Analysis of the Implementation and Computational Costs for the Cryptosystems on Suzuki Group

Gennady Khalimov
Information Security Department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
gennadykhalimov@gmail.com

Yevgen Kotukh
Information Security Department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
yevgenkotukh@gmail.com

Аналіз Реалізації та Обчислювальні Витрати для Криптосистем на SuzukiGroup

Геннадій Халімов
Факультет інформаційної безпеки
Харківський національний університет радіоелектроніки
Харків, Україна
gennadykhalimov@gmail.com

Євген Котух
Факультет інформаційної безпеки
Харківський національний університет радіоелектроніки
Харків, Україна
yevgenkotukh@gmail.com

***Abstract*—The paper considers the main implementations of cryptosystems in groups and an analysis of the estimation of complexity of calculations. The analysis of the cryptosystems implementation based on Suzuki group is presented. The design and implementation peculiarities of the Suzuki 2-group based MST3 cryptosystem are analyzed. The comparative results of encryption and decryption computation costs for the finite field of 128 bits, 256 bits, as well as implementation for the RSA algorithm are obtained. It follows from the evaluation that, for example, the encryption and decryption time of the RSA algorithm is 10 times bigger than the MST3 cryptosystem, but it much more cost effective in terms of the size of private and public keys.**

***Анотація*—В роботі розглядаються основні реалізації криптосистем у групах та аналіз оцінки складності розрахунків. Представлений аналіз впровадження криптосистем на групі Suzuki. Проаналізовано особливості розробки та реалізації криптосистеми MST3 на базі 2-х груп Suzuki. Отримані порівняльні результати розрахунку шифрування та дешифрування для кінцевого поля 128 біт, 256 біт, а також реалізація алгоритму RSA. З оцінки випливає, що, наприклад, час шифрування та дешифрування алгоритму RSA в 10 разів перевищує криптосистему MST3, але набагато більш економічно ефективний з точки зору розміру приватних та відкритих ключів. функціонування систем електронної взаємодії органів виконавчої влади.**

***Keywords*—Suzuki 2-group, logarithmic signature, Computational,MST_3**

***Ключові слова*—Сузукі 2-груп, логарифмічний підпис, обчислення, MST_3**

I. INTRODUCTION

In the early 80's, the use of group theoretical problems for cryptography was proposed by Wagner and Magyarik [1], Wagner [2], Magliveras [3]. Magliveras et al were made the proposals for cryptographic schemes based on special expanded finite groups (so-called logarithmic signatures) [3]. Logariphmic signatures and their cryptographic application were studied by Gonzağlez Vasco, Steinwandt, Birget, Bohliet, and others authors. These decompositions are interesting by themselves like mathematical objects. For example, Hajors work on Minkowski's hypothesis shows that this type of decomposition for abelian groups arises in the study of multidimensional coverings (see [4]).

MST_1, MST_2 and MST_3 are examples of public key cryptosystems. The construction of short logarithmic signatures is the actual issue of their implementation. Logarithmic signatures are the special type of group decomposition are presented as the main components of some cryptographic keys. In this connection, scientific interest corresponds to the search of the logarithmic signatures in the finite groups (such decompositions exist for solvable, symmetric and alternate groups) and assessment of their practical feasibility and secrecy. The basic definitions of logarithmic signatures, coverings for finite groups and their mapping generations, as well as the structure of the given cryptosystems are presented in [4].

II. DESING AND IMPLEMENTATION PECULIARITIES OF MST$_3$ CRYPTOSYSTEM ON SUZUKI 2-GROUP

Suzuki 2-group with order of q^2 is proposed in the generic implementation of MST_3 cryptosystem. Using the notation of Higman [5], Suzuki 2-group with order of q^2 is noted as $A(\mathrm{m},\theta)$. Let $q=2^m$ з $3\le m\in N$ is such, that F_q field has nontrivial automorphism θ of unpaired order. Here it means that m is not degree of 2. Than groups of $A(\mathrm{m},\theta)$ are exist.

In fact, if we determine $\varsigma:=\{S(a,b)\mid a,b\in \mathrm{F}_q\}$, where $S(a,b)=\begin{pmatrix}1&0&0\\ a&1&0\\ b&a^\theta&1\end{pmatrix}$ is the matrix 3×3 over the field F_q, it shows that group ς is isomorphic to $A(\mathrm{m},\theta)$. So, ς has the order of q^2 and we have

$$Z:=\mathbf{Z}(\varsigma)=\Phi(\varsigma)=\varsigma'=\Omega_1(\varsigma)=\{S(0,b)\mid b\in \mathrm{F}_q\}$$

Since the center $\mathbf{Z}(\varsigma)$ is an elementary Abelian group of the order q, it can be identified with the additive group of field F_q. Besides, factor-group $\varsigma/\Phi(\varsigma)$ is elementary Abelian group of order q. Then it's easy to check that the multiplication of the two elements in ς is carried out in accordance with the rule

$$S(a_1,b_1)S(a_2,b_2)=S(a_1+a_2,b_1+b_2+a_1^\theta a_2).$$

Finding the inverse element is performed by the formula

$$S(a,b)^{-1}=S(a,b+a^{\theta+1}).$$

The algorithm of the system for encryption has the following stages [6].

A. Generate of the key data

1. Choose the big group $G=A(m,\theta),\ q=2^m$.

2. Generate factorizable logarithmic signature $\beta=[B_1,...,B_s]=(b_{i,j})=(S(0,b_{i,j}.b))$ of $(r_1,...,r_s)$ type, where $b_{i,j}.b\in F_q$.

3. Generate random covering

$$\alpha=[A_1,...,A_s]=(a_{i,j})=(S(a_{i,j}.a,a_{i,j}.b))$$

of the same type of β, where $a_{i,j}.a\in F_q/\{0\}, a_{i,j}.b\in F_q$.

4. Generate random values $t_0,t_1,...,t_s\in G$, matrix of random bits $\sigma=[q\times q]$.

5. Construct homomorphism $f:G\to Z$, defined as $f(S(a,b))=S(0,g(a))$ (in this implementation, the multiplication by a random bit matrix $f(a)=a^\sigma$ was used).

6. Compute

$$\gamma=[H_1,...,H_s]=(h_{i,j})=(S(h_{i,j}.a,h_{i,j}.b)),$$

where $h_{i,j}=t_{i-1}^{-1}*a_{i,j}*t_i*b_{i,j}*f(a_{i,j})$.

7. Public key - $[\alpha,\gamma]$, private key - $[\beta,(t_0,t_1,...,t_s),f]$ and additional data which is needed for the factorization of β.

B. Encryption of the message m

1. Generate element $\chi = S(0,m) \in G$
2. Generate random number $R \in Z$
3. Compute the cryptogram

$$y_1 = \alpha'(R) * \chi,\ y_2 = \gamma'(R) * \chi .$$

Remark

To reduce the size of cipher text enough to save $(y_{1.a}, y_{1.b}, y_{2.b})$, when decrypting the component $y_{2.a}$ can be restored by the formula $y_{2.a} = y_{1.a} \oplus t_{0.a} \oplus t_{S.a}$

C. Decryption

1. Compute

$$\beta'(R) = f(y_1)^{-1} * y_1^{-1} * t_0 * y_2 * t_S^{-1} .$$

2. Factorize $R = \beta'^{-1}(R)$.
3. Compute $\alpha'(R)$.
4. Restore $m = y_{1.b} \oplus \alpha'(R)_{.b}$.

Encryption testing is performed on a computer running OS Ubuntu 16.04 with Intel® Core™ i7-4702MQ CPU @2,20 GHz processor, 12 Gb RAM. The results are presented in Tables 1,2.

TABLE I. ENCRYPTION AND DECRYPTION COMPUTATIONAL COSTS IN THE 128BITS FINITE FIELD

Partition classes	Time of the key data generation, ms	Private key size, bytes	Public key size, bytes	Encryption time for 100 KB, ms	Decryption time for 100 KB, ms
128[2] →64[4]	*56*	*78830*	*39761*	*4749*	*2711*
64[4] →32[16]	*59*	*111726*	*75217*	*2388*	*1487*
32[16] →16[256]	*169*	*671918*	*590609*	*1205*	*888*

TABLE II. ENCRYPTION AND DECRYPTION COMPUTATIONAL COSTS IN THE 256BITS FINITE FIELD

Partition classes	Time of the key data generation, ms	Private key size, bytes	Public key size, bytes	Encryption time for 100 KB, ms	Decryption time for 100 KB, ms
256[2] →128[4]	*57*	*249630*	*128593*	*14811*	*7911*
128[4]→ 64[16]	*106*	*361502*	*248657*	*7540*	*4196*
64[16] →32[256]	*798*	*2193054*	*1967569*	*3782*	*2318*

In the Table. III a comparison with RSA encryption algorithm is presented.

TABLE III. ENCRYPTION AND DECRYPTION COMPUTATIONAL COSTS FOR RSA

Bitness of key parameters, bit	Time of the key data generation, ms	Private key size, bytes	Public key size, bytes	Encryption time for 100 KB, ms	Decryption time for 100 KB, ms
512	*3,368*	*342*	*92*	*66,987*	*641,277*
1024	*8,685*	*632*	*160*	*117,947*	*2116,400*
2048	*63,658*	*1214*	*292*	*243,887*	*9853,580*
4096	*707,645*	*2373*	*548*	*591,868*	*64250,400*

CONCLUSIONS

1.It is necessary to select a partition class of the logarithmic sub-block into blocks to optimize the computational costs for the size of private and public keys, the time for encryption and decryption. Time costs can be reduced by several times. The use of the final field of 128, 256 bits is sufficient to provide the highest class of security in the cryptosystems` classification.

2. During the calculation of 2048 and 4096 bits in the finite field, the encryption and decryption time of the RSA algorithm is tens of times larger than the MST_3 cryptosystem, but it ensures significant cost savings for the size of private and public keys.

REFERENCES

[1] N.R. Wagner and M. R. Magyarik. "A Public Key Cryptosystem Based on the Word Problem." In Advances in Cryptology. Proceedings of CRYPTO 1984, pp. 19—36, edited by G. R. Blakley and D. Chaum, Lecture Notes in Computer Science 196. Berlin: Springer, 1985.

[2] N.R. Wagner. "Searching for Public-Key Cryptosystems." In Proceedings of the 1984 Symposium on Security and Privacy (SSP '84), pp. 91—98. Los Alamitos, CA: IEEE Computer Society Press, 1990.

[3] S.S. Magliveras. "A Cryptosystem from Logarithmic Signatures of Finite Groups." In Proceedings of the 29th Midwest Symposium on Circuits and Systems, pp. 972—975. Amsterdam: Elsevier Publishing Company, 1986.

[4] W. Lempken, S.S. Magliveras, Tran van Trung and W. Wei, A public key cryptosystem based on non-abelian finite groups, J. of Cryptology, 22(2009), 62–74.

[5] G. Higman, Suzuki 2-groups.Ill. J. Mathematic.-1963.-V.7. –P.79–96.

[6] Pavol Svaba. Covers and logarithmic signatures of finite groups in cryptography. Dissertation, Bratislava, Slowakische Republik – 2011.