A Tripartite Strong Designated Verifier Scheme Based On Threshold RSA Signatures

Jérôme Dossogne[1] **and Olivier Markowitch**[1]
[1]Computer Science Department, Université Libre de Bruxelles, Brussels, Belgium

Abstract— *In this paper we propose a new designated verifier signature scheme based on the threshold signature scheme presented [8] by Ghodosi and Pieprzyk. The advantages of the new scheme compared with previously proposed solutions are its computational efficiency and its simple and rational design that allows distributed implementations of the computations and suits the needs of both single individuals and organizations wishing to set a threshold on the number of required signers. The signing key size remains bounded by the size of a RSA modulus. Moreover, in addition to the computation of a classical RSA signature, the signature generation of our scheme needs only one modular multiplication; the verification process in our scheme remains the same than a classical RSA signature verification.*

Keywords: Digital Signature, Designated verifier, Threshold cryptosystems

1. Introduction

On the basis of the works of Chaum and van Antwerpen [1] on undeniable signatures and in order to prevent blackmailing [3], [4] and mafia attacks [2], Jakobsson, Sako and Impagliazzo introduced the notion of designated verifier signatures [5]. A valid designated verifier signature convinces only a specified recipient (nobody else may be able to deduce anything about the signature). To achieve this property, a designated verifier of a signature s must be able to produce a signature s' intended to himself that is indistinguishable from s.

In 2001, Rivest, Shamir and Tauman introduced a ring signature scheme [6] that allows a group member to generate a signature on the name of the group. A special case of these signatures provides designated verifier signatures.

In 2003 Saeednia, Kremer and Markowitch proposed an efficient strong designated verifier signature scheme [10]. As defined later by Lee and Chang [7], the strongness property refers to the requirement of the designated verifier to use his/her secret key to verify the validity of a signature.

In this paper we propose strong designated verifier schemes that are based on the threshold signature scheme of Ghodosi and Pieprzyk [8]. The advantages of our new scheme compared with previously proposed schemes are its computational efficiency and its simple and rational design. The straightforwardness of the design allows a great flexibility which leads to easily design several versions of the scheme depending of the context. The proposed design allows distributed implementations of the computations and suits the needs of both single individuals and organizations wishing to set a threshold on the number of required signers.

In this paper, we focus on a scheme that considers an individual signer and we present two versions of the resulting tripartite scheme. The first version is only intended to present our designated verifier signature scheme. The second scheme supplements the protocol in ensuring the confidentiality of the signed message.

Our schemes present the following advantages: (1) the signing key size is bounded by the size of a RSA modulus, (2) the schemes rely on simple RSA signatures, (3) in addition to the computation of a classical RSA signature, the signature generation of our scheme needs only one modular multiplication, (4) the verification process in our scheme remains the same than a classical RSA signature verification. Therefore, our schemes can easily be used in existing protocols that are based on RSA signatures in order to add the property of strong designated verifier. The existing keys can be reused as well as most of the existing software. The flexibility offered by sharing a RSA key, using a threshold cryptosystems, allows to adapt the scheme to a lot of contexts including a large variation in the number of participants. Therefore, our solution suits the needs of individuals as well as the needs of larger organizations.

The price to pay to achieve these properties is that the mininum number of entities involved in the scheme is three, therefore when considering an individual signer, the method considers the participation of a trusted third party.

The paper is organised as follows. In section 2, we present the threshold cryptosystem used to build our designated verifier signatures schemes. The first version of our designated verifier signature scheme is described in the section 3. In section 4 we address the problem of confidentiality of the signed message and present an adaptation of the designated verifier scheme that is based on hashes. In section 5 we present the strong property of our designated verifier schemes. We conclude in section 6.

2. A threshold cryptosystem

Our system is based on Ghodosi and Pieprzyk threshold signature scheme [8] which relies on Shamir Threshold

cryptosytems [9]. In our scheme a trusted third party is involved as a contributor in order to help in producing the signatures.

Therefore, the participants in the scheme are:

- u_1 the *signer*,
- u_2 the *contributor*,
- u_3 the *designated verifier*.

The third party u_2 is trusted to cooperate by signing the messages it receives and to keep secret its signing key.

Since our scheme is based on a (k,l)-threshold cryptosystem and RSA signatures, we share the RSA secret key d between l potential signers (the RSA public key (e,n) remains in one piece) . In practice, we propose here to split the secret key in three shares such that $d = (\sum_{u_i \in A} d_i) \pmod n$ where A is the set of signatories and to use a $(k=2, l=3)$-threshold scheme.

We define $f(x)$, a polynomial of order $k-1$: $f(x) = d + ax$, where $a \neq 0$ is randomly chosen. On that basis we create l shares $s_i = f(x_i)$, $x_i = i, \forall i : 0 < i \leq l$

$$\begin{cases} s_1 = f(1) = d + 1a \pmod n \\ s_2 = f(2) = d + 2a \pmod n \\ s_3 = f(3) = d + 3a \pmod n \\ d = f(0) \end{cases} \tag{1}$$

Therefore a linear combination of k distinct s_i allows to retrieve d.

On the basis of Lagrange polynomials, we can consider the two following cases:

1) if u_1 and u_2 are signing :

$$\begin{aligned} f(x) &= s_1 \frac{x - x_2}{x_1 - x_2} + s_2 \frac{x - x_1}{x_2 - x_1} \pmod n \\ &= s_1(2 - x) + s_2(x - 1) \pmod n \end{aligned} \tag{2}$$

2) if u_2 and u_3 are signing :

$$\begin{aligned} f(x) &= s_2 \frac{x - x_3}{x_2 - x_3} + s_3 \frac{x - x_2}{x_3 - x_2} \pmod n \\ &= s_2(3 - x) + s_3(x - 2) \pmod n \end{aligned} \tag{3}$$

Having

$$d = f(0) = \sum_{u_i \in A} (s_i \prod_{u_j \in A, j \neq i} \frac{0 - x_j}{x_i - x_j}) \pmod n \tag{4}$$

we can choose

$$d_i = s_i \prod_{u_j \in A, j \neq i} \frac{j}{j - i} \pmod n \tag{5}$$

Since we have:

$$\begin{cases} d_i < n, \forall i : 0 < i \leq l \\ d = (\sum_{u_i \in A} d_i) \pmod n \Rightarrow \sum_{u_i \in A} d_i = I_A n + d \\ \sum_{u_i \in A} d_i = I_A n + d < |A| n d < |A| n \end{cases} \tag{6}$$

where $|A|$ is the number of signatories.

Therefore, we obtain:

1) if u_1 and u_2 are signing :

$$\begin{cases} d_1 = s_1 \frac{2}{2-1} = 2s_1 \\ d_{2a} = s_2 \frac{1}{1-2} = -s_2 \end{cases} \tag{7}$$

$$\Longrightarrow^{(1)}$$

$$\begin{cases} d_1 = 2d + 2a \pmod n \\ d_{2a} = -d - 2a \pmod n \end{cases} \tag{8}$$

and

$$d_1 + d_{2a} = d \bmod n \tag{9}$$

2) if u_2 and u_3 are signing :

$$\begin{cases} d_{2b} = 3s_2 \\ d_3 = -s_2 \end{cases} \tag{10}$$

$$\Longrightarrow^{(1)}$$

$$\begin{cases} d_{2b} = 3d + 6a \bmod n \\ d_3 = -2d - 6a \bmod n \end{cases} \tag{11}$$

and

$$d_{2b} + d_3 = d \bmod n \tag{12}$$

Notice that even owning d_{2a} and d_{2b}, u_2 cannot reconstruct d since these two values are linearly dependent, therefore u_2 does not gain any additionnal information by having two shares instead of one. Nervertheless, we will still describe the protocol using two different names but keeping in mind that sending only one of the two during the distribution procedure is enough.

3. Our designated verifier signature scheme

Our objective is to obtain a signature

$$\sigma = m^d \pmod n \tag{13}$$

by computing

$$\begin{aligned} \sigma' &= \prod_{u_i \in A} \sigma_i \\ &= m^{\sum_{u_i \in A} (d_i \bmod n)} \\ &= m^{d + I_A n} \\ &= (\sigma m^{I_A n}) \pmod n \end{aligned} \tag{14}$$

where

$$I_A = \frac{(\sum_{u_i \in A} d_i) - d}{n} \tag{15}$$

Therefore, while creating the shares we have to compute $m^{-I_A n}$ in order to obtain

$$\sigma = \sigma^{'} m^{-I_A n} = (\sigma m^{I_A n} m^{-I_A n}) = m^d \pmod n \tag{16}$$

Thus, with $A1 = \{u_1, u_2\}$ and $A3 = \{u_3, u_2\}$ we compute

$$I_{A1} = \frac{d_1 + d_{2a} - d}{n} \tag{17}$$

and

$$I_{A3} = \frac{d_{2b} + d_3 - d}{n} \tag{18}$$

while creating the share so that :

1) if u_1 and u_2 are signing :

$$\begin{aligned} \sigma &= \sigma_1 \sigma_2 m^{-I_{A1} n} = (\sigma m^{I_{A1} n} m^{-I_{A1} n}) \\ &= m^d \bmod n \end{aligned} \tag{19}$$

2) if u_2 and u_3 are signing :

$$\begin{aligned} \sigma &= \sigma_2 \sigma_3 m^{-I_{A3} n} = (\sigma m^{I_{A3} n} m^{-I_{A3} n}) \\ &= m^d \bmod n \end{aligned} \tag{20}$$

Setup

The keys and values $(e, n), d_1, d_{2a}, d_{2b}, d_3$, I_{A1} and I_{A3} are distributed from the entity generating them to each participant: (d_1, n, I_{A1}) is transmitted to the signer u_1, (d_{2a}, d_{2b}, n) to the contributor u_2 and (e, n, d_3, I_{A3}) to the designated verifier u_3. We suppose that the transmissions are done via secure channels.

Signature generation

u_1 and u_2 are signing the message m coming from u_1

1) u_1 calculates $\sigma_1 = m^{d_1} \pmod n$
   $u_1 \rightarrow u_2 : m$

2) u_2 calculates $\sigma_2 = m^{d_{2a}} \pmod n$
   $u_2 \rightarrow u_1 : \sigma_2$

3) u_1 calculates $\sigma = \sigma_1 \sigma_2 m^{-I_{A1} n} \pmod n$
   $u_1 \rightarrow u_3 : \sigma, m$

Verification

The verifier u_3 checks if the signature is valid which is the case if $m = \sigma^e \pmod n$.

Simulation

u_2 and u_3 are signing the message m coming from u_3

1) u_3 calculates $\sigma_3 = m^{d_3} \pmod n$
   $u_3 \rightarrow u_2 : m$

2) u_2 calculates $\sigma_2 = m^{d_{2b}} \pmod n$
   $u_2 \rightarrow u_3 : \sigma_2$

3) u_3 calculates $\sigma = \sigma_3 \sigma_2 m^{-I_{A3} n} \pmod n$

u_3 checks if the signature is valid which is the case if $m = \sigma^e \pmod n$. Notice that $\sigma = m^d \pmod n$ is the same in both cases which means that having σ and m, no one can prove it was u_3 or u_1 who co-signed with u_2.

4. A designated verifier signature scheme with hashed values

In the designated verifier signature scheme presented in the previous section, the message to be signed is sent to u_2 in order to obtain his partial signature. If confidentiality is required, we propose an adapted version that makes use of hashed values.

To apply this solution, it is mandatory that the set of possible messages M is an infinite one or is so large that it would be impossible for u_2 to calculate all the hashes of all the equiprobable $m_i \in M, 0 < i < |M|$ (otherwise, u_2 could pre-compute some hashes in order to detect if u_1 sends a message corresponding to those hashes).

Setup

The keys and values $(e, n), d_1, d_{2a}, d_{2b}, d_3$, I_{A1} and I_{A3} are distributed from the entity generating them to each participant: (d_1, n, I_{A1}) is transmitted to the signer u_1, (d_{2a}, d_{2b}, n) to the contributor u_2 and (e, n, d_3, I_{A3}) to the designated verifier u_3. We suppose that the transmissions are done via secure channels.

Signature generation

u_1 and u_2 are signing the message m coming from u_1

1) u_1 calculates $h = H(m)$ and $\sigma_1 = h^{d_1} \pmod n$ where $H()$ is a known hash function
   $u_1 \rightarrow u_2 : \sigma_1$

2) u_2 calculates $\sigma_2 = h^{d_{2a}} \pmod n$
   $u_2 \rightarrow u_1 : \sigma_2$

3) u_1 calculates $\sigma = \sigma_1 \sigma_2 h^{-I_{A1} n} \pmod n$
   $u_1 \rightarrow u_3 : \sigma, m$

Verification

u_3 checks if the signature is valid which is the case if $hash(m) = \sigma^e \pmod n$

Simulation

u_2 and u_3 are signing the message m coming from u_3

1) u_3 calculates $h = H(m)$ and $\sigma_3 = h^{d_3} \pmod n$ where H is a known cryptographic hash function
   $u_3 \rightarrow u_2 : h$

2) u_2 calculates $\sigma_2 = h^{d_{2b}} \pmod n$
$u_2 \rightarrow u_3 : \sigma_2$

3) u_3 calculates $\sigma = \sigma_3 \sigma_2 h^{-I_{A3} n} \pmod n$

u_3 checks if the signature is valid which is the case if $H(m) = \sigma^e \pmod n$. Notice that $\sigma = H(m)^d \pmod n$ is the same in both cases which means that having σ and m, no one can prove it was u_3 or u_1 who co-signed with u_2.

5. Strong Designated Verifier Schemes

When considering our designated verifier signature schemes, since the verification key e is sent only to the designated verifier, we obtain strong designated verifier schemes. In particular this is possible since e does not have to be public because it is only required by u_3 to check the signatures.

6. Conclusion

In this paper we presented an efficient threshold designated verifier signature scheme that can be used by individual signers. It is cost effective and its design leads to an easy distribution of the computations, since the partial signatures can be computed in parallel by each participants. As mentioned here and explained in a soon to be submitted paper, the scheme can also be used by organizations wishing to set a threshold on the number of required signers.

References

[1] D. Chaum and H. Van Antwerpen, Undeniable signatures, Advances in Cryptology (Proccedings of Crypto '90), Lecture Notes in Computer Science, vol. 435, Springer-Verlag, 1991, pp. 212-216.

[2] Y. Desmedt, C. Goutier and S. Bengio, Special uses and abuses of the Fiat-Shamir passport protocol, Advances in Cryptology (Proccedings of Crypto '87), Lecture Notes in Computer Science, vol. 293, Springer-Verlag, 1988, pp. 21-39.

[3] Y. Desmedt and M. Yung, Weaknesses of Undeniable Signature Schemes (Extended Abstract), Advances in Cryptology (Proccedings of Eurocrypt '91), Lecture Notes in Computer Science, vol. 547, Springer-Verlag, 1992, pp. 205-220.

[4] M. Jakobsson, Blackmailing using Undeniable Signatures, Advances in Cryptology (Proccedings of Eurocrypt '94), Lecture Notes in Computer Science, vol. 950, Springer-Verlag, 1995, pp. 425-427.

[5] M. Jakobsson, K. Sako and R. Impagliazzo, Designated Verifier Proofs and Their Applications, Advances in Cryptology (Proceedings of Eurocrypt '96) , Lecture Notes in Computer Science, vol. 1070, Springer-Verlag, 1996, pp. 143-154.

[6] R. Rivest, A. Shamir and Y. Tauman, How to Leak a Secret, Advances in Cryptology (Proceedings of Asiacrypt '01), Lecture Notes in Computer Science, vol. 2248, Springer-Verlag, 2001, pp. 552-565.

[7] J.-S. Lee, J.H. Chang, Comment on Saeednia et al.'s strong designated verifier signature scheme, Computer Standards & Interfaces, vol. 31, no. 1, 2009, pp. 258-260.

[8] H. Ghodosi, J. Pieprzyk, An Ideal and Robust Threshold RSA, Progress in Cryptology - VIETCRYPT 2006, vol. 4341, Springer Berlin / Heidelberg, 2006, pp. 312-321.

[9] A. Shamir, How to share a secret, Commun. ACM, vol. 22, no. 11, 1979, pp. 612-613.

[10] S. Saeednia, S. Kremer, O. Markowitch, An efficient strong designated verifier signature scheme, ICISC'03, Lecture Notes in Computer Science, vol. 2971, Springer Berlin, 2004, pp. 40-54.