A Novel Approach to Ensure Data Confidentiality Using Encryption and Obfuscation

Alexander Kozachok

***Abstract*— In information society, data is become one of the most important part to company or individual. Documented file is one of the forms for storing data on a storage media, especially for use by computers. Thus, documented file protection has become urgent. This paper presents a novel approach to protect documented files from unauthorized access by combining encryption and obfuscation. Encryption is used to protect the confidentiality of document content, and obfuscation provides a way to protect program code by making it unreadable and use to control access. The proposed system protects documented files not only content confidentiality but files access control methods, and can reduce the chance of data leakage at utmost.**

***Tóm tắt*— Trong thời đại thông tin hiện nay, dữ liệu là một trong những thành phần quan trọng nhất của các tổ chức và cá nhân. Tập tin văn bản là một trong những hình thức phổ biến để lưu trữ dữ liệu trên các phương tiện lưu trữ, đặc biệt là trên máy tính. Vì vậy, việc bảo mật các tập tin văn bản là hết sức cần thiết. Bài báo này đề xuất một phương pháp mới cho việc bảo mật các tập tin văn bản chống lại việc thâm nhập trái phép dựa trên việc kết hợp mã hóa và thuật toán làm rối mã chương trình. Trong đó, mã hóa được sử dụng để bảo mật nội dung văn bản, còn thuật toán làm rối mã chương trình cho phép bảo mật mã nguồn và điều khiển truy cập. Hệ thống chương trình này cho phép bảo vệ văn bản dựa trên việc bảo mật nội dung và phương pháp điều khiển truy cập để giảm tối đa các nguy cơ rò rỉ dữ liệu.**

Keywords*— *obfuscation; unauthorized access; encryption documented file.

Từ khóa*— *thuật toán làm rối mã; xâm nhập trái phép; mã hõa tập tin văn bản.

I. INTRODUCTION

Today development in information system technologies have resulted in computerizing many applications in almost all aspects of our life. Valuable secret information is vulnerable during storing and transmission over network by unauthorized access. In this age of universal electronic connectivity, of viruses and hackers, of electronic eavesdropping and electronic fraud, there is indeed a need to protect information from passing before curious eyes or, more importantly, from falling into wrong hands. Encryption and obfuscation techniques are two important branches of information security. Encryption is the process of converting the readable form into unreadable form using an algorithm and key. Obfuscation is a process which illegal users do by implementing a particular mathematical function [1]. It is supposed to use GOST R 34.12-2015 [2] algorithm to encrypt documented files and then use indistinguishability obfuscation [3] to obfuscate Boolean access rights function. Based on these two techniques information would be more secure and protected from unauthorized access.

This paper describes a novel approach to protect information from unauthorized access which uses both methods: encryption and obfuscation.

II. PROPOSED SYSTEM

The objective of the proposed scheme is to increase security of documented files. In this age of universal electronic connectivity, of vulnerabilities and threats, of various techniques to analyze source code and computer memory, to secure information against security breaches and attacks there is a need of more sophisticated techniques to protect documented files. To avoid the problem of unauthorized data access using encryption along with obfuscation techniques is the right solution.

The main idea of the proposed system is the following. At first, a documented file is encrypted with GOST R 34.12-2015 (Kuznechik) encryption algorithm. GOST R 34.12-2015 is a symmetric-key block cipher having high efficiency with respect to security and speed. The proposed system generates one secured container for each encrypted document that is called Boolean access rights function. System uses cryptographic multilinear maps to obfuscate content of this container such that the access rights function is protected and its functionality is preserved. Then encrypted document and obfuscated access rights

function are embedded into the container is an executable linkable file (ELF file) format. The document is secured during storing and transmitting over network.

A block diagram of proposed system is shown in Figure 1. The detailed design flow is described below:

Step 1: User identity. When a user logins, the system will need to determine the identity of the user who requests a service (Block 1). This step is used to configure access rules and security policies based on the user identifier (ID_{user}). Each user has a unique ID_{user}, stored on the USB-token.

Step 2: Operating mode selection. After identification, the user can choose one of the two modes (Block 2): create a new documented file (Block 12) or selecting old one (Block 3). Now, it will be considered the process of creating a new document.

Step 3: Creating a new document. The proposed system refers to creating an isolated environment for running applications that are used to create documents (Block 7). Opening and running document in a completely isolated environment allows the unleashing of viruses, worms so as not to have an effect on the real system and also better control over processes use of system resources [4, 5]. This step outputs the documented file D.

Step 4: Set access rights. After the user has worked with the document D and closed it, user must to set access permissions to the document D for other users in system: read, write and full control (Block 8). The permissions are stored in an access matrix M. The access matrix represents the current access permissions structure. Its component M_{ij} records the modes in which subject S_i is permitted to access object O_j. The access matrix structure allows us to implement both mandatory access control and discretionary access control simultaneously. The table 1 below shows an example of setting permissions to the document.

Step 5: Encrypt data and generate access rights function. This step includes data encryption and access rights function generation. At first, documented file D and access matrix M are encrypted with the key K using GOST R 34.12–2015 encryption algorithm (Block 9).

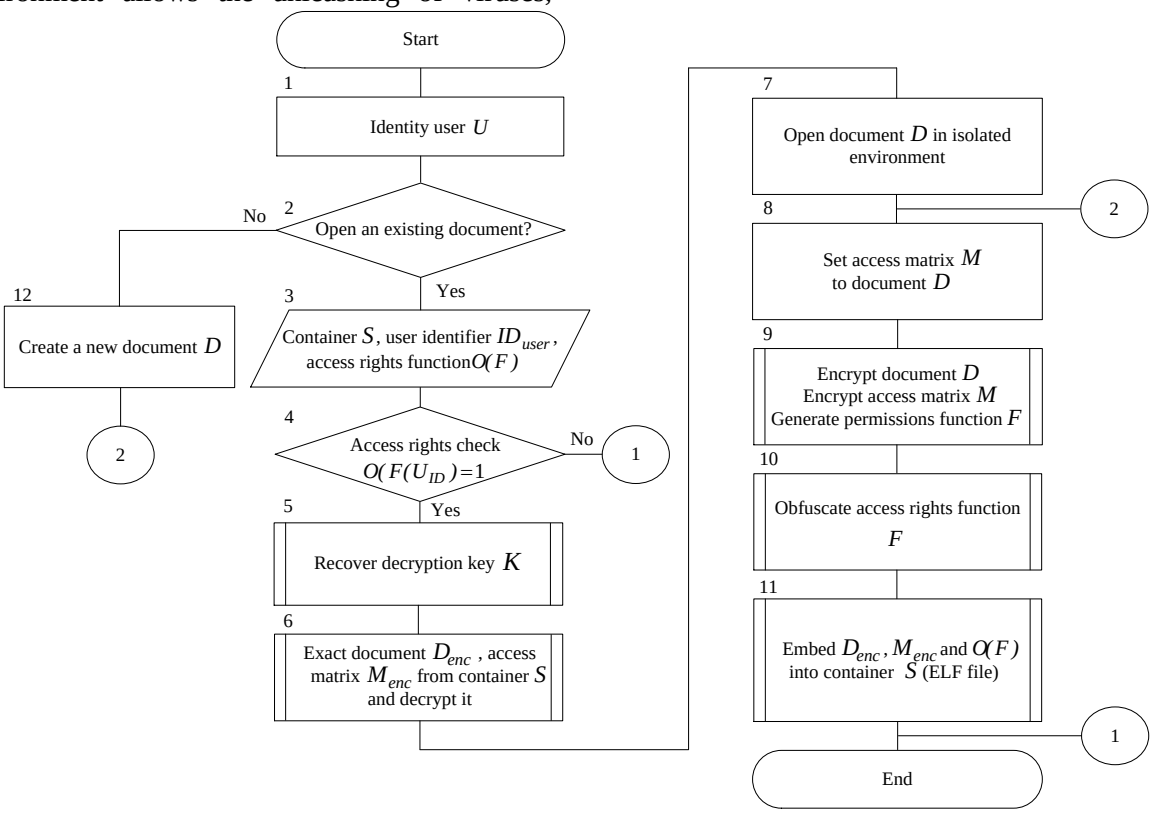

Figure 1. Block diagram of the proposed system

The encryption key K was generated by USB-token. The device not only generates encryption keys but it also generates an access rights function F. The access rights function F allows to determine who will have access to encrypted document D_{enc} and securely recover the decryption key K. This step outputs the encrypted document D_{enc}, encrypted access matrix M_{enc} and access rights function F.

TABLE 1. SETTING THE PERMISSIONS TO THE DOCUMENT

Sensitivity labels	User names	Group names	Access rights		
			Read	Write	Full control
*	User_2	*	1	1	1
2	*	*	1	1	0
*	*	Workers	1	0	0

Step 6: Obfuscate the access rights function. Since the code of access rights function F is delivered in plain, this function is required to be obfuscated (Block 10). Using code obfuscation technologies provides a mechanism for preventing tampering, deterring reverse engineering or recreational challenge for attacker or users analyzing the source code. This step outputs the obfuscated access rights function $O(F)$.

Step 7: Embed data into ELF file. Encrypted document D_{enc}, encrypted access matrix M_{enc} and obfuscated access rights function $O(F)$ are embedded in the container S (ELF file). This container is used to securely store information and can be sent to other users in the system.

When the user opens an existing document D_{enc} stored in a container S (Block 2, Block 3), obfuscated access rights function $O(F)$ is used to control who is able to access and execute a certain container (Block 4). If the access rights check is successful, obfuscated access rights function $O(F)$ recover key K to decrypt document D_{enc} (Block 5, Block 6).

III. PERMISSIONS FUNCTION

This part describes the process of generating access rights function and recovering decryption key based on user identifier. The following example explains how to do this.

A. Create a access rights function

- Generate an encryption key with 4-bit (as an example) length $K = 1001$ by using USB-token;
- Perform circular right shift operation on proposed destination user identifier $ID_{destination\_user} = 1011$ and corresponding bit values of encryption key as shown in Table 2.

TABLE 2. CIRCULAR RIGHT SHIFT FOR USER IDENTIFIER

Round	Circular shift	Bit values of encryption key
1	1011	1
2	1101	0
3	1110	0
4	0111	1

- Create a truth table for 4 variables as in Table 3. Its output is "true" (1) if inputs are the results of circular right shift operation and conresponding key bit equal 1. Otherwise, the output is "false" (0).

TABLE 3. TRUTH TABLE FOR ACCESS RIGHTS FUNCTION

Round	x_1	x_2	x_3	x_4	Output
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

- Create a access rights function in full disjunctive normal form from Table 3. In example it is:

$$f(x_1,x_2,x_3,x_4) = (\bar{x}_1 \wedge x_2 \wedge x_3 \wedge x_4) \vee (x_1 \wedge \bar{x}_2 \wedge x_3 \wedge x_4)$$

B. Recover decryption key

To recover a decryption key, perform circular right shift operation on proposed destination user $ID_{destination\_user} = 1011$ and compute the value of the access rights function by reading the values of

variables in all rounds in according to user identifier. The decryption key is K=1001.

In general case, encryption key and user identifier should be not less than 256-bit length to provide security purpose.

TABLE 4. EXAMPLE OF RECOVERING DECRYPTION KEY FROM ACCESS RIGHTS FUNCTION

Round	Circular shift	$f(x_1,x_2,x_3,x_4)$
1	1011	1
2	1101	0
3	1110	0
4	0111	1

IV. INDISTINGUISHABILITY OBFUSCATION

A. Preliminaries

Here, we first review the definition of cryptographic multilinear maps, which are the cornerstone of proposed algorithm.

Definition 1. (k–multilinear Map [6, 7]). For $k+1$ cyclic groups $G_1,...,G_k,G_T$ of the same order p, a k–multilinear map $e: G_1,...,G_k \to G_T$ has the following properties:

a) For elements $g_1 \in G_1,...,g_k \in G_k$, index $i \in [k]$ and an integer $\alpha \in Z_p$, we have:

$$e(g_1,...,\alpha.g_i,...,g_k) = \alpha.e(g_1,...,g_k)$$

b) The map e is non-degenerate, which means that if $g_i \in G_i (i=1,...,k)$ is a generator of G_i, then $e(g_1,...,g_k)$ is a generator of the target group G_T.

In the above definition, if $G_i (i=1,...,k)$ are all identical groups, it is called a symmetric multilinear map.

Cryptographic multilinear map has been a long sought–after and powerful tool. Not until recently it was approximately constructed in the form of Graded Encoding System. In particular, main algorithm is based on the construction over integers, which is a more practical one devise by Coron, Lepoint and Tibouchi [8]. Now we briefly recall the definition of Graded Encoding System and its associated efficient procedures of algorithm.

Definition 2. (k–Graded Encoding System [7]). A k–Graded Encoding System consists of a ring R and a system of sets $S = \{S_v^{(\alpha)} \in \{0,1\}^* : v \in N, \alpha \in R\}$, with the following properties:

a) For the every $v \in N$, the set $\{S_v^{(\alpha)} : \alpha \in R\}$ are disjoint.

b) There is an associate binary operation '+' and a self-inverse unary operation '–' (on $\{0,1\}^*$) such that for every $\alpha_1, \alpha_2 \in R$, every index $i \le k$, and every $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_i^{(\alpha_2)}$, it holds that

$$u_1 + u_2 \in S_i^{(\alpha_1+\alpha_2)}, -u_1 \in S_i^{(-\alpha_1)}$$

where $\alpha_1 + \alpha_2$ and $-\alpha_1$ are addition and negation in R.

c) There is an associate binary operation '$\times$' (on $\{0,1\}^*$) such that for every $\alpha_1, \alpha_2 \in R$, every i_1, i_2 with $i_1 + i_2 \le k$, and every $u_1 \in S_{i_1}^{(\alpha_1)}$ and $u_2 \in S_{i_2}^{(\alpha_1)}$, it holds that $u_1 \times u_2 \in S_{i_1+i_2}^{(\alpha_1 \cdot \alpha_2)}$. Here $\alpha_1 \cdot \alpha_2$ is multiplication in R, and $i_1 + i_2$ is integer addition.

Definition 3. (Efficient Procedures for k–Graded Encoding System [7, 8]). For graded encoding system, we have the following efficient procedures:

Instance Generation:

$(params, P_{zt}) \leftarrow InstGen(1^\lambda, 1^k)$,

where $params$ is a description of a k–Graded Encoding System with security parameter λ, and P_{zt} is a zero-test parameter for level k.

Ring Sampler:

$c \leftarrow samp(params)$.

This procedure outputs a "level-zero encoding" $c \in S_0^{(\alpha)}$ for a nearly uniform element $\alpha \in R$.

Encoding:

$c_k \leftarrow enc(params, k, c)$.

This procedure outputs the "level-i encoding" $c_k \in S_i^{(\alpha)}$ for a "level-zero encoding" $c \in S_0^{(\alpha)}$.

Re-randomization:

$c' \leftarrow reRand(params, i, c)$.

Procedure *reRand* can re-randomize encodings relative to the same level i.

Addition and negation:

$u' \leftarrow add(params, i, u_1, u_2)$

$u' \leftarrow neg(params, i, u_1)$.

These two procedures are corresponding to operation '+' and '–' in the above definition.

Multiplication:

$u' \leftarrow mul(params, i_1, u_1, i_2, u_2)$.

This procedure will check whether $u_k \in S_k^{(0)}$.

Zero Testing:

$isZero(params, P_{zt}, u_k) \stackrel{?}{=} 0/1$.

This procedure will check whether $u_k \in S_k^{(0)}$.

Extraction:

$sk \leftarrow ext(params, P_{zt}, u_k)$.

This procedure extracts a "canonical" and "random" representation of ring elements from their level-k encoding.

B. Obfuscation Algorithm

There will be introduced proposed framework which is comprised of two phases, the obfuscating phase and the evaluation phase.

A cryptographic obfuscator of a program can make new program performing the same functionality while the detailed underlying instructions are hidden. The first phase is the construction of an obfuscator O for access rights function F such that $F' = O(F)$. The second phase is the evolution of the obfuscated access rights function $F' = O(F)$.

Obfuscation Phase: In this first phase, obfuscator O takes as input access rights function F represented as a Boolean formula f [3].

INPUTS: Boolean formula f.

OUTPUTS: Obfuscated Boolean formula $f' = O(f)$.

a) The Boolean formula f is converted into a functionally equivalent branching program BP_f using the approach of Sauerhoff [9].

b) The branching program BP_f is mapped to a matrix branching program MBP_f, which consists of a sequence of pairs of matrices. A matrix branching program of width W for n-bit input is given by a tuple:

$$MBP_f = (I_{W\times W}, P_{rej}, inp(i), B_{i,0}, B_{i,1})_{i=1}^{L},$$

where $B_{i,b} \in \{0,1\}^{W\times W}$ – permutation matrices, $b \in \{0,1\}$; $inp(i):[L]\to[n]$ – is the input-bit selection function; $I_{W\times W}$ – identity matrices. Also create a dummy branching program $MBP'_f = (I_{W\times W}, P_{rej}, inp(i), \{I_{W\times W}\})_{i=1}^{L}$. The output of MBP_f on input $x \in \{0,1\}$ is defined as:

$$MBP_f(x) = \begin{cases} 1, \ if \ \prod_{i=1}^{L} B_{i,inp(i)} = I_{W\times W}; \\ 0, if \ \prod_{i=1}^{L} B_{i,inp(i)} = P_{rej}. \end{cases}$$

c) The matrix branching program MBP_f is randomized to get $M\tilde{B}P_f$.

Generate a graded encoding scheme with a zero-testing parameter at level – 1^{L+2}. Let $params$ be the public parameters of the GES. Choose a prime p such that message in Z_p can be encoded by the GES. And define a procedure randomization as follows:

- Choose random scalar

$$\{\alpha_{i,0}, \alpha_{i,1}, \alpha'_{i,0}, \alpha'_{i,1} \in Z_p : i \in [L]\}$$

such that

$\prod_{i\in I_j} \alpha_{i,0} = \prod_{i\in I_j} \alpha'_{i,0}$, $\prod_{i\in I_j} \alpha_{i,1} = \prod_{i\in I_j} \alpha'_{i,1}$;

- Choose two sets of random $2L\times 2L$ diagonal matrices $d_{i,b}, d'_{i,b}$ and compute

$$\begin{cases} D_{i,b} = \begin{bmatrix} d_{i,b} & 0 \\ 0 & \alpha_{i,b} B_{i,b} \end{bmatrix}; \\ D'_{i,b} = \begin{bmatrix} d'_{i,b} & 0 \\ 0 & \alpha'_{i,b} I_{W\times W} \end{bmatrix}. \end{cases}$$

- Choose two set of random $2(L+1)$ invertible matrices of $(2L+W)(2L+W)$ size $R_0, R_1, ..., R_L, R'_0, R'_1, ..., R'_L \in Z_p$;
- Let

$$\tilde{D}_{i,b} = R_{i-1} \cdot D_{i,b} \cdot (R_i)^{-1}$$

and

$\tilde{D}'_{i,b} = R'_{i-1} \cdot D'_{i,b} \cdot (R'_i)^{-1} \quad \forall i \in [n], b \in \{0,1\}$;

- Choose $(2L+W)$ vectors s, t and s', t' such that $s = (\bar{0}, \bar{s}_R, \hat{s})$, $t = (\bar{t}_R, \bar{0}, \hat{t})^T$, $s' = (\bar{0}, \bar{s}'_R, \hat{s}')$, $t' = (\bar{t}'_R, \bar{0}, \hat{t}')^T$, $\langle \hat{s}, \hat{t} \rangle = \langle \hat{s}', \hat{t}' \rangle$;
- Compute vectors $\tilde{s} = s \cdot R_0^{-1}$, $\tilde{t} = R_L \cdot t$ and $\tilde{s}' = s' \cdot (R'_0)^{-1}$, $\tilde{t}' = R'_L \cdot t'$.

d) The randomized matrix branching program is then encoded, matrix element by matrix element, using a graded encoding scheme.

Use the GES to generate level – e_1 encoding s_{enc}, s'_{enc} of $\tilde{s} = s \cdot R_0^{-1}$ and $\tilde{s}' = s' \cdot (R'_0)^{-1}$ respectively. Similarly, generate level – e_{L+2} encodings t_{enc}, t'_{enc} of $\tilde{t} = R_L \cdot t$ and $\tilde{t}' = R'_L \cdot t'$ respectively.

Encode the entries of each matrix $\tilde{D}_{i,b}$ and $\tilde{D}'_{i,b}$ at level – e_{i+1}. Call the encoded matrices $\hat{D}_{i,b}$ and $\hat{D}'_{i,b}$. Then the obfuscated program is comprised of:

$$O(f)=\begin{Bmatrix} s_{enc}, s'_{enc}, t_{enc}, t'_{enc}, \hat{D}_{i,b}, \hat{D}'_{i,b}, \\ params, inp(i) \end{Bmatrix}.$$

Evaluation phase: After receiving $f' = O(f)$ obfuscated access rights function performs the identity user task on the user identifier ID_{user} as the input x. We check whatether it sastisfies an underlying rule in f' sequentially.

INPUTS: x (user identifier ID_{user})

OUTPUTS: Obfuscated Boolean formula $O(f(x))$.

To evaluate an obfuscation of f on the input x use $params$ to zero-test the matrix

$$(\tilde{s}.\prod_{i=1}^{L}\tilde{D}_{i,inp(i)}.\tilde{t} - \tilde{s}'.\prod_{i=1}^{L}\tilde{D}'_{i,inp(i)}.\tilde{t}')$$

If the zero-test returns 1 (true), then the primal and dummy produce the same result on the input x, and since the dummy programs always compute $I_{W\times W}$, the primal program also computes $I_{W\times W}$. Therefore:

$$O(f(x))=\begin{cases} 1 \; if \; isZero = 1 \\ 0 \; otherwise \end{cases}.$$

After this phase, if $O(f(x))=1$ this means that access permissions check is successfull, access rights function will recover key to decrypt documented after extracting it from container.

C. Security proof

In [10], there were formulated and proved series of statements based on random oracle model to prove security of the proposed approach.

Using random oracle model is one of the main directions to prove cryptographic protocol security. Oracle is auxiliary algorithm that performs in a single step of main algorithm. The set of proposed and proved statements allowed constructing secure obfuscator satisfying virtual black box property. This fact means that no algorithm can distinguish obfuscated program more efficiently then hypothesis made by viewing inputs and outputs of obfuscated program and probability of getting information from obfuscation is about 2^{-l}, where l – user identifier length which should be not less than 256-bit to provide security purpose.

Firstly, we proved that indistinguishability obfuscation algorithm is polynomial time. Then using Killian theorem and Schwartz-Zippel lemma we proved that probability of getting zero in ZeroTest is negligible. That is why the task to guess input vector that gives zero in ZeroTest reduces to solving NP-complete propositional satisfiability problem (SAT).

IV. CONCLUSION

Security is very important for efficient communications. Encryption and obfuscation are two major branches of data security. In this proposed system encryption and obfuscation is combined to give two tier securities for data. Documented file is encrypted before embedding it into the container which gives high security to data. GOST R 34.12-2015 algorithm is used to encrypt documented file and indistinguishability obfuscation method is used to obfuscate contents of access rights function that allows determining who has access to document and recovering decryption key. We think the framework we proposed also works for other file format as audio and video file. But since their configurations and structures are different from documented file formats, so the specific solutions to them will also vary. We leave this as our future work.

REFERENCE

[1]. B. Barak, O. Goldreich, R. Impagliazzo, S. Rucich, A. Sahai, S. Vadhan, and K. Yang, “On the (Im) possibility of obfuscating programs”, J. ACM, vol. 59, no. 2, pp. 1-48, May 2012.

[2]. Federal Agency on Technical Regulating and Metrology, “Information technology. Cryptographic data security. Block ciphers. GOST R 34.12-2015”, 2015.

[3]. S. Garg, C. Gentry, S. Halevi , Candidate “Indistinguishability Obfuscation and Functional Encryption for all circuits”. IACR Cryptology ePrint Archive, no. 451, 2013.

[4]. I. Goldberg, D. Wagner, R. Thoms and E.A. Brewer, “A secure environment for untrusted helper applications: confining the wily hacker”, In USENIX Security Symposium.

[5]. Z. Liang, V. Venkatakrishanan and R. Sekar, “Isolated program execution: An application transparent approach for executing untrusted programs”, In Proceedings of Annual Computer Security Applications Conference.

[6]. D. Boneh and A. Silverberg, “Applications of multilinear forms to cryptography”, vol. 324, no. 1, pp. 71-90, 2003.

[7]. S. Garg, C. Gentry and S. Halevi, “Candidate multilinear maps from ideal lattices”. In Advances in Cryptology – EUROCRYPT, pp. 1-17, 2013.

[8]. J. S. Coron, T. Lepoint and M. Tibouchi, “Practical multilinear maps over the integers”, In Advances in Cryptology – CRYPTO, pp.476-493, 2013.

Hội thảo "Một số vấn đề chọn lọc về an toàn an ninh thông tin" (Symposium on Infomation Sercurity - SoIS) sẽ được tổ chức với sự phối hợp của 8 Cơ sở đào tạo trọng điểm về an ninh, an toàn thông tin tại Việt Nam.

Hội thảo sẽ là diễn đàn khoa học thường niên để cán bộ nghiên cứu, triển khai, giảng dạy và quản lý trong lĩnh vực an toàn thông tin trao đổi, chia sẻ kinh nghiệm và tìm kiếm sự hợp tác. Hội thảo khuyến khích các nghiên cứu sinh, học viên cao học và nhà khoa học trẻ tham gia báo cáo, trao đổi kết quả nghiên cứu của bản thân, và đặc biệt những kết quả nghiên cứu có tính ứng dụng trong thực tiễn.

Hội thảo lần thứ nhất "Một số vấn đề chọn lọc về an toàn an ninh thông tin" được tổ chức vào ngày 28/11/2016 tại Học viện Kỹ thuật Mật mã (Website: http://sois.actvn.edu.vn/).

Chủ đề chính của Hội thảo năm nay bao gồm: Mật mã ứng dụng; An toàn thông tin ứng dụng; An toàn, an ninh mạng; Quản lý và đào tạo an toàn, an ninh thông tin.

Trưởng Ban Tổ chức Hội thảo là TS. Nguyễn Minh Hồng, Thứ trưởng Bộ Thông tin và Truyền thông, Trưởng Ban điều hành Đề án 99. Phó Trưởng Ban Tổ chức Hội thảo là TS. Nguyễn Nam Hải, Giám đốc Học viện Kỹ thuật Mật mã. Ngoài ra Ban tổ chức còn 9 thành viên đại diện lãnh đạo Cục An toàn thông tin, Hiệp hội An toàn thông tin Việt Nam và các Cơ sở đào tạo trọng điểm về an toàn thông tin.

Ban Chương trình Hội thảo gồm các nhà khoa học, chuyên gia trong lĩnh vực an toàn thông tin trong và ngoài nước.

Các bài báo cáo gửi tham gia Hội thảo (theo định dạng của IEEE, gửi qua hệ thống EasyChair: *http://easychair.org/conferences/?conf=sois2016*) có chất lượng tốt, sau quy trình phản biện sẽ được lựa chọn đăng trong Kỷ yếu Hội thảo (xuất bản trước khi diễn ra Hội thảo).

Sau Hội thảo, những báo cáo đạt chất lượng tốt sẽ được giới thiệu để xét duyệt nhận trên Tạp chí Công nghệ thông tin và Truyền thông - Chuyên san Các công trình nghiên cứu, phát triển và ứng dụng Công nghệ thông tin và Truyền thông và một số bài khác được giới thiệu vào chuyên san "Nghiên cứu Khoa học và Công nghệ trong lĩnh vực An toàn thông tin" của Tạp chí An toàn thông tin.

[9]. M. Sauerhoff, I. Wegener and R. Werchner, "Relating branching program size and formula size over the full binary basis" In Proceedings of 16th STACS. pp. 57-67, 1999.

[10]. A. V. Kozachok, M. V. Bochkov and L. M. Tuan, "Indistinguishable obfuscation security theoretical proof", Cybersecurity issues, no. 1(14), pp. 36-46, 2016.

AUTHORS PROFILE

PhD. Alexander Kozachok

Workplace: The Academy of Federal Guard Service of the Russian Federation.

Email: alex.totrin@gmail.com

The education process: received his Ph.D. degree in Engineering Sciences in Academy of Federal Guard Service of the Russian Federation in Dec 2012.

Research today: information security, unauthorized access protection, mathematical cryptography, theoretical problems of computer science.