enigma

BRAZILIAN JOURNAL OF INFORMATION SECURITY AND CRYPTOGRAPHY

VOLUME 3 - ISSUE 1 - Brasilia - Brazil - 2016

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

ENIGMA — Brazilian Journal of Information Security and Cryptography

Eduardo Takeo Ueda

Mirela Sechi Moretti Annoni Notare
Rafael Timóteo de Sousa Júnior

VOLUME 3

ISSUE 1

Brasilia — Brazil
2016

ENIGMA — Brazilian Journal of Information Security and Cryptography

Editors

Editor in Chief
Eduardo Takeo Ueda

Associate Editors in Chief
Mirela Sechi Moretti Annoni Notare
Rafael Timóteo de Sousa Júnior (UnB, Brazil)

Editorial Board

André Luiz Moura dos Santos (UECE, Brazil)
Carlos Alberto Maziero (UTFPR, Brazil)
Denise Hideko Goya (UFABC, Brazil)
Eduardo James Pereira Souto (UFAM, Brazil)
Eduardo Martins Guerra (INPE, Brazil)
Leonardo Barbosa e Oliveira (UFMG, Brazil)
Raul Fernando Weber (UFRGS, Brazil)
Ruy José Guerra Barretto de Queiroz (UFPE, Brazil)
Rafael Rodrigues Obelheiro (UDESC, Brazil)

Coordinator of the National Network of Information Security and Cryptography

Antônio Carlos Menna Barreto Monclaro

Commander of Cyber Defense

General de Divisão Paulo Sergio Melo de Carvalho

Chief of the Science and Technology Department

General de Exército Juarez Aparecido de Paula Cunha

Brazilian Army Commander

General de Exército Eduardo Dias da Costa Villas Boas

Production

Matheus Barbosa, Graphic Designer of Logo
Daniel Alves Tavares, José Rafael A. de Amorim, Marcílio Costa Bezerra, Publication Designers

SUMMARY

ENIGMA – Brazilian Journal of Information Security and Cryptography Volume 3 Issue 1 September 2016

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 3 of ENIGMA – Brazilian Journal of Information Security and Cryptography. Submissions were accepted in English, Portuguese and Spanish. In this issue, 4 papers are published, of which 3 were peer-reviewed while another was invited and reviewed by the editorial board of the Journal. In addition, the invited paper was the Best Paper of the conference SBSeg'2015.**

***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 scientists in the history of computing.

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.

E. T. Ueda, Institute for Technological Research of the State of São Paulo, edutakeo@usp.br

M. S. M. A. Notare, IEEE Latin America Transactions Editor in Chief, FAERO Technology University in Fly Transportation, mirela@ieee.org

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

II. About Volume 3, Issue 1 of Enigma

In this first issue of Volume 3 of ENIGMA – Brazilian Journal of Information Security and Cryptography – 4 papers are published, and in this section we briefly describe the contribution of each of these papers.

The first selected paper, entitled "Machine Learning for Cryptographic Algorithm Identification", studies classical cryptographic algorithms identification with the support of machine learning. It shows the viability of classifying cryptograms, according to their encryption algorithm, by using data mining techniques. In this paper experiment, the random probability for guessing those algorithms is 25%. However, the mean value of correctness obtained reaches 97,23%. In addition, it seems that it is possible to increase this value.

The selected paper "A Wireless Physically Secure Key Distribution System" presents how to achieve wireless secure communication at fast speeds with bit-to-bit symmetric encryption. A fast and secure key distribution system is shown that operates in classical channels but with a dynamic protection given by the low noise of the light signal. The binary signals in transit in the channel are protected by coding with random bases and by the addition of physical noise that is recorded and added bit by bit to the signals. The hardware requirements is described as well as how to calculate the security level associated with the communication. A correct implemented system would offer privacy at a top-secret level for the users. Furthermore, the correct choice of parameters creates post-quantum security privacy.

In the next selected paper, "Future Internet and Reconfigurable Computing: Considerations on Flexibility and Security", the authors argue that it is necessary to approximate the areas of computer architecture and computer networks, or more specifically bridge the gap between research in Reconfigurable Computing and in the Future Internet Architectures. A brief survey with plainly successful examples indicates how some of the needs and future internet objectives can be met through reconfigurable computing, especially with respect to flexibility and security requirements.

The last paper in this ENIGMA issue, the invited paper "A Secure Protocol for Exchanging Cards in P2P Trading Card Games Based on Transferable e-Cash", which was considered the best paper of SBSeg´2015, presents a set of requirements for allowing secure trades in P2P TCGs, defining the cheating types that need to be detected. A transferable e-cash protocol

is adapted for creating a concrete scheme that fulfills those requirements. The proposed scheme is based on P-signatures, allowing a vector of messages to be signed, which is combined with a compact blind signature scheme in the asymmetric pairing setting to allow a more memory-efficient representation. According to preliminary analysis, the scheme is quite efficient to be used in practice.

III. Conclusion

ENIGMA – Brazilian Journal of Information Security and Cryptography – is now in its third year. By adopting since its creation the best practices from IEEE Transactions publications, we hope that soon this journal will become a reference among the leading international publications dedicated to information security and cryptography.

With the publication of this journal issue, Brazil is taking another step towards the future, because the 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, as threats to information security and privacy are risks for any nation, the ENIGMA journal can envision the international community.

Acknowledgments

We would like to thank all the authors who contributed with their papers for this 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 papers. We are grateful to National Network of Information Security and Cryptography (RENASIC) and the Cyber Defense Center (CDCiber) of the Defense Ministry 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.

Eduardo Takeo Ueda received the Ph.D. degree in Electrical Engineering in 2012, MSc degree in Computer Science in 2007, both from University of São Paulo (USP), and Specialist degree in Health Informatics in 2014 by Federal University of São Paulo (UNIFESP). He also holds a Mathematics BSc 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 Professor in Senac University Center of São Paulo, Master's Thesis Advisor in Institute for Technological Research of the State of São Paulo, member in conferences program commitees and reviewer of scientific journals. Currently, he is member of the 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. She is Professor at FAERO Tecnology University in Fly Transportation. Her main research of interest focuses on the proposition of security management solutions for Wireless, Mobile, Sensor Ad- Hoc Networks, Intelligent Vehicular Networks and Fly Transportation. 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 Regional Committees Chair of IEEE NoticIEEEro, Editorial Advisory Board of IEEE Spectrum/The Institute newsletter, 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 (22 years) of IEEE, and member of SBrT and SBC societies.
http://lattes.cnpq.br/8224632340074096.

Rafael Timóteo de Sousa Júnior 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. From 2006 to 2007, supported by the Brazilian R&D Agency CNPq, He took a sabbatical year in the Group for the Security of Information Systems and Networks, at Ecole Superiéure d´Electricité, Rennes, France. 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, where he is a member of the Post-Graduate Program on Electrical Engineering (PPGEE) and supervises the Decision Technologies Laboratory (LATITUDE). He is a member of the Brazilian Computer Society (SBC), member of the National Network of Information Security and Cryptography (RENASIC) and coordinates the Unit 6 of the Brazilian National Science and Technology Institute (INCT) on Cyber Defense. He has developed research in information and network security, distributed data services and knowledge discovery for intrusion and fraud detection, as well as signal processing, energy harvesting and security at the physical layer.
http://lattes.cnpq.br/3196088341529197.

Machine Learning for Cryptographic Algorithm Identification

F. M. Barbosa, A. R. S. F. Vidal and F. L. de Mello

***Abstract*—This paper aims to study encrypted text files in order to identify their encoding algorithm. Plain texts were encoded with distinct cryptographic algorithms and then some metadata were extracted from these codifications. Afterward, the algorithm identification is obtained by using data mining techniques. Firstly, texts in Portuguese, English and Spanish were encrypted using DES, Blowfish, RSA, and RC4 algorithms. Secondly, the encrypted files were submitted to data mining techniques such as J48, FT, PART, Complement Naive Bayes, and Multilayer Perceptron classifiers. Charts were created using the confusion matrices generated in step two and it was possible to perceive that the percentage of identification for each of the algorithms is greater than a probabilistic bid. There are several scenarios where algorithm identification reaches almost $97,23\%$ of correctness.**

***Keywords*—Cryptographic Algorithm Identification, Data Mining, Machine Intelligence.**

I. Introduction

The theme of this paper is cryptogram analysis in order to identify the cryptographic algorithm used for ciphering. Therefore, it aims to analyze segments from encrypted texts and use this information to identify those algorithms. Even though this test evaluates four cryptographic algorithms, the methodology is generic so that it can be applied to a greater set of algorithms.

The cryptographic algorithms are necessary in order to provide data confidentiality, integrity, authenticity and irreversibility, allowing only the emitter and the receptor of an encrypted message to access the original information content. Today, the cryptographic security depends on the key resistance to attacks and not on the obscurity of the algorithm, that is, the encryption key unknown but the algorithms method are notorious. There are several of such algorithms with different implementations, some are more popular than others are, either because their easiness for implementing or its performance.

Despite the common knowledge of algorithm implementation, the task of breaking the code is neither simple nor brief. First, it is necessary to find out the algorithm used for encoding, and once identified the algorithm, the efforts for obtaining the original information are restricted to attempts of breaking the cipher by using cryptanalysis. Hence, a straightforward cryptanalysis is a huge task. However, there are smaller and complex reduced activities that combined may allow the successful achievement of the task: to determine the cipher size, to retrieve the cipher key, to discover the type of encoding used for cyphering, and retrieve the encryption algorithm.

This work focus on the identification of algorithms used for encoding plain texts by classifying cryptograms trough data mining techniques. The action of finding the key used in such algorithms as well as reversing the encryption is beyond the scope of this article.

II. Related Work

There are a great variety of cryptography algorithms, a sort of procedure responsible for defining data transformations that cannot be easily reversed by unauthorized users. For instance, DES algorithm was developed by the former NIST institute and was widely adopted by industry. Kahate [1] states that this algorithm was the most used for two decades, although its popularity decreased due to its vulnerabilities. Tanenbaum [2] says that the original algorithm is not so secure, but some upgrades can adjust it to be useful. Pfleeger and Pfleeger [3] point out that its security might be achieved by applying successive techniques of substitution and transposition.

The Blowfish algorithm was proposed as an alternative for DES since this was vulnerable to brute force attacks and to others cryptanalysis approaches [4]. Since Blowfish was created to replace DES, some works focus on the comparison among those algorithms. Nie, Song and Zhi [5] provide interesting comparison of speed and energy consumption. Verma, Agarwal, Dafouti and Tyagi [6] demonstrated that the Blowfish is not only faster than DES, AES and Triple DES, but also provides a security enhancement because of its key size. Poonia and Yavad [7] show that some modifications can be made in order to make the algorithm more compact and safer than its original version.

RC4 is a patented algorithm widely used on stream cipher security software such as TLS, SSL and WEP [8]. It is also known as ARC4, since it was never released by RSA, even though its source code was leaked on the Internet [9]. Despite being a simple and efficient algorithm, easily implemented, and five times faster than DES [8], [21], there are several weaknesses that can be exploited [11], [12], [19]. According to Vanhoef and Piessens [10] RC4 should not be used any more.

RSA is the most known asymmetric algorithm [15] and was the first of such algorithms published in literature [16]. Its security relies on the difficulty of factoring very large prime numbers. Coutinho [15] shows that those prime numbers must be wisely chosen, otherwise it is relative simple to break this

F. M. Barbosa, Web Developer with mainframe platform experience, Rio de Janeiro, RJ, Brasil, flaviombarbosa@gmail.com

A. R. S. F. Vidal, Java Programmer with event processing experience, Rio de Janeiro, RJ, Brasil, arthurreimao@hotmail.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, Rio de Janeiro, RJ, Brasil, fmello@del.ufrj.br

kind of encryption. Notwithstanding, RSA has been used for encoding and decoding medical images [18] and for a hybrid Bluetooth communication algorithm [17].

Data mining is a process that uses several algorithms in order to retrieve valid patterns from large datasets that can be potentially useful in decision-making process [13], [14], [20]. J48 classifier is an implementation of the classic C4.5 decision tree data mining algorithm, and there are two possible pruning methods [22] to reduce time complexity. The Multilayer Perceptron, on its turn, is a neuron network classifier that is also widely known [26]. It consists of an input layer, intermediate layers and an output layer, where the classificatory training phase is supervised, with backpropagation as a method for minimizing error.

PART is a rules induction method that combines the approaches from C4.5 and RIPPER algorithms without performing a global optimization to produce rules set. The central idea is to create partial decision tree dividing the dataset as in C4.5. Once the partial tree is defined, one rule is obtained from the best leaf node. Gama [24] calls attention to functional trees as an ongoing approach for machine learning and decision models.

FT classifier [24] belongs to an algorithm family, which analyses the differences between decision models. It is similar to several other functional tree algorithms, but the nodes are created according to the samples provided. Additionally, the attributes used in the classification model are incorporated on demand. Besides, the algorithm provides decision lists organized by the set of rules [22].

Naive Bayes is a classifier commonly used as text classifier because its good speed performance, but it also some weaknesses. Rennie et al [25] present two of those weaknesses that influence its performance: 1) the different amount of data for each classes influence the decision weights definition for those categories; 2) the hypothesis of non-overlapping classes. The Complement Naive Bayes is a classifier proposed by Rennie et al [25] which aims to improve Naive Bayes by solving faults associated to misleading trainings.

III. Cryptographic Algorithm Detection

The strategy adopted to detect the algorithm used on a text encryption is to apply data mining algorithms over a set of encrypted files metadata. In order to support this experiment, several plain text files were collected from three different languages. Each file was encrypted with DES, Blowfish, RC4 and RSA algorithms. Then, descriptive metadata was extracted from the cryptograms, that is the sequences bit quantity. Afterward, the data mining procedures were executed by using J48, FT, PART, Complement Naive Bayes and Multilayer Perceptron classifiers. Finally, by using a confusion matrix generated at each mining algorithm execution, it was possible to create an estimative of successful identification of the cryptographic algorithm. Those stages are detailed as follows and illustrated at Fig. 1.

The plain texts used in this experiment encompass three distinct idioms corpora. It was chosen two latin idioms (Portuguese and Spanish) and one anglo-saxon idiom (English).

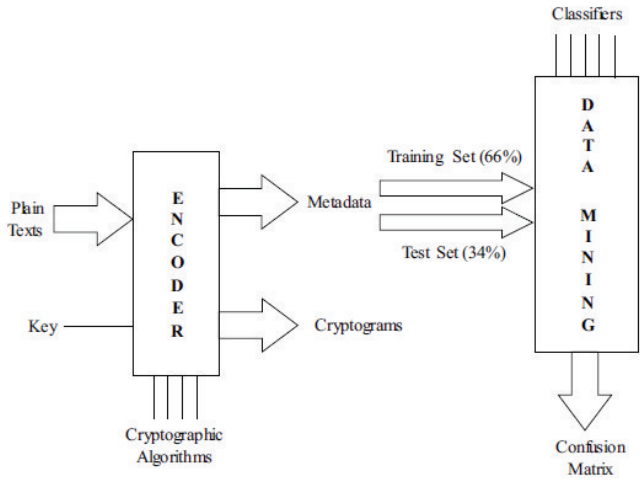

Figure 1. The block diagram of the experiment.

From the point of view of computer compilation, the two former idioms are linguistically more complex but they share some lexical and grammatical similarities. The latter idiom is quite different from the others but it has a simplest structure.

The choice for using those idioms is due to the possibility of comparing the results for different languages and evaluate the sensibility of the cryptographic algorithm classifiers under those circumstances. Each corpus is composed of 200 samples of distinct plain texts, not subject oriented, extracted from newspapers and magazines, without text fragments repetition, and at least with 6.000 characters each.

This study is exploratory in nature and therefore, there is no need for dealing with the most up to date cryptographic algorithms, as a result classical ones were chosen to be evaluated. The criterion for choosing the algorithms is to evaluate the detection behavior for block ciphers, stream ciphers, symmetric key algorithms and public key algorithms. For that reason, the instances of such algorithm classes that were chosen are DES, Blowfish, RSA and RC4.

By the end of the text encryption it is constructed a histogram of each file in order to provide metadata about the cryptograms. The first step for constructing this histogram is to express the number of bits that will define a block, and this block size varies from 4 bits to 16 bits. The bits inside a block correspond to values and then it counts how many values fall into that block.

The automation of text files encryption, for the four cryptographic algorithms, and the histogram construction conditioned to different block sizes was accomplished by using an application developed by Reimão [27]. Hence, three new encrypted files corpora were produced, with a set of corresponding histograms as metadata.

Finally, the metadata was submitted to the classification process containing a set of classifiers. Similar to the process of choosing the cryptographic algorithms, the choice for using classification algorithms instances is based on the viability of employing mining algorithms to the detection of the encryption procedure. By this reason, it was chosen representative algorithms of the classification categories, that is, bayesian

class, functional class, rules based class and decision trees generators class. Therefore, the instances of such classes used at this work are: J48, PART, FT, Complement Naive Bayes and Multilayer Perceptron.

Before performing the tests with the encrypted files corpora, it was necessary to construct the classification model for each classifier. Each encrypted files corpus was segmented into two distinct sets. The first one, containing 66% of the corpus, and was destined to the classification model creation. The second set, containing 34% of the corpus, was submitted to tests. For each individual set from the corpus, there was the same amount of cryptograms encoded with a given encryption algorithm. This feature avoids an algorithm identification enhancement against other algorithms.

IV. Results

The creation of all encrypted files took 42.3 minutes at an i3-2330M CPU @ 2.20 GHz with 4GB RAM memory. The files encoding is a fast procedure but the creation of all histograms, subjected to all possible block sizes, for all files from each idiom, is time costly. The next stage, the data mining stage, is highly dependent on the classifier algorithm. Besides, the execution time from all classifiers increases with the increment of the number of bits of the block, as expected. Moreover, the number of blocks increases as a power of two. These features create a bad environment for processing all combinations of <encrypted file, block size, mining algorithm> in order to create the classification models.

The J48 algorithm took 33.6 minutes of execution time, where 22.5 minutes were spent on constructing the classification model. The FT algorithm spent 7.46 hours executing, from which 2.46 hours were spent on constructing the model. The PART classifier constructed the model in 48.53 minutes from the total execution time of 73.33 minutes. The Complement Naive Bayes is the faster classifier, taking 55 seconds of execution time and just 4 seconds for building the model. At last, the Multilayer Perceptron was the most time consuming, it needed 62.09 hours for constructing the model and 68.91 hours of total execution time. One important notice is that the experiment with the Multilayer Perceptron was not fully accomplished because it was necessary to interrupt its execution. The neural network training is too slow, and thus the time for constructing the model became unfeasible as it increases exponentially. Therefore, it was necessary to limit the block size to 11 bits, that is, blocks from 12 to 16 bits were not evaluated because they are too time expensive. The block with 11 bits size, for instance, took 68.91 hours. Additionally, there were also problems with memory consumption.

The results analysis obtained from the classification process is based on confusion matrix. It aims to get an effective measure of the classification models, since those matrices makes explicit the number of correct classifications versus the number of inferred classifications for each cryptographic algorithm. This means that it was possible to compute the correctness percentage for each classifier applied to each encoding algorithm.

The plots presented in this section describe the classifiers performance for a sample with all three idioms mixed. There is a marginal difference between the performances of the classifiers when using distinct idiom corpus. In fact, the classification got better results for the English corpus, but those results are nor significantly different from the results obtained with the other two corpora. Therefore, it does not seem important to distinguish the idiom of the corpus, not only to define the classification model, but also to be used as test set.

The chart with the results obtained from J48 classifier is presented at Fig. 2. It shows that when using block size of 16 bits the correctness ratio for three algorithms (DES, Blowfish e RSA) is higher, while this classifier combined with the block size criterion is not much sensible for RC4. It is interesting that the smallest block size is a better descriptor because it reaches 61.88% of correctness, while the same block size of 4 bits provides an approximate correctness of 30% for the other classifiers. Moreover, it is observable that the better-identified algorithm under these circumstances is the RSA, with 87.77% of correctness mean value.

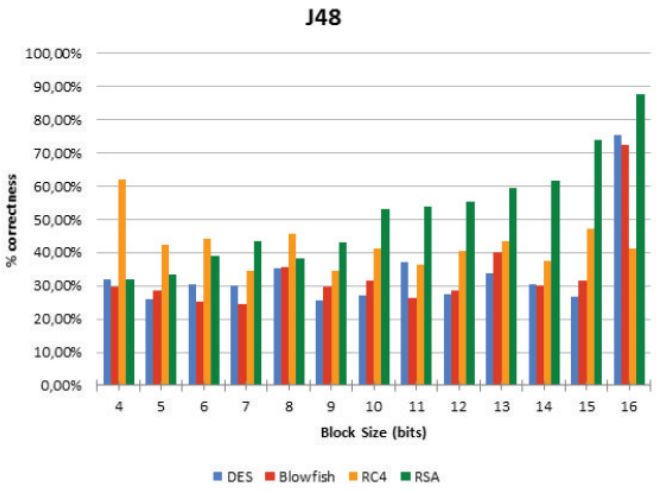

Figure 2. Correctness of J48 classifier, with block size of 4 to 16 bits, and corpus with tree idioms mixed up.

Fig. 3 shows the results for the FT classifier. The RC4 encoding is easily recognized by this mining algorithm because the correctness mean value is 66.01% (considering all block sizes). Notice that the 4 bits block size, as it happened to the J48 classifier, is also a good parameter to identify the RC4. However, block with 8 bits provide an even better discriminant. The correctness ratio for DES, Blowfish and RSA reaches its major value with 16 bits block size, as it happened to the J48.

The chart for the PART classifier is presented at Fig. 4 where it is possible to observe a scenario similar to what happened with J48 and PART classifiers. When using the 16 bits block size, the correctness ratio reaches the best results for DES, Blowfish and RSA algorithms. The RC4 algorithm identification is also not so sensible to this classifier. Nevertheless, the usage of PART indicates that the RC4 is again easier classified by using 4 bits block size, as it had already happened with J48 and somewhat with FT. Thus, it seems reasonable that the usage of 4 bits block size can be useful for RC4 identification. The DES and Blowfish identification became a little bit lower

when using 16 bits block size compared to FT, but the RSA identification increased the correctness mean value to 82.45%.

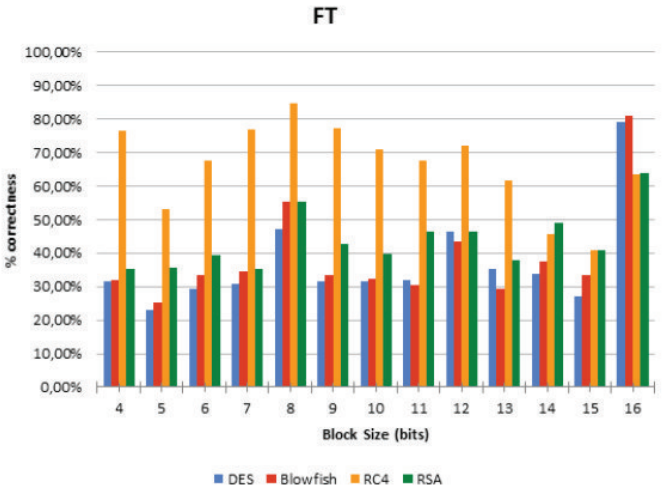

Figure 3. Correctness of FT classifier, with block size of 4 to 16 bits, and corpus with tree idioms mixed up.

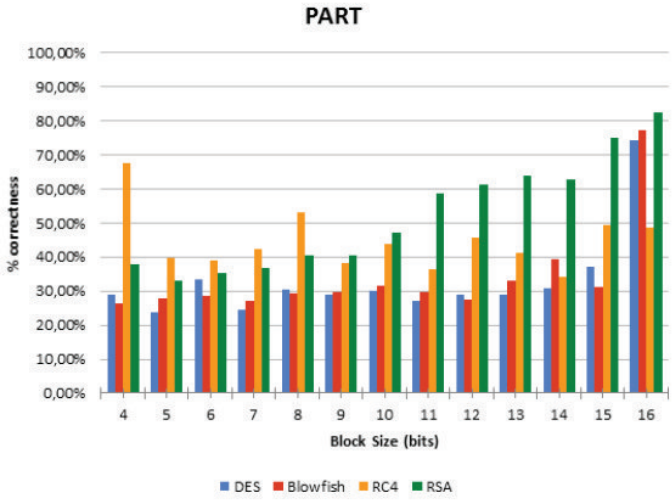

Figure 4. Correctness of PART classifier, with block size of 4 to 16 bits, and corpus with tree idioms mixed up.

The chart for Complement Naive Bayes classifier (Fig. 5 shows a completely different scenario. The correctness ratio increased significantly, whatever encoding algorithm is taken for analysis. The Blowfish algorithm is almost fully recognized, and its corresponding correctness mean ratio is 99.54%. The correctness mean value for RSA algorithm increased to 89.36%. DES and RC4 were fully recognized. Additionally, the RC4 algorithm is fully recognized using 8 to 16 bits lock size, in contrast to what had happened with the other classifiers.

At last, the Multilayer Perceptron chart is presented at Fig. 6. Remember that it was not possible to compute the classification model for blocks with 12 bits or more because of technical constraints. The computer platform used in this experiment (Intel i3, 4GB RAM and Windows 10) does not have enough memory to train a neural network with 2^{12} inputs (or more: $2^{13}, 2^{14}, 2^{15}, 2^{16}$) and 4 outputs. This causes a memory fault during this process. Additionally, the time spent to train the neural network was too long. Therefore, the chart from Fig. 6 shows results for 4 to 11 bits block size. The RC4 encoding is better recognized than the others are, and it has a higher correctness mean value for all block sizes. The 11 bits block size is the best parameter for RSA identification and the 8 bits block size is the best option for identifying DES and Blowfish.

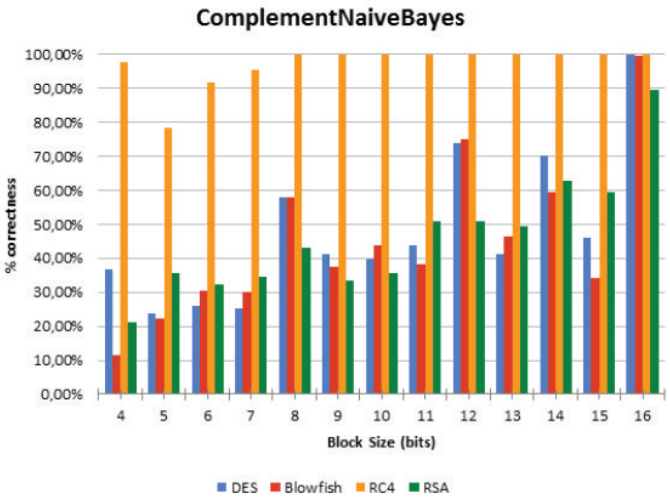

Figure 5. Correctness of Complement Naïve Bayes classifier, with block size of 4 to 16 bits, and corpus with tree idioms mixed up.

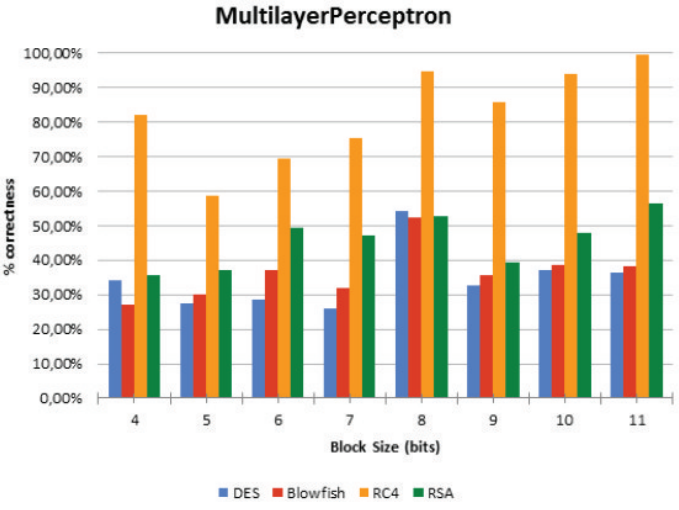

Figure 6. Correctness of Multilayer Perceptron classifier, with block size of 4 to 16 bits, and corpus with tree idioms mixed up.

Taking into consideration that four cryptographic algorithm identification is being studied, the bid for correctly choosing the class of algorithm for a given cryptogram is 25%, with any data analysis and with random selection. For all plots presented at this section it is possible to observe that the correctness mean values for all classifiers are greater than this 25%, even though considering the worst and marginal values of 30% to 33%. However, there are significantly higher

values for correctness ratio, which indicates that the usage of data mining algorithms is useful for encoding algorithm identification.

V. Conclusions

The task of discovering the algorithm used in the process of encoding plain text is not simple. During the cryptanalysis, this knowledge may reduce the efforts for obtaining the original message and thus compromising the information security. Even though this information is not deterministic for breaking the code, this approach of identifying the encoding algorithm can be a useful tool.

This work studied classical cryptographic algorithms identification with the support of machine learning. It showed the viability of classifying cryptograms, according to their encryption algorithm, by using data mining techniques. At this experiment, the random probability for guessing those algorithms is 25%. However, the mean value of correctness obtained here reaches $97,23\%$. In addition, it seems that it is possible to increase this value.

It is important to say that this study was subjected to several constraints. The results obtained here suggest that more efforts must be dedicated to this theme. Despite using a medium size sample space, it was possible to infer the correct cryptographic algorithm with a very good certainty. Therefore, in future works, the number of text samples should be enlarged. It is possible that the successful identification may saturate after using a given number of samples.

Moreover, the size of the blocks should also increased since small blocks do not provided many variation options and thus are not good discriminators. Blocks with 4, 8 and 16 bits size seems to be good parameters. It is reasonable to interested on the performance of block size with 24 and 32 bits. It is also interesting to evaluate if there is another block size that is significant for classification. Besides, it is curious why RC4 is so responsive for 4 bits block size, although this encryption algorithm is no longer relevant.

Furthermore, the set of idioms did not influence the classification process, but it is not guaranteed that those idioms are sufficiently different. The inclusion of other idioms with distinct alphabets and grammars, such as Bulgarian, Russian, Swedish, Chinese, and Arab, for instance, may provide the necessary variations for idiom sensitiveness.

Additional cryptographic algorithms are also a good improvement, so that the results obtained can be used in real applications. Different instances of encoders with Electronic Code Book (ECB), Cypher Block Chaining (CBC), Cypher Feedback (CFB) and Output Feedback (OFB) are also necessary to be studied.

The Complement Naive Bayes algorithm seems to be the best classifier, but this ranking can be changed when the number of different cryptographic algorithms increase. Likewise, more mining algorithms may be tested. One of those algorithms is the Weightless Neural Network, which can substitute the Multilayer Perceptron for fast training and classification.

Acknowledgments

The Federal University of Rio de Janeiro (UFRJ) Coppetec Foundation supported this research, under project Poli-19.257, which was carried out at the Machine Intelligence Laboratory of UFRJ.

References

[1] Kahate, A. *Cryptography and Network Security*, 3rd ed, Nova Deli, McGraw Hill Education, 2013.

[2] Tanenbaum, A. *Computer Network*, 5th edition, Boston, Pearson, 2011.

[3] Pfleeger, C. P. and Pfleeger, S. L. *Security in Computing*, Boston, Prentice Hall, 2006.

[4] Schneier, B. *Fast Software Encryption*, Cambridge Security Workshop Proceedings, pp. 191-204, 1994.

[5] [10] Nie, T., Song, C., Zhi, X. *Performance Evaluation of DES and Blowfish Algorithms*, International Conference on Biomedical Engineering and Computer Science (ICBECS), pp. 1-4, Wuhan, 2010.

[6] Verma, O. P., Agarwal, R., Dafouti, D., Tyagi, S. *Performance Analysis Of Data Encryption Algorithms*, 3rd International Conference on Electronics Computer Technology (ICECT), pp. 399-403, Kanyakumari, 2011.

[7] Poonia, V., Yadav, N. S. *Analysis of modified Blowfish Algorithm in different cases with various parameters*, International Conference on Advanced Computing and Communication Systems, pp. 1-5, Coimbatore, 2015.

[8] Hammood, M. M., Yoshigoe, K., Sagheer, A. M. *RC4-2S: RC4 Stream Cipher with Two State Tables*, Information Technology Convergence, v. 253, pp. 13-20, 2013.

[9] Paul, G., Maitra, S. *RC4 Stream Cypher and Its Variants*. Boston, CRC Press, 2012.

[10] Vanhoef, M., Piessens, F. *All Your Biases Belong To Us: Breaking RC4 in WPA-TKIP and TLS*, Proceedings of the 24th USENIX Conference on Security Symposium, pp. 12-14, Washington, 2015.

[11] Fluhrer, S., Mantin, I., Shamir, A. *Weakness in the Key Scheduling Algorithm of RC4*, Selected Areas of Cryptography, v. 2259, pp. 1-24, 2001.

[12] Mantin, I., Shamir, A. *A Pratical Attack on Broadcast RC4*, Fast Software Encryption, v. 2355, pp. 152-164, 2002.

[13] Navega, S., *Princípios Essenciais do Data Mining*, Anais de Infoimagem, Cenadem, 2002.

[14] Han, J., Kamber, M., Pei, J. *Data Mining Concepts and Techniques*, 3rd edition, Morgan Kaufmann, Waltham, 2011.

[15] , Coutinho, C. S. *Números Inteiros e Criptografia RSA*, IMPA, Rio de Janeiro, 2003.

[16] Das, A., Madhavan, C. E. V. *Public-key Cryptography Theory and Practice*, Deli, Pearson, 2009.

[17] Ren, W., Miao, Z. *A Hybrid Algorithm Based on DES and RSA in Bluetooth Comunication*, Second International Conference on Modeling, Simulation and Visualization Methods (WMSVM), pp. 221-225, Sanya, 2010.

[18] Anane, N., Anane, M., Bessalah, H., Issad, M., Messaoudi, K. *RSA Based Encryption Decryption of Medical Images*, 7th International Multi-Conference on Systems Signals and Devices (SSD), pp. 1-4, 2010.

[19] Goutam, P., Subhamoy, M. *RC4 State Information at Any Stage Reveals the Secret Key*, IACR Cryptology ePrint Archive, 2007.

[20] Witten, I. H., Frank, E., Hall, M. A. *Data Mining Practical Machine Learning Tools and Techniques*, 3rd edition, Morgan Kaufmann, Burlington, 2011.

[21] Gupta, S., Chattopadhyay, A., Sinha, K., Maitra, S., Sinha B. *High-performance hardware implementation for RC4 stream cipher*, IEEE Transaction Computers, v. 62(4), pp. 730-743, 2013.

[22] Mohamed, W. N. H. W., Sallen, M. N. M., Omar, A. H. *A Comparative Study of Reduced Error Pruning Method in Decision Tree Algorithms*, IEEE International Conference of Control System, Computing and Engineering, Penang, pp. 23-25, 2012.

[23] Frank, E., Witten, I. *Generating Accurate Rule Sets Without Global Optimization*, Proceedings of the Fifteenth International Conference on Machine Learning, pp. 144-151, São Francisco, 1998.

[24] Gama, J. *Functional Trees*, Machine Learning, v. 55(3), pp. 219-250, 2004.

[25] Rennie, J. D. M., Shih, L., Teevan, J., Karger, D. R. *Tackling the Poor Assumpltions of Naive Bayes Text Classifiers* Proceedins of the Twentieth International Conference on Machine Learning, Whasington DC, 2003.

[26] Silva, L. N. C. *Análise e Síntese de Estratégias de Aprendizado Para Redes Neurais Artificiais* Projeto de Mestrado, Universidade Estadual de Campinas, Setembro de 1998.
[27] Reimão, A. S. F. V. *Análise de blocos de arquivos criptografados para obtenção do algoritmo*, Projeto de Graduação, Universidade Federal do Rio de Janeiro, Fevereiro 2015.

Flávio Mendonça Barbosa did his MBA on computer and systems engineering at Federal University of Rio de Janeiro - UFRJ (2016) and undergraduation on computer science at Federal University of Rio de Janeiro - UFRJ (2012). Worked on mainframe platform for six years and has been developing web apps with Java as the backend platform since 2012.

Arthur Reimão Santos Figueiredo Vidal did his undergraduation on electronics engineer at Federal University of Rio de Janeiro - UFRJ (2014). Worked with event processing for a year and a half and is currently studying for a public sector job.

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 Mentor Group. Associate Professor at the Electronics and Computing Department (DEL) of Polytechnic School (Poli) at Federal University of Rio de Janeiro (UFRJ) since 2007.

A Wireless Physically Secure Key Distribution System

G. A. Barbosa

***Abstract*—A fast and secure key distribution system is shown that operates in classical channels but with a dynamic protection given by the shot noise of light. The binary signals in the communication channel are protected by coding in random bases and by addition of physical noise that was recorded and added bit by bit to the signals. While the resulting signals are classical they carry the uncontrollable randomness information in the signal sent. The legitimate users start with a shared secret between them creating a measuring advantage over the adversary. This way the introduced noise does not affect the users but frustrates the attacker.**

***Keywords*—Random, Physical Processes, Cryptography, Privacy Amplification.**

I. Introduction

A FAST and secure *key distribution* system is presented to operate in generic communication channels, including wireless channels. The transmitted signals are deterministic (or perfectly copied) but include continuously recorded random noise that frustrates an attacker to obtain useful information. This noise affects the attacker but not the legitimate users that share an initial shared secret bit sequence c_0. The legitimate users will end up with a continuous supply of fresh keys that can be used even to encrypt information bit-to-bit in large volumes and fast rates.

The wireless key distribution system discussed in this work uses the intrinsic light noise of a laser beam to frustrate an attacker to extract meaningful signals. However, this noise is not in the communication channel but it is recorded *before* reaching the channel.

Historically, cryptography using optical noise from coherent states in a communication channel can be traced back to [1] and [2]. The first uses quantum demolition measurements and quadrature measurements while the second uses direct measurements with no need for phase references or quantum features besides the presence of optical noise. The methods and techniques used are widely different. The use of the optical noise in this paper has a relationship to the one originally used in [2], where fiber optics communication in a noisy channel blocked information leakage to an adversary. An initial shared information on a M-ry coding protocol used by the legitimate users allowed them to extract more information from the channel than the one obtained by the adversary. More recently that original idea was improved with a specific privacy amplification protocol [3] while keeping the use of an optical communication channel.

The present work merges main ideas of the protection given by the light's noise in a protocol applied to wireless channels. Seed ideas on the use of a wireless channels using recorded physical noise were introduced from 2005 to 2007 [4]. This work brings those ideas of wireless channels secured by recorded optical noise to a practical level. It also opens up the possibility to immediate application of the technique to mobile devices. This new scheme is detailed and the associated security level is calculated. This system performs one-time-pad encryption with the securely distributed keys.

Symmetric keys with end-to-end encryption, where keys are kept secret by the users, may provide perfect secure communication for companies. Government distribution of keys for their users could guarantee secure communication among users as well as dispose of tools to access necessary exchanged information whenever a strong need exists. In the same way, companies that distribute keys for their users could comply with legal requirements such as the All Writs Act (AWA) - as long as their key repository are kept under control.

A step-by-step description of this system will be made along this paper. The key distribution system not just generates and distribute cryptographic keys but also provide functions like encryption and decryption between users (or "stations") A and B: It is a *platform* for secure communications.

II. Platform for secure communications

Fig. 1 shows a block diagram of this platform for one of the users, say A. Users A and B have similar platforms. Communication between A and B proceeds through the communication ports of a PC with access to the Internet (Top portion of Fig. 1). This PC works as the interface with the exterior and is isolated from the platform (Bottom of Fig. 1) by an air gap. In other words, the platform has no direct access to or from the Internet. Data flow in and out is done through a Dynamic memory conjugate with OR switches that only allow authenticated and fixed size packets.

The platform is roughly composed of two opto-electronic parts: 1) A fast Physical Random Bit Generator (PhRBG) and 2) a Noise Generator. The PhRBG delivers bits to a Bit Pool where a Privacy Amplification (PA) protocol is applied and encryption and decryption functions are performed. Description of these parts will be made along the paper; they are intertwined in their functionalities as as such their understanding are necessary for a full comprehension of the proposed system. A PC-mother board (not discussed in this paper) in the platform perform several operations and provide access for the users, including a graphical interface for platform control.

Although this is a quite general system allowing privacy in communications one could mention a few applications like secure communications for embassies or the secure transfer of large volumes of patient data among medical centers and insurance companies.

G. A. Barbosa, QuantaSec – Consulting and Projects in Physical Cryptography Ltda, Av. Portugal 1558, Belo Horizonte MG 31550-000, Brazil, e-mail: GeraldoABarbosa@gmail.com

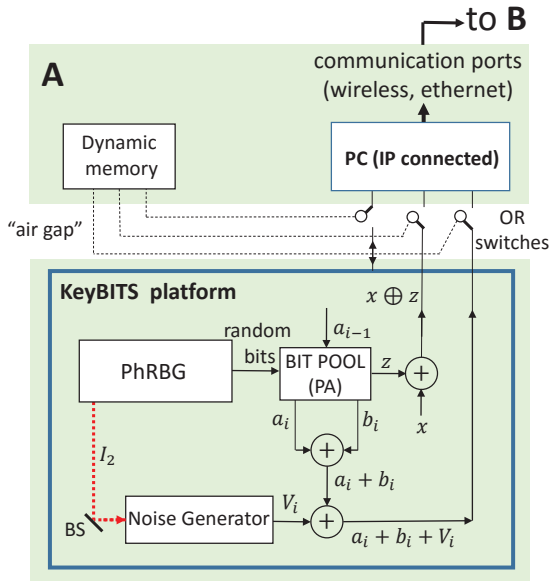

Figure 1. Each station (A or B) is physically controlled by each user and is composed of an IP connected PC that exchange communications between stations. The platform has no direct connection to the communication channels. Data flow in and out from the platform to the communication PC is done through a Dynamic memory and OR switches. This memory contains instructions only allowing transit of authenticated packets with fixed size. The platform contains the Physical Random Bit Generator (PhRBG), a Bit Pool and a Noise Generator to efficiently mimic the optical shot noise of a noisy optical communication channel.

III. Physical Random Bit Generator

The fast Physical Random Bit Generator (PhRBG) is of a novel type described in Reference [5]. The PhRBG extract broad bandwidth fluctuations (shot-noise) of a laser light beam and delivers random voltage signals (V_+, V_-) –signals that can be expressed as random bits– to the Bit Pool.

Fig. 2 provides more details. Left upper part of Fig. 2 shows the PhRBG. A laser beam excites a multi-photon detector and the voltage output pass through amplifiers G and an analog-to-digital (ADC) converter. The laser intensity I_1 and the gain G are adjusted to enhance the current from the noisy optical signals well above electronic noises:

$$\overline{(\Delta I_{light})^2} \gg \overline{(\Delta I_{electronic})^2}\,. \qquad (1)$$

It also necessary to work below the range where the ratio noise/signal is too small. In terms of the number of photons n:

$$\frac{\text{Noise}}{\text{Signal}} = \frac{\sqrt{(\langle \Delta n \rangle)^2}}{\langle n \rangle} = \frac{1}{\sqrt{\langle n \rangle}} \to \text{not small}\,. \qquad (2)$$

In other words, the desired signals are optimized optical shot-noise signals that allow a good number of detection levels from an ADC. The stream of digitalized fluctuating signals are classified within short time intervals in signals above the average value as bit 1 signals (V_+) while signals below the average are identified as bit 0 signals (V_-).

Sampling time for acquisition of the bit signals are set much shorter than the coherence time of the laser used. By doing so samplings occur within a fixed optical *phase* of the sampled photons. This leads to photon *number* fluctuations that are maximal: Although phase and number (or photon amplitude) are not strictly conjugate variables, there is an uncertainty relationship for number and phase.

The individual bit signals generated by the PhRBG around time instants t_i will be designated by a_i and a sequence of a_i by a. Notation a sometimes designates a sequence of bits or the size of this sequence whenever this does no give rise to notational problems.

User A wants to transmit in a secure way these random a bits to user B.

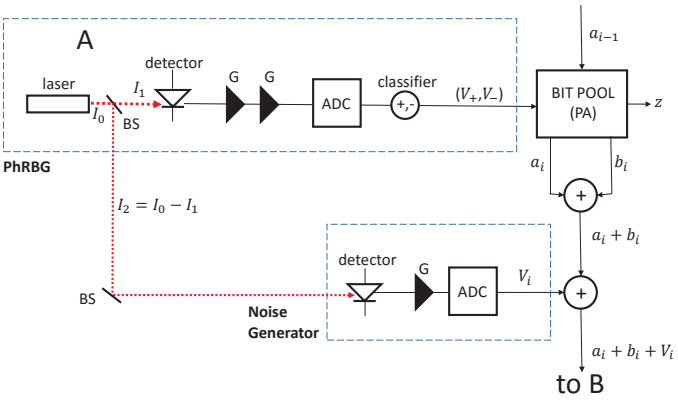

Figure 2. Hardware to generate random bits (PhRBG), Noise Generator and bit pool for Privacy Amplification. The bit pool contain memories and a FPGA (Field Programmable Gate Array) to perform fast operations locally.

Appendix A comments on the PhRBG.

IV. Bit Pool and Noise Generator

Fig. 2 also shows at the right upper side a Bit Pool where random bits generated from the PhRBG are stored together with bits b_i that were already acquired and recorded. The initial sequence $\{b_0\}$ is taken from a secret sequence c_0 of size $c_0 = ma$ initially shared between the legitimate users. The Bit Pool outputs signals $b_i + a_i$. Bits b_i act as a modulation or encryption signals to the random bits a_i. After application of the PA protocol a final distillation of z bits, over which the attacker has no knowledge, will be available for encryption and decryption purposes. A full discussion of these operations are made ahead when discussing the physical modulation of the signals and the Privacy Amplification protocol.

A. Noise Generator and recorded optical shot-noise

Bottom part of Fig. 2 shows the Noise Generator. A laser beam with intensity I_2 is detected, amplified to produce optical shot noise limited signals. These signals are digitalized producing a sequence of independent noise signals $V_N = \{V_i\}$. V_i is added to the signals $a_i + b_i$ giving $a_i + b_i + V_i$ and sent to B. The noise contribution V_N replaces the intrinsic optic noise in an optical channel. The magnitude and format of this noise will be shown after presenting the idea of $M-$ry bases.

The first modulation signal b_0 is defined by m random bits from c_0. In general the modulating random signal b_i can be seen as a transmission *basis* for a_i. One may as well see bits a_i as a message and b_i as an encrypting signal. To generate each b_i, or one number among M, m bits are necessary $(m = \log_2 M)$.

It is to be understood that the $M-$ry coding interleaves bits in the sense that the same bit signal superposed to a basis b_k representing a bit 1 (or 0) represents the opposite bit 0 (or 1) in a neighbor basis b_{k-1} or b_{k+1}. For example, see Fig. 1 in [3] for a physical representation of these interleaved bits in the optical phase space. Other possible realization of distinct neighboring levels with distinct bits could be made with levels separated by small physical displacements different from phase, e.g. amplitude, as shown in Fig. 3. This, together with the added noise V_N do not allow the attacker to obtain the bit a_i.

B. Noise Generator and $M-$ry bases

As shown in the bottom right part of Fig. 2 a random signal V_i is added to $a_i + b_i$, giving $a_i + b_i + V_i$ to be sent to B. It is emphasized that although the signal sent from A to B is deterministic, V_i is a *recorded random noise* that varies from bit to bit. A recorded signal is deterministic by definition because it can be perfectly copied. However, this recorded noise is an instance of an unpredictable event by nature.

In nature the noise intensity is continuous but the recorded digitalized noise is distributed among the M levels supplied by an Analog to Digital Converter. This statistical distribution among M levels also has a characteristic deviation σ_V. This will be discussed ahead.

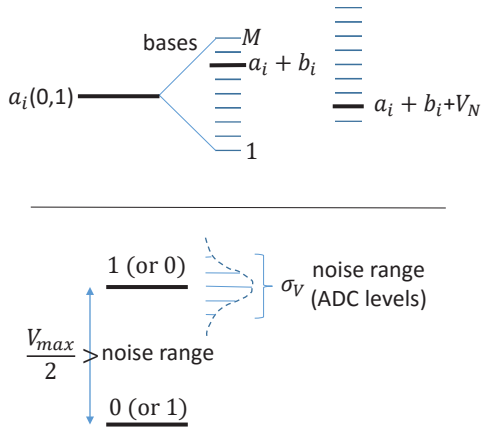

Figure 3. TOP - The physical amplitude signal representing a given basis b_i is added to the signal representing a bit a_i. The basis signal is known to users A and B but not to the attacker. A signal a_i is to be seen as a given bit in basis b_i (1 or 0) but this same signal a_i will be seen as the opposite bit (0 or 1) when attached to neighboring bases b_{i+1} or b_{i-1}. Therefore even a small noise V_i added to $a_i + b_i$ do not allow an attacker to know which bases have been used. However, both A and B know b_i and thus the bit sent a_i can be extracted. Physically, distinct modulation voltage signals may represent bits and bases. There are M bases; they can be assumed separated by V_{max}/M. A bit 1 and a bit 0 can be assumed physically separated by $V_{max}/2$. Voltage signals can be assumed cyclic in the sense that $V_{max} + \epsilon \to \epsilon$. BOTTOM - The amplitude between a signal a_i representing a bit and the opposite one is greater than the noise range. As voltage modulation of signals representing 1 and 0 are separated by $V_{max}/2$ this allows a precise bit determination by the legitimate users. On the contrary, the attacker struggles unsuccessfully with resolution of neighbouring levels.

The noise signal V_i is derived from a split beam of intensity I_2 (see left bottom part of Fig. 2). Light from this derived beam excites a multi-photon detector, the output is amplified by G and digitalized. An extra amplifier G may be adjusted to levels compatible to the signal $a_i + b_i$. In other words, the added noise mix potential bases and bits so that the attacker could not identify either the bit or the basis sent. Fig. 3 sketches the addition of the random basis b_i and the added noise V_i. The attacker does not know either the basis b_i neither the noise V_i and, therefore, cannot deduce the bit sent a_i from the total signal $a_i + b_i + V_i$.

V. Privacy Amplification and Fresh Bit Generation by A and B

The Privacy Amplification process to be utilized was first shown in Section IX of Reference [3]; it utilized the formalism originally developed in Reference [6]. In the present work it is applied to a classical communication channel instead of a noisy fiber optic channel.

Briefly, the following steps are performed:

1) The Bit Pool starts with the bit sequence of size $c_0 = ms$ (bits b_i) already shared by A and B. A sequence a of bits, $a = \{a_i\}$, is generated by the PhRBG and stored in the Bit Pool by user A. The sequence a is sent from A to B after the preparation that add bases and noise. A number of bits $a + ma$ is used for the task of creating bits a_i and bases b_i to be sent from A to B as $\{a_i + b_i\}$.

2) An instance of a universal hash function f is sent from A to B.

3) The probability for information leakage of bits obtained by the attacker over the sequence sent is calculated (as indicated ahead), generating the parameter t (number of possibly leaked bits). In other words, from sequence $\{0,1\}^n$ the attacker may capture $\{0,1\}^t$.

The PA protocol includes the following steps: From the $n = a + b$ bits stored in the Bit Pool, t bits are destroyed:

$$\{0,1\}^n \to \{0,1\}^{n-t}\,. \tag{3}$$

An extra number of bits λ is reduced as a security parameter [6]. This reduction in bit numbers is then

$$\{0,1\}^n \to \{0,1\}^{n-t} \to \{0,1\}^{n-t-\lambda} \tag{4}$$

where $\{0,1\}^{n-t-\lambda}$ is the final number of bits. The initial total amount of bits n in the Bit Pool was then reduced to $r = n - t - \lambda$. These remaining bits are then further randomized by the PA protocol [3]. The protocol establishes that the attacker has no information on these reduced and "shuffled" number of bits r.

The number r of bits can be rearranged in sizes as follows

$$\begin{aligned} r &= n - t - \lambda = (a+b) - t - \lambda = (a - t - \lambda) + b \\ &= (a - t - \lambda) + ma \equiv z + ma\,. \end{aligned} \tag{5}$$

The sequence of size $z \equiv (a - t - \lambda)$ (see output from Bit Pool in Fig. 2) will be used as *fresh bits* for encryption while the sequence of size ma will form the new bases $\{b_i\}$ for the next round of bit distribution. The process can proceed without the legitimate users having to meet or use a courier to refresh an initial sequence ma. Other rounds then may proceed.

The PA theory [6] says that after reducing the initial number of bits from $n = a + ma$ (ma initially shared and a fresh bits) to $r = n - t - \lambda$, the amount of information that may be acquired by the attacker is given by the *Mutual Information* I_λ. Corollary

TABLE I
PRIVACY AMPLIFICATION PROTOCOL FOR THE WIRELESS PLATFORM

PA protocol		
INITIALIZATION: A and B share c_0 of size and entropy $m\,s$.		
Station A		
#	ACTION	OBJECTIVE
1a	$a_i =$ GetString(PhRBG)	Get bitstring from PhRBG
1b	$b_i = c_{i-1}[1, ms]$	Extract $m\,s$ from pool for bases b
1c	Code&Send(a_i, b_i)	Send over classical channel
2	Send f	Send instance of universal hash f over classical channel
3a	$c_i = f(c_{i-1}\|\|a_i)$	A applies PA from $ms+s$ bits reducing them to $ms+s-t-\lambda$
3b	$z_i =$ $c_i[ms+1, ms+s-t-\lambda]$	A uses $s-t-\lambda$ bits from pool as the key stream z. The remaining $m\,s$ bits form the bases for next round.
Station B		
1a		no matching step to A's
1b	$b_i = c_{i-1}[1, ms]$	Get bases bits from initial pool value
1c	$a_i =$Receive&Decode(b_i)	Receive bits from classical channel
2	Receive f	receive instance of universal hash f
3a	$c_i = f(c_{i-1}\|\|a_i)$	B applies PA from $ms+s$ bits reducing them to $ms+s-t-\lambda$
3b	$z_i =$ $c_i[ms+1, ms+s-t-\lambda]$	B uses $s-t-\lambda$ bits from pool as the key stream z. The remaining $m\,s$ bits form the bases for next round.

5 (pg. 1920) in Reference [6], gives the information leaked to the attacker:

$$I_\lambda = \frac{1}{\ln 2 \times 2^\lambda} = \frac{1}{\ln 2 \times 2^{n-t-r}}\,. \tag{6}$$

A. Protocol steps

Table I list all steps of the protocol. The basis assigned for each bit sent uses $\log_2 M$ to encode it and the process is continuously sustained in rounds of s bits, in an unlimited way. This procedure has been shown to be very fast in hardware.

A and B use the protocols in a concerted manner and extract a sequence z of bits over which the attacker has no knowledge. One should recall that the communication channel is classical and the signals contain recorded optical noise modulating each bit sent. At every round A and B know the basis used and they use this to their advantage so that the noise V_N does not disturb identification of a_i. A secure distilled stream of bits from A is transferred to B.

The protocols proceeds to other similar runs. After n runs, Alice and Bob share nz bits.

VI. LEAKAGE PROBABILITY AND MUTUAL INFORMATION I_λ

Calculation of the mutual information I_λ that is directly connected to the probability for an attacker to extract useful information sent from A to B. It depends on the parameter t (number of possibly leaked bits in a sequence sent).

In the wireless scheme the number of levels used as bases depends on the digital hardware utilized (8 bits resolution$\rightarrow M = 256$, 10 bits resolution $\rightarrow M = 1024$ and so on). This converter sets the maximum number of levels M. Voltage signals $V_k\,, (k = 0, 1, 2 \ldots M)$ will represent these bases and to alternate bits in nearby bases one may chose bases by voltage values given by

$$V_k = V_{\text{max}} \left[\frac{k}{M} + \frac{1-(-1)^k}{2} \right]. \tag{7}$$

At the same time, as voltage signals V_N representing recorded optical noise will be added to these values, these values should have a span smaller than V_{max} (see Fig. 3). However, this span must be large enough to cover a good number of bases so that the attacker cannot resolve the basis b_i when a bit a_i is sent. The actual optical noise has a continuous span but the recorded region is set by digitalized levels of the ADC used. Setting the spacing of signals for bases similar to the spacing V_{max}/M of recorded noise levels, one could set the digitalized noise deviation, by adjusting the gain G, such that

$$V_{\text{max}}/M \ll \sigma_V \ll V_{\text{max}}\,. \tag{8}$$

This condition can be mapped to the same formalism utilized in the POVM (Positive Operator Valued Measure) calculation developed in [2] and from which the leakage bit probability t can be obtained. One may write the probability for indistinguishability between two levels separated by Δk, as

$$P_{\Delta k} = e^{-\frac{|\alpha|^2}{4}\left(\frac{V_{\Delta k}}{V_{\text{max}}}\right)^2} = e^{-\frac{|\alpha|^2}{4}\frac{(\Delta k)^2}{M^2}} \equiv e^{-\frac{\Delta k^2}{2(\sigma_k)^2}}\,. \tag{9}$$

The expected deviation σ_k in the number of levels is

$$\sigma_k = \sqrt{\frac{2}{\langle n \rangle}} M\,, \tag{10}$$

where $\langle n \rangle = |\alpha|^2$ and α is the coherent amplitude of a laser.

Calculation of the probability of error P_e for an attacker to obtain a bit sent follows what was done in [2]. Fig. 4 exemplifies these errors for a set of M values (number of bases) and number $\langle n \rangle$ of photons detected. For a sequence of s bits sent the parameter t (bit information leaked in s) in Eq. 6 will be $t = (0.5 - P_e) \times s$. With t calculated and the safety parameter λ defined, the probability for information that could be leaked to the attacker is calculated. It can be shown [2] that $t \sim 10^{-4}$ can be easily obtained; therefore with a sequence os $s = 10^6$ bits sent, this gives $t \sim 10^2$.

Fig. 5 exemplifies the PA effect by $(\log_{10} I_\lambda)$ (see Eq. 6) as a function of r, $(0,1)^n \rightarrow (0,1)^r$, and t, number of bits leaked to the attacker.

VII. OTHER APPLICATION EXAMPLES

Above sections described the basic parts of the platform for secure communications.

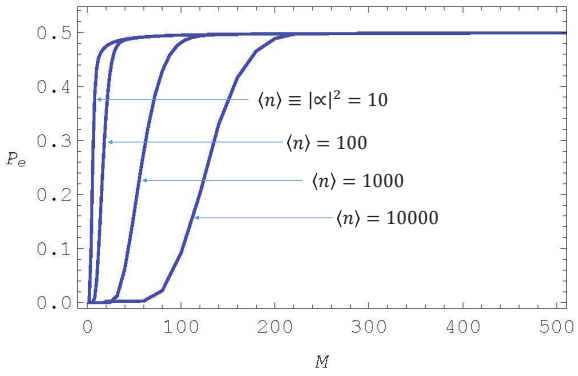

Figure 4. Probability of error for an attacker on a bit as a function of the number M of bases used and the average number of photons $\langle n \rangle$ carrying a bit.

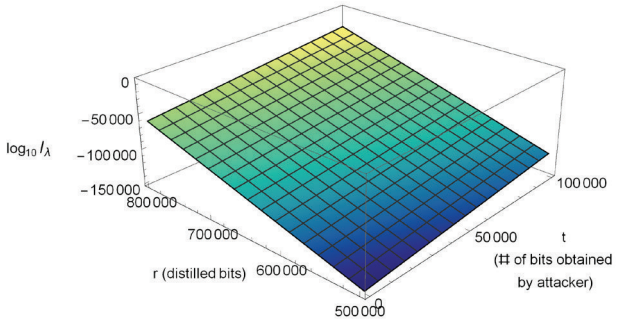

Figure 5. $\log_{10}$ of the Mutual Information I_λ leaked to the attacker after Privacy amplification is applied. In this example 10^6 bits are sent. t gives the number of bits leaked to the attacker before PA is applied and r is the distilled or useful remaining bits.

The use of a FPGA (Field Programmable Gated Array) and memory allow functions like bit storage and encryption and perform the "Bit Pool" functions necessary. Custom tailored applications can also be programmed under this fast hardware processing.

The current rack holding the PhRBG can be reduced to a small size with output directly coupled to a smart phone. This can provide true secure communications (bit-to-bit encryption) between cellular telephone users as another application example. It is also useful to call the attention to the reader that a decentralized protocol for bit-by-bit encryption for $N-$users exists [7]. Under this protocol, once N users acquire a long stream of random bits from the PhRBG, they can exchange secure information among them without any need to contact a central station to synchronize their bit streams.

Both Secure Data and Voice Over Internet (VOIP) can be implemented. It should be emphasized that it is important that the key storage must be kept "outside" of the mobile device and that the flow of information from the key generation and encrypting unit to the device connected to the Internet should be strictly controlled.

Other steps, more costly, can produce an ASIC (Application Specific Integrated Circuit) to reduce the system to a chip size device.

Another possible application example for the platform is to feed a Software-Defined-Radio (SDR) with cryptographic keys for bit-by-bit encryption/decryption capabilities. This could bring absolute security for Data and Voice communications through SDR.

VIII. Conclusions

It was shown how to achieve wireless secure communication at fast speeds with bit-to-bit symmetric encryption. The hardware requirements was described and it was shown how to calculate the security level associated to the communication. Miniaturization steps may allow easy coupling to mobile devices. The key storage have to be under control of the legitimate users and no key should ever be stored where a hacker could have command/control of the system. A correct implemented system would offer privacy at top-secret level for the users. Furthermore, the correct choice of parameters creates a post-quantum security privacy.

Appendix

A. Platform - rack implementation

The PhRBG, within the platform, is seen as a rack implementation in Fig. 6 and some details in Fig. 7. A detailed description of the PhRBG will be published elsewhere [8]. Just a brief description is presented here.

The PhRBG is an opto-electronic device designed to generate bits continuously to supply any demand for bits at high speeds. The physical principle involved, quantum vacuum fluctuations that produce the optical shot-noise, is not bandwidth limited and the device speed can be adapted to all electronic improvements. Among the differences with other quantum random bit generators the presented device has no need for interferometry and a single detector is used. This gives a time stable operation for the system.

The PhRBG was currently implemented with off-the-shelf components including low cost amplifiers (See G in Fig. 1). These amplifiers have a frequency dependent gain profile (a monotonous high gain at low frequencies) that introduces a low frequency bias in the bit generation. To compensate for this bias without increasing costs a Linear Feedback Shift Register (LFSR) is used in series with the bit output to produce an extra randomization. This breaks –the expectedly more rare– long sequences of repeated bits. This process does not reduce the speed of the PhRBG.

The currently implemented PhRBG works at ∼2.0 Gbit/sec and passes all randomness tests to which it was submitted, including the NIST suite described in "NIST's Special Publication 800 - A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications".

Besides passing conventional randomness tests, some visual information conveys the same idea. Fig. 8 shows amplitudes of a Fourier analysis of a bit stream revealing the white spectrum character of the generated bits. Figs. 9 and 10 show data and the expected occurrence of random bits for a distribution where the probability to occur 0 or 1s are equal, $p = 1/2$. It is expected that the probability to occur a sequence of k identical bits (either 0 or 1) is $p(k) = 1/2^k$. If one changes basis 2 to basis "e" one writes

$$p(k) = \frac{1}{2^k} = e^{-k \ln 2} \simeq e^{-0.693147\, k}. \qquad (11)$$

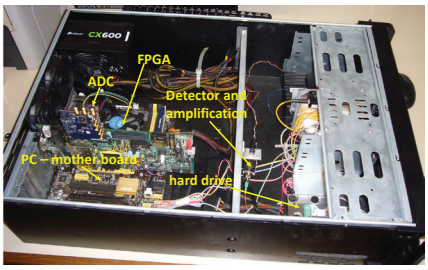

Figure 6. A top view with some components of the platform. The laser, detector, amplifier and hard drive are at the right side of the rack. The ADC that format analog signals from the optical amplification is connected to the FPGA for processing. A PC-motherboard provides management of several functions including a friendly graphical interface for the user.

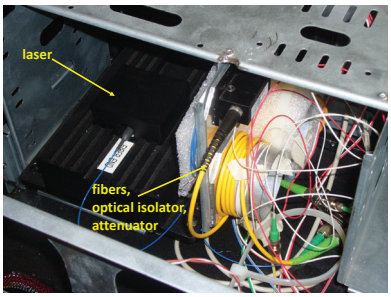

Figure 7. Detail of the laser location, optical isolator and attenuator.

Data in Figs. 9 and 10 were fitted to $p(n) = c\,e^{-an} = c\,e^{\ln 2^{1-\epsilon} n}$, where ϵ will indicate a depart from the distribution $p(k) = 1/2^k$. The raw data [9] for the histograms are given by lists L_1 and L_0:

$$L_1 = \{\{1,159676\},\{2,79651\},\{3,40253\},\{4,20017\},\{5,9864\},\{6,4960\},\{7,2567\},\{8,1239\},\{9,623\},\{10,313\},\{11,156\},\{12,59\},\{13,37\},\{14,21\},\{15,9\},\{16,8\},\{17,3\},\{18,4\},\{19,1\},\{20,0\},\{21,0\}\} \quad (12)$$

$$L_0 = \{\{1,159805\},\{2,79964\},\{3,39766\},\{4,20021\},\{5,9892\},\{6,4962\},\{7,2488\},\{8,1306\},\{9,630\},\{10,336\},\{11,148\},\{12,71\},\{13,42\},\{14,10\},\{15,11\},\{16,6\},\{17,2\},\{18,0\},\{19,1\},\{20,1\},\{21,1\}\}\,. \quad (13)$$

One should observe that the deviation parameter ϵ is exponentially small, giving an estimate of the randomness associated with the generated bits.

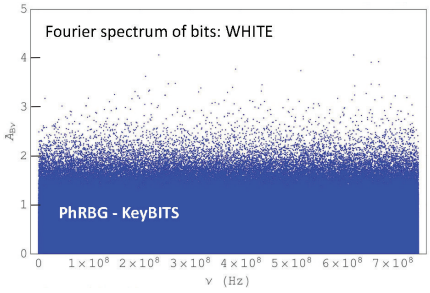

Figure 8. Plot of relative Fourier amplitudes A_ν as a function of the frequency ν. Transforming $(0,1)$ sequences onto $(-1,1)$ sequences allows easy Fourier spectrum analysis that show the "white-noise" character of the output signals.

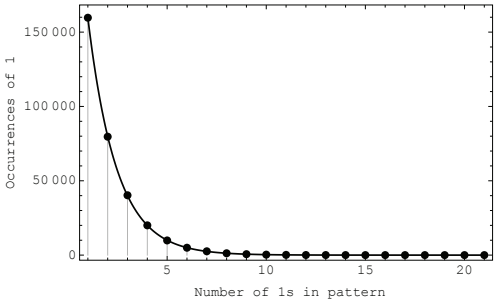

Figure 9. Histogram of 1s. Dots are obtained from 1,277,874 bits obtained and the solid line is the fit to $c = 319018 \pm 356$ and $\epsilon = -0.003 \pm 0.003$.

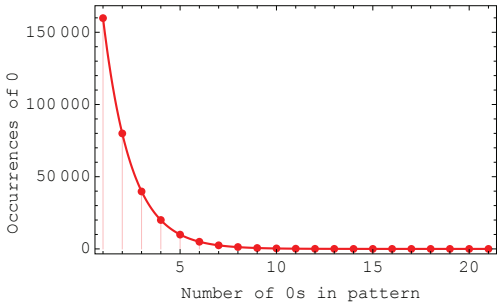

Figure 10. Histogram of 0s. Dots are obtained from 1,277,874 bits obtained and the solid line is the fit to $c = 319880 \pm 193$ and $\epsilon = -0.003 \pm 0.002$.

References

[1] F. Grosshans and P. Grangier, Continuous Variable Quantum Cryptography Using Coherent States, Phys. Rev. Lett. **88**, 057902 (2002).

[2] G. A. Barbosa, Fast and secure key distribution using mesoscopic coherent states of light, Physical Review A **68**, 052307 (2003).

[3] G. A. Barbosa and J. van de Graaf, Untappable Key Distribution System: a One-Time-Pad Booster, Enigma - Brazilian Journal of Information Security and Cryptography, Vol. **1**, No. 2, 16 (2015)

[4] G. A. Barbosa, arXiv:quant-ph/0510011 v2 16 Nov 2005 and arXiv:quant-ph/0705.2243 v2 17 May 2007.

[5] G. A. Barbosa, Harnessing Natures Randomness: Physical Random Number Generator, Enigma - Brazilian Journal of Information Security and Cryptography, Vol. **1**, No. 1, 47 (2014).

[6] C. H. Bennett, G. Brassard, C. Crepeau, U. M. Maurer, Generalized privacy amplification, IEEE Transactions on Information Theory **41**, 1915 (1995)

[7] Jeroen van de Graaf, Decentralized management of One-Time Pad key material for a group, XIV Simpósio Brasileiro em Segurana da Informação e de Sistemas Computacionais SBSeg 2014 - Brazil.

[8] The PhRBG implementation was carried out by a team from Universidade Federal de Minas Gerais and QuantaSEC Consulting, Projects and Research in Physical Cryptography Ltd. with support from Ministério da Ciência, Tecnologia e Inovação (MCTI)-Finep(0276/12)-Fundep(19658)-Comando do Exército(DCT)-RENASIC.

[9] RENASIC Reports: KeyBITS Report 3 (Universidade Federal de Minas Gerais and QuantaSEC).

Geraldo Alexandre Barbosa – PhD (Physics) / University of Southern California, 1974. Areas of work: Quantum Optics and Condensed Matter (Theory and Experiment), Physical Cryptography. 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.

Future Internet and Reconfigurable Computing: Considerations on Flexibility and Security

D. G. Mesquita and P. F. Rosa

***Abstract*—The Future Internet is expected to support services in both existing and new scenarios, in terms of mobility, quality, scalability and security, among other. In this work we present how Reconfigurable Computing (RC) may contribute to build Future Internet (FI) flexibility and security. Therefore, we discuss some aspects of FI initiatives that can be addressed by Reconfigurable Computing. Then we show some features of the Reconfigurable Computing enabling technology – FPGA – which can help to build a more flexible and secure Future Internet. The concluding remarks concern the need to bring together FI and RC researchers.**

***Keywords*—Future Internet Architecture, Reconfigurable Computing, Security.**

I. Introduction

THE Internet growing success is due, in part, to its achitectural simplicity. Designed in the 70's and consolidated in 1981 with the request for comments 791 which has standardized IPv4 - the Internet Protocol version 4 [1], the network of networks has kept, through time, the flexibility needed to incorporate new technologies in order to support a whole myriad of applications.

IPv4 initial specifications established its independence from the underlying layers, and also concerning the host architecture, as well as universal connectivity through entire network, point-to-point acknowledgements and standardized application protocols. Another success motive which should be highlighted is the Berkeley University decision of adopting the TCP/IP implementation in its 4.2 BSD Unix in 1983, and make its source code available as public domain software.

However, the TCP/IP idealizers could not foresee the reach which this network would achieve in the next decades. As can be seen in Fig. 1, in 1970 there were only 9 hosts on the Internet, while the current number is significantly greater.

If, initially, the concern was to interconnect a few computers in research centers, today, 35 years after the RFC:791, the Internet must cope with mobile devices and with the Internet of Things, where devices can "talk" to each other without direct human intervention. Considering the numbers related to this phenomenon, in 1985 there were 1,000 hosts connected, in 1995 it was 3,000,000 [3] and, today, there is approximately one billion hosts linked to the Internet [4]! Fig. 2 helps to visualize the Internet growth, and gives an idea of the complexity concerning the hosts interconnection of such numerous devices, especially if compared to Fig. 1.

D. G. Mesquita, Universidade Federal do Pampa (Unipampa), Santana do Livramento, RS, Brasil, mesquita@unipampa.edu.br

P. F. Rosa, Universidade Federal de Uberlândia (UFU), Uberlândia, MG, Brasil, frosi@ufu.br

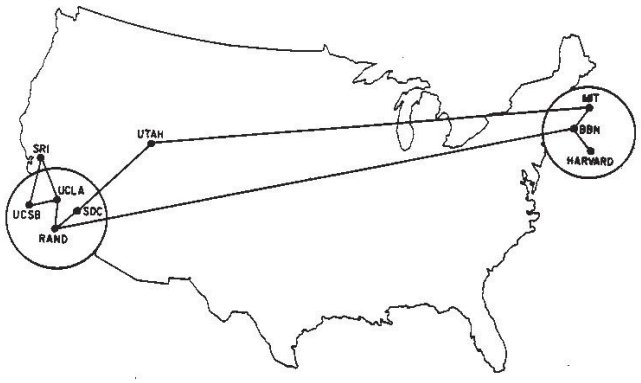

Figure 1. The Internet in 1970. Source: [2]

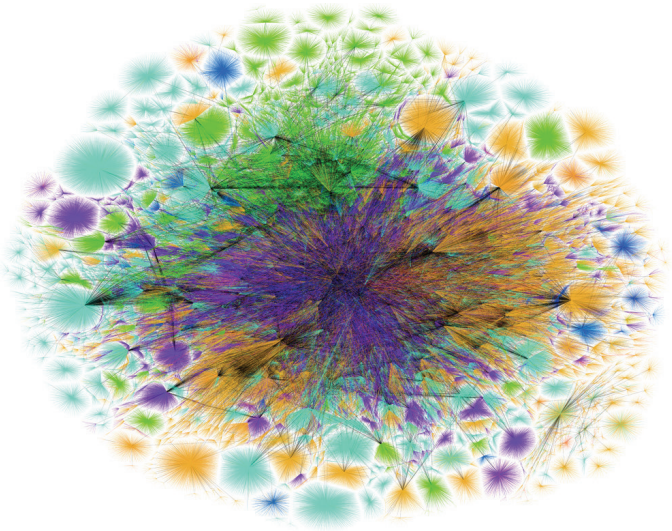

Figure 2. The Internet in 2015. Source: [5]

These numbers leads to the first IPv4 limitation: the scarcity of network addresses. Because of this limitation, many companies implement the network address translation (NAT) in order to map different private IPv4 addresses into an unique public IPv4 address. This technique has helped in conserving public IPv4 addresses, but also have some drawbacks. The NAT does not support the security standards of the network layer, and does not support mapping all protocols of superiors layers. Besides, the address configuration in various devices, whether static or dynamic, should be much simpler than the current way.

Another current Internet problem relates to quality of service (QoS). Despite IPv4 supports QoS, it has only 8 bit of the Type of Service (ToS) field and payload identification to performe it. The IPv4 ToS field has limited functionality and the packet identification is not possible when the datagram packet is encrypted [6].

Talking about cryptography, when the RFC:791 was published, almost none of the current security threats were anticipated. An IPv4 extension, called IPSec [7] was suggested in order to protect data transmitted through Internet, avoiding data visualization or modification by unauthorized people. However, the IPSec is not an Internet built-in protocol and, in many times, its implementations are proprietary.

Meanwhile, with the growing threat of cyber attacks, the security as a whole, and the cryptography - as its support pillar, became a central matter about the future of the Internet.

The scientific and industrial communities are aware of these problems and are proposing changes in the Internet for some time. Several initiatives have been proposed aiming to develop Future Internet Architectures. Some examples are the projects FIBRE (Brasil), FIRE (Europe), NETS-FIA (USA) and AKARI (Japan), among others. More information about this subject are found in Section II.

These initiatives have a lot in common, but it is worth to highlight two important aspects: flexibility and security. Flexibility is necessary both to conduct the transition from the current to future Internet as for the Internet's purpose itself, much more focused on associated services and content type rather than mere packet transmission. Security is another aspect that encourages the development of a new Internet. Cyber attacks are becoming more and more sophisticated and it threatens the economic order. If the current Internet was not conceived to deal with such threats, the next generation of Internet need to have security as a major concern.

Given this context, one technology - now mature - that could be largely used in the very conception of the Future Internet in order to build network flexibility and security has been neglected. This technology is the Reconfigurable Computing, which is based on field programmable gate arrays (FPGA). FPGA have features which can meet both the requirements of flexibility as the security for Internet next generation.

The idea of a reconfigurable device was first conceived by Gerald Estrin in 1960 [8], but only in 1985 they became commercially available [9]. Fig. 3 shows the experiment made as proof of concept by G. Estrin [10]. Besides all technology limitations at that time, the idea of a processing element having a fixed part and another with a flexible hardware, adaptable for any given application, was genius. The hardware flexibility was achieved through the inclusion or removal of specific hardware modules and by reconfiguring its interconnections. If in 1960 the task was manual and fastidious, today, with submicron fabrication technology with a tremendous evolution of the Electronic Design Automation (EDA) tools, FPGA have became an unavoidable solution, when searching for a good compromise among performance, low cost, low power, flexibility and security

Today there is FPGA based systems in sectors such as aerospace, automotive, defense, industrial automation and data-

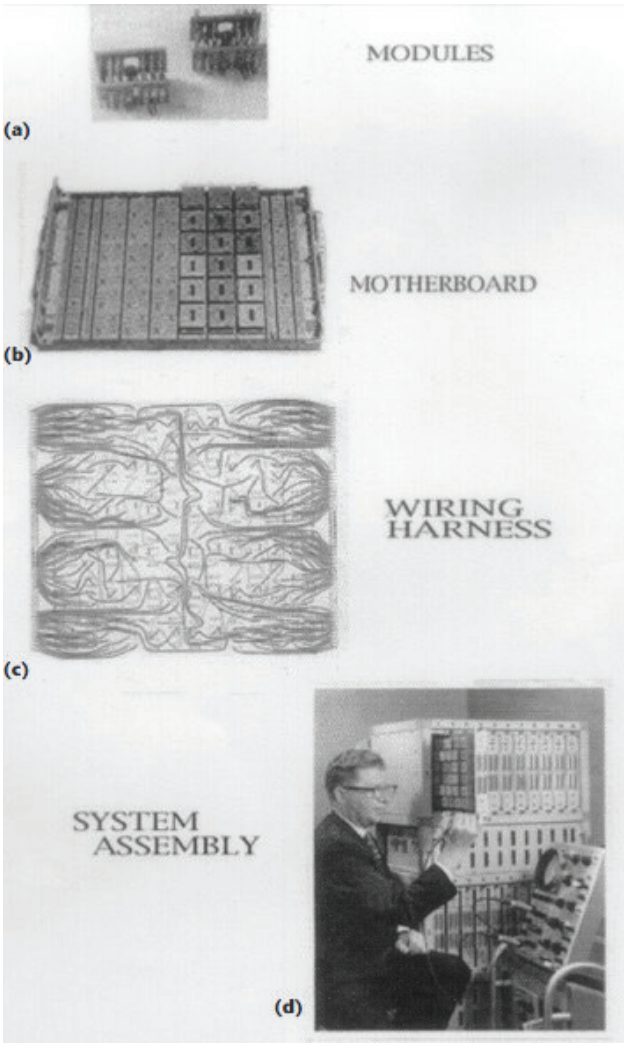

Figure 3. Basic reconfigurable modules (a), motherboard (b), wiring harness for the motherboard (c), and system assembly constructed for the supervisory control and transfer path between the fixed and variable structure computers. The author is shown using an oscilloscope probe to observe electrical activity of the system assembly (d). Source: [10]

centers, among many others. There are even FPGA prototyping boards designed specifically to support network research and development [11], [12].

This article aims to approach these two investigation fields Future Internet and Reconfigurable Computing, discussing how this approximation could lead to a next generation of networks more flexible and secure.

In order to do this, the article brings in its second Section an overview of Future Internet initiatives, with some project examples and a brief discussion about how flexibility and security are important to these enterprises. On the third Section, this article argue about the enabling technology of Reconfigurable Computing: the FPGA. The third Section shows briefly how FPGA works and highlights how its architecture helps to achieve flexibility and security. Finally, the last Section draws a roadmap in the direction of a convergence between researchers in "Future Internet"and Reconfigurable Computing, so the next generation of network can be more flexible and secure.

II. Future Internet Architecture

As mentioned before, there are many Future Internet Architecture initiatives around the world. In this Section I present an overview of a few of those initiatives.

A. Future Internet Research and Experimentation

FIRE (Europe): Future Internet Research and Experimentation is an initiative launched and funded by the European Commission that has been growing since its inception in 2010 with the ambition of being Europe's Open Lab for Future Internet research, development and innovation. One (of many) interesting aspects of the FIRE is the importance given to experimentation as development methodology. The white paper [13], a document on the FIRE initiative, reports some experiments that pushes the knowledge frontiers concerning connectivity. Among the examples there are the investigation of Facebook on economics of privacy; the Netflix experimentation platform to ensure optimal video streaming delivery with minimal playback interruption; Smart Cities using diverse network applications in fields such as transport, energy and environmental management.

Those well succeeded development strategies based on experimentation led the FIRE organizers to adopt a research methodology named experimentation-driven. Another interesting characteristic of the FIRE initiative is the engagement of the industry partners both in financing as in developing projects. Still in [13], the security theme is mentioned many times, from standards for mobile communication untill security and privacy in smart spaces, passing by aspects of trustworthiness, dependability and border control in autonomous cooperative robots.

B. Future Internet Brazilian Environment for Experimentation

Helped by the European Commission, Brazil has developed an experimentation environment - a testbed - which works as a large scale laboratory. The goal of this testbed is to serve as research and development infrastructure so students, researchers and industry may test new network applications and architectures. The project is named Future Internet Brazilian Environment for Experimentation (FIBRE)[1], and it is composed by eleven experimentation islands, sheltered in universities and research institutes. Each experimentation island has a set of equipments which supports experiments in both wired networks as wireless. Those islands are connected by an network on the brazilian RNP backbone, comprised of two network separate layers: a control plane and an experiment plane [14]. The FIBRE community is active, with recent publications, as [15] and [16]. Some projects related to FIBRE include OpenFlow [2] and Software Defined Networks (SDN). In the context of FIBRE, these concepts are related to FPGA implementations, as can be seen in Fig. 4.

[1]http://www.fibre.org.br/

[2]https://www.opennetworking.org/sdn-resources/openflow

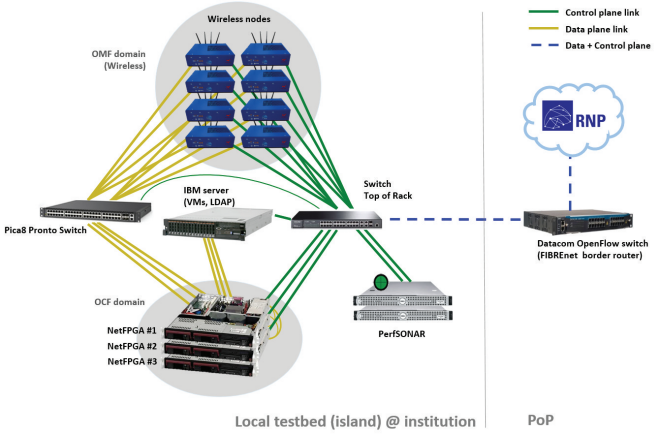

Figure 4. A hardware view of a typical FIBRE island. Note the NetFPGA as a part of the infrastructure. Source: [17]

C. NETS-Future Internet Architecture

NETS-FIA (USA): Recognizing the need for a secure and highly dependable information technology infrastructure and building on NSF's on-going investments in network science and engineering, the Directorate for Computer and Information Science and Engineering (CISE) has formulated this program to stimulate innovative and creative research to explore, design, and evaluate trustworthy future Internet architectures. The NETS-FIA objective is to engage the research community in collaborative, long-range, transformative thinking - unfettered by the constraints of today's networks yet inspired by lessons learned and promising new research ideas - to design and experiment with new network architectures and networking concepts that take into consideration the larger social, economic and legal issues that arise from the interplay between the Internet and society [18]. One of the many projects derived from the NETS-FIA initiative is the Named Data Networking (NDN) [19]. The NDN aims to develop a new Internet architecture keeping the strengths of the current one, and addressing its drawbacks. Its main aspect is to name the contents instead its location. For instance, the current Internet secure the data container, while the NDN secures the content. This is an architectural choice which decouples data confidence from hosts confidence. The project studies the technical challenges to be overcomed in order to validate the NDN as Future Internet: routing scalability, network security, content protection and privacy. Thus, as the previous initiatives, the NETS-FIA is strongly based on experimentation and on the prototyping of protocols. Also, there is an emphasis on the security theme.

D. AKARI

AKARI: The japanese AKARI[3] Architecture Design Project aims to implement the basic technology of a new generation network, developing a network architecture and creating a network design based on that architecture. Its philosophy is to pursue an ideal solution by researching new network

[3]The AKARI project aims to be "A small light (akari in Japanese) in the dark pointing to the future."

architectures from a clean slate without being prevented by existing constraints. The project principles are "crystal synthesis", "reality connected"and "sustainable and evolutionary". The explanation follows:

- *Crystal synthesis* means that the project must remain simple, even when integrating different functionalities.
- *Reality connected* stand for to separate physical and logical infrastructures.
- *Sustainable and evolutionary* represents the properties of self-organization and self-distribution, being flexible and open to future changes.

Projects related to AKARI deal with different aspects networks, and among them stand out those concerning data-centric networking systems [20].

Although the AKARI project was discontinued in 2013, the National Institute of Information and Communications Technology (NICT), from Japan, continued to stimulate research on the post-Internet network, whose goals include addressing current network issues such as reliability and security [21]. According to the authors, this new network must also be flexible and sustainable.

E. Section Summary

There are two major concepts standing out from the subsections above: flexibility and security.

Flexibility is needed to perform experimentations, but also to support future changes on the Internet architecture, so it can evolve. Nowadays, the flexibility is deeply related to SDN and to the data-centric networking systems. Security is essential to the economic success of any network [22]. As in the current Internet almost all security is performed at the application level, the Future Internet must ensure data content at lower levels. Fig. 5 depicts the concern with these aspects.

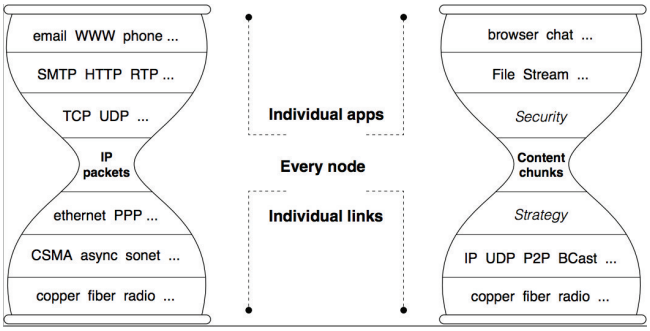

Figure 5. The Internet and the NDN hourglass Architecture. Source: [23]

Fig. 5 compares the current Internet (at the left side) with the next generation of Internet proposed by the NDN team (at the right site). The focus is no longer at the packets communication, but changes to content distribution. Rather than “fixed” protocols, the Future Internet deal with strategies in order to guarantee the data arrival at its due destination. Concerning security, a neglected feature at the original Internet architecture, receives a spotlight at the Future Internet Architectures as proposed by the Named Data Networking team.

The next Section presents the FPGA architecture characteristics that may contribute to achieve flexibility and security (and performance) in projects concerning Future Internet Architectures.

III. Reconfigurable Computing

Reconfiguable Computing is a relatively new computational paradigm which fills the gap between the software flexibility and the specific hardware performance [24]. As mentioned at the Introducion, the reconfigurable computing enabling technology is the FPGA. But unlike has been shown in Fig. 3, the current FPGA are fabricated in nanometric scale and the process of design and prototyping is all assisted by sophisticated Electronic Design Automation (EDA) tools, making possible a short time-to-market and a longer product life-cycle. Fig. 6 depicts the position of systems based on FPGA regarding those based on CPU (microprocessors) and those based on Application Specific Integrated Circuits (ASICs). As can be seen in this abstraction, FPGAs fill a gap between performance and flexibility. Yet, if we change the Y axis from flexibility to energy efficiency, the positions remain the same.

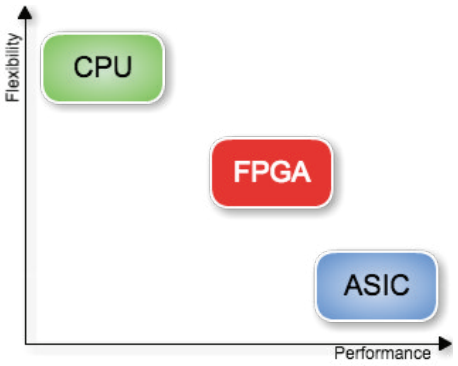

Figure 6. FPGA bridging the gap between microprocessors and application specific processors. Source: Adapted from [25]

Another important factor favouring the FPGA is its cost. Although for parallels applications its performance approaches to the ASIC, as the FPGA is an off-the-shelf product, there is not the Non-Recurring-Engineering (NRE) costs, tipically high for ASIC projects. Moreover, as the design time for FPGA is only a fraction of those related to ASIC, the development costs of systems based on FPGA are significantly lower than those associated with ASIC. Meanwhile the FPGA time-to-market is shorter [26], fact that can improve the profitability of those who adopt this technology. However, perhaps the main interest concerning FPGA is its reconfigurability, since this feature allows the hardware to be modified at the logic function level. This hardware flexibility may lead to a longer product life-cycle, since the hardware functionality can be upgradable without change any physical element.

In order to support these assertions, the next subsection gives an overview of the FPGA architecture. After, in the following two subsections two themes are discussed: (i) topics about FPGA used in rapid prototyping of digital systems, specially concerning networking and (ii) FPGA used to improve the security infrastructure, in particular for cryptographic applications. The Section ends by establishing the link between

the topics mentioned right above and those covered at the end of Section II: network flexibility and security.

A. FPGA

This subsection is a brief overview about FPGA. If you are interested in a detailed description of its functionality, please refer to [27]. For an extended survey on Reconfigurable Computing, We recommend to read [28]. For an up-to-date summary of the knowledge-base on FPGA architecture, tools and systems I suggest the article [26]. Finally, there are Reconfigurable Systems that are not based on FPGA. These are not covered by this work, but are discussed on [25].

FPGA are digital integrated circuits whose functionality is not pre-defined and can be changed after its fabrication. Basically it is an array of programmable logic elements, interconnected by a mesh of also programmable routing resources, which functionality can be field programmable. Fig. 7) depicts the FPGA basic architecture in a high level of abstraction. Depending on the fabrication technology - anti-fuse, Electrically-Erasable Programmable Read-Only Memory (EEPROM), static RAM (SRAM), magnetic RAM (MRAM) - the FPGA can be programmable once or many times. The most common is the SRAM based FPGA, which can be reconfigured numerous times [27]. The magic behind the FPGA is the capability of receiving any functionality that can be algorithmically described. This is possible because the logic elements composed by look-up tables (LUTs), multiplexers, flip-flops and some logic gates. Each LUT can be used as a small RAM, or, most commonly, as a logic function. Typically a FPGA has thousands of logic elements, communicating through interconnection resources composed by wires and muxes. Isolated, a logic element cannot do much, but with the rich interconnection, complex logic functions can be performed. The designer may describe the FPGA functionality using a hardware description language (tipically VHDL or Verilog). The codification style can be algorithimically (behavioural) or structural. The most common is a mix of both, emphasizing the structural one (for the synthesis sake). Once the description is finished, an EDA tool helps to transform it into a bitstream, which is the configuration file format for the FPGA. This process is called synthesis. There are intermediary steps, but most of it is automated. Fig. 9 (left) helps to visualize this process.

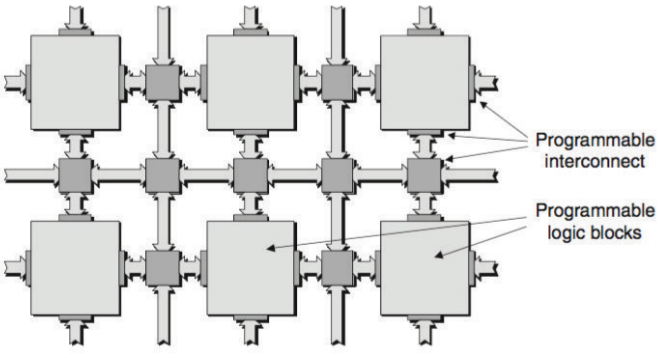

Figure 7. A simplified view of a generic FPGA. Source: [29]

Fig. 7 is a mere abstraction, because, actually, FPGA architecture include I/O pins, fast interconnection network, clock trees, multiply and accumulate (MAC) elements, memory blocs, small DSPs, and even, sometimes, one or more microprocessors (softcores or hardwireds). By adding some general purpose processing as small CPUs and DSPs into the FPGA matrix, the Reconfigurable Computing is improved, as it approaches the best of both worlds. Thereby, it is possible to see the FPGA as a co-processor of a conventional microprocessor (which is, in its turn, into the FPGA). It is also possible to imagine the FPGA as processing unity attached to the microprocessor via shared memory. On a third scenario, the FPGA can act as a stand-alone processor. Fig. 8 depicts those three possibilities of Reconfigurable Computing.

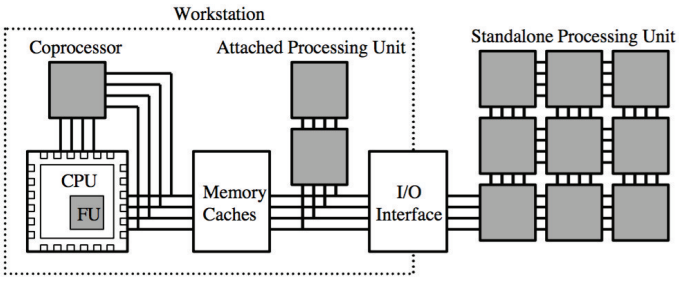

Figure 8. Different coupling scenari between Reconfigurable Computing (RC is shaded) and conventional computing (CPU). Source: [30]

However, each project must carefully analyse what is more suitable to its goals, because each scenario has strengths and weakness.

On the one hand the tighter the integration between the reconfigurable hardware and the microprocessor, more often the reconfigurable fabric can be used by a given application, due to the low communication overhead. On the other hand, the hardware is unable to operate for large time slices without the CPU intervention. Often in this case, the amount of logic elements available to the application is reduced, as part of it is used to implement the CPU itself. The more loosely coupled with the CPU, more room to explore the application parallelism, but with a penalty of increasing the communication overhead.

Regarding design costs of a digital system, the design flow for FPGA is simpler than that for ASIC, as can be seen in Fig. 9. Although both begin similarly, with the functional specification phase, the hardware description (with VHDL or Verilog) and a behavioural simulation, the ASIC flow needs static timing analysis and equivalence checks with the foundry parameters. Also, the ASIC flow must include verification of internal deep sub-micron effects (verification on second and third order effects). Concerning the FPGA these verification are previously made by the manufacturer so the end user (in this case the digital designer) do not to bother with it.

Not to mention that, once the flow is exectute for FPGA and the bitstream is generated, just download it into the FPGA on a prototyping board to test it. If something goes wrong, just go back to the HDL phase and start again.

In the case of the ASIC flow, it is not so easy. Once finished the system verification using EDA tools, the physical layout

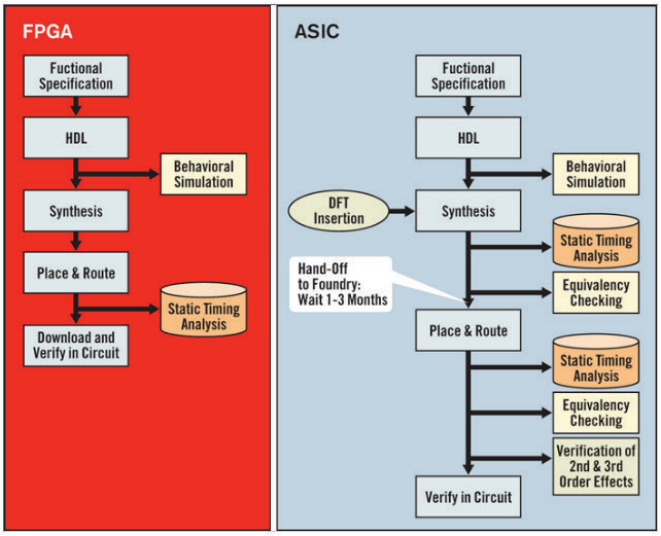

Figure 9. Comparison between FPGA and ASIC design flow. Source: [31]

must be done, and after that there is the post-layout technology checks. If everything is ok, the layout can be sent to the manufacturer (frequently overseas). Once the chip done, it is sent to the encapsulator, and then, sent back to the original designer in order to test. If the test fail, all process must restart. As you can see, it is an expensive process.

To illustrate an FPGA prototyping board and its capabilities, Fig. 10 shows a low-cost board, where a whole digital system can be implemented and tested. Once the goals achieved, the system can be sized to fit its needs, in a specically designed board with only the needed resources, with and FPGA equivalent to that on the prototyping board.

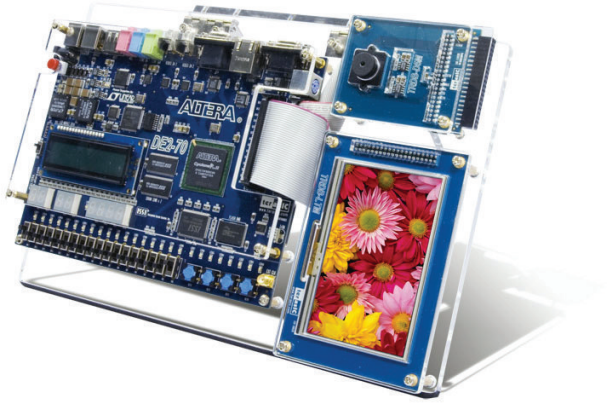

Figure 10. An Altera FPGA Prototyping Platform. Source: [32]

Fig. 10 depicts the I/O richness that can be attached to a FPGA, in the sense of prototyping a variety of applications. For instance, the camera and the display could be used as I/O for a facial recognition on a access control system to a mainframe room, where the heaviest processing algorithm are computed on the FPGA.

Such flexibility, and low implementation cost, besides the high performance when compared to purely software solution, indicate the FPGA as an ideal platform to prototype digital systems. The next subsections gives some examples network and security experimentations using FPGA.

B. FPGA and Networking

A recurring subject in developing the Future Internet is the Software Defined Network (SDN) model. In the SDN architecture, the control plan and the data plan are decoupled, the network inteligence and status are locally stored, and the underlying network infrastructure are abstracted from the application [33]. In short, SDN emphasizes the following characteristics:

1) Decoupling networking hardware and software;
2) Centralized network view and control;
3) Open interfaces between devices on the control plan and the data plan; and
4) Programmability by external applications, i.e., operators, independent sofware vendors and users - not just equipment manufacturers.

Decoupling networking hardware and software allows for centralization of the control portion (named the control plane) while keeping the actual packet forwarding function (the forwarding plane) distributed across many physical network switches. This provides a means to configure, monitor, troubleshoot, and automate a large network built of many discrete hardware components as a single network "fabric."

The centralized control plane can then enable new or different forwarding behaviors and broader, more precise control of traffic flow. Many products that encompass data center fabrics and flow control methods such as OpenFlow leverage this facet of SDN.

The switch required by the SDN model must process packet flows in a performing and secure way. So, basically, there are three elements to consider: performance and security, but also flexibility as preconized by the OpenFlow switches.

The reference [34] is a report on initiatives about SDN switches using different approaches: multicore CPU/GPP; Network Processing Units (NPU) / Network Flow Processors (NFP); PLD/FPGA; Application-Specific standard products; and ASICs. Fig. 11 relates them in a plan composed by the Programability (flexibility) axe and Performance (in Gb/s) axe. We can see a close relation with Fig. 6, where FPGA appears as a half-way between performance and flexibility. This finding is also supported by article [35].

One concrete example of convergence between Reconfigurable Computing researchers and network researchers is the NetFPGA project [36] from the University of Cambridge. This project, based on FPGA technology, provides software, hardware and community as a basic infrastructure to simulation and testing high-speed networks.One key point of this project is to maintain the platform as an open-source project, allowing the reuse of building blocks across various research projects. The current board (NetFPGA SUME) supports up to 100GB/s applications. The NetFPGA SUME shown in Fig. 12 uses a high-density Virtex 7 690T FPGA, supporting high-speed serial interfaces, and its format permits user-expansion.

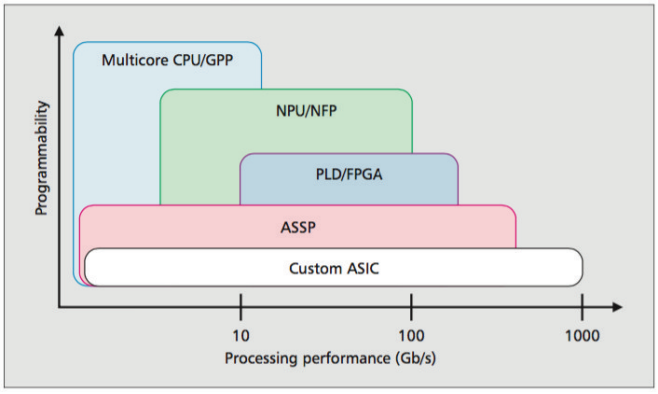

Figure 11. Network processing: performance x programmability Source: [34]

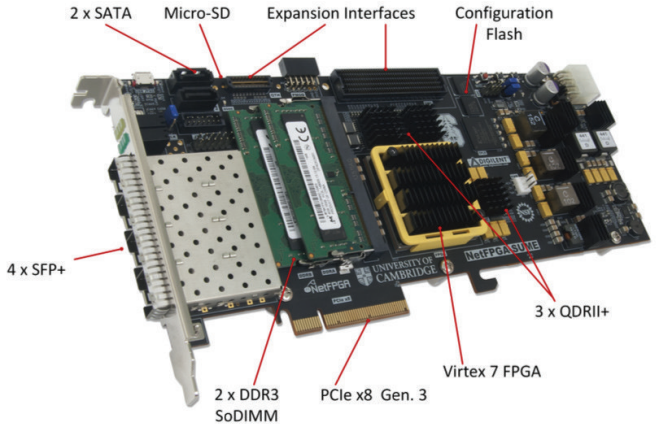

Figure 12. NetFPGA SUME board. Source: [36]

Acording to the project maintainers[4], there currently are more than two hundred scientific publications of results on network research makin use of NetFPGA, across 150 research groups in more than 40 countries. These project includes an open-source network tester [37], a high-resolution hardware based packet capture [38], an evaluation of native load distribuition of ARP-path in data centers [39] and a framework for trust and policy management for a secure internet [40]. In all these works, the common point is the claim that is through experimentation on a real hardware, rather than simulations, is that researchers can reach more assertive conclusions.

Another significant example is the development process of a high availability routing protocol - called HARP [41], researched on the context of the Future Internet project called Entity Title Architecture (ETArch) [42].

At the work presented in [43], the author has found several problems at the VRRP (Virtual Router Redundancy Protocol) that could lead to sensible downtime intervals at the network of a major telecom group. He identified the split brain and the no-brain situations as causes to these downtimes. Then[43] proposes n extension to the VRRP in order to address the problems found. However, when trying to implemente the work in [43], we found some inconsistencies, unobservable without physical experimentation. By using some low-cost FPGA we were able to recreate the virtual router scenario and to observe the failures in VRRP. By constructing a hardware model to validate high-availability protocols, we achieved to develop and test the HARP, solving the no-brain and the split-brain situations. Fig. 13 depicts the solution. The HARP protocol is essentially a finite state machine (at the center of Fig. 13). The surrounding blocks are part of the validation scheme. More information about HARP protocol can be found in [44].

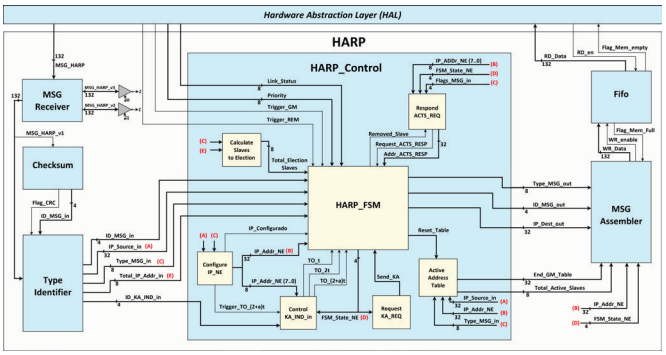

Figure 13. HARP Architecture. Source: Self.

As illustration, the Fig. 14 shows the FPGA boards where the HARP was prototyped. Each board acts as a router, and all three composes a virtual router. Many different scenarios were described and experimented, giving us confidence in our both theoretical and practical findings.

Figure 14. HARP prototype experimentation. Source: Self.

Both situations, the NetFPGA community and numerous research papers, and our own results, demonstrate the potential of FPGA as platform to implement future internet solutions, in particular in cases where the experimentation is part of the research and development methodology, as foreseen in FIBRE, NFS-NET, FIRE and AKIRA initiatives.

C. FPGA and Network Security

As mentioned before, FPGA has a regular and homogeneous internal architecture, with a rich interconnection network. These are important ingredients to compute parallel algorithms, as it is the case of cryptography or pattern recognition. Such algorithm classes are in the basis of network security solutions, as authentication (through public key cryptography),

[4]http://netfpga.org/site/\#/publications/

packets classification, or intrusion detection, only to mention a few examples.

The networks composing the Internet, the current one or the future, need security solutions with the following features [45] :

- Real-Time Protection. For an effective protection mechanism, it is important to achieve line-speed data processing with an affordable cost.
- Flexible Updating. As attackers are creative and their methods are continuously evolving, the protection system must also be adaptable.
- Scalability. This is a critical concern for actual deployment. Many approaches working in small networks have their performance deteriorated in real scale networks. Due to its regular architecture, FPGA can be added in a scalable way.

The features above are well addressed by FPGAs. Its intrinsic parallelism can meet the real-time requirements for packets classification, for instance, at a lower cost than ASICs. Although flexibility also can be achieved with software in CPU or NPU solutions, those does not necessarily meet the performance conditions, meanwhile the ASIC are fast, but expensive and not flexible at all. Once again, FPGA emerge as the alternative which fills the gap among all architectural approaches.

Still regarding these features - performance and flexibility - there is some examples that is worth to take a look:

- *Packet classification* this is one of the fundamental challenges in designing high speed routers. It enables the router to support firewall processing, quality of service differentiation, policy routing and other value added services. When a packet arrives at a router, its header is compared with a set of rules. Each rule can have one or more fields and their associated value, and an action to be taken if matched. To perform the needed comparison to each packed, FPGA has been successfully used [46], [47], [48]. In [49] we find a comparison between packet classification implementations in FPGA, GPP and GPU, with an impressive advantage to FPGA.
- *Pattern matching* is the most important and the most computationally intensive component of a network intrusion detection systems (NDIS). NDIS operates by monitoring network packets and matching them against user-defined rule set. Many successful works has been done using FPGA to perform pattern matching [50], [51], [52]. [53] brings us a comprehensive survey about pattern matching for deep packet inspection, including FPGA implementations.
- *Distributed Denial of Service (DDOS)* and Internet worms attacks are the two major security theats to the network infrastructure [54]. In [55] active scans and filters has been implemented, in order to detect Internet worms and viruses at a multi-Gigabit/second rates, using FPGA. In [56] the authors achieved 400Kilo Packets filtering in a anti-DDOS system based on the NetFPGA 1G. In [57] the authors implemented a real-time detection against DDOS and IDS (Intrusion Detection System) based in FPGA, achieving a throughput of 2 Gigabits per second.
- *Cryptography* is the fundamental component for securing the Internet traffic. However, cryptographic algorithms impose high processing time and efforts that can be a bottleneck to high-speed networks. It has been demonstrated the advantages of using FPGA for cryptographic applications. [58] shows a fast Elliptic Curve Cryptography in FPGA. In [59] the authors present three reconfigurable hardware architectures for modular exponentiation, the main function of the RSA cryptosystem.In [60] we found an efficient FPGA implementation of the AES private-key cryptographic algorithm.

However, FPGA implementations of cryptographic primitives deserve some attention concerning some potential vulnerabilities.

First of all, designers should be advised that any hardware implementation of cryptographic primitives also may be vulnerable to hardware-specific attacks. Even choosing to implement a computationally secure cryptographic algorithm (such as RSA [61] or ECC [62]), cryptographic hardware modules may suffer with direct or side-channel attacks [63], [64]. Direct attacks may attempt to change the implemented logic. To counteract it, FPGA vendors have developed tamper-evidence and tamper-resistance techniques [65],[66], [67]. Concerning SCA attacks, there also some techniques the designer must be aware to use [68].

Fig. 15 summarizes hardware threats and countermeasures. At the middle, the figure shows the abstraction levels, from the lower level, which is the fabrication technology up to the application level. Each level has known threats and corresponding countermeasures.

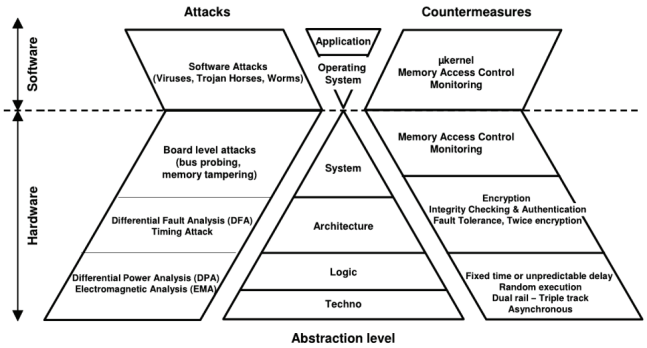

Figure 15. Hardware threats and countermeasures by abstraction level. Source: [69]

For instance, when implementing security modules in FPGA, we are working at the Architecture and Logic levels, which are vulnerable to Side Channel Attacks, such as Differential Power Analysis [64] and Timing Attacks [63]. These threats are called SCA because rather than attack cryptographic algorithms vulnerabilities, the attackers observe "leaked"information of the cryptographic circuit such as power consumption or the amount of time to compute one specific cryptographic function.

So we must worry about integrity checking, bitstream encryption, fault tolerance, and some low level countermeasures

such as unpredictable random (or fixed time) execution, dual-rail or asynchronous implementation. However, if carefully designed, cryptographic modules in FPGA can be efficient (in terms of performance, cost and flexibility) and secure.

In [70] and [71] we demonstrate the efficiency of FPGA to implement Montgmomery Modular Multiplications [72], essential to public key cryptosystems such as RSA. In [73] we show how to counteract DPA attacks with Reconfigurable Computing using a leak resistant arithmetic and architecture. Fig. 16 shows an ovewview of this approach.

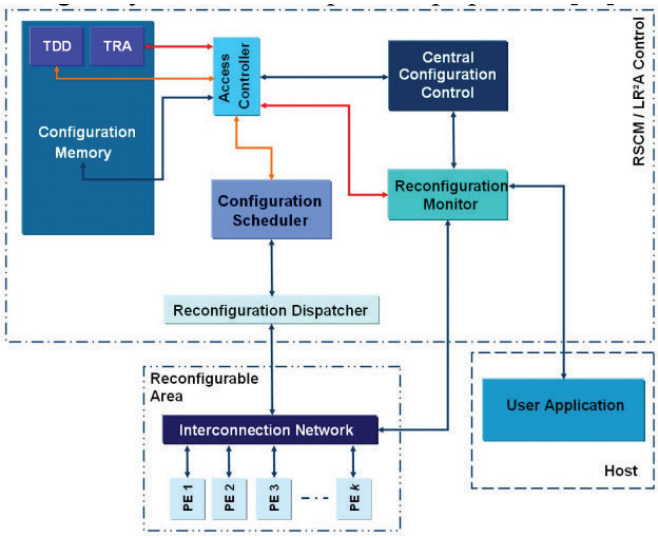

Figure 16. A leak resistant reconfigurable architecture. Source: [73]

The "user application"sends a request to the reconfigurable controller module in order to specify its needs in terms of cryptography. The reconfigurable monitor talks to the central configuration control to verify if the requested function is available in the configuration memory. If so, the configuration scheduler searches the corresponding cryptographic function in the memory and dispatches it to the reconfigurable area, in order to be executed. On the example, the cryptographic module is a Leak Resistant Arithmetic to perform Montgomery modular multiplications capable of masking power consumption and computing time so no DPA or Differential Fault Analysis (DFA) can be performed against the circuit.

These few works illustrates the capabilities of FPGA to cope with network cryptographic needs.

D. Section Summary

The examples in this section shows that Reconfigurable Hardware is already in use for both network and security applications, being useful for rapid hardware prototyping but also as market solution on these fields. However, the Future Internet initiatives do not mention explicitly the use of Reconfigurable Computing in its directive, nor its related projects give attention to this possibility.

IV. Conclusions

In this article we try to logically build the argument that it is necessary to approximate the areas of computer architecture and computer networks, or more specifically bridge the gap between research in Reconfigurable Computing those in the Future Internet Architectures. We show through a brief survey full of successful examples, such as some of the needs and future internet objectives can be met through the reconfigurable computing.

Finally, some considerations on lessons we learned in building this article.

First, we assume that to approach the research of Future Internet and Reconfigurable Computing, need to adopt an approach that, in parallel, is top-down and bottom-up.

On one hand we think the top-down approach in the sense that researchers in both areas perform specific meeting to discuss joint research. It is also necessary to articulate actions so that there is the funding for such initiatives, preferably through public-private partnerships.

On the other hand, the bottom-up approach means that complexity of current research and development has no room for compartmentalized knowledge. The undergraduate curricula should emphasize pedagogical approaches based on multidisciplinary problem solving [74], [75]. After all, accustoming students to think in a complex way, we increase the chances of developing the critical mass necessary for troubleshooting problems composed of variable increasingly numerous and diverse in nature. According to Edgar Morin[76]:

> *"We need a kind of thinking that reconnects that which is disjointed and compartmentalized, that respects diversity as it recognizes unity, and that tries to discern interdependencies. We need a radical thinking (which gets to the root of problems), a multidimensional thinking, and an organizational or systemic thinking."*

As shown in the examples brought previously, the merge of fields Reconfigurable Computing and Future Internet is feasible. However this integration needs to be intensified and adopted as early as possible so that, from the specification phase of the solutions for the Internet of the Future, already consider Reconfigurable Computing as architectural alternative for implementation.

Acknowledgments

The authors would like to thank the Brazilian CAPES agency for support the project "Rede de Cooperação Universitária para Ensino Superior e Pesquisa Avançada Científica e Tecnológica em Sistemas Eletrônicos e Interativos aplicados à Defesa Nacional"CAPES-PRODEFESA, CAPES-PRODEFESA number 23038.009307/2013-77.

References

[1] J. Postel, "Internet protocol," Internet Requests for Comments, RFC Editor, RFC 791, September 1981. [Online]. Available: https://tools.ietf.org/html/rfc791

[2] C. S. University. (2016) Figure of web. [Online]. Available: http://som.csudh.edu/cis/lpress/history/arpamaps/press6.jpg

[3] S. Shenker, "Fundamental design issues for the future internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, Sept 1995.

[4] CIA. (2016) Country comparison: Internet hosts. [Online]. Available: https://www.cia.gov/library/publications/the-world-factbook/rankorder/2184rank.html

[5] T. O. Project. (2016) Figure of web. [Online]. Available: http://www.opte.org/the-internet/

[6] M. Sailan, R. Hassan, and A. Patel, "A comparative review of ipv4 and ipv6 for research test bed," in *2009 International Conference on Electrical Engineering and Informatics*, vol. 02, Aug 2009, pp. 427–433.

[7] S. Frankel and S. Krishnan, "Ip security (ipsec) and internet key exchange (ike) document roadmap," Internet Requests for Comments, RFC Editor, RFC 6071, February 2011. [Online]. Available: https://tools.ietf.org/html/rfc6071

[8] G. Estrin, "Organization of computer systems: The fixed plus variable structure computer," in *Papers Presented at the May 3-5, 1960, Western Joint IRE-AIEE-ACM Computer Conference*, ser. IRE-AIEE-ACM '60 (Western). New York, NY, USA: ACM, 1960, pp. 33–40. [Online]. Available: http://doi.acm.org/10.1145/1460361.1460365

[9] J. Rose, A. E. Gamal, and A. Sangiovanni-Vincentelli, "Architecture of field-programmable gate arrays," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 1013–1029, Jul 1993.

[10] G. Estrin, "Reconfigurable computer origins: the ucla fixed-plus-variable (f+v) structure computer," *IEEE Annals of the History of Computing*, vol. 24, no. 4, pp. 3–9, Oct 2002.

[11] Digilent. (2016) Netfpga sume virtex 7 development board. [Online]. Available: goo.gl/0b4C6P

[12] Terasic. (2016) De5net fpga development kit. [Online]. Available: goo.gl/o3pWNM

[13] M. Boniface, M. Calisti, and M. Serrano, "Next generation internet experimentation." European Comission, Tech. Rep., June 2016. [Online]. Available: https://goo.gl/kZq0VG

[14] RNP. (2016) Internet do futuro. [Online]. Available: https://goo.gl/w8JjkD

[15] C. Rothenberg, A. Vidal, M. Salvador *et al.*, "Hybrid networking toward a software-defined era," in *Network Innovation through OpenFlow and SDN*, J. Fagerberg, D. C. Mowery, and R. R. Nelson, Eds. Oxford: CRC Press, 2014, ch. 8, pp. 153–198.

[16] L. Ciuffo, T. Salmito, J. Rezende, and I. Machado, "Testbed fibre: Passado, presente e perspectivas," in *Anais do Workshop de Pesquisa Experimental da Internet do Futuro*, vol. 1. SBC, Jun 2016, pp. 3–7.

[17] FIBRE. (2016) Figure of web. [Online]. Available: http://fibre.org.br/infrastructure/resources/

[18] NSF. (2016) National science foundation future internet architecture project. [Online]. Available: http://www.nets-fia.net/

[19] L. Zhang, E. Estrin, J. Burke *et al.*, "Named data networking (ndn) project," NDN Project, Tech. Rep., October 2010. [Online]. Available: http://goo.gl/K9YsV8

[20] M. Murata, "Goals of r&d on new-generation network project," *Journal of the National Institute of Information and Communications Technology*, vol. 62, no. 2, pp. 2–5, Mar 2015.

[21] N. Nishinaga and H. Harai, "Progress and results of new-generation network research and development," *Journal of National Institute of Information and Communications Technology*, vol. 62, no. 2, pp. 1176–1188, March 2016.

[22] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 26–36, July 2011.

[23] N. D. Networking. (2016) Figure of web. [Online]. Available: https://named-data.net/project/execsummary/

[24] S. Hauck, "The roles of fpgas in reprogrammable systems," *Proceedings of the IEEE*, vol. 86, no. 4, pp. 615–638, Apr 1998.

[25] R. Hartenstein, "A decade of reconfigurable computing: A visionary retrospective," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '01. Piscataway, NJ, USA: IEEE Press, 2001, pp. 642–649. [Online]. Available: http://dl.acm.org/citation.cfm?id=367072.367839

[26] R. Tessier, K. Pocek, and A. DeHon, "Reconfigurable computing architectures," *Proceedings of the IEEE*, vol. 103, no. 3, pp. 332–354, March 2015.

[27] S. Brown and J. Rose, "Fpga and cpld architectures: a tutorial," *IEEE Design Test of Computers*, vol. 13, no. 2, pp. 42–57, Summer 1996.

[28] K. Compton and S. Hauck, "Reconfigurable computing: A survey of systems and software," *ACM Computing Surveys*, vol. 34, no. 2, pp. 171–210, Jun. 2002. [Online]. Available: http://doi.acm.org/10.1145/508352.508353

[29] C. Maxfield, *The Design Warriorś Guide tho FPGAs: Devices, Tools and Flows*. Massachusetts, USA: Elsevier, 2004.

[30] K. Compton and S. Hauck, "An introduction to reconfigurable computing," Northwestern University, Tech. Rep., October 1999. [Online]. Available: http://goo.gl/gJgIEv

[31] XILINX. (2016) Figure of web. [Online]. Available: http://www.xilinx.com/fpga/asic.htm

[32] TERASIC. (2016) Figure of web. [Online]. Available: http://www.terasic.com

[33] O. N. Fondation. (2016) Software-defined networking (sdn) definition. [Online]. Available: https://www.opennetworking.org/sdn-resources/sdn-definition

[34] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for sdn? implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36–43, July 2013.

[35] A. Kalyaev and E. Melnik, "Fpga-based approach for organization of sdn switch," in *Application of Information and Communication Technologies (AICT), 2015 9th International Conference on*, Oct 2015, pp. 363–366.

[36] N. Zilberman, Y. Audzevich, G. Kalogeridou *et al.*, "Netfpga: Rapid prototyping of networking devices in open source," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM '15. New York, NY, USA: ACM, 2015, pp. 363–364. [Online]. Available: http://doi.acm.org/10.1145/2785956.2790029

[37] G. Antichi, M. Shahbaz, Y. Geng *et al.*, "Osnt: open source network tester," *IEEE Network*, vol. 28, no. 5, pp. 6–12, September 2014.

[38] Y. E. Kwasi and R. Rojas-Cessa, "High-resolution hardware-based packet capture with higher-layer pass-through on netfpga card," in *2014 23rd Wireless and Optical Communication Conference (WOCC)*, May 2014, pp. 1–6.

[39] G. Ibáñez, J. A. Carral, E. Rojas, and J. M. Giménez-Guzmán, "Evaluating native load distribution of arp-path bridging protocol in mesh and data center," in *2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 3769–3774.

[40] X. Liu, A. Wada, T. Xing, P. Juluri, Y. Sato, S. Ata, D. Huang, and D. Medhi, "Servitr: A framework for trust and policy management for a secure internet and its proof-of-concept implementation," in *2012 IEEE Network Operations and Management Symposium*, April 2012, pp. 1159–1166.

[41] D. Mesquita, R. Oliveira, and P. Rosa, "Harp - high availability routing protocol," 2014, bR Patent BR5120130001056.

[42] J. H. de Souza Pereira, F. de Oliveira Silva, E. L. Filho, S. T. Kofuji, and P. F. Rosa, *Title Model Ontology for Future Internet Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 103–114. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20898-0_8

[43] G. Hashimoto, E. Filho, J. Pereira, and P. Rosa, "High avaliability: A long-term feature in network elements," in *Proceedings of the Fifth International Conference on Systems and Networks Communications (ICSNC), 2010*, aug. 2010, pp. 201 –206.

[44] R. Oliveira, D. Mesquita, and P. Rosa, "Harp: A split brain free protocol implemented in fpga," in *Proceedings of the Ninth Advanced International Conference on Telecommunications, AICT 2013*, vol. 9, jun 2013, pp. 197–203. [Online]. Available: https://www.thinkmind.org/download.php?articleid=aict_2013_9_20_10173

[45] H. Chen, Y. Chen, and D. H. Summerville, "A survey on the application of fpgas for network infrastructure security," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 541–561, 2011.

[46] J. Li, Y. Chen, C. Ho, and Z. Lu, "Binary-tree-based high speed packet classification system on fpga," in *The International Conference on Information Networking 2013 (ICOIN)*, Jan 2013, pp. 517–522.

[47] Y. R. Qu and V. K. Prasanna, "High-performance and dynamically updatable packet classification engine on fpga," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 197–209, Jan 2016.

[48] A. Wicaksana and A. Sasongko, "Fast and reconfigurable packet classification engine in fpga-based firewall," in *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, July 2011, pp. 1–6.

[49] Y. R. Qu, H. H. Zhang, S. Zhou, and V. K. Prasanna, "Optimizing many-field packet classification on fpga, multi-core general purpose processor, and gpu," in *Architectures for Networking and Communications Systems (ANCS), 2015 ACM/IEEE Symposium on*, May 2015, pp. 87–98.

[50] K. Jaic, M. C. Smith, and N. Sarma, "A practical network intrusion detection system for inline fpgas on 10gbe network adapters," in *2014 IEEE 25th International Conference on Application-Specific Systems, Architectures and Processors*, June 2014, pp. 180–181.

[51] W.-S. Jung and T. G. Kwon, "An independently partial pattern matching for content inspection at multi gigabit networks," in *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, vol. 2, Feb 2010, pp. 1574–1579.

[52] T. T. Hieu and N. T. Tran, "A memory efficient fpga-based pattern matching engine for stateful nids," in *2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2013, pp. 252–257.
[53] C. Xu, S. Chen, J. Su, S. M. Yiu, and L. C. K. Hui, "A survey on regular expression matching for deep packet inspection: Applications, algorithms and hardware platforms," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.
[54] M. Cai, K. Hwang, Y.-K. Kwok, S. Song, and Y. Chen, "Collaborative internet worm containment," *IEEE Security Privacy*, vol. 3, no. 3, pp. 25–33, May 2005.
[55] J. W. Lockwood, J. Moscola, M. Kulig, D. Reddick, and T. Brooks, "Internet worm and virus protection in dynamically reconfigurable hardware," in *in '03 Military and Aerospace Programmable Logic Device (MAPLD)*, Washington, DC, Sep 2003, p. E10.
[56] K. Pandiyarajan, S. Haridas, and K. Varghese, "Transparent fpga based device for sql ddos mitigation," in *Field-Programmable Technology (FPT), 2013 International Conference on*, Dec 2013, pp. 82–89.
[57] J.-T. Oh, S.-K. Park, J.-S. Jang, and Y.-H. Jeon, "Detection of ddos and ids evasion attacks in a high-speed networks environment," *International Journal of Computer Science and Network Security*, vol. 7, no. 6, pp. 124–131, Jun 2007. [Online]. Available: http://paper.ijcsns.org/07_book/200706/20070617.pdf
[58] W. N. Chelton and M. Benaissa, "Fast elliptic curve cryptography on fpga," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 2, pp. 198–205, Feb 2008.
[59] N. Nedjah and L. M. Mourelle, "Three hardware architectures for the binary modular exponentiation: sequential, parallel, and systolic," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 3, pp. 627–633, March 2006.
[60] H. W. Kim and S. Lee, "Design and implementation of a private and public key crypto processor and its application to a security system," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 214–224, Feb 2004.
[61] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978. [Online]. Available: http://doi.acm.org/10.1145/359340.359342
[62] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computations*, vol. 48, pp. 203–209, 1987.
[63] P. C. Kocher, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 104–113.
[64] P. Kocher, J. Jaffe, and B. Jun, *Differential Power Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397.
[65] ALTERA. (2016) Anti-tamper capabilities in fpga designs. [Online]. Available: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/wp/wp-01066-anti-tamper-capabilities-fpga.pdf
[66] ——. (2016) Developing tamper resistant designs with xilinx virtex-6 and 7 series fpgas. [Online]. Available: http://www.xilinx.com/support/documentation/application_notes/xapp1084_tamp_resist_dsgns.pdf
[67] A. Seffrin, S. Malipatlolla, and S. A. Huss, "A novel design flow for tamper-resistant self-healing properties of fpga devices without configuration readback capability," in *Field-Programmable Technology (FPT), 2010 International Conference on*, Dec 2010, pp. 291–294.
[68] B. Chevallier-Mames, M. Ciet, and M. Joye, "Low-cost solutions for preventing simple side-channel analysis: side-channel atomicity," *IEEE Transactions on Computers*, vol. 53, no. 6, pp. 760–768, June 2004.
[69] B. Badrignans, J.-L. Danger, V. Fischer, G. Gognat, and L. Torres, *Security Trends on FPGA*. Montpellier, France: Springer-Verlag, 2011.
[70] D. G. Mesquita, G. Perin, F. L. Herrmann, and J. a. B. d. S. Martins, "An efficient implementation of montgomery powering ladder in reconfigurable hardware," in *Proceedings of the 23rd Symposium on Integrated Circuits and System Design*, ser. SBCCI '10. New York, NY, USA: ACM, 2010, pp. 121–126. [Online]. Available: http://doi.acm.org/10.1145/1854153.1854184
[71] D. G. Mesquita, G. Perin, and J. a. B. d. S. Martins, "Montgomery modular multiplication on reconfigurable hardware: Systolic versus multiplexed implementation," *International Journal of Reconfigurable Computing*, vol. 2011, pp. 1–10, January 2011. [Online]. Available: http://dx.doi.org/10.1155/2011/127147
[72] P. L. Montgomery, "Modular multiplication without trial division," *Mathematics of Computations*, vol. 44, no. 170, pp. 519–521, 1985.
[73] D. Mesquita, B. Badrignans, L. Torres, G. Sassatelli, M. Robert, and F. Moraes, "A cryptographic coarse grain reconfigurable architecture robust against dpa," in *2007 IEEE International Parallel and Distributed Processing Symposium*, March 2007, pp. 1–8.
[74] N. Linge and D. Parsons, "Problem-based learning as an effective tool for teaching computer network design," *IEEE Transactions on Education*, vol. 49, no. 1, pp. 5–10, Feb 2006.
[75] J. H. Lee, S. E. Lee, H. C. Yu, and T. Suh, "Pipelined cpu design with fpga in teaching computer architecture," *IEEE Transactions on Education*, vol. 55, no. 3, pp. 341–348, Aug 2012.
[76] E. Morin, *On Complexity*. Hampton Press, 2008.

Daniel Mesquita is associate professor at the Unipampa (Federal University of the Pampa, Brazil). He received his Ph.D. degree in Microelectronics from Université Montpellier II (France) for his thesis on "Reconfigurable architectures and cryptography". In 2007 and 2008 he worked as researcher at the Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento (INESC-ID), in Lisbon, Portugal. In 2008 and 2009 Daniel worked as developer at the CEITEC S.A., a Brazilian semiconductor company. Since 1999 Daniel discusses reconfigurable computing trends, tools and applications. His current research concerns the use of reconfigurable computing to improve security, reliability and flexibility to future internet.

Pedro Frosi is titular professor at the UFU (Federal University of Uberlandia, Brazil). He received his Ph.D. degree in Computer Engineering from the University of Sao Paulo (USP), Brazil and the Centre National pour la Recherche Scientifique, France, on the field of distributed systems. He holds a master degree on computer architecture from USP. He's research interest concern Future Internet, Internet of Things, Cloud Computing, High Availability Architectures and Scalability for Software Architecture.

A Secure Protocol for Exchanging Cards in P2P Trading Card Games Based on Transferable e-cash

M. V. M. Silva and M. A. Simplicio Jr.

***Abstract*—Trading card games (TCG) distinguish from traditional card games mainly because the cards are not shared between players in a match. Instead, users play with the cards they own (e.g., purchased or traded with other players), which corresponds to a subset of all cards produced by the game provider. Even though most computer-based TCGs rely on a trusted third-party (TTP) for preventing cheating during trades, allowing them to securely do so without such entity remains a challenging task. Actually, potential solutions are related to e-cash protocols, but, unlike the latter, TCGs require users to play with the cards under their possession, not only to be able to pass those cards over. In this work, we present the security requirements of TCGs and how they relate to e-cash. We then propose a concrete, TTP-free protocol for anonymously trading cards, using as basis a secure transferable e-cash protocol.**

***Keywords*—Trading Card Games (TCG), secure trading, TTP-free, transferable e-cash.**

I. Introduction

A trading card game (TCG) is a type of card game in which, instead of using a fixed deck, each player creates his/her own deck from a subset of all cards made available by the game provider [1]. During a match, players usually do not share their cards with their opponents; hence, as any different cards may exist, part of the game is to build decks that support a target strategy or game style. To build better decks, users may either trade cards with other users or purchase them directly from the game provider. To improve their revenue, in the last years some providers have expanded their markets beyond the realm of physical cards, including digital versions of their games. This is the case, for example, of "Magic: the Gathering™", one of the first TCGs ever released[1].

To set matches and avoid cheating, digital TCGs typically use a client-server architecture, where the centralized system acts as card market and referee for the matches between players. When considering mobile applications, however, a peer-to-peer (P2P) architecture may present advantages over the client-server one [1], [2]. The reason is that a client-server model obliges players to have a continuous Internet connection when trading or playing, preventing them to do any of those actions otherwise. If the game protocols are designed so it does not depend on a trusted third party (TTP) to prevent cheating, on the other hand, then a local connection would be enough, bringing convenience to users.

M. V. M. Silva, Laboratório de Arquitetura de Redes de Computadores, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil, mvsilva@larc.usp.br

M. A. Simplicio Jr, Laboratório de Arquitetura de Redes de Computadores, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil, mjunior@larc.usp.br

[1]http://magic.wizards.com/en/content/magic-duels

Playing traditional card games in a P2P model was firstly proposed in *mental poker* [3] and different solutions were proposed since them (for a survey, see [4]). These works also served as basis for TTP-free solutions for TCGs, such as Match+Guardian [2] and SecureTCG [1], which allow the detection of cheating attempts during a match with two or more players. Despite those advances concerning *in-game* cheating, such protocols still depend on a trusted entity for each card trading event, leaving the task of reducing this dependence as a subject for future work.

Trading cards in a TTP-free manner is a problem that resembles that tackled by transferable e-cash protocols [5], [6], where the cards replace the digital money. For example, as in e-cash, a player should be able to anonymously trade cards with other players without the need of a TTP for mediating the transactions; however, if he/she sends the same card to two or more players (i.e., "double-spends" it), this should be detectable and the transgressor's anonymity should be revoked.Nevertheless, TCGs also have additional requirements, as there is no concept similar to "playing with owned cards" in the context of e-cash. To the best of our knowledge, there is no definition in the literature of the full set of security requirements that apply to card trading, which hinders further progress in this area.

Aiming to tackle the above issues, in this work we: (1) define the requirements for secure card trading; and (2) instantiate a protocol that fulfills those requirements, allowing players to detect cheating attempts when exchanging cards which each other even before a match starts. The propose scheme is based on existing transferable e-cash protocols (namely, [6] and [7]), with the required adaptations for allowing players to: (1) purchase cards from the game provider in a privacy-preserving manner, meaning that a card cannot be linked to any user unless its owner generates a proof of ownership; (2) use the cards they own in a match; (3) trade cards with other players; (4) verify the validity of the card without the intervention of a TTP, independently of the number of previous owners the card has ever had; (5) let the game provider know about cheating events, such as a user playing with a card that has already been handed over to another user. Since the resulting protocol is transparent to how the matches themselves are handled, it can also be integrated with in-game cheating-detection mechanisms such as the aforementioned Match+Guardian or SecureTCG, thus allowing the construction of secure P2P-based TCG environment.

The rest of this document is organized as follows. Section II discusses the characteristics of TCGs, describing its security requirements compared to those of e-cash protocols.

Section III presents the notation and the building blocks of the proposed protocol, as well as the corresponding security assumptions. Section IV then uses these building blocks to describe a concrete instantiation of the proposed protocol. Finally, Section VI presents our final considerations.

II. Background

This section presents the basic concepts related to TCGs, including the game architecture, the representation of the cards, and the corresponding security requirements and threats.

A. Architecture

Following the notation of [1], [2], the architecture of a P2P TCG encompasses a game server and the players.

The game server is responsible for any action that requires a trusted authority or centralized information storage. One of its primary roles is to serve as a *registration center* for players: to enroll in the system, a user must register with a unique identifier (e.g. e-mail or social security number) and provide his/her public key; the game server then generates a digital certificate to assert this information, allowing anyone to verify who are the system's authorized users.

The game server also acts as a *card market*, being responsible for selling and digitally signing cards, so the buyer can prove that a card is valid as well as its ownership. As a result, the server does not need to keep record of the cards possessed by each player, as ownership varies with time and, as proposed in this work, trading may occur without the server's knowledge. The server is also responsible for informing players of the cards available in the game, as new releases usually add several new cards to the game.

Finally, the server is also the entity that plays the role of *game auditor*, verifying claims regarding cheating attempts and eventually punishing those responsible for misbehavior. For example, in [1], [2], the players may send after-match information to the server to prove that a user cheated, e.g., by modifying the sequence or contents of their deck during a match. If a player sends to the server the list of cards employed by an adversary, the server should also be able to verify the usage of cards that were not under a malicious player's possession at the time of the match (e.g., because he/she traded it earlier). Providing such after-match data is actually very common, as this information is normally required to rank players depending on the number of victories in matches.

Any other action that does not require a TTP, such as playing the game or trading cards, can be performed in purely P2P fashion and still be protected by cheating-detection mechanisms. As in-game cheating is quite thoroughly covered in [1], in this work we focus only on cheating-detection during card trading.

B. Representation of cards

The minimal representation of a card C in a typical TCG corresponds to a tuple $C = (ID, d, V, owner)$, where: ID is the card's unique identifier; d is the card's game-specific information, which defines how it affects the game, which are the conditions for it to be played, etc.; V is some validation information, which allows any player to verify that the card was indeed issued by the game provider; and $owner$ is the information that allows the card's current owner to be identified. Since V and $owner$ are directly related to a players' ability to detect invalid cards or attempts to play with cards that are not actually owned by a player, they are described in more detail later in Section IV, in which a concrete instantiation of the proposed protocol for this purpose is described.

C. Comparison with e-cash

The security issues that appear when trading cards are somewhat similar to those faced by transferable e-cash. Indeed, both systems must provide some sort of *balance*, so that the number of elements (coins or cards) of the system should not grow without the central server's authorization. Hence, no user should be able to produce more elements than what the central server has emitted, which could be done by forging a new element or duplicating an existing one. Many actions supported by card trading and transferable e-cash protocols are also similar: stamping new cards is similar to minting new coins, while trading cards is equivalent to spending coins.

It is, thus, reasonable to build a secure card trading protocol from a transferable e-cash scheme. In this case, like coins, the card's portion that indicates ownership ($owner$), grows in size with each transference [5], or need to be stored somewhere else to prevent such growth (e.g., in a receipt [8]). To avoid indefinite growth, players may *refresh* their cards, which is equivalent to deposit a coin and get a new, mint version of it. TTP-free transferability also raises the problem of duplicating existing elements, an issue that cannot be prevented but can be detected so that the culprit is identifiable when the coin is deposited at the central server. More precisely, in case of double-spending in transferable e-cash schemes, the central server is able to revoke the anonymity of the user responsible for misbehavior, and only of that user, independently of how many owners the coin had before or after it was copied.

In the context of TCGs, however, the double spending problem is more complicated because players may not only trade, but also use their cards without transferring its ownership. Therefore, TCGs also need mechanisms for detecting a scenario in which a user irregularly plays with a card that has been previously traded. As further discussed in Section IV, this can be accomplished if the server crosses the information about refreshed cards with those received from match reports. Hence, refreshing cards benefits both honest players and the game server: the former get a shorter copy of the card, which is less computationally expensive to verify and trade, while the latter is able to audit trades by using the information stored in the cards submitted for refreshing. The same mutual benefit applies to the match reports: honest players who win matches can raise their ranks by informing their victories to the server; honest players who lose matches can make sure the opponent played fairly; and the server can audit if some refreshed or traded card has been illicitly used in a match. It should, thus, be quite easy to encourage players to provide such information often to the server.

In summary, five types of cheating can appear when cards are traded: (1) *Double-refresh*: refreshing a same card twice, obtaining several valid instances of the same card but purchasing a single one; (2) *Double-trade*: sending copies of the same card to different users; (3) *Trade-then-play*: playing with a card that has already been passed to another user; (4) *Refresh-then-trade*: refreshing a card C to obtain a mint version of it, C', but then trading copies of C with other users; and (5) *Refresh-then-play*: refreshing a card C to obtain a mint version of it, C', but then using C in matches with other players.

D. System requirements

From the previous discussion, we can postulate that the following security and usability requirements must be met in by secure P2P-based TCG system.

Verifiable stamping: The card market must stamp cards, so their validity and ownership can be verified without the need of contacting the central server.

TTP-free transferability: Players should be able to trade cards with each other without the intervention of a TTP, and the new ownership can also be verified without the need of contacting a trusted server.

Anonymity: Suppose that U_0 purchases a given card C, and then that card is repeatedly traded among a set of users $\{U_{1...n}\}$ before the last owner, U_{n+1}, informs the server about this ownership. In this case, the server only learns the identity of U_{n+1}, while the C's previous owners remain anonymous. In addition, during this process user U_i only learns the identity of U_{i-1} and U_{i+1}.

Balance: The number of cards in the system cannot grow unless the central server stamps new cards, with invalid duplicates being detected and removed.

Cheat detection: Players cannot trade a card more than once without losing their anonymity toward the server, nor play with a card after having traded it.

Exculpability: The game server, even if in collusion with users, cannot falsely prove that an honest user has cheated, i.e., the cheating-detection mechanism only allows identifying users who have duplicated a card (either for trading or playing with it).

III. Building blocks

This section presents the mechanisms necessary for a concrete construction of a P2P TCG trading protocol. Specifically, the proposed scheme is based on the transferable e-cash scheme described in [6] and revisited in [7], which relies on asymmetric pairings, witness-indistinguishable non-interactive proofs, verifiable random functions and structure-preserving blind signatures.

A. Preliminaries and Notation

Assume three groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ of prime order q, and a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ having the following properties: (1) *bilinearity*: $\forall G \in \mathbb{G}_1, H \in \mathbb{G}_2, a, b \in \mathbb{Z}_q : e(G^a, H^b) = e(G, H)^{ab}$; (2) *non-degenerative*: $\forall G \neq 1_{\mathbb{G}_1}, H \neq 1_{\mathbb{G}_2} : e(G, H) \neq 1_{\mathbb{G}_T}$; and (3) e is *efficiently computable*. The pairing parameters $\Lambda = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, G, H, e)$ are a Type-3 (or asymmetric) pairing if $\mathbb{G}_1 \neq \mathbb{G}_2$ and there is no efficiently computable homomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.

In a finite set $\mathcal{S}$, $s \xleftarrow{\$} \mathcal{S}$ denotes that s is sampled uniformly at random from $\mathcal{S}$. If some protocol $\mathcal{R}$ is a multi-party algorithm between parties $\mathcal{A}$ and $\mathcal{B}$, then $\mathcal{R}\big(\mathcal{A}(a) \leftrightarrow \mathcal{B}(b)\big)$ is the execution of $\mathcal{R}$ with inputs a from $\mathcal{A}$ and b from $\mathcal{B}$. We also consider a cryptographic hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^q$ (e.g., SHA-3 [9]). If $\mathcal{H}$ has more than one input, we consider the inputs are concatenated in the order they are presented. We also define a set of map functions from $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{Z}_q$ to $\{0,1\}^*$, so that group elements can be used as input to the hash function.

A *co-* security assumption is the translation of an assumption from a symmetric pairing to an asymmetric pairing. The superscript numbers in these assumptions are the necessary number of duplicated elements from any source group of the pairing for validation of security, as discussed in details in [10].

B. Groth-Sahai proofs

Proofs of knowledge allow a party to prove knowledge of some secret value without revealing it, which is done by showing a witness satisfying some relation that depends on the secret. The most efficient proofs are usually interactive, based on a challenge-response method. For transferable elements, however, one cannot expect any interaction with the parties not directly involved in the current transference. Nevertheless, as shown in [11], non-interactive proofs can still be performed in an efficient manner when the relation to be proved is a set of equations in some defined format and the witnesses are variables that belong to the solutions set. Such Groth-Sahai proofs depend on a signature scheme, and the relation is defined by the verification equation. For a structure-preserving signature, which is adopted in this article, the relation is the following pairing product equation (PPE):

$$\prod_{i=0}^{m} e(X_i, B_i) \prod_{j=0}^{n} e(A_j, Y_j) \prod_{i=0}^{m} \prod_{j=0}^{n} e(X_i, Y_j)^{\gamma_{ij}} = t \qquad (1)$$

where $A_j \in \mathbb{G}_1$, $B_i \in \mathbb{G}_2$, $t \in \mathbb{G}_T$ and $\gamma_{ij} \in \mathbb{Z}_q$ are constants, and $X_i \in \mathbb{G}_1$ and $Y_j \in \mathbb{G}_2$ are variables. For a proof, we need 4 elements in $\mathbb{G}_1$ and 4 in $\mathbb{G}_2$ for each equation, whereas each variable will be committed to 2 elements in their group.

The algorithms employed by a Groth-Sahai proof are: $GSCommit(x, open) \rightarrow C$, that commits the variables to be used in a proof; $GSProve(\{x_i \ in \ C_i\}_{i=1..n}|eq) \rightarrow \phi$, that creates a proof that the prover knows witnesses that satisfies the equation; and $GSVerify(\phi, eq) \rightarrow \{0,1\}$, that verifies if the proof is valid.

We refer to [11] for a concrete instantiation under Symmetric External Diffie-Hellman (SXDH) assumption [12].

C. Verifiable random function

A verifiable random function (VRF) f_s is a especial type of pseudorandom function that allows anyone who knows the secret seed s to compute $f_s(x)$ for any x and also to

prove that $f_s(x)$ is indeed correct without compromising the unpredictability of f_s at any point $x^{'} \neq x$ [13]. Of especial interest to this work is the VRF instantiation described in [7], in which the verification of x uses the PPE $e(Y = G^{\frac{1}{s+x}}, H^s \cdot H^x) = e(G, H)$, so knowledge of s and x can be proved by the Groth-Sahai method. This specific instantiation is secure under the q-Decisional Diffie-Hellman inversion (q-DDHI) assumption (in $\mathbb{G}_1$) [14] and SXDH (for the Groth-Sahai proof).

D. Structure-preserving blind signature

Blind signatures were originally proposed in the context of anonymous e-cash [15], allowing a user to obtain a valid signature on values unknown to the signer. If transferability is required, the user doing the transfer also needs to prove knowledge of the signed values, which can be achieved using *structure-preserving* (or automorphic) signatures [16]. In such signature schemes, the verification keys lie in the message space, the messages and signatures comprise elements of $\mathbb{G}_1$ and $\mathbb{G}_2$, and the verification is done using a set of PPEs. Using the set of signatures, a prover can create a non-interactive proof of knowledge that some witnesses satisfy the PPE for verification.

There are few structure-preserving blind signature schemes in the literature, and even fewer for efficiently signing a set of messages. For the purposes of this work, we adapt the P-signature scheme proposed by [17], converting it to an asymmetric pairing setting by means of the method proposed in [10]. The reason for this modification is that, even though [17] is quite efficient, it uses symmetric pairing and supersingular elliptic curves, requiring fields of larger size to achieve a security level similar to what can be obtained with an asymmetric pairing [18]. The resulting scheme comprises the following operations:

- $PSetup(k) \rightarrow pparams$: Generates the set of public parameters $pparams$ for the signature, which corresponds to the parameters of Groth-Sahai proofs under an asymmetric pairing setting. For the sake of simplicity, these parameters are omitted in the descriptions of the remainder operations.
- $PKeyGen(n) \rightarrow (pk, sk)$: Choose $\alpha, \beta, \gamma, \omega \xleftarrow{\$} \mathbb{Z}_q^*$ and $U, U_0 \xleftarrow{\$} \mathbb{G}_2$. Compute $U_1 = G^{\beta}, \Omega = G^{\omega}, A = H^{\gamma}$, as well as $\forall i \in [1, 2n] \setminus (n+1) : G_i = G^{\alpha^i}, H_i = H^{\alpha^i}$, to sign n messages. Output the private key $sk = (\gamma, \omega, \beta)$ and the public key $pk = (U, U_0, U_1, \Omega, A, \{G_i\}_{i=1..n}, \{H_i\}_{i=1..n})$.
- $PSign(sk, \vec{m}) \rightarrow \sigma$: For $\vec{m} = (m_1, ..., m_n)$, pick $r \xleftarrow{\$} \mathbb{Z}_q^*$ and compute $K = H^r \cdot \prod_{j=1}^{n} H_{n+1-j}^{m_j}$. Choose $c \xleftarrow{\$} \mathbb{Z}_q^*$ and compute: $\sigma_1 = H^{\gamma/(\omega+c)}$, $\sigma_2 = G^c$, $\sigma_3 = U^c$, $\sigma_4 = (U_0 \cdot K^{\beta})^c$, $\sigma_5 = K^c$, $\sigma_6 = K$, $\sigma_r = r$.
Output the signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_r)$.
- $PVerifySig(pk, \sigma, \vec{m}) \rightarrow \{0, 1\}$: Return 1 if and only if the following equations hold: $e(G, A) = e(\Omega \cdot \sigma_2, \sigma_1)$, $e(\sigma_2, U) = e(G, \sigma_3)$, $e(G, \sigma_4) = e(\sigma_2, U_0) \cdot e(U_1, \sigma_5)$, $e(G, \sigma_5) = e(\sigma_2, \sigma_6)$, and $\sigma_6 = H^r \cdot \prod_{j=1}^{n} H_{n+1-j}^{m_j}$.
- $PCommit(pk, \vec{m}) \rightarrow (K, r)$: Choose $r \xleftarrow{\$} \mathbb{Z}_q^*$ and compute $K = H^r \cdot \prod_{j=1}^{n} H_{n+1-j}^{m_j}$. Output the commitment $comm = (K, r)$.
- $PUpdateComm(pk, \vec{m}, K) \rightarrow K'$: Compute and output $K' = K \cdot \prod_{j=0}^{n} H_{n+1-j}^{m_j}$.
- $PWitGen(pk, i, \vec{m}, K, r) \rightarrow W_i$: If K is a commitment to $\vec{m}$ with opening r, compute and output $W_i = G_i^r \cdot \prod_{j=1; j \neq i}^{n} G_{n+1+i-j}^{m_j}$.
- $PVerifyWit(pk, i, m_i, W_i, K) \rightarrow \{0, 1\}$: Return 1 if and only if the following equation holds: $e(G_i, K) = e(G_1, H_n)^{m_i} \cdot e(W_i, H)$.
- $PProveCom(pk, \vec{m}, K, r) \rightarrow \phi_K$: Generate witnesses for each message committed, $\forall i \in [1, n] : W_i = WitGen(pk, i, \vec{m}, K, r)$. Generate a Groth-Sahai proof of knowledge that the following pairing product equations are valid (group elements in bold are constants): $\forall i \in [1, n] : e(\mathbf{G_i^{-1}}, K) \cdot e(\mathbf{G_n}, H_1^{m_i}) \cdot e(W_i, \mathbf{H}) = \mathbf{1_{G_T}}$; $\forall i \in [1, n] : e(\mathbf{G_1}, H^{m_i}) \cdot e(\mathbf{G^{-1}}, H_1^{m_i}) = \mathbf{1_{G_T}}$; $\forall i \in [1, n] : e(\mathbf{G_{2n}}, H^{m_i}) \cdot e(\mathbf{G^{-1}}, H_{2n}^{m_i}) = \mathbf{1_{G_T}}$. Output the proof ϕ_K and the complementary commitments (from the Groth-Sahai system), $\forall i \in [1, n] : C_{m_i H_1}, C_{m_i H}, C_{m_i H_{2n}}, C_{W_i}; C_K$.
- $PVerifyProofCom(\phi_K) \rightarrow \{0, 1\}$: Verify if the Groth-Sahai proof of knowledge ϕ_K was correctly constructed.
- $\Big(PObtainSig(pk, \vec{m}_P) \leftrightarrow PIssueSig(sk, \vec{m}_S)\Big) \rightarrow \sigma$:

– The User commits the message $\vec{m}_P$ as $(K, r') = PCommit(pk, \vec{m}_P)$, then sends K to the Signer with a proof of knowledge $\phi_K = PProveCom(pk, \vec{m}_P, K, r')$ that the commitment is valid.
– The Signer verifies the proof of knowledge with a call to $PVerifyProofCom(\phi_K)$, updates the commitment to $K' = PUpdateCom(pk, \vec{m}_S, K)$, and blindly signs the commitment with random seeds $c, r'' \xleftarrow{\$} \mathbb{Z}_q^*$ as: $\sigma_1 = H^{\gamma/(\omega+c)}$, $\sigma_2 = G^c$, $\sigma_3 = U^c$, $\sigma_4 = (U_0 \cdot (K' \cdot H^{r''})^{\beta})^c$, $\sigma_5 = (K' \cdot H^{r''})^c$, $\sigma_6 = K' \cdot H^{r''}$, and $\sigma_r' = r''$. The Signer then sends $\sigma' = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_r')$ to the User.
– The User updates $\sigma_r = r' + \sigma_r'$ and outputs $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_r)$.

- $PProveSig(pk, \vec{m}, \sigma) \rightarrow \phi_\sigma$: Parse $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_r)$. Generate witnesses for each message signed $\forall i \in [1, n] : W_i = WitGen(pk, i, \vec{m}, \sigma_6, \sigma_r)$. Generate a Groth-Sahai proof of knowledge that the following pairing product equations are valid (the group elements in bold are constants):

– Signature equation validation: $e(\mathbf{\Omega}, \sigma_1) \cdot e(\sigma_2, \sigma_1) \cdot e(\mathbf{G}, A^{-1}) = \mathbf{1_{G_T}}$; $e(\sigma_2, \mathbf{U_0}) \cdot e(\mathbf{U_1}, \sigma_5) \cdot e(\mathbf{G^{-1}}, \sigma_4) = \mathbf{1_{G_T}}$; $e(\sigma_2, \mathbf{U}) \cdot e(\mathbf{G^{-1}}, \sigma_3) = \mathbf{1_{G_T}}$; $e(\mathbf{G}, \sigma_5) \cdot e(\sigma_2, \sigma_6)^{-1} = \mathbf{1_{G_T}}$.
– Message pertinence validation: $\forall i \in [1, n] : e(\mathbf{G_i^{-1}}, \sigma_6) \cdot e(\mathbf{G_n}, H_1^{m_i}) \cdot e(W_i, \mathbf{H}) = \mathbf{1_{G_T}}$; $\forall i \in [1, n] : e(\mathbf{G_1}, H^{m_i}) \cdot e(\mathbf{G^{-1}}, H_1^{m_i}) = \mathbf{1_{G_T}}$; $\forall i \in [1, n] : e(\mathbf{G_{2n}}, H^{m_i}) \cdot e(\mathbf{G^{-1}}, H_{2n}^{m_i}) = \mathbf{1_{G_T}}$.
– Equality commitment validation: $e(G, \mathbf{A}) \cdot e(G, A^{-1}) = \mathbf{1_{G_T}}$; $e(G, \mathbf{H}) = \mathbf{e(G, H)}$

Output the proof ϕ_σ and the complementary commitments (from the Groth-Sahai system) $\forall i \in [1, n] : C_{m_i H_1}, C_{m_i H}, C_{m_i H_{2n}}, C_{W_i}; \forall j \in [1, 6] : C_{\sigma_j}; C_{-A}, C_G$.

- $PVerifyProofSig(\phi_\sigma) \rightarrow \{0, 1\}$: Verify if the Groth-Sahai proof of knowledge ϕ_σ was correctly constructed.

Table I lists the number of elements necessary for the signature scheme when signing n messages.

TABLE I
NUMBER OF ELEMENTS FROM EACH GROUP WHEN SIGNING n MESSAGES

Object	$\mathbb{Z}_q$	$\mathbb{G}_1$	$\mathbb{G}_2$	*Object*	$\mathbb{Z}_q$	$\mathbb{G}_1$	$\mathbb{G}_2$
Private key (sk)	3	0	0	Opening ($open$)	1	0	0
Public key (pk)	0	$2+n$	$3+n$	Signature (σ)	1	1	5
Message ($\vec{m}$)	n	0	0	Proof of commitment (ϕ_K)	0	$4+2n$	$6+6n$
Commitment (K)	0	0	1	Proof of signature (ϕ_σ)	0	$8+2n$	$18+6n$

E. Compact e-cash

The compact e-cash scheme originally described in [6] and revised in [7] allows a user to withdraw several coins (i.e., a wallet) within a single message. In the context of TCGs, this scheme is interesting because it (1) allows several seed parameters (instead of coins) to be signed altogether and (2) it provides a direct method for identifying cheaters, who have their public key recovered, so the server do not need to screen the whole user database in search for the culprit. The version actually adopted in the proposed solution is based on the adaptation from [19], which achieves transferability with strong anonymity, by means of the following operations (for a concrete instantiation and details, see [7]):

- $Setup$: The bank generates a public/private key pair and publishes its public key together with the system's public parameters.
- $Register$: The user randomly generates a public/private key pair based on the system parameters and retrieves a certificate from the bank for the public key generated in this manner. The bank stores the user's identity and corresponding public keys, which allows users to be identified in case of double-spending.
- $Withdraw$: The user produces seed values and commits them to the bank, which in turn blindly signs those values. This creates a new anonymous wallet with as many coins as the number of seeds provided.
- $Spend$: Users may exchange either unspent coins from their wallets or coins previously received. In the former case, the user creates a new coin from the serial seed and treats it just like a received coin. Each time a coin is spent, a tag giving ownership of it to the receiver is added to the coin representation, making it grow in size. All tags must be verified by the receiver to ensure the previous transaction are valid and, thus, that the coin actually hold value.
- $Deposit$: The user sends the coin to the bank, which verifies if this coin had already been deposited. If it has, the bank verifies if this is a case of double-deposit (i.e., if the user is trying to deposit the same coin twice) or of double-spending (i.e., if it was sent to two different users at some point in time).
- $Identify$: In case of double-spending, the bank retrieves the public key of the perpetrator, so the required administrative penalties can be applied.

A wallet $W = (sk_U, s, t, \sigma)$ is composed by the private key sk_U of the owner, a serial seed s, a transfer seed t, and a signature σ on these values. A coin $C = (S, \phi_S, \phi_\sigma, \pi_T = \{T_j, \phi_{T_j}, r_j, i_j\})$ is identified by a serial number S and its proof of validity ϕ_S, proof of knowledge on the signature of the wallet ϕ_σ, and a set π_T of j transferences. Each transference is composed by a transference tag number T_j and its proof of validity ϕ_{T_j}, a tag of ownership r_j, and some public information i_j. When a coin is spent, a new tag indicating the transference of ownership is inserted into π_T.

The serial number S is picked at random to provide a unique identifier for each coin. It is then employed in the *serial number generation* function f_S, a VRF that is defined by Equation 2. In this equation, s is a seed signed in the wallet and sk_U is the private key of the owner (or the index of the coin, if more than one coin can be withdrawn).

$$f_S(sk_U, s) = G^{\frac{1}{s+sk_U}} \tag{2}$$

The transference tag T identifies each transference, also picked at random. It is then employed in a modified version of the VRF, the *transference tag generation* function f_T described in Equation 3. In this equation, t is a seed signed in the wallet or referenced by previous transference, sk_U is the secret key of the owner and R is the hash of the private (that contains the owner) and public (e.g., a timestamp) information of the transference.

$$f_T(sk_U, t, R) = \left(G^R\right)^{sk_U} G^{\frac{1}{t+sk_U}} \tag{3}$$

Finally, the ownership tag r is a randomly-picked value used to hide the private key of the coin's owner. Similarly to S, it is employed in a VRF, the *ownership tag generation* function f_r from Equation 4, where sk_U is the private key of the owner and i is some public information related to the transference. This tag is used to create the transference tag that allows the owner of the coin to prove that the last transference was directed to him/her, so this information is used to compute R, linking the transference tag T to the owner, represented by r.

$$f_r(sk_U, i) = G^{\frac{1}{sk_U+i}} \tag{4}$$

The revised version also presents a proof of knowledge protocol for the serial number generation $\Phi_S : \left(Prove(sk_U, s) \rightarrow \phi_S; Verify(\phi_S) \rightarrow \{0,1\}\right)$, for the transference tag Φ_T : $\left(Prove(sk_U, t, R) \rightarrow \phi_T; Verify(\phi_T) \rightarrow \{0,1\}\right)$ and for the ownership tag Φ_r : $\left(Prove(r, i) \rightarrow \phi_r; Verify(\phi_r) \rightarrow \{0,1\}\right)$, based on Groth-Sahai proofs which proves the values presented in the coin (S, T and R) were computed using the respective functions. Due to space limitation, we refer the reader to [7] for details.

IV. PROPOSED PROTOCOL

In this section we present a concrete instantiation of the proposed scheme for secure trading cards, using the building blocks described in Sec. III. The roles of registration center $\mathcal{C}$, card market $\mathcal{M}$ and game auditor $\mathcal{A}$ are played by the

game server $\mathcal{G} = \mathcal{C} \cup \mathcal{M} \cup \mathcal{A}$. A card C is represented by the tuple $C = (ID, d, V, owner)$, where: $ID \in \mathbb{G}_1$ is its unique identifier; $d \in \mathbb{Z}_q$ is the numeric representation of the card's description using some suitable encoding; $V = (\phi_{ID}, \phi_\sigma)$, where ϕ_{ID} and ϕ_σ are, respectively, proofs of knowledge of the construction of the ID and of the signature from the market; and $owner = \pi_T = \{T_j, \phi_{T_j}, r_j, i_j\}$ corresponds to the records of all owners of the cards, so that, for each index j in π_T, T_j is the transference tag with proof of knowledge of the construction ϕ_{T_j}, r_j is the ownership tag and i_j is the public information regarding the transference.

The operations comprised by the proposed scheme are, then:

- $Setup()$: The game server generates the system parameters $tcgparams = (pparams_C, pparams_M)$ where $pparams_C$ and $pparams_M$ are the parameters of two signature schemes, the first to register new players and the second to stamp new cards. Both of them contain parameters of a Groth-Sahai proof system, defined over an asymmetric pairing Λ. These parameters are used by the subsequent operations and, for shortness, are omitted in their descriptions. The game server also generates two keypairs: $(sk_C, pk_C) \leftarrow PKeyGen()$ to register players and $(sk_M, pk_M) \leftarrow PKeyGen()$ to stamp cards. It then publishes $tcgparams$, pk_C and pk_M.
- $Register(id_P[, sk_P])$: Player $\mathcal{P}$ with identity id_P generates a secret key $sk_P \xleftarrow{\$} \mathbb{Z}_q$ and computes the public key $pk_P = e(G, H)^{sk_P}$. $\mathcal{P}$ generates a proof of knowledge $\phi_P = GSProof\big(H^{sk_P}\ in\ C_{sk_P}, \theta = 1 | e(G^\theta, H^{sk_P}) = pk_P \wedge e(G^\theta, H) = e(G, H)\big)$. The triple (id_P, pk_P, ϕ_P) is sent to the registration center $\mathcal{C}$. If the proof ϕ_P is valid, $\mathcal{C}$ generates a signature $\sigma_P = PSign(sk_C, \{id_P, pk_P\})$. $\mathcal{P}$ can then present σ_P as his/her certificate.
- $Stamp\Big(\mathcal{P}(sk_P, pk_M, d) \leftrightarrow \mathcal{M}(sk_M, pk_M, d)\Big)$: To purchase an instance of a card with description d, player $\mathcal{P}$ generates a partial identifier seed $s' \xleftarrow{\$} \mathbb{Z}_q$ and a transference seed $t \xleftarrow{\$} \mathbb{Z}_q$, and the card market $\mathcal{M}$ generates the card's partial identifier component $s'' \xleftarrow{\$} \mathbb{Z}_q$. Both parties execute the interactive protocol to obtain a blind signature $\sigma_s = \Big(PObtainSig(pk_M, \{sk_P, s', t, 0\}) \leftrightarrow PIssueSig(sk_M, \{0, s'', 0, d\})\Big)$ that is returned to $\mathcal{P}$ together with s''. The player then generates a proof of knowledge $\phi_\sigma = PProveSig(pk_M, \{sk_P, s = s' + s'', t, d\}, \sigma_s)$. After that, $\mathcal{P}$ chooses some unique the public information $i_0 \leftarrow \{0,1\}^*$ (e.g., a timestamp) and computes $r_0 = G^{\frac{1}{sk_P + \mathcal{H}(i_0)}}$ and $R_0 = \mathcal{H}(r_0, i_0)$. $\mathcal{P}$ then generates the unique identifier $ID = f_S(sk_P, s)$ and the transference tag $T_0 = f_T(sk_P, t, R_0)$, together with proofs of knowledge $\phi_{ID} = \Phi_S.Prove(sk_P, s)$ and $\phi_{T_0} = \Phi_T.Prove(sk_P, t, R_0)$ of the construction, associated with the commitments in the proof of signature ϕ_σ. Finally, the player stores the card $C = (ID, d, \phi_{ID}, \phi_\sigma, \pi_T = \{T_0, \phi_{T_0}, r_0, i_0\})$.
- $Send\Big(\mathcal{P}_1(sk_{P_1}, pk_{P_2}, C) \leftrightarrow \mathcal{P}_2(sk_{P_2}, pk_{P_1})\Big)$: The receiver $\mathcal{P}_2$ chooses some public information $i \leftarrow \{0,1\}^*$ and computes $r = G^{\frac{1}{sk_{P_2} + \mathcal{H}(i)}}$ and a proof of validity $\phi_r = \Phi_r.Prove(sk_{P_2}, r, \mathcal{H}(i))$. $\mathcal{P}_2$ then sends the tuple (i, r, ϕ_r) to the current card hold, $\mathcal{P}_1$. $\mathcal{P}_1$ parses $C = (ID, d, \phi_{ID}, \phi_\sigma, \pi_T = \{T_j, \phi_{T_j}, r_j, i_j\}_{j=0\ldots h})$ and verifies the proof of validity $\Phi_r.Verify(\phi_r, pk_{P_2})$. If everything is correct, $\mathcal{P}_1$ first sets $i_{h+1} = i$ and $r_{h+1} = r$, and then computes $R_{h+1} = \mathcal{H}(r_{h+1}, i_{h+1})$ and $t = \mathcal{H}(S, \{T_j\}_{j=0\ldots h})$. Finally, $\mathcal{P}_1$ generates a new transference tag $T_{h+1} = f_T(sk_{P_1}, t, R_{h+1})$, as well as a proof of knowledge $\phi_{T_{h+1}} = \Phi_T.Prove(sk_{P_1}, t, R_{h+1})$ of the construction. The card $C' = (ID, d, \phi_{ID}, \phi_\sigma, \pi'_T = \{T_j, \phi_{T_j}, r_j, \phi_{r_j}, i_j\}_{j=0\ldots(h+1)})$ is sent to $\mathcal{P}_2$. Upon reception, $\mathcal{P}_2$ verifies the construction of the unique identifier ID by $\Phi_S.Verify(\phi_{ID})$ and the tags $\{T_j\}_{j=0..h}$ by $\bigwedge_{j=0}^{h+1} \Phi_T.Verify(\phi_{T_j})$, as well as that the proof of ownership $\phi_{r_{h+1}}$ is valid in respect to the public key pk_{P_2} by $\Phi_r.Verify(\phi_{r_{h+1}}, pk_{P_2})$. If all proofs are correct, $\mathcal{P}_2$ stores the card C' as his/her own.
- $Play\Big(\mathcal{P}_1(sk_{P_1}, C) \leftrightarrow \mathcal{P}_2(pk_{P_1})\Big)$: Player $\mathcal{P}_1$ prepares a card $C = (ID, d, \phi_{ID}, \phi_\sigma, \pi_T = \{T_j, \phi_{T_j}, r_j, i_j\}_{j=0..h})$ that has been updated h times. $\mathcal{P}_1$ chooses some arbitrary public information $i_{h+1} \leftarrow \{0,1\}^*$ and computes $r_{h+1} = G^{\frac{1}{sk_{P_1} + \mathcal{H}(i_{h+1})}}$, $R_{h+1} = \mathcal{H}(r_{h+1}, i_{h+1})$ and $t = \mathcal{H}(S, \{T_j\}_{j=0\ldots h})$. Then $\mathcal{P}_1$ generates a new transference tag $T_{h+1} = f_T(sk_{P_1}, t, R_{h+1})$, together with proof of knowledge $\phi_{T_{h+1}} = \Phi_T.Prove(sk_{P_1}, t, R_{h+1})$ of the construction. The card C is updated to $C' = (ID, d, \phi_{ID}, \phi_\sigma, \pi'_T = \{T_j, \phi_{T_j}, r_j, \phi_{r_j}, i_j\}_{j=0\ldots(h+1)})$ $\mathcal{P}_1$ also prepares two proofs of knowledge $\phi_{r_h} = \Phi_r.Prove(sk_{P_2}, \mathcal{H}(i_h))$ and $\phi_{r_{h+1}} = \Phi_r.Prove(sk_{P_2}, \mathcal{H}(i_{h+1}))$ to prove that the card was correctly prepared The triple $(C', \phi_{r_h}, \phi_{r_{h+1}})$ is sent to the match's opponent $\mathcal{P}_2$. Upon reception of C', $\mathcal{P}_2$ verifies the construction of the unique identifier ID by $\Phi_S.Verify(\phi_{ID})$ and transference tags $\{T_j\}_{j=0\ldots(h+1)}$ by $\bigwedge_{j=0}^{h+1} \Phi_T.Verify(\phi_{T_j})$, and that both proofs of ownership ϕ_{r_h} and $\phi_{r_{h+1}}$ are valid in respect to the public key pk_{P_1} by $\Phi_r.Verify(\phi_{r_h}, pk_{P_1}) \wedge \Phi_r.Verify(\phi_{r_{h+1}}, pk_{P_1})$. If they are all valid, $\mathcal{P}_2$ then stores this card locally, so it can report this information to the game server later, and uses the unique identifier ID to identify this card during the match.
- $Report\Big(\mathcal{P}(C) \leftrightarrow \mathcal{A}(\mathcal{RS})\Big)$: Player $\mathcal{P}$ sends to the game auditor $\mathcal{A}$ a card $C = (ID, d, \phi_{ID}, \phi_\sigma, \pi_T = \{T_j, \phi_{T_j}, r_j, i_j\}_{j=0\ldots h})$ that an opponent has used in some match. $\mathcal{A}$ stores C in the set of reported cards $\mathcal{RS}$ and verifies if there is any card $\bar{C}$ with identifier $\bar{ID} = ID$ already reported in $\mathcal{RS}$. For each card $\bar{C}$, $\mathcal{A}$ executes $Identify(C, \bar{C})$, retrieving the list of public keys of users who had illegally duplicated this card.
- $Refresh\Big(\mathcal{P}(sk_P, C) \leftrightarrow \big(\mathcal{G} = \mathcal{A}(\mathcal{RS}) \cup \mathcal{M}(sk_M, pk_P)\big)\Big)$: Player $\mathcal{P}$ prepares a card $C = (ID, d, \phi_{ID}, \phi_\sigma, \pi_T = \{T_j, \phi_{T_j}, r_j, i_j\}_{j=0\ldots h})$ that has been updated h times. $\mathcal{P}$ chooses some public information $i_{h+1} \leftarrow \{0,1\}^*$ (e.g., a timestamp) and computes $r_{h+1} = G^{\frac{1}{sk_P + \mathcal{H}(i_{h+1})}}$, $R_{h+1} = \mathcal{H}(r_{h+1}, i_{h+1})$ and $t = \mathcal{H}(S, \{T_j\}_{j=0\ldots h})$. Then $\mathcal{P}$ generates a new transference tag $T_{h+1} = f_T(sk_P, t, R_{h+1})$, together with proof of knowledge $\phi_{T_{h+1}} = \Phi_T.Prove(sk_P, t, R_{h+1})$ of the construction. The card C is updated to $C' = (ID, d, \phi_{ID}, \phi_\sigma, \pi'_T = \{T_j, \phi_{T_j}, r_j, i_j\}_{j=0\ldots(h+1)})$ and is sent to the game server $\mathcal{G}$. $\mathcal{A}$ stores C' in $\mathcal{RS}$ and verifies if there is any card $\bar{C}$ with identifier $\bar{ID} = ID$ already reported in

$\mathcal{RS}$. For each card $\bar{C}$, $\mathcal{A}$ executes $Identify(C', \bar{C})$, retrieving the list of public keys of users who had illegally duplicated this card. If identifying C' did not return any transgressor, both parties execute $Stamp\big(\mathcal{P}(sk_P, pk_M, d) \leftrightarrow \mathcal{M}(sk_M, pk_P)\big)$ to produce a fresh card C'' to $\mathcal{P}$.

- $Identify(C, \bar{C})$: The game auditor $\mathcal{A}$ parses cards $C = (ID, d, \phi_{ID}, \phi_\sigma, \pi_T = \{T_j, \phi_{T_j}, r_j, i_j\}_{j=0\ldots h})$ and $\bar{C} = (\bar{S}, \phi_{\bar{ID}}, \phi_{\bar{\sigma}}, \bar{d}, \bar{\pi_T} = \{\bar{T_j}, \phi_{\bar{T_j}}, \bar{r_j}, \bar{i_j}\}_{j=0\ldots \bar{h}})$ with the same identifier $ID = \bar{ID}$. It searches for the first index l in which $T_l \neq \bar{T_l}$, computes $R_l = \mathcal{H}(r_l, i_l)$ and $\bar{R_l} = \mathcal{H}(\bar{r_l}, \bar{i_l})$, and retrieves the public key of the perpetrator $\mathcal{D}$ as $pk_D = \left(\frac{T_l}{\bar{T_l}}\right)^{\frac{1}{R_l - \bar{R_l}}}$. If the index l is larger than the number of hops for any card (h or $\bar{h}$), this card had already been reported but had not been duplicated, so the output is empty.

The requirements of a secure card trading system, as presented in Sec. II, are fulfilled by the underlying e-cash scheme. Namely, the signature on stamping method guarantees verifiability ("own" property), anonymity when stamping (the signer cannot link signatures to new cards), and balance (if the signature is unforgeable, a new card cannot be inconspicuously created without authorization by the card market). The proof of knowledge provides transferability on trading and ad-hoc playing, given its non-interactivity property. It also keeps anonymity when trading, since it is witness-indistinguishable together with the VRF. Finally, the identification method of the e-cash scheme guarantees balance, cheat detection and exculpability.

V. Preliminary efficiency analysis

Signing, $n = 4$ messages ($\{sk_P, s, t, d\}$) with the presented P-signature scheme, a signature proof (ϕ_σ) requires 20 elements in $\mathbb{G}_1$ and 42 in $\mathbb{G}_2$ (see Table I). For the serial number generation proof (ϕ_{ID}), we need 24 elements in $\mathbb{G}_1$ and 26 in $\mathbb{G}_2$. For the transference tag generation proof (ϕ_T), we need 36 elements in $\mathbb{G}_1$ and 38 in $\mathbb{G}_2$. A card C is composed by: the unique identifier ID (1 element in $\mathbb{G}_1$, output from f_S) the proofs of knowledge of ID (ϕ_{ID}) and of the signature from the market (ϕ_σ), and a set of transferences (π_T), updated with each trade or play. Each transference j in the set is in turn composed by: the transference tag T_j (1 element in $\mathbb{G}_1$, output from f_T), the corresponding proof of knowledge ϕ_{T_j}, the ownership tag r_j (1 element in $\mathbb{G}_1$, output from f_r), and additional public information i_j, which may have variable length. Hence, for a total of t transferences and/or usages, a card needs $45 + 38t$ elements in $\mathbb{G}_1$ and $68 + 38t$ in $\mathbb{G}_2$.

The execution time is likely dominated by the pairing computations. Using a Groth-Sahai proof of knowledge, each time a card is traded or used in a match, the total execution cost corresponding basically to the 148 underlying pairing computations. As each pairing is expected to take on the order of 1 ms to run with [20] (Intel Core i5 1,6 GHz, 128-bit security level), the total time would be around 150 ms per card traded or played. We note that, while these timings are quite reasonable for trading, they may be somewhat cumbersome when playing with a deck having roughly 50 cards, as it is common in commercial TCGs, since the verification of a deck would take around 7.5 min. Nevertheless, it is important to have in mind that the preparation of a deck can be done beforehand, much before the match starts; in addition, the verification of the corresponding proofs of knowledge can happen in background during the match, which usually takes several minutes. Therefore, in practice those costs can be made transparent to players.

VI. Conclusions

In this paper, we presented the set of requirements for allowing secure trades in P2P TCGs, defining the cheating types that need to be detected. We then adapted a transferable e-cash protocol for creating a concrete scheme that fulfills those requirements The proposed scheme is based on the P-signatures described in [17], which allows a vector of messages to be signed, which is combined with a compact blind signature scheme in the asymmetric pairing setting to allow a more memory-efficient representation.

According to our preliminary analysis, the scheme is quite efficient to be used in practice, especially considering that the most expensive operations involved (namely, validating an entire deck of cards) can be performed in background, either before or during a match.

Acknowledgments

This work was supported by the São Paulo Research Foundation (FAPESP) under grant 2011/21592-8 and by the National Counsel of Technology and Scientific Development (CNPq) under grants 482342/2011-0 and 165874/2014-7.

References

[1] M. A. Simplicio, M. A. Santos, R. R. Leal, M. A. Gomes, and W. A. Goya, "SecureTCG: a lightweight cheating-detection protocol for P2P multiplayer online trading card games," *Security and Communication Networks*, vol. 7, no. 12, pp. 2412–2431, 2014.

[2] D. Pittman and C. GauthierDickey, "Match+Guardian: a secure peer-to-peer trading card game protocol," *Multimedia systems*, vol. 19, no. 3, pp. 303–314, 2013.

[3] A. Shamir, R. Rivest, and L. Adleman, "Mental poker," in *The Mathematical Gardner*, D. Klarner, Ed. Springer US, 1981, pp. 37–43. [Online]. Available: http://dx.doi.org/10.1007/978-1-4684-6686-7_5

[4] J. Castellà Roca, F. Sebé Feixas, and J. Domingo-Ferrer, *Contributions to mental poker*. Universitat Autònoma de Barcelona,, 2006.

[5] D. Chaum and T. P. Pedersen, "Transferred cash grows in size," in *Advances in Cryptology (Eurocrypt'92)*. Springer, 1993, pp. 390–407.

[6] J. Camenisch, S. Hohenberger, and A. Lysyanskaya, "Compact e-cash," in *Advances in Cryptology (Eurocrypt'05)*. Springer, 2005, pp. 302–321.

[7] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya, "Compact e-cash and simulatable vrfs revisited," in *Pairing'09*. Springer, 2009, pp. 114–131.

[8] G. Fuchsbauer, D. Pointcheval, and D. Vergnaud, "Transferable constant-size fair e-cash," in *Cryptology and Network Security*. Springer, 2009, pp. 226–247.

[9] National Institute of Standards and Technology, *DRAFT FIPS PUB 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. pub-NIST, May 2014. [Online]. Available: http://csrc.nist.gov/publications/drafts/fips-202/fips_202_draft.pdf

[10] M. Abe, J. Groth, M. Ohkubo, and T. Tango, "Converting cryptographic schemes from symmetric to asymmetric bilinear groups," in *Advances in Cryptology (CRYPTO'14)*. Springer, 2014, pp. 241–260.

[11] J. Groth and A. Sahai, "Efficient non-interactive proof systems for bilinear groups," in *Advances in Cryptology (Eurocrypt'08)*. Springer, 2008, pp. 415–432.

[12] L. Ballard, M. Green, B. de Medeiros, and F. Monrose, "Correlation-resistant storage," TR-SP-BGMM-050507, Johns Hopkins UDCS, Tech. Rep., 2005.

[13] S. Micali, M. Rabin, and S. Vadhan, "Verifiable random functions," in *Foundations of Computer Science, 1999. 40th Annual Symposium on*. IEEE, 1999, pp. 120–130.
[14] Y. Dodis and A. Yampolskiy, "A verifiable random function with short proofs and keys," in *Public Key Cryptography (PKC'05)*. Springer, 2005, pp. 416–431.
[15] D. Chaum, "Blind signatures for untraceable payments," in *Advances in cryptology*. Springer, 1983, pp. 199–203.
[16] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo, "Structure-preserving signatures and commitments to group elements," in *Advances in Cryptology (CRYPTO'10)*. Springer, 2010, pp. 209–236.
[17] M. Izabachène, B. Libert, and D. Vergnaud, "Block-wise P-signatures and non-interactive anonymous credentials with efficient attributes," in *Cryptography and Coding*. Springer, 2011, pp. 431–450.
[18] R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé, "A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic," in *Advances in Cryptology (Eurocrypt'14)*. Springer, 2014, pp. 1–16.
[19] S. Canard, A. Gouget, and J. Traoré, "Improvement of efficiency in (unconditional) anonymous transferable e-cash," in *Financial Cryptography and Data Security*. Springer, 2008, pp. 202–214.
[20] D. F. Aranha, K. Karabina, P. Longa, C. H. Gebotys, and J. López, "Faster explicit formulas for computing pairings over ordinary curves," in *Advances in Cryptology–EUROCRYPT 2011*. Springer, 2011, pp. 48–68.

Marcos Vinicius Maciel da Silva received his BSc (2013) in Electrical/Computing Engineering at the Escola Politécnica, Universidade de São Paulo, Brazil. He is currently following his MSc on Electrical/Computing Engineering at the same institution. He has experience in the area o Computer Science, with especial interest in the following topics: cloud computing, zero-knowledge proofs, blind signature and electronic cash protocols.

Marcos Antonio Simplicio Junior is an Assistant Professor in the Department of Computer and Digital Systems Engineering at the Escola Politécnica, Universidade de São Paulo, Brazil. He received his BSc (2006), MSc (2008) and PhD (2010) degrees in Electrical/Computing Engineering at the same institution, and also has a Master degree (2006) in Engineering conferred by the Ecole Centrale Des Arts Et Manufactures (Ecole Centrale Paris), France. His main research interests and the focus of the projects coordinated by him include: (applied) cryptography, including the design and analysis of algorithms and protocols; and network security, in particular solutions tailored for scenarios involving resource-constrained devices (e.g., sensor networks), and distributed systems (e.g., cloud computing and P2P networks).

Website of the National Network of Information Security and Criptography
http://www.renasic.org.br

ISSN 2358-8963