Paper

Simple Dynamic Threshold Decryption Based on CRT and RSA

Bartosz Nakielski and Jacek Pomykała

Abstract—In the paper we present a simple threshold decryption system based on the RSA cryptosystem. Our model avoids the application of the Shamir secret sharing protocol and is based only on the Chinese reminder theorem. The flexibility in the threshold level is attained due to the suitable preparation of the input data. The second part of the article describes a modification of the basic model, which admits the sender's impact on the choice of the real receiver's group.

Keywords*— *CRT, RSA, threshold decryption.

1. Introduction

Threshold cryptography is one of the most important directions in modern cryptology. The basic idea is the division of the private key (used to decrypt or sign the messages) into shares, such that at least the given number of them (called the threshold level) is necessary to its reconstruction. This allows to distribute the trust or responsiblity over the group members who are involved in the decryption or signing process, respectively. On the other hand, the distributed data or services allow to increase their availability, reliability or security.

The potential draw-back of the threshold cryptosystems (particularly in the encryption systems) is the lack of flexibility in the threshold level aspect. However in many applications the messages have different "priorities" or importance. In this connection we should require that some data should have higher threshold level than the others. In the classical approach this implies the necessity of generation of several polynomial threshold sharing protocols each responsible for the different threshold level. Recently there were some attempts to partially solve this problem (see, e.g., [1], [2], [3]) in the digital signature context.

As concerns the threshold decryption systems, very interesting solution was presented by H. Ghodosi, J. Pieprzyk, R. Safavi-Naini [4]. They apply the RSA (Rivest, Shamir, Adleman) cryptosystem [5] together with the Chinese reminder theorem (CRT) and the Shamir secret sharing protocol [6] to obtain the flexibility of the threshold level in dynamic group decryption process.

The Shamir protocol allowed the sender to share the "session key" among the members of the corresponding decryption group. In [7] the application of the above model for the databases systems was presented.

In this paper we were able to avoid the application of Shamir protocol completely, while still keeping the possibility to vary the threshold level together with the encrypted messages. In view of the additional "formatting" conditions concerning the encrypted data, our model is based only on the RSA cryptosystem and Chinese remainder theorem.

2. Mathematical Background

2.1. Chinese Reminder Theorem

Given the pairwise coprime positive integers $n_1, n_2, \ldots, n_k$ and any integers $a_1, a_2, \ldots, a_k$ one can compute the integer a satisfying the following conditions:
$a \equiv a_i \mod n_i$ (for $i = 1, 2, \ldots, k$).
It can be obtained explicitly from the formula below:

$$a = \left(\sum_{i=1}^{k} a_i z_i y_i \right) \operatorname{mod} n\,,$$

where:

$$n = \prod_{i=1}^{k} n_i\,,$$

$$z_j = \frac{n}{n_j} = \prod_{i=1}^{j-1} n_i \times \prod_{i=j+1}^{k} n_i\,,$$

$$y_j = z_j^{-1} \operatorname{mod} n_j\,.$$

2.2. RSA Cryptosystem

The RSA cryptosystem may be applied for the data encryption process as well as to the digital signatures. It's security is based on the factorization problem (for positive integers), which is believed to be computationally hard.

Parameters of RSA encryption scheme

Public key – (e, N) and private key d,

$N = p \cdot q$ (p, q are prime numbers),

$\varphi(N) = (p-1) \cdot (q-1)$,

e – a number coprime to $\varphi(N)$,

d – a number satisfyig the condition:
$e \cdot d \equiv 1 \bmod \varphi(N)$.

To encrypt the message m we compute the cryptogram $c = m^e \operatorname{mod} N$. To decrypt the ciphertext c we compute the value $m = c^d \operatorname{mod} N$.

More information concerning the RSA cryptosystems may be found in [5] and [8].

2.3. Dynamic threshold decryption

The idea of the threshold decryption cryptosystem is based on the splitting of the decryption key into several parts called the shares, which are applied for the reconstruction of the plaintext from the given ciphertext. The shares attached to the group members result in the fact that the decryption process has to be done collectively.
The typical threshold decryption systems use Lagrange interpolation formula to reconstruct the secret (being the free coefficient of the corresponding polynomial) from the shares being its values in positive integers. The degree of the polynomial defines the minimal number of shares needed to reconstruct the secret value. Thus the change of the threshold level causes the requirement of a new random polynomial to be generated and the corresponding shares to be distributed among the decryption group members. This makes the traditional approach to such systems completely impractical especially when the dynamic groups are considered. The solution proposed in [4] makes the sender responsible for the corresponding polynomial choice and pointing out the group of receivers of a given message, by means of some kind of "session keys". As a result the flexibility of the threshold level (depending on the message) is admissible. Moreover the impact of the sender on the choice of the "decryption" group is achieved. The more general dynamic decryption group model admitting the distribution of the same shares among the distinct members (c.f. [9], [10]) or hierarchical model (c.f. [11]) could be also considered within the similiar framework.
In this paper we present the simple threshold decryption protocol which avoids the application of the Shamir secret sharing protocol, still keeping the flexibility of the corresponding threshold level in the decryption process. It was possible due to the suitable preparation (representation) of the data, according to the assumed threshold level, before its encryption by the public keys of the decryption group members. In the decryption phase we use the corresponding private keys and the shares of the plaintext to reconstruct the original message.

3. Model

3.1. Notation

Let $G = \{P_1, ..., P_n\}$ be the group of users equipped with RSA keys $(d_i, (e_i, N_i))$, respectively.
Let us assume that the RSA moduli N_i are localized in the intervals:

$(*) \quad N_i \in (2^{i-1}N_0, 2^i N_0)$ for $i = 1, 2, ..., n$

and K be a fixed number (security parameter) satisfying the inequality:

$$(**) \quad 2\log_2\log_2 N_0 < K < \frac{\log_2 N_0}{10}.$$

The communication among the group G goes through the group message board (*GMB*), where all the decrypted fragments of the message are published. The access to the *GMB* requires RSA keys so only the members of G can read and write in *GMB*.
According to the application of CRT we denote:

$$N = \prod_{i=1}^{n} N_i,$$

$$N^j = \tfrac{N}{N_j} = \prod_{i=1}^{j-1} N_i \cdot \prod_{i=j+1}^{n} N_i,$$

$$Y_j = (N^j)^{-1} \bmod N.$$

3.2. Data Preparation

Let $\lfloor x \rfloor$ stand for the largest integer not exceeding x, while $\lceil x \rceil$ stand for the smallest integer not less then x. We define:

$$l_1 = l_1(t) = \lfloor \log_2 \prod_{i=n-t+2}^{n} N_i \rfloor,$$

$$l_2 = l_2(t) = \lfloor \log_2 \prod_{i=1}^{t} N_i \rfloor.$$

Assume for the moment that $l_1 + 4K < l_2$ (see Lemma 1). By the CRT any message M of the length contained in the interval $(l_1 + K, l_1 + 4K) \subset (l_1, l_2)$ is represented uniquely by the values of the residue classes: $M_{i_j} \bmod N_{i_j}$ for $j = 1, 2, ...t$.
If M has the length k greater or equal to $l_1 + 4K$ we can divide it on the suitable messages of length $l_1 + K$ and at most one message of length less than $l_1 + K$. For the message M of length $k < l_1 + K$ we shall apply the message padding procedure (desribed, e.g., in [8] in the framework of hash functions). Namely we require M to be represented by M_t according to the following steps:

1. Select a random number $l \in (l_1 + 3K, l_1 + 4K)$.
2. Add $l - k - \lceil \log_2 (l_1 + K) \rceil$ random bits to the message M (at the left side).
3. Add $\lceil \log_2 (l_1 + K) \rceil$ bits (at the right – hand side) denoting the length of the original message M (this part will contain a few zeros, since the length of M requires only $\lceil \log_2 k \rceil$ bits).

After the pading phase the message M_t is built of three parts:

Random bits	Message M	Bits denoting the length of M
$l - k - \lceil \log_2(l_1 + K) \rceil$	k	$\lceil \log_2(l_1 + K) \rceil$

Lemma 1. Let $n \geq 3$, $1 \leq t \leq n$ and $N_0 > \max(2^{n^2}, 2^{500})$. Assume that the RSA moduli satisfy the condition $(*)$ (see Subsection 3.1) and let K satisfying $(**)$ be fixed.
Then $l_1 + 4K < l_2$ and M_t contains at least K random bits.

Proof. By $(*)$ and $(**)$ we obtain that $l_2 - l_1 \geq \log_2 N_0 - (t-1)(n-t+1) = \log_2 N_0 - \frac{n^2-1}{4} \geq \log_2 N_0 - \frac{n^2}{4}$.
Therefore by the lower bound for N_0 and the upper bound for K we obtain that $l_1 + 4K < l_2$. Moreover, by the lower

bound $(**)$ for K the number of random bits in M_t is at least $l-k-\log_2(l_1+K) > l_1+3K-(l_1+K)-K > K$ as required. ∎

From the above we see that each fragment of M of length l_1+K has the K-bit margin against the possible decryption of the message by any group of at most $t-1$ members of G. On the other hand, the fragment of M of length $< l_1+K$ has, after the padding procedure, the corresponding margin of size K according to minimum K random bits in M_t. Finally, let us also remark that the length of M_t is chosen randomly in the interval of length K.

4. Encryption and Decryption Algorithms

4.1. Encryption

To send the encrypted message M to the group G the following steps are performed by the sender:

1. Select the threshold level t $(t \leq n)$.
2. Create the M_t from M according to the description in Subsection 3.2.
3. Using the public keys belonging to group members compute

   $c_i \equiv M_t^{e_i} \bmod N_i$ for $i = 1,2,...,n$.
4. With the aid of CRT compute C such that $C \equiv c_i \bmod N_i$ for $i = 1,2,...,n$.
5. Send the cryptogram (C,t) to the group G.

Remark 1. For the sake of complexity the values of N^i and Y_i (see Subsection 2.1) used in the Step 4 above should be precomputed and published for the users in advance.

4.2. Decryption

Decryption of the given ciphertext runs as follows:

1. Group members who decide to decrypt the message compute

   $m_i = C^{d_i} \bmod N_i$ and publish the triple $((C,t),m_i)$ on the *GMB*.
2. When t triples occur on *GMB* any member can combine the fragments and reconstruct the plaintext M_t (and then the original message M).

4.3. Modification

In the above model all the users have the same rights in the decryption process. However by the slight modification in the protocol the sender can have an impact on the choice of the members (say from some set $B \subset G$) participating in the decryption process. The modified protocol runs as follows:

1. Select the threshold level t $(t \leq |B|)$.
2. Create the M_t from M following the procedure described in Subsection 3.2.
3. Using the public keys belonging to group members compute

   $c_i \equiv M_t^{e_i} \bmod N_i$ for $\mathbf{i} \in \mathbf{B}$,
4. With the aid of CRT compute C such that $C \equiv c_i \bmod N_i$ for $\mathbf{i} \in \mathbf{B}$.
5. Send the cryptogram (C,t) to the group B.

If the message is declared only for users of B (and nobody else should read it) they should distribute the corresponding values among the group B (instead of publishing them in *GMB*).

5. Conclusion

In the paper we presented a threshold decryption protocol for the dynamic group based on the RSA cryptosystem and CRT, with the full flexibility of the threshold level. Our model avoids the application of the classical Shamir secret sharing protocol.
Instead we use some kind of formatting data technique which allows to replace Shamir protocol by the CRT method. The required ingredient was the application of the well known padding methodology in our context. The slight modification of the principal model admits the impact of the sender for the choice of the real receiver's group. The presented model can be extended for the more general framework of the threshold decryption systems (c.f. [10], [11]).

Appendix – an Example

In order to simplify the description we present the suitable example omitting the data preparation phase in the protocol.

1. Let $G = \{P_1, P_2, P_3\}$.
2. Values of the private and public keys belonging to the members of G:

   $e_1 = 17 \quad d_1 = 1289 \quad N_1 = 23 \cdot 167 = 3841$
   $e_2 = 11 \quad d_2 = 3459 \quad N_2 = 59 \cdot 83 = 4897$
   $e_3 = 13 \quad d_3 = 4501 \quad N_3 = 47 \cdot 107 = 5029$
3. The user S encrypts and sends the message $M = 452009$, with the threshold level $t = 2$ for the group G.
4. Encryption:

   $c_1 = 452009^{17} \bmod 3841 = 2053$
   $c_2 = 452009^{11} \bmod 4897 = 2197$
   $c_3 = 452009^{13} \bmod 5029 = 3845$

Parameters of the CRT:

$$N^1 = 4897 \cdot 5029 = 24\,627\,013$$
$$N^2 = 3841 \cdot 5029 = 19316389$$
$$N^3 = 3841 \cdot 4897 = 18809377$$

$$(N^1)^{-1} \bmod N_1 = 24\,627\,013^{-1} \bmod 3841 = 174$$
$$(N^2)^{-1} \bmod N_2 = 19316389^{-1} \bmod 4897 = 1973$$
$$(N^3)^{-1} \bmod N_3 = 18809377^{-1} \bmod 5029 = 2775$$

$$\begin{aligned} C = {} & (2053 \cdot 24627013 \cdot 174 + 2197 \cdot 19316389 \cdot 1973 \\ & + 3845 \cdot 18809377 \cdot 2775) \bmod 94592356933 \\ = {} & 79682507303 \end{aligned}$$

5. The user S sends the ciphertext $(79682507303, 2)$ to the group G.

6. Users P_1 and P_2 decrypt the message:

$$m_1 = C^{d_1} \bmod N_1 = 79682507303^{1289} \bmod 3841 = 2612$$
$$m_2 = C^{d_2} \bmod N_2 = 79682507303^{3459} \bmod 4897 = 1485$$

$$\begin{aligned} M_t = {} & 2612 \cdot 4897 \cdot (4897^{-1} \bmod 3841) \\ & + 1485 \cdot 3841 \cdot (3841^{-1} \bmod 4897) \\ = {} & (2612 \cdot 4897 \cdot 3139 \\ & + 1485 \cdot 3841 \cdot 895) \bmod 18809377 \\ = {} & 452009 \end{aligned}$$

References

[1] B. Nakielski, J. Pomykała, and J. A. Pomykała, "Multi-threshold signature", *J. Telecommun. Inform. Technol.*, no. 1, pp. 51–55, 2008.

[2] J. Pomykała and T. Warchoł, "Threshold signatures in dynamic groups ", in *Proc. Fut. Gener. Commun. Netw. 2007*, Jeju-Island, Korea, 2007, pp. 32–37.

[3] J. Pomykała and T. Warchoł , "Dynamic multi-threshold signatures without the trusted dealer", *Int. J. Multimed. Ubiquit. Eng.*, vol. 3, pp. 31–42, July 2008.

[4] H. Ghodosi, J. Pieprzyk, and R. Safavi-Naini, "Dynamic threshold cryptosystems: a new scheme in group oriented cryptography", in *Proc. Pragocrypt'96*, Prague, Czech Republic, 1996, pp. 370–379.

[5] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[6] A. Shamir, "How to share a secret", *Commun. ACM*, vol. 22, pp. 612–613, 1979.

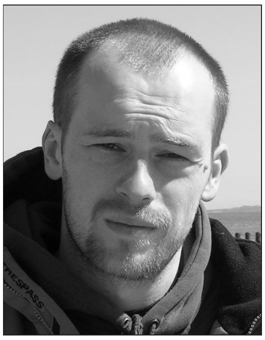
[7] B. Nakielski, J. Pomykała, and J. A. Pomykała, "Wykorzystanie deszyfrowania progowego w bazach danych", *Biul. WAT*, vol. LVII, no. 4, pp. 183–196, 2008 (in Polish).

[8] B. Schneier, *Applied Cryptography*. New York: Wiley, 1996.

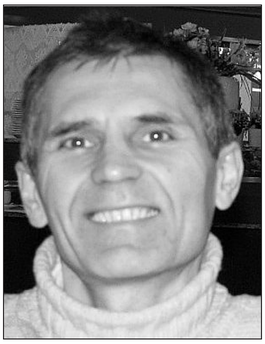
[9] R. Di Pietro, L. V. Mancini, and G. Zanin, *Efficient and Adaptive Threshold Signatures for Ad-Hoc Networks*. Electronic Notes in Theoretical Computer Science. Amsterdam: Elsevier, 2007, vol. 171, pp. 93–105.

[10] J. Pomykała and B. Źrałek, "Threshold flexible siganture scheme in dynamic groups", in *Proc. ACS Conf.*, Międzyzdroje, Poland, 2008.

[11] T. Tassa, "Hierarchical threshold secret sharing", *J. Cryptol.*, vol. 20, pp. 237–264, 2007.



Bartosz Nakielski was born in Warsaw, Poland, in 1979. He received his M.Sc. in mathematics from Department of Mathematics, Mechanics and Informatics of Warsaw University. His thesis was titled "Arithmetical aspects of digital signatures". He was working as a certificate authority administrator in Information Security Department in Social Insurance Institution (years 2004–2007). Since January 2008 he works in Security Department in National Bank of Poland.
e-mail: barteknakielski@aster.pl



Jacek Pomykała works at the Faculty of Mathematics, Informatics and Mechanics of Warsaw University, Poland, since 1985. He received the Ph.D. and D.Sc. degrees in 1986 and 1997, respectively. He has published over 20 papers mainly in mathematical journalas. He is also one of the authors of the book concerning the information systems and cryptography. He had many research visits (two long term visits) in Europe, America, Asia and has been an invited speaker in many international conferences in mathematics and computer science.
e-mail: pomykala@mimuw.edu.pl
Institute of Mathematics
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw
Banacha st 2
02-097 Warsaw, Poland