Paper

Multi-threshold signature

Bartosz Nakielski, Jacek Pomykała, and Janusz Andrzej Pomykała

Abstract—The work presents a new signature scheme, called the multi-threshold signature, which generalizes the concept of multisignature and threshold signature. This scheme protects the anonymity of signers in a way the group signature does – in exceptional circumstances the identities of signers may be revealed. Due to the new party – completer, in our scheme the threshold size may vary together with the message to be signed. The presented scheme is based on the RSA signature standard, however other signature standards might be applied to it as well.

Keywords*— *public key cryptography, threshold signature, multisignature, secret sharing.

1. Introduction

Threshold and multiparty cryptography represent a wide and important area of the modern cryptography. The large part of it deals with the signature schemes such as threshold signatures and multisignatures.

Threshold signatures (c.f. [3, 9]) allow any group of l users to create a signature provided $l \geq t$ (where t is a threshold level, fixed in advance). The multisignature allow any group of members to sign the given message. The identities of signers are recognized in the verification phase and then the decision if the signature is accepted is made (see [1, 4–6]). The verification of the signature applies the public keys of corresponding signers.

This paper is motivated by the following problem: given the group G of cardinality l and the pair (m,t) we are interested in the cryptographic scheme that allows any subgroup of at least t members to sign the message m. In distinction to the common (t,l)- threshold type scheme, here the value of t is not fixed in advance, but may vary together with the message m to be signed. Thus it might be very useful in applications, where the number of members required to agree upon the given document depends for instance on the document's "priority".

Another motivation is to propose the flexible signature scheme, which according to the requirement is anonymous or admits the signer's identification. This flexibility was not the subject of the previous papers, which generally speaking treat both solutions in separate schemes (c.f. [2] for example). From the practical point of view this ability seems to be significant in applications, and the proposed scheme provides the essential computational savings by joining both options within one cryptographic scheme. Therefore as an input for the signing algorithm is the triple (m,t,b), where $b \in \{0,1\}$ points out if the signature should be anonymous or with the signer's identification. The resulting signature is to be verified by any user applying the public key related to G. Similarly as in the conventional threshold signature scheme we require, that any subgroup of cardinality less than t is not able to generate the valid (i.e., accepted in the verification phase) signature, attached to the pair (m,t) (in fact it is not able to obtain any nontrivial information about the group G secret value related to its public key).

In the conventional threshold signature the group public key corresponds to the given value of the threshold size t. The idea of our solution relies on the enlarging somewhat the original group G, so that the public key corresponds to the bigger threshold size t'. Then the additional shares (handled by the additional (trusted) party C) will ensure the valid threshold size $t \leq t'$ of the original signer's group. One could extend the above idea considering not necessarily trusted completer (e.g., C being another group of signers). Such a development in direction of dynamic groups was considered in [10]. The presented scheme is based on the RSA [7] cryptosystem, and the Shamir secret sharing protocol [8]. The paper contains the detailed description of the corresponding multi-threshold signature scheme and the proof of its correctness.

2. General system model

2.1. Participants

We assume there are three parties involved in the protocol:

1. Group $G = \{P_1, P_2, ..., P_K\}$.
2. The trusted dealer D responsible for the generation of private and public key of G and the corresponding shares for the group members $P_1, P_2 ..., P_K$ and completer C.
3. The trusted completer C responsible for flexibility of the threshold level.

We assume that the dealer D is connected with the members P_i and the completer C by the secure channels. Communication between C and members of G goes through group message board (GMB) where all the partial signatures are published (only C and G have an access to it).

2.2. Notation

Throughout the paper N is a positive integer such that: $N = pq$, where $p = 2p' + 1$, $q = 2q' + 1$ and p, q, p', q' are prime numbers satisfying $\min(p'q') > 2K$, where $K = |G|$.

By λ we denote the Carmichael lambda function defined as

$\lambda(\prod_p(p^{\alpha_p}) = \mathrm{lcm}_p\lambda(p^{\alpha_p})$, where:

$\lambda(2) = 1 \quad \lambda(2^\alpha) = 2^{\alpha-2}$ for $\alpha \geq 3$,

$\lambda(4) = 2 \quad \lambda(p^\alpha) = p^{\alpha-1}(p-1)$ if p is an odd prime number.

Conventionally the elements e and d are mutually inverse elements in $Z^*_{\lambda(N)}$, i.e., $ed \equiv 1 \mod \lambda(N)$.
We assume that every member $P_i \in G$ is equipped with the RSA keys (N_i, e_i, d_i), respectively, needed for the authentication process within corresponding parties or members. To assure the uniqueness of m' at the end of the verification process we assume that $N_1 < N_2 < ... < N_K$:

$$A_i = \prod_{\substack{j=1 \\ j\neq i}}^{2K}(x_i - x_j) \mod \lambda(N) \text{ for } i = 1,2,...,2K$$

(x_i are numbers assigned to P_i).
Moreover we let $A'_i = A_i/2^{\varpi_i}$, where ϖ_i is the highest power of 2 dividing A_i and $\varpi = \max_{i\in I}\varpi_i$.
Throughout the paper h will denote a given secure hash function.

3. Initialization phase

In the initialization phase the dealer D performs the following steps:

1. Generates the key pair (d,e) and a random polynomial:
   $f(x) = d + c_1x + c_2x^2 + ... + c_Kx^K \in Z^*_{\lambda(N)}[x]$.

2. Computes the shares $s_i = f(x_i)(A'_i)^{-1} \mod \lambda(N)$ and sends them to P_i $(i \leq K)$ and to C (for $K+1 \leq i \leq 2K$).

   Remark 1. Since $\min(p',q') > 2K$ the odd numbers A'_i are invertible $\mod \lambda(N)$.

3. Selects $g \in Z^*_N$ of order equal to $\lambda(N)$ and sends $g^{-1} \mod N$ and $z_i = (g^{s_i})^{-1} \mod N$ to the completer C.

4. He publishes the group public key $gpk = (N, e, \varpi)$.

4. The anonymous signing phase

Assume that the tuple (m,t,b) (m is the message, $t \in \{1,2,...,K\}$ is the threshold level and $b \in \{0,1\}$ points out the signature type (anonymous or with signers identification)), is given to G and C in order to be signed by a given subgroup of G. Then the following steps are performed:

1. **Completer's computation**. The completer C computes $m^* = h(m,t,b)$ and applies the partial signature generation algorithm to compute and publish in the *GMB* the following partial signatures: $\sigma_{i_{t+1}}(m^*)$, $\sigma_{i_{t+2}}(m^*), ..., \sigma_{i_{K+1}}(m^*)$ (where $\sigma_i(m^*) = (m^*)^{s_i} \mod N$) together with the sequence $i_{t+1}, i_{t+2}, ..., i_{K+1}$ of terms contained in the interval $(K, 2K]$.

2. **Group signing**. The group members who decide to sign m, compute $m^* = h(m,t,b)$ and publish their partial signatures $\sigma_{i_1}(m^*), \sigma_{i_2}(m^*), ..., \sigma_{i_t}(m^*)$ $(1 \leq i_1 < i_2 < ... < i_t \leq K)$ in the *GMB*.

3. **Partial signature verification**. The completer selects a random $r \in Z_{\lambda(N)}$ computes $v^* = (\frac{m^*}{g})^r \mod N$ and sends it to the members participating in the signature generation. Next he computes $v_j = (z_{i_j}\sigma_{i_j}(m^*))^r \mod N$. Each member compute $v^*_j = (v^*)^{s_j} \mod N$ and sends it to the completer. Completer accepts the partial signature σ_{i_j} if and only if $v_j = v^*_j$.

4. **Generation of the full signature**. With the aid of the share combining algorithm the $K+1$ valid signatures $\sigma_{i_1}(m^*)$, $\sigma_{i_2}(m^*), ..., \sigma_{i_{K+1}}(m^*)$ allow any delegated signer to compute the anonymous signature $((m,t,0), \sigma_B(m^*))$, where:

   $B = \{i_1, i_2, \ldots, i_t, i_{t+1}, \ldots, i_{K+1}\}$,

   $\sigma_B(m^*) = \prod_{j\in B}\sigma^{a_i}_{i_j}(m^*) \mod n$ and

   $a_i = 2^{\varpi-\varpi_i}\prod_{\substack{j\in B \\ j\neq i}}(0 - x_j)\prod_{j\notin B}(x_i - x_j)$ is the appropriate Lagrange coefficient for the group B.

   When the signature $\sigma_B(m^*)$ (verified by any signer using gpk) occurs in the *GMB*, the anonymous signing is finished and $\sigma_B(m^*)$ is published.

5. The authorization phase

In the following part the members $P_i \in B$ authorize subsequently their signature using the private keys d_i. We remark that the description of B contains the subscripts of the corresponding signers. They perform the following steps:

1. P_1 computes the message $m' = h(m^*, \sigma, B)$, signs it using his private key d_1 and sends the obtained ciphertext $\delta_1 = (m')^{d_1} \mod N_1$ to the second member P_2.

2. P_2 verifies if $(\delta_1)^{e_1} \equiv m' \mod N_1$ if so, he computes $\delta_2 = (\delta_1)^{d_2} \mod N_2$ and sends it to P_3 (otherwise he publishes in *GMB* information about this disagreement and stops the protocol).

3. Similarly P_3 verifies the obtained ciphertext δ_2 using the public keys (e_2, N_2) and (e_1, N_1), respectively, computes $\delta_3 = (\delta_2)^{d_3} \mod N_3$, sends it to P_4 and so on.

4. The last member $P_t \in B$ verifies δ_{t-1} using the public keys:

   $(e_{t-1},N_{t-1}),(e_{t-2},N_{t-2}),...(e_1,N_1)$ and, if the verification is correct, he computes $\delta_t = (\delta_{t-1})^{d_t} \mod N_t$ and publishes it in *GMB*.

5. The full signature of the message m is the 5 – tuple: $((m,t,\sigma),B,\delta_t)$, where $P_i \in B$ are ordered as obove.

According to the requirements, the chosen member of the group G publishes the anonymous signature $((m,t,0),\sigma)$ or the full signature $((m,t,1),\sigma,B,\delta_t)$.
The anonymous signature does not imply any information about the identities of the members of B. It proves only that at least t members of group G have signed the document m.

6. The verification phase

After receiving the anonymous signature (m,t,σ) the verifier uses the group public key (e,N,ϖ) to compute $m^* = h(m,t,b) \mod N$ and then accepts it provided $\sigma^e \equiv (m^*)^{2^\varpi} \mod N$. To verify the full signature (m,t,σ,B,δ_t) one first computes $m' = h(m^*,\sigma,B)$ and then accepts the full signature provided $(...((\delta_t^{e_t} \mod N_t)^{e_{t-1}} \mod N_{t-1})...)^{e_1} \equiv m' \mod N_1$.

Theorem 1. The correctly created signature will be accepted in the verification phase.

Proof. First let us consider the anonymous signature (m,t,σ). It is sufficient to prove that $\sigma^e \equiv (m^*)^{2^\varpi} \mod N$. By definition we have $\sigma = \prod_{i\in B} \sigma_i^{a_i} = (m^*)^{\sum_{i\in B} s_i a_i}$, where:

$$s_i = f(x_i)(A_i')^{-1} \mod \lambda(N), \tag{1}$$

$$a_i = 2^{\varpi-\varpi_i} \prod_{j\in B, j\neq i} (0-x_j) \prod_{j\notin B} (x_i - x_j). \tag{2}$$

It remains therefore to show that $\sum_{i\in B} s_i a_i = 2^\varpi d = 2^\varpi f(0) = F(0) \mod \lambda(N)$.
In this connection we apply the Lagrange interpolation formula for

$F(x) = 2^\varpi f(x) \in Z^*_{\lambda(N)}[x]$ whose graph passes by the points $(x_{i_1},F(x_{i_1}))$,

$(x_{i_2},F(x_{i_2})),\ldots,(x_{i_{K+1}},F(x_{i_{K+1}}))$, where $B = \{i_1,i_2,\ldots,i_t, i_{t+1},\ldots,i_{K+1}\}$.

We have $F(x) = \sum_{i\in B} f(x_i)(2^\varpi \Lambda_i(x)) \mod \lambda(N)$, where:

$$2^\varpi \Lambda_i(x) = 2^{\varpi-\varpi_i} \prod_{j\in B, j\neq i} (2^{\varpi_i}(\frac{x-x_j}{x_i-x_j})) =$$

$$= 2^{\varpi-\varpi_i} \prod_{j\in B, j\neq i} (x-x_j) \prod_{j\notin B} (x_i-x_j) \prod_{\substack{j=1 \\ j\neq i}}^{2t} \frac{2^{\varpi_i}}{(x_i-x_j)} \mod \lambda(N).$$

Hence in view of Eqs. (1) and (2) and definition of A_i' we obtain:

$$F(0) = \sum_{i\in B} f(x_i)(A_i')^{-1} \cdot 2^{\varpi-\varpi_i} \prod_{j\in B, j\neq i} (0-x_j) \prod_{j\notin B} (x_i-x_j) =$$
$$= \sum_{i\in B} s_i a_i \mod \lambda(N), \text{ as claimed.}$$

To verify the full signature (m,t,σ,B,δ_t) we use the bijectivity of transformation $x \longmapsto x^{d_i} \mod N_i$ $(1 \le i \le t)$ and the inequalities:

$N_1 < N_2 < ... < N_K$ (that assure the uniqueness of m' at the end of the verification process).

Taking δ_t to the power e_t we obtain the unique $\delta_{t-1} \mod N_{t-1}$ then (using e_{t-1}) the unique $\delta_{t-2} \mod N_{t-2}$ and finally the unique
$((m')^{d_1})^{e_1} = m' \mod N_1$ as required.

■

7. Conclusions

Two basic benefits of the presented scheme are the scaleability (in threshold size) and generality – it might be useful for the applications typical for the threshold-type signatures or multisignatures.
The final output is the pair: anonymous G-signature and the full signature (containing the signers' identifications).
The completer can be regarded as a well protected machine which for the input value (m,t,b) outputs the corresponding partial signatures.
As proved in [10] the multi-threshold device with C regarded as another group of signers could be developed in the direction of dynamic groups signatures schemes.

Appendix – an example

1. System parameters:

   $p = 23$ $q = 47$ $N = 1081$ $\lambda(N) = 506$
   $p\prime = 11$ $q\prime = 23$ $t = 3$ $\min(p',q') > 6 = 2t$
   $e = 13$ $d = 39$ $m^* = 7$ $G = \{P_1,P_2,P_3\}$

2. Dealer generates random polynomial:

   $f(x) = 3x^3 + 5x^2 + 7x + 39$ and sets $x_i = i$ which implies:

   $A_i = \prod_{\substack{j=1 \\ j\neq i}}^{6} (i-j) \mod 506$

3. We have:

   $A_1 = 386$ $A_1' = 193$ $\varpi_1 = 1$ $(A_1')^{-1} = 409$ $f(1) = 54$
   $A_2 = 24$ $A_2' = 3$ $\varpi_2 = 3$ $(A_2')^{-1} = 169$ $f(2) = 97$
   $A_3 = 494$ $A_3' = 247$ $\varpi_3 = 1$ $(A_3')^{-1} = 295$ $f(3) = 186$
   $A_4 = 12$ $A_4' = 3$ $\varpi_4 = 2$ $(A_4')^{-1} = 169$ $f(4) = 339$
   $A_5 = 482$ $A_5' = 241$ $\varpi_5 = 1$ $(A_5')^{-1} = 21$ $f(5) = 68$
   $A_6 = 120$ $A_6' = 15$ $\varpi_6 = 3$ $(A_6')^{-1} = 135$ $f(6) = 403$
   $\varpi = \max_i \varpi_i = 3$

4. Dealer, using the table above, computes:
   $s_i = f(i) * (A'_i)^{-1} \mod \lambda(N)$
   $s_1 = (54 * 409) \mod 506 = 328$
   $s_2 = (97 * 169) \mod 506 = 201$
   $s_3 = (186 * 295) \mod 506 = 222$
   $s_4 = (339 * 169) \mod 506 = 113$
   $s_5 = (68 * 21) \mod 506 = 416$
   $s_6 = (403 * 135) \mod 506 = 263$

5. Dealer selects $g = 3$ and sends to the completer the following values:
   $g^{-1} \mod N = 3^{-1} \mod 1081 = 721$ and
   $z_1 = (g^{s_1})^{-1} \mod N = 331$
   $z_2 = (g^{s_2})^{-1} \mod N = 259$
   $z_3 = (g^{s_3})^{-1} \mod N = 639$
   $z_4 = (g^{s_4})^{-1} \mod N = 949$
   $z_5 = (g^{s_5})^{-1} \mod N = 538$
   $z_6 = (g^{s_6})^{-1} \mod N = 647$

6. We assume that $m^* = h(m,2,0) = 7$ and $B = \{2,3,4,6\}$.

7. P_2 and P_3 generate and send to the completer their partial signatures:
   $\sigma_2 = (m^*)^{s_2} \mod N = 7^{201} \mod 1081 = 711$
   $\sigma_3 = (m^*)^{s_3} \mod N = 7^{222} \mod 1081 = 3$

8. Completer verifies partial signatures created by P_2 and P_3.
   He selects $r = 5$ and sends v^* to P_2 and P_3, where
   $v^* = (m^* g^{-1})^r \mod N = (7 \cdot 721)^5 \mod 1081 = 732$.
   Next the completer computes:
   $v_2 = (z_2 \cdot \sigma_2(m^*))^r \mod N = (259 \cdot 711)^5 \mod 1081 = 948$
   $v_3 = (z_3 \cdot \sigma_3(m^*))^r \mod N = (639 \cdot 3)^5 \mod 1081 = 16$

9. Members P_2 and P_3 compute and send to the completer:
   $v_2^* = (v^*)^{s_2} = 732^{201} \mod 1081 = 948$
   $v_3^* = (v^*)^{s_3} = 732^{222} \mod 1081 = 16$

10. Completer accepts σ_2, σ_3 and creates two missing partial signatures:
    $\sigma_4 = (m^*)^{s_4} \mod N = 7^{113} \mod 1081 = 964$ and
    $\sigma_6 = (m^*)^{s_6} \mod N = 7^{263} \mod 1081 = 79$

11. P_2 (as a delegated user) computes the interpolation coefficients:
    $a_2 = 2^1(0-3)(0-4)(0-6)(2-1)(2-5) \mod 506 = 216$
    $a_3 = 2^2(0-2)(0-4)(0-6)(3-1)(3-5) \mod 506 = 262$
    $a_4 = 2^1(0-2)(0-3)(0-6)(4-1)(4-5) \mod 506 = 216$
    $a_6 = 2^0(0-2)(0-3)(0-4)(6-1)(6-5) \mod 506 = 79$
    and finally he computes the anonymous signature:
    $\sigma = \prod_{i \in B} \sigma_i^{a_i} =$
    $(711^{216} * 3^{262} * 964^{216} * 79^{386}) \mod 1081 = 354$

12. To verify the signature $((m,1,0),354)$ we use the public key $(1081,13,3)$ and compute:
    $\sigma^e \mod N = 354^{13} \mod 1081 = 909$
    $(m^*)^{2^{\varpi}} \mod N = 7^8 \mod 1081 = 909$ and accept the signature.

References

[1] C. Boyd, "Digital mutisignatures", in *Cryptography and Coding*, H. Baker and F. Piper, Eds. Oxford: Oxford University Press, 1989.

[2] C. C. Chang and H. F. Huang, "An efficient and practical (t,n) threshold signature scheme with known signers", *Fundam. Informat.*, vol. 53, no. 3, pp. 243–253, 2003.

[3] Y. Desmedt and Y. Frankel, "Threshold cryptosystems", in *Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology, LNCS*. Berlin: Springer, 1989, vol. 435, pp. 307–319.

[4] K. Itakura and K. Nakamura, "A public key cryptosystem suitable for digital multisignatures", *NEC Res. Develop.*, no. 71, pp. 1–8, 1983.

[5] S. Micali, K. Ohta, and L. Reyzin, "Accountable – subgroup multisignatures", in *Proc. 8th ACM Conf. Comput. Commun. Secur. CCS'01*, Philadelphia, USA, 2001.

[6] T. Okamoto, "A digital multisignature scheme using bijective public-key cryptosystems", *ACM Trans. Comput. Syst.*, vol. 6, iss. 4, pp. 432–441, 1988.

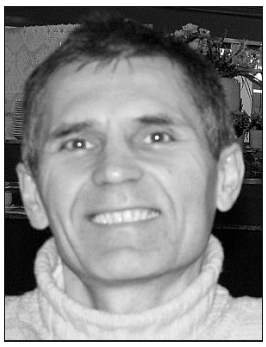
[7] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

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

[9] V. Shoup, "Practical threshold signatures", in *Proc. Adv. Cryptol. EuroCrypt 2000*, Bruges, Belgium, 2000.

[10] J. Pomykała and T. Warchoł, "Threshold signatures in dynamic groups", in *Proc. Fut. Gener. Commun. Netw.*, Jeju-Island, Korea, 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 has defended his Ph.D. thesis in 1986 in the area of sieve methods in number theory. In 1997 he has made the habilitation in the area of automorphic L-functions and distribution of their zeros. Now his main area of subject is cryptology and number theory. He has published over 20 papers mainly in mathematical journalas and has been directed two KBN grants on the arithmetics of elliptic curves and zeros distributions of Hecke's L-functions, respectively. 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.

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

Janusz Andrzej Pomykała received B.Sc. and M.Sc. in mathematics from the University of Warsaw in 1983, focusing on the thesis about iterated forcing method, written under prof. W. Guzicki. He has taught and done research at various colleges and universities in Poland. He was working also in ILLC at the University of Amsterdam, due to Tempus program. His fields of interest include mathematical logic, algebra, use of novel methods of data analysis, information systems and databases, design of safe systems. He is also active in the popularization of mathematics and computer science on elementary level. He works in the Department of Applied Computer Science and Management in Warsaw Management Academy. He has written about 20 scientific articles and one book (in Polish). He was on more then 10 international conferences and workshops in the field of computer science.

e-mail: pomykala_andrzej@mac.edu.pl
Department of Applied Computer Science and Management
Warsaw Management Academy
Kawęczyńska st 36
03-772 Warsaw, Poland