SECRYPT 2009, pages 266–269, 2009

INTERACTIVE SECRET SHARE MANAGEMENT

Constantin Catalin Dragan
Department of Computing, "Al.I.Cuza" University, Iasi, Romania
dragancc@yahoo.com

Keywords: Secret-sharing scheme, management, compartment schemes

Abstract: In this paper, we have proposed a method for the management of a compartmented secret sharing scheme that allows the increase of the global threshold without modifying the existent shares of the participants. We have considered the Trusted Authority the central point of the scheme as a management unit: it creates the shares, in a RSA manner, and distributes them, rebuilds the secret S, and allows the registration of new participants without modifying the existing shares.

1 INTRODUCTION

A secret sharing scheme (Menezes et al., 1998; Iftene, 2007) starts with a secret and then derives from it certain shares which are distributed to participants. The secret may be recovered only by certain predetermined sets which belong to the access structure (Ito et al., 1987). Secret sharing schemes have been independently introduced by Blakley (Blakley et al., 1993) and Shamir (Shamir, 1979) as a solution for safeguarding cryptographic keys. In the first secret sharing schemes only the number of the participants in the reconstruction phase was important for recovering the secret(threshold secret sharing schemes (Blakley et al., 1993; Shamir, 1979)). A scheme that deals with more complex access structures is compartmented secret sharing schemes (Simmons, 1990; Ghodosi et al., 1998), in which the set of participants is partitioned into compartments and the secret can be recovered if and only if the number of participants from any compartment is greater than or equal to a compartment threshold and the total number of participants is greater than or equal to a global threshold. In this paper, we propose a method for the management of compartmented secret sharing schemes that allows the increase of the global threshold without modifying the existent shares of the participants or the compartments thresholds. Moreover, the method is compatible with non-disjoint compartments.

2 INTERACTIVE SECRET SHARE MANAGEMENT

The scheme we are suggesting in the following paragraphs deals with the subsecret shared by an arbitrary secret sharing scheme. If $(a_1, ..., a_k)$ are all the shares that need to be used for the rebuild of the secret S, then we are interested in the methods used for the distribution of these k subsecrets to m participants. Each participant has the possibility to rebuild one or more subsecrets. The trusted authority coordinates every aspect of the scheme. It creates the shares, and gives them to the participants, encrypted in a RSA manner (Shamir, 1978). Furthermore, it also coordinates the reconstruction phase: it gathers the participants, communicates with them to establish their identity and receive their share, and rebuilds the secret S. In the end, it checks the correctness of the secret S using a method of proof.

2.1 Scheme settings

Participants. In order to set up the scheme let us assume that there are m participants $A_1, ..., A_m$. Assume that the participants are divided into small groups (compartments) $G_1, ..., G_l$, and $G_i \cap G_j = \emptyset$ for any $i \neq j$. For any compartment G_i a non-empty subset of

positions $P_i \subseteq \{1,...,k\}$ is given.

Choosing the secret. The trusted authority (TA) will choose a k -coordinate secret S =$(a_1,...,a_k)$, where each coordinate a_i is a large number and $k \leq m$.

Sharing the secret. A participant $A \in G_i$ is allowed to recover any subsecret $a_j \in P_i$. In order to do that the TA computes the polynomial f_A

$$f_A(x) = \sum_{j \in P_i} \left(c_j a_j \prod_{r \neq j, r \in P_i} (x - r) \right), \quad (1)$$

where c_j is a coefficient given by

$$c_j = \frac{1}{\prod_{r \neq j, r \in P_i} (j - r)}. \quad (2)$$

The polynomial f_A can also be written as $f_A(x) = x^{t-1} b_{t-1} + ... + x b_1 + b_0$, where $t = |P_i|$. The coefficients of this polynomial are distributed in a "blind" way to participants. To do that, TA follows the next steps:

- chooses two large prime numbers p and q with $p \neq q$ and computes $n = pq$;
- chooses $e \in \mathbb{Z}^*_{\phi(n)}$;
- encrypts each coefficient b_j by e , $d_j = b_j^e \bmod n$, for all $0 \leq j < t$;
- distributes the vector $(d_{t-1},...,d_0)$ to A by a secure channel.

The TA has to keep in a safe place only k, p, q and e.

Secret recovery. In order to recover the secret S , k distinct participants are needed. Each of them should provide a share of the secret. Let us assume that the participants are $A_{i_1},...,A_{i_k}$ with $i_s \neq i_r$ for any $s \neq r$.

Assume that $r \in P_{i_r}$, for any $1 \leq r \leq k$. That is, we assume that A_{i_r} can deliver the share a_r. In fact, this share is delivered by means of the vector $(d_{t-1},....,d_0)$. This vector is then processed by TA as follows:

- compute $e^{-1} \bmod \phi(n)$;
- decrypt $b_j = d_j^{e^{-1}} \bmod n$;
- compute $f_{A_{i_r}}(r) = a_r$.

Correctness. The correctness of this scheme easily follows from the description above. The pair (e, e^{-1}) is a RSA asymmetric key and, in our scheme, no element of this pair is public.

Security of the scheme. First, we remark that no participant A can recover the coefficients of f_A because these coefficients are encrypted in a RSA manner.

If two or more participants try to put together their secret information, they cannot mount any attack on the polynomial coefficients $(d_{t-1},...,d_0)$ more dangerous than any known cryptotext attack against RSA (Simmons, 1983; DeLaurentis, 1984). Therefore, we may say that the security of this scheme relies on the security of RSA (Simmons, 1983; DeLaurentis, 1984).

When secret recovery is needed, TA should make sure that the participants are distinct and each participant is able to deliver a share according to P_i. For this, each participant A should have a certificate

$$c(A) = (ID(A), P, Info, sig_{TA}(ID(A), P, Info)) \quad (3)$$

consisting of A's identity ID(A), A's compartment P, information, and TA's signature on these $sig_{TA}(ID(A), P, Info)$.

The TA checks the certificate and if it is valid, then it chooses $j \in P$ and computes $f_A(j)$.

Correctness proof for recovery. TA should not keep the secret S =$(a_1,....,a_k)$, but in the same time he should be able to make sure that at every recovery he obtain the same secret S . This can be done by using some methods of proof such as:

$$Proof = (\alpha_1^{a_1} + ... + \alpha_k^{a_k}) \bmod n; \quad (4)$$

$$Proof = hash(a_1,...,a_k); \quad (5)$$

where α_i could be a distinct prime number or a primitive root, for any $1 \leq i \leq k$.

At every recovery, TA computes the newProof using the secret he just obtain, and compares it to the Proof located in a safe place on TA. If they are equal then he obtained the correct secret.

2.2 Secret Share Management

In compartment schemes, the set of participants is partitioned into compartments (groups): $G_1,...,G_l$. Beside a global threshold k, a threshold k_i is assigned to the i^{th} group, for all $1 \leq i \leq l$. Most of the secret sharing schemes (Simmons, 1990; Ghodosi et al., 1998) that use compartments consider the more general case of disjoint compartments .

For an arbitrary scheme we will point out the differences between a participant that belongs to only one group and one that can be in more than one group. Moreover, a method of modifying a classic scheme with the intent of raising the schemes threshold without modifying the existent shares of the participants is presented.

2.2.1 A Participant can only be in one Group.

In this case, all participants in the same group G_i share the same subsecrets a_j with $j \in P_i$. The difference

between participants of the same group is made by certificates .

The registration of a new participant in a group G_i requires only the presence of one older participant. The older participant sends its share to TA, that validates it. Then TA gives it to the new participant along with a new certificate.

2.2.2 A Participant can be in more than one Group.

A participant A that is member of the groups $G_1,...,G_\alpha$ has access to all the subsecrets in each group therefore,

$$f_A(x) = \sum_{j \in P} \left(c_j a_j \prod_{r \neq j, r \in P} (x - r) \right), \qquad (6)$$

where $P = P_1 \cup ... \cup P_\alpha$ and c_j is a coefficient given by

$$c_j = \frac{1}{\prod_{r \neq j, r \in P} (j - r)}. \qquad (7)$$

Participants that are members of the same groups are distinguished by their certificate.

The registration of a new participant to groups $G_1,...,G_\alpha$ requires the presence of at least one participant from each group, in order to be able to form the new share. He would also receive a new certificate from TA.

The recovery of the secret must start after the registration of all participants. Because one participant A can be in multiple groups $G_1,...,G_\alpha$ he could be used to return the subsecret to any of his groups. That is why TA has to determine where the participant is needed first. TA should receive a set of participants and, before starting to compute the secret, he has to make sure of the following things:

- the number of participants is greater than or equal to the sum of the groups thresholds;
- and the number of participants that can occupy one group is greater than or equal to its threshold.

If at least one of the two conditions are invalid, TA can not rebuild the secret S , because no matter how he distributes the participants into groups, there will always be groups with less participants than their threshold.

Otherwise, TA should distribute the participants to each group. He could use backtracking and for a group G_i he should first put the participants that belong to only that group then the participants corresponding to multiple groups.

If the number of participants in each group is greater than or equal to the groups threshold the algorithm will return "Succes", otherwise "Fail".

Example Let us consider m= 14, k= 6, l= 3, $k_1 = 3$, $k_2 = 2$, and $k_3 = 1$ and the set of participants $A = \{A_1,...,A_6\}$, where $A_1 \in G_1 \cap G_2$, $A_2 \in G_2 \cap G_3$, $A_3 \in G_1$, $A_4 \in G_1$, $A_5 \in G_1$, $A_6 \in G_1$.

The evaluation of the conditions:

- the conditions are true
  - for G_1 we have $\{A_1,A_2,A_4,A_5,A_6\}$ and $5 \geq 2$;
  - for G_2 we have $\{A_1,A_2\}$ and $2 = 2$;
  - for G_3 we have $\{A_2\}$ and $1 = 1$.

The algorithm returns:

- the participants distributed in groups
  - for G_1 we have $\{A_3,A_4,A_5\}$;
  - for G_2 we have $\{A_1,A_2\}$;
  - for G_3 we have ?;
  - A_6 not used.
- "Fail".

2.2.3 The Scheme's Threshold is Greater than the Sum of the Groups Thresholds.

The previous paragraphs have discussed the most well known case of compartment schemes, where the scheme threshold equals the sum of the groups threshold $k = \sum_{i=1}^{l} k_i$.

For $k > \sum_{i=1}^{l} k_i$ it is not enough the information the participants offer, because TA still needs α subsecrets to form the secret S , where $\alpha = k - \sum_{i=1}^{l} k_i$. There is no major difference between the methods $G_i \cap G_j = \emptyset$ and $G_i \cap G_j \neq \emptyset$. We will present the same solution to both of them.

The simplest way is for TA to just keep stored in a safe place $(a_{k-\alpha+1},...,a_k)$, but that is not fair, and contradicts the security of TA. So we propose a set of functions - stored on TA, that receives as input some subsecrets from $(a_1,...,a_{k-\alpha})$ and returns $(a_{k-\alpha+1},...,a_k)$. That way TA should have access to $(a_{k-\alpha+1},...,a_k)$ only in the moment of reconstruction (after the entry of the participants).

2.2.4 The Number of Needed Subsecrets is less than or equal to the Number of Groups ($\alpha \leq l$).

Each group G_i has assign a function that receives as input the subset of subsecrets its participants generated $(a_1^{(i)},...,a_{k_i}^{(i)})$, and gives as output one of the needed subsecrets $a_{k-\alpha+i}$. One example of such a function could be $f_{G_i}(x_1,...,x_{k_i}) = a_{k-\alpha+i} - (a_1^{(i)} + ... + a_{k_i}^{(i)}) + (x_1 + ... + x_{k_i})$. In the end the groups would have obtain $(a_{k-\alpha+1},...,a_{k-\alpha+l})$. To simplify, only the first α groups would return the subsecrets needed and the rest would return 0.

2.2.5 The Number of Needed Subsecrets is Greater than the Number of Groups ($\alpha > l$).

We will assume that $\alpha \leq 2l$. Similar to the previous case ($\alpha \leq l$) we attribute a function f_{G_i} to each group G_i, $1 \leq i \leq l$. The rest of the subsecrets needed $(a_{k-\alpha+l+1}, ..., a_k)$ could be obtain from TA, through the use of a function F similar to the one used by the participants.

$$F(x) = \sum_{j=1}^{\alpha-l} \left(\prod_{i=1, i\neq j}^{l} \frac{(x - a_{k-\alpha+i})}{(a_{k-\alpha+j} - a_{k-\alpha+i})} a_{k-\alpha+l+j} \right)$$

The function F receives as input $a_{k-\alpha+i}$ and gives as output $a_{k-\alpha+l+i}$, where $1 \leq i \leq (\alpha - l)$.

2.3 Hardware Implementation.

We can implement our scheme by card-reader tamper-resistant devices and smart-cards.

The card-reader, which is a device associated to TA, should be capable of performing computations, because it should recover the secret S. The device has a rewritable permanent memory that stores k, p, q, e, a list of certificates $\{c(A_1), ..., c(A_m)\}$, and a list of groups with their corresponding threshold $\{(G_1, P_1, k_1), ..., (G_l, P_l, k_l)\}$.

The card-reader should be capable to do:

- register the participant. The card reader checks the participant's certificate against the list of allowed participants and does the following:
  - if the certificate is invalid the card reader rejects the participant;
  - if the participant is already registered in the current reconstruction of the secret, the card reader rejects the participant;
  - otherwise, it registers the participant.
- for each group G_i it computes its secret $g_i = (a_{i_1}, ..., a_{i_{k_i}})$, where $1 \leq i \leq l$;
- after obtaining all the subsets g_i, it (TA) uses concatenation on them to form one set, the secret $S = g_1||...||g_l$, where $||$ is the operation of concatenation.

After obtaining the secret it first checks the correctness by using proof. If the result is correct it discards any information related to the secret rebuilding.

The smart card is attributed to the participant and has stored on it all the relevant information (the shares $d_A(x)$ and the certificate $c(A)$). It is able to communicate with the card-reader by carrying out a process of identification. He sends the information, when needed, to the card reader and awaits the result, the secret S.

3 CONCLUSIONS

In this paper, we have proposed a method for the management of a compartmented secret sharing scheme that allows the increase of the global threshold without modifying the existent shares of the participants. We have considered the Trusted Authority the central point of the scheme as a management unit: it creates the shares, in a RSA manner, and distributes them, rebuilds the secret S, and allows the registration of new participants without modifying the existing shares.

REFERENCES

Blakley, B., Blakley, G. R., Chan, A. H., and Massey, J. L. (1993). Threshold schemes with disenrollment. In *Advances in Cryptology - CRYPTO 92*, volume 740 of Lecture Notes in Computer Science, pages 540–548. Springer-Verlag.

DeLaurentis, J. M. (1984). A further weakness in the common modulus protocol for the rsa cryptoalgorithm. *Cryptologia*, 8(3).

Ghodosi, H., Pieprzyk, J., and Safavi-Naini, R. (1998). Secret sharing in multilevel and compartmented groups. *Lecture Notes in Computer Science*, 1438:367–378.

Iftene, S. (2007). *Secret Sharing Schemes with Applications in Security Protocols*. PhD thesis, "Al.I.Cuza" University of Iasi, Iasi, Romania.

Ito, M., Saito, A., and Nishizeki, T. (1987). Secret sharing scheme realizing general access structure. In *IEEE Global Telecommunications Conference: Globecom 87*, pages 99–102. IEEE Press.

Menezes, A. J., van Oorschot, P. C., and Vanstone, S. A. (1998). *Handbook of Applied Cryptography*. CRC Press, volume 6 of discrete mathematics and its applications edition.

Shamir, A. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126.

Shamir, A. (1979). How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613.

Simmons, G. J. (1983). A 'weak' privacy protocol using the rsa cryptoalgorithm. *Cryptologia*, 7(2).

Simmons, G. J. (1990). How to (really) share a secret. In *Advances in Cryptology - CRYPTO 88*, volume 403 of Lecture Notes in Computer Science, pages 390–448. Springer-Verlag.