enigma

BRAZILIAN JOURNAL OF INFORMATION SECURITY AND CRYPTOGRAPHY

VOLUME 1 - ISSUE 1 - Brasília — Brazil - 2014

RENASIC
Rede Nacional em Segurança da Informação e Criptografia

National Network of Information Security and Cryptography

DCT
Science and Technology Department

Brazilian Army

CDCIBER
Cyber Defense Center

Website of the ENIGMA Journal
http://cdciber.enigmajournal.unb.br

ENIGMA — Brazilian Journal of Information Security and Cryptography

Eduardo Takeo Ueda

Mirela Sechi Moretti Annoni Notare
Rafael Timóteo de Sousa Júnior

VOLUME 1

ISSUE 1

Brasília — Brazil
2014

ENIGMA — Brazilian Journal of Information Security and Cryptography

Production
Matheus Barbosa, Graphic Designer of Logo
Marcílio Costa Bezerra, Publication Designer
Aristides Sebastião Lopes Carneiro, Linguistic Reviewer

SUMMARY

Accepted Papers

Invited Papers

Best Paper of CIBSI 2013

Best Paper of TIBETS 2013

ENIGMA – Brazilian Journal of Information Security and Cryptography Volume 1 Issue 1 September 2014

E. T. Ueda, *Editor in Chief,* M. S. M. A. Notare, *Associate Editor in Chief,* and R. T. de Sousa Júnior, *Associate Editor in Chief*

***Abstract*— This is the first issue of Volume 1 of ENIGMA – Brazilian Journal of Information Security and Cryptography. Papers submissions were accepted in English, Portuguese and Spanish. In this first issue, 10 papers are published, of which 4 were peer-reviewed while the other 6 were invited and reviwed by the editorial board of the journal. In addition, 2 of the 6 invited papers are Best Papers from conferences CIBSI'2013 and TIBETS'2013 respectively.**

***Keywords*— Brazilian Journal, Cryptography, Information Security.**

I. INTRODUCTION

ENIGMA – Brazilian Journal of Information Security and Cryptography – is a technical-scientific publication that aims at discussing theoretical aspect contributions and practical applications results in information security, cryptography and cyber defense as well as fundamental subjects in support of those issues.

The choice of the name ENIGMA for this publication is related to the ENIGMA cryptography machine. However, the main reason for this choice is to pay tribute to the mathematician and computer scientist Alan Mathison Turing (1912-1954), considered one of the leading scientist in the history of computing. The world as we all know today would probably be very different it were not Turing's scientific contributions to humanity.

This journal is directed to academia researchers, industry professionals, members of government and military organizations, and all people that have interest in the area of information security and cryptography in order to disseminate and share their new technologies, scientific discoveries and research contributions.

The creation of this periodical is due the necessity to solve a gap represented by the lack of a technical-scientific brazilian journal that emphasizes information security and cryptography. In this manner, ENIGMA – Brazilian Journal of Information Security and Cryptography – must provide this demand, publishing papers of high quality within the international state-of-the-art.

Therefore, ENIGMA – Brazilian Journal of Information Security and Cryptography – will fulfill this demand, and will publish state-of-art and original research papers and timely review articles on the theory, design, and evaluation of all aspects of information, network and system security.

II. ABOUT VOLUME 1, ISSUE 1 OF ENIGMA

In this first issue of Volume 1 of ENIGMA – Brazilian Journal of Information Security and Cryptography – 10 papers were published, and in this section we briefly describe the contribution of each of these papers.

The first paper entitled "*Unconditionally Secure Quantum Communications via Decoherence-Free Subspaces*" by E. B. Guedes and F. M. de Assis, shows how to use decoherence-free subspaces over collective-noise quantum channels to convey classical information in perfect secrecy. The results obtained show how secure communication protocols can be simplified while reducing significantly the communication overhead.

The second paper entitled "*Revocation of User Certificates in a Military Ad Hoc Network*" by J. Jormakka and H. Jormakka, presents a scheme for revoking certificates in a medium-small size semi ad-hoc military network. Note that the solution can also be used in the civilian applications, such as police and crisis management, among others. It describes the functionalities of a protocol to handle certificates, a set of policy rules in a node for handling certificates and an analysis how the proposed mechanisms can mitigate attacks on the certificate revocation solution.

The paper "*Synthetic Steganographic Series and Finance*" by P. C. Ritchey and V. J. Rego, provide a comprehensive methodology that enables an agent to embed secret messages in public data that is sent or broadcasted to a receiving agent. The experiments have shown that one can develop a relatively sophisticated and practical secret-key stego-systems for a variety of applications including financial market based applications.

The paper "*Securing Automation Systems against Malware Intrusion*" by R. Fitz and W. A. Halang, focuses on the fact that computers employed for automation and control purposes are today more and more connected to networks, and thereby could be endangered by malware. As such, new architectures for their hardware and software as presented in this paper and proven to be necessary to solve the security problem due to their intrinsic properties.

The paper entitled "*Isomorphism Theorem and Cryptology*" by R. L. de Carvalho and F. L. de Mello, presents a theory of computational study based on recursive functions computability and presents innovative parallel mechanism relevant to enhance the performance of cryptography schemes. The main issue, as discussed in this paper, is closley related to the Isomorphism Theorem which supports the Church-Turing

E. T. Ueda, Senac University Center of São Paulo, edutakeo@gmail.com

M. S. M. A. Notare, IEEE Latin America Transactions Editor in Chief, IEEE South Brazil, mirela@ieee.org

R. T. de Sousa Júnior, University of Brasilia, desousa@unb.br

thesis and provides a connection between cryptology and linguistics.

The paper entitled "*Harnessing Nature's Randomness: Physical Random Number Generator*" by G. A. Barbosa, presents some guidelines for construction of a fast (telecommunication speed) Physical Random Number Generator. It discusses the fundamental physical elements involved, technicalities of signal recording, its limitations, and the final bit extraction The need for randomness tests is emphasized and the impossibility of guaranteeing true randomness of a finite sequence is discussed.

The paper "*QC-MDPC McEliece: an Optimized Implementation of a New McEliece Variant*" by H. O. Martins and A. C. A. Nascimento, presents the implementation of an optimized version of a McEliece variant. The McEliece cryptosystem is an example of code-based cryptography which is an alternative to the most popular and commercial cryptosystems nowadays as it is believed to be immune to quantum computing. It has simple and uses fast algorithms. Its main drawback is the size of the keys it has to deal with. By substituting the Goppa codes of the McEliece original proposalby LDPC and MDPC codes it's possible to achieve much smaller keys.

The paper "*Securing Web Applications: Techniques and Challenges*" by M. Vieira, discusses key techniques for security testing and assessment, providing the basis for understanding existing research challenges on developing and deploying secure web applications. The paper highlighted several research challenges in an attempt to motivate further research in these topics. The paper did not intend to provide a comprehensive survey. However, it does focus on key promising aspects in which research is need, and can be applied in the context of large-scale software based industry.

The ninth paper entitled "*Design of a Set of Software Tools for Side-Channel Attacks*" by A. F. Rodrígues et al, is the Best Paper of CIBSI'2013 (Congreso Iberoamericano de Seguridad Informática). In this paper, the authors present the first experimental results of a set of software tools for side channels attacks on cryptographic devices. The authors discuss the main types of attack, with an emphasis on attack called for an analysis of power consumption.

The last paper entitled "*Content related to Computing Security on Computer Engineering Degree according to International Professional Certificates*" by D. G. Rosado et al, is an extension of the best selectd papers of TIBETS'2013 (Taller Iberoamericano de Enseñanza e Innovación Educativa en Seguridad de la Información). This paper establishes a transverse guide for implementing information security content for courses and modules in the area of informatics or computer. The authors argue that basic knowledge of information security should be taught to students from the beginning of their training at university or college. In addition, the integrated content in the curriculum of the institutions should be based on professional certifications to prepare students for the industry.

III. CONCLUSION

ENIGMA – Brazilian Journal of Information Security and Cryptography – is a young publication but the beginning follow the best practices adopted by IEEE Transactions publications. It is hoped that soon this journal will become an icon of reference among the leading international publication dedicated to information security and cryptography.

With the creation of this journal the Brazil a considerable step toward the future, because ENIGMA journal is an important tool for communication and integration of knowledge between universities, research centers, industries, government or military institutions around the world. Moreover, threats to security and privacy of information are the enemy of any nation, which justifies this creation of this ENIGMA journal, indeed a unique initiatives for Brazil.

ACKNOWLEDGEMENTS

Initially, we would like to thank all the authors who contributed with their papers for the first issue of the ENIGMA journal, this publication would not exist if not for the dedication to their research. We must also thank all reviewers who worked very hard and in a timely fashion, so that we could select high quality of papers. We are grateful to National Network of Information Security and Cryptography (RENASIC) and the Cyber Defense Center (CDCiber) of the Defense Minister of Brazil for their support in the creation of this journal, and the University of Brasilia (UnB) for providing space on one of their servers to host the official website of this journal. Moreover, we wish to thank Coronel Luiz Carlos Rodrigues Pereira and Major Luciano de Oliveira for their support and in believing in this journal as well as their continuous help to get the printed version of this first issue. Last but not least, thanks to Itamar Annoni Notare for his hard work and for his assistance in the revision process as well as on the formatting this journal.

Eduardo Takeo Ueda received the Ph.D. degree in Electrical Engineering in 2012, and MSc degree in Computer Science in 2007, both from University of São Paulo (USP). He also holds a Mathematics degree by the São Paulo State University (UNESP), year 2000. His research interest includes topics of Cryptographic Algorithms and Protocols, Models of Access Control, and Computational Trust and Reputation. He has been committee member in conferences and reviewer of scientifics journals. Currently, he is Professor in Senac University Center of São Paulo, Master's Thesis Advisor in Institute for Technological Research of São Paulo, member of the Brazilian Computer Society (SBC), member of National Network of Information Security and Cryptography (RENASIC), and Editor in Chief of ENIGMA – Brazilian Journal of Information Security and Cryptography. http://lattes.cnpq.br/8367973725203446.

Mirela Sechi Moretti Annoni Notare received her Ph.D. and MSc degrees from the Federal University of Santa Catarina (UFSC) and a BSc degree from Passo Fundo University – all the three degrees in Computer Science. Her main research of interest focuses on the proposition of security management solutions for Wireless, Mobile, Sensor and Ad-Hoc Networks. Dra. Mirela Notare published widely in these areas. She also received several awards and citations, such as National Award for Telecommunication Software, British Library, TV Globo, INRIA and Elsevier Science. She served as General Co-chair for the I2TS (International Information and Telecommunication Technologies Symposium) and Program Co-Chair for the IEEE MobiWac (Mobility and Wireless Access Workshop) and IEEE ISCC. She has been a committee member in several scientific conferences, including ACM MSWiM, IEEE/ACM ANSS, IEEE ICC, IEEE IPDPS/WMAN, IEEE/SBC SSI, and IEEE Globecom/Ad-Hoc, Sensor and Mesh Networking Symposium. She has been Guest Editor for several international journals, such as JOIN (The International Journal of Interconnection Networks), IJWMC (Journal of Wireless and Mobile Computing), JBCS (Journal of Brazilian Computer Society), Elsevier ScienceJPDC (The International Journal of Parallel and Distributed Computing), Wiley & Sons Journal of Wireless Communications & Mobile Computing, and Wiley InterScience Journal Concurrency & Computation: Practice & Experience. She has some Books and Chapters – Protocol Engineering with LOTOS/ISO (UFSC) and Solutions to Parallel and Distributed Computing Problems (Wiley Inter Science), for instance. She is the current Editor in Chief of IEEE Latin America Transactions magazine and Associate Editor in Chief of ENIGMA – Brazilian Journal of Information Security and Cryptography. She is the founding and president of STS Co, a senior member (19 years) of IEEE, and member of SBrT and SBC societies. http://lattes.cnpq.br/8224632340074096.

Rafael Timóteo de Sousa Júnior, was born in Campina Grande – PB, Brazil, on June 24, 1961. He graduated in Electrical Engineering, from the Federal University of Paraíba – UFPB, Campina Grande – PB, Brazil, 1984, and got his Doctorate Degree in Telecommunications, from the University of Rennes 1, Rennes, France, 1988. He worked as a software and network engineer in the private sector from 1989 to 1996. Since 1996, He is a Network Engineering Professor in the Electrical Engineering Department, at the University of Brasília, Brazil. From 2006 to 2007, supported by the Brazilian R&D Agency CNPq, on leave from the University of Brasília, He took a sabbatical year in the Group for the Security of Information Systems and Networks, at Ecole Superiéure d´Electricité, Rennes, France. He is a member of the Post-Graduate Program on Electrical Engineering (PPGEE) and supervises the Decision Technologies Laboratory (LATITUDE) of the University of Brasília. His field of study is distributed systems and network management and security. http://lattes.cnpq.br/3196088341529197

Unconditionally Secure Quantum Communications Via Decoherence-Free Subspaces

E. B. Guedes and F. M. de Assis

***Abstract*— We show how to use decoherence-free subspaces over collective-noise quantum channels to convey classical information in perfect secrecy. We argue that codes defined over decoherence-free subspaces are codes for quantum wiretap channels in which the gain of information by a non-authorized third part is zero. We also show that if some symmetry conditions are guaranteed, the maximum rate on which such secret communications take place is equal to the ordinary capacity of a quantum channel to convey classical information. As a consequence of these results, we show how some protocols for secure communication can be simplified, reducing significantly the number of communications performed.**

***Keywords*— Decoherence-free subspaces, Quantum wiretap channels, Unconditional Security.**

I. INTRODUÇÃO

PREVENIR erros na informação quântica é um dos principais objetivos da Teoria Quântica da Informação. Erros surgem do acoplamento entre um sistema de interesse o ambiente, em função da subsequente *descoerência* induzida por este acoplamento. Considerando a natureza frágil dos sistemas quânticos, a descoerência é tida como o principal obstáculo na transmissão de informação coerente [1].

No contexto das Comunicações Quânticas, a descoerência é responsável pelo *vazamento* da informação para o ambiente em um canal quântico ruidoso. Se mensagens secretas são transmitidas por este canal, pelo menos parte delas pode ser capturadas por um receptor não-autorizado, aqui chamado de *espião*. Esta situação é indesejada e, em um cenário criptográfico, deve ser evitada.

Cai et al. [2] e Devetak [3] modelaram este cenário por meio dos chamados *canais wiretap quânticos*. Eles também estabeleceram as condições para realizar a troca de informações clássicas por canais quânticos sem que o conteúdo das mensagens fosse descoberto por um espião. Nesta formulação, apenas são considerados adequados códigos que minimizem a probabilidade de erro de decodificação entre os participantes legítimos ao passo que maximizem a equivocação de um espião. Apesar disso, a taxa máxima em que estas comunicações secretas podem acontecer, a chamada *capacidade quântica de sigilo*, é usualmente menor que a capacidade ordinária para envio de informação clássica neste mesmo canal.

Para minimizar a descoerência, diversos métodos foram propostos, tais como códigos corretores de erros quânticos (QECC – *Quantum Error-Correcting Codes*), desacoplamento dinâmico, subespaços livres de descoerência (DFS – *Decoherence-Free Subspaces*), dentre outros [4]. Em se tratando dos DFS, em particular, se os operadores de erro que afetam os qubits possuírem algumas simetrias, então estes qubits irão sofrer o mesmo tipo de erro ao passarem pelo canal quântico. Em alguns casos, isto fará com que determinados estados sejam invariantes ao erro, significando que a descoerência não ocorre em determinados subespaços [5]. Desta maneira, verifica-se um potencial no uso destes subespaços para a construção de códigos que minimizem o vazamento da informação para o ambiente.

Nos dias atuais, alguns trabalhos na literatura já exploram o potencial dos DFS nas Comunicações Quânticas. Estes trabalhos consistem de protocolos contra certo tipo de ruído coletivo (a exemplo de rotação, defasamento, amortecimento de amplitude) e consideram o uso de DFS pequenos (com dois ou três qubits, por exemplo) [6]-[10]. Até mesmo realizações experimentais já foram construídas objetivando o processamento da informação quântica [11]-[14]. Na perspectiva destes trabalhos, a proteção da informação significa evitar a perda de coerência, mantendo a fidelidade dos estados quânticos.

Neste artigo, serão investigadas consequências mais gerais do uso de DFS em Comunicações Quânticas. Em particular, considerando a perspectiva da *troca segura de mensagens*. Para tanto, será apresentada uma definição formal de canais quânticos que satisfazem aos critérios de simetria para a existência de DFS, posteriormente serão definidos códigos sobre estes subespaços e, por fim, serão estabelecidas as condições para a realização de comunicações sigilosas.

A partir de uma análise formal realizada, foi possível concluir que os códigos definidos sobre os DFS também são códigos adequados para os canais *wiretap* quânticos. Isto significa que a utilização de DFS possibilita a realização de comunicações quânticas incondicionalmente seguras. Mais além, foi verificado que a capacidade de sigilo neste cenário iguala-se à capacidade para envio de informação clássica ordinária. Este é um caso particular em que a capacidade de sigilo é máxima.

Após a apresentação destes resultados, serão exploradas algumas implicações resultantes em determinados protocolos para comunicação quântica segura direta e para comunicação quântica segura determinística. Alguns autores apresentaram estratégias para comunicações seguras sobre canais quânticos com ruído coletivo via DFS, mas o esforço requerido por alguns destes protocolos para checagem de espionagem aumenta significativamente o número de operações a serem implementadas, bem como o número de qubits trocados. Em

E. B. Guedes, Escola Superior de Tecnologia, Universidade do Estado do Amazonas, Manaus, Amazonas Brasil, elloaguedes@gmail.com

F. M. de Assis, Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Campina Grande, Paraíba, Brasil, fmarassis@gmail.com

face dos novos resultados sobre segurança incondicional e DFS, serão sugeridas simplificações nestes protocolos que diminuem a complexidade de implementá-los e também que reduzem substancialmente o número de comunicações realizadas.

O artigo está organizado como segue. As condições para privacidade quântica, estabelecidas por Schumacher e Westmoreland [15], serão apresentadas na Seção II. Os conceitos dos canais *wiretap* quânticos serão recapitulados na Seção III. Os fundamentos em DFS serão introduzidos na Seção IV. As contribuições sobre o uso de DFS para a realização de comunicações incondicionalmente seguras serão apresentadas na Seção V. Na seção VI, será mostrado um exemplo detalhado de como enviar informação secreta utilizando DFS. Os impactos dos resultados obtidos na simplificação de alguns protocolos existentes serão mostrados na Seção VII. Por fim, as considerações finais serão apresentadas na Seção VIII.

II. PRIVACIDADE QUÂNTICA

Suponha que um emissor (Alice) prepare um sistema quântico B em um estado inicial ρ. O objetivo de Alice é enviá-lo a um receptor (Bob) por meio de um canal quântico ruidoso, denotado pelo superoperador $\mathcal{E}^B$. Desta maneira, o estado recebido por Bob é $\rho_{\text{Bob}} = \mathcal{E}^B(\rho)$.

Devido à presença do ruído, para prover uma descrição unitária da evolução de ρ_{Bob} ao longo do canal, é necessário considerar a interação com o ambiente, que é assumido iniciar em um estado puro $|0_E\rangle$. Neste caso, o superoperador é dado por

$$\mathcal{E}^B(\rho) \quad = \quad Tr_E U^{BE} \left(\rho \otimes |0_\text{E}\rangle \langle 0_\text{E}|\right) U^{BE\dagger} \qquad (1)$$

em que U^{BE} representa a operação unitária de interação.

A *troca de entropia*, denotada por S_e, é definida como uma medida da informação trocada entre o sistema B e o ambiente E durante o período de interação. Considerando que o ambiente inicia em um estado puro, a troca de entropia é dada por $S_e = S(\rho_E)$, em que ρ_E é o estado final do ambiente. A troca de entropia é determinada inteiramente pelo estado inicial ρ de B e pela dinâmica do superoperador $\mathcal{E}^B$, isto significa que a troca de entropia é uma propriedade "intrínseca" ao sistema B e à sua dinâmica [15].

Suponha que Alice esteja usando o canal quântico para enviar informações clássicas para Bob. Alice então prepara o sistema quântico B em um dos possíveis estados ρ_k com probabilidades *a priori*p_k . O estado ρ enviado por Alice pode ser denotado por uma média

$$\rho = \sum_k p_k \rho_k \qquad (2)$$

Bob obtém o k-ésimo estado como sendo $\rho_{\text{Bob},k} = \mathcal{E}^B(\rho_k)$. Uma vez que $\mathcal{E}^B$ é linear, a média do estado recebido por Bob é

$$\begin{aligned} \rho_{\text{Bob}} &= \sum_k p_k \cdot \mathcal{E}^B(\rho_k) && (3) \\ &= \mathcal{E}^B(\rho) && (4) \end{aligned}$$

Para decodificar a mensagem recebida, Bob realiza uma medição utilizando algum *observável de decodificação*. A quantidade de informação clássica transmitida de Alice para Bob, denotada por $\mathbf{H}_{\text{Bob}}$, é governada pela quantidade de Holevo χ^{Bob}, definida como

$$\chi^{\text{Bob}} = S(\rho_{\text{Bob}}) - \sum_k p_k S(\rho_{\text{Bob},k}) \qquad (5)$$

Algumas considerações sobre a quantidade de Holevo neste cenário devem ser mencionadas: (*i*) $\mathbf{H}_{\text{Bob}} \leq \chi^{\text{Bob}}$ independente do observável de decodificação escolhido; e (*ii*) $\mathbf{H}_{\text{Bob}}$ pode ser arbitrariamente próxima de χ^{Bob} por meio de uma escolha adequada de um código e de um observável de decodificação. Neste caso, χ^{Bob} representa um limitante superior para a informação clássica transmitida de Alice para Bob.

Ao considerar os fins criptográficos do canal, então é estabelecido que uma espiã (Eve) deve ter acesso a alguma parte ou a todo o ambiente E com o qual B interage. O superoperador de evolução $\mathcal{E}^B$ descreve todos os efeitos da espiã no canal ou, em outras palavras, todos os esforços para a espionagem de Alice e Bob estão contidos no operador de interação U^{BE}. Desta maneira, a informação acessível à Eve, denotada por $\mathbf{H}_{\text{Eve}}$, será limitada por

$$\chi^{\text{Eve}} = S(\rho_{\text{Eve}}) + \sum_k p_k S(\rho_{\text{Eve},k}) \qquad (6)$$

A desigualdade $\mathbf{H}_{\text{Eve}} \leq \chi^{\text{Eve}}$ é verdadeira quer Eve tenha acesso total ou não ao ambiente.

A *privacidade quântica* é definida como

$$P = \mathbf{H}_{\text{Bob}} - \mathbf{H}_{\text{Eve}} \qquad (7)$$

Alice e Bob desejam maximizar P ao máximo possível. Mas, eles também devem assumir que a espiã está adquirindo o máximo de informação disponível. A *privacidade garantida*, $P_G = \inf P$, é o ínfimo sobre todas as possíveis estratégias adoptadas por Eve. Uma vez que $\mathbf{H}_{\text{Eve}} \leq \chi^{\text{Eve}}$, então $P_G \geq \mathbf{H}_{\text{Bob}} - \chi^{\text{Eve}}$. Por outro lado, Alice e Bob desejam usar o canal de modo que tornem a privacidade garantida P_G tão grande quanto o possível. Seja $\mathcal{P} = \sup P_G$. O melhor esquema que Alice e Bob podem utilizar aproxima $\mathbf{H}_{\text{Bob}}$ de χ^{Bob}. Desta maneira, é possível denotar $\mathcal{P}$ como

$$\mathcal{P} = \chi^{\text{Bob}} - \chi^{\text{Eve}} \qquad (8)$$

Apesar da caracterização da privacidade quântica, é necessário estabelecer esquemas que descrevam como Alice e Bob devem proceder para estabelecer as propriedades necessárias para a realização de comunicações seguras mesmo na presença da espiã. Estes aspectos serão discutidos na próxima seção.

III. CANAIS *WIRETAP* QUÂNTICOS

Na tentativa de prover uma descrição das propriedades do canal para estabelecer comunicações secretas sem possibilitar a descoberta de informações por um espião, Cai et al. [2] e Devetak [3] simultaneamente definiram os *canais wiretap quânticos*, cuja formalização é apresentada a seguir.

Definição 1. *Um canal wiretap quântico sem memória é descrito por um par de superoperadores* $\mathcal{E}^B$ *e* $\mathcal{E}^E$ *de um espaço de Hilbert complexo* $\mathcal{H}$*. Quando Alice envia um estado quântico* ω *de* $\mathcal{H}^{\otimes n}$*, Bob recebe* $\mathcal{E}^{\otimes n,B}(\omega)$ *e Eve recebe* $\mathcal{E}^{\otimes n,E}(\omega)$ *, em que* n *é a dimensão do espaço de Hilbert de entrada.*

Os códigos utilizados pelos participantes legítimos da comunicação são caracterizados na Definição 2.

Definição 2. *Um conjunto de palavras código de comprimento* n $(n = dim(\mathcal{H}))$ *para um conjunto* $\mathcal{U}$ *de mensagens clássicas é um conjunto de estados de entrada rotulados por mensagens em* $\mathcal{U}$*,* $\Omega(\mathcal{U}) = \{\omega(u) : u \in \mathcal{U}\}$ *, e uma decodificação de medição de comprimento* n *com saída em* $\mathcal{U}$*, i.e., um conjunto de operadores* $\mathcal{D}_u$*,* $u \in \mathcal{U}$ *com* $\sum_{u\in\mathcal{U}} \mathcal{D}_u \leq \mathbb{1}$ *. O par* $(\Omega(\mathcal{U}), \{\mathcal{D}_u : u \in \mathcal{U}\})$ *é chamado de um código de comprimento* n *para o conjunto de mensagens* $\mathcal{U}$*. A taxa deste código é* $\frac{1}{n}\log|\mathcal{U}|$ *.*

De acordo com ambas as definições apresentadas, a Fig. 1 ilustra os procedimentos requeridos para o cenário quântico. Alice deve criar um estado $\omega(u)$ quando desejar enviar uma mensagem $u \in \mathcal{U}$ para Bob. Devido ao ruído, Bob recebe $\mathcal{E}^{\otimes n,B}(\omega(u)) = Tr_E\left[\mathcal{E}^{\otimes n}(\rho \otimes |0_E\rangle\langle 0_E|\right]$ e realiza a decodificação da mensagem original utilizando um POVM (*Positive Operator-Value Measurement*) $\{\mathcal{D}_u : u \in \mathcal{U}\}$, que resulta em uma estimativa u' para u. A espiã Eve, por sua vez, recebe o estado $\mathcal{E}^{\otimes n,E}(\omega(u)) = Tr_B\left[\mathcal{E}^{\otimes n}(\rho \otimes |0_E\rangle\langle 0_E|\right]$ e irá tentar obter o máximo possível de informação sobre a mensagem originalmente enviada por Alice. Para tanto, ela irá tentar construir um POVM baseando-se na tipicalidade dos estados que recebe do canal, seguindo uma estratégia apresentada em [2, Sec. 4].

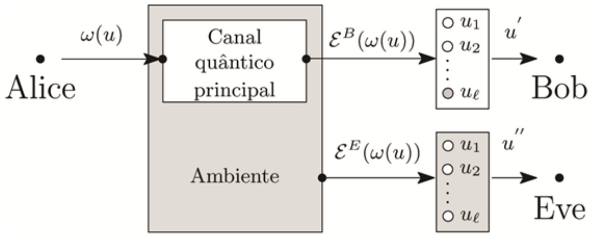


Figura 1. Idéia geral do canal *wiretap* quântico.

Entretanto, embora a estratégia de comunicação tenha sido detalhada, os argumentos de segurança sobre a troca de mensagens ainda não foram definidos. É necessário garantir uma baixa probabilidade de erro na decodificação entre os Alice e Bob, ao passo que Eve não aprende praticamente nada a respeito da mensagem secreta que passou pelo canal. A formalização destes dois requisitos é satisfeita pelos códigos da Definição 3.

Definição 3. *(Código Wiretap) Um código* $(\Omega(\mathcal{U}), \{\mathcal{D}_u : u \in \mathcal{U}\})$ *de comprimento* n *é chamado um código wiretap com parâmetros* $(n, |\mathcal{U}|, \lambda, \mu)$ *se, para* $\lambda, \mu > 0$

$$P_e = 1 - \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} Tr_E\left[\mathcal{E}^{\otimes n,B}(\omega(u))\mathcal{D}_u\right] \leq \lambda \tag{9}$$

e

$$\frac{1}{n}\left[S\left(\sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{U}|}\mathcal{E}^{\otimes n,E}(\omega(u))\right) - \sum_{u\in\mathcal{U}}\frac{1}{|\mathcal{U}|}S\left(\mathcal{E}^{\otimes n,E}(\omega(u))\right)\right] < \mu \tag{10}$$

em que $\frac{1}{n}\log|\mathcal{U}|$ *é a taxa deste código.*

Na definição de um código com parâmetros $(n, |\mathcal{U}|, \lambda, \mu)$, a Eq. (9) garante que a probabilidade média de erro na decodificação por Bob é menor que um parâmetro λ, e a Eq. (10) limita a informação media acessível por Eve, de tal maneira que esta não captura praticamente nada a respeito da mensagem secreta enviada por Alice.

Por fim, a *capacidade quântica de sigilo* é definida.

Definição 4. *(Capacidade Quântica de Sigilo [2]) A capacidade de sigilo de um canal quântico é o maior número real* C_S *tal que para todo* $\epsilon, \lambda, \mu > 0$ *e* n *suficientemente grande, existe um código* $(n, |\mathcal{U}|, \lambda, \mu)$ *com*

$$C_S < \frac{1}{n}\log|\mathcal{U}| + \epsilon \tag{11}$$

Apesar das definições anteriores assumirem mensagens uniformemente distribuídas, o seguinte teorema de [2, Sec. 5] sobre a capacidade quântica de sigilo é um resultado mais geral.

Teorema 1. *Para um canal wiretap quântico* $\mathcal{E}$ *como caracterizado na Definição 1, a capacidade quântica de sigilo satisfaz*

$$C_S(\mathcal{E}) \geq \max_{\{P\}}\left[\chi^{\text{Bob}} - \chi^{\text{Eve}}\right] \tag{12}$$

em que o máximo é tomado sobre todas as distribuições de probabilidade sobre $\mathcal{U}$*; e* χ^{Bob} *e* χ^{Eve} *são as quantidades de Holevo dadas nas Eqs. (5) e (6), respectivamente.*

A capacidade quântica de sigilo pode ser compreendida como a capacidade de um canal quântico para enviar informação clássica em sigilo absoluto. Esta capacidade é equivalente ao supremo da privacidade garantida definida na Eq. (8).

A capacidade de sigilo definida na Eq. (12) é o análogo quântico da capacidade de sigilo clássica proposta por Wyner [16]. É possível verificar algumas similaridades entre ambas as definições: as duas limitam a probabilidade de erro na decodificação e também a informação que deve ser acessível por um espião. Entretanto, o caso quântico utiliza-se de medidas de informação próprias deste domínio, tais como a entropia de von Neumann e a quantidade de Holevo. Uma característica particular da capacidade quântica de sigilo é que esta não possui uma caracterização de letra isolada, significando que a mesma não é computável por considerar todos os possíveis estados de entrada e todas as possíveis distribuições sobre eles [2], [3].

IV. SUBESPAÇOS LIVRES DE DESCOERÊNCIA

Devido à descoerência, um sistema quântico decai para um estado de baixa energia em função das perdas sofridas para o ambiente, tem sua fase desvanecida e, por fim, a informação que armazenava é perdida [17].

Seja um sistema quântico fechado composto por um *sistema de interesse* S e pelo *ambiente* E. O hamiltoniano que descreve este sistema é definido como segue

$$\mathbb{H} = \mathbb{H}_S \otimes \mathbb{1}_E + \mathbb{1}_S \otimes \mathbb{H}_E + \mathbb{H}_{SE} \tag{13}$$

em que $\mathbb{1}$ denota o operador identidade; e $\mathbb{H}_S$, $\mathbb{H}_E$ e $\mathbb{H}_{SE}$ denotam os hamiltonianos do sistema de interesse, do ambiente e da interação sistema-ambiente, respectivamente.

Para prevenir erros, seria ideal que $\mathbb{H}_{SE}$ fosse igual a zero, indicando que o sistema e o ambiente estão desacoplados e evoluem independentemente e unitariamente de acordo com seus respectivos hamiltonianos $\mathbb{H}_S$ e $\mathbb{H}_E$ [5]. Porém, em cenários práticos, esta situação ideal não é possível, uma vez que limitações tecnológicas impedem a construção de um sistema completamente livre de ruído. Assim, após isolar o sistema da melhor maneira possível, o adequado é vislumbrar objetivos realísticos para identificar e corrigir erros quando eles ocorrerem, evitar o ruído quando possível, ou então até mesmo tentar suprimir o ruído do sistema [4].

Se algumas simetrias existirem na interação entre sistema e ambiente, então é possível encontrar "locais seguros" no espaço de Hilbert do sistema que não sofrem os efeitos da descoerência. Seja $\{A_i(t)\}$ um conjunto de operadores na representação da soma de operadores (OSR – *Operator-Sum Representation*) correspondendo à evolução do sistema. Diz-se que a matriz densidade ρ_S é *invariante* perante os operadores OSR $\{A_i(t)\}$ se $A_i(t)\rho_S A_i^\dagger(t) = \rho_S$. Levando isto em consideração, é possível definir os subespaços livres de descoerência, cujos estados são invariantes apesar da existência de um acoplamento não-trivial entre sistema de interesse e ambiente.

Definição 5. *(Subespaço Livre de Descoerência [16]) Um subespaço $\tilde{\mathcal{H}}$ de um espaço de Hilbert $\mathcal{H}$ é chamado DFS em relação ao acoplamento entre sistema e ambiente se cada estado puro deste subespaço é invariante perante a correspondente evolução OSR para qualquer condição inicial do ambiente, isto é*

$$\sum_i A_i(t)|\tilde{k}\rangle\langle\tilde{k}|A_i^\dagger(t) = |\tilde{k}\rangle\langle\tilde{k}|, \forall|\tilde{k}\rangle\langle\tilde{k}| \in \tilde{\mathcal{H}}, \forall\rho_E(0) \tag{14}$$

Apesar da definição de DFS apresentada ter sido feita em termos de estados puros, um estado emaranhado que tenha suporte apenas em estados puros de um DFS também será invariante e, portanto, protegido da descoerência [18].

Sistemas quânticos definidos sobre DFS são totalmente desacoplados do ambiente e, por esta razão, completamente imunes aos efeitos da descoerência. Códigos quânticos construídos a partir de estados de um DFS são classificados como *códigos quânticos de prevenção de erros* (QEAC – *Quantum error-avoiding codes*) e as tarefas de perturbação e recuperação nestes códigos são triviais [19].

Seja o hamiltoniano da interação sistema-ambiente dado por $\mathbb{H}_{SE} = \sum_j \mathbf{S}_j \otimes \mathbf{E}_j$., em que $\mathbf{S}_j$ e $\mathbf{E}_j$ são os operadores do sistema e do ambiente, respectivamente. Considera-se que os operadores do ambiente $\mathbf{E}_j$ são linearmente independentes. As simetrias requeridas para a existência de DFS são descritas no teorema a seguir. Para um prova detalhada ou diferentes formulações ver [5, Sec. 5].

Teorema 2. *(Condições para DFS) Um subespaço $\tilde{\mathcal{H}}$ é um DFS se, e somente se, os operadores do sistema $\mathbf{S}_j$ atuarem proporcionalmente à identidade neste subespaço*

$$\mathbf{S}_j|\tilde{k}\rangle = c_j|\tilde{k}\rangle \qquad \forall j, |\tilde{k}\rangle \in \tilde{\mathcal{H}} \tag{15}$$

Na prática, identificar uma simetria útil e tirar proveito dela pode ser uma tarefa bastante difícil. Isto acontece porque é necessário (*i*) identificar a simetria; (*ii*) encontrar estados que sejam invariantes a interação e, por fim, (*iii*) construir, se possível, operações no sistema que preservem as simetrias necessárias. Apesar destas dificuldades, quando comparados aos QECCs, por exemplo, os DFS possuem algumas vantagens, a exemplo de frequentemente requererem menos qubits físicos para representar um qubit lógico e também de não demandarem uma repetida identificação e correção de erros [4].

Em se tratando dos DFS como QEACs, eles podem ser contrastados com os QECCs em alguns aspectos. Enquanto os QECCs são projetados para corrigir erros após a sua ocorrência, QEACs não possuem a habilidade de corrigir erros, uma vez que os previnem; QECCs adotados em cenários práticos pertencem a classe dos códigos não-degenerados, enquanto QEACs são códigos altamente degenerados; QEACs possuem distância infinita, enquanto os QECCs não-degenerados possuem distância infinita. Em particular, se a degenerescência atinge o máximo, um QECC se reduz a um QEAC, o que ilustra a circunstância em que um tipo de código torna-se equivalente ao outro [19].

A ausência de descoerência em DFS têm se mostrado de grande importância para implementações de memórias quânticas e algoritmos quânticos. Outras aplicações incluem codificação da informação em pontos quânticos, dissipação coletiva, redução de ruído, dentre outros [4], [5].

V. DFS EM COMUNICAÇÕES SEGURAS

A partir de agora serão consideradas as aplicações dos DFS nas Comunicações Quânticas. Será considerado como referência o modelo de *canais quânticos com ruído coletivo*, i.e., um modelo de canal no qual os qubits se acoplam identicamente ao mesmo ambiente, ao passo que sofrem defasamento e dissipação [20]. Apesar de não ser um modelo abrangente, este caso especial traz à tona algumas consequências particulares do uso de DFS em comunicações quânticas. O foco a ser considerado, em particular, será nos aspectos da *troca segura de mensagens*.

Considera-se o caso em que Alice quer enviar mensagens clássicas secretas para Bob por um canal quântico. Estas mensagens devem ser protegidas da espiã Eve, que tem acesso total ao ambiente. O canal entre Alice e Bob possui um subespaço livre de descoerência que será utilizado para codificar as mensagens secretas. A definição a seguir caracteriza este canal quântico.

Definição 6. *(Canal Wiretap Quântico com Ruído Coletivo) Um canal wiretap quântico com ruído coletivo $\mathcal{E}$ é um canal como na Definição 1, mas cuja decomposição de Kraus $\{A_i\}$ satisfaz o Teorema 2.*

Nesta definição, uma vez que $\{A_i\}$ satisfaz ao Teorema 2, então o canal $\mathcal{E}$ possui um DFS $\tilde{\mathcal{H}}$. Quando Alice deseja enviar um estado para Bob, ela o faz pelo canal quântico e este estado interage com o ambiente. Bob recebe o estado resultante do traço parcial sobre o ambiente. A espiã Eve, por sua vez, captura o que vazou para o ambiente.

Sem perda de generalidade, será considerado aqui que o ambiente inicia em um estado puro $|0_{\mathrm{E}}\rangle\langle 0_{\mathrm{E}}|$. Esta é uma hipótese plausível, pois sempre é possível imaginar que um ambiente "local" em um estado misto é apenas parte de um sistema maior em um estado puro emaranhado [15].

O passo seguinte é definir o QEAC sobre $\tilde{\mathcal{H}}$ para codificar as mensagens entre Alice e Bob.

Definição 7. **S***eja $\tilde{\mathcal{H}}$ um DFS gerado por um conjunto de autovetores $\{|\tilde{k}\rangle\}$, i.e., $\tilde{\mathcal{H}} = Span[\{|\tilde{k}\rangle\}]$. Um conjunto de palavras código de comprimento n $(n = dim(\tilde{\mathcal{H}}))$ para um conjunto $\mathcal{U}$ de mensagens clássicas é um conjunto de estados de entrada rotulados por mensagens em $\mathcal{U}$, $\tilde{K}(\mathcal{U}) = \{\tilde{k}(u) : u \in \mathcal{U}\} \subseteq \tilde{\mathcal{H}}$, e um processo de medição trivial composto por um conjunto POVM $\tilde{\mathcal{D}}_u$, $u \in \mathcal{U}$ com a restrição $\sum_{u\in\mathcal{U}} \tilde{\mathcal{D}}_u \leq \mathbb{1}$. O par $(\tilde{K}(\mathcal{U}), \{\tilde{\mathcal{D}}_u : u \in \mathcal{U}\})$ é chamado um QEAC de comprimento n para o conjunto de mensagens $\mathcal{U}$. A taxa deste código é $\frac{1}{n}\log|\mathcal{U}|$*

Utilizando o código definido, se Alice deseja enviar a mensagem u para Bob, ela deve codifica-la no QEAC definido sobre $\tilde{\mathcal{H}}$, obtendo $\tilde{k}(u)$. Quando ela envia o estado resultante pelo canal, este interage com o ambiente. Bob então recebe $\rho_{\mathrm{Bob}}(\tilde{k}(u))$ e Eve recebe $\rho_{\mathrm{Eve}}(\tilde{k}(u))$, os quais são dados por

$$\begin{aligned} \rho_{\mathrm{Bob}}(\tilde{k}(u)) &= \mathrm{Tr}_{\mathrm{E}}\left[\mathcal{E}^{\otimes n}(\tilde{k}(u) \otimes |0_E\rangle\langle 0_E|)\right] && (16) \\ \rho_{\mathrm{Eve}}(\tilde{k}(u)) &= \mathrm{Tr}_{\mathrm{B}}\left[\mathcal{E}^{\otimes n}(\tilde{k}(u) \otimes |0_E\rangle\langle 0_E|)\right] && (17) \end{aligned}$$

Uma vez que Alice utilizou um QEAC como na Definição 7, então a simetria dinâmica existente protegeu a informação quântica da interação com o ambiente. Isto significa que a evolução conjunta entre sistema e ambiente aconteceu de maneira desacoplada. Assim, o estado $\rho_{\mathrm{Bob}}(\tilde{k}(u))$ é

$$\begin{aligned} \rho_{\mathrm{Bob}}(\tilde{k}(u)) &= \mathrm{Tr}_{\mathrm{E}}\left[\mathcal{E}^{\otimes n}(\tilde{k}(u) \otimes |0_E\rangle\langle 0_E|)\right] && (18) \\ &= \mathrm{Tr}_{\mathrm{E}}\left[\sum_i A_i\left(\tilde{k}(u) \otimes |0_E\rangle\langle 0_E|\right)A_i^\dagger\right] && (19) \\ &= \mathrm{Tr}_{\mathrm{E}}\left[\tilde{k}(u) \otimes \rho_E\right] && (20) \\ &= \tilde{k}(u) && (21) \end{aligned}$$

em que a Eq. (20) deve-se à invariância dos estados do DFS perante os operadores OSR. Levando em consideração o hamiltoniano dado na Eq. (13) e o fato do sistema de interesse e do ambiente não terem interagido, este é o caso em que o ambiente sofreu apenas a ação de $\mathbb{H}_E$, indicando uma evolução unitária restrita ao ambiente. Isto significa que $\rho_{\mathrm{Eve}}(\tilde{k}(u)) = \rho_E$ é um estado puro.

O lema a seguir formaliza como o QEAC protege a informação transmitida pelo canal da atuação de um espião.

Lema 1. *Um QEAC como na Definição 7 sobre um canal wiretap quântico com ruído coletivo, como na Definição 6, é um código wiretap com parâmetros $(n, |\mathcal{U}|, \lambda, \mu)$.*

Prova. A prova é feita de maneira direta, mostrando como o QEAC satisfaz aos critérios das Eqs. (9) e (10).

Primeiro será analisada a probabilidade de erro na decodificação. Uma vez que $\tilde{k}(u)$ pertence a $\tilde{\mathcal{H}}$, sabe-se que este estado não interagiu com o ambiente. Então, $\rho_{\mathrm{Bob}} = \tilde{k}(u)$ como mostrado nas Eqs. (18)-(21). Verifica-se que o processo de decodificação é trivial e que a mensagem enviada por Alice pode ser perfeitamente recuperada, visto que há um operador de decodificação $\tilde{\mathcal{D}}_u$ para cada $u \in \mathcal{U}$. É possível concluir, portanto, que há uma probabilidade desprezível de erro na decodificação por Bob. Logo, o critério da Eq. (9) é satisfeito.

Prossegue-se para a análise do critério da Eq. (10). É interessante verificar que esta equação representa a média da informação acessível por Eve, a qual é limitada pela quantidade de Holevo definida na Eq. (6). A quantidade de Holevo será obtida primeiramente.

Apesar do estado final do ambiente ρ_E (vide Eq. (20)) não ser conhecido, o fato de Alice e Bob terem utilizado apenas estados de um DFS garantiu que o hamiltoniano de interação $\mathbb{H}_{SE}$ não governou a evolução conjunta entre sistema de interesse e ambiente. Ao contrário, é possível garantir que cada sistema evoluiu de maneira completamente unitária de acordo com seu próprio hamiltoniano, o que implica que o ambiente apenas sofreu a atuação de $\mathbb{H}_E$. No contexto em questão, isto significa que o ambiente terminou em um estado puro. Utilizando este resultado para calcular a quantidade de Holevo, tem-se

$$\begin{aligned}
\chi^{\text{Eve}} &= S(\rho_{\text{Eve}}(\tilde{k}(u))) - \sum_k p_k S(\rho_{\text{Eve},k}\tilde{k}(u)) && (22)\\
&= S(\rho_E) - \sum_k p_k S(\rho_{\text{Eve},k}\tilde{k}(u)) && (23)\\
&= 0 - \sum_k p_k S(\rho_{\text{Eve},k}\tilde{k}(u)) && (24)
\end{aligned}$$

Um fato conhecido sobre a quantidade de Holevo é que $\chi^{\text{Eve}} \geq 0$. Uma vez que $S(\rho) \geq 0$ para qualquer ρ, e que $p_k \geq 0$ para todo k, então este é o caso que o termo remanescente é igual a zero. Portanto, $\chi^{\text{Eve}} = 0$.

Dado que a quantidade de Holevo é um limitante superior para a informação acessível, este é o caso em que a Eq. (10) também é igual a zero. Este resultado significa que a quantidade de informação capturada por Eve não pôde reduzir a incerteza sobre a mensagem secreta u enviada de Alice para Bob – implicando em *sigilo absoluto*, um requisito essencial para códigos *wiretap*. Isto conclui a prova. ■

Outra medida de informação que enfatiza a ausência de interação entre sistema e ambiente é a troca de entropia, cuja medida é determinada inteiramente pelo estado inicial de B e pela dinâmica do canal [15]. Neste caso, esta medida é igual a $S_e = S(\rho_{\text{Eve}}(\tilde{k}(u))) = S(\rho_E) = 0$, uma vez que ρ_E é um estado puro. Assim é possível reforçar a conclusão que sistema e ambiente evoluíram de maneira completamente desacoplada.

Por fim, parte-se para a caracterização da capacidade de sigilo de um canal *wiretap* quântico com ruído coletivo.

Teorema 3. *A capacidade de sigilo de uma canal wiretap quântico com ruído coletivo $\mathcal{E}$, caracterizado na Definição 6, satisfaz*

$$C_{S,\text{DFS}}(\mathcal{E}) = \max_{\{P\}} \left[\chi^{\text{Bob}}\right] \qquad (25)$$

em que o máximo é tomado sobre todas as distribuições de probabilidade P sobre $\mathcal{U}$; e χ^{Bob} é a quantidade de Holevo dada na Eq. (5).

Prova. Seja um QEAC $(\tilde{K}(\mathcal{U}), \{\tilde{\mathcal{D}}_u : u \in \mathcal{U}\})$ utilizado no canal $\mathcal{E}$. Como visto no Lema 1, este é um código *wiretap*. Um fato verificado na prova deste lema foi que a quantidade de Holevo de Ev$\chi^{\text{Eve}} = 0$. Primeiro substitui-se este resultado na Eq. (12). A igualdade final é advinda como consequência do Teorema de Holevo-Schumacher-Westmoreland [21], que afirma que a taxa de um código deve ser limitada pela quantidade de Holevo. ■

Pode-se concluir, então, que é possível realizar comunicações quânticas com sigilo absoluto em canais quânticos espionados desde que os operadores de erro satisfaçam algumas simetrias. O critério de segurança incondicional é satisfeito, uma vez que $\chi^{\text{Eve}} = 0$.

A expressão resultante da capacidade de sigilo de um DFS possui relação com resultados apresentados por Schumacher e Westmoreland [15]. Estes autores argumentam que a habilidade de um canal quântico de enviar informação privada é pelo menos tão grande quanto a habilidade de enviar informação coerente. Uma vez que a informação codificada em um DFS não perde coerência, então a capacidade de enviar informação privada é maximizada, particularmente quando comparada a outros tipos de canais quânticos.

VI. Exemplo – Defasamento Coletivo

Para ilustrar os resultados descritos neste artigo, nesta seção será mostrado um exemplo detalhado do envio sigiloso de informação clássica por um canal quântico com defasamento coletivo $\mathcal{E}$. Neste modelo de canal, os qubits se acoplam ao ambiente de maneira simétrica ao passo que sofrem defasamento, definido como:

$$\begin{aligned}
|0\rangle &\rightarrow |0\rangle && (26)\\
|1\rangle &\rightarrow e^{\imath\phi}|1\rangle && (27)
\end{aligned}$$

Alice deseja enviar mensagens clássicas secretas para Bob, porém Eve espiona o canal com acesso total ao ambiente. Se há descoerência, então é possível que Eve adquira alguma informação acerca da mensagem secreta trocada entre Alice e Bob.

Para contornar os efeitos da descoerência, Alice e Bob podem tirar proveito de uma simetria existente no canal. Se eles codificarem as mensagens utilizando estados imunes à descoerência, então Eve não irá capturar nada a respeito das mensagens secretas. Para tanto, Alice e Bob utilizarão o seguinte esquema de codificação

$$\begin{aligned}
|0_L\rangle &= |01\rangle && (28)\\
|1_L\rangle &= |10\rangle && (29)
\end{aligned}$$

Um qubit pode, portanto, ser codificado como $|\psi_L\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$. É possível constatar que $|\psi_L\rangle$ não sofre os efeitos da descoerência ao passar pelo canal

$$\begin{aligned}
\mathcal{E}(|\psi_L\rangle) &= \mathcal{E}\left(\alpha|0_L\rangle + \beta|1_L\rangle\right) && (30)\\
&= \alpha e^{\imath\phi}|01\rangle + \beta e^{\imath\phi}|10\rangle && (31)\\
&= e^{\imath\phi}\left(\alpha|01\rangle + \beta|10\rangle\right) && (32)\\
&= e^{\imath\phi}|\psi_L\rangle && (33)\\
&= |\psi_L\rangle && (34)
\end{aligned}$$

porque o fator de fase global $e^{\imath\phi}$ adquirido durante o processo de defasamento não possui significância física. Isto significa que ambos os estados $|01\rangle$ e $|10\rangle$ pertencem a $\tilde{\mathcal{H}}$, um DFS do espaço de Hilbert $\mathcal{H}$ no canal quântico de defasamento coletivo.

Neste exemplo, as mensagens enviadas por Alice são binárias, logo $\tilde{K}(\mathcal{U}) = \{|01\rangle, |10\rangle\}$. Fazendo uso deste código, para enviar a mensagem u, Alice a codifica no estado correspondente $\tilde{k}(u)$, o qual será enviado pelo canal. Assume-se aqui que os bits 0 e 1 são equiprováveis e que o ambiente inicia no estado puro $|0_E\rangle\langle 0_E|$.

Dado que Alice utilizou estados do DFS para codificar mensagens para Bob, o sistema de interesse e o ambiente não interagiram. Como já provado anteriormente, Eve não captura qualquer informação a respeito da mensagem secreta.

Em relação a Bob, de acordo com o esquema caracterizado, o estado recebido $\rho_{\text{Bob}}(\tilde{k}(u))$ é dado por

$$\begin{aligned} \rho_{\text{Bob}}(\tilde{k}(u)) &= \text{Tr}_{\text{E}}\left[\mathcal{E}^{\otimes n}(\tilde{k}(u) \otimes |0_E\rangle \langle 0_E|)\right] & (35) \\ &= \tilde{k}(u) & (36) \end{aligned}$$

Para decodificar a mensagem recebida, Bob deve seguir as instruções do QEAC na Definição 7, devendo construir os projetores POVM $\tilde{\mathcal{D}}_0 = |01\rangle\langle 01|$ e $\tilde{\mathcal{D}}_1 = |10\rangle\langle 10|$.

A quantidade de Holevo de Bob, como definida na Eq. (5), assume o seguinte valor neste cenário:

$$\begin{aligned} \chi^{\text{Bob}} &= S\left(\rho_{\text{Bob}}\tilde{k}(u)\right) - \sum_u p_u S\left(\rho_{\text{Bob},u}\right) & (37) \\ &= S\left(\frac{1}{2}|01\rangle\langle 01| + \frac{1}{2}|10\rangle\langle 10|\right) - \frac{1}{2}S\left(\tilde{k}(0)\right) & (38) \\ & \quad -\frac{1}{2}S\left(\tilde{k}(1)\right) & (39) \\ &= 1 - 0 - 0 & (40) \\ &= 1 & (41) \end{aligned}$$

Utilizando este resultado na Eq. (25), é possível concluir que a capacidade quântica de sigilo para este cenário é igual a $C_{S,DFS}(\mathcal{E}) = 1$ bit por uso do canal. Este é um exemplo de como enviar mensagens secretas a uma taxa positiva utilizando DFS em um canal quântico ruidoso utilizando um procedimento de codificação-decodificação bastante simplificado.

VII. IMPACTOS EM PROTOCOLOS QSDC E DSQC

A Mecânica Quântica provê novas maneiras para realização de transmissão e processamento da informação. Em um contexto criptográfico, a *distribuição quântica de chaves* (QKD – *Quantum Key Distribution*) é uma das técnicas mais maduras atualmente, a qual possibilita a criação de chaves privadas clássicas com o intuito de permitir que duas partes realizem comunicações de forma segura [22, p. 586]. Estas chaves podem ser usadas para cifrar mensagens em esquemas de criptografias clássicos, tais como o *one-time pad*. Deste modo, percebe-se que há pelo menos duas transmissões em um protocolo QKD: a primeira delas via um canal quântico com o intuito de criar uma chave segura entre as partes; e a segunda, na qual a mensagem cifrada é transmitida. Muitos protocolos para QKD já foram propostos, inclusive com suas provas de segurança adequadamente estabelecidas [23].

Porém, em transmissões práticas, o ruído do canal não pode ser evitado completamente. Este ruído não apenas aumenta a taxa de erro no envio da mensagem, mas também pode dificultar a detecção de um espião num processo de checagem de segurança [10].

Recentemente, a *comunicação quântica segura direta* (QSDC – *Quantum Secure Direct Communication*) foi proposta como uma nova técnica de comunicação. Ela tem por objetivo transmitir mensagens secretas diretamente, sem o auxílio de chaves privadas nem de comunicações clássicas. Neste esquema, tem-se que a QKD e a transmissão clássica da mensagem cifrada são condensadas em uma única comunicação quântica. Por esta razão, considera-se que o QSDC como uma técnica completamente baseada na Mecânica Quântica [24].

Outra técnica que permite a comunicação quântica segura é intitulada *comunicação quântica determinística segura* (DSQC – *Deterministic Secure Quantum Communication*). Nesta técnica, a mensagem é enviada deterministicamente pelo canal quântico, mas pode ser deduzida apenas após uma transmissão de informação clássica [25]. De fato, a diferença fundamental entre QSDC e DSQC é esta necessidade de mais um envio de comunicação clássica [24].

Até os dias atuais, muitos protocolos para QSDC e DSQC já foram propostos na literatura, considerando o uso de diferentes recursos e métodos, tais como troca de emaranhamento [26], teleportação [25], [27], [28], *one-time pad* quântico [29], rearranjo da ordem de partículas [30], dentre outros. O *survey* de Long et al. [24] contempla desenvolvimentos recentes tanto sobre QSDC quanto DSQC.

O uso de canais com ruído coletivo também foi considerado na proposição de alguns protocolos de QSDC e DSQC em uma tentativa de prevenir o ruído. Tais protocolos exploram as simetrias existentes no DFS para transmitir informação. Decaimento de amplitude [9], rotação [6], [10] e defasamento [6] são modelos de canais com ruído coletivo que já foram considerados por estes protocolos. Porém, como provado na Seção V, codificar informação em um DFS habilita comunicação quântica incondicionalmente segura. Deste modo, uma questão que emerge é: existem modificações que podem ser feitas nestes protocolos visando uma simplificação ou aumento de eficiência? As subseções a seguir irão caracterizar estes protocolos de acordo com o tipo de canal e irão apresentar algumas sugestões nesta direção.

A. Canal de Decaimento de Amplitude Coletivo

O fenômeno de dissipação de energia ao transmitir um estado quântico é modelado pelo *canal de decaimento de amplitude coletivo*. Este canal possui a seguinte representação OSR

$$\mathcal{E}(\rho) = A_0 \rho A_0^\dagger + A_1 \rho A_1^\dagger \qquad (42)$$

em que os operadores A_0 e A_1 possuem a seguinte definição

$$A_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix} \qquad A_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix} \qquad (43)$$

em que γ denota a taxa de decaimento, que pode ser pensada como a probabilidade de perda de um fóton [22, p. 380].

Um protocolo QSDC sobre o canal de decaimento de amplitude coletivo foi proposto por Qin et al. [9]. Este protocolo faz uso de dois estados de um DFS definidos sobre este canal ($|0_L\rangle$ e $|1_L\rangle$) e também de dois outros estados baseados numa superposição deles ($|+_L\rangle$ e $|-_L\rangle$). Estes estados quânticos são

$$\begin{aligned} |0_L\rangle &= |00\rangle && (44) \\ |1_L\rangle &= \frac{|10\rangle - |01\rangle}{\sqrt{2}} && (45) \\ |+_L\rangle &= \frac{|0_L\rangle + |1_L\rangle}{\sqrt{2}} && (46) \\ |-_L\rangle &= \frac{|0_L\rangle - |1_L\rangle}{\sqrt{2}} && (47) \end{aligned}$$

O protocolo de Qin et al. é definido como segue:

1) Alice gera uma seqüência aleatória dos seguintes estados $\{|0_L\rangle, |1_L\rangle, |+_L\rangle, |-_L\rangle\}$ e os envia para Bob;
2) Bob escolhe alguns qubits para realizar uma checagem de espionagem. Ele os mede em uma das duas bases possíveis aleatoriamente escolhidas ($\{|0_L\rangle, |1_L\rangle\}$ ou $\{|+_L\rangle, |-_L\rangle\}$) e publica os resultados obtidos. Alice checa as saídas e julga quando existem espiões no canal. Bob realiza algumas operações nos qubits remanescentes, introduz alguns bits aleatórios, e os envia de volta para Alice;
3) Alice mede os qubits recebidos na mesma base que ela originalmente os preparou. A depender da relação entre as saídas obtidas e os estados originalmente preparados, Alice pode deterministicamente decodificar a mensagem enviada por Bob;
4) Bob declara a posição e os valores dos bits aleatórios. Alice julga a segurança e recupera a mensagem enviada.

Neste protocolo, o número de comunicações requerido para a troca da mensagem e também para a checagem de espionagem inclui redundância, bits aleatórios e também comunicações clássicas para divulgar a saída de determinadas medições. Vale mencionar também que a taxa deste protocolo é de 1 bit de informação por uso do canal, uma vez que um estado de dois qubits é utilizado.

Os estados $|0_L\rangle$ e $|1_L\rangle$ do DFS existente no canal também possibilitam o envio de 1 bit de informação por uso do canal, mas com a vantagem de que não é necessário realizar checagem de espionagem, pois os DFS habilitam segurança incondicional. Formalizando, Alice e Bob podem usar um QEAC $(\tilde{K}(\{0,1\} = \{\tilde{k}(0) = |0_L\rangle, \tilde{k}(1) = |1_L\rangle\}, \{\tilde{D}_0 = |0_L\rangle\langle 0_L|, \tilde{D}_1 = |1_L\rangle\langle 1_L|\})$ para realizar a troca de mensagens de forma segura.

Apesar da taxa resultante ser a mesma, o uso do QEAC sugerido provê uma redução significativa na Complexidade Comunicacional Quântica[1] deste protocolo. Na proposição original, se a mensagem possui tamanho m, então é evidente que mais de m bits e qubits precisam ser trocados para realizar a comunicação. Utilizando a simplificação apresentada, este número iguala-se exatamente a m, com a vantagem adicional de não ser necessário utilizar um canal clássico. Além disso, o processo de codificação-decodificação se torna menos complexo, o que reduz o número de portas quânticas requeridas para implementar este esquema.

B. Canal Quântico de Rotação Coletiva

Um canal quântico de rotação coletiva pode ser denotado como

$$\begin{aligned} |0\rangle &\rightarrow \cos\theta\,|0\rangle + \sin\theta\,|1\rangle && (48) \\ |1\rangle &\rightarrow -\sin\theta\,|0\rangle + \cos\theta\,|1\rangle && (49) \end{aligned}$$

em que θ denota o parâmetro de rotação o qual flutua com o tempo t. Dois estados imunes aos efeitos deste canal são estados de Bell

$$\begin{aligned} |\phi^+\rangle &= \frac{1}{\sqrt{2}}(|0_1 0_2\rangle + |1_1 1_2\rangle) && (50) \\ |\psi^-\rangle &= \frac{1}{\sqrt{2}}(|0_1 1_2\rangle - |0_1 1_2\rangle) && (51) \end{aligned}$$

Recentemente, Gu et al. [6] propuseram um DSQC o qual faz uso do DFS existente neste canal. De acordo com estes autores, a comunicação sigilosa pode ser feita da seguinte forma:

1) Alice prepara um estado emaranhado de três fótons

$$|\Phi^+\rangle_{AB_1B_2} = \frac{1}{\sqrt{2}}\left(|0_A\rangle\,|\phi^+_{B_1B_2}\rangle + |1_A\rangle\,|\psi^-_{B_1B_2}\rangle\right) \tag{52}$$

Ela mantém o qubit A e envia os qubits B_1 e B_2 para Bob;
2) Após receber a seqüência de Alice, Bob escolhe algumas amostras para checar a existência de espionagem. Para tanto, ele mede alguns qubits utilizando as bases $Z_{B_1} \otimes Z_{B_2}$, $Z_{B_1} \otimes X_{B_2}$ e $X_{B_1} \otimes Z_{B_2}$ escolhidas de maneira aleatória;
3) Bob divulga para Alice quais os qubits escolhidos para checagem de espionagem e o resultado obtido da medição de tais amostras;
4) Se Bob escolheu medir com a base $Z_{B_1} \otimes Z_{B_2}$, então Alice escolhe a base Z_A para medir seu fóton correspondente; em caso contrário, ela o mede utilizando X_A;
5) Alice e Bob utilizam a correlação existente entre as suas amostras para analisar a taxa de erro. Se o erro é maior que um limiar, eles repetem o protocolo desde o início. Em caso contrário, irão codificar as saídas $|0_A\rangle$, $|00_{B_1B_2}\rangle$, $|11_{B_1B_2}\rangle$, $|+_A\rangle$, $|0+_{B_1B_2}\rangle$, $|1-_{B_1B_2}\rangle$, $|-0_{B_1B_2}\rangle$ e $|+1_{B_1B_2}\rangle$ como correspondendo ao bit clássico 0, enquanto as saídas $|1_A\rangle$, $|01_{B_1B_2}\rangle$, $|10_{B_1B_2}\rangle$, $|-_A\rangle$, $|0-_{B_1B_2}\rangle$, $|1+_{B_1B_2}\rangle$, $|+0_{B_1B_2}\rangle$ e como correspondendo ao bit clássico 1;
6) Alice divulga para Bob a saída $C_A = O_A \oplus M_A$ em que O_A é o resultado das medições que ela obteve no fóton A e M_A é a mensagem secreta que ela deseja mandar para Bob de maneira privada;
7) Bob lê a mensagem secreta diretamente, i.e., $M_A = C_A \oplus O_B$, em que O_B é o resultado das medições que Bob efetuou nos fótons B_1 e B_2.

Antes de iniciar a análise deste protocolo, primeiramente algumas considerações serão feitas sobre ele. Os estados do DFS são estados de Bell, a correlação existente entre as

[1] A complexidade comunicacional é uma medida de quantas comunicações são necessárias para que duas partes possam concluir uma tarefa distribuída utilizando o mínimo de comunicações possível [31].

amostras de Alice e Bob permite uma checagem de espionagem e, por último, uma cifragem do tipo *one-time pad* é efetuada antes de a mensagem ser enviada.

Em face do DFS existente, algumas simplificações são passíveis de aplicação nesse protocolo. Se Alice e Bob desejam enviar a mensagem diretamente pelo canal quântico, uma codificação apropriada utilizando apenas os estados $|\phi^+\rangle$ e $|\psi^-\rangle$ pode ser feita alcançando a taxa de um bit de informação por uso do canal. Estes bits podem ser utilizados para criar uma chave secreta privada para codificar as mensagens utilizando *one-time pad*, de acordo com os dois últimos passos do protocolo de Gu et al. [6]. Outra sugestão que pode ser explorada tira proveito da correlação existente entre os estados de Alice e Bob para criar esta chave.

Em ambas as sugestões, a segurança incondicional provida pelo DFS é um ingrediente chave para as simplificações realizadas. Como pode ser observado, em nenhum dos casos a checagem de espionagem é requerida, o que reduz substancialmente o número de comunicações realizadas.

Estas sugestões podem ser aplicadas de maneira similar no DSQC proposto por Dong et al. [10] que é bastante similar ao protocolo de Gu et al [6] mostrado nesta seção. A principal diferença entre eles consiste no uso do canal clássico: enquanto o protocolo de Gu et al. utiliza este canal para enviar uma versão cifrada da mensagem, o protocolo de Dong et al. o utiliza para converter os resultados das medições nos bits apropriados da mensagem secreta. Neste protocolo, enviar uma mensagem de m bits requer que Alice e Bob troquem pelo menos $4{\cdot}m$ bits e qubits. Seguindo a primeira sugestão de modificação, este número de comunicações seria reduzido a m qubits.

C. Canal Quântico de Defasamento Coletivo

O canal quântico de defasamento coletivo já foi caracterizado anteriormente na Seção VI. Um protocolo que faz uso deste DFS foi proposto por Gu et al [6], o qual é bastante similar ao QSDC proposto por estes mesmos autores para o canal de rotação coletiva, discutido previamente na seção VII-B.

Também foi mostrado na Seção VI como enviar 1 bit de informação por uso do canal sem a necessidade de checagem de espionagem. A mesma idéia pode ser utilizada aqui para simplificar significativamente o protocolo em questão.

VIII. CONSIDERAÇÕES FINAIS

A partir da análise realizada, é possível concluir que se um canal quântico é caracterizado como na Definição 6, então a existência de certas simetrias pode ser explorada para enviar informação clássica com segurança incondicional. A codificação da informação em um DFS pode ser vista como uma instância de um código *wiretap*, com a particularidade de que nenhuma informação é capturada por um espião.

A expressão da capacidade de sigilo destes canais, mostrada na Eq. (25) é igual à capacidade de um canal quântico para o envio de mensagens clássicas [21]. Este é um caso particular em que a habilidade de um canal quântico para enviar informação secreta é tão grande quando a sua habilidade de enviar informação clássica ordinária.

Em se tratando da capacidade quântica de sigilo, Cai et al. [2] argumentam que esta não possui letra isolada e, em virtude disso, obter uma versão computável da mesma torna-se ainda mais difícil que obter uma versão computável para a capacidade clássica de um canal quântico. O caso particular para canais quânticos com ruído coletivo apresentado neste trabalho mostra que esta capacidade de sigilo é igual à capacidade clássica de um canal quântico, o que indica uma menor complexidade na obtenção desta capacidade.

Apesar das dificuldades existentes atualmente para construir sistemas quânticos completamente fechados [8], os resultados mostrados aqui podem ser aplicados para construir dispositivos que realizam a troca de mensagens com segurança incondicional mesmo na presença da descoerência. Isto é bastante promissor para implementações práticas, especialmente considerando os resultados já existentes sobre o uso de DFS em comunicações [32]-[34], incluindo de longa distância [35].

Uma primeira conseqüência dos resultados deste trabalho foi mostrar uma simplificação substancial em protocolos de QSDC e DSQC existentes na literatura. O número de comunicações realizadas e de operações pôde ser significativamente reduzido em função dos novos resultados sobre segurança incondicional e DFS. Isto reforça a viabilidade prática do uso de DFS em comunicações.

É importante enfatizar que os resultados apresentados não podem ser generalizados para todos os tipos de canais quânticos, pois nem todos eles satisfazem às condições para existência de DFS. Zanardi e Rasetti [20] argumentam que só existem DFS em cenários onde há descoerência coletiva. Apesar de ser um caso especial, as vantagens alcançadas em termos de segurança e taxa são significativas.

Em trabalhos futuros, sugere-se a investigação de condições mais gerais para a existência de sigilo absoluto em comunicações quânticas.

AGRADECIMENTOS

Os autores agradecem o apoio provido pelas agências de fomento brasileiras CAPES e CNPq e pelo projeto QUANTA/RENASIC/FINEP.

REFERÊNCIAS

[1] M. Schlosshauer, Decoherence and the Quantum-to-Classical Transition, Springer, Ed. Springer, 2007.

[2] N. Cai, A. Winter, and R. W. Yeung, "Quantum privacy and quantum wiretap channels," Problems of Information Transmission, vol. 40, pp. 318–336, 2004.

[3] I. Devetak, "The private classical capacity and quantum capacity of a quantum channel," IEEE Transactions on Information Theory, vol. 51, no. 1, pp. 44 –55, 2005.

[4] M. S. Byrd, L.-A. Wu, and D. A. Lidar, "Overview of quantum error prevention and leakage elimination," Journal of Modern Optics, vol. 51, no. 16-18, pp. 2449–2460, 2004.

[5] D. A. Lidar and K. B. Whaley, "Decoherence free subspaces and subsystems," arXiv:quant-ph/0301032v1, pp. 83–120, 2003.

[6] G. Bin, P. ShiXin, S. Biao, and Z. Kun, "Deterministic secure quantum communication over a collective-noise channel," Science in China Series G: Physics, Mechanics and Astronomy, vol. 52, no. 12, pp. 1913–1918, 2009.

[7] K. Majgier, H. Maassen, and K. Zczkowski, “Protected subspaces in quantum information,” Quantum Inf. Process, vol. 9, pp. 343–367, 2010.

[8] M. S. Byrd, D. A. Lidar, L.-A. Wu, , and P. Zanardi, “Universal leakage elimination,” Phys. Rev. A, vol. 71, p. 052301, 2005

[9] S. Qin, Q. Wen, L. Meng, and F. Zhu, “Quantum secure direct communication over the collective amplitude damping channel,” Science in China Series G: Physics, Mechanics and Astronomy, vol. 52, no. 8, pp. 1208–1212, 2009.

[10] H.-K. Dong, L. Dong, X.-M. Xiu, and Y.-J. Gao, “A deterministic secure quantum communication protocol through a collective rotation noise channel,” Int. J. of Quantum Inf., vol. 8, no. 8, pp. 1389–1395, 2010.

[11] L. Viola, E. M. Fortunato, M. A. Pravia, E. Knill, R. Laflamme, and D. G. Cory, “Experimental realization of noiseless subsystems for quantum information processing,” Science, vol. 293, pp. 2059–2063, 2001.

[12] A. Beige, D. Braun, B. Tregenna, and P. Knight, “Quantum computing using dissipation to remain in a decoherence-free subspace,” Phys. Rev. Lett., vol. 85, p. 1762, 2000.

[13] D. Kielpinski, “A decoherence-free quantum memory using trapped ions,” Science, vol. 291, p. 1013, 2001.

[14] P. G. Kwiat, A. J. Berglund, J. B. Altepeter, and A. G. White, “Experimental verification of decoherence-free subspaces,” Science, vol. 290, pp. 498–501, 2000.

[15] B. Schumacher and M. Westmoreland, “Quantum privacy and quantum coherence,” Phys. Rev. Lett., vol. 80, no. 25, pp. 5695–5697, 1998.

[16] A. D. Wyner, “The wire-tap channel,” The Bell System Technical Journal, vol. 1, pp. 1355–1387, 1975.

[17] A. Shabani and D. Lidar., “Theory of initialization-free decoherence-free subspaces and subsystems,” Phys. Rev. A, vol. 72, p. 042303, 2005.

[18] D. M. Bacon, “Decoherence, control, and symmetry in quantum computers,” Ph.D. dissertation, University of California at Berkeley, 2001.

[19] L.-M. Duan and G.-C. Guo, “Quantum error avoiding codes versus quantum error correcting codes,” Phys. Lett. A, vol. 255, pp. 209–212, 1999.

[20] P. Zanardi and M. Rasetti, “Noiseless quantum codes,” Phys. Rev. Lett., vol. 79, p. 3306, 1997.

[21] B. Schumacher and M. D. Westmoreland, “Sending classical information via noisy quantum channels,” Phys. Rev. A, vol. 56, pp. 131–138, 1997.

[22] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, C. U. Press, Ed. Bookman, 2010.

[23] D. Mayers, “Unconditional security in quantum cryptography,” Journal of the ACM, vol. 48, no. 3, pp. 351–406, 2001.

[24] G. Lu Long, F. Guo Deng, C. W. X. Han Lo, K. Wen, and W. Ying Wang, “Quantum secure direct communication and deterministic secure quantum communication,” Front. Phys. China, vol. 2, no. 3, pp. 251– 272, 2007.

[25] F. L. Yan and X. Q. Zhang, “A scheme for secure direct communication using EPR pairs and teleportation,” Eur. Phys. J. B, vol. 41, pp. 75–78, 2004.

[26] M. Zukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, ““Eventready- detectors” bell experiment via entanglement swapping,” Phys. Rev. Lett., vol. 71, p. 4287, 1993.

[27] T. Gao, “Controlled and secure direct communication using GHZ state and teleportation,” Z. Naturforsch, vol. 59, p. 597, 2004.

[28] T. Gao, F.-L. Yan, and Z.-X. Wang, “Controlled quantum teleportation and secure direct communication,” Chinese Phys., vol. 14, p. 893, 2005.

[29] F. G. Deng and G. L. Long, “Secure direct communication with a quantum one-time pad,” Phys. Rev. A, vol. 69, p. 052319, 2004.

[30] A. D. Zhu, Y. Xia, Q. B. Fan, and S. Zhang, “Secure direct communication based on secret transmitting order of particles,” Phys. Rev. A, vol. 73, p. 022338, 2006.

[31] R. de Wolf, “Quantum communication and complexity,” Theoretical Computer Science, vol. 287, no. 1, pp. 337–353, 2002.

[32] U. Dorner, A. Klein, and D. Jaksch, “A quantum repeater based on decoherence free subspaces,” Quant. Inf. Comp., vol. 8, p. 468, 2008.

[33] G. Jaeger and A. Sergienko, “Constructing four-photon states for quantum communication and information processing,” Int. J. Theor. Phys., vol. 47, p. 2120, 2008.

[34] Y. Xia, J. Song, Z.-B. Yang, and S.-B. Zheng, “Generation of fourphoton polarization-entangled decoherence-free states within a network,” Appl. Phys. B, vol. 99, pp. 651–656, 2010.

[35] P. Xue, “Long-distance quantum communication in a decoherence-free subspace,” Phys. Lett. A, vol. 372, pp. 6859–6866, 2008

Elloá B. Guedes é doutora em Ciência da Computação pela Universidade Federal de Campina Grande, pesquisadora do Instituto de Estudos em Computação e Informação Quânticas (IQuanta) e docente da Escola Superior de Tecnologia da Universidade do Estado do Amazonas (UEA). A autora já desenvolveu outros trabalhos nas áreas de Computação e Informação Quânticas, especialmente ligados à predição de geradores pseudoaleatórios criptograficamente seguros. Atualmente trabalha com simulação de algoritmos quânticos em computadores clássicos e também com comunicações seguras por canais quânticos ruidosos.

Francisco M. de Assis é professor titular da Universidade Federal de Campina Grande com pós-doutorado na Universidade de Toronto, Canadá. Os principais interesses de pesquisa do autor são Teoria da Informação Clássica e Quântica, Sistemas de Telecomunicação, Algoritmos e Complexidade Computacional. Atualmente é presidente do Instituto de Estudos em Computação e Informação Quânticas (IQuanta) e também coordenador do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande.

Revocation of User Certificates in a Military Ad Hoc Network

J. Jormakka and H. Jormakka

Abstract— **This paper presents a scheme for revoking certificates in a medium-small size semi-ad hoc military network, but the solution can be applied in the civilian side e.g. by police and crisis management. It describes the functionalities of a protocol to handle certificates, a set of policy rules in a node for handling certificates and an analysis how the proposed mechanisms can mitigate attacks on the certificate revocation solution. The mechanisms allows communication between the nodes on a lower security level even if the latest certificate revocation list is not available; protects against false revocations of certificates; and implements a mechanism for lowering trust levels of certificates.**

Keywords— **Ad hoc networks, Certificates distribution and revocation, Military.**

I. INTRODUCTION

A mobile ad hoc network is a wireless network where nodes transfer data to each other without the help of a base station. Usually data is passed through other nodes hop-by-hop. The paper describes a certificate revocation scheme of a military communication network for command posts and brigade headquarters. The network operates in semi-ad hoc mode, i.e., as an ad hoc network that is often connected to a fixed network.

Certificate revocation is the mechanism by which a Certification Authority (CA) announces that a certificate it has issued is no longer valid, even though its validity time has not expired. Certificate revocation is necessary if the private key corresponding to the public key in the certificate is suspected to be compromised, or for other reasons, e.g., if the user changes affiliation or name.

The certificate revocation mechanism in the ITU-T X.509 Recommendation uses Certificate Revocation Lists (CRL): the CA sends periodically new CRLs and puts them to the X.500 directory. Users can recover the list from Certificate Revocation List Distribution Points (CRL DP), a 1993 addition to the 1988 version of the X.500 Recommendation. The CRL mechanism is commonly used with other directory solutions.

Certificate Revocation Lists have a number of problems. One is the scalability of the mechanism in a very large network. Various solutions have been proposed in the literature. The network that is studied in this paper is so small that the scalability problem does not arise. Another problem with CRLs is that there is some time delay between the compromise of the certificate, e.g., loss of the private key, and the revocation of the certificate. Thus, there is always some time for authentication fraud. A problem that is characteristic to Mobile Semi-Ad Hoc Networks is the unavailability of the CRL: if the CA is reached through the fixed network and a wireless user is not sufficiently often connected to the fixed network, he cannot always have the latest CRL. Therefore he should not trust a certificate of another user. However, communication between users in the ad hoc network may be even more essential than reliable authentication. New mechanisms are required allowing communication between users on as high security level as can be offered and enabling sufficiently secure and efficient revocation of certificates in a semi-ad hoc network.

II. RELATED WORK

Certificate revocation is one of the known weaknesses in public key cryptography and a large number of research papers have been written on the subject. Much research has been directed to the scalability of the CRL mechanism by improving data structures, see e.g. [1], [2], [3], [4], [5]. The X.500 Recommendations already contain some options, like Delta CRLs. In a large network CRL distribution poses scalability challenges since the CRLs are typically very large. More general performance issues of large networks have also been treated, like in [6] and [7].

There are rather few proposals for certificate revocation in wireless ad hoc networks that address the problem that CRLs are not always available. This situation is most compelling in military ad hoc networks where connection to the fixed network that usually holds CLRs is often unavailable, and there exists a determined adversary, who tries to take advantage of the situation. In crisis and emergence response operations one usually may assume that the connections work and adversaries either to not exist or are not competent. In civilian ad hoc networks there often does not exist compelling reasons to secure the networks against imposters. However, in a military network one must secure communications. There are not that many alternatives if CRLs cannot be obtained. Either the protocol does not need certificate revocations – but then it requires renewals or other similar costly operations – or certificates are revoked by one or more participants in the ad hoc network.

Li et al in [8] describe a scheme for wireless ad hoc networks where each node keeps up the validity of its certificate using a One-Way Hash Chain. Other nodes can request the node to send a certificate with updated validity information. The method dispenses with certificate revocations, but it relies on keeping, in addition to the private key, another private secret in the mobile node. The authors propose keeping it in an USB-Key. This method can be useful in a network where the device may be lost but users carry the

J. Jormakka, Aalto University, Espoo, Finland, Comnet, (adjunct prof.) jorma.o.jormakka@gmail.com

H. Jormakka, Technical Research Centre of Finland (VTT), Espoo, Finland, henrykasj@gmail.com

USB-Key with them and this key is not lost. In a military ad hoc network the method does not give good security: if a node is lost it often means that not only the user, but also all he carries with him, is under adversary control. URSA in [9] is also a method, which dispenses with certificate revocation. It accomplishes this by requiring tickets. The drawback is a considerable traffic in renewing tickets. In Chinni et al. [10] certificate revocation is very shortly discussed in the context of renewing a certificate: the local environment is checked and if the node has not misbehaved or marked as convicted, it is granted a certificate. This description is too terse to be a method that can be implemented but it seems to suggest some kind of a vote.

Several authors have thought that if many users are needed for revoking a certificate, then certificate revocation must be slow. Thus, they have devised alternatives where only one user is needed to revoke a certificate. Naturally, this user may be mistaken or malicious, therefore the proposals require some cost to the revoking user, or assume that only trusted users can revoke certificates. As example of the first alternative is the suicide method in [11]. If a user revokes a certificate, he at the same time revokes his own certificate. In a military network this solution is not acceptable as the revoking soldier loses his communication capabilities and cannot fill his tasks. Two examples of the second alternative are [12] and [13]. In both cases only one user is needed to revoke a certificate, but the revoking user must be trustable. This does not work as we cannot know who is trustable.

Thus, a working solution for a military ad hoc network needs some kind of voting. Arboit et al in [14] present a revocation scheme based on a reputation protocol. It is voting but it focuses on network nodes independently collecting information of bad behavior from all other nodes. Because of this, the method will not revoke certificates sufficiently fast in order to protect military operations. The method presented in this paper is also based on the voting method. We require voting since one person can and does make mistakes and it is difficult to decide if a colleague should be excluded from the network. It is possible that his identity is stolen or that he is a spy, but this is never immediately clear. Thus more than one vote should be needed, but the votes are not accumulated over time as in [14]. Instead, if some suspicions arise, the soldier noticing them contacts physically another, usually higher ranking, member of the network, and they collectively exclude the potential threat.

Kitada et al [15] propose a Public Key Infrastructure (PKI) system where a node collects the certificates that it needs on demand. They also want to dispense with CRLs. The knowledge of revoked certificates is maintained by each node locally by asking each node that has issued a certificate if the certificate is valid. This solution is unsuitable to semi-ad hoc networks: if the issuer is in the wireless network it may be compromised, while if the issuer is the CA which is reachable through the fixed network, the connection may be broken. Morogan and Muftic [16] propose a validity time, the *grace period,* for CRL and a mechanism *channeling of the update information* by which ad hoc network nodes can obtain the latest CRL from each other. The mechanism for distribution of certificates is similar to the one presented in part *B.* of section 4 of this paper.

III. USE OF CERTIFICATES IN MANETS

The results presented in this paper were obtained in a project, where a medium-small size military mobile ad hoc network intended for the purposes of a command post or a brigade-level headquarter was designed and a mobile ad hoc node was implemented using, whenever possible, commercially available hardware and software. As a security solution in a military network has military specific requirements and cannot be directly adopted from civilian solutions, revocation of certificates was one of the parts that were designed specifically for that type of network.

The security requirements are stricter than in the civilian side since the adversary is better motivated and more capable than in many other usage scenarios, and the network gives access to classified material. Additionally, connections may be one-way only, i.e., existence of a jamming device, variations of signal power levels, or some radio propagation effects allow transmission to one direction only. Though the usual operation of the network requires high bandwidth bidirectional links, such as 802.11g or 802.16, the security solutions should be able to work on one-way connections if needed. As most protocols require responses, a very low bandwidth backward channel can be assumed to exist.

In many wireless ad hoc networks, both in the civilian and military sectors, there is a constraint on computing power imposed by small battery powered devices. This constraint is not critical in the intended network where users are usually in armored vehicles and the power source is provided by generators of these vehicles.

A stand-alone mobile ad hoc network can use whatever suitable authentication method between the users of the network nodes, but usually the ad hoc network is a part of a larger network and is in fact a semi-ad hoc network. The mobile ad hoc network considered in this paper is a typical military ad hoc network, which means semi-ad hoc network as the services to be used mostly reside in the fixed part and the network must provide organization-wide connectivity. Figure 1 shows the location of the brigade headquarter network between the fixed network and the tactical network (wireless military network for combat net radios).

The prevalent way of authentication in the wired network is based on the Public Key Infrastructure (PKI). The users in the mobile ad hoc network communicate with the services and users of the fixed network. This is why the authentication method in the mobile ad hoc network is most naturally also based on certificates. As the network is owned by one organization there is only one CA and all users have the CA certificate. Joint operations will become more common in the future and multiple CAs must be supported. The implied changes to handling certificate revocations are not large; mainly, a certificate path is needed.

We can assume that the CA can be trusted as CA compromise is expected to be very difficult. The mobile ad hoc network users can reach the CA through the fixed network.

In the fixed network user certificates, CA certificates and certificate revocation lists are stored in a directory. In the wireless ad hoc network public key cryptography should preferably be used without a directory because it cannot be assumed to be always available. A combination of the following two methods dispenses with the directory.

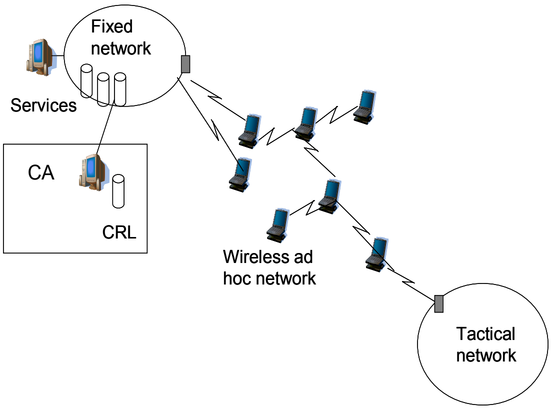


Figure. 1. Command-post/brigade headquarter ad hoc network. The terminals access services in the fixed network and take part in group decision making. The applications are database applications with graphical interfaces, voice and possibly video.

A. Storing certificates of all users locally

Storing certificates of other users locally is a possibility if the network is not too large. The memory of a smart card is not sufficient for certificates of all users even in a small network. Let us assume that only the certificates of the user himself, those of the services and the CA certificate(s) are stored on a smart card. Certificates of other users can be stored on a USB plug-in, Compact Disc (CD) or hard disc. Let us assume that the way the network is used is that activities are divided into operations which have a well-defined starting time and usually also a well-defined finishing time. At the beginning of each operation valid certificates are distributed to all users e.g. with a USB plug-in or a CD. During the time of the operation certificate revocations and new user certificates are added to the hard disc. A certificate of a new user can be obtained from the CA .

B. Sending own certificate

The user who wants to be authenticated sends his certificate during the authentication phase. The receiver checks the validity of the user certificate by checking the signature of the CA and the validity time of the certificate.

An adversary trying to impersonate a user may give a revoked certificate which is still valid. Therefore, either the validity time should be very short or users need certificate revocation lists in order to authenticate other users in a secure way. In the first case the certificates must be renewed during the operation. This may be impossible as connectivity to the fixed network can be broken. We will assume that the validity time of the certificate sent by a user has not expired during the operation time frame, thus certificate revocations are necessary.

IV. ATTACKS AGAINST CERTIFICATES

Public key cryptography has a quite high security level, but there are many potential threats to certificates.

- Attack 1: Registering a certificate on somebody else's name. The CA is assumed to be capable of validating the user before giving a certificate.
- Attack 2: Changing a certificate to a crafted certificate. Certificates are signed by the CA making this practically impossible.
- Attack 3: Pretending to be a CA and falsifying certificates of users. All trust is lost if the certificate path includes a CA that does not correctly check the identities of users to whom it issues user certificates. In practice, an adversary must be able to insert a false CA into a certificate path. In the 1988 version of the X.500 directory this was at least theoretically possible since user certificates and CA certificates had the same structure and an adversary might cause a user certificate to be taken as a CA certificate. A user gets a user certificate if a CA trusts that the user is whom he claims to be, whereas a CA Certificate is only granted to a party who is trusted to be able and willing to behave as a CA, e.g., to check the identity of other users. In X.500 versions since 1993 the possibility of this attack can be removed since there is a field that can be used to distinguish between certificate-types.
- Attack 4: Blocking CRL updates or removing them from the directory. This attack is solved in X.509 by requiring that CRLs are sent periodically on known times (i.e., CRL indicates the time the next CRL will be received). Then the user knows if he has the latest CRL. The update period for CRL may be too long for a particular application and there may be times when an immediate revocation is needed. The CA can issue a new CRL at any time but it cannot be assumed that a user will receive the CRL or know of its existence. Thus, the same revocation must be also in the periodically sent CRLs.
- Attack 5: Registering a valid user certificate with a name that is misleadingly similar to a name of a valid user. This mechanism is a form of social engineering: a user mistakenly takes another user for somebody else and authentication also succeeds: the cryptographic application authenticates him to be what his name literally states. This kind of acting is difficult if the CA is making intelligent decisions of what names users can have. There are cases when this type of misleading may still be possible. A user with the same name may be mistaken to be another user sending e.g., from the home terminal. If a company has a good security policy, such cases can be minimized.
- Attack 6: Threatening, bribing, blackmailing or in other way persuading a valid user to behave in the way the adversary desires. This threat cannot be removed by technical means.
- Attack 7: Stealing the private key and using it before the theft is noticed. This is always possible and cannot be well protected against.
- Attack 8: Hacking, social engineering, sending malicious code, or in other ways breaking into network nodes and stealing or modifying information.
- Attack 9: Announcing the certificate of a valid user compromised. This can cause a denial of service attack to the valid user.
- Attack 10: Downloading CRLs in order to cause a denial of service attack to the network.

- Attack 11: Checking revocation of a certificate in order to learn something of the state of a valid user. If lost private keys are revoked, this gives information of what nodes are known to be lost. In a military network this can be important information.
- Attack 12: Gaining access to the system with stolen credentials although loss of the private key was noticed. This may occur if certificate revocation has not been spoted.

Additionally, the following problems may occur in connection with a public key cryptosystem:

- Problem 1: Refusing communication with a valid user since the latest CRL is not available and one side of the communication cannot verify the validity of the public key. In time-critical activities this is an annoying problem.
- Problem 2: CRLs can be long and distribution of CRLs may cause excessive load to a mobile network.
- Problem 3: The CA may become a bottleneck.

Some of the above attacks are sufficiently well solved by the Public Key Infrastructure system in the fixed network (Attacks 1, 2, 3 and 4). However, we must still keep these attacks in mind when proposing any modification to the handling of certificates. For instance, a mechanism sending revocations on demand must consider Attack 4, while sending a validity statement of a certificate on demand can be quite secure. Some attacks are unsolvable or non-technical and cannot be dealt with (Attacks 5, 6, 7 and 8). We look for a solution mitigating or removing four attacks (9, 10, 11 and 12) and the three problems. As there are attacks that cannot be removed, the goal is not perfect security but rendering the mentioned attacks at least as difficult as the main remaining threats.

V. REVOCATION POLICY ALTERNATIVES

A typical service provided in a military network accesses sensitive data stored in data warehouses. It is most often a database application and the access paradigm is publish/subscribe. There are multiple copies of the same data and we can assume that the load on one data warehouse does not become a limiting factor and a user can obtain the service he needs from some of the data warehouses. Partial local copies of the data are made in the wireless network. Services are not mobile, i.e., moving from one computer platform to another, mainly for security reasons.

In a typical service usage scenario a user authenticates to a service with strong credentials using a certificate and a private key stored on a smart card and that the card is protected by a password. The services could be protected by an additional password that have to be memorized and that is one per user to all the services. The motivation for the password is to protect the services in case a node and a smart card are both captured, and the password (PIN code) to the smart card is available, or e.g., the lock on the smart card is removed by a suitable application of a voltage level.

It is common that a node or the whole wireless network is not connected to the fixed network. Usually this only means technical communication problems but in case of a military network it may mean that the smart card, possibly the whole node, is captured and a private key may be compromised.

There are at least three possible approaches to certificate revocation:

1. *If the wireless network is not connected to the fixed network do not usually revoke certificates corresponding to a lost smart card. Let the CA, or a node authorized by the CA, the possibility to revoke a specific certificate*

In this policy the assumption is that it is very difficult for the adversary to obtain the private key even though the valid user has lost his smart card. However, it should be able to revoke a specific certificate if there is reason to suspect that a private key is compromised.

2. *Use a shared and not electronically stored password for authentication as an additional mechanism to strong credentials*

Password authentication is here used in a way similar to a PIN code. If the password can be recovered from lost card, the mechanism does not add any strength to security. Therefore the password must not be stored. It is difficult to remember many passwords and therefore it should be shared by the users of the wireless network.

3. *Always revoke all certificates corresponding to a lost smart card.*

This is a natural policy: usage of a lost smart card indicates that authentication with the credentials is not from a valid user, or at least one should verify the user.

The advantage of the first policy is that revocation is done by the CA in the same way as in the fixed network, i.e., when the ad hoc network has connectivity and there are no military ad hoc network specific problems. In the intended application the material is highly sensitive and the risk that a private key is broken from stolen equipment or obtained from a captured user is too large and overcomes the problems of certificate revocation. This policy must be discarded.

The second policy does not need CRLs and access cannot be gained by using a stolen node or smart card. In a very small military network the users could be considered being able to keep the shared secret. However, there is a disadvantage - one more password has to be memorized. This means that the mechanism is too weak to protect sensitive data. We must discard this solution also.

The selected solution is the last policy. In that case certificate revocations can be common and occur when the network is disconnected from the CA. This implies that there is a need for a mechanism revoking certificates also when the CA is not reachable. The protocol enabling certificates revocation is briefly presented in the next section.

VI. CERTIFICATE REVOCATION PROTOCOL

The protocol supports four functionalities each containing some protocol actions. Only the main ideas of the functionalities are presented in this paper.

User credentials are considered lost if they are announced lost to the CA. This means that the network is not polling users in order to check whether their credentials are lost, instead there is normal communication between users and if a user thinks that credentials of another user are misused, he issues a *Doubt* message to the CA. The *Doubt* mechanism is described later. The CA makes the decision to consider credentials lost and sends a revocation of the certificate.

A. Verification and distribution of certificates

The first functionality that must be supported enables users communication even if the fixed network is not available. Although most user certificates are available in a local storage, CRL is not always up to date. Unencrypted communication is not an acceptable solution. Functionality *A.* comprises of three protocol actions and has the necessary security level.

Partial-authentication

In case when two users start communication and one of the nodes notices that it does not have the most current CRL, it concludes that it can make only partial authentication. Communication is possible with partial authentication (PA), but in that case a particular *PA security policy* is applied by both sides restricting the message types and services. Messages between parties using partial authentication are marked with a special *partial-authentication* flag. This makes it possible for other nodes to notice if partially authenticated communication is carried through them.

Send-own-certificate

A user can add its certificate to the message if it expects that the other side of the communication does not have it. This mechanism consumes bandwidth considerably and should be used only by users who enter a new, rarely used by him network, or if they have received a new certificate.

Distribute-certificate

A user may distribute a certificate to the whole mobile ad hoc network. An efficient distribution mechanism is assumed to exist in the network as the typical service in the network is group collaboration, e.g., planning military operations. Distribution can be made using a distribution node and one-to-one connections or by multicast. Multicast in mobile ad hoc networks is usually difficult, but in the application the nodes are typically not moving while taking part in group collaboration. The distribution mechanism is not assumed to be reliable, i.e., not all nodes always receive the messages.

B. Distribution of certificate revocations

The second functionality provides mechanism for distributing certificate revocations. CA, the party that revokes certificates, distributes CRLs to the data warehouses periodically, but also has the possibility to issue and distribute a CRL at any time. To prevent congestion the users of the wireless network do not fetch the CRL from the CA. Instead, each service located in any of the data warehouses of the fixed network after mutual authentication can provide it on a user's request. As the services in any case contain sensitive data, we may assume that they are well protected and trustable. If not, security has already been lost. Because of the military character of the network the users are grouped into units which are commonly working together. Therefore it is practical that a user requests a validity statement of the certificates of a group. The service requested forms a message:

Message := { group-id, start-time, (all-valid | (revoked, certificate)| (group-validity, bit-string))}*

The group-id is a two byte id of the group to which belongs the party whose certificate validity is being checked. If needed, the list of the group members can be obtained from the warehouse server. The two byte field *start-time* is the agreed starting time of the operation. It is measured in seconds, starting from 0. As the nodes know the validity duration, the ending time is not coded. The third field in this message has three alternatives: *all-valid=00*, *revoked=01*, *group-validity=10*. For optimization reasons the option *revoked* is used if the number of revoked certificates is less than eight. Otherwise the last option (*group-validity)* is used. After each *revoked* comes an identifier for a user whose certificate is revoked. The identifier is 6 bits, which gives one byte with the prefix *revoked*. In case where more than eight certificates were revoked, the *group-validity* option is used. After *group-validity* follows a bit string where each bit corresponds to a group member: a revoked certificate is coded as bit one and each valid certificate as bit zero. The group of the brigade headquarter network has less than 64 members. The group validity bit string together with the two bites (10) is eight bytes long. *Message* is padded to full bytes with a bit string of ones.

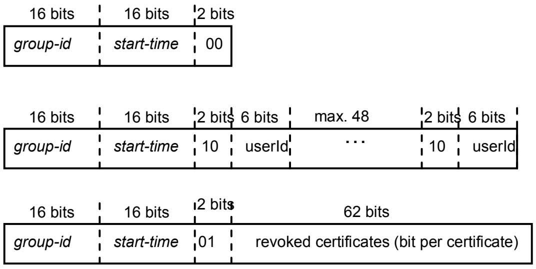


Figure 2. The structure of the distribution of certificate revocation message.

CRL-update

The service sends the *Message* to a user. The message is protected by the shared session key, so the user trusts the message because it trusts the service. This protocol action produces very small messages and can effectively cope with Problem 2.

CRL-update-signed

The service signs the *Message* and sends it to the user. This protocol action produces larger messages, but there is the advantage that the user can pass the message to other users. If the bandwidth of the network allows, this action is the preferred one.

Announce-CRL-DP

If a node in the network has the latest signed CRL (from *Message*) for a member of a group and it notices that communication marked with the flag *partially authenticated* is passing through it, the node sends an *Announce-CRL-DP* message to the parties in the communication. This message informs the nodes that an up-to-date *Message* can be obtained from this node.

Request-CRL
A user requests *Message* from a service or from another user with this request. A user receiving this request first checks that the requester is announced valid in the CRL it has from the *Message*, then authenticates the user, and if everything is verified, sends a *Message* signed by a service. The requesting user does not need to trust the user who sends the *Message*, only to trust the service.

Send-own-CRL
A user may send *Message* signed by a service. Sending *Message* that contains a validity proof of the user certificate removes the problem of revoked certificates in the *Send-own-certificate* action.

C. Revoking certificates

The third functionality is revoking certificates. The usual way of revoking certificates is that the CA issues a CRL where the certificate is revoked.

Revoke
The *Revoke* action is a command for revoking a particular certificate at any time. It can be given by the CA or by another user to whom the rights of the CA has been transferred. The issuer of *Revoke* distributes the message, in the ad hoc network it is distributed to the whole network.

Doubt
Announcing revocations is a problem in any system using certificate revocation lists. A user who has lost his credentials cannot be authenticated in a strong way before he obtains new certificate. In an ad hoc network he most probably announces the loss through the same network as usually no external communication network is available. As he must revoke his certificates without credentials, he must access as another user. This means that an adversary can equally well try to revoke certificates of any valid user and in this way block the user from the network. The tasks are typically time-critical so even temporary denial of service situations must be avoided. The *Doubt* mechanism is designed so that blocking valid users is difficult.

Any node can announce to the network that it suspects that another node is not trustable. *Doubt* is a one-way protocol requiring sending the message doubt $\{b, k_b\}$ where b is a user id and k_b is the user's public key. There are three principal reasons for a node to send the doubt message. One is the loss of a user's private keys. In such case the user must access some other node and have a valid user of the node issue a *Doubt* on his certificate. Another reason is when a node monitoring traffic passing through notices that another user makes several failed attempts to access services. Services are protected by passwords and if a user accesses the services and becomes refused several times there is good reason to suspect a compromised node. A third reason to issue a *Doubt* is when a group of users have agreed that some node is compromised and decide to force the believe level on that user's certificate to zero in order to exclude the user from the network.

Clear
The nodes keep a believe level for each certificate. Receiving a *Doubt* message lowers the believe level of a certificate. Receiving *Clear* initializes the believe level of a certificate. *Clear* can be issued by the CA or by a user authorized by the CA.

Check-up-question
There are cases when a user has to prove his identity by answering check.-up questions. One of such cases is when the user has lost his smart card and tries to revoke his certificate by sending *Doubt and* the CA cannot authenticate him in the usual way. Another one is when an adversary tries to invalidate a certificate of a valid user. There is no especially good method for solving this problem. The usual way is to store some questions of personal information with answers to an announcement centre (here, the CA) and require a correct answer for revoking the certificate. Personal information is rather easily obtained and temporary PIN codes known to the user and the CA may be slightly stronger. The certificate could be revoked by the user giving this PIN and his name to the CA. The information can only be used once.

D. Authorization

The network has a trusted entity, the CA. As the CA is not always available and as the solution is intended to a military network, a trusted entity capable of revoking certificates, issuing new certificates and clearing doubts is needed. The CA is authorizing an entity to act as a trusted entity by issuing *Transfer-of-rights*.

Transfer-of-rights
The CA can transfer rights to another user. It is outside to scope of the technical solution to guarantee that the user is trustworthy. *Transfer-of-rights* is not distributed to the network. If a user is transferred the rights of the CA and wants to do an operation with CA rights, it must include the transfer message structure to the message so that the other nodes know it is authorized.

VII. CERTIFICATE REVOCATION POLICY RULES

In this section we will derive rules and conclusions concerning the *Doubt* protocol. The presented below activity consist of operations which have a definite starting time and a known duration. At the starting time all parameters are initialized.

The policy language used here is a modified and simplified version of the formalism presented in [17] for public key systems in such a way that it can support the above *Doubt* protocol using a *believe* set that expresses the level of user's trust on other users certificates.

We assume that there is only one CA; called ca. It is directly trusted by all users and all certificates are signed by ca. This assumption reflects the situation that the network is owned by one organization and it is not very large. As organizations today often enter into joint activities, in the future this assumption will have to be relaxed and certification paths must be allowed.

Expressions

- The expression "a `says` S" means that the user a sends an electronically signed message stating that the expression S is true.
- The expression "a `transfer` b" means that the user a transfers rights to the user b.
- The expression "a `doubts` S" means that the user a has sent a message *Doubt* for the expression S.
- The expression "a `revoke` S" means that the user a has sent a CRL revoking the expression S.
- The expression "`trust` $\{b,k_b\}$" means that the user trusts the public key k_b to belong to the user b.
- The expression "`clear` $\{b,k_b\}$" restores the trusts on the expression that the public key k_b belongs to the user b.

Each node a keeps a table of values of the type $believe_a[b]$, where each of the values expresses the level that user a believes in validity of user b belonging to certain group B. Because of the specific nature of military ad hoc network we assume that the set B covers a unit to which belongs the user's group, but the solution could be generalized by extending the table dynamically whenever a new member accesses the network. Each of the $believe_a[b]$ is initialized to a small number M at the time the operation is started. Each node a keeps also a queue $queue_a(b)$ of identities of users who have submitted *Doubt* messages for b. The size of the queue is the number of users allowed to lower the believe level of a certificate to zero. This number is expected to be small (2-5), so the memory requirements are not too large. The queue is initialized to zero. The queue is a First-In-First-Out queue. POP takes the first identity to be serviced from the queue and PUSH puts an identity to the end of the queue.

The policy rules are read from up down, that is whatever rule is first filled, its conclusion is taken. The policy rules in each node are as follows:

```
R1: $ca$ says transfer $a$, $a$ says S $\Rightarrow$ $ca$ says $S$
R2: $ca$ says revoke $\{b,k_b\}$ $\Rightarrow$ $believe_a[b]=0$
R3 $a$ says doubt $\{b,k_b\} \wedge a \in queue_a(b)$
    $\Rightarrow$ GO TO R5:
R4: $a$ says doubt $\{b,k_b\} \wedge ca$ says $\{b,k_b\}$
    $\Rightarrow believe_a[b]--$
    PUSH $a$ $queue_a(b)$
R5: $ca$ says $\{b,k_b\} \wedge believe_a[b]>0$
    $\Rightarrow$ trust $\{b,k_b\}$
R6: $service$ says $\{b,k_b\} \wedge believe_a[b]>0$
    $\Rightarrow$ trust $\{b,k_b\}$
R7: $ca$ says clear $\{b,k_b\}$ $\Rightarrow$
$believe_a[b]=M$
WHILE ( $\exists c \in queue_a(b)$ ) POP $queue_a(b)$
```

VIII. DISCUSSIONS

The *Doubt* mechanism can realize to some extent the idea of the RUMOR protocol proposed by [18]. Using the RUMOR protocol any node may announce its doubt that a node is compromised. This protocol is only a communication mechanism and does not specify how a node concludes that it sends a RUMOR and what a node receiving a RUMOR should do. The *Doubt* mechanism has a smaller scope and is more precisely defined. Any node can send the *Doubt* message indicating that the binding between the user and the public key is in doubt. A node, which receives a *Doubt* message will lower the believe level of this certificate.

There are a number of issues that must be considered in the *Doubt* protocol. Let us assume that there is a compromised host in the network. If it can send a sufficient number of *Doubt* messages and in this way force the believe level on a valid certificate to zero, it can create a denial of service to a valid user. If on the other hand, there is only one valid user that knows that a certificate should be revoked, for instance a valid user who has lost its private key, his announcement should be acted on. A cryptographic shared secret is a too heavy mechanism for this purpose. A reasonable compromise has been achieved with the $queue_a(b)$ mechanism. It is a modification of the simple CHOKe mechanism that has been proposed in [19] for limiting UDP flows.

Announce-CRL-DP has a similar purpose as the *channeling of the update information* in [16]. The main difference is how a node which has an up-to-date CRL notices that it should give the CRL to other nodes. In a military network it is occasionally necessary to restrict communication to the minimum. In this low activity mode of the network, each node monitors traffic passing through it and if it notices that a connection has the *partial-authentication* flag set, it sends *Announce-CRL-DP* to the parties in the communication. This mechanism saves bandwidth and makes detection of the network less likely.

IX. ANALYSES

Let us see how the proposed solution mitigates the problems and attacks listed in section 4. It is not possible to completely remove these threats, only to make the attacks more difficult than some other attacks, such as social engineering, hacking and malicious code. No probability measure can be assigned to such attacks: therefore we will create a measure by a tactical argument. Let us say that an attack is a *serious threat* if a form of the attack that is likely to work can be designed. There usually are errors in the code and with a finite amount of work an adversary can find a successful attack. Many people can be misled and it is possible to find users that are likely to fall on a well-designed social engineering attack. Thus, these attacks make a serious threat. Let us say that an attack is a *minor threat* if using it requires much of good luck. The assumption is that a professional attacker prefers to plan less random attacks using mechanisms that require less of a good luck. The goal is to show that the revocation scheme renders attacks 9-12 and the problems 1-3 to minor threats.

Proposition 1. Attack 9 is a minor threat.

Argument: If an adversary tries to invalidate a certificate of a valid user, he issues a *Doubt* message. There are the following alternatives:

1. The *Doubt* message reaches the CA. The CA sends a *Check-up-question* to both the adversary and to the valid user. The valid user answers correctly. If the adversary answers incorrectly, he cannot invalidate the certificate. If the adversary answers correctly, the issue is investigated further according to the CA policy. In this case the certificate of a valid user is revoked only by good luck.
2. The connection to the fixed network is broken. The adversary must convince *k* other users to send a *Doubt* message in order to force the believe level of a valid user's certificate to zero (the length of the believe queue is in this case *k)*. If *k* is selected sufficiently large, this can be considered to require too much luck.

*Proposition 2***:** Attacks 10 and 11 are minor threats.
Argument: User authentication, with checking of certificate revocations, is required before a request of certificate revocations is accepted by a service or another user. This means that the system must already be compromised if Attacks 10 or 11 succeed.

*Proposition 3***:** Attack 12 is a minor threat.
Argument: The mechanisms for distributing certificate revocations mitigate this attack.

If the CA decides that a certificate is not trustable, it issues a CLR and sends it to services. Revocation is made by the CA if the network is connected to the fixed network. Otherwise, a user which has been transferred CA rights can revoke certificates. It is also possible for a set of users to lower the believe level of a certificate to zero by the *Doubt* mechanism.

If a user notices that his certificate should be revoked, for instance, has lost the private key, he issues *Doubt* on his certificate. If the CA receives *Doubt*, it will pose a *Check-up-question* to the sender of the *Doubt* message and to the user with the certificate to be revoked. If the user can answer the *Check-up-question* correctly, and the user of a compromised node does not answer at all, or answers incorrectly, the CA revokes the certificate. Thus an adversary has a high risk that the revocation of a lost certificate is distributed.

It may be argued that proposition 3 is not quite filled. CA rights are transferred to a single person and thus compromising this person has fatal consequences. This is true, but the military nature of operations implies that the commanders always have an important role in each operation. It is more important to grant a single person CA rights than to protect the network against attacks.

Proposition 4: Problem 1 is a minor threat.
Argument: Functionality *A.* (section 6) allows communication to the extent that the policy rules for partial authentication allow. Serious lack of communication requires an unfortunate combination of events.

Proposition 5. Problems 2 and 3 are minor threats.
Argument: We can argue that the protocol in section 6 can be applied in a way that requires the minimum possible amount of transfer of data from the fixed network to wireless network for certificate revocation purposes.

Due to sensitivity of the material in the services, revocation of certificates is necessary. In the presented method obtaining information that a certificate is revoked requires in minimum one bit of information per certificate (see Fig. 2). Compressed form of this information is the smallest amount of data and the proposed compression method is nearly optimal (it is octet-aligned for efficiency). Encrypting it with a symmetric crypto-algorithm is needed for security purposes and it does not increase the size of data. Therefore *CRL-update* contains for practical reasons the smallest possible amount of data: Problem 2 is then minimized to the extent that it can be. Additionally, as in the designed method load is distributed to services, Problem 3 is not a threat.

X. RUNNING ON UNIDIRECTIONAL LINKS

Let us briefly discuss running the revocation protocol on unidirectional links, as was desired in section 3. In that case all the protocol messages needed for certificate revocation must either be one-way or require answers that can be given on a low bandwidth connection in the reverse direction.

A simple protocol between two terminals *A* and *B* is presented below. It protects against: man-in-the-middle attack, replay, eavesdropping, impersonating *A* and *B*, as well as capturing *A* and *B*.

Let us assume that *A* can reach *B*, but not vice versa. Both *A* and *B* share predefined knowledge - tables of numbered keys (passwords), called *local-keys*. Let us introduce the following notations: K_B is the public key of *B*, $[X]_k$ means that data *X* is encrypted with the key *k*, $signed_A\{X\}$ denotes data *X* with electronic signature made by *A*. Let $cert_A$ denote the certificate of *A* and *key1|key2* mean a bitwise concatenation of two keys, *key1* and *key2*.

In order to send data to *B*, *A* has to authenticate itself. *A* sends to *B* the following data: *Message* signed by a service *S, (*optionally*)* its own certificate, and a signed message encrypted with the public keys of *B*. The message contains a seed for the session key, called here *session-key1*. The session key is obtained by a bitwise concatenation of *session-key1* and one of the numbered keys from the shared table. The message also contains a serial number:

$$M_{auth} = signed_S(Message),\ (cert_A\ ),$$
$$[signed_A\{session\text{-}key1,\ local\text{-}key\text{-}number,\ serial\text{-}number_A\ \}]_{KB}$$

$$local\text{-}key = local\text{-}key\text{-}table[local\text{-}key\text{-}number]$$
$$session\text{-}key = session\text{-}key1|local\text{-}key.$$

When *B* receives the message, it must use its private key in order to open it. Thus, *A* can trust that either the message was received by *B*, or the receiver cannot open the message. To obtain the session key the user needs a shared key pointed by the *local-key-number* in the message. The key concatenated with *session-key1* gives the session key to be used in one way communication where *A* sends to *B* messages of the form:

$$M_{data}=[data]_{session\text{-}key}\ .$$

As there is no return channel, or it has very low bandwidth, strong forward error correction is needed for all messages.

The use of tables with passwords provides access control, as the tables of passwords are protected by the access rights of their users. It also protects against using a stolen or lost terminal. The serial number prevents replays. *A* can either keep track of all users *B* and keep a counter for every *B*, or it can use one counter for all sent packets. In either case, *B* should not receive multiple times a packet with the same serial number from *A*. Both *A* and *B* share the same CA and the public key of the CA and usually the certificates of *A* and *B* are locally stored in both *A* and *B*, thus *B* can verify the certificate. *B* can check if the certificate of *A* is revoked from the *Message*. If needed, *A* also sends its certificate.

Often some low bit channel from *B* to *A* is available, it can be HF radio, some covert channel or the actual signal channel which sometimes works. This low bit channel can be used by *A* for verifying any information, such as that *B* obtained the message. *A* poses a question and gets the answer in some simple code. An example is a code where *A* encrypts with the session key an integer and *B* must reply if the integer is odd or even:

$$A \to B\ ,\ [number]_{session\text{-}key}$$
$$B \to A\ ,\ [answer]_{session\text{-}key}\ ,$$

$$answer = number \bmod 2.$$

Correct answers to N questions can be given by chance with probability 0.5^N. For high security N to should be very large, but in this case N=3 or N=5 might suffice: the probability of recognizing the enemy correctly before shooting is only between 80% and 99%, therefore the needed confidence can be of the same range.

This simple protocol for unidirectional links solves revocation of certificates with *Message* from *B*., section 6. A useful subset of the actions in the protocol in 6 can run on top of this protocol as most of the actions are one-way.

XI. Conclusion

The paper proposes a scheme for certificate revocation in a mobile military ad hoc network. The intended application is from the military side but some of the mechanisms may have wider use. The outlined protocol for revocation of certificates fills the needs of the intended application and gives a sufficient security level for practical purposes.

The solution can be applied in the civilian side e.g. by police and crisis management. There are some conditions that the proposed mechanism assumes. The activity should be organized in operations which start at specific times because the solution synchronizes shared secrets at the start time. If such times are not available, difficulties in synchronizing shared secrets lower the security level of check-up questions and other similar mechanisms. The adversary also should plan his actions in a cost-effective way, rather than the ad hoc way of script kiddies. If this kind of an assumption cannot be made, the division of threats to major and minor threats is obviously not useful.

Acknowledgments The work presented in the paper was supported by the anonymous.

References

[1] M. Naor and K. Nissim, Certificate Revocation and Certificate Update, *IEEE J. on Selected Areas Comm.,* Vol. 18, No. 4, pp. 561-570, 2000.

[2] P. Kocher, On certificate revocation and validation," in Financial Cryptography-FC'98, *Lecture Notes in Computer Science*, Berlin, Springer-Verlag, pp. 172-177, 1998.

[3] S. Micali, Efficient certificate revocation, Tech. Memo MIT/LCS/TM-542b, 1996.

[4] M.E. Nowatkowski and H.L. Owen, Certificate Revocation List Distribution in VANETs Using Most Pieces Broadcast, *Proc. IEEE SoutheastCon 2010*, pp. 238-241, 18-21. March 2010.

[5] J. J. Haas, Y-C. Hu, and K. P. Laberteaux, Efficient Certificate Revocation List Organization and Distribution, *IEEE J. on Selected Areas Comm.* Vol. 29, No. 3, March 2011.

[6] C. B. Popescu, B. Crispo, and A. S. Tanenbaum, A Certificate Revocation Scheme for a Large-Scale Highly Replicated Distributed System, *Proc. 8th IEEE International Symposium on Computers and Communication* (ISCC′03), 2003.

[7] B-H. Li, Y-B. Hou, and Y-L. Zhao, A Scalable Scheme for Certificate Revocation, *Proc. 4th International Conf. on Machine Learning and Cybernetics,* Guangzhou, 18-21, pp. 3852-3856, Aug. 2005.

[8] J. Li, Y. Zhu, H. Pan, and S. Liu, A Distributed Certificate Scheme Based on One-Way Hash Chain for Wireless Ad Hoc Networks, *Mobile Technology, Applications and Systems, 2nd International Conference*, 2005.

[9] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks, *IEEE/ACM Tr. on Networking*, Vol. 12, No. 6, Dec. 2004.

[10] S. Chinni, J. Thomas, G. Ghinea, and Z. Shen, Trust model for certificate revocation in ad hoc networks, *Ad Hoc networks*, No. 6, pp. 441-457, 2008.

[11] J. Clulow and T. More, Suicide for the Common Good. a New Strategy for Credential revocation in Self-Organizing Systems, *AMCSIGOPS Operating Systems reviews*, vol. 40,no. 3, pp. 18-21, Jul. 2007.

[12] W. Liu, H. Nishiyama, N. Ansari, and H. Kato, A Study on Certificate revocation in Mobile Ad Hoc Networks, *Proc. IEEE ICC 2011*. 2011.

[13] K. K. Chauhan, and S. Tapaswi, A Secure Key Management System in Group Structured Mobile Ad Hoc Networks, *Proc. WCNIS,* pp. 307-311, 25-27 June 2010.

[14] G. Arboit, C. Crepeau, C.R. Davis, and M. Maheswaran, A localized certificate revocation scheme for mobile ad hoc networks, *Ad Hoc Networks*, No. 6, pp. 17-31, 2008.

[15] Y. Kitada, A. Watanabe, and I. Sasase, On demand distributed public key management for wireless ad hoc networks, *Communication, Computers and Signal Processing*, 2005. PA CRI M. 2005 IEEE Pacific Rim. Conf. 24-26, pp. 454-457, Aug. 2005

[16] M. C, Morogan and S. Muftic, Certificate Management in Ad Hoc Networks, IEEE Database, 2002.

[17] R. Kohlas and U. Mauer, Reasoning About Public-Key Certification: On Bindings Between Entities and Public Keys, *IEEE J. on Selected Areas Comm.*, Vol. 18, No. 4, pp. 551-560, 2000.

[18] C. Candolin and H. H. Kari, Distributing incomplete trust in wireless ad hoc networks, *Proc. IEEE SoutheastCon*, pp. 68-73, 2003.

[19] P. Pan; B. Prabhakar and K. Psounis, CHOKe: A stateless AQM scheme for approximating fair bandwidth allocation, *Proc. IEEE INFOCOM*, Mar. 2000.

Jorma Jormakka (M'98) received the Ph.D. degree in mathematics 1988 from the University of Helsinki. Currently he is an adjunct professor at the Aalto University and at the National Defence University. In the years 2000-2010 he was the professor of command and control systems at the National Defence University. During 2000-2004 he was professor of information technology at the Helsinki University of Technology, and in 1997-1999 he was professor of telecommunications at the Lappeenranta University of Technology.

Henryka Jormakka obtained her Master and PhD degrees in Mathematics from University of Lodz, Poland. She has worked as a research professor of information technology at Technical Research Centre of Finland and at Lappeenranta University of Technology. Her research interests include telecommunication protocols, service architecture, mobility management, agent technology, middleware platforms and security.

Synthetic Steganographic Series and Finance

P. C. Ritchey and V. J. Rego

Abstract— **Whenever agent Alice is in a position to send a nontrivially generated data series to agent Bob, Alice is in a position to signal Bob in various ways, including an ability to hide a message for Bob in the series. Such data series may be generated in a variety of situations, including stock prices in any time scale, patterns, textures, tilings, and virtually any processes in which seemingly random values may appear. We focus on stock-price series and demonstrate each step of the data generation and hiding process in detail, using authentic data from the underlying application to generate synthetic data that cannot be compared to data that will naturally be absent in real applications.**

Keywords— **Steganography, TES, Financial, Time Series.**

I. Introduction

In this paper we detail a methodology that enables an agent Alice to embed secret messages in public data that is sent or broadcast to a receiving agent Bob. While we have applied the method successfully in various settings, including patterns, textures and games, we focus here on random series because they (a) yield a powerful class of applications, and (b) require a special treatment. Thus, to begin, we focus on a (random) time series that Alice sends to Bob, and we study a number of steps involving data generation, message hiding, and message retrieval. For completeness, we demonstrate the process through experimentation with real data.

The first question to ask is: where does this time series come from? In a typical setting, Alice works in an application space where complex operations, perhaps involving groups of people and/or multiple transactions, proceed in sequence to generate unpredictable time-series values. To drive home this point, consider that Alice is a market-maker or specialist at a financial exchange where she is in control of a relatively illiquid stock. In such situations it is well-accepted that Alice or accomplices of Alice can exhibit control of price (or rearrange group activity) for periods of time that are sufficiently long to embed secret messages of nontrivial length in given price sequences. The second question to ask is: would it not be readily apparent to data viewers that the “controlled” data generated by Alice is suspect? The answer is no, because Alice can call upon a virtually unlimited history of prior data, under similar environmental conditions, to generate on-the-fly synthetic data that is impossible to distinguish from true data which is necessarily absent because synthetic data operates in its place.

For brevity, we will thus accept that Alice can generate synthetic data at will. We will equip her with a strong methodology to do so. Because we have motivated the financial specialist setting in prior work in great detail, we will simply focus on how Alice generates on-the-fly data that cannot be distinguished from real data that would exist otherwise, and on how she embeds secret information for recipient Bob in this data. Observe that in the case of stock data, Alice generates a stego price series that is broadcast to the entire world via public market services. The methodology will make clear how Bob recovers hidden messages from synthetic data.

The paper is laid out as follows: Section II presents a simple steganographic system for hiding information in an arbitrary series of values. Section III reviews methods for modeling time series data with various distributions. Section IV applies the steganographic system of Section II to the time series generated by the methods of Section III. Section V reviews the TES (transform-expand-sample) method of modeling time series with arbitrary distributions. Section VI applies the steganographic system of Section II to the time series generated by TES.

II. A Simple Steganographic System

We present a systematic method for hiding information in an arbitrary series of values. A secret key stego-system [1] is defined as a quintuple $\mathcal{S} = \langle C, M, K, D_K, E_K \rangle$, where C is the set of possible cover objects, M the set of secret messages with $|C| \geq |M|$, K the set of secret keys, $E_K : C \times M \times K \to C$ and $D_K : C \times K \to M$, with the property that $D_K(E_K(c, m, k), k) = m$ for all $m \in M$, $c \in C$ and $k \in K$.

To specify a stego-system, one must define the sets C, M, K and the functions E_K and D_K. The function E_K is the embedding function which takes as input the cover-object $c \in C$ and the message $m \in M$ to be hidden in c as well as any additional (key) parameters $k \in K$ required to hide the information. It returns a stego-object containing the hidden information. The function D_K is a disembedding function which takes as input a stego-object c' and key k and outputs the message m which is hidden in the stego-object.

For the purposes of this paper, we define the set C to be the set of positive real numbers $\mathcal{R}^+$, and set M to be the set of binary strings of arbitrary length, i.e., $M = \{0, 1\}^n$ for $n \geq 0$. In the remainder of this section we will develop our definitions of the embedding and disembedding functions, E_K and D_K, by starting with a simple function and adding layers of complexity to create a more general function.

A. Embedding Function

A family of embedding methods emerges from a simple example of embedding. To hide a binary string — a representation of some secret information — in time series data, we divide the real number line into cells based on some set of parameters and label these cells alternately as 0 and

P. C. Ritchey, Department of Computer Science, Purdue University, West Lafayette, Indiana, USA, pritchey@purdue.edu

V. J. Rego, Department of Computer Science, Purdue University, West Lafayette, Indiana, USA, rego@purdue.edu

1, beginning by labeling the first cell 0. The information is embedded by allowing the label sequence of the data to match message bits. To do this, align the first bit of the message with the first value of the cover-object and compare each bit of the message with the label of the corresponding data value. If the data label matches the message bit, the data is left unperturbed. If the data label differs from the message bit, however, the data value is minimally augmented so that the label changes. A simple policy for this would be to bump 0s up to 1 and 1s down to 0 by adding ±1 cell-width to the data value.

This stego scheme can have several layers of complexity based on the key (set of parameters) used. We assume the key is exchanged securely prior to stego transmission.

Layer 0: In its simplest form all cell widths are a constant value $w = \frac{1}{\alpha}$, yielding the following embedding function:

$$S'(i) = S(i) + w\,(M(i) - m(S(i), w)) \tag{1}$$

where $m(s, w) \equiv \left\lfloor \frac{s}{w} \right\rfloor \ (mod\, 2)$. The key for the layer-0 stego-system is $k = \{\alpha\}$.

Layer 1: In the next layer of complexity we may allow an offset value equal to some constant δ_0, so that the embedding can begin at any point in the coverdata, not just at the very beginning. This allows us to control the location of the information within the covertext. The embedding function is:

$$S'(i) = S(i) + I_{\{i>\delta_0\}} w\,(z_i(\delta_0) - m(S(i), w)) \tag{2}$$

where $z_i(j) = M(i-j)$ and $I_{\{cond\}}$ is 1 if and only if $cond$ is true and 0 otherwise. The key for this system is $k = \{\alpha, \delta_0\}$.

Layer 2: Once we have the machinery to handle a constant message offset, we can easily handle using only certain values to hide message bits. To do so, we use a monotonically increasing function as a variable offset. It must be monotonically increasing or else the function will try to embed two message bits in the same value, which would cause the first message bit to be lost. We choose whether or not to use the current data value to embed a message bit by tossing a coin that lands heads up with probability p. If this event occurs, we embed the value, and otherwise skip it. Embedding a message bit in the current value does not increase the offset. But, skipping a value does. So, we need to keep track of our offset as we go. We do this with the $\delta(i)$ function defined in Equation 4. This additional layer of complexity allows us to control the distribution of the message within the covertext. The embedding function is now

$$S'(i) = S(i) + I_{\{i>\delta(i)\}} w\,(z_i(\delta(i)) - m(S(i), w)) \tag{3}$$

where

$$\delta(i) = \delta(i-1) + I_{\{r_1(i)>p\}},\ \delta(0) = r_1(0) \tag{4}$$

The key for this system is $k = \{\alpha, p, R_1\}$, where R_1 is a random number generator which generates $r_1(i)$.

Layer 3: An additional (and, for now, final) layer of complexity can be had by allowing cell widths to vary. We accomplish this by using the function $w(i) = 1/r_2(i)$. The embedding function is now:

$$S'(i) = S(i) + I_{\{i>\delta(i)\}} w(i)\,(z_i(\delta(i)) - m(S(i), w(i))) \tag{5}$$

The key for this system is $k = \{p, R_1, R_2\}$, where R_2 is a random number generator to generate $r_2(i)$, which replaces the previously used α.

The above becomes our embedding function E_K, which allows us to specify through the key k: the granularity of the embedding (through the specification of R_2) and the spread of the information through the cover-object (via the specification of R_1 and value of p). This function completely captures the behavior of each of its predecessor layers. To implement Layer-0, let R_2 be a stream with constant value α, set $p = 1$ and let R_1 be any stream who's first value is 0.

B. Disembedding Function

To disembed a hidden message from a data series we need to know the key k. Once we know k, we split the data range into cells based on the parameters used and label them as before, alternating between 0 and 1. The message is the bit string obtained by simply reading off the labels of the data values. Thus, the disembedding function is simply a labeling function.

Unlike the case of the embedding function, where each new layer of complexity further complicates embedding, the disembedding function's form stays very much the same even despite the complexity layers. The disembedding functions are listed below in quick succession as Equations 6-9.

$$M(i) \equiv \left\lfloor \frac{S'(i)}{w} \right\rfloor mod\ 2 \tag{6}$$

$$M(i) \equiv \left\lfloor \frac{S'(i+\delta_0)}{w} \right\rfloor mod\ 2 \tag{7}$$

$$M(i) \equiv \left\lfloor \frac{S'(i+\delta(i))}{w} \right\rfloor mod\ 2 \tag{8}$$

$$M(i) \equiv \left\lfloor \frac{S'(i+\delta(i))}{w(i+\delta(i))} \right\rfloor mod\ 2 \tag{9}$$

C. Good Cell Widths

Through the random number generator R_2, we have control over the range of our cell widths. Cell widths which are too large will result in large data perturbations which may be easily detectable. However, the embedding is fairly tolerant of external augmentation since a large perturbation is needed to bump data values out of their actual labels. Conversely, cell widths which are too small will result in small perturbations which succumb to external augmentation very easily. But here, however, the embedding is difficult to detect since data values exhibit negligible change, and the underlying structure of the original data is preserved.

A good cell width, then, is one which balances two requirements: (1) minimizes the evidence of embedding and

(2) maximizes the tolerance of the embedding to external perturbations. If we know the maximum amount by which data values may be perturbed, we can calculate the minimum cell width we are required to have so that the probability that a value survives the perturbation is at least some value q. The cell width required is given by Equation 10.

$$w \geq \frac{2p_{max}}{1-q}, \tag{10}$$

where $q \in (0,1)$ is the probability that a value survives the maximum perturbation p_{max}.

Using this fact along with $w(i) = 1/r_2(i)$ from above, we can compute an upper and a lower bound for good random variates from R_2, shown in Equation 11 where w_{max} is the maximum cell width allowed.

$$\frac{1}{w_{max}} \leq r_2 \leq \frac{1-q}{2p_{max}}, \tag{11}$$

III. Methods For Modeling Time Series

To illustrate the above ideas, we focus on time series embeddings. There are two main classes of methods used to model time series data: autoregressive (AR) and moving average (MA) models. The choice of model is based on the properties of the data one is trying to model. Autoregressive models are useful in modeling stationary data, while moving average models are better equipped to handle non-stationarity. It is possible to model non-stationary data with an autoregressive model by using differencing. In this section we briefly present the first order autoregressive model and how to use it to model data with various distributions. While many time series can be modeled by a Gaussian process, many naturally occurring time series, however, are non-Gaussian. For this reason, several different methods for generating non-Gaussian time series have been constructed [2]. These include models which utilize exponential [3], [4], Laplace [5] and gamma [6] marginals.

A. Autoregressive Model

The first order autoregressive model is given by Equation 12, where ϕ is the parameter, c is a constant (often $c = 0$ for simplicity) and ϵ_t is the noise term. Note that ϵ_t can also be considered an error or innovation term.

$$X_t = c + \phi X_{t-1} + \epsilon_t \tag{12}$$

Gaussian Autoregressive Model. If the noise term ϵ_t is a Gaussian process, then X_t is also a Gaussian process. The resulting model is the first order Gaussian autoregressive model GAR(1).

Laplace Autoregressive Model. If the noise term ϵ_t is defined

$$\epsilon_t = \begin{cases} 0 & w.p. \quad \phi^2 \\ L & w.p. \quad 1-\phi^2 \end{cases} \tag{13}$$

where L is a Laplace distributed random variable, the sequence X_t is a stationary time series with Laplace marginal distribution for all t [5].

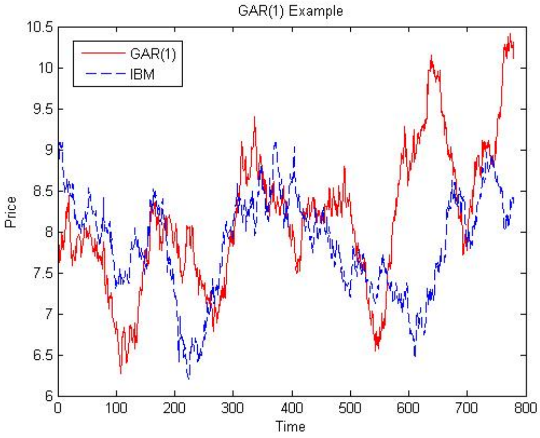


Figure 1. Synthesized GAR(1) for IBM

Exponential Autoregressive Model. The conventional exponential first-order autoregressive model EAR(1) [3] has the form

$$X_t = \phi X_{t-1} + \begin{cases} 0 & w.p. \quad \phi \\ E_t & w.p. \quad 1-\phi \end{cases}, t = 0,1,2,\cdots \tag{14}$$

where $0 \leq \phi < 1$ is a parameter and the $E_t, t = 0,1,2,\cdots$, are independent exponential variables with parameter $\lambda > 0$. This model generates paths in which large values are followed by runs having geometrically distributed lengths of falling values [4].

IV. Steganograhpy with AR Models

We are now ready to present steganographic methods based on synthetic data and, in particular, data generated via the methods of Section III. For ease of explanation we will use the GAR(1) model as our example. Fig. 1 shows an example synthetic data series generated to model an IBM stock price trajectory where prices are reported at end-of-day. The methods apply to any time scale, including tick, minutes, days, weeks and months.

A. Embedding and Disembedding

Embedding in data generated via an AR model such as GAR(1) can be done in at least two ways, which is to say that the message can be hidden in at least two places in the model: the innovations and the final time series. The choice of where to embed decides which information is necessary in the key in order to disembed the message. Embedding in the innovations requires the key to contain ϕ whereas embedding in the final time series does not require ϕ to be known.

Disembedding is accomplished by isolating the portion of the model which was used to hide the message and applying the disembedding function in order to extract the hidden information.

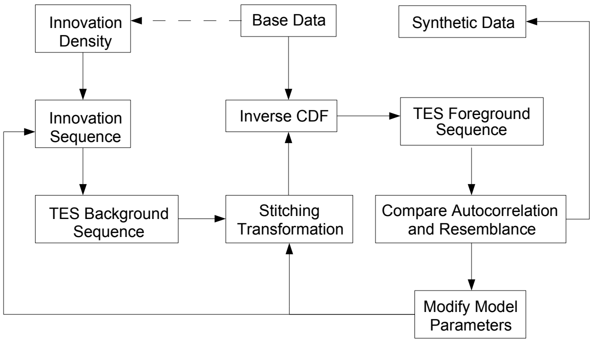


Figure 2. The TES Methodology

V. Modeling Time Series With TES

The TES (transform-expand-sample) [7], [8] methodology is a versatile method for generating a class of stochastic stationary time series which exhibit general marginal distributions and a broad range of dependence structures. TES is designed to generate sequences which satisfy the following three goodness-of-fit requirements [7], in descending order of rigor:

1) The marginal distribution of the sequence should match its empirical counterpart.
2) The autocorrelation of the sequence should approximate its empirical counterpart well.
3) The trajectory of the sequence should "resemble" that of the empirical data.

A. TES Methodology Outline

In [7], the TES methodology is enabled through execution of the follow steps (also shown in Fig. 2):

1) Construct histogram H of the base data
2) Using H, construct the CDF of the base data
3) Using the CDF, invert H to obtain H^{-1}
4) Construct the density f_V of innovation sequence V
5) Choose the sign of the TES class (TES+ or TES-)
6) Generate initial random variate $U_0 \in (0, 1)$
7) Using U_0 and V, generate U
8) Choose stitching parameter ξ
9) Compute background process $Y_i = S_\xi(U_i)$
10) Compute foreground process $X_i = H^{-1}(Y_i)$
11) Compare the distribution and autocorrelation of X with those of the base data. If the two are similar to their empirical counterparts, compare the appearance of X with that of the base data. If X is unacceptable, repeat the above from any previous step; otherwise select X as the final TES process.

In this work, we have used a similar set of steps to implement the TES methodology. Instead of constructing a histogram off the base data, we directly construct the empirical CDF. Also, we only use the TES+ class, though nothing in our methodology precludes the use of the TES- class. The choice to use TES+ was made based on the data we are trying to model. Instead of an innovation density f_V, we use an innovation CDF F_V. Having the innovation CDF is equivalent to having the innovation density and enables a more precise path. While initial parameter values can be set arbitrarily, in this work we have chosen to precompute suitable values — a process that is explained later. Our modified TES methodology is as follows:

1) Construct the empirical CDF F of the base data
2) Choose stitching parameter ξ
3) Construct the empirical CDF F_V of innovations
4) Generate innovation sequence V
5) Compute unstitched background sequence U
6) Compute stitched background process $Y = S_\xi(U)$
7) Compute foreground process $X = F^{-1}(Y)$
8) Compare the distribution and autocorrelation of X with those of the base data. If the two are similar to their empirical counterparts, compare the appearance of X with that of the base data. If X is unacceptable, repeat the above from any previous step; otherwise select X as the final TES process.

In the remainder of this section we discuss our implementation of the TES procedure in more detail.

B. Empirical CDF

The empirical CDF is a 2-dimensional vector which tells us what proportion of the data is less than a given value. The first dimension of the vector is the X-axis and covers the range of the data values. The second dimension contains the proportion of the data which is less than the corresponding X-axis value.

$$F(x) = \frac{1}{|D|} \sum_{d \in D} I_{\{d<x\}} \tag{15}$$

where $I_{\{cond\}} = 1$ iff $cond$ is true and 0 otherwise.

The empirical CDF can be smoothed through the use of interpolation. Since our CDF is a vector, obtaining its inversion is simple. The inverse CDF is the distortion used to transform the background process Y into the foreground process X, i.e., $Y = F(X)$ and $X = F^{-1}(Y)$.

When X is a foreground process, such as the base data, evaluating the CDF at each of the points in X recovers the stitched background process Y. We use this fact later when constructing a good innovation density.

C. Stitching Transformations

The purpose of the stitching transformation S_ξ is to smooth the generated time series while preserving uniformity. The necessity of such a function is apparent when we consider the sequence of values comprising U. Each U_n is the sum of the previous value U_{n-1} and an innovation value V_n taken modulo 1. We may regard U as a random walk around the unit circle. When we cross over the zero-one boundary of the circle, the value drops from a large fraction to a small fraction or vice versa, depending on the direction we cross the boundary. Left as it is, this large jump would result in an even larger jump in the foreground process, say from near the minimum value to near the maximum. In certain cases, this may be acceptable, but in general it is undesirable.

A stitching transformation S_ξ maps the interval $[0,1)$ to itself and is determined by a stitching parameter $\xi \in (0,1)$. For a given ξ, the stitching transformation is defined in [7] as

$$S_\xi(y) = \begin{cases} \dfrac{y}{\xi}, & 0 \leq y \leq \xi \\ \dfrac{1-y}{1-\xi}, & \xi \leq y < 1 \end{cases} \tag{16}$$

While it is possible to invert a stitching transformation, doing so requires additional information beyond the stitched value. Due to the piecewise definition of the stitching transformation, the direction the original value was stitched is required in order to invert it. Therefore, we must record this direction if we wish to invert the stitching transformation in such a way as to obtain the true original value.

$$y = S_\xi^{-1}(x) = \begin{cases} \xi x, & d_\xi(y) = 1 \\ 1-(1-\xi)x, & d_\xi(y) = 0 \end{cases} \tag{17}$$

where $x = S_\xi(y)$ and $d_\xi(y)$ is defined as

$$d_\xi(y) = \begin{cases} 1, & 0 \leq y \leq \xi \\ 0, & \xi \leq y < 1 \end{cases} \tag{18}$$

If the stitching directions are unknown, one of several approaches can be taken: 1) randomly choose the direction, 2) choose a string of directions and repeat it over the length of the data, or 3) choose the direction which results in the smallest difference (mod 1) from the previous unstitched value.

D. Innovation Sequence

The innovation sequence V is a sequence of random numbers drawn from the innovation density f_V. The innovation sequence tries to approximate the underlying pseudo-random process occurring in the original data. The values in the innovation sequence, called innovations, are transition values, i.e., amounts by which a next value in the sequence differs from a previous value. The innovation sequence is used to generate the background process which is then transformed into the foreground process which becomes a candidate for the final TES series.

E. Innovation Density

In [7], [8], the innovation density f_V consists of $K > 0$ non-overlapping regions, called steps, whose positive heights sum to 1. Each step k is therefore represented by a 3-tuple (L_k, R_k, P_k), where L_k and R_k are the left and right endpoints of the step, and P_k is the height. Thus,

$$f_V(x) = \sum_{k=1}^{K} 1_{[L_k,R_k)}(x)\frac{P_k}{\alpha_k}, x \in [-0.5, 0.5) \tag{19}$$

where $\alpha_k = R_k - L_k$ is the width of step k. The innovation density is essentially the desired histogram for the innovation sequence. To generate innovations from the density as defined above, with probability P_k we generate a uniform random number between L_k and R_k.

Since the innovation density is one of two parameters we can use to find the best foreground process, it is important to choose a good one. Our method for finding a good innovation density gives us the empirical CDF of the innovation sequence, which is equivalent to having the density f_V.

F. Background Process

Given the explanation of stitching above, we make a distinction between the stitched background process Y and the unstitched background process U. The TES methodology divides unstitched background processes into two classes, TES+ and TES-. TES+ consists of sequences U_n^+ of the form

$$U_n^+ = \begin{cases} U_0, & n = 0 \\ \left\langle U_{n-1}^+ + V_n \right\rangle, & n > 0 \end{cases} \tag{20}$$

and TES- consists of random sequences U_n^- of the form

$$U_n^- = \begin{cases} U_n^+, & n \ even \\ 1 - U_n^+, & n \ odd \end{cases} \tag{21}$$

The choice of whether to use TES+ or TES- depends on the autoregression of the original data. An oscillatory autoregression is modeled more easily by TES-.

The stitched background process Y is obtained by applying a stitching transformation to U, $Y = S_\xi(U)$.

G. Foreground Process

The final product of the TES methodology is the foreground process X, which should have the same marginal distribution and autocorrelation as the base data and should also "resemble" the base data in its path. The foreground process is generated by applying a distortion to the background process. Conceptually, while this distortion could be any monotonic function, TES uses the inverse CDF of the base data. Since the CDF is a monotonically increasing function, it is always invertible. The equation for obtaining X from the background process Y is $X = F^{-1}(Y)$.

1) Searching for Good Parameter Values: We compute good initial model parameters (F_V and ξ) by viewing the base data as a TES foreground process and reversing the TES methodology to find the innovation sequence. We then use the empirical CDF of the innovation sequence as the CDF of our future innovation sequences. In order to do this, we must know ξ, and so we find a suitable ξ along the way. The process works as follows:

1) Evaluate F at each of the points in the base data, obtaining Y'
2) For each of several candidate values of ξ,
   a) Unstitch Y', obtaining U'
   b) Difference U', obtaining V'
   c) Construct the empirical CDF of V', $F_{V,\xi}$
   d) Generate an innovation sequence V_ξ
   e) Generate an unstitched background sequence U_ξ
   f) Stitch U_ξ, obtaining Y_ξ
   g) Evaluate the empirical inverse CDF at each of the points in Y_ξ, obtaining X_ξ

h) Construct the empirical CDF F_ξ of X_ξ

3) Let $\hat{\xi}$ be the ξ which minimizes $\sum |F - F_\xi|$

4) $F_V = F_{V,\hat{\xi}}$

We use the sum of the absolute values of the differences between the empirical CDF of the base data and that of a foreground process X_ξ generated with stitching parameter ξ as a goodness test for the candidate values of the stitching parameter. Once we know a good ξ, we can compute the empirical CDF of the innovation sequence obtained by differencing the unstitched background process. Other measures of goodness may also used, such as, for example, similarity of autocorrelations and visual resemblance.

When unstitching the background process, since we do not know the original direction the values were stitched (we cannot know because they were not actually stitched), we must make a guess as to what the correct directions to unstitch the values are. There are several ways to do this and we have chosen to unstitch in the direction which results in the smallest difference (mod 1) from the previous unstitched value. This approach keeps the innovation sequence values small and results in a more precise innovation density.

When differencing the unstitched background process, due to the modulo-1 arithmetic used, sometimes this difference will exceed the innovation boundaries, in which case the innovation simply needs to be in the other direction. That is, you need to cross over the zero/one boundary to get to the next value using a valid innovation value as shown in Equation 22.

$$V_n = \begin{cases} U_n - U_{n-1}, & -0.5 < U_n - U_{n-1} < 0.5 \\ U_n - U_{n-1} + 1, & U_n - U_{n-1} < -0.5 \\ U_n - U_{n-1} - 1, & U_n - U_{n-1} > 0.5 \end{cases} \tag{22}$$

VI. Steganography with TES

We are now ready to discuss how the TES methodology can be used for a stego-system based on financial series (i.e., stock market) data.

A. Data-hiding locales

TES allows us to embed a message in several places. However, the choice of where to embed mainly affects the complexity of the key needed to recover the message. The effect on the final foreground process is negligible.

1) Innovation Sequence: If Alice hides the message in the innovation sequence, the stego-V (see Fig.s 7 through 10 for the different possible arrangements) is used to make the BP (background process) which is then stitched and finally transformed into the FP (foreground process). To recover the message, Bob (the receiving accomplice) must take the FP, transform it back into the stitched BP, unstitch the BP and then find the innovation sequence, which contains the message. To do all this, he must know the transformation from BP to FP, the stitching directions, the initial random variate (only really needed for very first bit; but if first bit is agreed to always be garbage, then it isn't actually required by the receiver) and the key to the secret key stego-system.

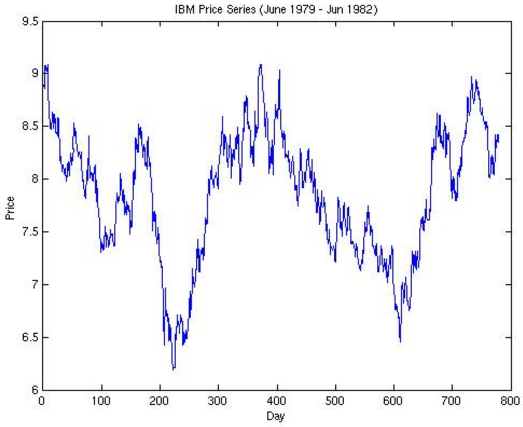


Figure 3. IBM Price Series Data.

2) BP prior to stitching: If Alice hides the message in the BP prior to stitching, the stego-BP gets stitched then transformed into the foreground process. Bob must then know the transformation to obtain the stitched BP. Then, he must also have the stitching directions in order to unstitch the BP. Once unstitched, disembedding the message using the stego-system key is simple.

3) BP after stitching: If Alice hides the message in the BP after it has been stitched, the stego-BP is just transformed into the FP. Bob must then know the transformation to obtain the stitched BP, which contains the hidden message. Again, disemebedding is simple using the stego-system key.

4) FP: If Alice simply hides the message in the FP, our accomplice just disembeds it directly from there and so that he/she only needs to know the stego-system key. As should be clear from the prior explanation, TES is only used synthesize the data.

VII. Experimental Results

We demonstrate the stego methodology through use of an actual IBM stock price trajectory (see Fig. 3) for the period of June 1979 through June 1982. This series becomes our base dataset.

TES is able to match the distribution and autocorrelation of the base data closely, and also produces a trajectory that shows a good visual resemblance to the base data. This is borne out by the close agreement of the CDF curves in Fig. 4, and the virtually identical autocorrelations in Fig. 5.

In general, our experiments show that Alice is able to generate a suitable synthetic data series very rapidly, through either an automatic selection procedure or through visual selection. In Fig. 6, for example, the foreground process that TES offers Alice yields identical second order properties and close visual resemblance to patterns in the base data, which is what Alice wants. Information can be hidden in several places during the TES procedure. The embedding location chosen has a negligible effect on the foreground process.

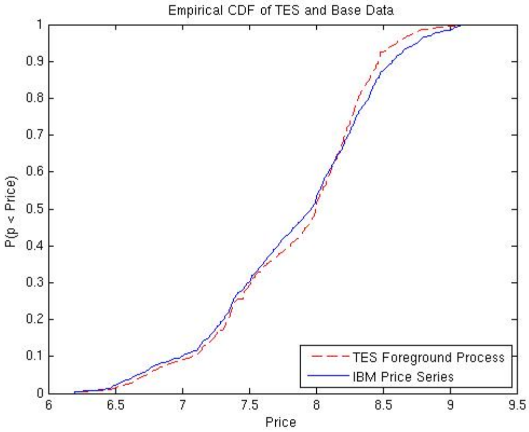


Figure 4. TES Foreground Process and Base Data Empirical CDFs.

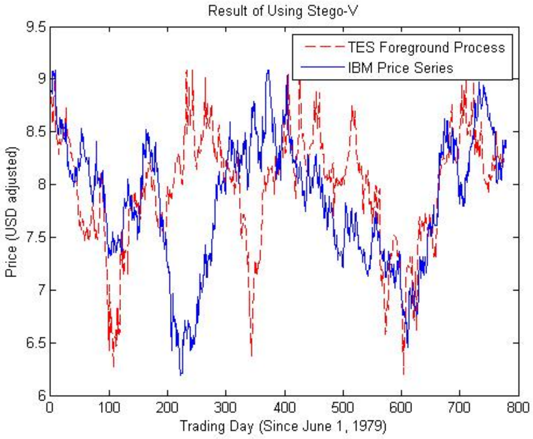


Figure 7. TES Foreground Process Using Stego-V.

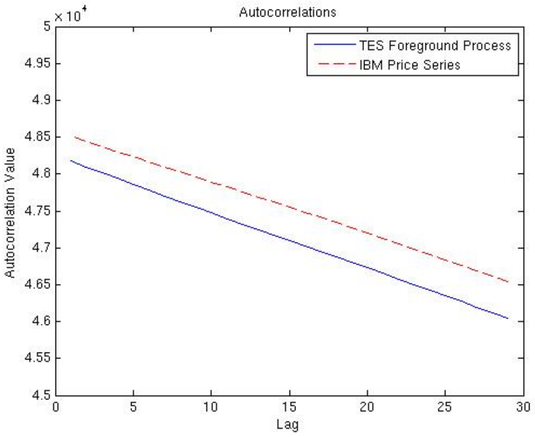


Figure 5. TES Foreground Process and Base Data Autocorrelations.

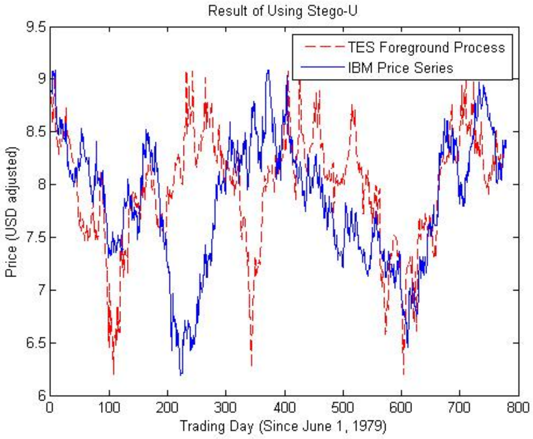


Figure 8. TES Foreground Process Using Stego-U.

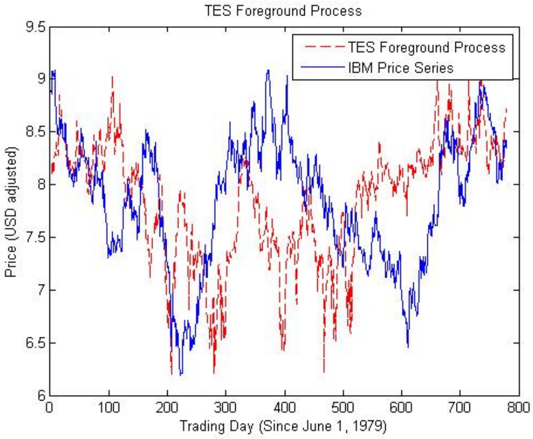


Figure 6. TES Foreground Process with Good Autocorrelation Match.

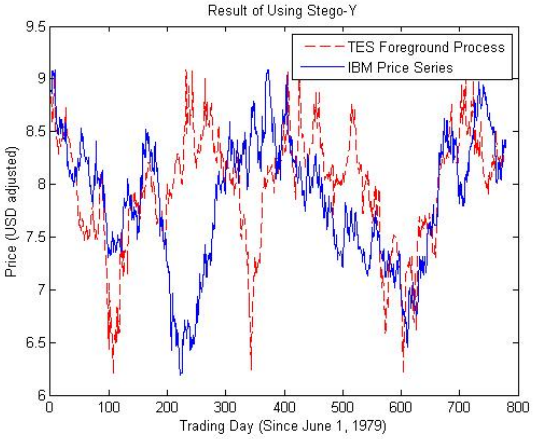


Figure 9. TES Foreground Process Using Stego-Y.

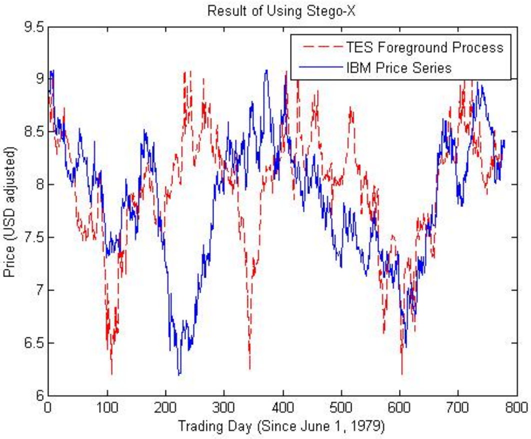


Figure 10. TES Foreground Process Using Stego-X.

VIII. Conclusion

Our experiments have shown that we can develop relatively sophisticated and practical secret-key stego-systems in a variety of applications including the kinds seen in financial markets. Layers of complexity enable information-hiding in an arbitrary series of values with control over information density, location and size of perturbation. We presented the TES methodology for modeling time series, with our implementation of the procedure, outlining the differences between our methods and ones laid out by Melamed [7], [8]. By piggybacking the proposed stego-system on TES, we are able to embed information in price series at various stages of the TES procedure. Our experience is that hiding effects on the final foreground process are negligible, and the main effect of the choice of where to do the embedding shows up in the complexity of the key required to recover the hidden information. In independent but supporting work we have developed theoretical results to show that detection of such stego-systems is virtually impossible under general conditions.

Acknowledgment

This research has been supported by NSF CNS-0716398 and NSF CCF-0939370.

References

[1] S. Katzenbeisser and F. A. Petitcolas, Eds., *Information Hiding: techniques for steganography and digital watermarking*. Artech House, 2000.

[2] M. Novkovic, “On exponential autoregressive time series models,” *Novi Sad Journal of Mathematics*, vol. 29, pp. 97–101, 1999.

[3] D. Gaver and P. Lewis, “First order autoregressive gamma sequences and point processes,” *Advances in Applied Probability*, vol. 12, pp. 727–745, 1980.

[4] A. Lawrance and P. Lewis, “A new autoregressive time series model in exponential variables (near(1)),” *Advances in Applied Probability*, vol. 13, pp. 826–845, 1981.

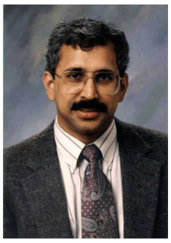
[5] L. S. Dewald and P. A. W. Lewis, “A new laplace second-order autoregressive time-series model - nlar(2),” *IEEE Transactions On Information Theory*, vol. IT-31, pp. 645–651, 1985.

[6] P. Lewis, E. McKenzie, and D. Hugus, “Gamma processes,” *Stochastic Models*, vol. 5, pp. 1–30, 1989.

[7] B. Melamed, “An overview of tes processes and modeling methodology,” in *Performance Evaluation of Computer and Communication Systems*, L. Donatiello and R. Nelson, Eds. Springer-Verlag, 1993, pp. 359–393.

[8] ——, “The empirical tes methodology: Modeling empirical time series,” *Journal of Applied Mathematics and Stochastic Analysis*, vol. 10, pp. 333–353, 1997.

Philip C. Ritchey received the B.S. degree in computer engineering in 2008 from Texas A&M University. He is currently a PhD candidate in Computer Science at Purdue University, where he is also a member of the Center for Education and Research in Information Assurance and Security (CERIAS) and the Center for Science of Information (CSoI). His research interests include information hiding, censorship-resistant technologies, privacy protection, and multi-agent systems.



Vernon J. Rego received the M.Sc. degree in mathematics in 1979 from the Birla Institute of Technology and Science, and the M.S. and Ph.D. degrees in computer science in 1982 and 1985, respectively, from Michigan State University. He is currently Professor of Computer Science at Purdue University, and a faculty member of the Center for Science of Information (CSoI). His research interests include software systems for high-performance distributed computation, parallel stochastic simulation, and software engineering.

Securing Automation Systems Against Malware Intrusion

R. Fitz and W. A. Halang

***Abstract*— Conventional measures do not sufficiently protect computing systems anymore against intruders and malware of any kind. The main reason for this is that the system architectures are based on highly insecure and error-prone foundations. Whereas some time ago this shortcoming could still be partially coped with by swift counteraction, today this "race" must be considered lost right from the start due to the fast data networks. There are no reactive measures anymore that could compensate for the aggressors' temporal advantage. Since computers employed for automation and control purposes are more and more connected to networks and are, thus, endangered by malware, new architectures for their hardware and software as presented in this paper are necessary, which solve the security problems by their intrinsic properties.**

***Keywords*—** *Computer control, automation, safety-related control, security, malware, intrusion prevention, hardware-based security measures.*

I. Introduction

It has become fashionable to employ even for safety-related tasks in automation technology computers whose hardware and software are neither secure against intruders nor able to provide acceptable real-time performance. Thus, to avoid conversions and to minimise times necessary to become acquainted with adequate industrial systems, more and more automation applications are implemented on the basis of cheap PCs as control computers and the popular Windows operating systems. As such computers are swamped with attacks for already some time now, there is a considerable risk also for industrial computing systems to be infected by malware like, for instance, Stuxnet [6] and, thus, to become unsafe. This is exacerbated by the presence of almost any enterprise in the Internet, and since firewalls are unable to protect the intranets of enterprises against external attacks.

Primarily the fast, high-capacity global communication networks and the monoculture in hardware and software technology has led to this situation, which favours the swift spreading of malware. When an electronic intruder was detected in former years, the companies dealing with counteracting them usually had sufficient time to update their products. Owing to the fact that customary software products can provide just a certain degree of protection against already known and analysed electronic malware, most computers are defenceless in the hands of new, not yet sufficiently analysed destructive programs. As there are some tens of thousands new ones of such programs every day according to studies of Bundesamt für Sicherheit in der Informationstechnik (German Federal Agency for Security in Information Technology) [1], some experts now recommend to update the installed "antivirus software" already on an hourly basis, in order to be able to provide, at least, a certain "basic level of protection".

Due to the system homogeneity mentioned above and the increased speed of the proliferation of malware, by now it is a generally accepted fact that trying to warrant security with malware detection programs and firewalls is not an adequate solution anymore. Hence, the security problem must be solved in a fundamentally different way by appropriate architectures of hardware and software. To this end, constructive security measures are presented in this article, which render the virus problem manageable and, thus, contribute to the ultimate solution of this kind of security problems. The feasibility of building systems which can match the contemporary potential of threat will be shown constructively. Moreover, it turns out that such systems can even be maintained more easily as well as can provide higher performance and greater robustness as the automation systems presently prevailing.

An analysis of the various intruders, particularly in form of programs and executable Internet content with malicious intentions, reveals that they are based on some common principles of operation. If these operation principles are thwarted by appropriate measures, malware is prevented from spreading and from launching its destructive effects. The security measures presented in the sequel disable the operation principles of all known malevolent programs in an effective way. In developing them, great importance was attached to the presented solutions being simple and easy to duplicate, in order to be understood and applied without any problems by the users of computers, as unnecessary complexity is the enemy of any effort towards enhancing security.

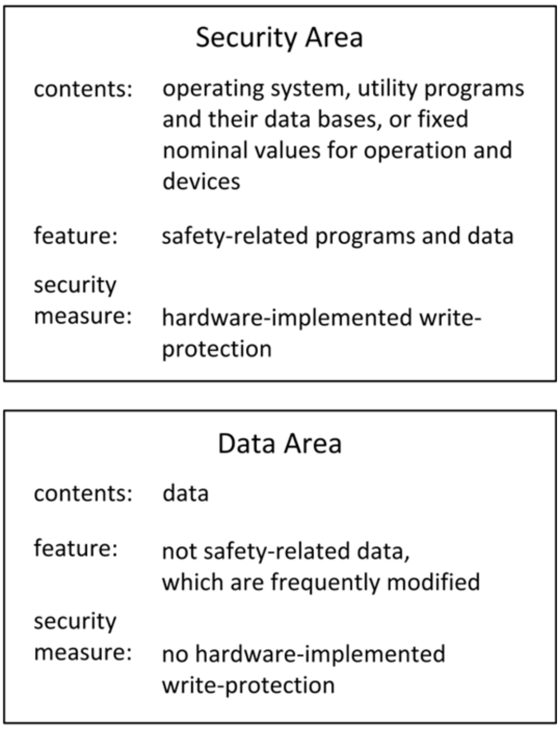
Recently discussed approaches based on cryptography can be ruled out because of their lacking verifiability and unnecessary complexity, in particular for use in automation technology. Their benefit for the users in improving the security of conventional systems is very doubtful in consideration of the fact that there is no practically applicable cryptographic method known which could not be deciphered by attackers – let alone the costs incurred and the performance absorbed by encoding and decoding data. Moreover, in the past experience has shown that cryptographic solutions provoke playfulness, and even persons without malicious intentions feel urged to decipher systems protected this way: a kind of competition or popular sport has emerged.

R. Fitz, Hochschule für Angewandte Wissenschaften Hamburg, Germany, robert.fitz@haw-hamburg.de

W. A. Halang, Fernuniversität in Hagen, Germany, wolfgang.halang@fernuni-hagen.de

II. MEMORY SEGMENTATION

Software with malicious intentions often interferes with application programs or even with operating system routines in order to manipulate them for its destructive purpose or to deactivate software-implemented security functions. Here a memory segmentation measure as developed in [2] takes effect. It reliably prevents unauthorized accesses to the storage areas of operating system and application programs. To this end, a hardware-supervised segmentation of memory is introduced, which protects programs against modifications not permitted. The mass storage of a computing system must, accordingly, be partitioned into at least two segments. At least one of these segments has to be provided with a hardware-implemented write-protection to allow for the storage of safety-related programs and data such as operating system, utility programs and their databases, or fixed nominal values for operation and devices whose failure to be met could lead to the destructions of devices. As shown in Fig. 1, in further segments not protected this way data are stored which, according to experience, are subject to frequent changes. At the same time, these segments can be used to test programs. This protection needs to be ensured throughout all storage levels.



Security Area

contents: operating system, utility programs and their data bases, or fixed nominal values for operation and devices

feature: safety-related programs and data

security measure: hardware-implemented write-protection

Data Area

contents: data

feature: not safety-related data, which are frequently modified

security measure: no hardware-implemented write-protection

Figure 1. Hardware-supervised segmentation of memory.

The more than half a century old and still predominant *Von Neumann architecture* with its minimalistic principles is totally inadequate for systems that need to be safe and secure, as it does not separate *data* from *instructions* and, thus, does not permit to protect both kinds of information in an optimum way (see Fig. 2).

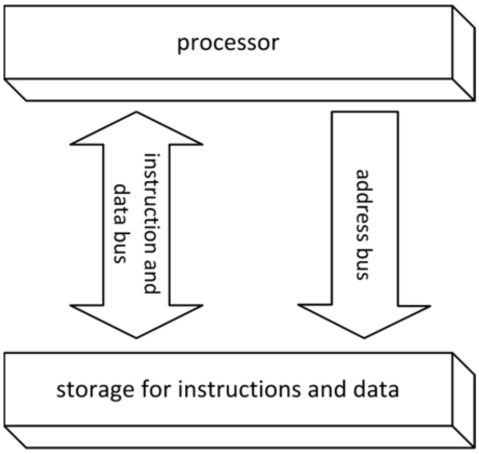


Figure 2. Von Neumann architecture.

The *Harvard architecture* (see Fig. 3), on the other hand, provides this separation throughout and, therefore, represents an adequate construction principle. It is a pleasant side-effect that systems based on this architecture are faster than the currently prevailing ones.

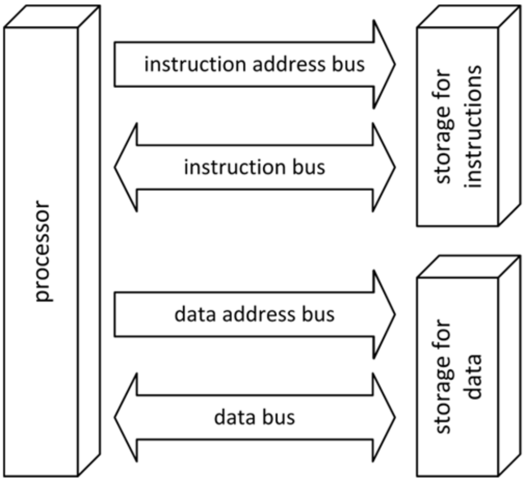


Figure 3. Harvard architecture.

III. CONTEXT-SENSITIVE MEMORY ALLOCATION

In contrast to programs, data are subject to frequent modifications. Therefore, a hardware-implemented write-protection as in [2] is not feasible for reasons of handling. Data can be protected against programs for spying out and modification, however, by a context-sensitive memory allocation according to [3], as shown in Fig. 4. Applying this measure, any unauthorised access to data is precluded. To this end, a system's mass storage, in particular the data area, is further subdivided by a partitioning into context-dependent segments. In an installation mode it is precisely specified which accesses to these segments are permitted to the programs. This is oriented at the data to be protected and not the programs, i.e. in general to each program there exist several data segments separated from one another.

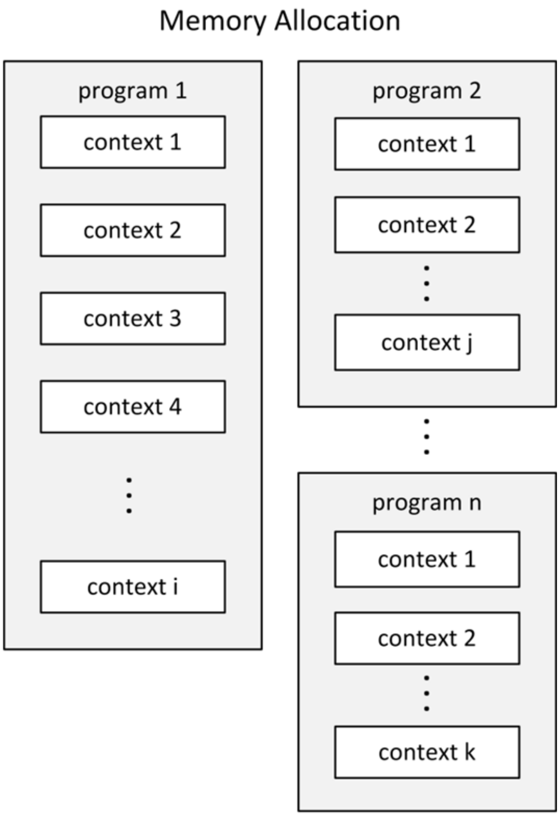


Figure 4. Context-sensitive memory allocation.

In other words, this method is characterised by permitting memory references to any application program and operating system service only by means of using access functions write-protected by hardware, which release the storage areas required for the respective application case for writing and reading or just for reading accesses. Accordingly, in a hardware-protected installation mode the users must establish for any program at least one access function, if they want to use this program in the application mode. As the protection mechanism shall not hinder the users in their daily work and, in particular, shall not hamper the systems' real-time behaviour, the bounds of the memory segments assigned to the different access functions are, for instance, stored in write-protected electrically erasable programmable read only memories (EEPROM), and loaded from there to control accesses to mass storage. Not all admissible memory areas are masked. It suffices to merely supervise the address lines and to control the write or read signals, respectively, of the mass storage media. If an access not permitted is requested, the processor is halted and a signal is generated, which allows the user to uniquely identify the incorrectly working program. For especially endangered programs a variety of access functions should be provided in order to keep the effects of infection by malware as low as possible. Electronic requests arriving from the outside, for instance, always ought to be placed first with their attachments, if any, in a separate and enclosed data area, and processed there.

This way, a spying or modification program that has infiltrated into a data segment without permission can be denied to spread to other segments leaving possible damage narrowly bounded. Based on the segmentation measures presented, a protection against unnoticed modification of data within such a segment can reliably be implemented by already established redundancy measures. Moreover, a finely structured segmentation also protects well against the negative effects of common programming errors, and provides a basis for lucid system maintenance.

IV. HARDWARE-IMPLEMENTED COUPLING OF WRITE-PROTECTION TO AUTHENTIFICATION AND AUTHENTIFICATION-DEPENDENT VIRTUAL ADDRESS SPACE

In order not to endanger the advantages of memory areas write-protected by hardware measures during the installation phases of programs, and to ensure separation on all storage levels throughout, it is necessary to accommodate service programs and their databases also in areas write-protected by hardware and separated from the program area. In doing so, it must be prevented that program and service areas are enabled for writing at the same time, and the memory management must be extended in such a way that the virtual addresses can be used to supervise the computer, since such addresses are linear and, thus, much easier to observe. This is achieved by utilising a hardware device according to [4] generating a write enable signal, which inherently prevents that more than one such signal is generated at a given instant. For this it is necessary to ensure a unique and safe authentification of the user, which is dependent on this person's momentary function, and by means of which the access rights required for the computer's protection are selected. This implies that these systems do not designate omnipotent administrators with rights, which cannot be controlled or are extremely difficult to protect, as they always proved to be a considerable weakness in a vast number of systems under different operating systems. Therefore, almost all attackers seek to gain administrator rights, in order to exercise complete control over a system. This possibility is constructively excluded in the here presented solution, since there is a kind of self-supervision of the correspondingly structured systems at any point in time. Expressed more precisely, hardware-protected and, thus, by software not attackable components of these systems control the respective other parts, even in installation phases. Suitable for user authentification are those methods which cannot be influenced by programs and which are, for instance, based on personal property or biometrical features. To supervise the address space of a computer, safe virtual addresses dependent on authentification and start addresses of page directories are used. The memory management unit is placed between storage and processor to protect the former against direct access by the processor. The unit is equipped with a hardware-implemented protection mechanism, which transmits the required programming signals of the processor in case of correspondingly privileged authentification, only.

V. DISCLOSURE OF REQUIRED RESOURCES

Destructive programs and software-based aggression from the Internet often use components of digital systems, which they would not need for their feigned nominal functions. Here the hardware-supported security measure detailed in [5] takes effect. For instance, for program modules responsible to receive electronic mail the access to communication components must be permitted, but not for those modules which display or even interpret the messages received. This example makes clear that to fulfill their nominal function programs do not need many of the available resources at all or, at least, not all the time, and that permanent release of all resources represents an irresponsible security risk, particularly as common programming errors without malicious intentions cannot be precluded for complex software. On the other hand, users cannot be expected to disable resources on a case by case basis, especially not for automation systems for which this is not possible at all. Therefore, measures need to be devised which protect the users, but do not unduely trouble or restrict them.

All these problems can be solved if any program, any interpretable file and any executable Internet content first discloses which resources it requires for execution. The disclosure of a program's nominal functions enables to install boundary values for systems and to supervise their operation in an effective way. By this supervision and, at any point in time, by locking all resources not needed at that time by means of hardware as shown in Fig. 5, it can be safely warranted that the desired nominal functionality is observed.

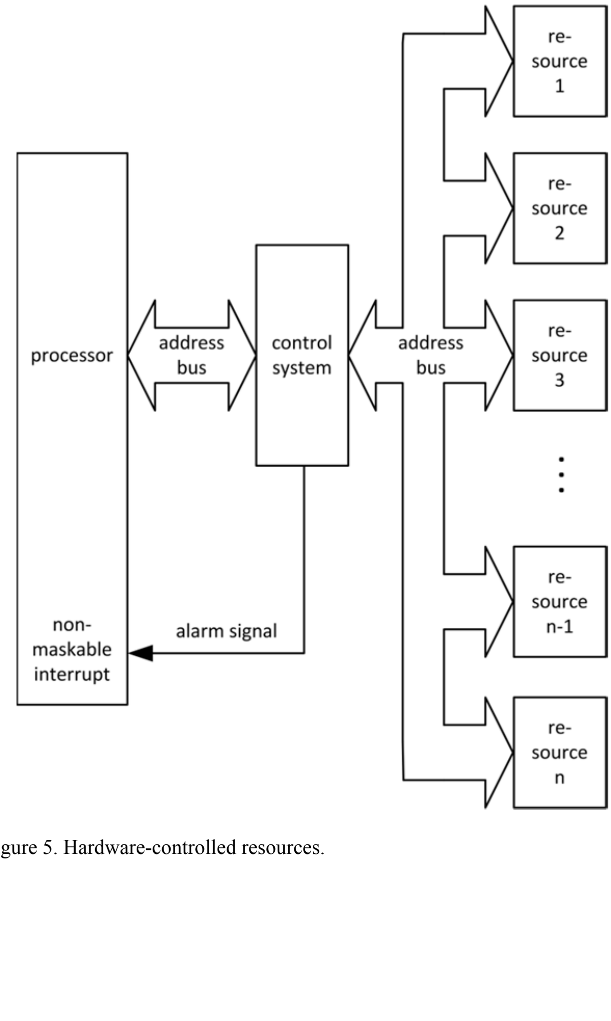


Figure 5. Hardware-controlled resources.

For, in installation modes, during which application software may not access processors, memory nor communication equipment, the users set the limits for resource accesses. Only after that it is possible to execute application programs under permanent hardware-supported supervision based on the constraints defined before. Hereby, not only the resource accesses are supervised, but also the execution times. Thus, the real-time capability is guaranteed. As a positive side-effect, this approach also prevents, up to a certain degree, damage caused by common programming errors without malevolent intentions. Upon deviation from its required nominal function the corresponding program is aborted. All resources seized before are reset and released again. This has the advantage that another program can immediately be put in execution after an illicit action, i.e. the system remains available.

The here presented methods solve the problem of executable Internet contents as well, which is currently of extreme urgency. For, executable Internet contents can be considered as programs for potential spying out and modification, whose program code resides on remote computers. To cope with them, the following procedure is to be adhered to.

- Before a program stored on a different, remotely located computer may become active, it must first provide information about its nominal functionality and the resources required for this.
- If the intended activities are considered uncritical, the execution is initiated without asking the users unnecessary questions. What hereby is regarded as uncritical was defined before by the respective users themselves, and stored in an area write-protected by hardware. Since a program's alleged activities are securely supervised in any case, the credibility of communication partners is not of such a decisive importance for a computer's security as it is the case for the currently prevailing solutions. Confidential information as exchanged, for instance, in electronic commerce is secured and encrypted for transmission here as well.
- If the data disclosed indicate critical functions, the further proceeding depends on whether there is already a certain trust in the source of the data, and which actions were permitted to it. In case the actions requested are within the framework already authorised, also here there is no feedback to the users. If the range of actions of an application or a data source is to be extended, first the users disconnect the communication links to extend the conceded framework of actions, and resume the connection to the communication partners not before the supervisor data and all resources not required have been hardware-protected against unauthorised access.

This solution by far outperforms established methods such as, for instance, the trust-based one of “ActiveX” or the “sandbox” method of “Java”, as decisions can be made on the basis of a much finer granularity, without imposing on the users unreasonable restrictions or urge them to admit everything.

VI. CONCLUSION

To be secure, automation systems must fulfill the following requirements.

- Data and instructions have to be separated throughout.
- Authentifications may not be influenceable by software.
- Protection systems may not be attackable themselves. This means that their implementation must be proven correct and safely protected against modifications not permitted.
- The protection of systems may not be put out of effect during the installation phases of application programs or of operating system components as well.
- All storage levels (main memory, mass storage etc.) have to be protected throughout against unauthorised accesses by means of authentification-dependent virtual address spaces.
- Constraints and nominal functionalities of programs are defined in installation phases, and permanently supervised in the course of operation. Their observance is guaranteed even under real-time conditions.
- To protect data against effects of common program errors or malicious interpretable files and to enable context-sensitive memory allocation, a means for the instantiation of programs must be provided, which employs access functions.

Utilising the presented measures industrial computer control systems are effectively protected against inadmissible accesses. This holds in particular for still unknown attack patterns or malware, too, because there is no more need for databases of malicious code or attack prototypes, which become obsolete within hours anyway due to the swift spreading of current malware via the Internet. It has been shown that it is possible to build systems which are immune against intruders and espionage. In addition, it was shown that separation and structuring considerably facilitates the maintainability of computer control systems, too, and even increases their performance. Furthermore, it became clear that systems protected by the above mentioned measures exhibit, on the basis of disclosing their nominal functions, of the permanent supervision against set bounds, of the context-sensitive allocation of data and of the impossibility to attack operating systems and application programs, a degree of *robustness* which allows them to maintain their functionality despite some failing application programs – a property being of fundamental importance for automation systems and highly safety-critical applications.

The measures presented here guarantee, with reference to [7], the observance of the *protection objectives*

1. **Privacy:** unauthorised gain of information is made impossible, i.e. spying out of data is obviated,
2. **Integrity:** unauthorised modification of information is precluded,
3. **Availability:** unauthorised influence on the functionality is precluded and
4. **Attributability:** at any point in time the responsible persons can be identified with certainty.

REFERENCES

[1] “Der Schädlings-Flut Herr werden”, *Bundeswehr aktuell*, 48(4)5, 30 January 2012, on-line: s337251796.online.de/2012/KW4/html/10005.html

[2] W.A. Halang and R. Fitz, “Speichersegmentierung in Datenverarbeitungsanlagen zum Schutz vor unbefugtem Eindringen”, German patent registration DE10031212A1, 2000

[3] W.A. Halang and R. Fitz, “Kontextsensitive Speicherzuordnung in Datenverarbeitungsanlagen zum Schutz vor unbefugtem Ausspähen und Manipulieren von Daten”, German patent registration DE10031209A1, 2000

[4] W.A. Halang and R. Fitz, “Gerätetechnische Schreibschutzkopplung zum Schutz digitaler Datenverarbeitungsanlagen vor Eindringlingen während der Installationsphase von Programmen”, German patent 10051941 since 20 October 2000

[5] W.A. Halang and R. Fitz: “Offenbarendes Verfahren zur Überwachung ausführbarer oder interpretierbarer Daten in digitalen Datenverarbeitungsanlagen mittels gerätetechnischer Einrichtungen. German patent registration DE10055118A1, 2000

[6] R. Langner, “Stuxnet: Dissecting a Cyberwarfare Weapon”, *IEEE Security & Privacy*, 9(3)49–51, 2011

[7] K. Rannenberg, A. Pfitzmann and G. Müller, “Sicherheit, insbesondere mehrseitige IT-Sicherheit”, in: *Mehrseitige Sicherheit in der Kommunikationstechnik*, pp. 21–29, Bonn: Addison-Wesley 1997.

Robert Fitz, born 1965 in Villingen-Schwenningen, Germany, received BSc and MSc degrees in electronic engineering in 1988 and 1999, respectively, and was awarded Dr.-Ing. in electrical and computer engineering in 2001. After working in industry and academia, he was appointed professor of electronic engineering and computer science in 2002. His current research interests are systems on chip as well as availability and security of data processing systems.

Wolfgang A. Halang, born 1951 in Essen, Germany, received doctorates in mathematics (1976) and computer science (1980). After working in industry and academia, he was appointed chair of computer engineering and department head at the University of Groningen in the Netherlands, and 1992 at Fernuniversität in Hagen. He was co-director of the 1992 NATO Advanced Study Institute on Real-Time Computing and visiting professor in Maribor, Slovenia, and Rome, founded the journal Real-Time Systems, is member of four further journals' editorial boards, (co-)authored 40 books and 400 refereed publications, holds 20 patents, gave 80 guest lectures worldwide, and is active in various technical committees and 240 programme committees.

Isomorphism Theorem and Cryptology

R. L. de Carvalho and F. L. de Mello

Abstract— **This paper presents a Theory of Computation study based on recursive functions computability and innovates by performing parallels to relevant themes of Cryptography. Hence, it is presented the Hennie's notion of "abstract family of algorithms" (AFA, for short) according to the authors' understanding, and also more judicious theorems demonstrations, many times completely different from those ones available in literature. The main issue is the Isomorphism Theorem which supports the Church-Turing Thesis and provides a connection between Cryptology and Linguistics.**

Keywords— **Algorithm, Recursive functions, Church-Turing Thesis, Fixed point theorem, Recursion theorem, Isomorphism theorem.**

I. Introduction

THE CODIFICATION is based on conversion rules whose objective is to transform a piece of information, a message, into a new representation of the same information. Under the Cryptology point of view, this concept remains valid, but it is added to it a strong constraint associated to the intention of keeping the message content restricted to an entity group, and obscured to everyone else. Thus, the informative content representation is transformed by an algorithm, producing a symbol sequence which belongs to the new depiction universe. Those transformation procedures are known by the Theory of Computation as Transductive Formal Systems [7].

Algorithms that convert a sequence of symbols into a new one, preserving the informative content, are known as compilers and interpreters. Besides the existence of important differences between compilers and interpreters, both of them have the responsibility of performing a language translation. Usually, this kind of translation is done from a high-level language to a low-level one. However, there is no obstacle against a translation performed on the other way round, or even from high-level to high-level, or from low-level to low-level.

Similarly, a sequence of symbols from a language plaintext is converted by a encoding algorithm into a new sequence from a cryptographic language. By this reason, it possible to be more specific by asserting that cryptographic algorithms translate a plaintext message to other languages such as Triple-DES, Blowfish, SEAL, MD5, among others. This approach is unconventional in Cryptology. Nevertheless, the existence of terms such as alphabet, communication, dictionary, corpus and corpora, available not only on Cryptography studies, but also on general Linguistics studies and on Theory of Computation, suggests an obtainable intersection regions among these knowledge areas.

On Theory of Computation, the Church-Turing Thesis states the direct relationship between algorithms and languages. It provides a correlation between the act of calculating and the algorithms materialization, that is, the computer programs. The calculus is the execution of methodic sequence actions, and the programs representation is provided by the language.

Thus, this article aims to use Theory of Computation concepts so as to produce a deeper comprehension about the algorithms and the objects to be represented. This approach provides a rigorous understanding about the subject and suggests a correlation between Cryptology and Linguistics.

II. Church-Turing Thesis

The Church-Turing Thesis, shown in Fig. 1, is not a theorem but an epistemic result which acceptance is almost universal. The intuitive concept of computable is associated to a class of arithmetic functions called recursive functions [1]. It can be reasoned that if a hypothesis cannot be directly proved, then maybe it can be refutated. Consequently, in order to deny the thesis it is sufficient to find out just one procedure that cannot be demonstratively computed by a Turing Machine. This procedure has not been found so far, and more over, because there are a considerable number of favorable experimental data, researches tend to accept the thesis. In addition, several attempts to specify the algorithm concept resulted into formalisms that can be demonstrated as equivalent to the Turing Machine.

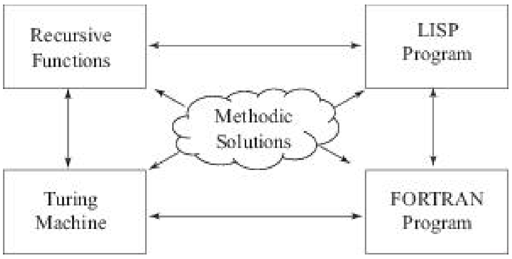


Figure 1. Illustrative scheme from Church-Turing Thesis suggesting that all formalisms to define an algorithm are supposed to be equivalent among themselves.

Therefore, the Church-Turing Thesis is a hypothesis on the mechanical nature of the act of calculating, describing a direct relationship to the computer, and to the different types of algorithms that can be executed by a machine. Thus, every function considered to be systematic can be computed by a Turing Machine. Any kind of programs can be translated to a Turing Machine, and also, any Turing Machine can be translated to a programming language. Consequently, any ordinary programming language is sufficient to represent any algorithm, whatever is its purpose.

R. L. de Carvalho (Ph.D.), Witty Group leader for artificial intelligence and knowledge visualization fields, rlins@globo.com

F. L. de Mello (D.Sc.), Assistant Professor at Electronics and Computation Engineering Department from Polytechnic School at Federal University of Rio de Janeiro, fmello@del.ufrj.br

Invited Paper

III. Algorithms

An algorithm a description of a function calculus or evaluation performed in a systematic way. The main elements associated to this idea are enumerated as follows:

1) *Finite size instruction set*: an algorithm must be described by a language into a finite fashion. Assuming an enumerable alphabet, the algorithm must be composed by a finite string over this alphabet.
2) *Algorithm domain*: composed by data, the set objects processed by an algorithm, for instance, a chain of symbols, natural numbers, etc.
3) *Computer agent*: the description computation results into a well defined sequence of operations or steps, that depends on an agent associated to the algorithm. This agent must be deterministic, that is, he should react to the algorithm instructions forever in the same way. For each input, the algorithm must have always the same behavior: if it halts, it will have always the same output; if it does not halt, it diverges.
4) *Facilities to execute, store and retrieve steps*: the intuitive notion of memory emerges as an agent resource. The maximum size of the memory, and consequently the input size of the algorithm, is an aspect to be taken for each computer agent. However, once this limit can be indefinitely extended, its existence is not considered.
5) *Agent capability*: by using a limited set of skills, the agent must be capable of computing any algorithm.
6) *The end of computation*: once all instructions have been executed, the agent provides the appropriate results, according to the input arguments of the domain. This does not mean that the algorithm should halt to any domain input. For instance, the natural numbers division can indefinitely operate if the dividend is not divisible by the divisor.

One of the first models of abstract machine, as an attempt to define an algorithm, was the Turing Machine. The following excerpt is a abridgement of Alan Turing understanding of what is a computer [12]. The reader must be aware that Turing wrote this text before the invention of the machine called computer, but it is really fascinating how it is still acceptable and precise.

Computing is normally done by writing certain symbols on paper[1]*. We may suppose this paper is divided into squares like a child's arithmetic book. [...] it will be agreed that two-dimensional character of paper is no essential of computing. [...] I shall also suppose that the number of symbols which may be printed is finite. [...] The behavior of the computer at any moment is determined by the symbols which He is observing, and his "‘state of mind"’ at that moment. We may suppose that there is a bound B to the number of symbols or squares which the computer can observe at one moment. If he wishes to observe more, he must use successive observations. We will also suppose that the number of states of mind which need be taken into account is finite. [...] Let us imagine the operations performed by the computer to split up into "‘simple operations"’ which are so elementary that it is not easy to imagine them further divided. Every such operation consists of some change of the physical system consisting of the computer and his tape. [...] We may suppose that in a simple operation not more than one symbol is altered. [...] Besides these changes of symbols, the simple operations must include changes of distribution of observed squares. The new observed squares must be immediately recognizable by the computer. I think it is reasonable to suppose that they can only be squares whose distance from the closest of the immediately previously observed square does not exceed a certain fixed amount. [...] The operation actually performed is determined, as has been suggested, by the state of mind of the computer and the observed symbols. In particular, they determine the state of mind of the computer after the operation is carried out.*

***Definition** 1:* A Turing Machine is a quintuple $M = \langle k, \Sigma, \delta, s, F\rangle$ where [4]:

- k is the finite set of STATES;
- Σ is an ALPHABET which contains symbols $\rhd$ and $\sqcup$, but doesn't contains $\rightarrow$ and $\leftarrow$;
- $s \in k$ is the INITIAL STATE;
- $F \subseteq k$ is the set of HALTING STATES;
- δ is a TRANSITION FUNCTION of $k \times \Sigma$ where: (a)
  1) for all $q \in (k - F)$, if $\delta(q, \rhd) = (p, b)$, then $b = \rightarrow$;
  2) for all $q \in (k - F)$ and $a \in \Sigma$, $a \neq \rhd$, if $\delta(q, a) = (p, b)$, then $b \neq \rhd$.

□

IV. Recursive Functions

One usual approach to define a mathematical function is the *recursive definition*: some initial function values are defined and the other ones are computed based on the priors. It is essential to understand the recursive functions computational model that is presented at this section. On the other hand, the reader who is acquainted with this subject may proceed to section VI without prejudice to the understanding of this work.

The recursive definition method is used to characterize the primitive recursive function class. First, some initial functions are defined, whose simplicity suggests their unconditional computability. These functions are described as follows [1][4]:

Successor: $\underline{suc} : \mathbb{N} \rightarrow \mathbb{N}$, so that for all $x \in \mathbb{N}$

$$\underline{suc}(x) = x + 1$$

Zero: $\underline{zero} : \mathbb{N} \rightarrow \mathbb{N}$, so that for all $x \in \mathbb{N}$

$$\underline{zero}(x) = 0$$

Projection: for each $n > 0$ and each $1 \leq i \leq n$, $\underline{pr}_i^n : \mathbb{N}^n \rightarrow \mathbb{N}$, so that for all $\underline{x}_n =< x_1, x_2, \cdots, x_n >\in \mathbb{N}^n$

$$\underline{pr}_i^n(\underline{x}_n) = x_i$$

Subsequently, it is necessary to define a procedure responsible for describing functions by using the previous ones as support, on the early case, the initial functions. Assume the functions $f : \mathbb{N}^m \rightarrow \mathbb{N}$ and $g_1, \cdots, g_m : \mathbb{N}^n \rightarrow \mathbb{N}$

[1] This is a 1937 text.

exists. The composition h of f and $g_1, \cdots, g_m$ is the function $h : \mathbb{N}^n \to \mathbb{N}$, defined by [1][4]:

$$h(x_1, \cdots, x_n) = f(g_1(x_1, \cdots, x_n), \cdots, g_m(x_1, \cdots, x_n))$$

The composition of a function f with other function g is usually denoted by $f \circ g$.

At last, let $f : \mathbb{N}^n \to \mathbb{N}$ and $g : \mathbb{N}^{n+2} \to \mathbb{N}$. It is said that $h : \mathbb{N}^{n+1} \to \mathbb{N}$ is defined by a primitive recursion if the h values are obtained by [1][4]:

$$\begin{array}{lcl} h(0, x_1, \cdots, x_n) & = & f(x_1, \cdots, x_n) \\ h(y+1, x_1, \cdots, x_n) & = & g(y, h(y, x_1, \cdots, x_n), x_1, \cdots, x_n) \end{array}$$

The sum of natural numbers, for instance, can be defined as follows:

$$\begin{array}{lcl} 0 + x & = & x \\ (y+1) + x & = & (y+x) + 1 \end{array}$$

So, by taking: $f = \underline{pr}_1^1 : \mathbb{N} \to \mathbb{N}$ and $g = \underline{suc} \circ \underline{pr}_2^3 : \mathbb{N}^3 \to \mathbb{N}$, obtained by composition of the initial function, it is possible to write:

$$\begin{array}{lcl} h(0, x) & = & f(x) = \underline{pr}_1^1(x) \\ h(y+1, x) & = & g(y, h(y, x), x) = \underline{suc}(\underline{pr}_2^3(y, h(y, x), x)) \end{array}$$

Notice that for all x and all y, $h(x, y) = x + y$, therefore h is the natural numbers addition. Moreover, h was obtained by composition and primitive recursion based on the initial functions. So, to calculate the sum of two numbers, such as 3 and 5, the computation is given by:

$$\begin{array}{lcl} h(0, 5) & = & 5 \\ h(1, 5) & = & \underline{suc}(h(0, 5)) = \underline{suc}(5) = 6 \\ h(2, 5) & = & \underline{suc}(h(1, 5)) = \underline{suc}(6) = 7 \\ h(3, 5) & = & \underline{suc}(h(2, 5)) = \underline{suc}(7) = 8 \end{array}$$

Therefore, a function $f : \mathbb{N}^n \to \mathbb{N}$ is *primitive recursive* if f is a initial function or if it is obtained by using the composition and the primitive recursion based on the initial functions.

In order to prove that a function $f : \mathbb{N}^n \to \mathbb{N}$ is primitive recursive, it is sufficient to show its informal definition as the one used for the sum. Thus, for example, the product of natural numbers is defined by:

$$\begin{array}{lcl} 0.x & = & 0 \\ (y+1).x & = & y.x + x \end{array}$$

And there is no need to write

$$\begin{array}{lcl} .(0, x) & = & \underline{zero}(x) \\ .(y+1, x) & = & +(\underline{pr}_2^3(y, .(y, x), x), \underline{pr}_3^3(y, .(y, x), x)) \end{array}$$

which is the strict formal definition using primitive recursion.

Nevertheless, not all functions define by recurrent schemes are primitive recursive. As an example, assume the Ackermann function [1][4]:

$$\begin{array}{llcl} a.1 & a(0, y) & = & y + 1 \\ a.2 & a(x+1, 0) & = & a(x, 1) \\ a.3 & a(x+1, y+1) & = & a(x, a(x+1, y)) \end{array}$$

Notice that this definition uses a recursion scheme, and that for all naturals m and n it is always possible to compute $a(m, n)$. By computing $a(1, 2)$ it is obtained the value 4, but the definition of the Ackermann function is quite different from the primitive recursive definition. It is interesting to observe that the general recursion scheme may define functions that are computable for a certain instances, but eventually they diverge.

A detailed study of general recursive schemes, which are different from the recursive primitive recursion, is not the purpose of this article. The existence of computable functions defined by those schemes, and that the general recursive schemes may define computable but divergent functions, suggests that besides composition and primitive recursion, it is necessary to define an additional functional scheme. This need is fulfilled by the *minimization* scheme, responsible for producing total functions.

Let $f : \mathbb{N}^{n+1} \to \mathbb{N}$ be a total function. The $h : \mathbb{N}^n \to \mathbb{N}$ is defined by minimization of f if and only if the values of h are obtained by [13]:

$$h(x_1, \cdots, x_n) = \begin{cases} y & \text{if } f(y, x_1, \ldots,) = 0 \text{ and if} \\ & \forall i < y, f(i, x_1, \ldots, x_n) \neq 0 \\ diverges & \text{otherwise} \end{cases}$$

The minimization of f is denoted by:

$$h(x_1, \cdots, x_n) = \mu y(f(y, x_1, \cdots, x_n))$$

The function $f : \mathbb{N}^n \to \mathbb{N}$ is called partial recursive if f is an initial function or if it is obtained by the usage of composition, primitive recursion and minimization from the initial functions. If f is also a total function, it is simply stated that f is a recursive function.

V. The Gödel β Functions

The programming languages include certain programming facilities in order to manipulate vectors, matrixes, etc. Likewise, the Gödel β functions are introduced as an mathematical utility for manipulating tuples. This functions transform tuples into natural numbers, and are also known as pairing functions. Shall define the primitive recursive functions $\beta : \mathbb{N}^2 \to \mathbb{N}$, ${}_1\beta_2 : \mathbb{N} \to \mathbb{N}$ and ${}_2\beta_2 : \mathbb{N} \to \mathbb{N}$ by:

$$\begin{array}{rcl} \beta(x, y) & = & x + \frac{(x+y).(x+y+1)}{2} \\ {}_1\beta_2(\beta(x, y)) & = & x \\ {}_2\beta_2(\beta(x, y)) & = & y \end{array}$$

The β function provides a linearization of the ordered pair according to sequence of Fig. 2 such as it is performed by space filling curves. Besides, the functions ${}_1\beta_2$ and ${}_2\beta_2$ implement the inverse operation of this linearization.

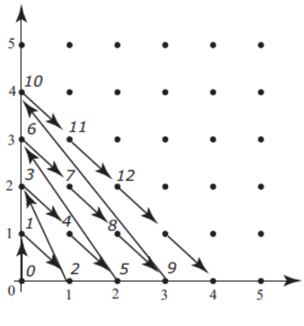


Figure 2. Gödel β pairing function.

Let $\beta_r : \mathbb{N}^r \rightarrow \mathbb{N}$ be inductivelly defined by:

$$\begin{array}{rcl} \beta_1(x) & = & x \\ \beta_2(x_1, x_2) & = & \beta(x_1, x_2) \\ \beta_{k+1}(x_1, \ldots, x_{k+1}) & = & \beta(\beta_k(x_1, \ldots, x_k), x_{k+1}),\ k \geq 2 \\ \bar{\beta}(\underline{x}_m, \underline{y}_n) & = & < x_1, \ldots, x_m, y_1, \ldots, y_n > \end{array}$$

VI. Abstract Family of Algorithms (AFA)

Each programming language is conceived to be applied to a specific problem type, and for this reason, those programming languages provide several features oriented to their purpose. However, the Church-Turing Thesis states that all languages are equivalent, and as a consequence, any kind of programming language may reach a problem solution whatever is the problem nature since this solution exists. In fact, an algorithm may be implemented by several programming languages, and as a result, all tasks resolved by the algorithm must also be resolved by its implementations [2]. Hence, the programming languages are materializations of algorithms, responsible for computing functions.

An algorithm is a set of deterministic procedures which are applied to a symbolic input class that eventually may result, for each input, a symbolic output [1]. Note that an algorithm is always finite even though it execution not necessary is finite. This happens because the algorithm is a symbol string of an alphabet Σ.

The set of all possible symbols combination, regardless the generated string size, is known as Σ^*. So, this combination produces non-meaning string, but it also produces the family of all algorithms A describable by using alphabet Σ. Fig. 3 highlight that for each algorithm there is a natural number m associated by a biunivocal relationship ε_A. The relationship ε_A is a primitive recursive function that enumerates all the representable algorithms using alphabet Σ, and so each algorithm is indexed by a $m \in \mathbb{N}$.

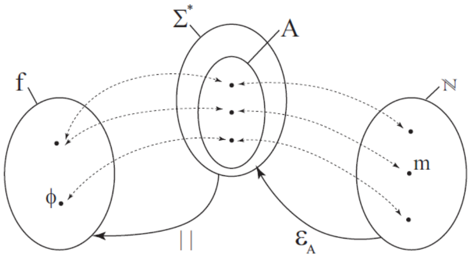


Figure 3. Relationships associated to an Abstract Family os Algorithms.

In addition, each algorithm A has a semantic meaning that is the consequence of its execution, denoted by the operator $|\ |$. It is the computation ϕ of an algorithm $A_m \in A$, and ultimately, it is the computation related to index m of an enumeration because $\phi_m = |A_m| = |\varepsilon_A(m)|$.

An AFA, described by a language L with a finite alphabet Σ, is an enumerable set where each algorithm is associated to a natural number, its index, and a partial function. Given an algorithm of this AFA, it is possible to obtain its index, and given a natural number, it is possible to obtain the algorithm associated to this index. For a given enumerable infinite alphabet, the enumeration must bem more sophisticated, such as the Gödel enumeration [5], based on prime numbers. However, considering the programming languages, the finite alphabet constrain is always applicable.

In spite of the biunivoque relationship ε_A, the relationship $|\ |$ allows the association between one element of f with more than one element of A, which means that may exist more than one algorithm capable to perform a unique task. This set f is composed by partial recursive functions [3][4], that is, functions built by using the initial functions, primitive recursive functions, primitive recursion, minimization and repetition.

***Definition** 2 (Abstract Family of Algorithms):* An AFA is a triple $\phi =< A, \varepsilon_A, |\ | >$ where:

A: is an enumerable set of objects called algorithms.
ε_A: is an function of $\mathbb{N}$ in A called ϕ enumeration, where $\varepsilon_A(\mathbb{N}) = A$.
$|\ |$: is an function of A in $\mathbb{N}^{\mathbb{N}}$ called computation.

with the following properties:

P_1 For all partial recursive functions ϕ, $f : \mathbb{N} \rightarrow \mathbb{N}$ exits $m \in \mathbb{N}$ where $\phi_m = |\varepsilon_A(m)|$. When it is applied to an string x, it is written $\phi_m(x)$;
P_2 Exists a natural number $u_{\mathcal{F}}$ which indexes $\phi_{u_{\mathcal{F}}} = |\varepsilon_A(u)|$ where if x is an input data string of an algorithm, then $\phi_{u_{\mathcal{F}}}(n, x) = \phi_n(x)$;
P_3 Exists a natural number $c_{\mathcal{F}}$ which indexes $\phi_{c_{\mathcal{F}}} = |\varepsilon_A(c_{\mathcal{F}})|$ where if n and m are an input data string of an algorithm, then $\phi_{c_{\mathcal{F}}(n,m)} = \phi_n \circ \phi_m$. When it is applied to a string x, it is written $\phi_{c_{\mathcal{F}}(n,m)}(x) = \phi_n \circ \phi_m(x)$;

The property P_1 indicates that all partial recursive functions are computed by an algorithm from the AFA $\mathcal{F}$ indexed by a natural number m. The property P_2 is important because it concerns to the universal function of $\mathcal{F}$ whose execution is the appliance of the algorithm of index n for the input data x. Note that the function $\phi_{u_{\mathcal{F}}}$ is associated to an algorithm that executes algorithms, a meta-algorithm, sometimes called universal algorithm. Despite this concept is special, the meta-algorithms are not rare but quite common, such as compilers and operating systems. Finally, the composition property P_3 aims to obtain, from two programs ϕ_n and ϕ_m, a new program ϕ_{nom}, and to accomplish this task it is necessary that the function $\phi_{c_{\mathcal{F}}}$ provides an modification on ϕ_n and ϕ_m in such way to avoid conflicts between the names of the variables and others tags. Thus, it isolates the algorithm scopes, forbidding their overlap.

The following theorem is known as the Parameterization Theorem or the s-m-n Theorem [1][11][9], and it has a great importance recursive functions theory. It reflects the possibility of packing function arguments in order to obtain programs that use the mechanism of argument passing to increase their modularity and reduce their coupling.

Let a function ϕ with $m + n$ arguments. Suppose that the first m arguments are fixed and the other n arguments allowed to change, resulting into a ϕ function with n arguments. The index of this functions depends on the index of the original function ϕ and the m arguments $x_1, \ldots, x_m$.

***Theorem** 1 (s-m-n Theorem):* For any acceptable $\phi_0, \phi_1, \ldots$, there is a total recursive function $s : \mathbb{N}^{m+1} \rightarrow \mathbb{N}$, where for all values $e, x_1, \ldots, x_m, y_1, \ldots, y_n$, with $m, n \geq 1$, we have:

$$\phi_{s(e,x_1,\ldots,x_m)}(y_1, \ldots, y_n) = \phi_e(x_1, \ldots, x_m, y_1, \ldots, y_n)$$

By taking $m = n = 1$, it is obtained the s-m-n theorem in its simple form,

$$\phi_{s(e,x)}(y) = \phi_e(x, y)$$

In order to demonstrate this theorem, let $j, k \in \mathbb{N}$, such as $j, k \geq 1$, f and g primitive recursive functions defined as:

$$\begin{array}{ll} (i) & f(k) = \bar{\beta}(0, k) \\ (ii) & g(\bar{\beta}(j, k)) = \bar{\beta}(j+1, k) \end{array}$$

As $\mathcal{F}$ is an AFA, there are indexes p and q for f and g, where $f = \phi_p^{\mathcal{F}}$ and $g = \phi_q^{\mathcal{F}}$. Let H be defined by:

$$\begin{array}{lcl} H(0) & = & p \\ H(x+1) & = & q \circ H(x) \end{array}$$

and so H is primitive recursive. First, it is necessary to demonstrate by induction that $\phi_{H(j)}(k) = \bar{\beta}(j, k)$.

- For $j = 0$

$$\phi_{H(0)}(k) = \phi_p(k) = f(k) = \bar{\beta}(0, k)$$

- Now, suppose that the proposition is valid for j, that is:

$$\phi_{H(j)}(k) = \bar{\beta}(j, k)$$

So, we have:

$$\begin{array}{lcl} \phi_{H(j+1)}(k) & = & \phi_{q \circ H(j)}(k) \\ & = & \phi_q \circ \phi_{H(j)}(k) \\ & = & g(\phi_{H(j)}(k)) \\ & = & g(\bar{\beta}(j, k)) \\ & = & \bar{\beta}(j+1, k) \end{array}$$

Let k be the index of the function $\phi_k(\bar{\beta}(\bar{\beta}(\underline{x}_m), \bar{\beta}(\underline{y}_n))) = \bar{\beta}_{m+n}(\underline{x}_m, \underline{y}_n)$. We define $s(e, \bar{\beta}(\underline{x}_m)) = e \circ k \circ H(\bar{\beta}(\underline{x}_m))$, so:

$$\begin{array}{l} \phi_{s_(e,\bar{\beta}(\underline{x}_m))}(\bar{\beta}(\underline{y}_n)) = \\ = \phi_e \circ \phi_k \circ \phi_{\underline{H(\bar{\beta}(\underline{x}_m))}}(\bar{\beta}(\underline{y}_n)) \\ = \phi_e \circ \phi_k(\bar{\beta}(\bar{\beta}(\underline{x}_m), \bar{\beta}(\underline{y}_n))) \\ = \phi_e(\bar{\beta}_{m+n}(\underline{x}_m, \underline{y}_n)) \\ = \phi_e(\underline{x}_m, \underline{y}_n) \end{array}$$

This theorem claims that for a given program with $m+n$ input arguments, if m arguments are set to be fixed, than it is possible to get a specialized new program with n input arguments. For example, suppose an application designed to attack cryptographic systems configured by $m+n$ variables. In this application, m arguments describe the cryptographic protocol features such as key Exchange, authentication, signature, etc; and n arguments describe cryptographic techniques such as key length, key manager, algorithm type, etc. The first parameters specify the cryptographic agreement between the actors of the process, and the parameters specify the group of methods used to construct the cryptographic object. If the application designed to attack cryptographic systems is sold for an entity concerned to messages that use public-key authentication cryptography, than the s-m-n Theorem ensures that there is an specific instance of this application that simulates the behavior of an attack against the public-key authentication cryptography by simply instancing the correspondent m parameters.

Another important theorem from the AFA is called Translation Theorem. It provides a relationship between two different AFA.

***Theorem** 2 (Weak Translation Theorem):* Given two abstract algorithm families $\mathcal{F}$ e $\mathcal{G}$ there is a primitive recursive function $\underline{tr}_{\mathcal{F}}^{\mathcal{G}}$, $\mathcal{G} \rightarrow \mathcal{F}$ such that for all $m \in \mathbb{N}$:

$$\phi_m^{\mathcal{G}} = \phi_{\underline{tr}_{\mathcal{F}}^{\mathcal{G}}(m)}^{\mathcal{F}}$$

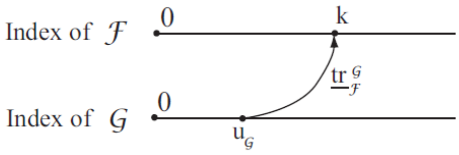


Figure 4. Translation from $\mathcal{G}$ to $\mathcal{F}$ illustrating the Weak Translation Theorem.

The Fig. 4 illustrates that there is a transformation of an indexed algorithm $\mathcal{G}$ whose image is an algorithm indexed on $\mathcal{F}$. In order to demonstrate the existence of this function $\underline{tr}_{\mathcal{F}}^{\mathcal{G}}$, keep in mind that as any AFA, $\mathcal{G}$ has an universal algorithm indexed by $u_{\mathcal{G}}$ which allows to say that $\phi_{u_{\mathcal{G}}}^{\mathcal{G}}(m, x) = \phi_m^{\mathcal{G}}(x)$. Moreover, it is important to remember that the AFA $\mathcal{F}$ computes all computable recursive functions, including the AFA $\mathcal{G}$ universal function. Thus, if k is the index of the function $\phi_{u_{\mathcal{G}}}^{\mathcal{G}}$ in AFA $\mathcal{F}$, it is defined $\underline{tr}_{\mathcal{F}}^{\mathcal{G}}$ by:

$$\underline{tr}_{\mathcal{F}}^{\mathcal{G}}(y) = s(k, y)$$

where y is the index of the algorithm in $\mathcal{G}$ on which to find its equivalent. So:

$$\phi_{\underline{tr}_{\mathcal{F}}^{\mathcal{G}}(m)}^{\mathcal{F}}(x) = \phi_{s(k,m)}^{\mathcal{F}}(x)$$

By using the s-m-n Theorem on $\phi_{s(k,m)}^{\mathcal{F}}(x)$ we have:

$$\begin{array}{lcl} \phi_{\underline{tr}_{\mathcal{F}}^{\mathcal{G}}(m)}^{\mathcal{F}}(x) & = & \phi_{s(k,m)}^{\mathcal{F}}(x) \\ & = & \phi_k^{\mathcal{F}}(m, x) \\ & = & \phi_{u_{\mathcal{G}}}^{\mathcal{G}}(m, x) \\ & = & \phi_m^{\mathcal{G}}(x) \end{array}$$

The Weak Translation Theorem permits to associate one AFA to another. It seems to be natural that this theorem will be used at the next sections to qualify the relationship between algorithms from two different abstract algorithm families. Furthermore, it will be important to support some considerations about Church-Turing Thesis.

One of the most important results for an Abstract Algorithm Family is called the Recursion Theorem [1][6]. This theorem will be presented in the next section.

VII. Recursion Theorem

Let $\mathcal{F} =< A, \varepsilon_A, |\,| >$ be an AFA and $f : \mathbb{N} \rightarrow \mathbb{N}$ a total recursive function which transforms a given algorithm. The f specification states that when it is applied to an algorithm index it results into a new index, and there is no need to consider what is the new indexed algorithm. Suppose that the function f applied to an index i maps a new index $f(i)$ associated to an algorithm $\varepsilon_A(f(i))$. A specific relevant scenario occurs when an ordinary algorithm $\varepsilon_A(j)$ computes exactly the same partial recursive function ϕ that the algorithm $\varepsilon_A(f(i))$ does, that is, $|\varepsilon_A(f(i))| = |\varepsilon_A(j)|$. In fact, it is expected that there will be algorithm capable of executing the same task, and also, there might be an unlimited relationships of this kind.

Hence, the set of all algorithms that compute the same partial recursive function ϕ is given by:

$$\mathcal{T}_{\mathcal{F}} = \left\{ < i, j > \in \mathbb{N}^2 | \phi_{f(i)} = \phi_j \right\}$$

***Theorem** 3 (Fixed Point Theorem):* The particular case where the algorithm $\varepsilon_A(f(i))$ computation produces the same function produced by $\varepsilon_A(i)$ is a circumstance when it is said that $i \in \mathbb{N}$ is a fixed point of ϕ [1][4], that is,

$$\phi_{f(i)} = \phi_i$$

A function fixed point is a value which remains unchanged, even though the function is applied to it. Note that the self-reference usage tends to be undesirable by the mathematical logic, such as in the lair paradox. However, the self-reference is important since the primitive recursion is based on it, that is, it is based on function definition in terms of the same functions.

The Recursion Theorem, as will be demonstrated in the following, is related to mathematical logic and self-reproducing systems, that is, functions that produce functions, and finally, the theorem is correlated to the possibility of creating machines that constructs replicas of them. Usually, non-researches of this subject have a huge resistance to accept this possibility, mainly because they accept as a dogma that machines cannot self-reproduce. This is clearly a mistake and the reason will be explained.

The mistaken reasoning works as follows: first, consider the matter of a machine be capable to produce another machine, such as a microprocessors factory. Electronic supplies are provided to this factory, it employs a robotic manufactory according a pre-defined instruction set, and at the end of the process it deploys a microprocessor. This factory must be more complex than the produced microprocessors since this factory needs to internalize not only the microprocessor design, but also the governance project of all its robots. The same reasoning might be used for a more abstract situation, where a machine A constructs a machine B. Therefore, the machine A must be more complex than the machine B. Additionally, it is correct to assert that any machine cannot be more complex than itself. Consequently, no machine can construct itself, turning the self-reproduction an impossible operation. Nevertheless, the Recursion Theorem refutes categorically this conclusion. The main explanation is because it is wrong the argument that there is a relationship between the ability of executing several functionalities and the possibility of replicating these functionalities.

***Theorem** 4 (Recursion Theorem):* Given an AFA and a recursive function f, there is a natural number n where

$$\phi_{f(n)} = \phi_n$$

This theorem means that there is a natural number n associated to a partial recursive function ϕ which submitted to a function f produces as a result the same function ϕ. Therefore, this procedure replicates the function ϕ.

In order to demonstrate this property, let n be a natural number and A_n an algorithm dependant on n with the following specification:

1) Takes n as input and applies ε_A to it;
2) Selects the algorithm $\varepsilon_A(n)$ from the AFA $\mathcal{F}$ to be used;
3) Computes $\varepsilon_A(n)$ using the same entrance n, and so obtainning $|\varepsilon_A(n)|\,(n)$;

$$A_n = \begin{cases} |\varepsilon_A(|\varepsilon_A(n)|\,(n))| & se\ |\varepsilon_A(n)|\,(n)\ converge \\ \\ diverge & se\ |\varepsilon_A(n)|\,(n)\ diverge \end{cases}$$

The classical Mathematics that ordinary function, when submitted to an input, will always produce an output, no matter what is this result. On the other hand, the Computability does not share this security degrees with classical Mathematics. An algorithm may receive a certain input and it may never stop processing it, an issue known as the halting problem. Under this point of view, the A_n specification needs to consider the divergent situation into its definition.

The Fig. 5 shows the strategy A_n, as previously defined, that will be used in order to demonstrate the theorem. The A_n goal is to show that n is a fixed point of ϕ_n. To accomplish this, it is necessary to choose the ϕ associated to n, compute the result of $\phi_n(n)$, and then, it is essential to demonstrate that this result remains unchanged when it is submitted to ϕ_n again.

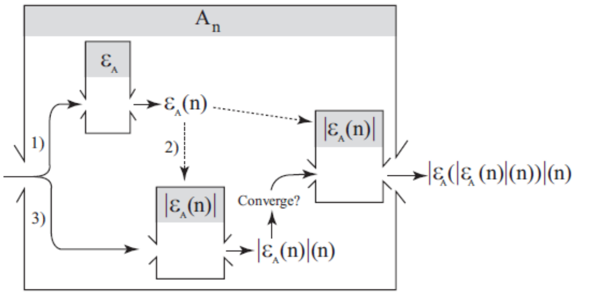


Figure 5. Recursion Theorem demonstrating strategy using the proposed algorithm A_n.

The property P_2 of the AFA $\mathcal{F}$ allows to say that:

$$|\varepsilon_A(n)|\,(n) = \phi_{u_{\mathcal{F}}}(n, n)$$

which indicates that the meta-algorithm will provide the execution of the algorithm indexed by n and whose input will be the string n. So, let n be a natural number submitted to the execution stream of Fig. 5. Therefore:

$$\begin{aligned} |A_n|\,(y) &= |\varepsilon_A(|\varepsilon_A(n)|\,(n))|\,(y) \\ &= \phi_{u_{\mathcal{F}}}(|\varepsilon_A(n)|\,(n), y) \\ &= \phi_{u_{\mathcal{F}}}(\phi_u(n,n), y) \end{aligned}$$

Let H be a function defined by:

$$H(n,y) = |A_n|\,(y) = \phi_{u_{\mathcal{F}}}(\phi_{u_{\mathcal{F}}}(n,n), y)$$

and let m_H the index of H. Define g by:

$$g(n) = s(m_H, n)$$

Then, performing the g substitution and by using s-m-n Theorem in its simple form we have:

$$\phi_{g(n)}(y) = \phi_{s(m_H,n)}(y) = \phi_{m_H}(n,y) = H(n,y)$$

Let $n_f = g(c_{\mathcal{F}}(m_f, m_g))$ an ordinary natural number which specifies a certain algorithm $\varepsilon_A(n_f)$, and c a composition function that merges the machine index m_f, proposed on the theorem, and the machine m_g. Computing this algorithm $\varepsilon_A(n_f)$ for the entry y we obtain:

$$\begin{aligned} \phi_{n_f}(y) &= \phi_{g(c_{\mathcal{F}}(m_f,m_g))}(y) \\ &= H(c_{\mathcal{F}}(m_f,m_g), y) \\ &= \phi_{u_{\mathcal{F}}}(\phi_{u_{\mathcal{F}}}(c_{\mathcal{F}}(m_f,m_g), c_{\mathcal{F}}(m_f,m_g)), y) \\ &= \phi_{u_{\mathcal{F}}}(\phi_{c_{\mathcal{F}}(m_f,m_g)}(c_{\mathcal{F}}(m_f,m_g)), y) \\ &= \phi_{c_{\mathcal{F}}(m_f,m_g)(c_{\mathcal{F}}(m_f,m_g))}(y) \\ &= \phi_{f \circ g(c_{\mathcal{F}}(m_f,m_g))}(y) \\ &= \phi_{f(g(c_{\mathcal{F}}(m_f,m_g)))}(y) \\ &= \phi_{f(n_f)}(y) \end{aligned}$$

So $n_f = g(c_{\mathcal{F}}(m_f,m_g)) = s_F(m_H, c_{\mathcal{F}}(m_f,m_g))$ is the fixed point of ϕ.

The Recursion Theorem can be used to construct some interesting recursive functions, such as the Self-replication Theorem [15].

***Theorem** 5 (Self-replication Theorem):* There is an algorithm that prints its own description, given any input.

Define f by $\phi_{f(x)}(y) = x$, that is, $f(x) = s(e,x)$ where e is an index for the projection function so that $\phi_{f(x)}(y) = \phi_{s(e,x)}(y) = \phi_e(x,y) = \underline{pr}_1^2(x,y) = x$. Since $f(x)$ is recursive, by the Recursion Theorem there is a n such that $\phi_{f(n)} = \phi_n$, hence $\phi_n(y) = \phi_{f(n)}(y) = n$.

The function ϕ_n is a function whose constant value is its index n. The algorithm which computes ϕ_n always outputs its own description. The word n is called a description of the algorithm because the algorithm with index n is $\varepsilon_A(n)$.

As a result, there is a natural number n associated to an AFA $\mathcal{F}$ that once submitted to function f it produces, as a result, the function ϕ. A computer virus and a computer worm [8] are designed to replicate themselves among several computers. These viruses are inactive when analyzed exclusively as block of programming code. However, when deployed into a host, it may become active and may start to transmit copies of itself to other accessible computers. Consequently, with the purpose of accomplishing its replicating task, the viruses contain the type of scheme described at the Recursion Theorem demonstration [10]. The quines are another variety of this kind of application [9]. They are programs with no entry that produce copies of themselves.

VIII. Isomorphism Theorem

In this section we will present some consequences of the: Fixed Point Theorem, Translation Theorem and Recursion Theorem. These issues will provide support to the main topic called Isomorphism Theorem.

Let $x_1 \neq x_2$ such that $\phi^{\mathcal{G}}_{x_1}(y) \neq \phi^{\mathcal{G}}_{x_2}(y)$. With the assistance of the universal algorithm from the AFA $\mathcal{G}$, it is possible to say that $\phi^{\mathcal{G}}_{u_{\mathcal{G}}}(x_1,y) \neq \phi^{\mathcal{G}}_{u_{\mathcal{G}}}(x_2,y)$. Though, the universal algorithm from $\mathcal{G}$ has an index k on AFA $\mathcal{F}$. So:

$$\begin{aligned} \phi^{\mathcal{G}}_{u_{\mathcal{G}}}(x_1,y) &\neq \phi^{\mathcal{G}}_{u_{\mathcal{G}}}(x_1,y) \\ \phi^{\mathcal{F}}_{k}(x_1,y) &\neq \phi^{\mathcal{F}}_{k}(x_1,y) \\ \phi^{\mathcal{F}}_{s(k,x_1)}(y) &\neq \phi^{\mathcal{F}}_{s(k,x_2)}(y) \\ \phi^{\mathcal{F}}_{\underline{tr}^{\mathcal{G}}_{\mathcal{F}}(x_1)}(y) &\neq \phi^{\mathcal{F}}_{\underline{tr}^{\mathcal{G}}_{\mathcal{F}}(x_2)}(y) \end{aligned}$$

If, for the same entry y two algorithm produce distinct computations, then it is because these algorithm are also distinct, and consequently are their indexes. Therefore, $\underline{tr}^{\mathcal{G}}_{\mathcal{F}}(x_1) \neq \underline{tr}^{\mathcal{G}}_{\mathcal{F}}(x_2)$. If this happens and if $x_1 \neq x_2$ then the function $\underline{tr}^{\mathcal{G}}_{\mathcal{F}}$ must be injective. For this reason, an injective $\underline{tri}^{\mathcal{G}}_{\mathcal{F}} = \underline{tr}^{\mathcal{G}}_{\mathcal{F}}$.

***Theorem** 6 (Strong Translation Theorem):* Given two abstract algorithm families $\mathcal{F}$ e $\mathcal{G}$ there is a primitive recursive injective function $\underline{tri}^{\mathcal{G}}_{\mathcal{F}}$, $\mathcal{G} \to \mathcal{F}$ such that for all $m \in \mathbb{N}$:

$$\phi^{\mathcal{G}}_{m} = \phi^{\mathcal{F}}_{\underline{tri}^{\mathcal{G}}_{\mathcal{F}}(m)}$$

Moreover, pay attention to some features from a peculiar function p called padding function. Its role is to add instructions to the algorithm without changing its functionality. At first, it may seems to be useless under the traditional computing point of view, but this behavior occur in several everyday forms. For instance, when a programmer add coments to his programming code, the size of the text increases, and the desired execution behavior is kept unchanged. So, the act of documenting the source-code of a program is a padding expression.

The current programming languages clearly prioritize the usage of code interpretation (Java, PHP, IL from .Net, script languages, ...), against code compilation. The main reason is the benefits provided by the code portability. Though, the side effect is that the interpreted code may easily be decompilated. Notwithstanding, software developers are still building interpreted application, distributing them through download or physical medias. Under these circumstances, it is critical to protect application source-code and intellectual capital, no matter if this application is freeware, shareware, or commercially licensed. The traditional cryptography is not a solution for this problem, because it would prevent the computer processor to have access to the program instructions. Therefore, the cryptographic solution is the code obfuscation, that is, to make the source-code less understandable for a human by adding irrelevant and outwit instructions, but keeping the functionality, exactly as the code padding.

The understanding of how the padding operates is better comprehended by using Labeled Markov Algorithm language [4],which will be summarized as follows.

***Definition** 3 (Labeled Markov Algorithm):* A LMA with an input Σ, or shortly a LMA in Σ, is a sequence of $n > 0$ expressions such as $l:\ x \to\ y/l'$, called commands, where:

- l and l' are natural numbers, or their representation, known as command labels and transferring labels, respectively.
- x is a word from the alphabet, called command' left.
- y is a word from the alphabet, called command' right.

Additionally, for all $i,\ 1 \leq i \leq n$, the i command has the $i-1$ label.

As it is common on programming languages, the sequence commands are separated by the symbol ;. Despite the data do not belong to the LMA definition (there are no variables or registers), the central point is to submit a Σ^* string to the LMA. The input string is transformed by the execution of the LMA, and its output is a modified string. Let a LMA be $\varepsilon_A(e)$ and $l : x_l \to y_l/l'$ a command from $\varepsilon_A(e)$. It is defined the function $\phi_e = |l : x_l \to y_l/l'|$ with values given by:

$$|l : x \to y/l'|(w) = \begin{cases} \underline{subst}(w,x,y) & if\ x \preceq w \\ \\ w & if\ x \not\preceq w \end{cases}$$

where $\underline{subst}(w,x,y)$ is the substitution result of the first occurrence of x in w for y, and $x \preceq w$ means that x is a substring from w.

The following sequence of commands is a LMA form from an alphabet Σ:

$$\begin{array}{l} 0 : 0 \to a/1; \\ 1 : a1 \to 1a/1; \\ 2 : a2 \to 2a/1; \\ 3 : a \to 2/5; \end{array}$$

Let $w \in \Sigma^*$. The command label 0 means: substitute the first occurrence of the null word in w for a, that is, put the marker a at the beginning of the word w and go to the execution of command label 1. The command label 1 means: substitute the first occurrence of the word $a1$ in w for $1a$ and execute again the command label 1.

Intuitively, it is known that an algorithm may be implemented into several different ways, even though theses different implementations compute the same function. As the index i from a function φ_i is given by the LMA enumeration $\varepsilon_A(i)$ that compute this function, in order to ensure the mapping to infinite indexes, it sufficient to take $\varepsilon_A(i)$ and add innocuous commands to produce a new $\varepsilon_A(i')$.

Therefore, let $p(e,x)$ be a padding function, recusive primitive and injective, which concatenates the algorithm $\varepsilon_A(e)$ with an irrelevant algorithm $\varepsilon_A(x)$. The concatenation operation may occur at the beginning ou at the end of the algorithm, but it may also more sophisticated and intercalate patches of algorithm $\varepsilon_A(e)$ with patches from $\varepsilon_A(x)$. Whatever the case, we have $\phi_{p(e,x)} = \phi_e$ and if $p(e,x) = p(e,y)$ then it is because $x = y$.

The main idea is to concatenate the useless commands from algorithm $\varepsilon_A(x)$ with the LMA algorithm index$e$. So, if $\varepsilon_A(e)$ is the LMA:

$$\varepsilon_A(e) = \begin{cases} 0 : x_0 \to y_0/l_0; \\ 1 : x_1 \to y_1/l_1; \\ \dots \\ k : x_k \to y_k/l_k; \end{cases}$$

Let x be the index of the program:

$$\varepsilon_A(x) = \begin{cases} 0 : 0 \to 0/1; \\ 1 : 0 \to 0/2; \\ \dots \\ y : 0 \to 0/y+1; \end{cases}$$

The LMA $\varepsilon_A(p(e,x))$ is the LMA:

$$\varepsilon_A(p(e,x)) \begin{cases} 0 : 0 \to 0/1; \\ 1 : 0 \to 0/2; \\ \dots \\ y : 0 \to 0/y+1; \\ y+1 : x_0 \to y_0/l_0; \\ y+2 : x_1 \to y_1/l_1; \\ \dots \\ y+k+1 : x_k \to y_k/l_k; \end{cases}$$

Then, the program created by the concatenation is the composition of the programs indexes x e e, and so it is sufficient to take $p(e,x) = c_{\mathcal{F}}(e,x)$, because then:

$$\phi_{p(e,x)} = \phi_{c_{\mathcal{F}}(e,x)} = \phi_e$$

Remembering that by definition the function p is injective.

Let $\mathcal{F}$ and $\mathcal{G}$ be two abstract algorithm families. Then, there is a recursive function $\underline{trias}_{\mathcal{G}}^{\mathcal{F}}$ such that for all $x \in \mathbb{N}$:

$$\phi^{\mathcal{G}}_{\underline{tris}_{\mathcal{F}}^{\mathcal{G}}(x)} = \phi^{\mathcal{F}}_x$$

and additionally:

$$0 < \underline{tris}_{\mathcal{F}}^{\mathcal{G}}(x) < \underline{tris}_{\mathcal{F}}^{\mathcal{G}}(x+1)$$

In order to make this happens, it is necessary to translate each program x and then add superfluous commands until the desired length is obtained. Note that this feature from function $\underline{tris}_{\mathcal{F}}^{\mathcal{G}}$ allows to state that for all algorithm from A, there is a natural number $x \in \mathbb{N}$ such that this algorithm is determined by $\varepsilon_A(x)$, characterizing a surjection.

Finally, it is possible to present the Isomorphism Theorem [1][4], which ensures a bijective mapping between abstarct algorithm families, that is, all abstract algorithm families are equivalent. The Isomorphism Theorem shows not only two acceptable algorithms might be ttranslated from one to another, but also that there is a one-to-one translating between these two acceptable algorithms.

***Theorem** 7 (Isomorphism Theorem):* Let $\mathcal{F}$ and $\mathcal{G}$ be two abstract algorithm families. If the translation fuction is injective and surjective then there is a bijective recursive function $\underline{trb}_{\mathcal{F}}^{\mathcal{G}}$ such that for all $x \in \mathbb{N}$:

$$\phi^{\mathcal{G}}_{\underline{trb}_{\mathcal{F}}^{\mathcal{G}}(x)} = \phi^{\mathcal{F}}_x$$

The demonstration os this theorem involves the usage of the function $\underline{tris}$, and it is presented a more complete adaptation from the one described by Machtey e Young [16]. As it was studied, this function is injective, that is, distinct indexes i and j from $\mathcal{G}$, are mapped through translation into distinct indexes from $\mathcal{F}$. This function is also surjective because all indexes from $\mathcal{F}$ are translation images from a index in $\mathcal{G}$.

Note that $\underline{tris}_{\mathcal{F}}^{\mathcal{G}}$, by construction, is monotonically increasing. Thus, given a natural number x, the composition result

$\underline{tris}^{\mathcal{F}}_{\mathcal{G}} \circ \underline{tris}^{\mathcal{G}}_{\mathcal{F}}$ produces a numerical value greater than x, that is, $\underline{tris}^{\mathcal{F}}_{\mathcal{G}}(\underline{tris}^{\mathcal{G}}_{\mathcal{F}}(x)) > x$. Once the function $\underline{tris}$ is based on padding we have that $x < \underline{tris}^{\mathcal{G}}_{\mathcal{F}}(x) < \underline{tris}^{\mathcal{F}}_{\mathcal{G}}(\underline{tris}^{\mathcal{G}}_{\mathcal{F}}(x))$. Consequently, the inverse translation functions must be monotonically decreasing. Therefore, we define the inverse functions s^* e t^* by (see Fig. 6):

$$s^*(x) = \begin{cases} y & \text{if exists } y \text{ such that } \underline{tris}^{\mathcal{G}}_{\mathcal{F}}(y) = x \\ 0 & \text{otherwise} \end{cases}$$

$$t^*(x) = \begin{cases} y & \text{if exists } y \text{ such that } \underline{tris}^{\mathcal{F}}_{\mathcal{G}}(y) = x \\ 0 & \text{otherwise} \end{cases}$$

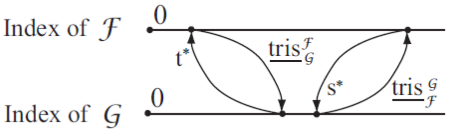

Index of $\mathcal{F}$
0
t^*
$\underline{tris}^{\mathcal{F}}_{\mathcal{G}}$
s^*
$\underline{tris}^{\mathcal{G}}_{\mathcal{F}}$
Index of $\mathcal{G}$
0

Figure 6. The inverse functions s^* e t^* associated to the translations between $\mathcal{F}$ and $\mathcal{G}$.

Given an index x from AFA $\mathcal{F}$, it is possible to define a sequence:

$$I_x = \{x, s^*(x), t^* \circ s^*(x), s^* \circ t^* \circ s^*(x) \cdots\}$$

As s^* and t^* are strictly decreasing (see Fig. 7), this sequence has, at most, $x+1$ elements, and also, the last value is always 0, that is, for some $l \in \mathbb{N}$:

1) the sequence ends on $\mathcal{F}$, and then $(t^* \circ s^*)^l(x) = 0$, or
2) the sequence ends on $\mathcal{G}$, and then $s^* \circ (t^* \circ s^*)^l(x) = 0$.

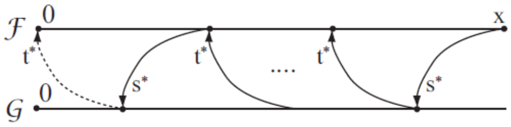


Figure 7. The s^*'s and t^*'s strictly decreasing sequences converging to 0 as the last value.

The dot line from Fig. 7 indicates that the sequence ends in $\mathcal{F}$ according to one of the options:

- $\underline{tris}^{\mathcal{F}}_{\mathcal{G}}(0) = (t^* \circ s^*)^l(x)$, or;
- $\nexists y$ in $\mathcal{F}$ such that $\underline{tris}^{\mathcal{F}}_{\mathcal{G}}(y) = (t^* \circ s^*)^l(x)$.

Analogously, note that a sequence ends on $\mathcal{G}$ according two situations:

- $\underline{tris}^{\mathcal{G}}_{\mathcal{F}}(0) = s^* \circ (t^* \circ s^*)^{l-1}(x)$, or;
- $\nexists y$ in $\mathcal{G}$ such that $\underline{tris}^{\mathcal{F}}_{\mathcal{G}}(y) = s^* \circ (t^* \circ s^*)^{l-1}(x)$.

Lets define the function:

$$\underline{term\mathcal{G}}(x) = \begin{cases} 1 & if\ I_x\ ends\ on\ \mathcal{G} \\ 0 & if\ I_x\ ends\ on\ \mathcal{F} \end{cases}$$

And the function:

$$\underline{trb}^{\mathcal{G}}_{\mathcal{F}}(x) = \begin{cases} \underline{tris}^{\mathcal{G}}_{\mathcal{F}}(x) & if\ \underline{term\mathcal{G}}(x) = 1 \\ t^*(x) & if\ \underline{term\mathcal{G}}(x) = 0 \end{cases}$$

Hence, it is expected to prove that $\underline{trb}^{\mathcal{G}}_{\mathcal{F}}$ is the desired translation function. Let x be an index from $\mathcal{G}$:

1) if $\underline{term\mathcal{G}}(x)=1$ then $\phi^{\mathcal{G}}_{\underline{trb}^{\mathcal{G}}_{\mathcal{F}}(x)} = \phi^{\mathcal{G}}_{\underline{tris}^{\mathcal{G}}_{\mathcal{F}}(x)} = \phi^{\mathcal{F}}_x$
2) if $\underline{term\mathcal{G}}(x)=0$ then $\phi^{\mathcal{G}}_{\underline{trb}^{\mathcal{G}}_{\mathcal{F}}(x)} = \phi^{\mathcal{G}}_{t^*(x)} = \phi^{\mathcal{F}}_{\underline{tris}^{\mathcal{F}}_{\mathcal{G}}(t^*(x))} = \phi^{\mathcal{F}}_x$

In both cases $\phi^{\mathcal{G}}_{\underline{trb}^{\mathcal{G}}_{\mathcal{F}}(x)} = \phi^{\mathcal{F}}_x$. Finally, it is necessary to prove that $\underline{trb}^{\mathcal{G}}_{\mathcal{F}}$ is bijective.

1) $\underline{trb}^{\mathcal{G}}_{\mathcal{F}}$ is injective: let $x, y \in \mathbb{N}$ such that $\underline{trb}^{\mathcal{G}}_{\mathcal{F}}(x) = \underline{trb}^{\mathcal{G}}_{\mathcal{F}}(y)$. Then it is mandatory that $\underline{term\mathcal{G}}(x) = \underline{term\mathcal{G}}(y)$:
   a) If $\underline{term\mathcal{G}}(x) = 1$ then $\underline{tris}^{\mathcal{G}}_{\mathcal{F}}(x) = \underline{tris}^{\mathcal{G}}_{\mathcal{F}}(y)$ thus, $x = y$;
   b) If $\underline{term\mathcal{G}}(x) = 0$ then $s^*(x) = s^*(y)$ thus, $x = y$;
2) $\underline{trb}^{\mathcal{G}}_{\mathcal{F}}$ is surjective: let y be an index from $\mathcal{F}$, then there is an index x from $\mathcal{G}$ such that $\underline{trb}^{\mathcal{G}}_{\mathcal{F}}(x) = y$, because analysing $\underline{tris}^{\mathcal{G}}_{\mathcal{F}}(y)$, it is verified that:
   a) if $\underline{term\mathcal{G}}(\underline{tris}^{\mathcal{G}}_{\mathcal{F}}(y)) = 1$ then $x = s^*(y)$ once that, in this case, $y = \underline{tris}^{\mathcal{F}}_{\mathcal{G}}(x) = \underline{trb}^{\mathcal{F}}_{\mathcal{G}}(x)$;
   b) if $\underline{term\mathcal{G}}(\underline{tris}^{\mathcal{G}}_{\mathcal{F}}(y)) = 0$ then $x = \underline{tris}^{\mathcal{G}}_{\mathcal{F}}(y)$ once that, in this case, $y = t^*(\underline{tris}^{\mathcal{G}}_{\mathcal{F}}(y)) = \underline{trb}^{\mathcal{G}}_{\mathcal{F}}(x)$;

The importance of this theorem is a reinforcement to Church-Turing Thesis, whatever computer model with the three basic properties (P_1, P_2, P_3), even possessing supposedly more powerful properties, is recursively isomorphic to the already known models. This result is important because it provides a convincing argument to whatever particular computing model is choosen. The Isomorphism Theorem states that all abstract algorithm families as equivalent, and that there do exists an effective and bijective translation function between them.

Note that the Cryptography studies the artifices responsible for transforming the information from its plaintext form into a ciphered form, comprehensible by a select group of people. The evaluation of the ciphered message provides the informative content. The results of the cryptographic techniques implementation comprise some kind of computation, such as the operator $|\ |$.

On the other hand, the Linguistics is the study of the language, that is, the schemes used by mankind to comprehend its ideas and feelings (the authors are consciously avoiding the word "communicate" because the Pêcheux [17] concept of imaginary formation). This comprehension is a consequence of carrying out several human abilities that are associated to an operator of semantic meaning, again, just like the operator $|\ |$.

Therefore, it is possible to incorporate these two points of view because both of them study principles and techniques which define a set of functions whose ultimate objective is to perform an evaluation, such as the algorithms.

By an AFA we mean a denumerable set of undefined objects called algorithms, each of which has associated with exactly one partial n-variable function for each positive integer n. Consequently, we claim that that Linguistics and Cryptography are associated to two different abstract families of algorithms. Moreover, the Isomorphism Theorem provides support to assert that the Linguistics and Cryptography set of objects are equivalent. Thus, a computable function, whose results are acceptable by one study approach, might be successfully used on the other knowledge area. Undoubtedly, an algorithm $\varepsilon_A^{\mathcal{G}}(n)$ from one knowledge area does not need to be the same one from the other area. However, there do exists an analogous algorithm on the other family, $\varepsilon_A^{\mathcal{F}}(n)$, and the mapping between them is performed by the translation function $\underline{trb}_{\mathcal{F}}^{\mathcal{G}}$, even though this translation might be unknown.

Hence, it seems to be reasonable to use computational linguistics and modern information retrieval to perform cryptographic analysis, not only to recover the informative content of a ciphered message, but also to locate features about the cryptographic key and the type of algorithm used for encryption. Alternatively, the Linguistic comparative method and the pragmatic factors study can be enhanced by Cryptography quantitative and qualitative analysis algorithms.

IX. Conclusions

This article has presented several remarks about how recursive functions computability study can be directly associated to Cryptography themes. In order to accomplish this task, the abstract family of algorithms theory was revised producing a different approach from the ones currently available in literature. The Isomorphism Theorem demonstration provides the technical support for a comparison between two AFA, establishing a connection between Cryptology and Linguistics.

Further studies on this matter include, but not restricted to, a research on cryptogram patterns associated to the keys used on AES and RSA. It also seems to be possible to use corpora techniques to group cryptographic texts according to their cipher keys.

References

[1] Hartley Rogers Jr. *Theory of Recursive Functions and Effective Computability*, 1st ed, McGraw-Hill Book Company, 1967.

[2] Roberto Lins de Carvalho. *Máquinas, Programas e Algoritmos*, 2^a Escola de Computação, Campinas, Universidade Estadual de Campinas, 1981.

[3] Yu I. Manin. *A Course in Mathematical Logic*, 1st ed, Graduate Texts in Mathematics 53, Springer-Verlag, 1977.

[4] Claudia Maria Garcia Medeiros de Oliveira and Roberto Lins de Carvalho. *Modelos de Computação e Sistemas Formais*, 11^a Escola de Computação, Rio de Janeiro, Universidade Federal do Rio de Janeiro, 1998.

[5] Elliott Mendelson. *Introduction to Mathematical Logic*, 3rd ed, Cole Mathematics Series, The Wadsworth and Brooks, 1987.

[6] Michael Sipser. *Introduction to The Theory of Computation*, 2rd ed, Course Technology Series, Thomson, 2006.

[7] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory*, Language and Computation. USA, Addison-Wesley Publishing Company, 1979

[8] Adam Young and Moti Yung. *Malicious Cryptography: Exposing Cryptovirology*, John Wiley and Sons Inc., 2004.

[9] Nigel J. Cutland. *Computability: An introduction to recursive function theory*, Cambridge University Press, 1980.

[10] Guillaume Bonfante and Matthieu Kaczmarek and Jean-Yves Marion. *A Classification of Viruses through Recursion Theorems*, CiE 2007, volume 4497 of Lecture Notes in Computer Science, 2007.

[11] Steven Homer and Alan L. Selman. *Computability and Complexity Theory*, Texts in Computer Science, 2nd ed, Springer, 2011.

[12] Alan M. Turing. *The Undecidable*, chapter On Computable Numbers with an Application to the Entscheidungsproblem, pages 115-151. Raven Press, New York, 1965.

[13] R. J. Nelson. *Introduction to Automata*, John Wiley and Sons, Inc., USA, 1965.

[14] Frederick C. Hennie. *Introduction to Computability*, Series in Computer Science and Information Processing, Addison-Wesley, Reading, Massachusets, USA, 1977.

[15] Walter S. Brainerd and Lawrence H. Landweber. *Theory of Computation*, John Wiley and Sons, 1974.

[16] Michael Machtey and Paul Young. *An Introduction to the General Theory of Algorithms*, Theory of Computation Series, Elsevier North Holland Inc., 1978.

[17] Michel Pêcheux. *Análise Automática do Discurso*, In: Por uma Análise Automática do Discurso, Editors: Françoise Gadet and Tony Hak, Editora Unicamp, 3.ed., 1997.

Roberto Lins de Carvalho did his PhD. on informatics at University of Toronto (1974), MSc. on informatics at Pontifical Catholic University of Rio de Janeiro - PUC Rio (1969), under graduation on telecommunication engineering at Military Engineering Institute - IME (1967) and under graduation military studies at Military Academy of Agulhas Negras - AMAN (1961). Lectured at Pontifical Catholic University of Rio de Janeiro, Campinas University, Fluminense Federal University, Aeronautics Technological Institute and Military Engineering Institute, where supervised several master dissertations and doctoral thesis on theorem proving, knowledge base systems and knowledge representation. Over fourteen years is the leader of Witty Group coaching artificial intelligence computer applications and is retired from Scientific Computing Brazilian National Laboratory - LNCC.

Flávio Luis de Mello did his DSc. on theory of computation and image processing at the Federal University of Rio de Janeiro - UFRJ (2006), MSc. on computer graphics at Federal University of Rio de Janeiro - UFRJ (2003), under graduation on systems engineering at Military Engineering Institute - IME (1998). Developed command and control systems and implemented military messages interchange applications during twelve years as Brazilian Army officer. Responsible for developing software applications based on theorem proving, knowledge base systems and knowledge representation from Witty Group. Associate Professor at the Electronics and Computing Department (DEL) of Polytechnic School (Poli) at Federal University of Rio de Janeiro (UFRJ) since 2007.

Harnessing Nature's Randomness: Physical Random Number Generator

G. A. Barbosa

Abstract— **Random number generators are indispensable for a multitude of tasks; from electronic games to secure communications. Most generators have been made either in software or determinist hardwired devices such as the Linear-Feedback-Shift-Registers; while gaining in costs or speed, the "random" sequences generated are actually deterministic, obeying clear generating algorithms, despite all randomness appearance of their outputs. From the other side, Nature presents a multitude of sources of true randomness that can be explored. Commercial random generators exist based on physical processes as the source of randomness. Difficulties are always present to extract Nature's randomness. This paper presents guidelines for construction of a fast (telecommunication speed) Physical Random Number Generator. It discusses the fundamental physical elements involved, technicalities of signal recording and its limitations, and the final bit extraction. The need for randomness tests is emphasized and the impossibility of guaranteeing true randomness of a finite sequence is discussed.**

Keywords— **Random, Physical processes, Cryptography.**

I. Introduction

RANDOM processes are intrinsic to Nature. This statement is usually accepted as a fundamental truth in Physics, together with the assumption that this randomness pervades the whole Universe. Quantum Mechanics itself is based on this randomness assumption. However, these beliefs are not unanimous [1].

A reason for this non-unanimity –even nowadays– is that probing Nature's randomness is a daunting task. Man's interaction with a random process interferes with the process itself or forcefully introduce filters during observation or detection of the involved phenomenon. The obtained outcome is always some biased picture of the fundamental process. This biasing is mostly created by the detecting instrumentation. Another fundamental problem is that the finite time window necessary to acquire data leads to samples of finite length. Being finite, they cannot fully characterize the random-like phenomenon: momentum powers of all orders necessary to a full characterization of a generic probability distribution cannot be obtained. One has to be satisfied with approximate results, not with the un-achievable idealized goal. Nevertheless, it is assumed that it is possible to obtain records of this "filtered" and finite set of data that passes many or all available statistical tests for a random phenomenon. One should be satisfied if no deterministic patterns are seen –the pragmatic approach normally used.

In a distinct way, man-made devices –consisting of electronic circuitry or software based– designed to produce the most possible randomness are *deterministic* devices by principle, regardless the complexity level that could be associated to them [2] such as the use of nonlinearities or superpositions of complex processes. These Pseudo Random Number Generators (PRNG) encompass the large majority of random generators in use nowadays.

Besides the randomness necessary for security applications other predicates are usually considered for a random number generator, including speed and cost. Speed is the second most desired feature, necessary for telecommunications.

Physical Random Number Generators (PhRG), by its turn, are devices trying to harness the random characteristics inherent to some physical phenomena. A PhRG designed to last in the existing fast advancing technological scenario should operate in *principles* that are untouched by the technology itself. As such, technological improvements can be incorporated in the system without modifications of the physical source being probed.

PRNGs have been widely described in the literature while PhRG implementations are not so common. In principle, PhRGs are free of the "deterministic" tag. Recent PhRG implementations include devices recording single-photon events [3] (detectors placed at the two ports of a beamspliter (BS)), Nyquist electrical noise [4] and chaotic lasers [5]. These PhRG implementations are not free of problems such as: the existence of bounds on speed due to the need of weak laser intensities [3] to avoid appearance of photons in both BS ports, slowness of electrical noise based processes [4], instabilities [5]. Nevertheless, they are a step forward in achieving true randomness, when compared to PRNGs.

This work shows steps necessary to construction of a Physical Random Generator (PhRG) based on the observation (fast detection and recording) of an elementary random *physical* phenomenon: photon number fluctuations at very short sampling times. The discussed device is aimed to extract intrinsic short-time intensity fluctuations of a coherent field (laser light) to produce random streams in a rate adequate for telecommunications.

II. Basic conditions for light sampling

A lasing device, fed by a current with a random stream of a large number of electrons can be used as the light source for a PhRG. A normal laser, gaseous or semiconductor, would fulfil this condition. The light state generated by a laser is well described by a coherent state where $\langle n\rangle = |\alpha|^2$ is the average number of photons in one coherence time τ_c, α is the complex laser amplitude, and the photon number variance is $\sigma^2 = \langle(\Delta n)^2\rangle = \langle n\rangle$; σ is the standard deviation. The probability for occurrence of n photons in a coherent state within sampling times $\Delta t \ll \tau_c$ is Poissonian distributed

$$p(n) = \frac{e^{-\langle n\rangle}\langle n\rangle^n}{n!}\,. \tag{1}$$

The probabilistic occurrence of these photon numbers reflects existing quantum fluctuations inherent to Nature and, in principle, they exist at all frequencies. The Poissonian occurrence of photon numbers has been called light's "shot-noise", like bal-

G. A. Barbosa, PhD, CEO of QuantaSec – Consulting and Projects in Physical Cryptography Ltd., Av. Portugal 1558, Belo Horizonte MG 31550-000 Brazil. Phone: +11 55 (31) 3441–4121, e-mail: GeraldoABarbosa@gmail.com

Invited Paper

listic occurrences of independent events (e.g., ∼ rain drops on a roof).

A single-mode laser will be discussed for this work. In the photon shot-noise limit (where the light noise predominates over other noise sources), intensity measurements can be performed to observe short time fluctuations ΔI, that deviate from the mean intensity $\langle I\rangle$ according to the Poissonian statistics (1):

$$\frac{\sqrt{\langle(\Delta I)^2\rangle}}{\langle I\rangle} = \frac{\sqrt{\langle(\Delta n)^2\rangle}}{\langle n\rangle} \rightarrow \frac{1}{\sqrt{\langle n\rangle}} . \tag{2}$$

Eq. (2) shows that the relative noise decreases as $\langle n\rangle$ increases. This makes deviations from the average intensity of an intense laser very difficult to be detectable.

A crucial characteristics associated to the statistical distribution given by (1) is that successive photon numbers, n_1 and n_2, present *no* correlation: $\langle n_1 n_2\rangle = \langle n_1\rangle\langle n_2\rangle$. This is the main property that guarantees that *if* one is able to extract these inherent fluctuations to generate bits, no correlations will appear among them.

As the physical phenomenon itself presents no bandwidth limitation, the PhRG can be made to follow any advances in optoelectronic technology. Usually the main speed restrictions arise from the light detector itself and the amplification circuitry bandwidth.

The device to be discussed [6] relies on the properties of a coherent light state, such as the one produced by a laser working well above threshold but with a damped intensity to increase the relative light fluctuations. This damping should be made by gray light filters (absorbers) without decreasing the laser current itself. This way, the coherent properties are preserved while the relative fluctuations are increased (See Eq. (2)). It may appear that decreasing the laser current could be a simpler way to get the desired low intensity. However, decreasing the current to obtain the desired light levels could put the laser close to the lasing threshold. Close to this threshold, the photon statistics are similar to the statistics of thermal fields. Differently from coherent state statistics, thermal fields present photon number correlations. These correlations are reflections of photon bunching that, given the occurrence of a photon, it is quite probable that a second photon will occur [7]. Therefore, whenever low intensities are desired the coherent state intensity shall not be diminished by drastically reducing the laser current. Instead, it should be damped with neutral filters. This avoids the mixing with thermal field statistics. This mixing produces detrimental features that will show up when statistical tests for randomness become more stringent.

Eventual integration of light source and detector on the same chip should produce the best light source and detector combination. However, this integration is not trivial nowadays but one should expect it to become more accessible in the near future. At the moment, coupling a laser source with a on-chip electronics circuitry is the way to go.

The detection circuitry, following a fast optical detector and a fast analog to digital conversor (AtoD) should discretize the input analog signals and discriminate for signals above and below the average intensity value. This “above” or “below” signals will be converted in fixed amplitude signals + or −, constituting the bit sequence. It is frequent that detection electronics are not perfectly symmetric in the charging and discharging of its circuitry. This may lead to asymmetric amplitude distributions and procedures are usually taken to minimize this problem. One way is to work with the time derivatives of the fluctuating signals [5], or else, a differential phase shift keying scheme (DPSK) can be used, where the difference of two successive modulations defines the bit, either 0 if no change occurs or 1 if a change has occurred.

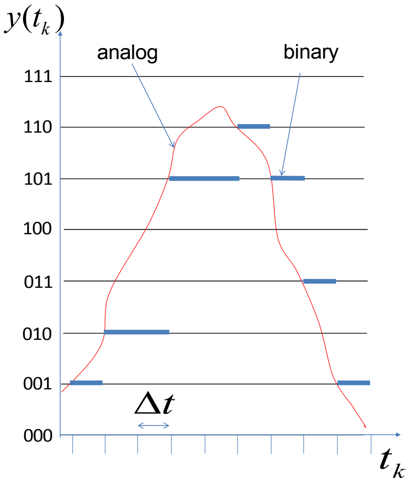


Figure 1. Binary levels (blue) representing an analog signal (red) varying in time t, with a 3-bit ADC. The number of available levels, above zero, is $2^3 - 1$. With an n-bit ADC and digital time units Δt, the digital output values are given by $y(t_k) = b_n(t_k)2^{n-1} + b_{n-1}(t_k)2^{n-2} + \ldots + b_2(t_k)2^1 + b_1(t_k)2^0$ ($t_k = k\Delta t$). Following this rule, the notation at the plot ordinate represents the sequence of three bits b_3, b_2, b_1.

III. Mesoscopic States and Bit Recording

As Eq. (2) indicates, for large intensity, the relative number fluctuation goes to zero. Some constraints already discussed are here repeated to emphasize general requirements to achieve the operational level:

1) A low current, lasers deviate from the coherent state operation. Therefore, to obtain coherent states in the *mesoscopic* regime (above strictly quantum but below very large intensities where fluctuations are negligible), the laser intensity is decreased through use of neutral filters and *not* by decreasing the electronic current.

2) Usual communication detectors do not have single photon sensitivities. Their minimum detection threshold are usually of the order of a few hundred of photons. This indicates that one has to utilize photon numbers, in Δt, above $\langle n\rangle_{\Delta t} \sim 10^3$.

3) A third constraint relates to the use of fast (linear) analog-to-digital (AtoD) recorders. An analog signal s is digitally recorded using b bits that produce 2^b levels equally spaced. See Fig. 1. Analog signal levels occurring within bit levels are rounded by the digital technique. The spacing between the binary levels define the available ADC's resolution. An ADC with

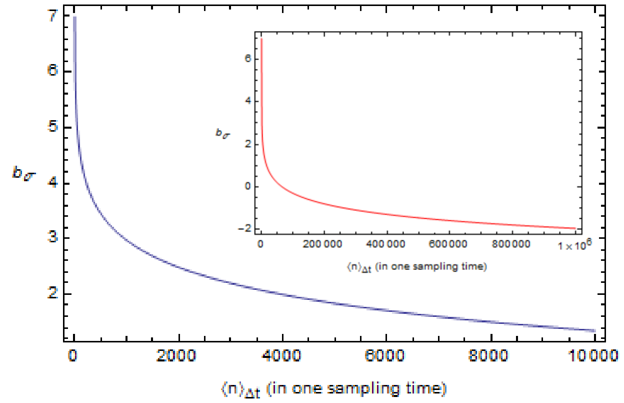


Figure 2. b_σ versus $\langle n\rangle_{\Delta t}$, for $b = 8$ ($2^b = 256$ levels), in the mesoscopic range. Inset for higher intensities. $\lambda = 1.55\mu$m, $\tau_c = 0.3\mu$s, or $\tau_c/\Delta t = 300$. It is seen that for higher intensities, the AtoD recorder saturates ($b_\sigma < 1$).

a high number of bits has an increased resolution ($1/2^n$ of the full signal range) but usually has a decreased speed. Small bit-number ADC (e.g., 8 bit) are usually faster and preferable, say, for telecommunication uses.
4) Use of a single detector reduce costs and eliminates the need for intensity balance in homodyne setups. Homodyne detection, while presenting a simple way to eliminate the average intense signal and extracting the desired noise, demands a constant precise balance, both optically as well as electronically, of the two detectors. The single detector use, when optimized for extraction of signals in the desired intensity range, offers a lower cost system without compromising speed.

A. *Bits for average signal and noise*

The digital recording of a laser light intensity signal as a function of time should reveal both the average signal level (or $\langle n\rangle$) and the fluctuations ($\pm\sigma$) around the average. For high intensity, the σ contribution for the signal gets smaller than the ADC's resolution and only the average signal is detected. Working in such conditions would rule out the possibility to have a record of σ. An ADC's with b bits has to accommodate both average and signals around average: $\langle n\rangle_{\Delta t} \pm \sigma$. Moreover, $\pm\sigma$ should be detectable by the ADC. Assuming that the laser intensity $I \propto \langle n\rangle_{\Delta t}/\Delta t$ and that the optical detector operates in a linear regime, it is expected that the relative proportion holds:

$$\frac{\langle n\rangle_{\Delta t}+\sigma}{\sigma} = \frac{2^b}{2^{b_\sigma}} \rightarrow b_\sigma = b + \log_2 \frac{\sigma}{\langle n\rangle_{\Delta t}+\sigma}, \qquad (3)$$

where b_σ is the number of bits "reserved" for σ (within the set of bits b).

Fig. 2 illustrates the dependence of b_σ with $\langle n\rangle_{\Delta t}$ for an analog-to-digital recorder of $b = 8$. It is seen that for a few thousands of photons a few bits b_σ are available to record the fluctuation σ. However, as the number of photons increase, the inset shows that the analog-to-digital recorder saturates and no bits b_σ are available to record the fluctuation σ.

The operation regime for the laser should be "shot-noise" limited, where the Poissonian statistics of light predominates well above thermal radiation residues and electronic noises. The establishment of conditions to guarantee shot-noise operation is not trivial. Section "Methods" sketches general guidance lines to achieve shot-noise limited signals.

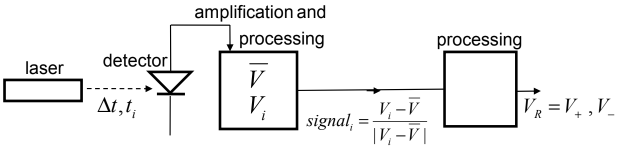


Figure 3. A shot-noise limited laser illuminates a fast light detector. After amplification, the signals V_i are recorded within a time interval Δt around time t_i. A processing circuit average the signals and classify each of them above or below the average $\overline{V}$. Random signals above or below the average value are converted in constant voltage amplitudes V_+ or V_- representing the random bits.

IV. PhRG's block diagram and commercial components

The PhRG's block diagram is sketched on Fig. 3. A laser with sufficient intensity to produce shot-noise limited light – where the light noise predominates over all electronic noises– illuminates a fast light detector. The processing units may contain an ADC for fast processing. Just to consider some concrete examples, using commercially available components, a fiber-optic connected continuous wave (CW) laser operating in single mode can be used, at $\lambda = 1.550\mu$m, bandwidth $\Delta\nu < 477$ kHz, and with a controlled temperature of 25^0C; its coherence time is $t_C \simeq 0.3\mu$s. An InGaAs PIN detector could be used, with a bandwidth $\Delta\nu \simeq 2$GHz, with a transimpedance amplifier operating under a battery power source supplying 5V and a photovoltage bias of 10V; the detector responsivity is 0.8A/W. The amplified signals can be recorded by a 1GHz analog-to-digital (AtoD) circuitry with 1Gb of memory and 8 bit resolution. The signals are to be acquired within time windows $\Delta t \simeq 10^{-9}$s, much shorter than the laser coherence time t_C and, therefore, representing the true statistics of the light fluctuations, as given by Eq. (1). A signal processor average the signals in the AtoD and classify the recorded data as being above or below their average with signal$_i = (V_i - \overline{V}/|V_i - \overline{V}|)$. Further processing converts the signal$_i$ sequences into constant amplitude signals V_+ or V_- that represent the sequence of random bits. Any formatting can be applied to this output stream.

A. *Signal simulations*

The described PhRG is part of an effort [8] to develop new cryptographic schemes. As it is not yet ready for operation, some computer simulations will be presented for pedagogical purposes. This way the reader can better understand the comments already made.

For the moment, the ADC operation will be ignored and analog signals will be treated for simplicity (for the ADC operation just think of discretized levels). Fig. 4 shows a sample of an analog CW signal (simulated as a signal taken at the amplification output stage). Comparisons, say, at each Δt (e.g., 2×10^{-9}s), between the instantaneous obtained value for the intensity (or an output voltage V) with respect the average intensity produce the bit values bit_i according to the rule

$$\text{signal}_i = \frac{I_i - \overline{I}}{|I_i - \overline{I}|}, \text{ and } \text{bit}_i = \frac{1+\text{signal}_i}{2}. \qquad (4)$$

For the fluctuations within the inset in Fig. 4, rule (4) gives the binary sequence 0,1,0,1,1,1,0,1,1,1,0,1,0,1, 0,1,0,0,1,0,1,1,1,0,1,0,1,1,1,0,1,1,0,0,1,0,0,0,1,1,1,0,0,0,1,0, 1,0,1,1.

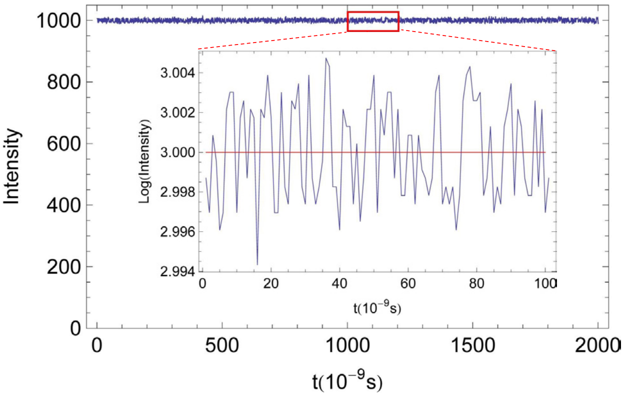


Figure 4. Simulation of a low laser intensity with Poissonian fluctuations as a function of time (the Intensity units are arbitrary). Inset: Detail of the laser intensity (log) within the small time window (in red). The red line indicates the mean intensity value.

A PhRG should operate continuously generating bits in a very high rate. The single laser and single detector scheme are overly superior with respect to stability than homodyne systems or comparison systems where two detectors are used.

V. Random number sequences: generalities

Given a finite sequence of supposedly random numbers, say the bit sequence above, one may ask if this sequence is truly random or not.

Quite generally, a physical source of entropy or numbers sequences n can be characterized by a probability distribution $p(n)$. Equivalently, $p(n)$ can be characterized by its moments of all orders (e.g., $\langle n \rangle, \langle (n - \langle n \rangle)^2 \rangle, \ldots$). However, being finite a sequence could not reveal all moments of the statistical source. Similarly, the occurrence of 0s and 1s, and distinct groups of 0 and 1 have to occur randomly. Observation of a finite sequence of 0s and 1s may reveal group patterns in the sequence. Patterns can be generated deterministically. That means that the given finite sequence could be compressed. Differently, a true random source should generate a sequence that, as its length increases, the associated entropy will increase linearly producing a sequence that could not be compressed.

The idea of randomness is perhaps better appreciated through the concept of "complexity of a string" [9]: The complexity of a string s is the length of the string's shortest description in some fixed universal description language. In other words, the complexity of a string is defined by the length of the *program* that describe that string. For example, a sequence of 10^6 consecutive "1"s, followed by another sequence of 10^6 consecutive "0"s produce a sequence of 2×10^6 bits. However, a short program such as "From $i = 1$ to 10^6, Print 1. From $i = 10^6 + 1$ to 10^6, Print 0 " produces the same sequence – with a short program. In other words, the sequence can be highly compressed and, therefore, is not random.

A description of s of minimal length, $d(s)$, uses the fewest number of characters and it is called a *minimal* description of s. The length of $d(s)$, i.e. the number of characters in the description, is the Kolmogorov complexity of s, written $K(s) = |d(s)|$. Unfortunately, K is not a computable function [10].

Nevertheless, it is clear that, under this definition, a perfect random sequence will need a program at least as long as the string itself to define the string, that is to say, the sequence cannot be compressed. Although these definitions may clarify the difficulties involved, they do not help much in the practical sense.

A. Statistical tests

The best one can do evaluate randomness is to apply a variety of tests. A satisfactory sequence that passes a given test will be said "random for that particular test".

There are known statistical test suites developed for this purpose. An example is the "A Statistical Test Suite for Random and Pseudo-random Number Generators for Cryptographic Applications", described in NIST's Special Publication 800 - 22/Revision 1:
http://csrc.nist.gov/groups/ST/toolkit/rng/documentation-software.html .

Another one is the "DieHard" battery of tests:
http://www.stat.fsu.edu/pub/diehard/ .

VI. From signal detection to signal-to-noise ratio: METHODS

This section assumes that the readers have some familiarity with basic concepts of quantum mechanics and that some of the cited references are to be consulted. It also supposes familiarity with basic thermodynamics. Those not interested in these formalisms should skip the derivations. However, the resulting equations can be used; they are the end-products of the section.

The laser source characteristics adequate as an optical noise source for a PhRG were briefly discussed. Fast detectors are another essential part of the device. Detectors are also sources of noise (purely shot and thermal noises) and the understanding of the mixtures of light noise and other noises arising from detectors have to be understood to allow one to control or balance these sources and to make possible extraction of uncorrelated bits.

Above all, detectors are an important part of our tool set to understand the Universe. As a short comment, at the instant of their actions, detectors define our interface between the past and the future, in the classical view of time as a constantly moving arrow. What they record (=past) can be checked against our predictions (=future) of this same event and contribute to our primary sketch of the Universe. These logs are classical, in the sense that they can be faithfully copied. Interpretation of these records one-by-one or, in a correlated form, gives support or not to our immediate expectations or even to more broad concepts as our views of a classical or quantum world. Therefore, our understanding of the detector's construction not only show

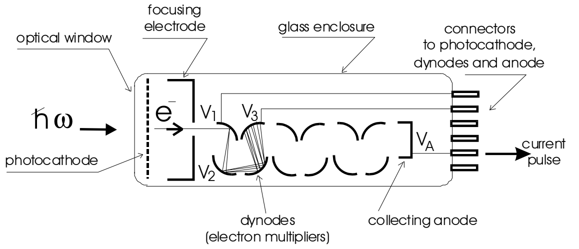


Figure 5. *PMT elements and gain mechanism* – A photon may transfer energy to a photoelectron in the photocathode. This electron hits the first dynode after acceleration by a voltage difference, ejecting electrons from the material. Successive accelerations and collisions result in a charge pulse at the collecting anode.

their limitations but also allow us to improve them, widening our perceptions about the Universe.

This section discusses optical detectors and, in particular, detectors that operate by photo-electron absorption processes. The optically sensitive materials used in their construction limit their wavelength bandwidth and their amplification electronics usually impose their frequency bandwidths. Noise sources will also be discussed as well as mathematical tools to understand their inner-workings. Only *direct* detection, used for the presented PhRG, is treated here; homodyne and heterodyne detection will not be discussed.

Among commercial detectors are the single photon sensitivity detectors, known as photon-counting detectors (SPCDM, or single photon counting detection modules), *photon multipliers* (PMT), and integrated semiconductor detectors known as *avalanche photo-diodes* (APD). Silicon PIN detectors are also used for low-light detection but their sensitivity is much lower than APD's and they are not adequate for single photon counting. However, an APD can be built with a faster electronics and have a wide use in telecommunication with a corresponding lower cost. Less common are the cryogenic detectors operating by photon absorption and using photon-to-thermal energy conversion. They can have very high sensitivities and resolution but they are slower. For a review on single-photon detectors see [11]. This note is not concerned with single photon detectors but with detectors for telecommunication that, apart of being low-cost compared with single photon detectors, could be able to detect mesoscopic number of photons. The expected photoelectron signals at the detector output are well represented by analog signals.

Usually, silicon APDs are optimized to work between 300 and 1100 nm, germanium between 800 and 1600 and InGaAs from 900 to 1700nm. Only silicon APDs present dark current low enough for commercial use in *non-gated* single photon detectors. Non-silicon APDs can also be used in gated operation for single-photon counting; the gate operation avoids the excessive thermal noise that builds up if they were used in a continuous operation and help diminishing after-pulses when operated to be sensitive to weak avalanches [12], [13].

The detectors based on the photoelectric effect can be classified as *annihilation* detectors. Figs. 5 and 6 sketch the gain mechanisms in these devices.

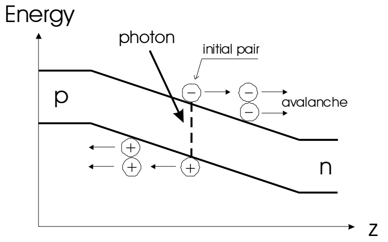


Figure 6. *Gain mechanism for an APD* – A photon creates an electron-hole pair in a semiconductor *p-n* junction. A strong field is maintained at the junction such that these charges are accelerated and, whenever they gain energy ($\geq E_g$, the gap energy) a secondary pair will be created at the collisions. Each new pair may contribute to create an "avalanche" of charges at the collecting outputs. The physical structure of an APD may contain a volume where light absorption creates electron-hole pairs and an electric field separates the two kinds of charges and sweeps one of the carriers towards a multiplication region where a strong electric field accelerates this charge causing impact ionizations as a gain mechanism.

A. Symmetric and asymmetric devices

Annihilation detectors differ from field detectors (antennas) in fundamental aspects. One of them is that annihilation detectors are not sensitive to the field polarization, whereas an antenna is. Another difference resides in the action of the photon annihilation and creation operators $\widehat{a}$ and $\widehat{a}^+$ (These are the fundamental quantum operators that annihilate and create photons). In the classical limit of optics $\hbar\omega \to 0$ and, therefore, a detector in contact with the heat bath at temperature T_K may, with high probability, gain sufficient energy to emit a photon with energy $\hbar\omega \ll k_B T_K$ (k_B is the Boltzmann's constant). In other words, $\widehat{a}$ and $\widehat{a}^+$ will have similar contributions in the interaction process. In the microwave range, $\hbar\omega \sim k_B T_K$. In the optical range $k_B T_K \ll \hbar\omega < 2m_0c^2$ (the upper limit taken as the energy of an electron-positron pair creation); therefore, the heat bath will have a small probability to create a photon. One may also say that in the optical range annihilation processes dominate over creation ones—an asymmetry between these processes. Single-photon sensitivity detectors are usually submitted to cooling processes to further reduce the probability of *dark* noise or electron emission in the detector when no desired light is present. In the optical range, the interaction between detector and field then proceeds mainly through the electric field operator $\widehat{E}^{(+)}$ (that involves only annihilation field operators).

B. Quantum Efficiency

The intrinsic bandwidth $\Delta\omega$ of a detector is basically determined by the material employed to make the photocathode and other elements such as the optical material of the collecting window. The electronic circuitry after the detecting elements will also contribute to the effective bandwidth of the detecting system. In general, one is interested in a detection bandwidth $\delta\omega$ quite narrow compared with the optical frequency ω_0 of the incident photons ($\delta\omega \ll \omega_0$.)

The photoelectric material is made with a very low value of the *work function* (the energy necessary to extract an electron from the material), which defines the lower limit for the detectable optical frequency. The material employed in the optical window of a cooled detector frequently defines the maximum optical frequency detectable.

The emission probability associated with the photoelectric effect has a conversion or *quantum efficiency* η_{pe} smaller than unity ($\eta_{pe} < 1$). The *quantum efficiency* is measured in the averaging process of converting photons to photoelectrons (electrons emitted in the photoelectric effect), and is given by

$$\langle n_{pe} \rangle = \eta_{pe} \langle n \rangle \,, \tag{5}$$

where $\langle n_{pe} \rangle$ is the average number of photoelectrons emitted and $\langle n \rangle$ is the average number of incident photons reaching the detector area A. This implies that there is *no* assurance that a photoelectron will appear after a given photon hits the photoelectric material – (5) only express *average* quantities.

This parameter η_{pe} has a stochastic origin related to atomic processes. One may interpret this uncertainty in the emission time of an excited atom as due to stochastic fluctuations of the *vacuum* electric field (existing field in the absence of photons). $\eta_{pe} < 1$ also impose limitations on experiments involving photon pair detection with two detectors (*coincidence* detection), because each undetected photon results in a loss of coincidence between the detectors.

A photon detector is usually made to multiply the initial charge ejected from the photocathode to result in a charge pulse easily treated by conventional electronics. This amplification process is known as the detector *gain G*. After this gain process an electric current $i(t) = Ge(dn_{pe}/dt)$ appears at the anode. The ratio σ between the photoelectric current density and the incident photon intensity is

$$\sigma = \frac{e\frac{dn_{pe}}{dt}}{\hbar\omega \frac{dn}{dt}} = \frac{e}{\hbar\omega}\frac{dn_{pe}}{dn} = \frac{e}{\hbar\omega}\eta_{pe} \,. \tag{6}$$

The ratio σ, known as "radiant sensitivity" of the photocathode, is usually furnished by the detector's maker. In a photomultiplier, it can be measured extracting the charge pulse directly from the first dynode (See Fig. 5), thus avoiding the gain mechanism.

C. *Temporal Response; Amplification and Discrimination; Formatting*

The gain involve processes occurring in a time interval τ_d. In APDs, within this time, a newly arriving photon cannot produce a distinct amplification pulse and thus produces no count in the external electronics. τ_d defines the detector's dead time. Some new detectors, including cryogenic ones, aim to identify the arrival of two or more photoelectrons within τ_d.

For some applications, such as coincidence counts between two detectors, in order to shorten the time resolution below τ_d, electronic techniques utilize the rate of increase (or decrease) of charge variation (time derivatives during a pulse formation) from one detector to trigger a time counter. A time is measured when the second detector gives a signal. This way, current commercial detectors may present a time resolution of $\sim 10^{-10}$s between two events (shorter than the dead time τ_d).

Some noise sources contribute to degradation of the photodetection, among them thermionic emission (proportional to temperature and dependent on specific materials) and cosmic rays. Photodetector engineering tries to optimizes their signal to noise ratio. Cosmic rays can be eliminated in coincidence detection, due to the negligible probability of both detectors being excited simultaneously. They cannot be eliminated in a single detector but are minimized by a small detection area.

In a general way, an electronic circuit *amplifies* the signal appearing after the gain process and chops some of them in a *discrimination* process to reduce events that come from thermal noise. Electrons emitted due to thermal emission produce charge pulses of less intensity in the gain stage, because they usually have much less initial kinetic energy than those produced in the photoelectric effect. This fact can be used to set a discrimination level to result in one charge pulse for each photoelectron emitted. *Formatting* electronics are now usually built into many detecting systems giving approximately a standard digital output (TTL, ECL, etc.) for each analog charge pulse generated.

D. *The Quantum Process of Photodetection (basics)*

The theory of photo-detection is an area of study by itself [14]. Some outstanding landmarks were established by Glauber with his work on optical coherence and by Mandel on the theory of photon statistics [15]. The reader is strongly suggested to consult these references. In this section, a simplified approach to the generation of photoelectrons from single photon streams, using a phenomenological response function [16], is utilized to introduce some readers in this subject.

An electric field quantum operator $\widehat{E}(z,t) = \widehat{E}^{+}(z,t) + \widehat{E}^{-}(z,t)$, where $\widehat{E}^{-}(z,t) = (\widehat{E}^{+}(z,t))^{+}$, describes light propagation along the z-axis in an isotropic medium with dielectric susceptibility ϵ, where $\omega = kv = kc/n$. For example, one could write, for a x-polarized field

$$\widehat{\mathbf{E}}^{-}(z,t) = \hat{\mathbf{x}} \sum_{\omega} \sqrt{\frac{\hbar\omega}{2\epsilon V}}\, \widehat{a}^{+}_{\omega} \exp\left[-i\omega\left(z/v - t\right)\right] \,, \tag{7}$$

where V is the quantization volume. In the classical limit, quantum operators $\widehat{a}_{\omega}$ and $\widehat{a}^{+}_{\omega}$ become the field amplitudes a and a^{*}.

If one considers ω in a continuum, it may be convenient to write the Hamiltonian $\widehat{H}$ for a free mode and the number operator $\widehat{N}$ as

$$\widehat{H} = \int_{0}^{\infty} \hbar\omega\, \widehat{a}^{+}(\omega)\widehat{a}(\omega)\, d\omega \,,\ \widehat{N} = \int_{0}^{\infty} \widehat{a}^{+}(\omega)\widehat{a}(\omega)\, d\omega, \tag{8}$$

where $[\widehat{a}(\omega), \widehat{a}^{+}(\omega')]_{-} = \delta(\omega - \omega')$. The practical transition from a discrete to a continuum is made by substituting $\sum_{k_z} \rightarrow (L_z/2\pi)\int dk_z$, where L_z is the quantization length of the field and writing the quantization volume V as the product of the mode area A_c times the length $L_z = v\delta t$ ($v = c/n$), where $\delta t = 1/\delta\nu$ is the separation time interval between modes. Also $\widehat{a}(\omega) \rightarrow \widehat{a}_{\omega}/\sqrt{\delta\omega}$, giving

$$\widehat{E}^{-}(z,t) = \frac{i}{\sqrt{2\pi}} \int_{0}^{\infty} d\omega \sqrt{\frac{\hbar\omega}{2\epsilon A_c v}}\, \widehat{a}^{+}(\omega) \exp\left[-i\omega\left(z/v - t\right)\right]. \tag{9}$$

As discussed previously, one is usually interested in a narrow frequency range around the average field frequency ω_0. Writing $\omega = \omega_0 + \omega'$, this condition is $\omega'/\omega_0 \ll 1$ and in this case,

$$\widehat{E}^{-}(z,t) \simeq i\frac{1}{\sqrt{2\pi}} \sqrt{\frac{\hbar\omega}{2\epsilon A_c v}} \exp\left[i\omega_0\left(z/v - t\right)\right]$$

$$\times\int_{-\infty}^{\infty} d\omega'\,\widehat{a}^{+}(\omega')\,\exp\left[-i\omega'(z/v-t)\right] . \tag{10}$$

The field intensity operator, $\widehat{I}(t)=\widehat{E}^{-}(t)\widehat{E}^{+}(t)$, is connected to the photoelectric current operator $\widehat{I}_e$ through a response function $D(t-t')$ of the photodetector by

$$\widehat{I}_e(z,t)\equiv\frac{\widehat{N}_e(t)}{dt}=e\int_{-\infty}^{\infty}dt'\,D(t-t')\widehat{E}^{-}(z,t')\widehat{E}^{+}(z,t') . \tag{11}$$

The phenomenological function $D(t-t')$ defines the causal process generating the electric current in time t. If one considers the time response of the photodetector to be much shorter than the frequency bandwidth considered, the response time can be approximated by $D(t-t')\simeq D\delta(t-t')$ and, therefore, substituting the above definitions in Eq. (11) results in

$$\begin{aligned}\widehat{I}_e(z,t)=eD\frac{\hbar\omega_0}{4\pi vA}\int_{-\infty}^{\infty}d\omega'\int_{-\infty}^{\infty}d\omega''\,\widehat{a}^{+}(\omega')\,\widehat{a}(\omega'')\\ \times\exp\left[-i(\omega'-\omega'')(z/v-t)\right] .\end{aligned}\tag{12}$$

The operator number for the photoelectrons and the current operator for these same electrons, in this "instantaneous" response approximation, are related by

$$\begin{aligned}\widehat{N}_e(z)&\equiv\int_{-\infty}^{\infty}\widehat{I}_e(z,t)dt\\ &=eD\frac{\hbar\omega_0}{4\pi\epsilon vA}\int_{-\infty}^{\infty}d\omega'\int_{-\infty}^{\infty}d\omega''\widehat{a}^{+}(\omega')\,\widehat{a}(\omega')\\ &\times\exp\left[-i(\omega'-\omega'')z/v\right]\\ &\times\int_{-\infty}^{\infty}dt\exp\left[-i(\omega'-\omega'')t\right]\\ &=eD\frac{\hbar\omega_0}{2\epsilon vA}\int_{-\infty}^{\infty}d\omega'\widehat{a}^{+}(\omega')\widehat{a}(\omega')=eD\frac{\hbar\omega_0}{2\epsilon vA}\widehat{N}.\end{aligned}\tag{13}$$

An average can be taken over the number operators for the photoelectrons and for the photons, giving

$$D\langle\widehat{N}\rangle=\frac{2\epsilon vA}{e\hbar\omega_0}\langle\widehat{N}_e\rangle . \tag{14}$$

Using the definition of the detector efficiency,

$$D=\frac{2\epsilon vA}{e\hbar\omega_0}\frac{\langle\widehat{N}_e\rangle}{\langle\widehat{N}\rangle}\equiv\frac{2\epsilon vA}{e\hbar\omega_0}\eta_{pe} . \tag{15}$$

The photoelectron current operator can now be written

$$\begin{aligned}\widehat{I}_e(z,t)=e\frac{\eta_{pe}}{2\pi}\int_{-\infty}^{\infty}d\omega'\int_{-\infty}^{\infty}d\omega''\widehat{a}^{+}(\omega')\,\widehat{a}(\omega'')\\ \times\exp\left[-i(\omega'-\omega'')(z/v-t)\right] ,\end{aligned}\tag{16}$$

and from the definition of a Fourier transform $f(t)=(1/\sqrt{2\pi})\int_{-\infty}^{\infty}d\omega f(\omega)\exp i\omega t$, one arrives at the *instantaneous* photoelectron current operator

$$\widehat{I}_e(z,t)=\frac{d\widehat{N}_e}{dt}=e\eta_{pe}\widehat{a}^{+}(t-z/v)\widehat{a}(t-z/v) . \tag{17}$$

From now on z will be taken as $z=0$. The number operator $\widehat{a}^{+}(t)\widehat{a}(t)$ is the photon number *intensity* at time t (Units of $\widehat{a}(t)$ are $t^{-1/2}$; see Eqs. (8)). Taking averages of the operators one arrives at

$$\langle d\widehat{N}_e\rangle=e\eta_{pe}\langle\widehat{a}^{+}(t)\widehat{a}(t)\rangle dt\equiv R_1(t)dt=dP_1(t) , \tag{18}$$

which defines the differential photodetection probability $dP_1=R_1(t)dt$ for *one* photoelectron in t within dt. The rate of photodetection—for single events—is then

$$\frac{dP_1}{dt}=R_1(t)=e\eta_{pe}\langle\widehat{a}^{+}(t)\widehat{a}(t)\rangle . \tag{19}$$

Consider a photon state $|\psi\rangle$ in the number representation, describing single photons at instant times $t_1,t_2,\ldots t_n$: $|\psi\rangle=|1_{t_1},1_{t_2},\ldots 1_{t_n}\rangle$. Applying the photoelectron current operator $\widehat{I}_e$ to $|\psi\rangle$, one has

$$\begin{aligned}\widehat{I}_e|\psi\rangle&=e\eta_{pe}\widehat{a}^{+}(t)\widehat{a}(t)|1_{t_1},1_{t_2},\ldots 1_{t_n}\rangle\\ &=e\eta_{pe}\widehat{a}^{+}(t)\widehat{a}(t)\left[\widehat{a}^{+}(t_1)\widehat{a}^{+}(t_2)\ldots\widehat{a}^{+}(t_n)|0\rangle\right].\end{aligned}\tag{20}$$

Successive applications of $[\widehat{a}(t),\widehat{a}^{+}(t_j)]=\delta(t-t_j)$ to the products of operators in this equation gives

$$\begin{aligned}\widehat{I}_e|\psi\rangle&=e\eta_{pe}\widehat{a}^{+}(t)\left[\delta(t-t_1)\widehat{a}^{+}(t_2)\ldots\widehat{a}^{+}(t_n)\right.\\ &\quad\left.+\widehat{a}^{+}(t_1)\widehat{a}(t)\widehat{a}^{+}(t_2)\ldots\widehat{a}^{+}(t_n)\right]|0\rangle=\ldots\\ &=e\eta_{pe}\left(\sum_{i=1}^{n}\delta(t-t_i)\right)|\psi\rangle,\end{aligned}\tag{21}$$

showing that the eigenvalue of the photoelectron current operator is a succession of sharp charge pulses at instants t_i (this result is qualitatively intuitive). Of course, different models for the response function D, instead of $D(t-t')=D\delta(t-t')$, give different distributions for the resulting current. With these, the delta pulses in Eq. (21) will be modified to pulses with less sharp shapes. In fact, good descriptions of practical current pulses can be achieved with simple models for D. For example, functions that depend only on a small number of "moments" (or Fourier components) are particularly useful; such as

$$D(t-t')\simeq\frac{\mu}{\sqrt{\pi}}e^{-\mu^2(t-t')^2} \tag{22}$$

where μ is adjusted to fit charge pulses that usually depend on the particular detection system in use.

However, Eq. (21) gives a good pictorial view of the "shot-noise" process.

E. Photon Detection Probability and Field Distributions

A useful connection between photon number and field distributions can be derived using the coherent basis representation $|\alpha\rangle$ defined [14] as (k_s designate modes) $\widehat{a}_{k_s}|\alpha_{k_s}\rangle=\alpha_{k_s}|\alpha_{k_s}\rangle$. The coherent state $|\alpha\rangle$ introduced by Glauber is

$$|\alpha\rangle=e^{-|\alpha|^2/2}\sum_{n=0}^{\infty}\frac{\alpha^n}{\sqrt{n!}}|n\rangle . \tag{23}$$

In the coherent basis the average photon number will be $\langle \alpha_{k_s} | \hat{a}^\dagger_{k_s} \hat{a}_{k_s} | \alpha_{k_s} \rangle = |\alpha_{k_s}|^2$. A field distribution in the diagonal representation [14] is written $P(\{\alpha_{k_s}\})$. $P(\{\alpha_{k_s}\})$ is associated to each specific field (laser, thermal etc). A density operator ρ for this field can be written

$$\rho = \int P(\{\alpha_{k_s}\}) |\{\alpha_{k_s}\}\rangle\langle\{\alpha_{k_s}\}| d^2\{\alpha_{k_s}\} \; , \tag{24}$$

and normalized with $Tr[\rho] = \int P(\{\alpha_{k_s}\}) d^2\{\alpha_{k_s}\} = 1$. A photon number probability distribution can be defined as

$$p(\{n_{k_s}\}) = Tr\left[\rho |\{n_{k_s}\}\rangle\langle\{n_{k_s}\}|\right] \; . \tag{25}$$

Working with these equations and summing the total number of photons counted $n = \sum_{\{n_{k_s}\}} n_{k_s}$, the Mandel's relationship connecting $p(n)$ and $P(\{\alpha_{k_s}\})$ is obtained:

$$p(n) = \int P(\{\alpha_{k_s}\}) \frac{\left(\sum_{k_s} |\alpha_{k_s}|^2\right)^n}{n!} e^{-\sum_{k_s} |\alpha_{k_s}|^2} d^2\{\alpha_{k_s}\} \tag{26}$$

and for a single mode case $\{\alpha_{k_s}\} \to \alpha$

$$p(n) = \int P(\alpha) \frac{|\alpha|^{2n}}{n!} e^{-|\alpha|^2} d^2\alpha \; . \tag{27}$$

Several probability distributions can be calculated as, for example,

1. *Laser with amplitude and phase constant or uniform phase*

$$p(n) = e^{-\langle n \rangle} \frac{\langle n \rangle^n}{n!} \; , \langle n \rangle = |\alpha|^2 \, . \tag{28}$$

2. *Thermal field (random phases)* (See Section VIII, Eq. 8.8. in Ref. [14])

$$p(n) = \frac{1}{1 + \langle n \rangle} \left(\frac{\langle n \rangle}{1 + \langle n \rangle} \right)^n \; . \tag{29}$$

3. *Superposition of laser and thermal light*

$$p(m) = \frac{\langle n_T \rangle^m}{(1 + \langle n_T \rangle)^{m+1}} \times L_m \left[-\frac{\langle n_L \rangle}{\langle n_T \rangle (1 + \langle n_T \rangle)} \right] e^{-\frac{\langle n_L \rangle}{1 + \langle n_T \rangle}} \; , \tag{30}$$

where L_m is the Laguerre function, $\langle n_L \rangle$ and $\langle n_T \rangle$ are the average numbers for the laser photons and thermal photons, respectively.

F. Noise considerations

The detection process has some fundamental random contributions: 1) The photon absorption is statistical in nature. 2) The immersion of detector and associated electronics in the environment at temperature T_K produces electronic thermal excitations or thermal noise that are also recorded [4]. 3) The electronic avalanche in the gain process is statistical.

Data recording at high speeds, such as done by AtoD converters, introduce their particular error sources. Traditional technical sources of error include nonlinearities in the conversion processes, electro-magnetic interferences, gain error produced by amplifier distortions, offset error, and AtoD conversion errors. Some of these may become a significant source of error. Clock jitter, for example, introduces uncertainty in the collection time of the signal. Fast AtoD converters may present cross-talk between the analog and digital components. Many of these error sources can be technically reduced.

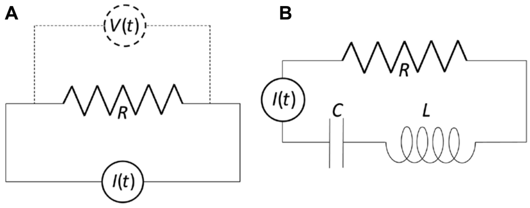


Figure 7. A: A real resistance R at temperature T_K can be represented as a lossless resistance R where a current $I(t)$ appears due to the thermally excited electrons. A voltage $V(t)$ appears at the resistance ends. B: RLC components in series.

A parameter that gives a good estimate of the signal that could be obtained under presence of fundamental noises is the *signal-to-noise* ratio *SNR*. This ratio could be defined, in the *number basis*, as the ratio of the average signal square to the variance

$$SNR = \frac{\langle n \rangle^2}{\langle (n - \langle n \rangle)^2 \rangle} \; , \tag{31}$$

where $\langle n \rangle$ is the average detected number of photons. SNR parameters can be written for any quantity of interest such as voltages, currents, phase and so on. As a simple warning, another common use is writing SNR as the square root of (31).

G. Fluctuation spectra

Detectors are usually connected to an impedance that could be, in the simplest case, a resistor or the effective resistance of a pre-amplifier. Understanding the effects of the thermal noise in this resistor by itself is important to derive the effective noise of a detector coupled to an external circuit. Even neglecting microscopic aspects describing the behavior of electrons in the resistor, thermodynamic arguments and macroscopic reasonings are of great help to understand this noise source.

A resistor R coupled to an ideal amplifier tuned at a frequency $\overline{\omega}$ with a bandwidth $\Delta\omega$ will produce a fluctuating signal in the amplifier. This signal can be traced by the current $I(t)$ generated by electrons set in motion by thermal energy. A corresponding fluctuating voltage emf $V(t) = RI(t)$ will be detected across the resistor. An equivalent circuit is shown in Fig. 7-A. A more general circuit to represent a real resistance connected to LC components in series is in Fig. 7-B. All components are assumed to be at thermal equilibrium at temperature T_K under ergodicity conditions. A voltage $V_j(t)$ will appear at each j-component ends. The equipartition theorem for the energy establishes that for each degree of freedom the average energy is $k_B T_K/2$. Therefore,

$$\langle \frac{1}{2} L I(t)^2 \rangle = \frac{1}{2} k_B T_K \text{ and } \langle \frac{1}{2} C V_C^2 \rangle = \frac{1}{2} k_B T_K \, , \tag{32}$$

where V_C is the potential difference across the capacitor. Thus, for example

$$\langle I(t)^2\rangle = \frac{k_B}{L} T_K \tag{33}$$

Parseval's theorem gives $\int_{-\infty}^{\infty} I(t)^2 dt = \int_{-\infty}^{\infty} |\widetilde{I}(\nu)|^2 d\nu$. The average $\langle I(t)^2\rangle$ can be expressed by

$$\begin{aligned}\langle I(t)^2\rangle &= \lim_{\tau\to\infty}\frac{1}{\tau}\int_{-\tau}^{\tau} I(t)^2 dt = \lim_{\tau\to\infty}\frac{1}{\tau}\int_{-\tau}^{\tau}|\widetilde{I}(\nu)|^2 d\nu \\ &\equiv \int_{-\infty}^{\infty} S_I(\nu) d\nu\,, \end{aligned}\tag{34}$$

where $S_I(\nu)$ is the spectral density of the photo-current $I(t)$. Therefore, for the current $I_n(t)$ caused by the thermal noise

$$\int_{-\infty}^{\infty} S_{I_n}(\nu) d\nu = \frac{k_B}{L} T_K\,. \tag{35}$$

Common responses from light detectors are voltage outputs. One may want to write the SNR ratio as a function of the voltage

$$SNR = \frac{\langle V\rangle^2}{\langle (V-\langle V\rangle)^2\rangle} = \frac{\langle V\rangle^2}{\langle V^2\rangle - \langle V\rangle^2} = \frac{\langle V\rangle^2}{\langle (\Delta V)^2\rangle}\,. \tag{36}$$

One then need to obtain the average and fluctuation of V to calculate Eq. (36). For example, the current in the circuit shown in Fig. 7-B is given by

$$L\frac{d}{dt}I(t) + RI(t) + \frac{1}{C}\int_{-\infty}^{t} I(t')dt' = V\,. \tag{37}$$

Looking at $e^{i2\pi\nu t}$ $(= e^{i\omega t})$ fluctuations, one obtains the circuit impedance $Z(\omega) = V(\omega)/I(\omega) = R + i\left(\omega L - \frac{1}{\omega C}\right)$. For a circuit where the energy is mostly stored in the inductance field, one may neglect the stored charge given by $\int_{-\infty}^{t} I(t')dt' \to 0$, that would otherwise reside in the capacitor. This gives a LR circuit whose response extends to very high frequencies. This gives $Z = Z(\omega) = R + i\omega L$. From $Z(\omega)I(\omega) = V(\omega)$ one may infer the relationship between S_I and the corresponding voltage spectrum S_V:

$$S_V = |Z|^2 S_I = |R + i\omega L|^2 S_I\,. \tag{38}$$

Therefore, for the noise

$$\int_{-\infty}^{\infty} S_{I_n}(\nu) d\nu = \frac{k_B}{L} T_K = \int_{-\infty}^{\infty} \frac{S_{V_n}(\nu)}{|Z|^2} d\nu\,. \tag{39}$$

Considering that the frequency response associated with $S_{V_n}(\nu)$ is uniform up to very high frequencies, one may write $S_{V_n}(\nu) \to S_{V_n}(0)$. This gives

$$\begin{aligned}\int_{-\infty}^{\infty} \frac{S_{V_n}(\nu)}{|Z|^2} d\nu &\simeq S_{V_n}(0)\int_{-\infty}^{\infty}\frac{1}{R^2 + (2\pi\nu L)^2} d\nu \\ &= \frac{S_{V_n}(0)}{2}\frac{1}{LR} = \frac{k_B}{L} T_K\,, \end{aligned}\tag{40}$$

and therefore

$$S_{V_n}(\nu) = S_{V_n}(0) = 2k_B T_K R\,, \tag{41}$$

$$S_{I_n}(\nu) = \frac{S_{V_n}(\nu)}{R^2} = \frac{2k_B T_K}{R}\,. \tag{42}$$

The detector output usually goes to a bandwidth limited preamplification stage, that will set the overall bandwidth limit in frequency $\Delta\nu_B$. Similarly to Eq. (34), the connection between the average $\langle V(t)^2\rangle$ and $S_{V_n}(\nu)$ is

$$\begin{aligned}\langle V(t)^2\rangle &= \int_{-\Delta\nu_B}^{\Delta\nu_B} S_V(\nu) d\nu = 2k_B T_K R\int_{-\Delta\nu_B}^{\Delta\nu_B} d\nu \\ &= 4k_B T_K R\Delta\nu_B\,. \end{aligned}\tag{43}$$

This treatment exemplifies the use of fluctuations and laws of energy equipartition to derive connections between frequency spectra and thermodynamic quantities. Similar treatment can be applied to distinct circuits. Eq. (43) was investigated by J. B. Johnson in [17].

For an electric current originated from laser excitation the instants t_i will be randomly (Poissonian) distributed, and for a given light power P the average value of the excited photoelectron current is (see Eq. 6)

$$\langle I_{pe}(t)\rangle = \sigma P = \eta_{pe}\frac{e}{\hbar\omega}P\,. \tag{44}$$

The instantaneous value $I_G(t)$ of the amplified $I_{pe}(t)$ by a circuitry with a time constant t_c and gain G is

$$I_G(t) = \int_0^{\infty} \frac{e^{-t'/t_c}}{t_c} G I_{pe}(t-t') dt' \tag{45}$$

The current $I_G(t)$ will show asymmetric amplified spikes instead of the point-like Dirac's deltas and with a decaying time given by t_c.

H. Signal to noise ratio

Let us consider that the output current $I(t)$ after an amplifier stage of gain G, and time constant t_c (e.g., $t_c = RC$), is constituted by the sum of the current contributions given by:
1) light of average frequency ω_0 with power $P(t)$ giving the average current $G\eta_{ep} e\langle P(t)\rangle/(\hbar\omega_0)$,
2) electronic thermal excitations (Johnson's noise) and,
3) dark current I_{dk} generated by crystallographic defects within the depletion region of the semiconductor being used as the photo-sensitive material (dark currents in PIN photodiodes could be of order $\sim$ 100pA or less).

In avalanche photodetectors (APD), intermediate energy levels can also be populated by the electronic avalanche. These energies would decay shortly after causing "after pulses" that may modify substantially the photo-electron statistics. For each detecting system used, one should understand and consider the causes of deviations from the direct photo-current caused by the primary photo-excitation.

Collecting the above contributions,

$$\langle I_e(t)\rangle \simeq G\eta_{ep}\, e\frac{\langle P(t)\rangle}{\hbar\omega_0} + \langle I_{e,s}(t)\rangle + \langle I_{dk}\rangle\,. \tag{46}$$

The shot noise current will fluctuate around this mean value.

The distinction between Johnson's noise and the electric shot noise is not always clear. While some forms of shot noise occurs even at a temperature of $0\,K$ (e.g., if originated by light's incidence), Johnson's noise is caused by thermal excitations and

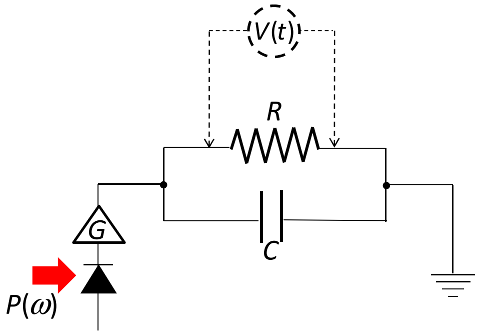


Figure 8. Voltage output. The detector current is amplified with gain G; the voltage is measured at the ends of the parallel RC circuit being probed.

do not exist at $0\,K$. Shot noise in matter is also called ballistic noise, and it is connected with processes where the mean free path of a particle (e.g., electron) is long compared with the atomic positions in the medium. Thermal equilibrium noise appear even with no net current present. Sometimes the physical process is such that a clear distinction between shot and Johnson noise cannot be made. However, for cases where this distinction can be made more apparent, a simplified derivation can be seen in [18].

Eq. (34) shows the connection between $\langle I^2\rangle$ and the current spectral density S_I. For current fluctuations due to the shot noise, this connection holds and for a narrow band S_I, one obtains the spectral density proportional to $\langle I\rangle B$:

$$S_I(\nu_0)B = \langle I^2\rangle = 2e\langle I\rangle B\,. \tag{47}$$

Actually, the total current density *fluctuation* spectrum is given by all contributions, and considering that a gain G also exists, combining Eqs. (47) and (46), one obtains

$$S_I(\nu) = G^2\eta_{ep}\,e^2\frac{P(\nu_0)}{\hbar\omega_0} + \frac{2k_BT_K}{R} + S_{I_{dk}}(\nu)\,. \tag{48}$$

Considering that samplings are taken at a specific frequency such that the t_c cutoff is very low compared to it, using Eq. (38) one may write the mean-square voltage *fluctuation*

$$\langle\Delta V(t)^2\rangle = \int_{-\infty}^{\infty} S_I(\nu)\,|Z(\nu)|^2\,d\nu \simeq S_I\int_{-\infty}^{\infty}|Z(\nu)|^2\,d\nu, \tag{49}$$

where S_I, from Eq. (48), includes the main contribution from the dark noise. In general, the variance of the filtered current *fluctuation* is

$$\langle\Delta I(t)^2\rangle = \int_{-\infty}^{\infty} F(\omega,\tau_c)S_I(\omega)\frac{d\omega}{2\pi}\,, \tag{50}$$

where $F(\omega,\tau_c)$ is the applied linear filter.

For a detector system ending in a parallel RC combination, where the voltage $V(t)$ is probed between the capacitor or resistor ends (see Fig. 8), the impedance Z to be inserted in Eq. (49) is given by

$$\frac{1}{Z} = \frac{1}{Z_R} + \frac{1}{Z_c} = \frac{1}{R} + \frac{1}{\frac{-i}{\omega C}}\,. \tag{51}$$

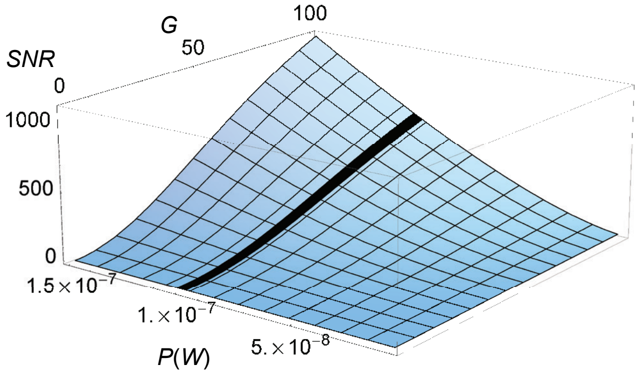


Figure 9. SNR as a function of the optical power P and the gain G. Used parameters are $T_K = 300K$, $\hbar = 1.055\times10^{-34}$Js, $k_B = 1.38\times10^{-23}$J/K, $e = 1.60\times10^{-19}$C, $\eta_{ep} = 0.8$, $R = 50\Omega, C = 20p$F, $\lambda = 1.55\mu$m, $P_{dk} = 1\times10^{-10}$W. The solid line indicates SNR values at $P = 100n$W and for variable gain.

Thus

$$\int_{-\infty}^{\infty}|Z(\nu)|^2d\nu = \frac{1}{2}\frac{R}{C} \to \langle\Delta V(t)^2\rangle \simeq S_I(\nu_0)\frac{1}{2}\frac{R}{C}\,, \tag{52}$$

and therefore

$$\langle\Delta V(t)^2\rangle = \frac{1}{2}\frac{R}{C}\left[G^2\eta_{ep}\,e^2\frac{P(\nu_0)}{\hbar\omega_0} + \frac{2k_BT_K}{R} + S_{I_{dk}}(\nu_0)\right] \tag{53}$$

$$\langle V(t)\rangle = G\eta_{ep}\,e\frac{\langle P(\nu_0)\rangle}{\hbar\omega_0}R\,. \tag{54}$$

$S_{I_{dk}}$ can be written using an equivalent power P_{dk} for the dark noise

$$S_{I_{dk}} = eG\left(\eta_{ep}\,e\frac{P_{dk}}{\hbar\omega}\right)\,. \tag{55}$$

Using the obtained relationships, the SNR ratio with respect to voltage measurements is

$$SNR = \frac{\langle V(t)\rangle^2}{\langle(V(t)-\langle V(t)\rangle)^2\rangle} = \frac{\langle V(t)\rangle^2}{\langle\Delta V(t)^2\rangle} = \frac{\eta_{ep}\,(P(\omega)/(\hbar\omega))2RC}{\left[1 + \frac{2k_BT_K}{RG^2e^2\eta_{ep}(P(\omega)/(\hbar\omega))} + \frac{P_{dk}(\omega)}{GP(\omega)}\right]}\,. \tag{56}$$

Eq. (56) is one of the main results in this section. It incorporates the leading aspects of the detection process and noise from fundamental sources. It can be used as a guidance tool in the optimization process to obtain a good signal to noise ratio, with light effects predominating over thermal sources and others. Fig. 9 shows SNR as a function of the optical power P and the gain G.

As a warning, the idealized voltmeter in Fig. 8 is, in practice, an instrument with particular noise sources. Although it is usually assumed that the voltage probes have a negligible effect on the measurement, their influence may be detected. In particular, for AtoD converters, one should examine the instrument's noise to understand its influence on the obtained data. For example,

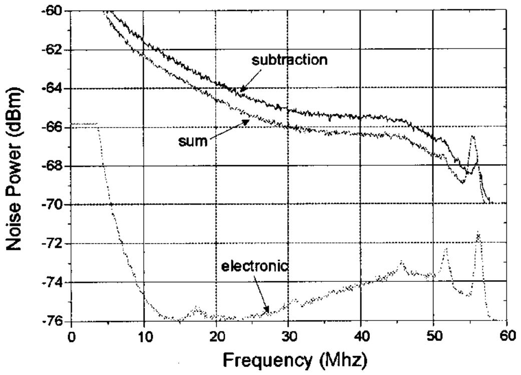


Figure 10. Example of background electronic noise compared with optical shot-noise signals for a diode laser with an external cavity. The lowest line is the electronic level (peaks are resonances in the detecting system) and upper lines are optical signals.

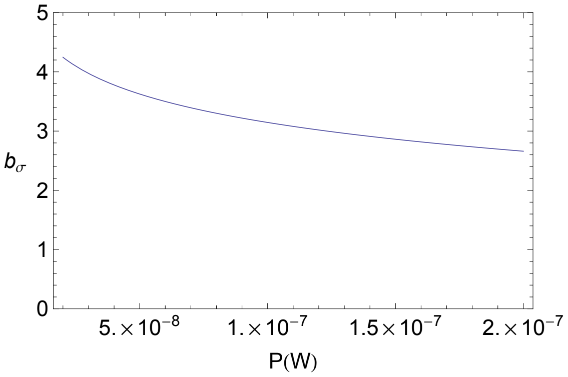


Figure 11. Number of bits available to record the fluctuations around the average optical signal. Laser power $P \sim 100n$W, $\Delta t = 1\ ns \rightarrow 3$ bits.

turning a light source off, one can measure the background electronic noise. Fig. 6 in [19], reproduced in Fig. 10 shows a measurement of the electronic noise for a particular laser, using a spectrum analyzer.

To record fluctuations of the optical field around the average optical signal itself, a couple of conditions have to be obeyed: Firstly, the average intensity reaching the detector in Δt has to excite it. This requires a minimum number of photons (detector dependent) and, for telecommunication detectors, here estimated at ~ 600 photons in Δt. Assuming a detecting system with parameters as given in Fig. 9, an attenuated laser power of $P = 100\,n$W, probed at intervals of 1 ns, produce an average of ~ 780 photons. Secondly, the background noises (noises associated to the detecting system) should give a smaller contribution than the signal corresponding to the fluctuations of the optical signal around its average. The value $\langle n\rangle$ (in 1 sec) is given by the laser power (in MKS units), as $P = \langle n\rangle_{1s}\hbar\omega_0$. Within a sampling time Δt, the average number of photons is $\langle n\rangle_{\Delta t} = \langle n\rangle\Delta t$. Assuming $\Delta t \ll \tau_c$, the standard deviation σ from $\langle n\rangle$, or average fluctuating number of photons, is $\sigma_{\Delta t} = \sqrt{\langle(n-\langle n\rangle)^2\rangle_{\Delta t}} = \sqrt{\langle n\rangle_{\Delta t}}$. With an applied gain $G \sim 100$, Fig. 9 shows a $SNR \sim 600$. Assuming that the total signal fulfills all of 2^b levels in the b-bits recording system, Eq. (3) gives the available bits for the signal fluctuation around its average for an 8-bits recording system (256 levels). Fig. 11 shows that for $P \sim 100\,n$W, there are ~ 3 bits available to record the fluctuating signal (± 8 in 256 levels).

Details related to a specific laser as well as to the ADC system used may heavily influence the final results due to the order of magnitude variations for some parameters; case-by-case have to be studied.

VII. Conclusions

Guidelines for construction of a basic fast multi-purpose PhRG were described as a initial contribution for construction of PhRGs for practical use in a variety of situations. Understanding the principles in these guidelines will also help the development of PhRG miniaturizations; speed gain and reduction in cost are to be expected.

Acknowledgments

This work is inserted in the project "key**BITS**" being implemented under support of RENASIC (Rede Nacional de Excelência em Segurança da Informação e Criptografia)–Ministério da Defesa/Ministério da Ciência e Tecnologia/Financiadora de Estudos e Projetos–BRAZIL.

References

[1] A. Einstein: "...I do not approve of the purely statistical way of thinking on which the new theories are founded...". From a letter to H. A. Lorentz in June 17, 1927, about Quantum Statistics.

[2] http://en.wikipedia.org/wiki/ Pseudorandom_number_generator

[3] 16 MHz Quantis-Quantum Random Number Generator, id Quantique SA (http://www.idquantique.com).

[4] For example: 250k bit/sec, True Random Number Generation IC "RPG100B" (http://www.fdk.com/whatsnew-e/release050930-e.html); VIA PadLock RNG. See also a cryptographic system in R. Mingesz, Z. Gingl, L. B. Kish, "Johnson(-like)-Noise-Kirchhoff–loop based secure classical communicator characteristics, for ranges of two to two thousand kilometers, via model-line", Phys. Letters A **372**, 978-984 (2008).

[5] I. Reidler, Y. Aviad, M. Rosenbluh, I. Kanter, "Ultrahigh-Speed Random Number Generation Based on a Chaotic Semiconductor Laser", Phys. Rev. Letters **103**, 024102/1-5 (2009).

[6] G.A. Barbosa, United States Patent #US 7,831,050 B2 (Filed on Dec. 1, 2004; Prior Publication Data Ju. 14, 2005). Fig. 3, descriptions and claims. See also INPI-Brazil, PI0405814-3 (2004).

[7] B. Qi, Y-M Chi, H-K Lo, and L. Qian, "High-speed quantum random number generation by measuring phase noise of a single-mode laser", Optics Letters **35**, 312-314 (2010).

[8] G. A. Barbosa and UFMG Team, *Telecomm. Platform*, See www.renasic.org.br/comsic/bin/view/LAPROJ/WebHome.

[9] A. Kolmogorov, "Logical Basis for Information Theory and Probability Theory", IEEE Transactions on Information Theory **14**, 662664 (1968).

[10] G. Chaitin, *Meta Math!:The Quest for Omega*. New York: Pantheon Books, 2005.

[11] R. H. Hadfield, "Single-photon detectors for optical quantum information applications", Nature Photonics **3**, 696-705 (2009).

[12] Z. L. Yan, A. W. Sharpe, J. F. Dynes, A. R. Dixon, and a. J. Shields, "Multi-gigahertz operation of photon counting InGaAs avalanche photodiodes", Appl. Phys. Letters **96**, 071101/1-3 (2010).

[13] T. Mueller, F. Xia and P. Avouris, "Graphene photodetectors for high-speed optical communications", Nature Photonics **4**, 297 (2010).

[14] R.J. Glauber, "Coherent and Incoherent States of the Radiation field", Physical Review **131**, 2766-2788 (1963). iptions and claims.

[15] L. Mandel and E. Wolf, *Optical Coherence and Quantum Optics* (Cambridge University Press, New York, 1995).

[16] B. Yurke, *Squeezed Light* (Course notes, University of Rochester, 1989).

[17] J. B. Johnson, "Thermal agitation of electricity in conductors", Phys. Rev. **32**, 97-109 (1928).

[18] L. Callegaro, "Unified derivation of Johnson and shot noise expressions", Am. J. Phys. **74**, 438-440 (2006).

[19] C. L. Garrido-Alzar, S. M. de Paula, M. Martinelli, R. J. Horowicz, A. Z. Khoury, G. A. Barbosa, "Transverse Fourier analysis of squeezed light in diode lasers", J. Opt. Soc. Am. B **18**, 1189-1195 (2001).

G. A. Barbosa was born in Brazil, in 1943. PhD (Physics)/University of Southern California, 1974. Areas of work: Quantum Optics and Condensed Matter (Theory and Experiment). Full Professor, Universidade Federal de Minas Gerais/MG/Brazil (up to 1995); Northwestern University (2000/2012) and CEO, QuantaSec Consultoria e Projetos em Criptografia Física Ltda /Brazil. Member of American Physical Society and Sociedade Brasileira de Física. Established several research laboratories including the first Quantum Optics laboratory in Brazil, where a quantum image was first demonstrated worldwide. Has patents in cryptography, including the very first Brazilian patent in quantum cryptography: INPI-PI9806314. This system is similar to a BB84 protocol but works in time coincidences with two independently polarized photons to eliminate background noises. Another patent covers the alpha eta *encryption* system (inventors: H. P. Yuen, P. Kumar, and G. A. Barbosa), developed under support from Defense Advanced Research Projects Agency (DARPA)- Department of Defense. A patent being implemented through support from Renasic (Rede Nacional de Segurança da Informação e Criptografia) is entitled "Fast multi-photon key distribution scheme secured by quantum noise" (US-2005-0152540-A1 and Brazil-INPI 002872/G. A. Barbosa).

QC-MDPC McEliece: an Optimized Implementation of a New McEliece Variant

H. O. Martins and A. C. A. Nascimento

***Abstract*— This paper presents the implementation of an optimized version of a McEliece variant.The McEliece cryptosystem is an example of code-based cryptography which is an alternative to the most popular and commercial cryptosystems nowadays as it is believed to be immune to quantum computing. It has simple and fast algorithms, but its drawback is the size of the keys it has to deal with. By substituting the Goppa codes of the McEliece original proposalby LDPC and MDPC codes it's possible to achieve much smaller keys. And by applying programming technicssuch as parallelization of operations and also utilizing efficient decoders of LDPC codes it's possible to achieve really good results and optimal performances of the code-based cryptosystem showing that it really has to be considered as a strong substitute to RSA and DSA as quantum computers emerge to easily compute discrete logarithms and factor large integers.**

***Keywords*— Post-quantum cryptography, Code-based cryptography, Coding-theory, Efficient decoding.**

I. INTRODUÇÃO

A SEGURANÇA dos criptossistemas de chave pública utilizados na atualidade tais como o RSA e o DSA está fundamentada no enorme esforço computacional necessário para a fatoração de números inteiros grandes ou para o cálculo de logaritmos discretos. Porém, estes criptossistemas podem ser quebrados em tempo polinomial por intermédio de computadores quânticos executando o algoritmo de Shor ou de Grover [1].

Os computadores quânticos poderão vir a representar o fim do RSA e do DSA, mas não da criptografia assimétrica, pois existem criptossistemas como os baseados em códigos corretores de erro que continuam imunes tanto à computação quântica quanto à clássica ou tradicional.

Se correspondem a uma possível alternativa para a ameaça representada pelos computadores quânticos, o grande problema dos criptossistemas baseados em códigos é a sua eficiência, basicamente no que diz respeito ao tamanho das suas chaves. O criptossistema de chave pública de McEliece com códigos Goppa proposto a mais de 30 anos [2], configurado para padrões de segurança atuais de 128 bits, apresenta chaves com tamanho de mais de um milhão e meio de bits, enquanto o RSA, para o mesmo nível de segurança, trabalha com chaves de alguns poucos milhares de bits.

Pesquisas atuais estão sendo conduzidas na busca de códigos alternativos que substituam os códigos de Goppa e possibilitem aos criptossistemas baseados em códigos tamanhos de chaves significativamente menores, mas que preservem as propriedades de segurança destes criptossistemas. Recentemente, Misoczkiet al. [3] apresentaram um artigo no qual propõem a utilização de códigos quase-cíclicos com verificação de paridade de densidade moderada (códigos QC-MDPC) em substituição aos códigos de Goppa. Como resultado, alcançaram os 128 bits de nível de segurança com uma chave de tamanho inferior a 1000 bits.

II. CONTRIBUIÇÃO

Este trabalho consiste na implementação de uma versão otimizada de um criptossistema de McEliece baseado em códigos QC-MDPC. Esta implementação está fundamentada na proposta de Misoczkiet al [3].

Como a decodificação é em regra a etapa mais custosa em criptossistemas baseados em códigos, o trabalho procurou centralizar todos os seus esforços na construção de uma versão otimizada do decodificador.

Na construção do decodificador o presente trabalho agregou à implementação proposta por Misoczkiet al. a contribuição de von Maurich eGüneysu[4], que corresponde a uma variante eficiente do melhor decodificador para códigos QC-MDPC em dispositivos embarcados.

O trabalho também procurou fazer uso de algumas técnicas computacionais que economizam repetições na execução de tarefas exaustivas tais como multiplicações de matrizes, bem como recorreu a extensões do conjunto de instruções do processador, as quais possibilitam o acesso a registradores específicos do hardware de modo a obter ganho de performance no processamento de operações lógicas e matemáticas através da paralelização.

III. PRELIMINARES

O primeiro criptossistema baseado na teoria de códigos foi proposto em 1978 por R. J. McEliece [2]. Em sua descrição original, a chave secreta do criptossistema de McElice corresponde a um código de Goppa.

H. O. Martins, Universidade de Brasília (UnB), Brasília, Brasil, homer@unb.br

A. C. A. Nascimento, Universidade de Brasília (UnB), Brasília, Brasil, andclay@ene.unb.br

Invited Paper

Em [1] encontra-se bem detalhado o funcionamento do criptossistema de McEliece com códigos Goppa. Em [3] existe uma descrição bastante completa de códigos QC-MDPC, bem como da codificação/decodificação McEliece baseada em códigos QC-MDPC. A seguir segue um resumo das principais ideias acerca dos códigos QC-MDPC, assim como do criptossistemaMcEliece.

A. Códigos QC-MDPC

Um **código binário (*n*,*r*)-linear** C de comprimento *n*, dimensão $n - r$ e codimensãor corresponde a um subespaço vetorial (*n* – *r*) dimensional de$\mathbb{F}_n^2$. Ele é gerado pelas linhas de uma matriz $G \in \mathbb{F}_2^{(n-r)\times n}$chamada matriz geradora de C. E também é o núcleo de uma matriz $H \in \mathbb{F}_2^{r\times n}$conhecida como matriz de verificação de paridade de C.

Um **código (*n*,*r*)-linear** é considerado **quase-cíclico** (**QC**) se existir um inteiro $n_0$tal que qualquer deslocamento circular de uma palavra-chave por n_0 posições dá como resultado uma outra palavra-chave.

Um **código (*n*,*r*,*w*)-MDPC** é um código linear de comprimento *n*, codimensãor e que admite uma matriz de verificação de paridade com peso de *Hamming* constante de suas linhas igual a *w*.

Para a construção de um **código (*n*,*r*,*w*)-QC-MDPC**escolhe-se uma palavra aleatória de comprimento$n = n_0.p$e peso *w* que corresponde à primeira linha da matriz H. As outras *r* - 1 linhas são obtidas por intermédio de *r* - 1 deslocamentos quase-cíclicos, de tal maneira que cada bloco $H_i$tem peso de linhaw_i tal que $w = \sum_{i=0}^{n_0-1} w_i$. A matriz geradora G pode ser calculada a partir dos blocos $H_i$como $G = [I \mid Q]$, onde I é a matriz identidade e $Q = \begin{bmatrix} (H_{n_0-1}^{-1}.H_0)^T \\ (H_{n_0-1}^{-1}.H_1)^T \\ \vdots \\ (H_{n_0-1}^{-1}.H_{n_0-2})^T \end{bmatrix}$.

B. Codificação/Decodificação QC-MDPC McEliece

As operações de geração de chaves, codificação e decodificação McEliece baseadas num código (*n*,*r*,*w*)-QC-MPDC com capacidade para correção de *t* erros são definidas como:

Geração das chaves: produza um vetor aleatório binário *h* de peso *w*. Ele corresponde à primeira linha da matriz H e as demais *r* - 1 linhas são obtidas através de deslocamentos quase-cíclicos de *h*. Obtenha a matriz G correspondente a H na forma reduzida escalonada. A chave pública é G e a chave privada é H.

Codificação: para codificar uma mensagem *m* produza um vetor aleatório binário *e* de peso menor ou igual a *t*. A mensagem codificada *x* é dada por $x \leftarrow mG + e$.

Decodificação: para decodificar *x* em *m*, calcule

$$mG \leftarrow \Psi_H(mG + e),$$

ondeΨ_Hé um decodificador para o código QC-MDPC com conhecimento da matriz H. A mensagem *m* é recuperada a partir das primeiras (*n* - *r*) posições de *m*G.

C. Decodificação Eficiente

A decodificação é a etapa que mais consome tempo na criptografia baseada em códigos. Portanto, a opção pelo algoritmo decodificador mais apropriado é crucial para a obtenção de uma performance desejável na tarefa de decodificação. Para decodificar códigos MDPC existem duas categorias de algoritmos: os mais simples, como aqueles que usam a técnica de inversão de bits (*bit-flipping*) e têm menor capacidade de correção de erros, e aqueles mais elaborados e que alcançam uma melhor capacidade corretora, como por exemplo o algoritmo soma-produto. Misoczkiet al [3] sugerem a utilização dos algoritmos de inversão de bits, tendo em vista que estes têm baixa complexidade, são iterativos e rápidos.

Em termos gerais, todos os algoritmos que usam a técnica de inversão de bits seguem a mesma ideia. Primeiramente, a cada iteração eles calculam a síndrome da mensagem a ser decodificada. A seguir são computados os números de equações de verificação de paridade não satisfeitas associadas a cada bit da mensagem. Cada bit associado a mais que *b* equações não satisfeitas é invertido. Este processamento é repetido até que a síndrome seja completamente zerada ou até que seja alcançado um número limite de iterações, quando considera-se que o processo de decodificação falhou. A grande diferença entre os diversos decodificadores é na maneira como o limite *b* é calculado, o que pode variar desde o maior número de equações não satisfeitas até o pré-cálculo de limites baseados em parâmetros do código utilizado.

O Algoritmo1 é apresentado em [3] e ilustra o parágrafo acima.

Algoritmo 1 variante do algoritmo de *bit-flipping* para decodificação de códigos QC-MDPC

```
Entrada: $y = mG + e$, com peso $w$ e dimensão $n$
Saída: $c$ tal que $Hc^T = 0$, ou FALHA.
whileδ∈ N > 0
    $s \leftarrow Hc^T$
    fori = 1 to dim($s$) do
      if$s$[i] = 1 then
          for j = 1 to$w$do
              contador[j] ← #upc para cada bit
   //#upc = número de equações de verificação de paridade
   //nãosatisfeitas
          end for
      end if
    end for
    Maxupc ← o maior #upc
    fori = 1 to$n$do
      ifcontador[i] ≥ (Maxupc − δ) then
          Flip$c_i$
      end if
    end for
    if$Hc^T = 0$ then
      return$c$
    endif
    δ ← δ – 1 // No caso de falha na decodificação
endwhile
return FALHA
```

Recentemente vários algoritmos de decodificadores foram analisados por Heyseet al.[5] quanto a sua adequação e performance ao trabalhar com os códigos QC-MDPC em dispositivos embarcados. O Algoritmo 2abaixo superou todos os outros tanto em tempo de processamento bem como na taxa de falha de decodificação.

Este decodificador pré-calcula seus limiares b_i baseado nos parâmetros do código como proposto por Gallager [6] e, ao contrário de outras soluções que recalculam a síndrome após cada iteração, apenas faz a atualização da mesma enquanto processa a decodificação da mensagem cifrada.

Algoritmo 2 varianteeficiente de algoritmo de *bit-flipping* para decodificação de códigos QC-MDPC

```
Entrada: $y = m\mathrm{G} + e$, com peso $w$ e dimensão $n$
Saída: $c$ tal que $\mathrm{H}c^{\mathrm{T}} = 0$, ou FALHA.
while$\delta \in \mathbb{N} > 0$
    $s \leftarrow \mathrm{H}c^{\mathrm{T}}$
    fori = 1 to dim($s$) do
      if$s$[i] = 1 then
          for j = 1 to$w$do
              contador[j] ← #upc para cada bit
  //#upc = número de equações de verificação de paridade
  //nãosatisfeitas
          end for
      end if
    end for
    Maxupc ← o maior #upc
    fori = 1 to$n$do
      ifcontador[i] ≥ $b_i$then
          Flip$c_i$
          $s$← $s$XOR$h_j$
      end if
    end for
    if$s$ = 0 then
      return $c$
    endif
    $\delta \leftarrow \delta - 1$ //No caso de falha na decodificação
endwhile
returnFALHA
```

IV. IMPLEMENTAÇÃO

A proposta do trabalho é a construção de uma implementação eficiente do criptossistemaMcEliece baseado em códigos QC-MDPC.

O desenvolvimento do criptossistema foi feito em linguagem C na plataforma Intel e desconsiderou qualquer limitação de memória ou capacidade de processamento e armazenamento do hardware.

Para sua realização, teve como ponto de partida a proposta [3] com a construção do codificador e do decodificador lá descritos, adotando os seguintes parâmetros apresentados no artigo (em bits), para um nível de segurança de 80 bits: $n_0 = 2$, $n = 9600$, $r = 4800$, $w = 90$, $t = 84$, tamanho da chave = 4800.

Portanto, uma mensagem de 4800 bits (r) é codificada em uma mensagem cifrada de 9600 (n) bits à qual 84 bits de erro (t) são adicionados. A matriz de paridade H possui linhas com peso constante 90 (w) e consiste de 2 blocos H_0 e H_1 (n_0).

Após a construção inicial do criptossistema e a tomada de tempo de execução, o esforço todo passou a ser direcionado para a otimização e melhora do desempenho da rotina de decodificação pois, como mencionado, esta etapa é a maior consumidora de tempo de processamento.

É importante ressaltar que tanto na codificação quanto na decodificação a aleatoriedade para a geração dos vetores binários (vetor h e vetor$e$) foi simulada, uma vez que a utilização de um gerador de números aleatórios verdadeiro está fora do escopo deste trabalho.

A seguir encontra-se detalhado o trabalho de implementação:

Geração das chaves: como não havia restrição de projeto, os vetores e matrizes binários, mesmo sendo esparsos, ficaram todos armazenados integralmente em memória durante a execução do programa, o que possibilitou ganho em algumas operações tais como as atualizações de contadores na decodificação, por exemplo. Para a manipulação dessas estruturas de dados foi utilizada a biblioteca numérica de domínio público para representação de vetores e matrizes *Meschach* [7], a qual provê os tipos básicos de dados matriz e vetor bem como algumas operações primárias tais como a cópia de blocos de dados entre ambos. A biblioteca foi devidamente customizada para que os tipos de dados armazenados pelas matrizes e vetores binários fossem representados como números inteiros de 32 bits sem sinal, os quais foram substituídos por números inteiros de 64 bits sem sinal numa segunda versão.

Durante a tarefa de geração das chaves do criptossistema existe a necessidade de inversão de matrizes binárias. Para a codificação desta tarefa foi utilizado o trabalho de Jasinskiet al. [8], o qual traz um algoritmo eficiente que implementa uma versão melhorada do método de Gauss-Jordan para a inversão de matrizes binárias.

Codificador: para a construção do codificador, com relação a aspectos de desempenho, o maior cuidado tomado foi com a representação de conjuntos de bits por números inteiros sem sinal de 32 bits inicialmente e 64 bits numa segunda etapa. Desta maneira os bits puderam ser processados em blocos, o que garante melhora no desempenho de tarefas que necessitem ser aplicadas em linhas de matrizes ou vetores, tais como a multiplicação de matrizes por matrizes ou de vetores por matrizes, nas quais são executadas operações lógicas (XOR's e AND's) nas linhas binárias.

Decodificador: assim como na geração de chaves e na codificação, as matrizes e vetores foram definidos como elementos de tipos de dados inteiros sem sinal de 32 bits e 64 bits numa segunda etapa. Na decodificação procurou-se fazer uso ao máximo de recursos e técnicas que permitissem a economia no número de operações executadas pelo processador e a consequente melhora no desempenho do processamento.

Como mostrado, numa tarefa de decodificação existe o cálculo inicial da síndrome, que corresponde à multiplicação da matriz de paridade H pelo vetor transposto que representa a mensagem codificada c. Medindo-se o tempo de processamento de todos os procedimentos que compõem a rotina de decodificação, percebe-se que esta multiplicação corresponde à etapa de maior custo no algoritmo. Um truque bastante simples e eficiente e que reduz em muito o tempo gasto para se decodificar uma mensagem é, na computação da síndrome, calcular o produto do vetor transposto pela matriz transposta. E na realização deste cálculo percorre-se cada elemento individual binário do vetor e as posições dos bits não nulos selecionam as linhas da matriz transposta às quais deve ser aplicado o XOR. O resultado desta operação é a síndrome. Como o vetor mensagem codificada tem cerca de apenas um terço de suas posições diferentes de zero, isto significa que somente aproximadamente um terço dos XOR's das linhas da matriz H precisa efetivamente ser computado para se chegar ao valor da síndrome. Este truque foi implementado e mostrou-se bastante útil devido à configuração da matriz e do vetor que deviam ser multiplicados.

No algoritmo da decodificação, após o cálculo da síndrome há um passo no qual é executado o cálculo dos contadores que registram a quantidade de equações de checagem de paridade não satisfeitas associadas a cada bit da mensagem. Esta contagem é repetida a cada atualização da síndrome e, portanto, foi testada alternativa na qual os contadores eram apenas atualizados em vez de zerados e recalculados. Nos testes armazenou-se a síndrome anterior e comparou-se a mesma com a síndrome atual e apenas os contadores relacionados às posições dos bits que mudaram entre ambas as versões da síndrome eram incrementados ou decrementados, dependendo da mudança do bit de zero para um ou de um para zero. A medição de tempo dos testes descartou esta solução alternativa ao mostrar que a recontagem de todos os contadores da maneira como está proposto no algoritmo é a estratégia mais rápida.

Uma outra otimização que representou impacto positivo no tempo de processamento foi a utilização de instruções vetoriais intrínsecas nas operações de XOR que existem tanto na multiplicação matriz por vetor quanto no processamento da síndrome durante a decodificação.

Instruções intrínsecas são aquelas que o compilador mapeia em uma sequência de uma ou mais instruções em linguagem de máquina com o grande benefício de possibilitar o acesso a novos registradores específicos do hardware assim como a novos tipos de dados que não estão disponíveis através dos métodos padrão das linguagens de programação.

As funções intrínsecas são inerentemente mais rápidas que as convencionais providas pela linguagem, O código para uma função intrínseca geralmente é inserido em linha, evitando a sobrecarga de uma chamada de função e permitindo instruções de máquina altamente eficientes, pois o compilador tem o conhecimento prévio e específico de como as instruções intrínsecas se comportam.

Mediante o uso de instruções vetoriais intrínsecas – instruções SIMD (*single instructionmultiple data*) - é possível a paralelização do código com a realização de operações lógicas ou matemáticas em múltiplos pares de operandos simultaneamente.

Para o processamento das matrizes e vetores foram utilizadas instruções vetoriais intrínsecas Intel do conjunto de extensões AVX e AVX2. Com o conjunto de instruções AVX é possível o acesso a registradores de 256 bits da família de processadores Intel com a microarquitetura *Sandy Bridge*. O conjunto de instruções AVX2 torna disponíveis registradores de 512 bits da família de processadores com microarquitetura *Haswell*. O ganho de tempo resultante do uso das instruções vetoriais nestes casos deve-se à paralelização: como afirmado, após serem carregados conjuntos de elementos de 32 ou 64 bits múltiplos inteiros do tamanho dos registradores, é executado um XOR simultâneo em vários pares de operandos.

Uma última otimização diz respeito à redução no número de operações de cálculo de síndromes, que correspondem a multiplicações de vetores mensagem por matrizes de paridade, o que, como já foi dito, é o maior gargalo de desempenho no algoritmo de decodificação. Comparando os Algoritmos 1 e 2 apresentados observa-se que em 2 a síndrome é calculada apenas uma vez e a partir de então ela passa a ser atualizada apenas através de XOR's. As duas propostas foram implementadas e observou-se ganho significativo no tempo de processamento da rotina de decodificação que faz economia na quantidade de multiplicações realizadas. Há que ser ressaltado que o tempo de execução é bastante melhorado com a redução do número de multiplicações necessariamente em conjunto com a utilização dos limiares $b_i$pré-calculados em cima dos parâmetros do código. Estes limiares diminuem significativamente a quantidade de iterações executadas pela rotina de decodificação.

V. RESULTADOS

Os resultados apresentados abaixo foram obtidos a partir de execuções das várias versões implementadas em linguagem C do criptossistema de McEliece baseado em códigos QC-MDPC. Para tais foi utilizada uma estação de trabalho com CPU Intel Core I7-4770 trabalhando na frequência de 3,40 GHz e rodando sistema operacional Linux Ubuntu12.04 LTS.

Em cada versão implementada do criptossistema, os tempos medidos foram calculados a partir da média dos tempos de execução de mil decodificações de uma mesma mensagem cifrada. Procedeu-se da mesma maneira para a obtenção do número médio de iterações para cada decodificação.

Inicialmente foi construída uma versão do criptossistema que corresponde literalmente ao que está proposto em [3]. Não foi utilizada nenhuma estratégia de otimização do código fonte, apenas alterou-se o tipo de dados dos elementos armazenados por vetores e matrizes para inteiros de 32 bits sem sinal e 64 bits sem sinal e mediu-se os tempos respectivos, apresentados na Tabela I. A ideia é estabelecer um marco inicial comparativo e indicativo da evolução dos resultados do trabalho. No artigo os autores mencionam que o tempo de decodificação por eles alcançado é de cerca de 3ms.

TABELA I. VERSÃO ORIGINAL DO CRIPTOSSISTEMA PROPOSTO EM [3].

Tipo de dados (vetores/matrizes)	Tempo de decodificação (ms)
Inteiros de 32 bits sem sinal	8.230
Inteiros de 64 bits sem sinal	6.740

A seguir o trabalho concentrou-se na construção e teste de versões alternativas otimizadas. O passo inicial da decodificação corresponde ao cálculo da síndrome, que nada mais é que o resultado do produto da mensagem codificada pela matriz de paridade. Analisando-se a performance da rotina de decodificação, observa-se que esta multiplicação é uma etapa de alto custo de processamento. Portanto, uma primeira otimização significou a aplicação de um truque na rotina de multiplicação de matriz por vetor, a qual passou a selecionar para a operação de XOR apenas as linhas da matriz transposta associada cujas posições correspondem a valores não nulos do vetor. Como o vetor possui apenas um terço de seus elementos diferentes de zero, esta estratégia economizou dois terços de operações XOR desnecessários para a obtenção da síndrome.

Além da redução do número de XOR's para o cálculo da síndrome, o uso das instruções vetoriais intrínsecas na rotina de multiplicação possibilitou o paralelismo com a execução de XOR's simultâneos em múltiplos pares de operandos carregados nos registradores específicos de 256 bits (AVX) e 512 bits (AVX2).

Todas estas ações refletem-se nos tempos mostrados na Tabela II, onde a decodificação passou a acontecer em menos de 3ms.

TABELA II. VERSÃO OTIMIZADA 1 DO CRIPTOSSISTEMA QC-MPDC MCELIECE.

Instruções vetoriais	Tempo de decodificação (ms)
AVX	2.595
AVX2	2.362

Conforme afirmado, a multiplicação para o cálculo da síndrome é a etapa que mais demanda esforço computacional na tarefa de decodificação de uma mensagem cifrada. No Algoritmo 1 proposto em [3] esta multiplicação é executada em cada uma das iterações que fazem parte das tentativas de decifração da mensagem.

O Algoritmo 2 apresentado em [4] elimina esses cálculos repetidos da síndrome, assim como reduz a quantidade de iterações necessárias para a decodificação da mensagem. Uma versão alternativa do decodificador foi implementada contemplando o Algoritmo 2 e os tempos medidos, como esperado, espelharam a melhora significativa da decodificação como resultado da eliminação das multiplicações sucessivas do vetor mensagem cifrada pela matriz de paridade e a substituição desta tarefa pela simples atualização da síndrome calculada no passo inicial da decodificação, assim como a redução do número de iterações do algoritmo com a substituição dos seus limiares por valores pré-calculados baseados em parâmetros do código.

A Tabela III mostra que esta ação de substituição do algoritmo de decodificação original contemplou o objetivo do trabalho proposto com a produção de uma implementação eficiente do criptossistemaMcEliece baseado em códigos QC-MDPC.

TABELA III. COMPARAÇÃO ENTRE VERSÕES ALTERNATIVAS DO CRIPTOSSISTEMA QC-MPDC MCELIECE

Instruções vetoriais	Tempo de decodificação (ms)	Algoritmo	Iterações
AVX	2.595	1	5
AVX	0.945	2	2
AVX2	2.362	1	5
AVX2	1.173	2	2

VI. CONCLUSÃO

Neste trabalho foi construída uma versão otimizada eficiente de um criptossistema de McEliece baseado em códigos QC-MDPC. Este criptossistema acena como um possível substituto dos criptossistemas assimétricos populares imune ao computador quântico. Na sua concepção original o criptossistema de McEliece baseado em códigos de Goppa apresentava como inconveniente o tamanho grande das suas chaves. A proposta [3] apresenta uma versão na qual níveis de segurança atuais são alcançados com chaves de dimensões compatíveis com as dos criptossistemas de chave pública atualmente difundidos. A implementação de uma versão com tempos de processamento eficientes é mais uma evidência que reforça estes criptossistemas como candidatos a alternativas viáveis caso a computação quântica se estabeleça além dos modelos teóricos e dos laboratórios.

REFERÊNCIAS

[1] D. J. Bernstein, J. Buchmann, and E. Dahmen, editors. *Post-Quantum Cryptography*. Springer-Verlag, 2009.

[2] R. J. McEliece, *"A Public-Key Cryptosystem Based On Algebraic Coding Theory"*, Deep Space Network Progress Report, vol. 44, pp. 114–116, Jan. 1978.

[3] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. L. M. Barreto, *"MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes"*, IEEE International Symposium on Information Theory, vol. 2013, 2013.

[4] I. von Maurich, T. Güneysu,*"Lightweight Code-based Cryptography: QC-MDPC McEliece Encryption on Reconfigurable Devices"*, Procedings of the IEE Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014, pp. 1-6.

[5] S. Heyse, I. von Maurich, and T. Güneysu, *"Smaller Keys for Code-Based Cryptography: QC-MDPC McEliece Implementations on Embedded Devices"*, in CHES, ser. Lecture Notes in Computer Science, G. Bertoni and J.-S. Coron, Eds., vol. 8086. Springer, 2013, pp. 273– 292.

[6] R. Gallager, *"Low-density Parity-check Codes"*, Information Theory, IRE Transactions on, vol. 8, no. 1, pp. 21–28, 1962.

[7] Meschach: Matrix computations in C. Disponível em: <http://homepage.math.uiowa.edu/~dstewart/meschach/html_manual/manual.html> Acesso em 20 de maio de 2014.

[8] R. P. Jasinski, V. A. Pedroni, A. Gortan, W. Godoy Jr, *"An Improved GF(2) Matrix Inverter with Linear Time Complexity"*, Procedings of the IEE International Conference on Reconfigurable Computing and FPGAs (ReConFig), 2010, pp. 322-327.

[9] Intel IntrinsicsGuide. Disponível em:<htps:t//software.intel.com/sites/landingpage/IntrinsicsGuide/> Acesso em 20 de maio de 2014.

[10] W. C. Huffman and V. Pless, *"Fundamentals of Error-Correcting Codes"*, 2010.

Homero de Oliveira Martins é graduado em Engenharia Elétrica pela Universidade de Brasília (UnB), Brasília, Brasil, em 1997. Recebeu também o título de Bacharel em Ciência da Computação pela Universidade de Brasília (UnB), Brasília, Brasil, em 1993. Atualmente é aluno de Mestrado em Engenharia Elétrica, na área de redes de comunicação, da Universidade de Brasília (UnB). Suas principais áreas de pesquisas são: Criptografia e Segurança da Informação. Trabalha no Centro de Informática da Câmara dos Deputados desde 1999 na Coordenação de Engenharia de Sistemas.

Anderson A. C. Nascimento é atualmente professor do departamento de Engenharia Elétrica da Universidade de Brasília (UnB), Brasília, Brasil. Recebeu o título de Doutor em *Information and Communication Engineering* da Universityof Tokyo (U.T.), Japão em 2004; Mestre em *Information and Communication Engineering* da Universityof Tokyo (U.T.), Japão em 2001; e graduado em Engenharia Elétrica pela Universidade de Brasília (UnB), Brasília, Brasil, em 1998. Suas principais áreas de pesquisas são: computação segura de duas partes e multipartes; teoria quântica da informação; segurança demonstrável e criptografia baseada em códigos.

Securing Web Applications: Techniques and Challenges

M. Vieira

Abstract— **Software security is nowadays a hot research topic, particularly in the Web domain. In fact, due to the impressive growth of the Internet and of Web applications, software security has become one vital concern in any information infrastructure. This paper discusses key techniques for security testing and assessment, providing the basis for understanding existing research challenges on developing and deploying secure Web applications.**

Keywords— **Security, Web applications, Vulnerabilities, Benchmarking, Secure Processes.**

I. INTRODUCTION

THE GOAL of security is to protect systems and data from intrusion. The risk of intrusion is related to the system vulnerabilities and the potential security attacks. The **system vulnerabilities** are an internal factor related to the set of security mechanisms available (or not available) in the system, the correct configuration of those mechanisms, and the hidden flaws on the system implementation. Many types of vulnerabilities are known and also taxonomies to classify them [1]. Vulnerability prevention consists on guarantying that the software has the minimum vulnerabilities possible (e.g. using security testing). On the other hand, vulnerability removal is the process of mitigating the vulnerabilities found in the system (e.g. by applying new security patches released by software vendors).

Security attacks are an external factor that mainly depends on the intentionality and capability of humans to maliciously break into the system tacking advantage of vulnerabilities. In fact, the success of a security attack depends on the vulnerabilities of the system and attacks are harmless in a system without vulnerabilities. On the other hand, vulnerabilities are harmless if the system is not subject of security attacks. The prevention against security attacks includes all the measures needed to minimize or eliminate the potential attacks against the system (by reducing the potential attack surface). Attack removal is related to the adoption of measures to stop attacks that have occurred before (e.g. using intrusion detection).

Secure Software behaves correctly in the presence of a malicious utilization (attack), even though software failures may also happen when the software is used correctly [2]. Thus, many times software development and testing concerns only with what happens when software fails and not with the intentions. This is where the difference between software safety and software security lies: in the presence of an intelligent adversary with the intention of damaging the system.

In the last two decades, the World Wide Web radically changed the way people communicate and do business. The problem is that, as the importance of the assets stored and managed by web applications increases, so does the natural interest of malicious minds in exploiting this new streak. In fact, web applications are so widely exposed that any existing security vulnerability will most probably be uncovered and exploited by hackers. Hence, the security of web applications is a major concern and is receiving more and more attention from the research community. However, in spite of this growing awareness of security aspects at web application level, there is an increase in the number of reported attacks that exploit web application vulnerabilities [3], [4].

To prevent vulnerabilities developers must apply best coding practices, perform security reviews, execute penetration testing, use code vulnerability detectors, etc. Still, many times developers focus on the implementation of functionalities and on satisfying the costumer's requirements and disregard security aspects. Also, most developers are not security specialists and the common time-to-market constraints limit an in-depth search for vulnerabilities. Another problem is that, traditional security mechanisms like network firewalls, intrusion detection systems (IDS), and encryption, are not able to mitigate web application attacks because they are performed through ports that are used for regular web traffic [5] and even application layer firewalls can not protect the applications as that requires a deep understanding of the business context [6]. In this scenario, a large effort should be put on improving the state of the art in the security of software systems.

This paper discusses key concepts, techniques and tools for testing and assessing security in the context of Web applications and services. First, we discuss techniques and tools for detecting vulnerabilities, which have the greatest importance to help developers producing more secure code. Second, we introduce the concept of Vulnerability and Attack Injection, whose goal is to provide the means to introduce realistic vulnerabilities in applications code. This is extremely useful in different contexts, including: 1) for training security teams; 2) to evaluate security teams in a controlled environment; and 3) to estimate the total number of vulnerabilities still present in the code. Then we discuss security evaluation from the benchmarking (i.e. comparison) point-of-view, which allows assessing and comparing the security of systems and/or components, supporting informed decisions while designing, developing, and deploying complex software systems. Finally, we put security in the context of the **development lifecycle**, emphasizing the key security aspects that should be kept in mind when developing Web applications.

M. Vieira, Universityof Coimbra (UC), Coimbra, Portugal, mvieira@dei.uc.pt

II. SECURITY TESTING

To identify security issues, developers must focus not only on testing the functionalities of the application but also on searching for dangerous security vulnerabilities that are present in the code and that can be maliciously exploited [2]. In this scenario, automated tools have a very important role on helping the developers to produce less vulnerable code.

Different techniques for the detection of vulnerabilities have been proposed in the past [1], but in practice these techniques can be divided in two main groups: **white-box** analysis, which consists of examining the code of the application without executing it (this can be done in one of two ways: manually during code inspections and reviews or automatically by using automated analysis tools); and **black-box** testing, which refers to the analysis of the program execution from an external point-of-view (in short, it consists of exercising the software and comparing the execution outcome with the expected result). Black-box testing is probably the most used technique for verification and validation of software.

In the context of security, both black-box testing and white-box analysis have limitations that are intrinsic to their characteristics. Black-box testing is based on the effective execution of the code and in practice vulnerability identification is only based on the analysis of the web application output. This way, the effectiveness of the process is always limited by the lack of visibility on the internal behavior of the application. On the other hand, white-box approaches like static analysis are normally based on the examination of the source code. The main problem here is that exhaustive source code analysis may be difficult and cannot find many security flaws due to the complexity of the code and the lack of a dynamic (runtime) view. Of course, black-box testing does not require access to the source code while static analysis does.

The effectiveness of automated vulnerability detection tools is frequently very low, thus using the wrong tool may lead to the deployment of applications with undetected vulnerabilities. The work presented in [7] shows the main findings of a practical study that compares the effectiveness of very well known and largely used penetration testing and static analysis tools in the detection of SQL Injection vulnerabilities in Web Services. Results show that the coverage of static code analysis tools (including FindBugs, Fortify 360, and IntelliJ IDEA) is typically much higher than of penetration testing tools (including HP WebInspect, IBM Rational AppScan, and Acunetix Web Vulnerability). False positives are a problem for both approaches, but have more impact in the case of static analysis. A key observation is that different tools implementing the same approach frequently report different vulnerabilities in the same code.

The challenge is that, although we frequently trust vulnerability detection tools, results highlight their limitations suggesting that it is necessary to improve the state of the art in vulnerability detection, for instance by combining different approaches. Also, it is important to define mechanisms to evaluate and compare different tools in order to select the tools that best fit each development scenario.

III. VULNERABILITY AND ATTACK INJECTION

Fault injection has become an attractive approach to validate specific fault handling mechanisms and to assess the impact of faults in actual systems, allowing the estimation of fault-tolerant system measures such as fault coverage and error latency [1]. In the past decades, research on fault injection has specially targeted the emulation of hardware faults, where a large number of works has shown that it is possible to emulate these faults in a quite realist way (e.g. [8], [9]). More recently the interest on the injection of software faults has increased, giving raise to several works on the emulation of this type of faults (e.g., [10], [11]). In practice, software fault injection deliberately introduces faults into the system in a way that emulates real software faults. A reference technique is G-SWFI (Generic Software Fault Injection Technique [10]), which supports the injection of realistic software faults (i.e. faults most likely present in a software) using educated code mutation. The faults injected are described in a library derived from an extensive field study aimed at identifying the types of bugs that can reasonably be expected to occur frequently in a software system.

The use of fault injection techniques to assess security is a particular case of software fault injection, focused on the software faults that represent security vulnerabilities or may cause the system to fail in avoiding a security problem. Security vulnerabilities are in fact a particular case of software faults, which require adapted injection approaches.

In [12] the vulnerabilities of six web applications were analyzed using field data based on a set of 655 security fixes. Results show that only a small subset of 12 generic software faults is responsible for all the security problems. In fact, there are considerable differences by comparing the distribution of the fault types related to security with studies of common software faults.

Neves et al. proposed a tool (AJECT) focused on discovering vulnerabilities on network servers, specifically on IMAP servers [13]. In their work the fault space is the binomial (attack, vulnerability) creating an intrusion that may cause an error and, possibly, a failure of the target system. To attack the target system they used predefined test classes of attacks and some sort of fuzzing.

A procedure inspired on the fault injection technique (that has been used for decades in the dependability area) targeting security vulnerabilities is proposed in [14]. In this work, the "security vulnerability" plus the "attack" represent the space of the "faults" that can be injected in a web application; and the "intrusion" is the "error". To emulate with accuracy real world web vulnerabilities this work relies on the results obtained in a field study on real security vulnerabilities, which were used to develop a novel Vulnerability Injection tool.

Conceptually, attack injection is based on the injection of realistic vulnerabilities that are automatically attacked, and finally the result of the attack is evaluated. As proposed in [15], a tool able to perform vulnerability and attack injection is a key instrument that can be used in several relevant scenarios, namely: building a realistic attack injector, train security teams, evaluate security teams, and estimate the total number of vulnerabilities still present in the code, among others.

The challenge is that current knowledge on vulnerability and attack models is quite limited, and additional studies are required to better understand how, where and when such faults should be injected (in a way that assures high representativeness). Also, existing work is focused on very specific types of vulnerabilities in the Web domain. Extending such approaches to additional domains is a relevant research challenge.

IV. SECURITY BENCHMARKING

Computer benchmarks are standard tools that allow evaluating and comparing different systems or components according to specific characteristics (e.g. performance, robustness, dependability, etc.) [16]. The work on performance benchmarking has started long ago. Ranging from simple benchmarks that target very specific hardware systems or components to very complex benchmarks focusing complex systems (e.g. database management systems, operating systems), performance benchmarks have contributed to improve successive generations of systems. Research on dependability benchmarking boosted in the beginning of this century [17]. Several works have been done by different groups and following different approaches (e.g. experimental, modeling, fault injection). Finally, work on security benchmarking is a new topic with many open questions.

Several security evaluation methods have been proposed in the past [18]–[21]. The Orange Book [20] and the Common Criteria for Information Technology Security Evaluation [19] define a set of generic rules that allow developers to specify the security attributes of their products and evaluators to verify if products actually meet their claims. Another example is the red team strategy [21], which consists of a group of experts trying to hack its own computer systems to evaluate security.

The work presented in [22] addresses the problem of determining, in a thorough and consistent way, the reliability and accuracy of anomaly detectors. This work addresses some key aspects that must be taken into consideration when benchmarking the performance of anomaly detection in the cyber-domain.

The set of security configuration benchmarks created by the Center for Internet Security (CIS) is a very interesting initiative [23]. CIS is a non-profit organization formed by several well-known academic, commercial, and governmental entities that has created a series of security configuration documents for several commercial and open source systems. These documents focus on the practical aspects of the configuration of these systems and state the concrete values each configuration option should have in order to enhance overall security of real installations. Although CIS refers to these documents as benchmarks they mainly reflect best practices and are not explicitly designed for systems assessment or comparison.

Vieira & Madeira proposed a practical way to characterize the security mechanisms in database systems [24]. In this approach database management systems (DBMS) are classified according to a set of security classes ranging from Class 0 to Class 5 (from the worst to the best). Systems are classified in a given class according to the security requirements satisfied. In [25] the authors analyze the security best practices behind the many configuration options available in several well-known DBMS. These security best practices are then generalized and used to define a set of configuration tests that can be used to compare different database installations. A benchmark that allows database administrators to assess and compare database configurations is presented in [26]. The benchmark provides a trust-based security metric, named minimum untrustworthiness, that expresses the minimum level of distrust the DBA should have in a given configuration regarding its ability to prevent attacks.

The use of trust-based metrics as an alternative to security measurement is discussed in [27]. This work also proposed a trustworthiness benchmark based on the systematic collection of evidences (collected using static analysis techniques) that can be used to select one among several web applications, from a security point-of-view.

Security benchmarking, and security assessment in general, is an open research problem. In fact, although there are several works in the literature, there is no “good enough” model for assessing and comparing the security of alternative systems and components. A key issue is that security is largely related with the “unknown” vulnerabilities and attacks, and comparing systems based on well defined attackloads may lead to conclusions that ultimately do not hold in the field (e.g. when a new vulnerability or attack type is discovered). Thus additional work is required to best understand the problem, propose generic frameworks and models for security comparison, studying the representativeness of attackloads, understand how new vulnerability and attack types can be considered, etc.

V. SECURITY IN THE SOFTWARE PROCESS

A software development process is composed of multiple phases [28]. To improve the situation in software security it is important not only to understand the existing approaches and tools but also to adequately integrate them in the development process, i.e. to use such approaches and tools in the points of the process where they can make the difference. Different authors divide the software process in different ways, but usually software development includes the following phases (which can be repeated in an iterative manner): initialization, design, implementation, testing, deployment and decommissioning.

The process starts with requirements gathering (including security requirements), followed by specification and design, implementation (coding), testing and deployment. Decommission takes place when the product is not useful/used anymore. Although code security concerns should be addressed during the entire software product development lifecycle, as highlighted by [29] especial focus should be put in three key phases [30]: implementation, testing, and deployment. The next points summarize the main challenges and put in the context of these three phases the concepts and techniques introduced before:

- **Implementation**: during coding we must use best practices that avoid the most critical vulnerabilities in the specific application domain. Examples of practices include input and output validation, the escaping of malicious characters, and the use of parameterized commands [1]. Vulnerability and attack injection techniques have in this phase a very important job in the evaluation of the best security testing tools to use. Also, for the success of this phase, it is essential to adequately train the development teams. For instance, experience shows that the main reason for the vulnerabilities existing in web application's code is related to training and education. First, there is a lack of courses/topics regarding secure design, secure coding, and security testing, in most computer science degrees [30]. Second, security is not usually among the developers' main skills as it is considered a boring and uninteresting topic (from the development point-of-view), and not as a way to develop new and exciting functionalities.
- **Testing**: as introduced before, there are many security testing techniques available for the identification of vulnerabilities during the testing phase [1]. To mitigate vulnerabilities, it is necessary to have well-trained teams read that adequately apply those techniques during the development of the application. The problem is that software quality assurance teams typically lack the knowledge required to effectively detect security problems. It is necessary to devise approaches to quickly and effectively train security assurance teams in the context of web applications development, by combining vulnerability injection with relevant guidance information about the most common security vulnerabilities. Also, benchmarking techniques should be applied to assess, compare, and select the most adequate security testing tools for each concrete scenario.
- **Deployment**: at runtime, it is possible to include in the environment different attack detection mechanisms, such as Intrusion Detection Systems (IDS) and Web Application Firewalls (WAF), among others. These mechanisms can operate at different levels and use different detection approaches. The main problems preventing their use are related to the performance overheads and to the false positives that disrupt the normal behavior of the system. In this phase, security benchmarking plays a fundamental role in helping to select the best alternatives (in terms of servers, security mechanisms, etc.) to use, according to specific security requirements. Also, vulnerability and attack injection techniques represent in this phase an efficient way to evaluate the effectiveness of attack detections mechanism to be installed.

VI. CONCLUSION

In this paper we introduced techniques for security testing and assessment in the context of web applications (some of them quite novel such as security benchmarking and vulnerability and attack injection). As an essential condition for deploying secure systems, we also discussed aspects related to the software development process. These are of extreme importance for software designers and developers and allow an effective assessment of the security attributes of the software components being designed/deployed.

The paper highlighted several research challenges in an attempt to motivate further research in these topics. The paper did not intend to provide a comprehensive survey, but to focus on key promising aspects in which research is need, but that can already be applied in the context of the software industry.

REFERENCES

[1] D. Stuttard and M. Pinto, *The web application hacker's handbook: discovering and exploiting security flaws*. Wiley Publishing, Inc., 2007.

[2] G. McGraw and B. Potter, "Software Security Testing," *IEEE Security and Privacy*, vol. 2, no. 5, pp. 81–85, 2004.

[3] S. Christey and R. A. Martin, "Vulnerability type distributions in CVE," *V1. 0*, vol. 10, p. 04, 2006.

[4] A. Stock, J. Williams, and D. Wichers, "OWASP Top 10," 2007.

[5] A. Singhal, T. Winograd, and K. Scarfone, "Guide to Secure Web Services: Recommendations of the National Institute of Standards and Technology," *Report, National Institute of Standards and Technology, US Department of Commerce*, pp. 800–95, 2007.

[6] OWASP Foundation, "OWASP Application Security FAQ Version 3," 2010. [Online]. Available: http://www.owasp.org/index.php/OWASP_Application_Security_FAQ. [Accessed: 09-Aug-2010].

[7] N. Antunes and M. Vieira, "Comparing the Effectiveness of Penetration Testing and Static Code Analysis on the Detection of SQL Injection Vulnerabilities in Web Services," in *15th IEEE Pacific Rim International Symposium on Dependable Computing, 2009. PRDC '09*, Shanghai, China, 2009, pp. 301–306.

[8] J. Carreira, H. Madeira, and J. G. Silva, "Xception: A technique for the experimental evaluation of dependability in modern computers," *IEEE Transactions on Software Engineering*, vol. 24, no. 2, pp. 125–136, 1998.

[9] M. Rodríguez, F. Salles, J.-C. Fabre, and J. Arlat, "MAFALDA: Microkernel Assessment by Fault Injection and Design Aid.," in *EDCC*, 1999, vol. 1667, pp. 143–160.

[10] J. A. Duraes and H. S. Madeira, "Emulation of Software Faults: A Field Data Study and a Practical Approach," *IEEE Transactions on Software Engineering*, vol. 32, no. 11, pp. 849–867, 2006.

[11] J. Durães and H. Madeira, "Definition of Software Fault Emulation Operators: A Field Data Study.," in *DSN*, 2003, pp. 105–114.

[12] J. Fonseca and M. Vieira, "Mapping software faults with web security vulnerabilities," presented at the IEEE International Conference on Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008., 2008, pp. 257–266.

[13] N. Neves, J. Antunes, M. Correia, P. Verissimo, and R. Neves, "Using Attack Injection to Discover New Vulnerabilities," in *International Conference on Dependable Systems and Networks, 2006. DSN 2006*, 2006, pp. 457–466.

[14] J. Fonseca, M. Vieira, and H. Madeira, "Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks," in *13th Pacific Rim International Symposium on Dependable Computing (PRDC 2007)*, Melbourne, Australia, 2007, pp. 365–372.

[15] J. Fonseca, M. Vieira, and H. Madeira, "Vulnerability & attack injection for web applications," in *IEEE/IFIP International Conference on Dependable Systems & Networks, 2009. DSN '09*, 2009, pp. 93–102.

[16] J. Gray, *Benchmark Handbook: For Database and Transaction Processing Systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992.

[17] K. Kanoun and L. Spainhower, *Dependability Benchmarking for Computer Systems*. Wiley-IEEE Computer Society Pr, 2008.

[18] Commission of the European Communities, *The IT Security Evaluation Manual (ITSEM)*. 1993.

[19] P. K. Infrastructure and T. P. Profile, "Common Criteria for Information Technology Security Evaluation," 2002.

[20] L. Qiu, Y. Zhang, F. Wang, M. Kyung, and H. R. Mahajan, "Trusted computer system evaluation criteria," in *National Computer Security Center*, 1985.

[21] Sandia National Laboratories, "Information Operations Red Team and Assessments^{TM}." [Online]. Available: http://www.sandia.gov/iorta/. [Accessed: 23-Sep-2012].

[22] R. A. Maxion and K. M. C. Tan, "Benchmarking anomaly-based detection systems," in *Proceedings International Conference on Dependable Systems and Networks, 2000. DSN 2000*, 2000, pp. 623 – 630.

[23] "Center for Internet Security." [Online]. Available: http://www.cisecurity.org/. [Accessed: 23-Sep-2012].

[24] M. Vieira and H. Madeira, "Towards a security benchmark for database management systems," in *International Conference on Dependable Systems and Networks, 2005. DSN 2005.*, Yokohama, Japan, 2005, pp. 592 – 601.

[25] A. A. Neto and M. Vieira, "Towards assessing the security of DBMS configurations," in *IEEE International Conference on Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008*, 2008, pp. 90 –95.

[26] A. A. Neto and M. Vieira, "A Trust-Based Benchmark for DBMS Configurations," in *15th IEEE Pacific Rim International Symposium on Dependable Computing, 2009. PRDC '09*, 2009, pp. 143 –150.

[27] A. A. Neto and M. Vieira, "Benchmarking Untrustworthiness," *International Journal of Dependable and Trustworthy Information Systems*, vol. 1, no. 2, pp. 32–54, 32 2010.

[28] C. Ghezzi, M. Jazayeri, and D. Mandrioli, *Fundamentals of software engineering*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2002.

[29] G. McGraw, *Software Security: Building Security In*. Addison-Wesley Professional, 2006.

[30] M. Howard and D. E. Leblanc, *Writing Secure Code*, 2nd ed. Redmond, Washington: Microsoft Press, 2002.

Marco Vieirais an assistant professor at the University of Coimbra, Portugal. His interests include dependability and security benchmarking, experimental dependability evaluation, fault injection, software development pro-cesses, and software quality assurance. Vieira received a PhD in computer engineering from the University of Coimbra. He is a member of the IEEE Computer Society.

Design of a Set of Software Tools for Side-Channel Attacks

A. Fuentes, L. Hernández, A. Martín and B. Alarcos

***Abstract*— This contribution presents the design and the first experimental results of a set of software tools to carry out side-channel attacks against cryptographic devices, especially smartcards. To this aim, the main attacks of this class are commented, with special emphasis in power analysis attacks. The final objective is to make this set of tools available to the scientific community, so that it can be improved and enlarged according to particular needs.**

***Keywords*— side channels, cryptography, software tools, security.**

I. Introducción

TRADICIONALMENTE se acepta que la Criptología se divide en dos partes claramente definidas, la Criptografía y el Criptoanálisis [8], [15]. La primera de ellas tiene como objetivo principal el diseño y la elaboración de sistemas que permitan el cifrado de información, de modo que sólo los usuarios autorizados puedan recuperar la información original, mediante el correspondiente proceso de descifrado. Ambos procesos, cifrado y descifrado, utilizan algoritmos cuyas entradas son la información que se desea cifrar o descifrar y determinadas claves. Por el contrario, el Criptoanálisis tiene como fin romper el secreto de tales comunicaciones, ya sea determinando las claves que se utilizaron en los procesos de cifrado y descifrado, o rompiendo el algoritmo en el que se basan tales procesos.

Como ya se ha dicho, una de las características fundamentales de los sistemas criptográficos en general, ya sean criptosistemas simétricos o asimétricos, esquemas de firma digital, protocolos de acuerdo o intercambio de clave, etc., es el estudio de su seguridad. La primera de las medidas que debe garantizarse para considerar que un sistema criptográfico es seguro es que el espacio de todas sus posibles claves ha de ser lo suficientemente elevado como para hacer inviable un ataque por fuerza bruta, es decir, por la prueba exhaustiva de todas las claves. La segunda medida que se debe considerar es la imposibilidad de romper el algoritmo en el que se basa el sistema criptográfico bajo estudio.

De este modo, se debe asegurar que el algoritmo no es vulnerable a ataques del tipo hombre en el medio (*man in the middle*), a medio camino, diferencial, etc. [6]. En otras ocasiones, la seguridad de los sistemas es hipotética (aunque aceptada universalmente), como en el caso de algoritmos que basan su seguridad en determinados problemas matemáticos, fundamentalmente de la teoría de números, como la factorización de enteros [17], el logaritmo discreto o elíptico [7], [9], la suma de un subconjunto o mochila [5], etc.

La clasificación más general de los ataques (teóricos) consiste en considerarlos como pasivos o activos [15]. Un ataque se dice pasivo si el atacante sólo monitoriza el canal de comunicaciones e intenta vulnerar la confidencialidad de los datos, esto es, obtener el texto claro a partir del texto cifrado; mientras que un ataque se llama activo si el adversario intenta borrar, añadir o, en general, alterar la transmisión de la información; es decir, su objetivo es amenazar la integridad de los datos, su autenticidad y su confidencialidad. Los ataques pasivos se suelen dividir en los siguientes tipos, dependiendo de la clase de información a la que el atacante tenga acceso: ataque al texto cifrado, al texto claro conocido, al texto claro elegido, al texto claro elegido adaptativo, al texto cifrado elegido y al texto cifrado elegido adaptativo. En cuanto a los ataques más extendidos a los protocolos destacan los siguientes: a la clave conocida, por repetición, por suplantación, por diccionario, por búsqueda hacia adelante y por intercalado.

Por otra parte, es sabido que sólo los algoritmos criptográficos cuya seguridad teórica ha sido demostrada o garantizada son los que pasan a la fase de ser implementados en dispositivos criptográficos como etiquetas RFID, tokens USB y, principalmente, tarjetas inteligentes. Las principales razones que motivan las implementaciones de protocolos criptográficos en estos dispositivos son su ubicuidad, facilidad de uso, comodidad y el hecho de que incorporen medidas de seguridad. Todo ello lleva al uso de procesadores criptográficos optimizados para tareas específicas que son embebidos en dichas tarjetas.

Todos estos dispositivos capaces de realizar operaciones criptográficas se han convertido en una herramienta indispensable en muchas actividades de nuestra vida cotidiana, como por ejemplo, la identificación de individuos (autenticación de usuarios en sistemas de información, tarjetas de identificación personal, pasaportes, etc.), la firma electrónica para garantizar el acceso a determinada información o el pago por bienes y servicios (tarjetas de firma, de identificación personal, etc.) y el almacenamiento de información cuya manipulación está restringida a entidades específicas (tarjetas prepago, tarjetas SIM, etc.).

A. Fuentes, Departamento de Tratamiento de la Información y Criptografía, Instituto de Tecnologías Físicas y de la Información, Consejo Superior de Investigaciones Científicas, Madrid, España, alberto.fuentes@iec.csic.es

L. Hernández, Departamento de Tratamiento de la Información y Criptografía, Instituto de Tecnologías Físicas y de la Información, Consejo Superior de Investigaciones Científicas, Madrid, España, luis@iec.csic.es

A. Martín, Departamento de Tratamiento de la Información y Criptografía, Instituto de Tecnologías Físicas y de la Información, Consejo Superior de Investigaciones Científicas, Madrid, España, agustin@iec.csic.es

B. Alarcos, Departamento de Automática, Escuela Politécnica Superior, Universidad de Alcalá, Madrid, España, bernardo.alarcos@uah.es

Sin embargo, la seguridad teórica de los sistemas criptográficos no garantiza su seguridad práctica, dado que en su implementación intervienen otros muchos factores que es preciso considerar y que no suelen ser tenidos en cuenta desde el punto de vista del diseñador de un protocolo criptográfico.

La información almacenada en los dispositivos criptográficos (típicamente una tarjeta inteligente), sólo puede ser utilizada a través de los algoritmos definidos por los desarrolladores del dispositivo, por lo que un usuario de dicha tarjeta sólo puede hacer uso de datos y algoritmos tras ciertas comprobaciones y con las restricciones que se hayan implementado. Por todo ello, una constante fundamental en todos los productos criptográficos y, por tanto, en sus implementaciones, es el análisis de su seguridad, tanto desde el punto de vista teórico como práctico.

Un atacante a un protocolo de acuerdo de clave, de firma digital o de cifrado/descifrado, por ejemplo, siempre buscará su parte más débil. Y ésta es, en muchos casos, su implementación en una tarjeta inteligente.

Los tipos de ataque que tienen como objetivo obtener información a partir de la implementación insegura de un protocolo criptográfico en una tarjeta inteligente, se han dado en llamar ataques por canal lateral (*side channel attacks*) o por inducción de fallos (*fault attacks*).

Debe tenerse en cuenta que cuando un criptoprocesador ejecuta determinado código, que implementa un protocolo o algoritmo específico, conlleva, entre otras características, un tiempo de computación, un consumo de potencia eléctrica, la generación de ondas electromagnéticas, etc.

La hipótesis en la que se basan estos ataques es que las magnitudes e intensidades de dichas características dependen directamente de las instrucciones, operaciones matemáticas y datos utilizados por el procesador. De esta forma, la clave criptográfica empleada (ya sea secreta o privada) puede ser inferida mediante el análisis de la información obtenida midiendo las fugas que se producen por estos denominados canales laterales.

En este trabajo se estudian los ataques por canal lateral a sistemas criptográficos implementados en tarjetas y se presenta el diseño de un conjunto de herramientas que permitan llevar a cabo ataques por canal lateral. Se pretende poner dicha herramienta al servicio de la comunidad científica interesada.

El resto del contenido de esta comunicación se estructura de la siguiente manera. En la sección II se hace un repaso general de los principales tipos de ataques a los diferentes dispositivos físicos. El diseño de las herramientas de ataque por canal lateral se lleva a cabo en la sección III; mientras que en la sección IV se presentan algunos resultados obtenidos durante la fase de diseño de tales herramientas. Finalmente, las principales conclusiones de este trabajo se muestran en la sección V.

II. Ataques a Dispositivos Físicos

Como ya se ha mencionado, hasta hace unos años la garantía de los sistemas criptográficos se basaba en el estudio teórico de su seguridad; sin embargo, desde la publicación de los artículos de Kocher [12] y Boneh et al. [3], esta garantía ya no es suficiente si se desea que los sistemas criptográficos implementados en dispositivos físicos sigan considerándose seguros. Dado que estos ataques están relacionados con la implementación física, se suelen denominar genéricamente ataques físicos [8].

Los problemas de seguridad en las implementaciones surgen bien por la existencia de determinados canales por los que es posible obtener información sensible de la zona considerada segura del dispositivo, bien por la posibilidad de inducir fallos en el comportamiento del circuito electrónico y, analizando el comportamiento del sistema, deducir información sobre las claves.

Dado que los ataques físicos son menos generales que los criptoanálisis clásicos, puesto que están específicamente ligados al modo de implementación, arquitectura del chip, etc., se suelen clasificar, bien como invasivos, semiinvasivos o no invasivos [2], [18] según que se manipule el dispositivo o sólo se haga uso de información disponible; bien como activos o pasivos, en analogía con los criptoanálisis clásicos, según que los ataques traten de manipular el funcionamiento del dispositivo o sólo observen el comportamiento del mismo durante el procesado del algoritmo criptográfico.

En los ataques físicos se supone, de forma análoga al criptoanálisis tradicional, que se verifica el principio de Kerckhoffs [11], esto es, el atacante tiene acceso al dispositivo y conoce el algoritmo criptográfico que está implementado en el chip, así como los detalles de la implementación; de modo que lo único que no se conoce es la clave. Por otra parte, se supone que el dispositivo se podrá hacer funcionar tantas veces como se desee, eligiendo los valores de entrada, y se podrá actuar sobre él o medir ciertos parámetros en su entorno.

A continuación se describen los principales tipos de ataques físicos.

A. Ataques por análisis temporal

Los ataques por análisis temporal (*timing attack*) pretenden obtener información sobre un criptosistema midiendo el tiempo que el dispositivo tarda en realizar las operaciones del algoritmo que implementa [12].

A modo que ejemplo, supóngase que se desea conocer la clave privada de un criptosistema de clave asimétrica mientras se ejecuta la exponenciación modular mediante el algoritmo de elevar al cuadrado y multiplicar. Es sabido que el tiempo de ejecución de la multiplicación es constante, pero si el resultado de la multiplicación es mayor que el módulo considerado, se debe hacer una reducción adicional y el tiempo de ejecución aumenta.

Mediante esta hipótesis, es claro que se obtiene información acerca de los tamaños de los números que intervienen en cada paso de la operación, lo que ayuda a la determinación de la clave si se ejecutan un gran número de muestras de texto claro.

Es sabido que la memoria caché almacena copias de los datos usados con mayor frecuencia. Cuando el procesador necesita leer una celda de la memoria principal, primero comprueba si el dato está ya en la caché. Si ése es el caso (*cache hit*), el procesador usa el dato de modo inmediato sin acceder a la memoria principal; en caso contrario (*cache miss*), el dato se lee de la memoria y se almacena en la caché.

Este comportamiento se aprovecha en los ataques denominados por análisis de la caché, que se suelen emplear en criptosistemas de clave simétrica, dado que su tiempo de ejecución no presenta suficiente correlación con los valores cifrados. En este caso se hace uso del hecho de que el tiempo requerido para acceder a la caché cuando los datos están presentes en ella es mucho menor que cuando no están [10].

Otro tipo de ataque por análisis temporal contra las CPU de los ordenadores se denomina por análisis de la predicción de saltos (*branch prediction analysis*). En este caso se hace uso del procesado determinístico de las unidades de predicción de saltos de las modernas CPU, que predicen la secuencia de instrucciones esperada antes de obtener el resultado real de la directiva de salto. El ataque aprovecha la penalización en tiempo (ciclos extra de reloj) que supone un error en la predicción de salto en el flujo de un programa cuya secuencia depende de los datos utilizados [1]. La relación entre la penalización en tiempo y los datos procesados es utilizada por el atacante para obtener información sobre dichos datos, entre los que se encuentra la clave buscada.

B. *Ataques por análisis de potencia*

Dado que el consumo de potencia de los microprocesadores actuales está muy relacionado con el número de bits que cambian en memoria o registros, un atacante puede aprovechar este hecho para intentar adivinar un valor secreto utilizado en una operación criptográfica observando la curva de consumo de potencia o traza. A continuación se comentan de forma resumida los principales ataques que emplean este método.

Los ataques por análisis simple de potencia (*Simple Power Analysis*, SPA) emplean trazas de consumo de potencia medidas durante el funcionamiento del dispositivo criptográfico. Estas medidas se determinan con un osciloscopio digital que mide la caída de tensión en una resistencia que se conecta en serie con el dispositivo, a la alimentación. Para que el atacante pueda obtener la clave más o menos directamente a partir de una determinada traza (o de un conjunto de unas pocas trazas), lo más normal es que necesite un conocimiento detallado del algoritmo criptográfico que está siendo ejecutado y, a ser posible, de las características de su implementación en el dispositivo atacado. En algunas ocasiones, un análisis visual de la traza permite obtener información relevante sobre el tipo de criptosistema empleado.

A modo de ejemplo, en la Fig. 1 se muestra una traza de potencia de una ejecución de un DEA (*Data Encryption Algorithm*), donde se aprecia la repetición de un patrón 16 veces.

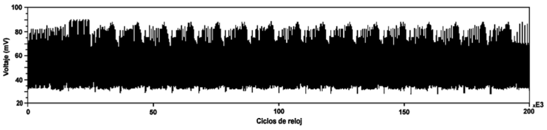


Figura 1. Traza de potencia de un DEA.

Si la relación entre la potencia consumida y la clave criptográfica no es clara, la señal que se obtiene suele ser muy débil y los ataques SPA no dan información suficiente para obtener la clave. En este caso, se suelen emplear técnicas estadísticas y ejecutar un ataque por análisis diferencial de potencia (*Differential Power Analysis*, DPA). Para llevar a cabo este ataque se hacen muchos cifrados con diferentes entradas, se miden y almacenan las correspondientes curvas de consumo de potencia y se sincronizan las medidas (ver Fig. 2 es decir, se procede a un alineamiento de las trazas de modo que se garantice la comparación de valores de potencia consumida en los mismos instantes a lo largo de la ejecución del algoritmo [13].

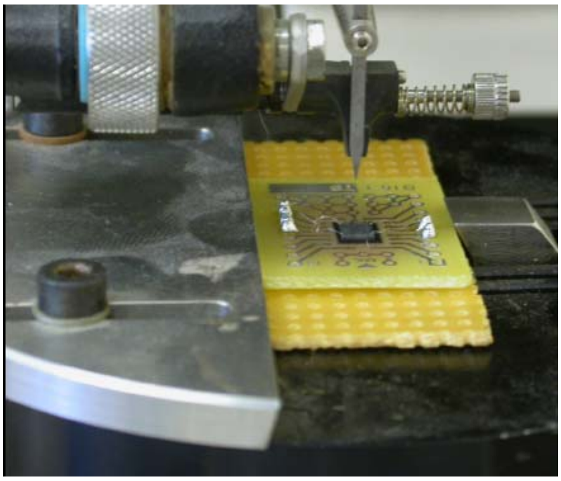


Figura 2. Cableado para medida del consumo de potencia del chip.

Los ataques por correlación de la potencia consumida (*Correlation Power Analysis*, CPA) consideran la correlación entre los datos procesados y la medición de la potencia instantánea consumida por el dispositivo [4]. Dado que esta correlación suele ser muy pequeña, se llevan a cabo muchas medidas para disponer de muchas trazas y utilizar después métodos estadísticos para comparar dichas trazas con las salidas de un determinado modelo del dispositivo.

Finalmente, señalar que los ataques DPA se pueden generalizar a los llamados ataques por análisis diferencial de orden superior (*Higher-Order Differential Power Analysis*, HODPA) en los que no se usa un único punto de la traza de potencia sino varios. Se trata de analizar si el consumo en su conjunto está correlado con el valor del bit según la conjetura que se haya hecho sobre la clave. En general, un ataque DPA de orden *n* hace uso simultáneamente de *n* muestras que corresponden a *n* valores intermedios diferentes (medidos en diferentes instantes) de la misma traza.

C. *Ataques por análisis de emanaciones electromagnéticas*

Los ataques por análisis de emanaciones electromagnéticas (*ElectroMagnetic Analysis*, EMA) consideran las emanaciones electromagnéticas emitidas por un circuito debidas, según las ecuaciones de Maxwell, al desplazamiento de cargas a lo largo de las pistas de las capas de metal del circuito. Estas emanaciones son especialmente significativas cuando los transistores conmutan de estado [16]. En este tipo de ataque se colocan sondas en las cercanías del chip para medir el campo electromagnético (ver Figura 3).

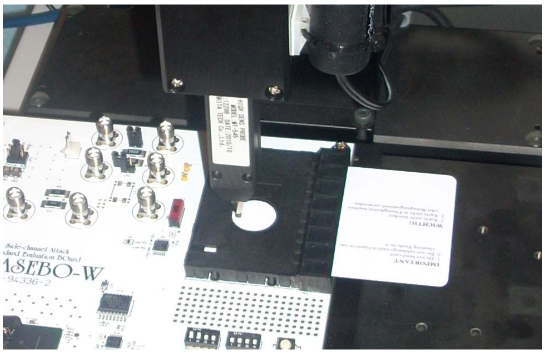


Figura 3. Medidas con sonda para un ataque EMA.

Aunque en la práctica se puede medir tanto el campo eléctrico como el magnético, para los ataques por canal lateral se obtienen mejores resultados haciendo uso del campo magnético. Por otra parte, es posible aumentar significativamente la determinación de este campo si se lleva a cabo un decapado previo del chip, permitiendo acercar más la sonda de medida a las zonas del chip donde están las fuentes de emanación.

La información recogida se analiza de forma similar a como se hace con las trazas de potencia y entonces se habla de análisis electromagnético simple (*Simple ElectroMagnetic Analysis*, SEMA) o de análisis electromagnético diferencial (*Differential ElectroMagnetic Analysis*, DEMA).

Los ataques de tipo EMA pueden proporcionar más información que los análisis de potencia consumida, pues, por ejemplo, pueden señalar la distribución de los componentes del chip si se conoce la orientación del campo electromagnético medido. En ocasiones puede darse el caso de que es posible medir el campo electromagnético y no el consumo de potencia.

D. *Ataques por inducción de fallos*

Los ataques por inducción de fallos son ataques activos que intentan manipular un dispositivo criptográfico para alterar su funcionamiento normal, de modo que dicha manipulación proporcione como salida del dispositivo un resultado erróneo en un cálculo, un mensaje de error, etc. Se trata, en definitiva, de obtener algún tipo de información no prevista originalmente en su implementación y explotar dicha información.

Para realizar este tipo de ataque puede ser necesario conocer la distribución de los componentes del chip dentro del mismo, es decir, en qué posición del chip se encuentra el criptoprocesador, determinadas celdas de memoria, etc., de tal manera que sea posible inducir el fallo en una localización específica del chip (ver Fig. 4). Para ello suele ser necesario un considerable trabajo de ingeniería inversa.

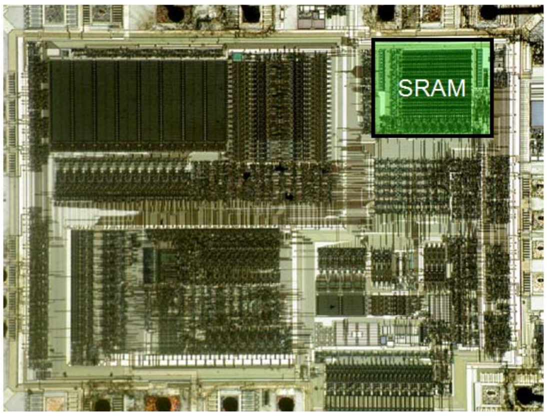


Figura 4. Identificación de regiones sensibles por ingeniería inversa.

Algunas de las técnicas para inducir fallos más comúnmente utilizadas son las siguientes.

Por calor o radiación infrarroja, de modo que si se consigue hacer funcionar un circuito fuera de los márgenes de la temperatura señalada por el fabricante, se puede explotar el hecho de que los umbrales de temperatura para la lectura y escritura no coinciden en la mayoría de las memorias no volátiles.

Los picos de tensión por encima del nivel de tolerancia de los dispositivos pueden provocar un resultado erróneo en un cálculo o fallos en la memoria.

Las variaciones en la frecuencia de reloj, al igual que en el caso anterior, pueden provocar fallos en la memoria o ejecuciones incorrectas de los programas.

Si fuera posible acceder a las capas de silicio de un chip, mediante el uso de rayos láser (rojos o verdes) o luz ultravioleta convenientemente enfocada, se pueden destruir algunas de sus estructuras o alterar valores de la memoria (ver Fig. 5).

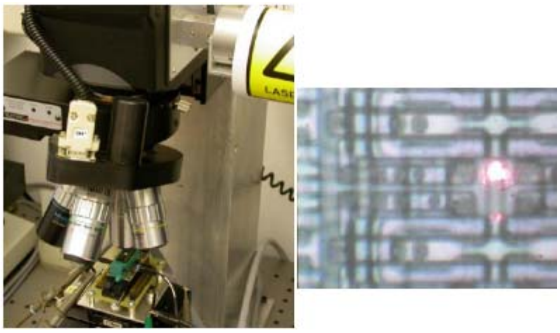


Figura 5. Disparo de luz láser contra un área de memoria.

El paso de corriente eléctrica por una bobina produce un campo magnético, de modo que si se coloca la bobina cerca de la superficie de un chip se inducen en él corrientes de Foucault que pueden afectar al correcto funcionamiento de un transistor o de un bloque de memoria.

En otras ocasiones se emplean haces de iones con el fin de perforar agujeros en las capas del chip e introducir por ellos material conductor de modo que sea posible acceder con equipos de medida a elementos individuales del chip.

Por otra parte, además de las técnicas de fallo, se suelen considerar modelos de fallo, que son representaciones de cómo el fallo ha influido en el comportamiento del dispositivo, de modo que se hace una hipótesis de cuántos bits han sido perturbados y cuál ha sido el impacto de la perturbación sobre tales bits.

E. Ataques por métodos combinados y contramedidas

Además de llevar a cabo los ataques ya presentados, es posible realizar ataques en los que intervengan varios de ellos de modo que la potencia del ataque se vea incrementada. De este modo, es posible combinar dos o más ataques utilizando, por ejemplo, el consumo de potencia y emanaciones electromagnéticas simultáneamente.

Es fácil suponer que cada vez que se publica un determinado tipo de ataque que pueda afectar a una serie de dispositivos o de algoritmos implementados en ellos, enseguida se plantea la necesidad de estudiar cómo los dispositivos pueden prevenir dichos ataques. Se trata, en definitiva, de implementar determinadas contramedidas que hagan inviable o, al menos, dificulten, tales ataques.

En todo caso, la decisión de implementar o no la contramedida, si es que se encuentra, dependerá de la valoración entre el coste de la contramedida y el valor de lo que se desea proteger. Las principales consideraciones que se suelen tener en cuenta para elaborar contramedidas son las que se mencionan a continuación.

Para evitar ataques por canal lateral se utilizan estrategias, tanto en hardware como software, como decorrelar las trazas de salida de ejecuciones individuales introduciendo retrasos temporales aleatorios y estados de espera, insertando instrucciones inútiles, haciendo aleatoria la ejecución de ciertas operaciones, etc.

Otras estrategias consisten en cambiar instrucciones críticas por otras cuya traza de consumo de potencia sea difícil de analizar; rediseñar la circuitería encargada de las operaciones aritméticas o de las transferencias a memoria; modificar los algoritmos criptográficos para que los ataques resulten ineficientes, como por ejemplo enmascarando de modo aleatorio datos y claves, etc.

III. Diseño de Herramientas para Ataques por Canal Lateral

Una vez que se han presentado los principales ataques por canal lateral y por fallos, consideraremos de modo especial los ataques de tipo DPA para los que estará especialmente diseñado el conjunto de herramientas presentado en este trabajo.

Como ya se ha mencionado anteriormente (ver §II-B), en los ataques DPA, al contrario de lo que sucede con los ataques SPA, un atacante no necesita un conocimiento detallado del dispositivo a atacar. Además, con estos ataques es posible obtener la clave secreta, incluso cuando las trazas de potencia sean ruidosas. Por otro lado, este tipo de ataque requiere una gran cantidad de trazas de potencia.

Un ataque DPA consta de los siguientes cinco pasos [14].

1) *Elegir un resultado intermedio del algoritmo ejecutado*. Este primer paso consiste en elegir un resultado intermedio del algoritmo criptográfico que ha sido ejecutado por el dispositivo a atacar. Dicho valor se obtiene a partir de una función que utiliza como entradas parte de las claves criptográficas y datos conocidos.
   Dicho de otro modo, el valor intermedio es el resultado de una función $f(c, k)$, siendo c un valor no constante conocido y k una parte pequeña de la clave buscada. En general, c es el texto claro o el texto cifrado que se combina con la clave o con una clave derivada.
2) *Medición del consumo de potencia*. En el segundo paso se mide el consumo de potencia del dispositivo criptográfico mientras cifra o descifra n bloques de datos (textos claros o textos cifrados).
   Para cada una de estas n ejecuciones de cifrado o descifrado, el atacante necesita conocer el valor, c, correspondiente que está implicado en el cálculo del resultado intermedio mencionado en el paso 1. Estos datos conocidos pueden escribirse como un vector $C = (c_1, \ldots, c_n)$, donde c_i denota el dato en la i-ésima ejecución del cifrado/descifrado.
   Durante cada una de estas ejecuciones, el atacante almacena una traza de potencia. Se denota por $T_i = (t_{i,1}, \ldots, t_{i,m})$ la traza correspondiente al bloque de datos c_i, siendo m la longitud de la traza. El atacante mide una traza para cada uno de los n bloques, de modo que las trazas se pueden representar como una matriz $T = (t_{i,j})$ de tamaño $n{\cdot}m$:
   $$T = \begin{pmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,m} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n,1} & t_{n,2} & \cdots & t_{n,m} \end{pmatrix}, \qquad (1)$$
   siendo $t_{i,j}$ el valor del voltaje medido por el osciloscopio al ejecutar el dato c_i en el instante j.
   Para un ataque DPA, es importante que las trazas medidas estén correctamente alineadas, es decir, los valores del consumo de potencia de cada columna de T deben corresponder a la misma operación. Para obtener este alineamiento en las trazas, la señal del disparador (*trigger*) del osciloscopio necesita ser generada de modo que el osciloscopio registre el consumo de potencia de modo exacto de cada secuencia de operaciones durante cada proceso de cifrado o descifrado.
   En el caso de no disponer de la señal del disparador, se debe realizar una alineación de las trazas mediante coincidencia de patrones.
3) *Cálculo de valores intermedios hipotéticos*. En este paso se determina un valor intermedio hipotético para cada elección de k, denotándose por $K = (k_1, \ldots, k_l)$ las posibles elecciones y por l su número. Los elementos del vector K suelen llamarse claves hipotéticas.
   Dado el vector de datos C y la clave hipotética K, un atacante puede calcular fácilmente los valores hipotéticos intermedios para las n ejecuciones de cifrado y para las l claves hipotéticas. Estos valores determinan una matriz $V = (v_{i,j})$, de tamaño $n{\cdot}l$, cuyos elementos vienen dados por:
   $$v_{i,j} = f(c_i, k_j), \quad 1 \le i \le n, \quad 1 \le j \le l, \qquad (2)$$
   es decir, se tiene que

$$V = \begin{pmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,m} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n,1} & v_{n,2} & \cdots & v_{n,m} \end{pmatrix}. \quad (3)$$

Las columnas de V contienen los resultados intermedios calculados y el objetivo del ataque DPA es determinar cuál de las columnas de V ha sido procesada durante las n rachas de cifrado o descifrado. En cuanto se conozca qué columna ha sido procesada en el dispositivo atacado, se conocerá la clave del dispositivo.

4) *Asignación de valores intermedios a los valores de consumo de potencia*. Este paso asigna los hipotéticos valores intermedios de V a una matriz $H = (h_{i,j})$, de tamaño $n{\cdot}l$, de valores hipotéticos de consumo de potencia. Para ello, el atacante utiliza diferentes técnicas de simulación.

   La calidad de la simulación depende en gran medida del conocimiento que el atacante tenga del dispositivo. Los modelos de potencia más utilizados para asignar V a H son los modelos basados en la distancia y el peso de Hamming.

5) *Comparación de los valores hipotéticos de consumo de potencia con las trazas de potencia*. El último paso del ataque consiste en comparar los valores hipotéticos del consumo de potencia con los valores del consumo de potencia medidos.

   Cada columna de H se compara con todas las columnas de T, es decir, el atacante compara los valores hipotéticos de consumo de potencia de cada clave hipotética con las trazas almacenadas en cada posición. El resultado de esta comparación es otra matriz $R = (r_{i,j})$, de tamaño $l{\cdot}m$, de modo que cada elemento $r_{i,j}$ contiene los resultados de la comparación entre la columna i-ésima de H y la j-ésima de T. Dependiendo del modelo de potencia utilizado, podrá existir una relación entre el consumo de potencia hipotético y el consumo de potencia real. Para determinar esta relación se utilizan métodos estadísticos.

Es claro que cuantas más trazas mida un atacante, más elementos habrá en cada columna de las matrices H y T, en cuyo caso será más probable que en R aparezcan datos destacados puesto que las relaciones entre las columnas serán más fáciles de determinar.

A. Obtención de trazas y calibración del osciloscopio

En los ataques por canal lateral se emplea un osciloscopio digital para obtener las trazas cuando el dispositivo está ejecutando operaciones criptográficas. Los osciloscopios pueden ser configurados de diferentes maneras, conforme a las características de los elementos a medir. Los principales parámetros que deben ser tenidos en cuenta para el almacenamiento de las trazas son los siguientes:

1) *Sensibilidad vertical*. Proporciona el rango de valores verticales (voltaje) que mide el osciloscopio, que suele ser un rango alrededor del cero, típicamente, ±100 mV, ±200 mV, ±500 mV. Cada osciloscopio posee un valor por defecto de estos valores que el usuario debe ajustar a las necesidades del problema a resolver.
2) *Frecuencia de muestreo*. Este valor especifica la velocidad a la que se sincroniza el convertidor analógico a digital en el osciloscopio para digitalizar la señal de entrada. Tal valor puede ser especificado en una frecuencia de captura (1 GS/s) o en un intervalo de tiempo entre capturas (1 ns).
3) *Resolución*. Los osciloscopios digitales discretizan los valores verticales (voltaje), de modo que la resolución proporciona el número de valores intermedios que pueden ser distinguidos por el osciloscopio, dentro del rango del valores del voltaje configurado por la sensibilidad vertical. Este valor se mide en bits y el número de valores intermedios se puede calcular como $2^{\text{resolution}}$. Por ejemplo, con una sensibilidad vertical de ±100 mV y 8 bits de resolución (256 valores intermedios), los valores adyacentes tienen una separación de 200/256 mV.
4) *Tamaño de memoria*. Los osciloscopios digitales almacenan valores de entrada en una memoria interna. Dependiendo de la cantidad de entradas capturadas por unidad de tiempo (sensibilidad horizontal), los datos pueden ser transferidos directamente a un PC en flujo (*streaming*) o en bloque (*block*). En este último caso, la longitud de la traza capturada está limitada por la memoria interna del osciloscopio.
5) *Offset de corriente continua (DC)*. El offset DC permite al usuario añadir un voltaje de corriente continua constante a la señal de entrada. Como ya se ha dicho, la sensibilidad vertical siempre está centrada en el valor cero; no obstante, cuando la señal es pequeña y distante de este valor, el offset puede ser utilizado para llevar la señal al rango adecuado para conseguir la sensibilidad vertical óptima.

Antes de iniciar el proceso de captura de las trazas, se deben especificar los parámetros a utilizar. Una vez que la captura finaliza, los datos son transferidos desde la memoria del osciloscopio a la memoria del PC. Las trazas capturadas son vectores de muestra y el tamaño de cada muestra es igual al parámetro de resolución. Este valor de muestra se presenta en bruto y el valor del voltaje se puede calcular como

$$V = S \cdot \frac{Raw}{Max} + O, \quad (4)$$

donde S es el máximo de la sensibilidad vertical, es decir, 50 mV, 100 mV o 200 mV; *Raw* es el valor en bruto; *Max* es el valor máximo representado por la resolución y O es el offset DC.

B. Método de programación

El primer paso a considerar en la implementación software es la determinación del método de programación que se va a utilizar, en función de los objetivos planteados y de las clases a lograr. Así, a la hora de establecer cuál es el mejor método de programación, se trata de decidir entre dos paradigmas de la programación para la reutilización y extensibilidad de código, el uso de herencia o el uso de plantillas (*templates*). En teoría, la herencia permite hacer desarrollo de código más limpio pero menos eficiente.

Para evaluar hasta qué punto penaliza el uso de herencia, se han desarrollado dos versiones de una parte crítica del código, con herencia y plantilla, y se han medido tiempos de procesamiento en cada una de las versiones. En la sección IV se indicarán con más detalle las versiones de código desarrollado y los resultados de las pruebas.

En la implementación de la versión con plantilla, ha sido necesario trasladar este método a varias clases. Se han realizado varias optimizaciones pero aun así el código no ha quedado lo suficientemente limpio. En la sección IV se comentarán los resultados de las pruebas y la decisión final sobre la implementación.

C. Representación de las trazas y su almacenamiento en memoria

Dado que la herramienta básica para obtener datos sobre los que trabajar en los ataques DPA son las trazas, se ha estudiado, en primer lugar, la manera más eficiente de almacenar en memoria los valores de las trazas capturadas. Uno de los principales objetivos es el de intentar minimizar la memoria usada, por lo que cada elemento de la traza deberá ocupar un espacio lo más cercano posible a la resolución del osciloscopio. Esto lleva consigo la ventaja de que además de almacenar más datos, se procesan más rápidamente, gracias al menor número de fallos de caché.

Se ha implementado otro punto de referencia para comprobar si es mejor representar las matrices que definen las trazas mediante columnas o filas para ser almacenadas en memoria y con el fin de determinar qué efectos tiene dicha representación sobre los fallos en la caché, es decir, analizar su rendimiento.

D. Implementación

En el desarrollo de la implementación, las clases mas importantes representadas son las siguientes:

CTrace: clase para contener una traza. Los elementos de la traza pueden tener una resolución de 8, 16 o 32 bits (CTrace8, CTrace16, CTrace32, respectivamente). Esta clase abstrae a otras clases de la resolución de los elementos (ver Fig. 6).

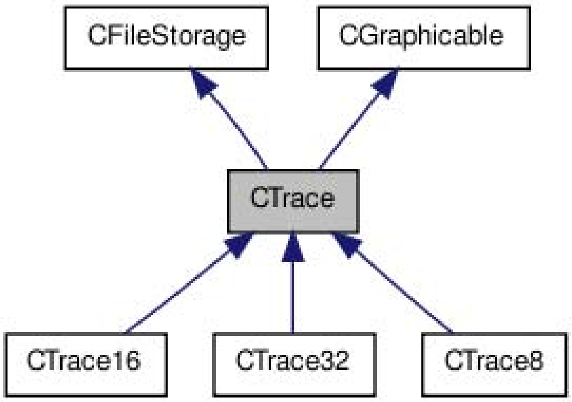


Figura 6. Clase CTrace.

Esta clase hace uso de las siguientes funciones.

- getVoltConversion(): devuelve el atributo voltConversion, es decir, el multiplicador para pasar las mediciones de *Raw* a voltaje.
- getDisplacement(): devuelve el atributo displacement, que es valor del offset de DC aplicado en las muestras.
- getSamplingRate(): devuelve el atributo samplingRate, esto es, el tiempo entre muestras.
- getValue(size_t pos)}: devuelve el valor de la traza, en mV, en la posición especificada.
- getRawValue(size_t pos): devuelve el valor obtenido en el osciloscopio.
- setValue(uint32_t value, size_t pos): establece la posición de la traza especificada con el valor dado.
- size_t size(): obtiene el tamaño de la traza (número de valores).
- eraseLast(): borra el último elemento de la traza.
- sizeElem(): obtiene la resolución de los valores en bruto, en Bytes.

CTimeSlice: clase que representa los valores de múltiples trazas en un instante de tiempo. Sobre estas trazas se realizará la correlación por distintos métodos, con el fin de obtener la clave u otros datos secretos (CTimeSlice8, CTimeSlice16, CTimeSlice32). Esta clase también requiere abstracción de la resolución de los elementos (ver Fig. 7).

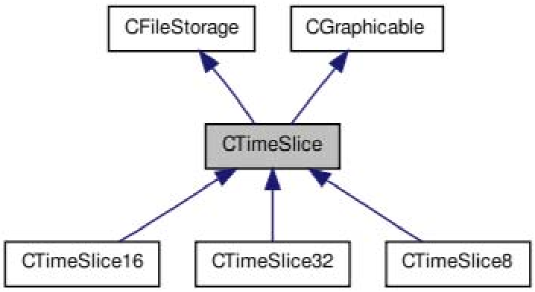


Figura 7. Clase CTimeSlice.

Las funciones incluidas en esta clase son las siguientes.

- getVoltConversion: ver CTrace.
- getDisplacement: ver CTrace.
- getValue: ver CTrace.
- getRawValue: ver CTrace.
- setValue: ver CTrace.
- size: ver CTrace.
- sizeElem: ver CTrace.
- getValue: ver CTrace.
- toDisk(int fd): almacena el objeto en el descriptor del fichero (ver §IV-B).
- fromDisk(int fd): recupera el objeto almacenado en el descriptor del fichero (ver §IV-B). El objeto deberá haber sido almacenado previamente en el fichero señalado por el descriptor mediante la función toDisc.
- toPng(char*fileName, list<string> *gnuplotCmds=NULL): produce un fichero de tipo png con la interpretación gráfica del objeto (ver §IV-C).

CTraceSet: clase que contiene un conjunto de trazas (ver Fig. 8). De esta clase se puede extraer CTimeSlice indicando el instante de tiempo. Además, se pueden obtener valores estadísticos, como la media y la varianza, del conjunto de trazas, útil para obtener, por ejemplo, la relación señal/ruido. Pueden ser preprocesables (alineamientos). Se han implementado dos algoritmos de alineamiento: mediante integración (*Integration*) y mediante reconocimiento de patrones (*Square match pattern*).

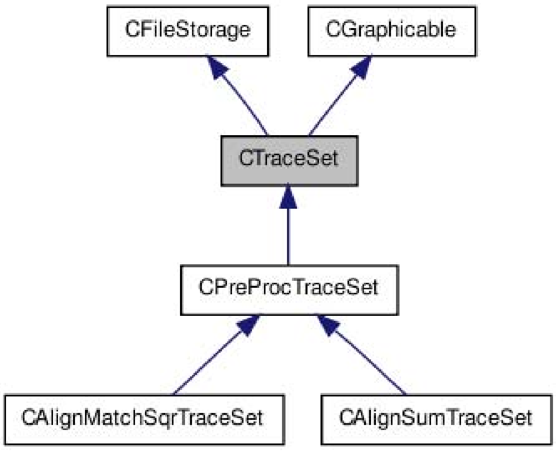


Figura 8. Clase CTraceSet.

Las funciones que se emplean en esta clase se describen a continuación.

- getLenTrace(): devuelve el número de elementos de cada traza.
- getLenSlice(): devuelve el número de elementos en cada instante de tiempo.
- addTrace(CTrace *trace): inserta una traza al final del conjunto de trazas. Las trazas se almacenan como referencias. El destructor CTraceSet libera la memoria. Este método sólo puede ser utilizado en el modo de alineamiento.
- statMode(): cambia la representación interna al modo estadístico. Mejora la ejecución de las operaciones que requieren el uso de instantes de tiempo.
- getNTrace(int pos): devuelve una copia de la traza almacenada en la posición especificada. El que la llama debe liberar el valor devuelto cuando se requiera (es más rápido en el modo de alineamiento).
- getNSlice(int pos): devuelve una copia del instante de tiempo almacenado en la posición especificada. El que la llama debe liberar el valor devuelto cuando se requiera (es más rápido en el modo estadístico).
- getNTraceRef(int pos): devuelve el puntero de la traza almacenada en la posición especificada. El que la llama no debe liberar el valor devuelto (es más rápido en el modo de alineamiento).
- getNSliceRef(int pos): devuelve un puntero al instante de tiempo almacenado en la posición especificada. El que la llama no debe liberar el valor devuelto (es más rápido en el modo de alineamiento).
- meanTraces(): devuelve un vector (CStatTrace) con el valor de la media aritmética de todas las trazas almacenadas. El que la llama debe liberar el valor devuelto cuando se requiera (es más rápido en el modo estadístico).
- varianceTraces(CStatTrace *mean=NULL): devuelve un vector (CStatTrace) con el valor de la varianza de todas las trazas almacenadas. El que la llama debe liberar el valor devuelto cuando se requiera (es más rápido en el modo estadístico).
- toDisk: ver CTimeSlice.
- fromDisk}: ver CTimeSlice.
- toPng: ver CTimeSlice.

CStatTrace: esta clase proporciona algunos datos estadísticos obtenidos a partir de varias trazas como la media, varianza, etc. (ver Fig. 9).

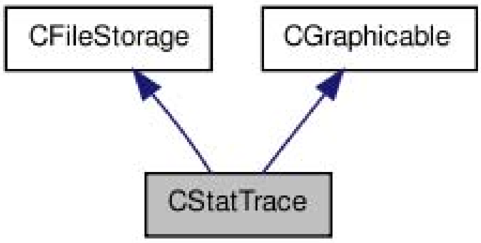


Figura 9. Clase CStatTrace.

Las principales funciones que emplea esta clase son:

- getSamplingRate: ver CTrace.
- getValue: ver CTrace.
- setValue: ver CTrace.
- size: ver CTrace.
- toDisk: ver CTimeSlice.
- fromDisk: ver CTimeSlice.
- toPng: ver CTimeSlice.

Como se puede ver en las Figs. 6-9, las clases anteriores hacen uso de otras dos clases, CFileStorage y CGraphicable. La primera de ellas contiene las funciones toDisk y fromDisk y especifica las subclases que pueden ser almacenadas en un fichero (ver Fig. 10). Por su parte, la segunda (ver Fig. 11) contiene la función toPng y señala las subclases que pueden ser imprimidas, si está instalado gnuplot.

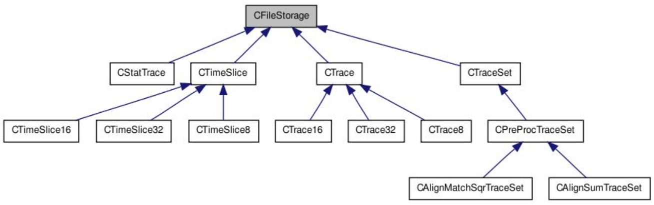


Figura 10. Clase CFileStorage}.

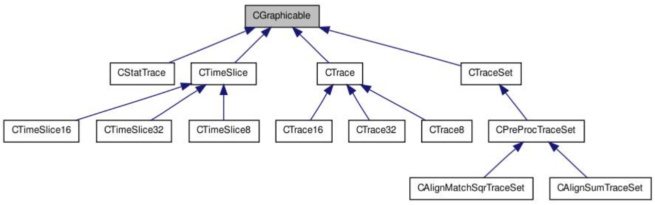


Figura 11. Clase CGraphicable.

IV. Resultados experimentales

Durante el diseño de este conjunto de herramientas se han evaluado varias opciones con el fin de resolver distintos problemas.

En primer lugar, y dado que tales herramientas se pretenden poner a disposición de los investigadores interesados, se ha estudiado la mejor forma de abstraer al usuario de propiedades de bajo nivel a la hora de trabajar con las trazas. De forma análoga, se ha analizado cómo se almacenarían las mismas en un soporte físico de forma óptima, como un disco duro; o qué clases podrían tener representación gráfica y qué tipo de representación gráfica se amoldaría mejor a cada clase.

Por todo ello, se han realizado los siguientes estudios.

A. Comparación entre plantillas y herencias

A la hora de tratar el diseño del software de almacenamiento de trazas (ver sección III), se han implementado dos posibles soluciones con el fin de analizar las ventajas e inconvenientes de cada una de ellas. Estas dos soluciones resuelven el problema de la abstracción en la precision de las trazas de dos modos distintos.

La implementación mediante plantillas proporciona mejores tiempos teóricos de ejecución que los correspondientes a las herencias, dado que las herencias tienen que resolver los métodos *virtuales* en tiempo de ejecución. No obstante, en la práctica, las optimizaciones de los compiladores pueden igualar dichos tiempos. En teoría la penalización de la versión con herencia es del 10%, si bien, en la práctica, el compilador con las opciones de optimización resuelve las llamadas a métodos virtuales, con lo que la diferencia es mínima.

Dado que se trata de una *toolbox*, es decir, un conjunto de herramientas que cualquier usuario puede utilizar y que está diseñado para ser extendido a su conveniencia, es fundamental elegir un método que ofrezca una alta claridad en el código. Con relación a este tema, hay que diferenciar entre las clases de más bajo nivel y las clases de más alto nivel que, por su parte, hacen uso de las primeras:

- *Clases de bajo nivel con plantillas*. La codificación mediante plantillas (ver Tabla I) sólo requiere una clase, ya que la resolución de la traza viene dada por la plantilla (template<class T> class CTrace). Cuando se ejemplifica dicha clase, se especifica la resolución denotando el tipo que se le asignará a T de la siguiente manera: CTrace<uint8_t> trace;. Este tipo puede ser uint8_t, uint16_t o uint32_t, representando resoluciones de 8, 16 y 32 bits, respectivamente.

TABLA I. EJEMPLO DE CÓDIGO DE BAJO NIVEL CON PLANTILLA.

```
template <class T> class CTrace : public valarray<T>{
protected:
    uint64_t samplingRate;
    double voltConversion;
public:
    CTrace (int fd);
    CTrace (uint64_t samplingRate, double voltConversion,
                T* vals, unsigned int lengh);
    CTrace (uint64_t samplingRate, double voltConversion);
    CTrace (CTrace<T> &trace, unsigned int begin,
                unsigned int len);
    double getVoltConversion (){return voltConversion;};
    uint64_t getSamplingRate (){return samplingRate;};
    int toDisk (int fd);
    int toPng (char *fileName, list<string> *gnuplotCmds
                = NULL);
};
```

- *Clases de bajo nivel con herencias*. La codificación usando herencias requiere de una clase por cada resolución posible, por ejemplo, CTrace8, CTrace16 y CTrace32 (ver Tabla II para una resolución de 8 bits) y una clase abstracta que sea superclase de las anteriores (ver Tabla III).

TABLA II. EJEMPLO DE CÓDIGO DE BAJO NIVEL CON HERENCIA.

```
class CTrace8 : public CTrace{
protected:
    vector <uint8_t> values;
public:
    CTrace8 (uint64_t samplingRate, double voltConversion,
                double displacement, uint8_t* values, size_t size);
    CTrace8 (uint64_t samplingRate, double voltConversion,
                double displacement, size_t size);
    CTrace8 ();
    float getValue (size_t pos){return float(values[pos]) *
                voltConversion + displacement;};
    uint32_t getRawValue (size_t pos) {return uint32_t
                (values[pos]);};
  int setValue (uint32_t value, size_t pos)
                {values[pos] = value;};
    size_t size (){return values.size();};
    void eraseLast (){values.pop_back();};
    size_t sizeElem (){return 8;};
    int toDisk (int fd);
    int fromDisk (int fd);
    int toPng (char *fileName, list<string> *gnuplotCmds
                = NULL);
};
```

TABLA III. EJEMPLO DE CÓDIGO DE BAJO NIVEL CON PLANTILLA.

```
class CTrace : virtual public CFileStorage,
                virtual public CGraphicable{
protected:
    uint64_t samplingRate;
    double voltConversion;
    double displacement;
public:
    CTrace (uint64_t samplingRate, double voltConversion,
                double displacement);
    CTrace ();
        // get methods for class parameters
    double getVoltConversion (){return voltConversion;};
    double getDisplacement (){return displacement;};
    uint64_t getSamplingRate (){return samplingRate;};
    virtual float getValue (size_t pos) = 0;
    virtual uint32_t getRawValue (size_t pos) = 0;
        // returns the oscilloscope value.
    virtual int setValue (uint32_t value, size_t pos) = 0;
        // set oscilloscope value.
    virtual size_t size () = 0; // returns its size.
    virtual void eraseLast () = 0;
        // erases last element (req. by CTraceSet::statMode)
    virtual size_t sizeElem () = 0;
        // used to know the value resolution (subClass used).
};
```

- *Clases de alto nivel con plantillas*. Cuando se emplean plantillas, las clases de alto nivel que hacen uso de clases de más bajo nivel, también se han de implementar con plantillas, de forma que se les pueda indicar la resolución de las trazas representadas por las clases de bajo nivel (ver Tabla IV). De este modo, las plantillas se extienden por gran parte del código de la toolbox.

TABLA IV. EJEMPLO DE CÓDIGO DE ALTO NIVEL CON PLANTILLA.

```
template <class T> class CTraceSet {
protected:
    vector < CTrace <T> > traces;
    int lenTrace;
    uint64_t samplingRate;
    double voltConversion;
public:
    CTraceSet (uint64_t samplingRate, double voltConversion,
              int lenTrace);
    CTraceSet (vector <CTrace<T> > &traces);
    CTraceSet (CTraceSet<T> &set, unsigned int begin,
              unsigned int len);
    CTraceSet (int fd);
    int getLenTrace (){return lenTrace;};
    uint64_t getSamplingRate (){return samplingRate;};
    double getVoltConversion (){return voltConversion;};
    int size (){return traces.size();};
    int addTrace (const CTrace<T> &trace);
    CTrace <T> getNTrace (int pos);
    CTimeSlice <T> getNSlice (int pos);
    virtual CTrace <double> meanTraces ();
    virtual CTrace <double> varianceTraces (CTrace <double>
                                              *mean = NULL);
    virtual int alignTraces (){};
    int toPng (char *fileName);
    int toDisk (int fd);
};
```

- *Clases de alto nivel con herencias*. En este caso, las clases de alto nivel hacen uso de la clase abstracta que es una superclase de las clases que especifican la resolución de las trazas (CTrace8, CTrace16, CTrace32), lo cual permite abstraerse de la subclase que está siendo utilizada (ver Tabla V).

TABLA V. EJEMPLO DE CÓDIGO DE ALTO NIVEL CON HERENCIA.

```
class CTraceSet : virtual public CFileStorage,
                 virtual public CGraphicable{
protected:
    vector <CTrace*> traces;
    vector <CTimeSlice*> slices;
    uint64_t samplingRate;
    size_t lenTrace;
    size_t lenSlice;
    bool StatMode;
    float getValue (size_t NTrace, size_t NSlice);
public:
    CTraceSet ();
    CTraceSet (vector <CTrace*> &traces);
    CTraceSet (CTraceSet &set, size_t begin, size_t len);
    ˜CTraceSet ();
    uint64_t getSamplingRate (){return samplingRate;};
    bool getStatMode (){return StatMode;};
    int getLenTrace (){return lenTrace;};
    int getLenSlice (){return lenSlice;};
    int addTrace (CTrace *trace); // only in alignment mode.
    void statMode ();
    CTrace* getNTrace (int pos);
    CTimeSlice* getNSlice (int pos);
    CTrace* getNTraceRef (int pos); // only in alignment mode.
    CTimeSlice* getNSliceRef (int pos);
        // only in statistical mode.
    virtual CStatTrace* meanTraces ();
    virtual CStatTrace* varianceTraces (CStatTrace* mean
              = NULL);
    int toPng (char *fileName, list<string> *gnuplotCmds
              = NULL);
    int toDisk (int fd);
    int fromDisk (int fd);
};
```

Para la implementación final, se han elegido las herencias, ya que se genera una codificación más limpia para las clases de alto nivel, que son más susceptibles de modificación o ampliación por la comunidad científica (por ejemplo, nuevos métodos estadísticos). Además, consideramos que la penalización que introduce el uso de herencia es asumible. Las clases de bajo nivel son simples y una vez optimizadas, es menos probable que se plantee su modificación.

B. *Interfaz para almacenamiento en disco*

Esta interfaz, cuyo diagrama de herencias se puede encontrar en la Fig. 10, permite almacenar en disco los objetos representados por las clases que la usan: CTrace, CTimeSlice, CTraceSet y CStatTrace.

La necesidad de esta interfaz surge por el coste de tiempo necesario para capturar el número de trazas requerido para el análisis estadístico. El almacenamiento permite capturar trazas en distintas tandas, así como hacer uso de ellas en momentos o localizaciones diferentes.

Dada la cantidad de información a almacenar, ha sido preciso intentar optimizar el espacio requerido para ello. Con este fin los atributos se almacenan de forma secuencial, ajustándolos al mínimo número de bits requerido para cada uno de ellos.

A modo de ejemplo, un objeto de la clase CTrace almacenaría la siguiente información.

- *Muestreo* (samplingRate): indica el muestreo, en picosegundos, que el osciloscopio utilizó para tomar la traza.
- *Conversión* (voltConversion): es el valor de la sensibilidad vertical que utilizó el osciloscopio para tomar la traza.
- *Desplazamiento* (displacement): señala el desplazamiento que hay que añadir para obtener el valor

de voltaje real.

- *Resolución* (bytesData): es el número de bits que ocupa cada dato (punto de la traza).
- *Tamaño* (sizeData): corresponde al número de datos que contiene la traza.
- *Datos* (Data): son los datos expuestos de forma secuencial.

Cada una de las clases almacena los datos correspondientes de modo secuencial como sigue:

- CTrace: samplingRate || voltConversion || displacement || bytesData || sizeData || Data …
- CTimeSlice: voltConversion || displacement || bytesData || sizeData || Datav
- CTraceSet: samplingRate || lenTrace || lenSlice || StatMode || data resolution || if (StatMode) Data Slice 0 … || Data Slice 1 … else Data Trace 0 … || Data Trace 1 …
- CStatTrace: samplingRate || sizeData || Data …

C. *Interfaz para representaciones gráficas*

Esta interfaz ha sido implementada para las clases que pueden ser representadas gráficamente. Como se puede comprobar en el diagrama de herencias de la Fig. 11, las clases que la implementan son CTrace, CTimeSlice, CTraceSet y CStatTrace.

Dependiendo de cada clase, el tipo de representación gráfica será diferente:

- CTrace: se representa como una gráfica en dos dimensiones. En el eje de abscisas se muestra el tiempo, mientras que en el eje de ordenadas se presenta el voltaje. En la Fig. 12 se muestra la representación de una traza de longitud 1000, habiéndose tomado medidas cada 2 ns.

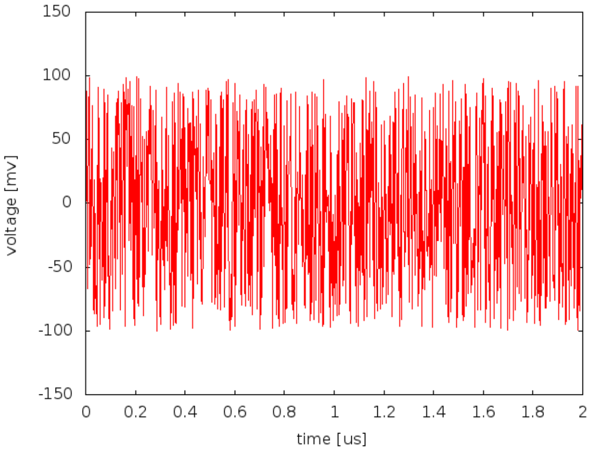


Figura 12. Gráfica proporcionada por CTrace.

- CTimeslice: se muestra como un histograma, por lo que hace referencia al número de ocurrencias de distintos valores del voltaje en un instante de tiempo determinado. La Fig. 13 presenta el histograma de los valores de la traza de la Fig. 12 para el primer instante de tiempo.

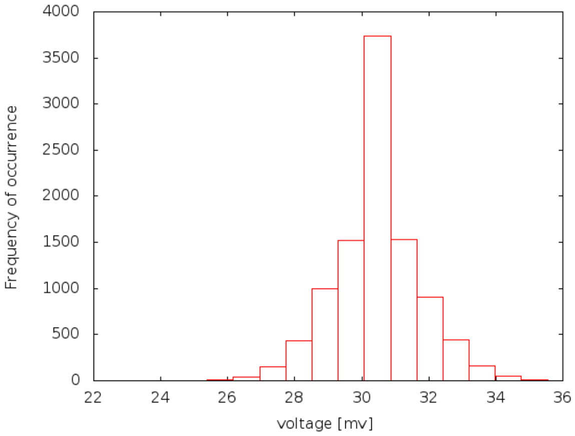


Figura 13. Histograma proporcionado por CTimeSlice.

- CTraceSet: se muestra como una gráfica en tres dimensiones. En realidad, la gráfica aparece en dos dimensiones, estando la tercera dimensión representada por la intensidad del color. Para realizar dicha gráfica se superponen varias trazas representadas como en CTrace. Es claro que cuanto mayor sea el número de trazas que pasen por un determinado punto, más intenso será el color en dicho punto. En la Fig. 14 se puede observar la salida de esta clase para la traza de la Fig. 12 y permite hacerse una idea del valor medio y de la varianza en los distintos instantes de tiempo.

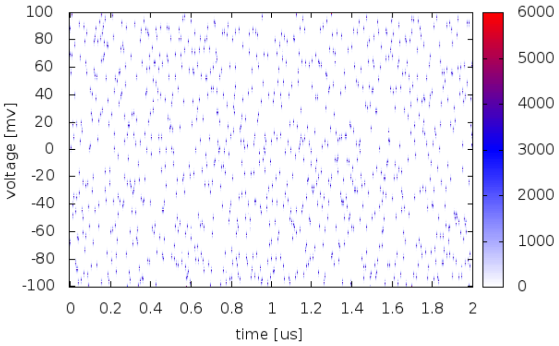


Figura 14. Gráfica proporcionada por CTraceSet.

V. Conclusiones

A partir del diseño y de los resultados experimentales presentados en las secciones III y IV, se ha llegado a las siguientes conclusiones con el fin de avanzar en la implementación de las herramientas software para realizar ataques por canal lateral.

Con relación al uso del método de programación (ver §III-B), la Tabla VI muestra una comparación de estas dos posibles soluciones. Según lo señalado anteriormente, consideramos que, globalmente, desarrollar el código utilizando herencia aporta mayores ventajas que hacerlo utilizando plantillas, ya que la penalización de tiempos es inferior a un 10% y el código sigue una estructura orientada a objetos, lo cual es más apropiado con vistas a desarrollar una caja de herramientas o toolbox.

TABLA VI. COMPARACIÓN ENTRE PLANTILLAS Y HERENCIAS.

	ALTO NIVEL	BAJO NIVEL
PLANTILLAS	Las plantillas tienden a extenderse por la implementación	Solo requiere una clase
HERENCIAS	Permite abstraerse de los detalles de bajo nivel	Es necesario crear una clase por resolución

Como consecuencia, para la implementación final se han elegido las herencias, ya que se genera una codificación más limpia para las clases de alto nivel y son más susceptibles de modificación o ampliación por la comunidad científica. La inclusión de nuevos métodos o valores estadísticos, por ejemplo, es más sencilla de este modo. Además, las clases de bajo nivel son simples y una vez optimizadas es menos probable que se plantee su modificación en el futuro (ver §IV-A).

En lo que hace referencia a la forma de representar las trazas para su optimización a la hora de ser almacenadas en memoria, se ha decidido implementar los dos métodos de representación mencionados en §III-C: uno para tareas de alineación y preprocesado (ordenamiento por filas) y otro para realizar labores estadísticas (ordenamiento por columnas). El paso de un modo de representación a otro se lleva a cabo mediante un método.

Ligado a este tema y con relación al uso del procesador, también se ha evaluado el uso de múltiples hilos (alineamiento concurrente de trazas, cálculos estadísticos concurrentes, etc.) de forma que se maximice el uso de los núcleos del procesador.

Con las clases presentadas y comentadas en §III-D, se tendría resuelta la representación de las medidas tomadas, es decir, se habría determinado cómo trabajar y preprocesar la matriz *T* mencionada en la sección III. Además, se habrían evaluado las distintas alternativas de implementación de modo que se maximice la eficiencia y claridad del código.

Finalmente, señalar que se han desarrollado dos interfaces específicas, una para el almacenamiento en disco (ver §IV-B) y otra para las representaciones gráficas (§IV-C).

AGRADECIMIENTOS

Este trabajo ha sido parcialmente subvencionado por el Ministerio de Economía y Competitividad bajo el proyecto ProCriCiS, TIN2014-55325-C2-1-R, y por la Comunidad de Madrid bajo el proyecto S2013/ICE-3095-CIBERDINE-CM.

REFERENCIAS

[1] O. Aciiçmez, J.P. Seifert, and Ç.K. Koç, “Predicting secret keys via branch prediction”, Lecture Notes in Comput. Sci., vol. 4377, pp. 225-242, 2007.

[2] R. Anderson, M. Bond, J. Clulow, and S. Skorobogatov. “Cryptographic processors-A survey”. Proc. IEEE, vol. 94, 2, pp. 357-369, feb. 2006.

[3] D. Boneh, R.A. DeMillo, and R.J. Lipton, “On the importance of checking cryptographic protocols for faults”, Lecture Notes in Comput. Sci., vol. 1233, pp. 37-51, 1997.

[4] E. Brier, C. Clavier, and F. Olivier, “Correlation power analysis with a leakage model”, Lecture Notes in Comput. Sci., vol. 3156, pp. 16-29, 2004.

[5] B. Chor and R.L. Rivest, “A knapsack-type public key cryptosystem based on arithmetic in finite fields”, IEEE Transactions on Information Theory, vol. 34, pp. 901-909, 1988.

[6] J. Daemen and J. Rijmen, “The Design of Rijndael: AES-The Advanced Encryption Standard”, Springer Verlag, Berlín, Germany, 2002.

[7] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms”, IEEE Transactions on Information Theory, vol. 31, pp. 469-472, 1985.

[8] A. Fúster Sabater, L. Hernández Encinas, A Martín Muñoz, F. Montoya Vitini, and J. Muñoz Masqué, “Criptografía, protección de datos y aplicaciones. Una guía para estudiantes y profesionales”, RA-MA, Madrid, Spain, 2012.

[9] D. Hankerson, A.J. Menezes, and S. Vanstone, “Guide to elliptic curve cryptography”, Springer, New York, NY, USA, 2004.

[10] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, “Side channel cryptanalysis of product ciphers”, J. Comput. Secur., vol. 8, 2,3, pp. 141-158, 2000.

[11] A. Kerckhoffs. “La cryptographie militaire”. Journal des sciences militaires, vol. IX, pp. 1-2, 5-38, 161-191, 1883.

[12] P.C. Kocher, “Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems”, Lecture Notes in Comput. Sci., vol. 1109, pp. 104-113, 1996.

[13] P.C. Kocher, J. Jaffe, and B. Jun, “Differential power analysis”, Lecture Notes in Comput. Sci., vol. 1666, pp. 388-397, 1999.

[14] S. Mangard, E. Oswald, and T. Popp, “Power analysis attacks: Revealing the secrets of smart cards”, Advances in Information Security, Springer Science+Business Media, NY, USA, 2007.

[15] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, “Handbook of Applied Cryptography”, CRC Press, Inc., Boca Raton, FL, USA, 1997.

[16] J.-J. Quisquater and D. Samyde, “ElectroMagnetic Analysis (EMA): Measures and counter-measures for smart cards”, Lecture Notes in Comput. Sci., vol. 2140, pp. 200-210, 2001.

[17] R.L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, Communications of the ACM, vol. 21, 2, pp. 120-126, 1978.

[18] S.P. Skorobogatov, “Semi-invasive attacks-A new approach to hardware security analysis”, PhD thesis, University of Cambridge, Darwin College, UK, 2005. http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-630.pdf.

Alberto Fuentes received his MSc. degree in Computer Science from the KTH Kungliga Tekniska Högskolan, Stockholm, Sweden in 2007. He has been employed by INTA as computer security specialist from 2007 to 2011 evaluating products and systems with Common Criteria and ITSEC standards and from 2011 to 2013 in EADS as common criteria consultant and I+D researcher. He has wide experience in PKI platforms, ad-hoc application or system penetration testing and system vulnerability analysis, and is currently involved in the design and implementation of software tools to carry out side channel attacks to cryptographic devices, pursuing his Ph.D. degree.

Luis Hernández obtained his Graduate in Mathematics at the University of Salamanca (Spain) in 1980 and his Ph.D. in Mathematics at the same university in 1992. He is a researcher at the Department of Information Processing and Cryptography (TIC) at the Institute of Physical and Information Technologies (ITEFI), Spanish National Research Council (CSIC) in Madrid (Spain). He has participated in more than 30 research projects. He is author of 9 books, 9 patents, more than 150 papers, more than 100 contributions to workshops and conferences, he has supervised 3 doctoral thesis and has served as referee for different SCI journals and for many international conferences. His current research interests include Cryptography and Cryptanalysis of public key cryptosystems (RSA, ElGamal and Chor-Rivest), Cryptosystems based on elliptic and hyperelliptic curves, Graphic Cryptography, Pseudorandom number generators, Digital signature schemes, Authentication and Identification protocols, Crypto-Biometry, Secret sharing protocols, Side channel attacks, and Number Theory problems. Dr. Hernández belongs to the International Association for Cryptologic Research (IACR).

Agustín Martín obtained his Graduate in Physics at the Complutense University of Madrid (Spain) in 1988 and his Ph.D. in Physics at the same university in 1995. He is currently a researcher at the Department of Information Processing and Cryptography (TIC) of the Institute of Physical and Information Technologies (ITEFI), Spanish National Research Council (CSIC), in Madrid. He started his research career dealing with the development of numerical methods in electromagnetics to study the interaction of electromagnetic radiation, especially in the radiofrequency range, with different objects as satellites or biological tissues. His latest research interests are focused on the analysis of possible vulnerabilities of radiofrequency identification systems (RFID) and the study of techniques of physical attacks to cryptographic devices. He has published 20 research papers in international scientific journals and is co-author of a book and more than 60 contributions to peer reviewed workshops and conferences. He has participated in 20 research projects and contracts, has taught several courses and seminars, and is co-author of 3 patents. Dr. Martin is member of the International Association for Cryptologic Research (IACR).

Bernardo Alarcos received his Ph.D. degree in telecommunications from the University of Alcalá, Spain in 2004. He received his M.Sc. degree in Telecommunications Engineering from the UPM, Polytechnic University of Madrid, Madrid, Spain in 1979. Since 1999, he has been employed as an Associate Professor at the Alcalá University. The subject of the thesis was about proposal of security architecture in Programmable Networks. He has participated in more than 5 projects developing activities related with the cybersecurity. Last projects that is working on is about Honeynet services using virtualization. Since 2010, he has been a member of Research Institute for the Police Science (IUICP), in the area of Computer forensics.

Content Related to Computing Security on Computer Engineering Degree According to International Professional Certificates

D. G. Rosado, L. E. Sanchez, D. Mellado and E. F. Medina

Abstract— **Companies and professionals are currently demanding increasingly more specialized profiles, and it is therefore desirable for future graduates to have obtained one or more international professional certificates in computing security and auditing, or to at least to have received the preparation required to obtain them. It is therefore of the utmost importance that new studies be focused on professional needs without losing the scientific rigor demanded in engineers. If this objective is to be achieved, it is fundamental that these new study plans be oriented toward facilitating the attainment of these professional certificates. In this paper we establish transversal guidelines for the implementation of content related to computing security in all the subjects, materials and modules of the new degree in Computer Engineering. This will fit perfectly with the material already being taught, will be an enriching element and will allow students to obtain the basic minimum knowledge on security required by any computer engineer from the beginning of their education. The security-related content that is required to be taught during the degree course will additionally be focused on industry and present-day society by means of existing professional security and auditing certificates that will provide future professionals with the knowledge and skills needed as regards security.**

Keywords— **security and auditing, professional certificates, contents, subjects, implementation.**

I. INTRODUCCION

A LO largo del curso 2009-2010 se empezó a implantar el primer curso del Grado en Ingeniería Informática en la Universidad de Castilla-La Mancha. Los detalles del grado, que ha sido adaptado al Espacio Europeo de Educación Superior (EEES) [1, 2], se recogen en una Memoria de Grado, que entre otras cosas ofrece información sobre su organización en módulos, que a su vez contiene materias, y que éstas están formadas por asignaturas, que son definidas en términos de unos descriptores generales, basándose en las recomendaciones de los principales currículos internacionales [3-9]. Para estas asignaturas, se incluye también información sobre las competencias a las que da cuenta, las prácticas docentes, métodos de evaluación, etc., y en todo caso, queda para el momento de la implantación de las asignaturas, el trabajo de definir detalladamente los contenidos de las mismas. De entre todas las asignaturas definidas en el grado, hay varias dedicadas exclusivamente a seguridad y auditoría, y hay otras asignaturas que definen implícitamente aspectos de seguridad ya sea en las competencias a alcanzar o en los descriptores a desarrollar. De cualquiera de las maneras, hay que detallar el contenido de seguridad y auditoría de todas estas asignaturas que se ajusten a sus competencias y descriptores de forma coordinada, y que se acerquen lo máximo posible a las necesidades que demanda la sociedad a través de las principales certificaciones profesionales de seguridad y auditoría [10-12].

Los contenidos de seguridad y auditoría dentro del grado en Ingeniería Informática deben estar perfectamente acoplados y organizados de forma que sea una progresión de conocimientos conforme se vaya avanzando en el grado, tengan una relación directa entre contenidos, estén ajustados a las competencias y objetivos de las asignaturas y estén orientados a las necesidades más demandadas por la sociedad [13, 14].

Las certificaciones profesionales internacionales son un excelente recurso para medir la demanda existente de profesionales en seguridad y auditoría que el mercado requiere [15-17]. Estas certificaciones definen un contenido especializado en seguridad y auditoría que podemos utilizar para incorporarlos en el grado ajustándolos y adaptándolos a las competencias, descriptores y objetivos de cada asignatura del grado.

Por lo tanto, con este trabajo pretendemos definir los contenidos, competencias, objetivos, prácticas docentes, etc. de cada asignatura donde se definan implícita o explícitamente temas de seguridad y auditoría descritos en el plan de estudios del grado, intentando que ese contenido se acerque lo máximo posible a los contenidos y competencias definidas en las principales certificaciones profesionales en seguridad y auditoría, de forma que haya una relación entre los contenidos de seguridad del grado y los contenidos de seguridad exigidos por las certificaciones profesionales que marcan las necesidades del mercado. Esto se debe hacer sin condicionar excesivamente la implantación del grado, pero de modo que se favorezca un acercamiento a estas certificaciones, tanto para que el alumno tenga una mejor formación, como para que opte a conseguir los certificados.

Además, hemos definido un mapa de conocimientos donde se describen el contenido de seguridad de cada asignatura, la relación con las competencias y objetivos de la asignatura y la

D. G. Rosado, Universidad de Castilla La Mancha, Spain, david.grosado@uclm.es

L. E. Sanchez, Universidad de las Fuerzas Armadas, Ecuador. luisenrique@sanchezcrespo.org

D. Mellado, Agencia Tributaria, Spain, damefe@esdebian.org

E. F. Medina, Universidad de Castilla La Mancha, Spain, eduardo.fdezmedina@uclm.es

relación con el contenido de las certificaciones profesionales.

En este trabajo se presentan los resultados conseguidos y elaborados como consecuencia de la ejecución del proyecto que fue presentado en el Primer Taller Iberoamericano de Enseñanza e Innovación Educativa en Seguridad de la Información TIBETS 2011 donde se presentó los objetivos que se plantearon, las actividades a desarrollar, las tareas y pasos a seguir para alcanzar los objetivos, el plan de trabajo seguido y se expuso un conjunto de entregables que se perseguían como resultado de todas las actividades y tareas definidas en el proyecto para conseguir los objetivos previstos. Por tanto, en este trabajo, se presentan dichos entregables que encajan perfectamente con los resultados esperados y que ha sido fruto de la ejecución de todas las actividades y tareas que se definieron y siguiendo la metodología de trabajo presentada en el artículo previo. Estos resultados nos servirán como guía de implantación de contenidos de seguridad en el Grado de Informática.

II. RESUMEN DE OBJETIVOS Y TAREAS

A. Objetivos

El objetivo de este proyecto es establecer una guía transversal para la implantación de contenidos relacionados con la seguridad informática en todas las asignaturas, materias y módulos del nuevo grado de Ingeniería Informática, que encajen perfectamente con las materias que se cursan, que sirvan de elemento enriquecedor y que sirva a los alumnos para adquirir los conocimientos básicos de seguridad mínimos que a cualquier ingeniero informático se le exige desde el principio de su formación. Además, esta integración debe asegurar un camino que les lleve a adquirir la base del conocimiento y materias relacionadas y exigidas en las diferentes certificaciones profesionales de seguridad.

Además, estos contenidos deben estar bien definidos y orquestados con las materias que se cursan, con el curso donde se imparta y con el nivel y competencias exigidas a los alumnos. Por tanto, los contenidos deben repartirse por todo el grado de tal forma que no se repitan entre asignaturas afines, que vayan de niveles básicos a más avanzados conforme se vaya progresando en el grado, y que estén relacionados unos con otros dentro de la misma materia e incluso dentro del mismo curso.

Lo que se pretende es que una vez superado los tres primeros cursos del grado, los alumnos hayan adquirido los conocimientos básicos de seguridad necesarios dentro de todos los ámbitos y campos de la informática (sistemas, software, redes, bases de datos, web, programación, arquitecturas, tecnologías, etc.) antes de iniciar la especialización, donde, dependiendo de ésta, se profundice con más detalles sobre seguridad para un ámbito específico de la informática (con alguna asignatura exclusiva de seguridad), como pueden ser los sistemas software, redes, etc. Así, nos aseguramos que aunque sea experto en, por ejemplo, seguridad software, también haya adquirido los conocimientos básicos para el resto de ámbitos de la informática. Tanto las materias de la especialización como las del resto del grado con contenidos de seguridad deben encajar perfectamente con las materias exigidas por las certificaciones profesionales o al menos sentar las bases y principios para conseguirlas.

De esta forma, un futuro graduado en Informática tendrá las nociones y conocimientos básicos sobre Seguridad, con amplios conocimientos sobre un área en concreto y con las bases necesarias para optar a alguna de las acreditaciones profesionales de seguridad que existen y que son demandadas en la industria de las TIC.

B. Tareas

Para conseguir los objetivos establecidos, se divide el proyecto en 4 actividades bien diferenciadas y que son necesarias para la consecución del proyecto. A estas 4 actividades se le añade una actividad más que es la de Coordinación donde se lleva a cabo las tareas de coordinación, integración y seguimiento del proyecto. Las actividades son:

1. Análisis de las certificaciones profesionales: Se debe realizar un estudio profundo y análisis de las diferentes certificaciones profesionales existentes en temas de Seguridad y del contenido establecido para cada una de ellas. Se deberá seleccionar los aspectos claves del contenido de las certificaciones profesionales e identificar los aspectos comunes a ellas.
2. Análisis de las asignaturas del grado: Se debe realizar un estudio y análisis de todas las asignaturas y materias del grado, así cómo conocer los diferentes descriptores y contenido para cada una de ellas con el fin de poder identificar temas y contenidos afines a las certificaciones profesionales para poder incorporar cierto contenido en la siguiente actividad.
3. Establecer guía de implantación en el grado: En esta actividad se hace el trabajo importante del proyecto de innovación dónde detallamos los contenidos más apropiados a ser incorporados en las asignaturas del grado e indicamos en qué asignaturas deben ser incorporados, estableciendo una guía de implantación de los contenidos de seguridad acorde a las certificaciones profesionales dentro de las asignaturas y/o materias del grado, además de actualizar las competencias, tareas, contenidos, objetivos y planificación de cada asignatura modificada. También se define y detallan coherentemente los contenidos, prácticas, actividades docentes, ejercicios, etc. de las asignaturas involucradas. Además, se establece relaciones entre asignaturas con contenidos de seguridad y auditoría con el fin de evitar repetir contenidos y con el propósito de que el aprendizaje de los contenidos sea de forma continua y progresiva conforme se vaya avanzando en el grado.
4. Definir mapas de conocimiento orientados a certificaciones: En esta actividad se hace un análisis y estudio de los contenidos de seguridad y auditoría incorporados, de las asignaturas involucradas y de las competencias que definen aspectos de seguridad dentro del plan de estudios. Con todo esto se hace un mapa indicando qué contenido es cubierto por cuales asignaturas y si cumplen con las competencias

establecidas en el plan de estudios. Así, sabremos el mapa de asignaturas que cubren la mayor parte de los contenidos definidos por qué certificación, de forma que el alumno sepa las asignaturas que debe cursar para estar mejor preparado para una certificación u otra y qué aspectos no son cubiertos y debería reforzar.

III. RESULTADOS ALCANZADOS

Para conseguir los objetivos establecidos, vamos a dividir el plan de trabajo en las 5 actividades bien diferenciadas y que son necesarias para la consecución de los objetivos que han sido mostradas anteriormente. Mostraremos los resultados alcanzados para cada una de estas actividades. Las actividades y los resultados se explican a continuación.

A. Coordinación

Se establece un plan de seguimiento y coordinación entre todos los involucrados en el proyecto, para establecer plazos de entrega y conseguir resultados que alimenten al resto de actividades, de forma ordenada, coherente y a tiempo.

Resultado: Se ha realizado correctamente y se ha llevado un control permanente de todos los trabajos y de los plazos de ejecución. La coordinación se ha llevado de forma correcta y en todo momento cada uno de los participantes sabía que tenía que hacer, qué tenía que entregar y de qué plazo disponía para su entrega. Además, todos los participantes tenían el informe completo de los trabajos que se iban realizando por el resto de participantes para poder integrar y continuar con lo ya realizado.

Entregable: Se han generado varios informes de seguimiento indicando las tareas a realizar, los involucrados en cada tarea, las fechas de inicio y fin de cada actividad, y los resultados y trabajo realizado hasta el momento (en porcentaje de realización). Esto nos ha servido para conocer el estado real del proyecto y si las actividades se iban realizando conforme la planificación inicial (ver sección IV.A).

B. Análisis de las certificaciones profesionales

Se debe realizar un estudio profundo y análisis de las diferentes certificaciones profesionales existentes en temas de Seguridad y del contenido establecido para cada una de ellas. Se deberá seleccionar los aspectos claves del contenido de las certificaciones profesionales e identificar los aspectos comunes a ellas.

Resultado: Aquí han intervenido participantes con un marcado carácter empresarial y certificados profesionalmente que conocen cuáles son las más importantes certificaciones y más demandadas por las empresas, e incluso tienen amplio conocimiento sobre los contenidos de muchas de ellas, lo cual nos ha permitido establecer discusiones sobre el contenido más interesante y que mejor encaja con el plan de estudios y con la situación actual del mercado y de la sociedad.

Entregable: Se ha elaborado una lista de los contenidos más interesantes y adecuados en temas de seguridad y auditoría, extraídos de los contenidos exigidos por las certificaciones profesionales. Este documento ha sido revisado por todos los participantes y ha sido producto de muchos cambios y modificaciones hasta encontrar un consenso entre todos (ver sección IV.B).

C. Análisis de las asignaturas del grado

Se debe realizar un estudio y análisis de todas las asignaturas y materias del grado, así cómo conocer los diferentes descriptores y contenido para cada una de ellas con el fin de poder identificar temas y contenidos afines a las certificaciones profesionales para poder incorporar cierto contenido en la siguiente actividad.

Resultado: En esta actividad contamos con profesores que participaron en la creación y elaboración del actual plan de estudios del grado de informática, por lo que el conocimiento de las competencias, materias y asignaturas es total, lo cual facilitó la labor de analizar en detalle todas las competencias relacionadas con la seguridad, y las asignaturas junto con los descriptores para extraer toda la información necesaria para tomar la decisión de qué asignaturas son más apropiadas, por tener una relación directa o indirecta con aspectos de seguridad, para que pueda incorporarse nuevos contenidos de seguridad y describirlas en detalle.

Entregable: El resultado de esta actividad ha sido una lista de asignaturas del grado junto con sus competencias, orientadas o relacionadas de alguna forma con la seguridad y auditoría, que sirvan como candidatas para poder incorporar los nuevos contenidos de seguridad extraídos de la actividad anterior. Esta lista también ha sido consensuada con el resto de participantes expertos en distintas disciplinas y que imparten asignaturas de diversa índole (ver sección IV.C).

D. Establecer guía de implantación en el grado

En esta actividad se hace el trabajo importante dónde detallamos los contenidos más apropiados a ser incorporados en las asignaturas del grado e indicamos en qué asignaturas deben ser incorporados, estableciendo una guía de implantación de los contenidos de seguridad acorde a las certificaciones profesionales dentro de las asignaturas y/o materias del grado, además de actualizar las competencias, tareas, contenidos, objetivos y planificación de cada asignatura modificada. También se define y detallan coherentemente los contenidos, prácticas, actividades docentes, ejercicios, etc. de las asignaturas involucradas. Además, se establece relaciones entre asignaturas con contenidos de seguridad y auditoría con el fin de evitar repetir contenidos y con el propósito de que el aprendizaje de los contenidos sea de forma continua y progresiva conforme se vaya avanzando en el grado.

Resultado: Sin duda esta actividad ha sido la más complicada de realizar y la que ha supuesto un esfuerzo extraordinario por parte de todos para su consecución. Aunque tenemos, por un lado los contenidos de seguridad a integrar, y por otro lado, las asignaturas candidatas, no es fácil integrar los contenidos en asignaturas, y requiere un análisis profundo y detallado tanto de los contenidos como de los descriptores de las asignaturas, para que dicha integración sea correcta. Además, dicha integración no es directa, sino que hay que estudiar en qué nivel de detalle esos contenidos deben ser integrados, si se integra como un todo o se divide por partes distribuidos en cierto número de asignaturas de una misma materia, si se integra de forma que sea una evolución sobre algún tema concreto a lo largo de los cursos, si se define una trazabilidad de contenidos relacionados con contenidos afines pero adaptados al nivel de exigencia y competencias exigido, y un largo etcétera que se ha tenido en cuenta para realizar dicha integración.

A pesar de la dificultad de esta actividad, todos los participantes han colaborado activamente en el rol de encargados de sus asignaturas, y otro grupo de participantes más expertos en temas de seguridad (algunos de ellos certificados profesionalmente), han sabido cooperar y trabajar de forma coordinada para que la integración de contenidos de seguridad en asignaturas se haga de la mejor forma posible, de forma no invasiva para que todo encaje a la perfección sin modificar los contenidos de las asignaturas existentes.

Entregable: Este es el entregable más importante y de mayor valor de este proyecto de innovación docente porque es el resultado final al que queríamos llegar. Este entregable, que se describirá en la sección III, resume los entregables de las actividades anteriores, presentado la lista de contenidos de seguridad seleccionados de las certificaciones profesionales, la lista de las asignaturas que han sido seleccionadas como candidatas para incorporar contenidos de seguridad, y también se muestra la relación entre ambas dejando claro qué contenidos en qué asignaturas han sido integradas y de qué forma (ver sección IV.D).

E. Definir mapas de conocimiento orientados a certificaciones

En esta actividad se hace un análisis y estudio de los contenidos de seguridad y auditoría incorporados, de las asignaturas involucradas y de las competencias que definen aspectos de seguridad dentro del plan de estudios. Con todo esto se hace un mapa indicando qué contenido es cubierto por cuales asignaturas y si cumplen con las competencias establecidas en el plan de estudios. Así, sabremos el mapa de asignaturas que cubren la mayor parte de los contenidos definidos por qué certificación, de forma que el alumno sepa las asignaturas que debe cursar para estar mejor preparado para una certificación u otra y qué aspectos no son cubiertos y debería reforzar.

Resultado: El propósito de esta actividad final es la de definir un mapa de conocimientos a partir de los contenidos de seguridad y auditoría que han sido integrados en las distintas materias y asignaturas a lo largo de todo el plan de estudios. El objetivo es indicar qué conjunto de asignaturas tienen ciertos contenidos relacionados de seguridad que juntos establecen un conocimiento completo de algún aspecto o ámbito de seguridad. Además, también se define un camino de asignaturas donde, a partir de los contenidos de seguridad incorporados, facilitan o se acercan más a una certificación profesional u otra, o con el que se consigue un conocimiento más detallado en temas de seguridad para una disciplina determinada dentro de la seguridad y auditoría.

Entregable: Este entregable es una especie de tabla con todo el plan de estudios (cursos, materias y asignaturas) y donde cada casilla indica el contenido de seguridad que ha sido implantado. Así, podemos establecer qué asignaturas tienen relación con respecto a los temas de seguridad que se abordan, para finalmente, poder indicar el camino o conjunto de asignaturas que el alumno puede cursar para adquirir un conocimiento más o menos profundo en aspectos de seguridad que más se ajustan a una u otra certificación profesional (ver sección IV.E).

IV. RESULTADOS DE LOS ENTREGABLES

A. Informes de seguimiento.

Son documentos donde se indica cuál ha sido la evolución del proyecto, quienes son los implicados, qué se debe presentar y qué queda por hacer, aclarando los plazos y la coordinación entre todos, y definiendo los hitos futuros. Además, también sirve para aclarar dudas y ver el estado actual del proyecto.

B. Lista de contenidos a incorporar.

Es un documento donde se definen qué certificaciones profesionales de seguridad y auditoría han sido seleccionadas, junto con los contenidos más adecuados y más interesantes de cada una de las certificaciones seleccionadas. Esta lista de contenidos no debe ser muy amplia ya que de lo contrario, imposibilitaría la incorporación en el plan de estudios.

Existen numerosas entidades y organismos acreditadores en todo el mundo. La lista de certificaciones disponibles en el mercado es también inagotable. Por ello, hemos seleccionado algunas de las más importantes, prestigiosas, avaladas por una larga experiencia y reconocidas por el sector.

- CISA (Certified Information System Auditor) y CISM (Certified Information Security Manager) por ISACA.
- CISSP (Certified Information System Security Manager) por (ISC)^2
- GIAC (Global Information Security Assurance Certification)
- CIA (Certified Internal Auditor) por el Institute of Internal Auditors.
- CIPP (The Certified Information Privacy Professional) por el the International Association of Privacy Professionals.
- CPP (Certified Protection Professional) por ASIS International.
- CCSP (Cisco Certified Security Professional) por CISCO Systems.

De entre toda esta lista de certificaciones, se han extraído los contenidos que se repiten en todas debido a la importancia que tienen esos aspectos de seguridad, y algún otro contenido que hemos considerado importante. Los contenidos, descripciones y descriptores se describen en la Tabla I.

TABLA I. CONTENIDO, DESCRIPCIÓN Y DESCRIPTORES DE LAS PRINCIPALES CERTIFICACIONES PROFESIONALES.

ID	Contenido	Descripción	Descriptores
AUD	Auditoría de sistemas de información	Para brindar servicios de auditoría de sistemas acorde a las normas, guías, estándares y mejores prácticas para apoyar a la organización a asegurar que sus sistemas y la tecnología de información están protegidos y controlados	Estándares, directrices, y herramientas. Controles. Planificación y gestión de proyectos de auditoría. Leyes y regulaciones aplicables. Recopilación de evidencia. Muestreo, reporte y comunicación. Sistemas y marcos de aseguramiento de la calidad de la auditoría.
GOB	Gobierno y gestión de TI	Para proporcionar aseguramiento de que la organización tiene la estructura, las políticas, los mecanismos de reporte y las prácticas de monitoreo necesarias para cumplir los requisitos del gobierno corporativo y la gestión de las TI	Metas y objetivos del negocio. Relación entre la seguridad de la información y las funciones del negocio. Alcance y los estatutos del gobierno de seguridad de la información. Estrategias de planificación presupuestaria. Desarrollo de casos de negocio. Requerimientos regulatorios y su impacto potencial. Gestión de responsabilidad común y relaciones con terceros y su impacto. Roles, responsabilidades y estructuras organizacionales. Lazos entre políticas y objetivos de negocio de la empresa. Estándares. Métodos centralizados y distribuidos para coordinar actividades de seguridad de la información y para establecer canales de comunicación y notificación en toda la organización.
RIE	Gestión de riesgos de la información	Identificar y gestionar los riesgos de seguridad de la información para lograr los objetivos del negocio.	Riesgos, vulnerabilidades y exposiciones de la información. Metodologías de evaluación y análisis de riesgos. Controles y contramedidas. Estrategias de mitigación de riesgos. Técnicas de análisis costo-beneficio. Principios y prácticas de gestión de riegos basados en el ciclo de vida.
PRO	Desarrollo del programa de seguridad de información	Crear y mantener un programa para implementar la estrategia de seguridad de información	Tipos de actividades. Planificación, diseño, desarrollo, prueba e implementación de los controles de seguridad de la información. Alineación de requisitos de seguridad. Arquitecturas de seguridad. Tecnologías y controles de seguridad. Desarrollo de políticas de seguridad. Cultura y comportamiento. Métodos para desarrollar, implementar, comunicar y mantener políticas, estándares, procedimientos, guías y otros documentos. Diseño, desarrollo e implementación de métricas de seguridad. Certificación y acreditación del cumplimiento. Métodos de seguimiento, medición y sostenimiento.
ARQ	Arquitectura y Modelos de Seguridad	Conceptos, principios, estructuras y estándares empleados para diseñar, monitorizar y asegurar sistemas, equipos, redes, aplicaciones y controles usados para reforzar los diversos niveles de la disponibilidad, integridad y confidencialidad	Conceptos de control y seguridad. Modelos de seguridad. Criterios de evaluación. Seguridad en entornos cliente/servidor y host. Seguridad y arquitectura de redes. Arquitectura de la seguridad IP.
CON	Sistemas y Metodología de Control de Acceso	Conjunto de mecanismos que permiten crear una arquitectura segura para proteger los activos de los SI	Conceptos y tópicos. Identificación y autenticación. Equipo de e-security. Single sign-on. Acceso centralizado / descentralizado / distribuido. Metodologías de control. Monitorización y tecnologías de control de acceso. Modelos de control de acceso (DAC, MAC, RBAC). Mejore prácticas (denegación implícita, menos privilegio, separación de responsabilidades, rotación de trabajo). Fundamentos biométricos. Claves
DAP	Seguridad en el Desarrollo de Aplicaciones y Sistemas	Define el entorno donde se diseña y desarrolla el software y engloba la importancia crítica del software dentro de la seguridad de los SI	Definiciones. Amenazas y metas de seguridad. Ciclo de vida. Arquitecturas seguras. Control de cambios. Medidas de seguridad y desarrollo de aplicaciones. Bases de datos y data warehousing. Knowledge-based systems. herramientas y técnicas de monitoreo
CRI	Criptografía	Los principios, medios y métodos de protección de la información para asegurar su integridad, confidencialidad y autenticidad	Historia y definiciones. Aplicaciones y usos de la criptografía. Protocolos y estándares. Tecnologías básicas. Sistemas de encriptación (AES, DES, PGP, RSA). Criptografía simétrica / asimétrica. Firma digital. Seguridad en el correo electrónico e Internet empleando encriptación. Gestión de claves. Public key infrastructure (PKI). VPN. IPSec. Ataques y criptoanálisis. Cuestiones legales en la exportación de criptografía
FIS	Seguridad Física	Técnicas de protección de instalaciones, incluyendo los recursos de los SI	Gestión de las instalaciones. Seguridad del personal. Defensa en profundidad. Controles físicos

INT	Seguridad en Internet, Redes y Telecomunicaciones	Incluye los dispositivos de la red, los métodos de transmisión, formatos de transporte, medidas de seguridad y autenticación	Gestión de la seguridad en la comunicaciones. Protocolos de red. Identificación y autenticación. Comunicación de datos. Seguridad de Internet y Web. Métodos de ataque. Seguridad en Multimedia. Firewalls. VPN. Seguridad de Perímetros
NEG	Recuperación ante Desastres y Planificación de la Continuidad del Negocio	Planificar, desarrollar y gestionar la capacidad para detectar, responder y recuperarse de incidentes de seguridad de información. Dirige la preservación del negocio en el caso de producirse situaciones de parada para la restauración de las operaciones	Conceptos de recuperación ante desastres y de negocio. Procesos de planificación de la recuperación. Gestión del software. Análisis de Vulnerabilidades. Desarrollo, mantenimiento y testing de planes. Prevención de desastres. Requisitos forenses. Prácticas de revisión e investigación. Cuantificación de daños, costos y otros impactos empresariales
LEY	Leyes, investigaciones y Ética	Engloba las leyes y regulaciones de los crímenes informáticos, las técnicas y medidas de investigación, recuperación de evidencias y códigos éticos	Leyes y regulaciones. Conducción de investigaciones. Ética en la seguridad de la información. Código ético

TABLA II. LISTA DE ASIGNATURAS Y DESCRIPTORES.

Asignaturas	Descriptores
Administración de Bases de Datos	Introducción a la administración de bases de datos. Diccionarios y repositorios de datos. Seguridad de bases de datos. Control de concurrencia y recuperación. Optimización y ajuste.
Análisis Forense Informático	Evidencias digitales. Recolección y manejo de evidencias. Detección de intrusiones informáticas. Redes trampa. Normativa legal y técnica en el tratamiento de evidencias. Herramientas de análisis forense.
Aplicaciones Distribuidas en Internet	Introducción a los modelos arquitecturales de bajo acoplamiento. Desarrollo de sistemas distribuidos. Plataformas de desarrollo basadas en estándares de mensajería y en paso de mensajes. Aspectos de escalabilidad y rendimiento en aplicaciones distribuidas en Internet.
Arquitectura de Computadores	Introducción a los tipos de arquitecturas y modelos de programación. Paralelismo a nivel de instrucción: conceptos y métodos para su explotación. Técnicas de optimización del software.
Aspectos Profesionales de la Informática	Gestión de Proyectos Informáticos. Aspectos jurídicos del uso de las TIC. Legislación y normativa. Propiedad intelectual. Firma electrónica. Ética y responsabilidad profesional. Delitos informáticos. Técnicas de comunicación efectivas para la elaboración del pliego de condiciones.
Auditoria en Sistemas de Información	Control interno y auditoría de sistemas de información. Metodologías de evaluación, control interno y auditoría. Departamento de auditoría. Entorno jurídico de la auditoría. Principales áreas de auditoría de sistemas de información. Herramientas para la auditoría.
Bases de Datos	Ficheros. Conceptos básicos de bases de datos. Sistemas de gestión de bases de datos. Modelos de datos. Modelo relacional. Estándar SQL. Programación y uso de bases de datos. Acceso programático a bases de datos. Introducción a otros modelos de datos.
Bases de Datos Avanzadas	Necesidades de información de las organizaciones. Modelado conceptual y lógico de datos. Bases de datos avanzadas: objeto-relacionales, orientadas a objeto, XML, web, multimedia, distribuidas, librerías digitales. Bases de datos y grid. Bases de datos y computación en nube. Procesamiento y gestión de transacciones.
Cálculo y Métodos Numéricos	Nociones básicas de los distintos conjuntos numéricos. Cálculo diferencial. Desarrollo de Taylor. Optimización. Cálculo integral y sus aplicaciones. Algunos métodos numéricos. Algorítmica numérica.
Comercio electrónico	Modelos de comercio electrónico. Seguridad en el comercio electrónico. Legislación. Transacciones electrónicas. Medios de pago electrónico. Lenguajes para el comercio electrónico Modelos de cliente.
Desarrollo de Bases de Datos	Requisitos de Datos. Diseño conceptual. Diseño lógico. Diseño Físico. Seguridad en BBDD. Diseño avanzado de datos: Objeto-relacional, XML-semiestructurado, multidimensional.
Desarrollo de Sistemas Web	Desarrollo de aplicaciones para la Web. Técnicas de modelado para la Web. Modelado de la interacción y la navegación. Arquitecturas para sistemas basados en web. Servidores web. Sistemas de gestión de contenidos. Dominios de aplicación Web.
Diseño y Gestión de Redes	Conceptos básicos sobre planificación de redes. Cableado estructurado de red. Diseño de LANs. Monitorización de una red. Control de una red. Protocolos de mantenimiento. Protocolos de monitorización. Herramientas de gestión de red.
Gestión de proyectos Software	Planificación estratégica. Planificación de proyectos software. Estimación. Seguimiento y control de proyectos software. Gestión de riesgos. Herramientas de gestión de proyectos.
Gestión de Sistemas de Información	El sistema de información y el negocio, adquisición, despliegue y gestión de soluciones y servicios TIC, técnicas avanzadas de manejo y recuperación de información, bases de datos de propósito especial (documentales, multimedia, espacio-temporales), sistemas de soporte a la decisión, almacenes de datos, minería de datos e inteligencia de negocio
Gestión y Administración de redes	Introducción a los Sistemas de mantenimiento y gestión de Red. Monitorización de una red. Control de una red. Protocolos de mantenimiento. Protocolos de monitorización. Herramientas de gestión de red. Gestión de la calidad de servicio
Ingeniería de Negocio	Requisitos organizacionales. Modelado de empresas. Procesos de negocio. Modelado y gestión de procesos de negocio. Desarrollo de software dirigido por procesos de negocio. Sistemas para toma de decisiones. Procesamiento OLAP. Procesos ETL. Minería de datos. Herramientas de inteligencia de negocio

Ingeniería de Requisitos	Fundamentos de análisis del software. Requisitos software. Tipos de requisitos. Elicitación, análisis, especificación y validación de requisitos software. Análisis orientado a objetos. Notaciones avanzadas. Herramientas de gestión de requisitos. Métodos de gestión de requisitos.
Ingeniería del Software II	Ciclos de vida del software. Procesos de ingeniería del software. Calidad de los productos y procesos del software. Verificación y validación del software. Pruebas del software. Mantenimiento del software. Gestión de configuración del software. Metodologías de desarrollo de software
Multimedia	Contenidos y composición multimedia, estándares para contenidos digitales, técnicas y estándares de compresión multimedia, distribución de contenidos multimedia. Sistemas y aplicaciones multimedia
Redes de Computadores II	Tecnologías de red. Interconexión de dispositivos de red. Protocolos de encaminamiento en Internet. Movilidad y multidifusión. Capa de transporte en TCP/IP. Diseño y programación de aplicaciones en red. Capa de aplicación en TCP/IP: servicios estándares más comunes. Conceptos básicos de la gestión de redes. Conceptos básicos de seguridad en redes.
Redes y Servicios Móviles	Características y diseño de aplicaciones sobre dispositivos móviles. Casos de estudio de plataformas comerciales. Desarrollo de sistemas basados en redes de sensores. Desarrollo de servicios para teléfonos móviles.
Seguridad de los Sistemas Informáticos	Políticas, técnicas y mecanismos de seguridad en los sistemas informáticos. Legislación y estándares de seguridad en las TIC. Vulnerabilidades de seguridad, análisis y clasificación de ataques, planes de seguridad y contingencia.
Seguridad de Sistemas Software	Fundamentos de seguridad. Seguridad organizativa. Requisitos de seguridad. Seguridad en desarrollo de software. Seguridad de sistemas de información. Riesgos de seguridad. Servicios de seguridad. Gestión de seguridad. Certificación, normas y estándares para la seguridad.
Seguridad en redes	Principios de seguridad en redes. Cortafuegos. Redes Privadas Virtuales. Acceso Remoto Seguro. Seguridad en capa de transporte. Seguridad en capa de Aplicación.
Sistemas Distribuidos	Conceptos fundamentales de sistemas distribuidos. Comunicación de procesos y grupos de procesos distribuidos. Objetos distribuidos e invocación remota. Sincronización distribuida. Transacciones y control de concurrencia. Programación de aplicaciones distribuidas.
Sistemas Operativos I	Características, funciones y estructura de los sistemas operativos: procesos, planificación, concurrencia, memoria, entrada y salida, sistemas de ficheros. Entorno de programación del sistema. Nociones de administración de sistemas.
Tecnologías y Sistemas Web	Plataformas web. Arquitecturas de sistemas web. Protocolos y estándares web. Programación de aplicaciones web. Tecnologías de acceso a bases de datos. Tecnologías avanzadas. Seguridad.

C. Lista de asignaturas candidatas.

Este documento define una lista de asignaturas, extraídas del plan de estudios del grado de Ingeniería Informática, que son las más adecuadas, por tener relación con algún aspecto de seguridad, para que se puedan incorporar nuevos contenidos de seguridad en su temario, ya sean como temas, ejercicios, casos prácticos, o prácticas de laboratorio. Para la selección de estas asignaturas se tiene en cuenta tanto los descriptores de cada asignatura, como las competencias con las que están relacionadas, además del visto bueno de la mayoría de tutores de las asignaturas involucradas. Por tanto, la lista de asignaturas candidatas servirá de entrada para el siguiente entregable y se pueden ver en la primera columna de la Tabla III.

~~D.~~A. Guía de Implantación.

Una vez que tenemos identificadas las asignaturas a las que se le puede incorporar ciertos contenidos de seguridad, los cuales son extraídos de las principales certificaciones profesionales, queda asignar o relacionar esos contenidos con esas asignaturas, de forma que se tenga claro dónde encaja esos contenidos en ese conjunto de asignaturas.

En la Tabla III podemos ver una visión general de las asignaturas del grado donde podemos incorporar, definir y planificar contenidos de seguridad y auditoría dentro de la propia guía docente de las asignaturas, ya sean como temas, ejercicios, ejemplos, casos prácticos o material complementario de la asignatura. No se detalla el contenido exacto que debe aparecer en cada asignatura, sólo una visión general del posible contenido, referido a una temática específica extraído de la lista de contenidos de seguridad, que pudiera ser incorporado a dicha asignatura. La definición detallada queda para un trabajo posterior, en el momento de definir la guía docente de la asignatura. Ahora sólo se ofrece unos descriptores a tener en cuenta en temas de seguridad y auditoría. Después del análisis y estudio llevado a cabo en esta actividad, hemos llegado al resultado mostrado en la Tabla III.

Aquí podemos ver cómo hemos relacionado las asignaturas candidatas generadas en la actividad III.C, con los posibles contenidos definidos en las principales certificaciones profesionales en seguridad y auditoría definidas en la actividad III.B.

De esta forma, hemos identificado qué contenido más apropiado de las certificaciones profesionales puede ser integrado y encajado en las diferentes asignaturas del Grado. Queda el siguiente paso que es una vez conocido el contenido que mejor encaja con la asignatura a partir de sus competencias y descriptores, falta definir dicho contenido y la guía docente de la asignatura con los objetivos, competencias, el temario, ejercicios, etc., que se deja para trabajo futuro conforme vayamos avanzamos en la implantación del Grado.

TABLA III. RESUMEN DE LA GUÍA DE IMPLANTACIÓN.

	ARQ	AUD	CON	CRI	DAP	FIS	GOB	INT	LEY	NEG	PRO	RIE
Administración de Bases de Datos					■	■			■	■		
Análisis Forense Informático		■							■	■		
Aplicaciones Distribuidas en Internet	■		■		■			■				
Arquitectura de Computadores	■											
Aspectos Profesionales de la Informática							■		■			■
Auditoria en Sistemas de Información		■							■		■	
Bases de Datos			■		■							
Bases de Datos Avanzadas			■		■	■			■	■		
Cálculo y Métodos Numéricos				■								
Comercio electrónico					■		■	■	■			
Desarrollo de Bases de Datos			■		■							
Desarrollo de Sistemas Web	■		■		■			■				
Diseño y Gestión de Redes	■			■				■				
Gestión de proyectos Software		■					■			■	■	■
Gestión de Sistemas de Información					■		■			■	■	■
Gestión y Administración de redes	■							■				
Ingeniería de Negocio							■			■	■	
Ingeniería de Requisitos					■						■	■
Ingeniería del Software II					■					■		■
Multimedia				■								
Redes de Computadores II	■		■					■				
Redes y Servicios Móviles	■		■	■				■				
Seguridad de los Sistemas Informáticos							■		■	■		■
Seguridad de Sistemas Software		■			■		■		■	■	■	■
Seguridad en redes	■		■	■				■				
Sistemas Distribuidos	■				■			■				
Sistemas Operativos I			■								■	
Tecnologías y Sistemas Web	■		■	■	■			■				

ARQ: Arquitectura y Modelos de Seguridad
AUD: Auditoría de sistemas de información
CON: Sistemas y Metodología de Control de Acceso
CRI: Criptografía
DAP: Seguridad en el Desarrollo de Aplicaciones y Sistemas
FIS: Seguridad Física
GOB: Gobierno y gestión de TI
INT: Seguridad en Internet, Redes y Telecomunicaciones
LEY: Leyes, investigaciones y Ética
NEG: Recuperación ante Desastres y Planificación de la Continuidad del Negocio
PRO: Desarrollo del programa de seguridad de información
RIE: Gestión de riesgos de la información

A. Mapa de certificaciones.

Este último material generado es un esquema o mapa de todo el plan de estudios del grado en Ingeniería Informática, y las relaciones en cuanto a contenidos de seguridad que se imparten en esas asignaturas con respecto a las certificaciones profesionales más destacadas. Aquí se muestran las asignaturas del grado y los posibles caminos hacia posibles certificaciones profesionales indicando qué conjunto de asignaturas pueden ser cursadas durante el grado para aproximarse al contenido exigido por las principales certificaciones profesionales.

La Tabla IV muestra las asignaturas de dos de las 4 intensificaciones para el Grado en Informática de la UCLM, junto con las optativas, y la relación existente con el contenido de las certificaciones profesionales seleccionadas. Debido a restricciones de espacio, la Tabla IV sólo muestra las dos intensificaciones (Ingeniería el Software (IS) y Tecnologías de Información (TI)) que tienen más relación con temas de seguridad, aunque el trabajo completo ha considerado todas las asignaturas de las 4 intensificaciones que no han sido mostradas aquí. Para el resto de intensificaciones (Ingeniería de Computadores y Computación) hay muy poco contenido en las asignaturas que forman esas intensificaciones que tengan relación con temas de seguridad identificados en las certificaciones profesionales.

De esta forma, podemos crear los diferentes mapas para las diferentes certificaciones profesionales, dando a conocer el conjunto de asignaturas y optativas que mejor encajan y que te dan una aproximación más completa a una determinada certificación profesional. Así, por ejemplo, quien esté interesado en tener mayor conocimiento sobre los contenidos de la certificación CISA, debe saber que tiene que cursar preferentemente la intensificación de Ingeniería de Software (IS) y elegir un conjunto concreto de optativas. Si lo que está interesado es en obtener conocimientos más relacionados con la certificación CISSP, deberá cursar la intensificación de Tecnologías de la Información (TI) y las cuatro asignaturas optativas que se indican en la Tabla IV.

TABLA IV. MAPA DE CERTIFICACIONES.

	Asignaturas	CISA	CISM	CISSP	GIAC
IS	Ingeniería de Requisitos	✗	✗		✗
	Diseño de Software	✗			
	Procesos de Ingeniería del Software				
	Calidad de Sistemas Software				
	Gestión de Proyectos Software	✗	✗		✗
	Desarrollo de Bases de Datos	✗		✗	✗
	Sistemas de Información Empresariales	✗			
	Seguridad de Sistemas Software	✗	✗		✗
TI	Integración de Sistemas Informáticos				
	Interacción Persona-Ordenador II				
	Diseño y Gestión de Redes			✗	✗
	Gestión de Sistemas de Información	✗	✗		✗
	Tecnologías y Sistemas Web	✗		✗	✗
	Comercio Electrónico	✗	✗	✗	✗
	Multimedia				✗
	Seguridad en Sistemas Informáticos	✗	✗		✗
Opt.	Ingeniería de Negocio	✗	✗		
	Bases de Datos Avanzadas	✗	✗	✗	✗
	Auditoría de Sistemas de Información	✗	✗		
	Administración de Bases de Datos	✗	✗		
	Desarrollo de Sistemas Web	✗		✗	
	Análisis Forense Informático	✗	✗		
	Redes y Servicios Móviles			✗	
	Aplicaciones Distribuidas en Internet	✗		✗	✗

V. CONCLUSIONES.

Este trabajo es el resultado de la ejecución de un proyecto de innovación docente donde se pretende plasmar la relación de temas de seguridad y auditoría entre los contenidos de las asignaturas del grado y los contenidos de las principales certificaciones profesionales de seguridad y auditoría que miden la demanda existente de profesionales en seguridad y auditoría que el mercado requiere, lo cual le da un aspecto más profesional, más orientado a las necesidades y más especializado en un ámbito concreto e importante de la Informática como es el de la Seguridad.

Con los resultados conseguidos, podemos tener la certeza de comprobar si el contenido más apropiado, que se ajustan a las necesidades reales de las empresas, ha sido debidamente incorporado e implementado en el grado, y si con dicha incorporación se cubre con alto porcentaje de competencias especificadas en el plan de estudios para las asignaturas. Además, con esta asociación, se puede extraer información de los puntos débiles en cuanto a contenidos y lo que el alumno tendría que reforzar para optar a alguna de las acreditaciones profesionales en seguridad y auditoría.

Para concluir, tan sólo me gustaría mencionar que todos los resultados obtenidos en este proyecto se comenzarán a aplicar en los cursos futuros, en el momento de empezar a definir las guías docentes de las asignaturas implicadas, principalmente en las asignaturas de las distintas intensificaciones y optativas, que son las más específicas y dónde tienen más cabida aspectos específicos de seguridad.

AGRADECIMIENTOS

Esta trabajo es el resultado de un proyecto de innovación docente titulado "Implantación y Orquestación de los Contenidos de Seguridad en el Grado en Ingeniería Informática que Favorezca en Acercamiento a las Principales Certificaciones Profesionales de Seguridad y Auditoría", concedido dentro de la 6ª Convocatoria de Ayudas para Proyectos de Innovación Docentes promovidos por el Vicerrectorado de Ordenación Académica y Formación Permanente de la Universidad de Castilla-la Mancha. También es parte de los siguientes proyectos: SIGMA-CC (TIN2012-36904) and GEODAS (TIN2012-37493-C03-01) financiados por el "Ministerio de Economía y Competitividad" (España). Y por el proyecto PROMETEO financiado por la Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) del Gobierno de Ecuador.

REFERENCIAS

[1] EEES. Espacio Europeo de Educación Superior. Available from: http://www.eees.es/.

[2] ECTS. European Credit Transfer System. Available from: http://www.ects.es/.

[3] ACM/AIS, MSIS 2006: Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems. 2006.

[4] ACM/IEEE, Computer Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering. 2004.

[5] ACM/IEEE, Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. 2004.

[6] ACM/IEEE, Computing Curricula 2005. The Overview Report. 2005.

[7] ACM/IEEE, Computer Science Curriculum 2008. 2008.

[8] ACM/IEEE, Information Technology 2008. Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. 2008.

[9] ISACA, ISACA Model Curriculum for Information Security Management. 2008.

[10] Seidman, S., The Emergence of Software Engineering Professionalism, in IFIP International Federation for Information Processing. 2008, Springer.

[11] Crowley, E., Information system security curricula development, in 4th conference on Information technology curriculum. 2003. p. 249-255.

[12] Suarez, B. and E. Tovar, Accreditation in engineering, in Plenary Sessions of Int. Conf. Engineering Computer Education 2005 (ICECE05). 2006.

[13] Seidman, S.B., Software Engineering Certification Schemes in Computer. 2008.

[14] Batchman, T. and E. Tovar, Advantages and challenges which the accreditation process with ABET offers to engineering and computer science programs. Perspective of the engineering college, in Plenary Sessions of Int. Conf. Engineering Computer Education 2005 (ICECE05). 2005.

[15] (ISC)2. The International Information Systems Security Certification Consortium, Inc., (ISC). Available from: http://www.isc2.org/.

[16] GIAC. GIAC –Global Information Security Assurance Certification. Available from: www.giac.org.

[17] ISACA. Information Systems Audit and Control Association. Available from: www.isaca.org.

David G. Rosado has an MSc and PhD. in Computer Science from the University of Málaga (Spain) and from the University of Castilla-La Mancha (Spain), respectively. His research activities are focused on security for Information Systems and Cloud Computing. He has published several papers in national and international conferences on these subjects, and he is co-editor of a book and chapter books. Author of several manuscripts in national and international journals (Information Software Technology, System Architecture, Network and Computer Applications, etc.). He is member of Program Committee of several conferences and workshops nationals and internationals such as ICEIS, ICCGI, CISIS, SBP, IAS, SDM, SECRYPT, COSE and international journals such as Internet Research, JNCA, KNOSYS, JKSU, and so on. He is a member of the GSyA research group of the Information Systems and Technologies Department at the University of Castilla-La Mancha, in Ciudad Real, Spain.

Luis Enrique Sánchez is PhD and MsC in Computer Science and is an Professor at the Universidad de las Fuerzas Armadas (ESPE) of Latacunga (Ecuador), MSc in Information Systems Audit from the Polytechnic University of Madrid, and Certified Information System Auditor by ISACA. He is the Director of Professional Services and R&D departments of the company Sicaman Nuevas Tecnologías S.L. COIICLM board or committee member and responsible for the professional services committee. His research activities are management security system, security metrics, data mining, data cleaning, and business intelligence. He participates in the GSyA research group of the Department of Computer Science at the University of Castilla-LaMancha, in Ciudad Real (Spain). He belongs to various professional and research associations (COIICLM, ATI, ASIA, ISACA, eSEC, INTECO, etc)

Daniel Mellado holds a PhD and MSc in Computer Science from the Castilla- La Mancha University (Spain) and holds a degree in Computer Science from the Autonomous University of Madrid (Spain), and he is Certified Information System Auditor by ISACA (Information System Audit and Control Association). He is Assistant Professor of the Department of Information Technologies and Systems at the Rey Juan Carlos University (Spain). He participates at the GSyA research group of the Department of Information Technologies and Systems at the Castilla- La Mancha University. He is civil servant at the Spanish Tax Agency (in Madrid, Spain), where he works as IT Auditor Manager. His research activities are security governance, security requirements engineering, security in cloud computing, security in information systems, secure software process improvement and auditory, quality and product lines. He has several dozens of papers in national and international conferences, journals and magazines on these subjects and co-author of several chapter books. He belongs to various professional and research associations (ASIA, ISACA, ASTIC, ACTICA, etc).

Eduardo Fernández-Medina holds a PhD. and an MSc. in Computer Science from the University of Sevilla. He is associate Professor at the Escuela Superior de Informática of the University of Castilla-La Mancha at Ciudad Real (Spain), his research activity being in the field of security in databases, datawarehouses, web services and information systems, and also in security metrics. Fernández-Medina is co-editor of several books and chapter books on these subjects, and has several dozens of papers in national and international conferences (DEXA, CAISE, UML, ER, etc.). Author of several manuscripts in national and international journals (Information Software Technology, Computers And Security, Information Systems Security, etc.), he is director of the GSyA research group of the Information Systems and Technologies Department at the University of Castilla-La Mancha, in Ciudad Real, Spain. He belongs to various professional and research associations (ATI, AEC, ISO, IFIP WG11.3 etc.).

Website of the National Network of Information Security and Cryptography
http://www.renasic.org .br

ISSN 2358-8963

9 772358 869004