Cryptosystems for Hierarchical Groups *

Hossein Ghodosi
Josef Pieprzyk
Chris Charnes
Rei Safavi-Naini

Center for Computer Security Research
Department of Computer Science
University of Wollongong
NSW 2522
Australia
e-mail: hossein/josef/charnes/rei@cs.uow.edu.au

Abstract

This paper addresses the problem of information protection in hierarchical groups. Higher level groups of participants can control the information flow (the decryption ability) to lower level groups. If a higher level group decides to allow a lower level group to read the message, it passes a go ahead ticket so the lower level group can decrypt the cryptogram and read the message. The formal model of top-down hierarchical cryptosystems is given.

Two practical and efficient schemes are described. The first is based on the ElGamal system. The second applies the RSA system. In proposed schemes the dealer publishes a public key such that an individual can use it to send an encrypted message to the (hierarchical) group. Publication of both the group public key and the encryption method does not reveal the decision of the group.

The lack of verifiability of retrieved messages in threshold ElGamal cryptosystems is also discussed.

1 Introduction

The concept of society or group-oriented threshold cryptography was originally formulated by Desmedt [2] for "democratic" groups where every participant has equal rights

*Support for this project was provided in part by the Australian Research Council under the reference number A49530480

in performing cryptographic operations. There were numerous attempts to generalize the group-oriented cryptography for an arbitrary access structure (for extensive review refer to Desmedt [2]). At the same time, most of the organizations (or groups) exhibit distinct hierarchical structures. Cryptographic operations in hierarchical groups can be done in two different ways: from the top to the bottom or from the bottom to the top. "Top-down" cryptographic operations have to be performed after "the go ahead" is given by the higher level. The permission (or denial) for the lower level may depend upon the result of the operation performed on the higher one. "Bottom-up" cryptographic operations permit the delegation of a specific right from lower levels to the top. The top level successfully executes the cryptographic operation when all lower levels have prepared their partial results.

This paper investigates the problem of a secure top-down information flow. The proposed cryptosystems are designed in such way that the information flow can be stopped by higher level participants. The proposed "top-down" hierarchical cryptosystems are based on the ElGamal and the RSA cryptosystems.

The paper is organized as follows. Section 2 briefly describes both the ElGamal and the RSA cryptosystems. The next section examines the way the top-down information flow can be controlled. Sections 4 and 5 discuss possible implementations. In Section 6, the lack of verifiability of retrieved messages in threshold ElGamal cryptosystems is discussed. Finally, conclusions are given in Section 7.

2 Background

The ElGamal and the RSA cryptosystems are based on the discrete logarithm problem. Computations in the ElGamal system are done in a Galois field – the modulus is a prime number [6]. This cryptosystem can be seen as an extension of the Diffie-Hellman public key distribution system [5]. On the other hand computations in the RSA public key cryptosystem are done in a ring – the modulus is a composite number [8]. The two different definitions of the moduli in these cryptosystems have a profound influence into their structure and security. The security of ElGamal system is equivalent to the difficulty of computing a corresponding an instance of the discrete logarithm [7]. While the security of the RSA system is related to the difficulty of factoring the modulus.

2.1 Secret sharing and group-oriented cryptography

The concept of secret sharing was independently introduced by Blakley [1] and Shamir [9]. Their schemes allow the recreation of a secret from the collection of participant shares if the number of co-operating participants equals or exceeds some fixed threshold t. Such secret sharing schemes are called (t, n) *threshold schemes*, where $t \leq n$ is the threshold and n is the size of the group.

Shamir's (t, n) threshold scheme is based on the interpolation property of polynomials. Consider a polynomial $f(x)$ of degree $(t-1)$ over a field $GF(q)$ such that $s = f(0)$ is secret. In Shamir's scheme, $y_i = f(x_i)$ are secret shares held by participants P_i, while the values x_i are publicly known $(i = 1, \ldots, n)$. The reconstruction of the polynomial with a

subset B of t shareholders can be done as:

$$f(x) = \sum_{i \in B} y_i \prod_{\substack{j \neq i \\ j \in B}} \frac{(x - x_j)}{(x_i - x_j)} \bmod q$$

and thus the secret $s = f(0)$ can be reconstructed as:

$$s = \sum_{i \in B} y_i \prod_{\substack{j \neq i \\ j \in B}} \frac{(-x_j)}{(x_i - x_j)} \bmod q$$

In group-oriented cryptography, cryptographic operations can be successfully performed only if there is a big enough set of co-operating participants. If the co-operation of each set of t participants of a group suffices to perform the cryptographic operation, then the cryptosystem is called a (t, n) *threshold cryptosystem*, where n is the number of participants in the group.

2.2 The RSA cryptosystem

The original RSA system uses the modulus $N = pq$ where p, q are strong primes, that is $p = 2p' + 1$ and $q = 2q' + 1$ (p' and q' are large and distinct primes). The creator of the system R who knows p and q, selects a random integer e such that e and $\lambda(N)$ are coprime ($\lambda(N)$ is the least common multiplier of two integers $p - 1$ and $q - 1$ so $\lambda(N) = 2p'q'$). Next R publishes e and N as the public parameters of the system, while keeps secret the values of p, q, and d (d has the property that $ed = 1 \bmod (\lambda(N))$). It is clear that computing of d is easy for R, who knows $\lambda(N)$, but is difficult (equivalent to the factoring of N) for someone who does not know the $\lambda(N)$.

Suppose an individual wants to send R a message M, where $0 \leq M \leq N - 1$. The sender computes the cryptogram $C = M^e \bmod N$ and communicates it to R. Since for every integer a ($gcd(a, p) = 1$ and $gcd(a, q) = 1$) the equation $a^{\lambda(N)} = 1 \bmod N$ holds, R can decrypt the cryptogram using:

$$C^d = (M^e)^d = M^{ed} = M^{1(\bmod \lambda(N))} = M \bmod N$$

Desmedt and Frankel [3] proposed a (t, n) threshold RSA cryptosystem which incorporates the Shamir secret sharing scheme. In their proposed scheme, any subset B ($|B| = t$) of participants can retrieve the secret

$$s = \sum_{i \in B} s_i \bmod \lambda(N)$$

from their shares s_i. Note that $s = d - 1$. Now, to decrypt the cryptogram $C = M^e \bmod N$, all participants of an authorized subset, B, compute their own partial decryption, C^{s_i}, and send to the combiner. The combiner can decrypt the cryptogram using:

$$\prod_{i \in B} C^{s_i} \times C = C^{(\sum_{i \in B} s_i + 1)} = C^d = (M^e)^d = M \bmod N.$$

2.3 The ElGamal cryptosystem

Computations in the ElGamal cryptosystem are performed in $GF(p)$, where p is a sufficiently large prime number. The system is set up by the receiver R who selects a primitive element g – a generator of $GF(p)$ – and chooses an integer $s \in GF(p)$ at random ($s \neq 0$). Next R computes $y = g^s \bmod p$ and publishes g, p, and y as the public parameters of the system. The secret component s is known to R only.

Suppose that a user P wants to send a message M to R, where $1 \leq M \leq p-1$. First P chooses a random integer $k \in GF(p)$ and sends the pair $C = (g^k, My^k)$ as the cryptogram of M. In order to decrypt the cryptogram C and retrieve the message, R needs only raise g^k to the power s (to get y^k). Then the multiplicative inverse, y^{-k}, can be used to decrypt the cryptogram as

$$My^k \times y^{-k} = M \bmod p.$$

Desmedt and Frankel [4] proposed a threshold cryptosystem which uses the ElGamal system. In their construction the dealer uses the Shamir (t, n) threshold scheme to distribute the secret s among the shareholders, such that for any subset B of t participants $s = \sum_{i \in B} s_i \bmod \phi(p)$, where $\phi(p)$ is the Euler totient function of p and s_i is the share of participant P_i. Now, to decrypt the cryptogram $C = (g^k, My^k)$, each $P_i \in B$ calculates $(g^k)^{s_i} \bmod p$ and sends the result to the combiner. The combiner computes $y^k = g^{ks} = \prod_{i \in B} g^{ks_i}$ (knowing that $s = \sum_{i \in B} s_i \bmod \phi(p)$) and decrypts the message (using the multiplicative inverse y^{-k}).

3 Top-down cryptography in hierarchical groups

In (t, n) threshold cryptography, any participant contributes equally to the cryptographic process. This means that threshold cryptography is well suited for democratic organizations with a "flat" structure. Unfortunately, most organizations exhibit a hierarchical structure. In hierarchical groups cryptography can be used to control the flow of information between two consecutive levels. If cryptography is used to protect information flowing from top to bottom, we talk about "top-down" cryptography. If the flow of information is from bottom to top, we deal with "bottom-up" cryptography. In this paper we consider top-down cryptography.

In top-down cryptography participants on higher levels of the hierarchy are required to perform cryptographic operations before the lower level participants. Lower level participants can only proceed with the cryptographic processing after receiving a "go ahead" from the higher levels. Let us consider the case when a cryptogram is broadcast to all participants of a hierarchical organization. A top-down cryptosystem should allow the participant on the top of the hierarchy to retrieve the plain-text message from the cryptogram. Once the higher level participants have decrypted the cryptogram, they may allow the lower level participants to decrypt the cryptogram by sending a suitable permission – a *go ahead ticket.*

3.1 The model

We assume that the hierarchy of participants is described by a tree structure, where $\mathcal{P}(0)$ is the set of the highest level participants. Every participant $P_j \in \mathcal{P}(0)$ is the head of the group $\mathcal{P}(1;j)$. In general each participant of the group $\mathcal{P}(i;j_1,\ldots,j_i)$ is the head of the group $\mathcal{P}(i+1;j_1,\ldots,j_i,j_{i+1})$. Any participant can be a member and the head of a single group. It is easy to see that each path in the hierarchy is uniquely identified by a suitable sequence of heads (supervisors). We also assume that all participants of the group defined on the i-th level have the same power, i.e. we are dealing with threshold access structures.

The following notation will be used. The set of all participants is $\mathcal{P}$. The set $\mathcal{E}$ is the collection of all cryptographic keys (either for encryption or decryption). The set of cryptograms is $\mathcal{C}$, and the set of plain-text messages is $\mathcal{M}$. $\mathcal{T}$ denotes the set of all go ahead tickets. The share set is $\mathcal{S}$.

Definition 3.1 *A top-down cryptographic system consists of a single trusted dealer and a collection of combiners. Each combiner on the i-th level acts on behalf of a specific group.*

- *The trusted dealer (D) sets up the system. D chooses both the encryption and decryption keys. The encryption key $e \in \mathcal{E}$ is public. The decryption key $d \in \mathcal{E}$ is secret and is used to calculate shares. Next D determines shares for all participants (of all groups of the hierarchy) according to the following function:*

  $$f_D^{(i)} : \mathcal{E} \times \mathcal{P}(i;j_1,\ldots,j_i) \to \mathcal{S} \times \mathcal{T}, \quad (i = 0,1,\cdots,\ell-1)$$

  for the groups on the i-th levels, and

  $$f_D^{(\ell)} : \mathcal{E} \times \mathcal{P}(i;j_1,\ldots,j_\ell) \to \mathcal{S}$$

  for the lowest level. All secret sharing schemes used for the groups are designed for threshold access structures. The value of the threshold may be different for each group. Note that the dealer sets up all static secret elements which do not change throughout the life time of the system.

- *The combiner of the top level group (on 0-th level) collects partial results from participants of the group $\mathcal{P}(0)$ and applies the function*

  $$f_C^{(0)} : f_0(\mathcal{C},\mathcal{S}_1) \times \cdots \times f_0(\mathcal{C},\mathcal{S}_t) \to M \times f_{tic}^0(\mathcal{C},\mathcal{T})$$

  where $f_{tic}^0(\bullet,\bullet)$ is a function which assigns a go ahead ticket for a cryptogram and a static ticket. If the number of participants who have provided their shares to combiner equals or exceeds the threshold, $f_C^{(0)}$ generates the plain message and a valid go ahead ticket which may be given to the lower level groups.

  A combiner on the i-th level collects partial results of its group $\mathcal{P}(i;j_1,\ldots,j_i)$, a go ahead ticket from the higher level and uses the function

  $$f_C^{(i)} : f_i(\mathcal{C},\mathcal{S}_1) \times \cdots \times f_i(\mathcal{C},\mathcal{S}_t) \times f_{tic}^{i-1}(\mathcal{C},\mathcal{T}) \to M \times f_{tic}^{i+1}(\mathcal{C},\mathcal{T}) \quad (i = 1,2,\cdots,\ell-1)$$

to retrieve the message and compute the ticket for the lower level groups. The message and the ticket are valid only when the number of contributing participants equals or exceeds the threshold of the group. The combiner on the lowest level, ℓ, retrieves the message (using partial results of the participants and the go ahead ticket) as:

$$f_C^{(\ell)} : f_\ell(\mathcal{C}, \mathcal{S}_1) \times \cdots \times f_\ell(\mathcal{C}, \mathcal{S}_t) \times f_{tic}^{\ell-1}(\mathcal{C}, \mathcal{T}) \to M.$$

The structure of a top-down cryptographic system reflects the hierarchy of the whole group. To avoid unnecessary details and to simplify our presentation, we will examine the simplest case of hierarchical cryptosystems where there is only one participant on every level of the hierarchy. Consequently, there is only one path in the hierarchy and the go ahead decision is taken by a single participant who acts as the combiner.

A generalization for the case with full hierarchical structure can be obtained by multiple application of single path hierarchy (with single participants or groups of participants on each level as appropriate). In this case there are many combiners on each level (except the first one). They collect partial results and go ahead tickets, decrypt the cryptogram and compute the go ahead tickets for the lower groups.

Go-ahead tickets may be computed by their immediate predecessors – groups one level above the current ones. More specifically, for the group $\mathcal{P}(i; j_1, \ldots, j_i)$, only the group $\mathcal{P}(i-1; j_1, \ldots, j_{i-1})$ can produce the valid ticket. This system is called a *hierarchical cryptosystems with sequential go ahead.*

The alternative to the above are hierarchical systems in which any higher level group belonging to a single path can authorize the lower level groups by passing go ahead tickets to them. So the group $\mathcal{P}(i; j_1, \ldots, j_i)$ can obtain a valid ticket from a one of the following groups: $\mathcal{P}(i-1; j_1, \ldots, j_{i-1})$, ..., $\mathcal{P}(1; j_1)$, and $\mathcal{P}(0)$. These systems are called *hierarchical cryptosystems with jumping go ahead.*

Next we will discuss two implementations of hierarchical cryptosystems. The first is based on the ElGamal cryptosystem. The second uses the RSA system.

4 ElGamal hierarchical cryptosystem

Assume that we have a group of participants $\mathcal{P} = \{P_0, P_1, \ldots, P_\ell\}$. P_0 is on the top of the hierarchy. P_1 belongs to the second level, P_2 to the third, etc. P_ℓ is at the bottom of the hierarchy. Every level is occupied by a single participant.

Algorithm for the dealer.

Let p be a large prime and g be a primitive element of $GF(p)$. The dealer selects $\ell + 1$ random integers $s_0, s_1, \cdots, s_\ell$, where $s_i \in GF(p)$ $(i = 0, 1, \cdots, \ell)$. Next the dealer computes $s = \sum_{i=0}^{\ell} s_i \bmod \phi(p)$, $y = g^s \bmod p$ and publishes p, g and y as the public parameters of the system. The secrets, (degenerate shares) $s_j, s_{j+1}, \cdots, s_\ell$, are distributed via a secure channel to the participant P_j $(j = 0, \cdots, \ell)$.

Algorithm for the sender.

The sender follows the steps of the ElGamal system. First he selects a random integer $k \in GF(p)$. Next he computes the cryptogram $C = (C^{(1)}, C^{(2)}) = (g^k, My^k)$ for the message $M \in \mathcal{M} = GF(p)$. Finally the sender broadcasts the cryptogram C.

Algorithm for the combiner

All participants P_i can be considered as degenerate combiners ($i = 0, \ldots, \ell$). The combiner P_0 knows the parameter $s = \sum_{i=0}^{\ell} s_i$ so he can compute $(g^k)^s$ and retrieve the message M. After reading the message, P_0 may send a go ahead ticket $(g^k)^{s_0} \pmod p$ to participant P_1 (enabling P_1 to retrieve the message). In general the participant P_i ($i = 1, \ldots, \ell$) knows $s_i, s_{i+1}, \cdots, s_\ell$ and after receiving the ticket $(g^k)^{s_0+\ldots+s_i-1}$ from P_{i-1}, can compute $(g^k)^s$ and decrypt the cryptogram. P_i decides whether to send the go ahead ticket $g^{s_0+\ldots+s_i}$ to P_{i+1}.

Note that this system is with jumping go ahead. After receiving a valid go ahead ticket, any higher level participant can issue a correct go ahead ticket to any lower level participant.

4.1 Variant of the hierarchical ElGamal cryptosystem

Assume that the dealer distributes only two secret elements to each participant. So P_0 knows the pair $(s_0, S_0 = \sum_{j=0}^{\ell} s_j)$. In general each $P_i (i = 1, 2, \cdots, \ell)$ knows the pair $(s_i, S_i = \sum_{j=i}^{\ell} s_j)$. Note that s_i is used to create a valid go ahead ticket for one level below and S_i along with the ticket from one level above allows P_i to decrypt cryptogram.

After receiving a valid ticket from the higher level, each participant can decrypt the cryptogram. The participant P_j ($j = 1, \ldots \ell$) who has received a correct ticket from the higher level, may issue a valid ticket (using the received ticket and s_j) for the $(j+1)$-th level. Hence, in this variant, each level can only issue a go ahead ticket to its immediate lower level. The obvious way to skip to lower levels, is by an interaction of two levels. Suppose that the j-th level wants to pass a ticket to level i ($j < i$, $j \neq i-1$). First the participant on the level i passes $(g^k)^{S_i}$ to level j. Now the j-th level can compute the ticket:

$$\frac{(g^k)^{S_0}}{(g^k)^{S_i}}$$

and can pass it to the level i.

Hierarchical ElGamal cryptosystems (both with jumping and sequential go ahead) have a property which allows higher level participants to modify their tickets so that lower level participants will retrieve false messages. This property is discussed in the following theorem.

Theorem 4.1 *In hierarchical ElGamal cryptosystems, the higher levels can modify their go ahead tickets in such a way that the lower levels retrieve a false message M_f from the cryptogram C.*

Proof: Assume that P_i wants P_j to accept $M_f \neq M$. Although P_i cannot change the cryptogram $C = My^k$, they can compute the following (invalid) ticket

$$\alpha = \frac{C}{M_f (g^k)^{S_j}}$$

provided P_i knows $(g^k)^{S_j}$, where $S_j = s_j + \cdots + s_\ell$. It is easy to see that P_j retrieves M_f using the ticket α. □

Note that if P_j retrieves a false message M_f (instead of the original M), all lower levels will also retrieve M_f (for valid tickets). These schemes can be useful for hierarchical groups where lower levels always expect to be able to read the cryptogram. Higer levels can always choose whether to send the original or false message.

4.2 Security

Security of the system for an outsider is equivalent to the security of the original ElGamal cryptosystem, that is, equivalent to the security offered by the discrete logarithm problem. From an outsider point of view the value of p and the public key y are independent from the internal structure of the group. For an insider (member) of the group, obtaining the secret from a partial decryption g^{ks_j}, $0 \leq j \leq \ell$, is equivalent to solving the corresponding instance of the discrete logarithm.

Although (t, n) threshold schemes are vulnerable to conspiracy attack, the proposed scheme resists against this attack as long as the participant on the highest level does not reveal their secret elements.

5 RSA hierarchical cryptosystem

Assume as before that we have a group of participants $\mathcal{P} = \{P_0, P_1, \ldots, P_\ell\}$. P_0 is on the top of the hierarchy. P_1 creates the second level, P_2 the third, etc. P_ℓ is at the bottom of the hierarchy. Every level is occupied by a single participant.

Algorithm for the dealer.

Let $N = pq$ where p, q are two safe primes, that is $p = 2p' + 1$ and $q = 2q' + 1$ (p' and q' are large and distinct primes). The dealer (D) who sets up the system, selects p, q and a random integer e such that e and $\lambda(N)$ are co-prime ($\lambda(N)$ is the least common multiple of $(p-1)$ and $(q-1)$, and therefore, is equal to $2p'q'$). Next D publishes e and N as public parameters of the system while the values of p, q, and d (d has the property that $ed \equiv 1 \pmod{\lambda(N)}$) are kept secret. Also the dealer selects ℓ random integers $s_1, s_2, \cdots, s_\ell$ and solves the equation $s_0 + s_1 + \cdots + s_\ell = d \bmod \lambda(N)$ for unknown integer s_0 (note that in this scheme $\sum_{i=0}^{\ell} s_i$ is equal to d, instead of $d-1$). Finally D distributes the secret elements (degenerated shares) via secure channels to all participants. P_0 gets the sequence $(s_0, \ldots, s_\ell)$. The participant P_i receives $(s_i, \ldots, s_\ell)$ where $i = 1, \ldots, \ell$.

Algorithm for the sender.

As in the original RSA system, the sender computes cryptogram C as $M^e \bmod N$. The cryptogram is broadcast to all participants of the hierarchy.

Algorithm for the combiner.

Participants can be considered as degenerate combiners. Combiner P_0 knows that $s_0 + s_1 + \cdots + s_\ell = d \bmod \lambda(N)$ and can decrypt the cryptogram C as $C^d = (M^e)^d = M \bmod N$. After reading the message, he can control P_i's ability to read the message by sending a correct ticket $C^{s_0+\ldots+s_i-1}$. After receiving the ticket $C^{s_0+\ldots+s_i-1}$, the combiner P_i can retrieve the message as he also knows the sequence $(s_i, \ldots, s_\ell)$ therefore

$$C^{s_0+\ldots+s_i-1} \times C^{(s_i+\ldots+s_\ell)} = C^d = M \bmod N.$$

5.1 Variant of the hierarchical RSA cryptosystem

Assume that the dealer distributes only two secret elements to each participant. So P_0 knows the pair $(s_0, S_0 = \sum_{j=0}^{\ell} s_j)$ and in general each $P_i (i = 1, 2, \cdots, \ell)$ knows the pair $(s_i, S_i = \sum_{j=i}^{\ell} s_j)$. Note that in case of the ElGamal hierarchical cryptosystem, the dealer may compute the sum modulo $\phi(p)$, but in this case the addition must be done in Z_N (otherwise, some levels may collaborate to find the $\lambda(N)$, however, due to exponentiation the decryption is actually performed in $Z_{\lambda(N)}$). The number s_i is used to create a valid go ahead ticket for one level below and S_i along with the go ahead ticket from one level above allows P_i to decrypt the cryptogram.

After receiving a valid ticket from the higher level, each participant can decrypt the cryptogram. A participant P_j, who has received a correct ticket from the higher level, can issue a valid ticket (using the received ticket and s_j) for the $(j+1)$-th level. Hence, in this variant, each level can only issue a go ahead ticket for the levels immediately below it. The obvious way to skip lower levels, is by the collaboration of two levels. Suppose the j-th level wants to pass a ticket to the i-th level where $j < i$ and $j \neq i - 1$. First the level i passes $(C)^{S_i}$ up to the level j. Now the j-th level can compute the ticket (assuming that $(C)^{S_i}$ and N are coprime):

$$\frac{M}{(C)^{S_i}} \bmod N$$

and passes it to the level i. Again, this type of skipping requires interaction between the levels.

The RSA implementation of hierarchical cryptosystems allows every participant in the hierarchy to verify if the retrieved message is genuine. This is explained by the following theorem.

Theorem 5.1 *In hierarchical RSA cryptosystems, the recipients of tickets can always distinguish valid tickets from corrupt ones.*

Proof: Assume that P_i wants P_j to accept $M_f \neq M$. Although P_i cannot change the cryptogram $C = M^e$ (it has been broadcast to every participants), they can compute a corrupt ticket

$$\alpha = \frac{M_f}{(C)^{S_i}} = M_f \times C^{-S_i} \bmod N$$

provided P_i knows C^{S_i}, where $S_i = s_i + \cdots + s_\ell$. Having got the ticket, P_i retrieves the message $M_f = \alpha \times C^{S_i}$. Now P_i can verify the validity of both retrieved message M_f and the ticket α by computing ${M_f}^e \bmod N$ which is not equal to the cryptogram C (as long as $M_f \neq M$ and $M_f < N$). □

Hierarchical RSA cryptosystems can be used where all participants want to verify the validity of retrieved messages and received tickets.

5.2 Security

The security of the system for an outsider who knows only the public values of e and N is equivalent to the security of the original RSA cryptosystem, and therefore, it is equivalent

to the factorization problem.

For an insider (member of the hierarchy), finding s_j, from the partial decryption value C^{s_j}, is equivalent to the solution of the discrete logarithm problem in the ring Z_N. Security of this system is equivalent to the threshold signature scheme of [3]. (In the RSA cryptosystem signing a message is equivalent to decryption.)

Although (t, n) threshold schemes are vulnerable to conspiracy attack, the proposed scheme resists against this attack as long as the participant on the highest level does not reveal their secret elements.

6 Verifiability in ElGamal threshold cryptosystem

Consider some repercussions of Theorem 4.1 in the context of security of ElGamal threshold cryptosystems. The ElGamal threshold cryptosystem was proposed by Desmedt and Frankel [4]. They request the shares and modified shares (shadows) to be secret. Partial results (generated from modified shares) are assumed be be protected by exponentiation - an attacker faces an intractable instance of the discrete logarithm problem). The partial results are communicated to a trusted combiner who retrieves the message from a public cryptogram. Note that in the light of Theorem 4.1, the retrieved meaningful message cannot be verified – it has to be accepted as genuine.

This becomes crucial if the message source has no redundancy. A dishonest participant can send a random (invalid) partial result and "force" the combiner to generate a random (but meaningful) false message. There is also another possible attack on the system if one of the participants is able to see other participants partial results. For an arbitrary false message M_f, this participant can modify his or her partial result in such a way that the combiner retrieves the false message M_f. This is the case when one of participants agrees to be the combiner.

Although this attack does not break the system, it compromises the goal of threshold cryptography. Note that the lack of verifiability of retrieved messages in ElGamal threshold cryptosystems is a weakness never indicated in the open literature before. On the other hand, in hierarchical cryptosystems this feature is a useful property which enables higher level participants to keep lover level participants informed and in the "dark" at the same time.

In the final version of this paper, we will generalize the design of cryptosystems for arbitrary hierarchy and show how this problem can be fixed.

7 Conclusions

We have proposed two cryptosystems as solutions to open problems related to hierarchical groups. The dealer publishes a public key such that an individual can use that public key to send an encrypted message to the whole (hierarchical) group. Publication of both the group public key and the encryption method does not reveal the decision of the group. We have assumed that higher level groups (or participants) can allow lower level groups (or participants) to retrieve the message by passing go ahead tickets. We have proposed two solutions: one based on the ElGamal system and the other based on the RSA system. The

cryptosystems are efficient, practical and non-interactive. The security of the proposed schemes is equivalent to the security of underlying cryptosystems.

Note that in both solutions there are two classes of secret elements (or shares): static (used for decryption), and dynamic (used for ticket generation). Static shares are more or less used in traditional way to compute partial results which later are sent to the combiner who can compute the final result for an authorized subset of participants. On the other hand dynamic shares are used to control the information flow from one level to another.

There is a crucial difference between threshold and hierarchical cryptosystems. In threshold cryptosystems the dealer distributes all shares in one go and does not allow participants to interfere in the distribution process. On the contrary, the dealer in hierarchical cryptosystems sets up the skeleton of the system by the distribution of static shares and deposits dynamic shares with higher level participants so they can make proper decisions about the distribution of dynamic shares later.

In the final version of the paper, we will generalize the concept of dynamic shares and we will show how this generalization can be used to design a cryptosystem for an arbitrary hierarchy.

This paper has also revealed a weakness in existing ElGamal based threshold cryptosystems.

References

[1] G. R. Blakley. Safeguarding cryptographic keys. Proceedings of AFIPS 1979 National Computer Conference, 1979.

[2] Y. Desmedt. Society and group oriented cryptography: a new concept. In *Advances in Cryptology, Proceedings of CRYPTO'87, Ed C. Pomerance, Lecture Notes in Computer Science, Vol.293*, pages 120–127. Springer-Verlag, 1988.

[3] Y. Desmedt. Threshold cryptosystems. In *Avances in Cryptology - AUSCRYPT'92, Lecture Notes in Computer Science, Vol. 718, J.Seberry, Y.Zheng (Eds.)*, pages 3–14. Springer-Verlag, 1993.

[4] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *Advances in Cryptology, Proceedings of CRYPTO'89, Ed G. Brassard, Lecture Notes in Computer Science, Vol.435*, pages 307–315. Springer-Verlag, 1990.

[5] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22(6):644–654, November 1976.

[6] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31:469–472, 1985.

[7] A.M. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. *Proceedings of EUROCRYPT'84, Lecture Notes in Computer Science, Advances in Cryptology, T. Beth, N. Cot, I. Ingemarsson (Eds)*, 209:224–314, 1985.

[8] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[9] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.