ElGamal with Differentiated Decryption on K+1 Access Levels

Stelian Flonta, Liviu Cristian Miclea and Izabella Diana Păun
Technical University of Cluj-Napoca, Romania
sflonta@colim.ro, Liviu.Miclea@aut.utcluj.ro, paunitza1@yahoo.com

Abstract—The algorithm is an extension of the El Gamal encryption system. The difference from the classical form is the way the keys are generated and distributed. This system allows the encryption of the message $m_i$ by Alice and its decryption by Bob. Moreover, there exists the possibility of decrypting a group of messages or all the messages by other beneficiaries, each using a secret special key. This key is distributed depending on the access level, meaning, on the access right. The user of the key of level 0 can decrypt all messages, the user which holds one of the keys of level 1 can decrypt a subset of the set of messages that can be decrypted with the key of level 0, the one which holds one of the keys of level 2 can decrypt a subset of the set of messages that can be decrypted with the key of level 1 and so on, and the one which holds one of the keys of level k can decrypt a message $m_i$.

Keywords: The system EG(k+1)GA, ElGamal with Differentiated Decryption, public key of level P(k+1)K, secret key of level k SK(k+1)Gk.

I. INTRODUCTION

The paper starts with a few mathematical preliminaries which have the scope of presenting the terms used throughout the presentation. By the term information it is presented the following set:

$$M = \{m_{ijk} \mid m_{ijk} \in \{0,1\}^P, \forall ijk \in N\}$$

which is a set of messages. The encryption of the information M can be realized by the encryption of each message. The access to an encrypted information means that the user can decrypt a part of or all of the encrypted messages, depending on the user’s rights. The central subject of this paper is the description of an encryption system elaborated to ensure the access to an encrypted information in a differentiated way. This system is structured on levels of access [7][8]. The algorithm is based on the El Gamal encryption system [1][3][4] and, in this article, it will be further called the ElGamal with differentiated decryption on k+1 access levels (EG(k+1)GA), this taking part of the category encryption systems with a public key which is based on the discrete logarithm problem. The extended system (EG(k+1)GA) is described in the next part of the paper. Generally, the encryption systems allow a user to encrypt the information M and another one to decrypt it. Such a possibility is offered by the ElGamal system. Unlike these, (EG(k+1)GA) allows the information M to be encrypted by more than one user, and the access to it, by decryption, to be possible to more users. The messages from the information M, which will be accessed by decryption by a user, will be those for which the user has rights. These rights can be organized with the help of access levels [8][9]. Between levels there are hierarchical links. These valences of (EG(k+1)GA) are possible because of the links between the keys. The presentation emphasizes the way the keys are generated, distributed and the links between them. The personal contributions, conclusions and application ideas are illustrated at the end of the paper.

II. PRELIMINARIES

Next, a few notions regarding trees will be presented in order to introduce the terminology. The nodes of a tree are placed on levels. So, the root of the tree is situated on level 0, its direct descendants on level 1, the direct descendants of a node on level k are situated on level k+1. The maximum height of a tree (depth of the tree) is the maximum level of a node in the tree. If there is an arc from node x to node y it is said that y is the child of node x or x is the parent of node y. A node y is the sibling of node z if they have the same parent. A leaf is a node without any children.
Stepping through the items of a tree can be realized in numerous ways, out of which A-preorder means visiting the root, then the subtrees which have as roots its descendants. For the following tree:

Figure 1. The representation of a tree.

The order of traversal in the A-preorder way is:
1, 2, 5, 6, 7, 3, 8, 9, 4.

A family of subsets of set E is called a partition of set E if:
• No subset is null
• All subsets are disjunctive two by two
• The union of all the subsets is equal to E

Within the framework of the operations that follow, in order to simplify the expression, it will be calculated modulo q without writing explicitly, where q is the one chosen in the key generation stage. For instance \( h=g^x \) means \( h=g^x \mod q \).

III. ELGAMAL WITH DIFFERENTIATED DECRYPTION ON K+1 ACCESS LEVELS

Let \( I \subseteq N^* x N^* x N^* \ldots N^* \) be a set of k indices. Next, the following notations will be used:
\[
i_{1..k} = (i_1, i_2, \ldots, i_k) = \overline{i_1 i_2 \ldots i_k} \\
\]
\[
k_{-j} = (i_1, i_2, \ldots, i_{k-j}) = \overline{i_1 i_2 \ldots i_{k-j}} \\
\]
The information \( \{m_{i..k} | i \in I \} \) is considered.

For a given node, the indices of the parent node are written, and finally, the number which represents the order in the set of siblings of this node. It is obvious that at the exponent the level of access is written. The level \( k+1 \) of the leaves is implicitly understood, and it is not written in order not to create confusion. The

<table>
<thead>
<tr>
<th>( f' )</th>
<th>( f'_1 )</th>
<th>( f'<em>1</em>{11} )</th>
<th>( f'<em>1</em>{12} )</th>
<th>( f'_2 )</th>
<th>( f'<em>2</em>{21} )</th>
<th>( f'<em>2</em>{22} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z_{111} )</td>
<td>( z_{112} )</td>
<td>( z_{113} )</td>
<td>( z_{121} )</td>
<td>( z_{122} )</td>
<td>( z_{211} )</td>
<td>( z_{212} )</td>
</tr>
<tr>
<td>( m_{111} )</td>
<td>( m_{112} )</td>
<td>( m_{113} )</td>
<td>( m_{121} )</td>
<td>( m_{122} )</td>
<td>( m_{211} )</td>
<td>( m_{212} )</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 3. The representation of AK5GA and of the messages.
above scheme “says” which is the “power” of decryption of a key, meaning that using a certain key, all the messages which are below it can be decrypted.

Following the facts stated above, AK(k+1)GA is constructed and it looks like in the Figure 6.

For a clearer mode of expression, the following notations will be used:
- The level which ensures the access to all messages \( m_{L} \) is 0, the immediately following one is 1, then 2 and the one which ensures the access to a single \( m_{L} \) is \( k \).
- For the decryption of message \( m_{L} \), the procedure is:
  1. The access level 0 is ensured by \( f^0 \).
  2. The access level 1 is ensured by \( f^1 \).
  3. The access level 2 is ensured by \( f^2 \).

Graphically, these ideas can be expressed in the following way:

\[
\begin{align*}
\text{Decryption of order 0 :} & \quad f^0 \\
\text{Decryption of order 1 :} & \quad f^1 \\
\text{Decryption of order 2 :} & \quad f^2 \\
\text{Decryption of order } k & : f^k
\end{align*}
\]

B. The keys generation

Next, a mathematical model for generating keys and distributing them will be presented, such that the requirements of the presented system to be fulfilled. In the computations that follow, in order to simplify the mode of expression, modulo \( q \) will be computed, if the elements are from \( Z_q \), without explicitly writing this fact.

A cyclic group of order \( q \) prime number is chosen, for which the problem of discrete logarithm is difficult, \( g \) being one of its generators. From the set \( \{0, \ldots, q-1\} \) k different elements \( x_1, x_2, \ldots, x_k \) are chosen. Also, the following functions are chosen:

\[
\begin{align*}
\theta_i : & N^* \rightarrow N^*, \quad i = 1, \ldots, k \\
\end{align*}
\]

These functions are irreducible polynomial. The secret keys \( f \) are functions defined as shown below:

\[
\begin{align*}
\text{SK(k+1)G1} & \\
& f^0 : N^* \times N^* \times N^* \times \ldots \times N^* \rightarrow Z_q^* \\
& f^1_{\theta_1(n_1), \ldots, \theta_k(n_k)} = x_1^{\theta_1(n_1)} \times x_2^{\theta_2(n_2)} \times \ldots \times x_k^{\theta_k(n_k)} \\
& \text{SK(k+1)G2} \\
& f^1_{\theta_1(n_1), \ldots, \theta_k(n_k)} = x_1^{\theta_1(n_1)} \times x_2^{\theta_2(n_2)} \times \ldots \times x_k^{\theta_k(n_k)} \\
& \text{SK(k+1)G3} \\
& f^1_{\theta_1(n_1), \ldots, \theta_k(n_k)} = x_1^{\theta_1(n_1)} \times x_2^{\theta_2(n_2)} \times \ldots \times x_k^{\theta_k(n_k)}
\end{align*}
\]
A function of order $s$ is obtained from that of order $s-1$, assigning values to the variable $n_s$, starting with the natural number 1. If the system has $k+1$ access levels, then the function $f_0$ has $k$ variables, $f_1$ has $k-1$ variables, $f_2$ has $k-2$ variables, and finally, after the values of all the variables are given, the secret key $z_{i...}$ is obtained. This fact is proved by the following equalities:

$$f^0(1,2,3,...,k) = f^1(1,2,...,k) = \quad = f^2_{ij}(1,2,3,...,i,k) = \quad = \ldots = f^{k-1}_{ij,...,k}(1,2,3,...,i,j,...,k) = z_{i...}$$

The element $h_{i...} = g^{z_{i...}}$ is associated to the element $z_{i...}$. The set of public keys $P(k+1)K$ is $\{q,g,h_{i...}\}$ and the set of secret keys $SK(k+1)GA$ is $\{z_{i...}, f_{j...}, j=0,...,k-1\}$.

The secret keys are distributed to the beneficiaries, so each user $Bob_{i...}$ receives the key $z_{i...}$ and the beneficiaries with lower access levels receive keys of type $f^j$, $j=0,...,k-1$. Of course $f^j$ will be distributed to the user that will be able to decrypt all messages. Knowing $z_{i...}$ the formulas of the functions $f^j$ are difficult to compute. The system of keys generated in this way ensures solving the proposed exercise.

Due to the way the keys are defined, these can be determined ones from others only in the following way:

$$f^0 \rightarrow f^1 \rightarrow f^2 \rightarrow \ldots \rightarrow z_{i...}$$

$$c^2_{\alpha \alpha \ldots \alpha \alpha} \frac{m_{\alpha \alpha \ldots \alpha \alpha} (i,j,y,...,j_v)}{g^{y (f^{ji}_{a_{i,j},a_{y,j}}) (i,j,y,...,j_v})} = (g^{y (f^{ji}_{a_{i,j},a_{y,j}}) (i,j,y,...,j_v}) = \frac{m_{\alpha \alpha \ldots \alpha \alpha} (i,j,y,...,j_v)}{g^{y (f^{ji}_{a_{i,j},a_{y,j}}) (i,j,y,...,j_v})}$$

for each $\alpha \alpha \ldots \alpha$

From these computations, it results the fact that in order to decrypt the message with a secret key, a chain is necessary to exist formed only by descending nodes from the analyzed key to the leaf $z_{i...}$ corresponding to the message $m_{i...}$. Of course, the difference between the initial tree and the final tree is greater than a level, the computations are executed in more than one step.

C. Message Encryption

Knowing the public key, Alice_{i...} encrypts the message $m_{i...}$ in the following way: the elements $y_{i...}$ from the set $\{0,...,q-1\}$ are chosen and $c^1_{i...} = g^{y_{i...}}$, $c^2_{i...}$ are computed. The encrypted message is $(c^1_{i...}, c^2_{i...})$. Obviously, the same set of indices will be used everywhere. In this case, $D(k+1)K = \{y_{i...}\}$.

D. Message Decryption

In order to decrypt the message $(c^1_{i...}, c^2_{i...})$, user Bob_{i...} employs $q$ and the secret key $\{z_{i...}\}$, computing $c^2_{i...} = c_1^{y_{i...}} - m_{y_{i...}} h_{i...} = q^{y_{i...}} - m_{y_{i...}} g^{z_{i...}} = c_{i...}$

This way, the decryption of order $k$ is realized. If the decryption of order $k-j$ is opted for, the below computations are needed:

$$m_{\alpha \alpha \ldots \alpha \alpha} (i,j,y,...,j_v) = m_{\alpha \alpha \ldots \alpha \alpha} (i,j,y,...,j_v)$$

IV. THE EXTENSION AND RESTRICTION OF AK(k+1)GA

If it is necessary to add or to remove an access level, AK(k+1)GA can be extended or restricted by constructing AK(k+2)GA, respectively AK(k)GA. These computations are realized by adding respectively, removing nodes and assigning a new set of indices. If the difference between the initial tree and the final tree is greater than a level, the computations are executed in more than one step.

A. The restriction of AK(k+1)GA

In order to obtain a AK(k)GA, a subtree is chosen which has as its root the key which will become of level 0, the number that indicates the level is modified.
by decrementing with 1 from the initial one and the indices are modified by eliminating the first index from left, as it is shown in Figure 6. The functions from $SK(k)G$ are computed as follows:

$$SK(k)G_0 = SK(k+1)G_1 \cap AK(k)G \quad \text{or} \quad SK(k)G_{k-1} = SK(k+1)G_k \cap AK(k)G$$

B. The Extension of $AK(k+1)G$A

In order to obtain a $AK(k+2)G$, a new root is added which will become the key of level 0, the number that indicates the level is modified by incrementing with 1 the initial one and the indices are modified by adding in front of the first index from left the index 2, as it is shown in Figure 7. The functions from $SK(k+2)G$A are computed as follows:

From the set \( \{0, \ldots, q-1\} \) \( x_0 \) is chosen so that \( x_0, x_1, x_2, \ldots, x_k \) to be different, where \( x_1, x_2, \ldots, x_k \) are from $SK(k+1)G_0$. Also, it is chosen :

\( \theta_1 : N^* \rightarrow N^* \), a polynomial function so that \( \theta_0(1) = 0 \).

$$SK(k+2)G_0 = x_0 \theta_1(n_0) \ast SK(k+1)G_0$$

or $SK(k+2)G_0$ is made up of the function

$$f^0 : N^* \times N^* \times N^* \times \ldots \times N^* \rightarrow Z_q^*$$

$$f^0 (n_0, n_1, n_2, \ldots, n_k) = x_0 \theta_1(n_0) \theta_2(n_1) \theta_3(n_2) \ldots \theta_k(n_k)$$

From this construction it immediately results

$$SK(k+1)G_0 \subseteq SK(k+2)G_1$$

$$SK(k+1)G_1 \subseteq SK(k+2)G_2$$

Finally, $SK(k+2)G$A is filled with secret keys generated according to the model shown in the algorithm.
V. CONTRIBUTIONS

EG(k+1)GA emphasizes a specific method of generating keys, of organizing AK(k+1)GA and of distributing them. In this manner, a control of the access to the information through access levels is ensured. The construction AK(k+1)GA is based on the notions of trees, functions and sets of the remainder classes.

AK(k+1)GA has as an important characteristic: openness, meaning that it can be extended or restricted by adding or removing access levels, which means that it can add or remove levels in the tree.

VI. CONCLUSIONS

The working speed of EG(k+1)GA is similar to that of the ElGamal system. The above extension is based on the specific way of generating and distributing keys. Starting from this idea, extensions of some systems that are based on the discrete logarithm problem can be realized, for example the MOR Encryption System or the Cramer-Shoup Encryption System.

The EG(k+1)GA is advantageous in the situations when information with a large number of messages is encrypted. This information can be organized in a database where different fields are encrypted with different keys or in a distributed database.

The method of generating AK(k+1)GA allows relatively easy replacement of a AK(k+1)GA with a new one, different from the initial one.

REFERENCES