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 Suzuki Group

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. Представленій 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, MST3

Ключові слова—Сузукі 2-груп, логарифмічний підпис, обчислення, MST3
I. INTRODUCTION

In the early 80's, the use of group theoretical problems for cryptography was proposed by Wagner and Mayarik [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]. Logarithmic signatures and their cryptographic application were studied by González 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]).

$\text{MST}_1$, $\text{MST}_2$ and $\text{MST}_3$ 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. DESIGN AND IMPLEMENTATION PECULIARITIES OF MST, CRYPTOSYSTEM ON SUZUKI 2-GROUP

Suzuki 2-group with order of $q^2$ is proposed in the generic implementation of $\text{MST}_1$, cryptosystem. Using the notation of Higman [5], Suzuki 2-group with order of $q^2$ is noted as $A(m, \theta)$. Let $q = 2^m \times 3 \leq m \in N$ is such, that $F_q$ field has nontrivial automorphism $\theta$ of unpaired order. Here it means that $m$ is not degree of 2. Than groups of $A(m, \theta)$ are exist.

In fact, if we determine $\zeta := \{ S(a, b) | a, b \in F_q \}$, where $S(a, b) = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & a^\theta & 1 \end{pmatrix}$ is the matrix $3 \times 3$ over the field $F_q$, it shows that group $\zeta$ is isomorphic to $A(m, \theta)$. So, $\zeta$ has the order of $q^2$ and we have $Z = Z(\zeta) = \Phi(\zeta) = \zeta' = \Omega(\zeta) = \{ S(0, b) | b \in F_q \}$.

Since the center $Z(\zeta)$ is an elementary Abelian group of the order $q$, it can be identified with the additive group of field $F_q$. Besides, factor-group $\zeta / \Phi(\zeta)$ is elementary Abelian group of order $q$. Then it's easy to check that the multiplication of the two elements in $\zeta$ is carried out in accordance with the rule $S(a, b)S(a, b) = S(a_1 + a_2, b_1 + b_2 + a_2^\theta a_1)$. 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 $\beta$, 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 $h_{i,j} = t_0^{-1} \ast a_{i,j} \ast t_1 \ast b_{i,j} \ast f(a_{i,j})$. 

209
7. Public key \([-[\alpha, \gamma], \text{ private key } [\beta, (t_0, t_1, \ldots, t_s), f]\) and additional data which is needed for the factorization of \(\beta\).

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
   \[y_1 = \alpha'(R) \star \chi, \quad y_2 = \gamma'(R) \star \chi.\]

Remark
To reduce the size of cipher text enough to save \((y_{1.\alpha}, y_{1.\beta}, y_{2.\beta})\), when decrypting the component \(y_{2.\alpha}\), can be restored by the formula \(y_{2.\alpha} = y'_{1.\alpha} \oplus t_{0.\alpha} \oplus t_{s.\alpha}\).

C. Decryption

1. Compute
   \(\beta'(R) = f(y_{1.\beta})^{-1} \star y_{1.\alpha} \star t_0 \star y_{2.\beta} \star t_s^{-1}.\)
2. Factorize \(R = \beta^{-1}(R)\).
3. Compute \(\alpha'(R)\).
4. Restore \(m = y_{1.\alpha} \oplus \alpha'(R)_{\beta}\).

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>
<thead>
<tr>
<th>Partition classes</th>
<th>Time of the key data generation, ms</th>
<th>Private key size, bytes</th>
<th>Public key size, bytes</th>
<th>Encryption time for 100 KB, ms</th>
<th>Decryption time for 100 KB, ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>32[16] \rightarrow 16[256]</td>
<td>169</td>
<td>671918</td>
<td>590609</td>
<td>1205</td>
<td>888</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Partition classes</th>
<th>Time of the key data generation, ms</th>
<th>Private key size, bytes</th>
<th>Public key size, bytes</th>
<th>Encryption time for 100 KB, ms</th>
<th>Decryption time for 100 KB, ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>128[4] \rightarrow 64[16]</td>
<td>106</td>
<td>361502</td>
<td>248657</td>
<td>7540</td>
<td>4196</td>
</tr>
<tr>
<td>64[16] \rightarrow 32[256]</td>
<td>798</td>
<td>2193054</td>
<td>1967569</td>
<td>3782</td>
<td>2518</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Bitness of key parameters, bit</th>
<th>Time of the key data generation, ms</th>
<th>Private key size, bytes</th>
<th>Public key size, bytes</th>
<th>Encryption time for 100 KB, ms</th>
<th>Decryption time for 100 KB, ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
<td>3,368</td>
<td>342</td>
<td>92</td>
<td>66,987</td>
<td>641,277</td>
</tr>
<tr>
<td>1024</td>
<td>8,685</td>
<td>652</td>
<td>160</td>
<td>117,947</td>
<td>2116,400</td>
</tr>
<tr>
<td>2048</td>
<td>65,658</td>
<td>1214</td>
<td>292</td>
<td>243,887</td>
<td>9853,580</td>
</tr>
<tr>
<td>4096</td>
<td>707,645</td>
<td>2373</td>
<td>548</td>
<td>591,868</td>
<td>64250,400</td>
</tr>
</tbody>
</table>
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


