Fully Distributed Proxy Signature Schemes

Javier Herranz and Germán Sáez

Dept. Matemàtica Aplicada IV, Universitat Politècnica de Catalunya
C. Jordi Girona, 1-3, Mòdul C3, Campus Nord, 08034 Barcelona, Spain
e-mail: {jherranz,german}@mat.upc.es

Abstract

In a proxy signature scheme, a potential signer delegates his signing capability to a proxy entity, who signs a message on behalf of the original signer. All the proposals of proxy signature schemes made until now have been based on Schnorr's signature scheme. Threshold versions of these schemes have also been proposed, in which the power of the proxy signer is distributed among a group of players, in such a way that any subset with a minimum number (threshold) of players can sign a message on behalf of the original signer.

We consider a model that is fully distributed, because we want to distribute not only the power of the proxy signer, but also the original signer ability to delegate his signing capability. Furthermore, we consider general structures, instead of only the threshold ones, for both the tolerated subsets of dishonest players and the subsets of honest players authorized to execute a valid instance of the protocol, and in both the original and the proxy signer entities. We find sufficient combinatorial conditions that these structures must satisfy in order to design a fully distributed, secure and robust proxy signature scheme for this general scenario.

We propose such a scheme for this setting. It is based on the results of [8] and [15], and inherits the security of these two works.

Keywords. Proxy signature schemes, distributed cryptographic protocols, secret sharing schemes.

Fully Distributed Proxy Signature Schemes

Javier Herranz and Germán Sáez

1 Introduction

Sometimes a person or a company that has the capability and the necessity of signing a document does not have enough time to do so. Or perhaps this person, A, is keen to delegate his signing capability to another person, B, so B would sign documents on behalf of A if A had some (technical, logistical) problem.

In a more concrete (or practical) situation, we can imagine a company with many departments. One of them, A (finances, business connections, loans in a bank, for example) must sign documents regularly, but A has a lot of things to do in addition to signing, and besides A wants its documents to be signed even if it is not able to do so because of some problem. A solution for this company could be to have a department B, the *proxy department*, whose only job would be to sign documents on behalf of the other departments of the company.

This is the scenario for a proxy signature scheme: a potential signer A delegates his signing capability to a proxy signer, B (in some way, A tells B what kind of messages B can sign), and B signs a message on behalf of the original signer, A. The receiver of the message verifies the signature of B and the delegation of A together.

Proxy signature schemes must have some security properties; we list them in Section 2. According to these properties, the most complete proxy signature schemes proposed until now are that of Lee, Kim and Kim [8] and that of Kim, Park and Won [7]. These schemes, as well as the previous proposals [9, 16] of proxy signature schemes, are based on Schnorr's signature scheme [12], which is also revisited in Section 2.

In [15], Stinson and Strobl propose a distributed version of Schnorr's signature scheme, which is as secure as the non-distributed one; that is, existentially unforgeable under adaptively chosen message attacks (as Pointcheval and Stern proved in [11]). This distributed scheme is based on the joint generation of a random secret value. Distributed protocols provide more security and reliability than individual ones, because they tolerate some coalitions of participants to be corrupted or non-working at the moment of the execution of the protocol. In Section 3 we propose a general framework for distributed protocols; that is, we consider general structures (families of subsets of players) that determine both which subsets of players can perform some specific actions and which subsets of dishonest players the system will be able to tolerate. The threshold case, in which these subsets are defined according to their cardinality, is a particular case. We adapt to this general framework the verifiable secret sharing scheme of Pedersen [10], the joint generation of a random secret value of Gennaro et al. [5] and the threshold Schnorr's signature scheme of Stinson and Strobl [15].

In Section 4, we construct a fully distributed and secure proxy signature scheme, in the sense that we distribute not only the proxy signer (that is, B), but also the original signer, A, who delegates his signing capability. This scheme runs in the general framework introduced in Section 3. If the structures satisfy some combinatorial conditions that we state, the scheme is robust and unforgeable in the random oracle model under chosen message attacks, because it inherits its security from the security of the distributed Schnorr's signature scheme of [15] and the proxy signature scheme of [8]. The distribution of the original signer, the level of security of the scheme, and the fact that we consider a scenario which is more general than the threshold one, make our proposal more complete than the previous threshold proxy signature schemes ([7, 16, 6]).

Finally, in Section 5 we conclude by summing up our contribution and discussing some problems that remain open in the area of proxy signatures.

2 Proxy Signatures

The concept of proxy signature was introduced by Mambo, Usuda and Okamoto in [9]. They classified these signatures according to the delegation type and the protection of the proxy signer. Kim et al. [7] included warrant information in these schemes; that is, the signer A sends to the proxy B a signed message in which A explicitly delegates its signing capability to B, allowing B to sign some kind of messages (specified in the warrant information) on behalf of A.

The idea of these proxy signature schemes is the following: A sends a message and its signature to a proxy signer, B, who uses this information to construct a proxy key, which B will use to sign messages on behalf of A. This proxy key must contain some authentic information about the proxy signer, if we want these schemes to satisfy the security requirements of proxy signatures listed in the work of Mambo et al. [9]:

(i) Strong unforgeability: only a designated proxy signer can create a valid proxy signature for the original signer (even the original signer cannot do it).

(ii) Verifiability: a verifier of a proxy signature will be convinced in any way of the original signer's agreement on the signed message.

(iii) Strong identifiability: a proxy signature determines the identity of the corresponding proxy signer.

(iv) Strong undeniability: after creating a valid proxy signature for an original signer, the proxy signer cannot repudiate this signature against anyone.

In [8] Lee, Kim and Kim briefly modify the proposal of [7]: now the proxy signer B and the original signer A play asymmetric roles in the generation of a proxy signature, and so the warrant information must not contain an explicit delegation of A's signing capability. Besides, A does not need to designate a specific proxy signer. In [8], the authors add a new security requirement to proxy signature schemes (which their scheme, as well as that proposed in [7], satisfies):

(v) Prevention of misuse: the proxy signer cannot use the proxy key for other purposes than generating a valid proxy signature. That is, he cannot sign, with the proxy key, messages that have not been authorized by the original signer.

All the proposals of proxy signature schemes, like [8] and [7], are based on Schnorr's signature scheme ([12]).

2.1 Schnorr's Signature Scheme

In [12], Schnorr introduced the following signature scheme.

Let p and q be large primes with $q|p-1$. Let g be a generator of a multiplicative subgroup of $\mathbb{Z}_p^*$ with order q. $H()$ denotes a collision resistant hash function. (This will be the mathematical scenario in the rest of the paper.)

A signer A has a private key $x_A \in \mathbb{Z}_q^*$ and the corresponding public key $y_A = g^{x_A}$. To sign a message M, A acts as follows:

1. choose a random $k \in \mathbb{Z}_q^*$
2. compute $r = g^k \mod p$ and $s = k + x_A H(M, r) \mod q$
3. define the signature on M to be the pair (r, s)

The validity of the signature is verified by the recipient by checking that $g^s = r y_A^{H(M,r)}$.

In [11], Pointcheval and Stern proved that, in the random oracle model, existential forgery under adaptively chosen message attack of Schnorr's scheme is equivalent to the discrete logarithm problem in the group generated by the element g.

2.2 The proposal of Lee, Kim and Kim

The following proxy signature scheme has been introduced in [8]. It is based on the proposal of Kim et al. [7], with the difference that the warrant information signed by the original signer must not explicitly include either his identity or the identity of the proxy signer. This is possible because the original signer and the proxy signer do not play the same role in the generation of a proxy signature, and so the verifier can identify both of them.

Original signer A has the key pair (x_A, y_A), with $y_A = g^{x_A}$, whereas the (future) proxy signer B also has his user key pair (x_B, y_B), with $y_B = g^{x_B}$.

Generation of the proxy key: the original signer A uses Schnorr's scheme to sign warrant information M_ω, which should specify which messages A will allow the proxy to sign on his behalf.

That is, A chooses at random $k_A \in \mathbb{Z}_q^*$, and computes $r_A = g^{k_A}$ and $s_A = k_A + x_A H(M_\omega, r_A) \bmod q$. Signer A sends (M_ω, r_A, s_A) to a proxy signer B secretly (in fact, only the value s_A must remain secret, the values M_ω and r_A should be broadcast). Then B verifies the validity of the Schnorr's signature:

$$g^{s_A} = r_A y_A^{H(M_\omega, r_A)}$$

If the verification is correct, B computes his proxy key pair (x_P, y_P) as

$$x_P = x_B + s_A \ , \ y_P = g^{x_P} \ (= y_B r_A y_A^{H(M_\omega, r_A)})$$

Proxy signature generation: in order to create a proxy signature on a message M conforming to the warrant information M_ω, proxy signer B uses Schnorr's signature scheme with keys (x_P, y_P) and obtains a signature (r_P, s_P) for the message M. The valid proxy signature will be the tuple

$$(M, r_P, s_P, M_\omega, r_A)$$

Verification: a recipient can verify the validity of the proxy signature by checking that M conforms to M_ω and the verification equality of Schnorr's signature scheme with public key $y_A^{H(M_\omega, r_A)} r_A y_B$ $(= y_P)$; that is

$$g^{s_P} = r_P (y_B r_A y_A^{H(M_\omega, r_A)})^{H(M, r_P)}$$

This proxy signature scheme satisfies the security requirements (i), ..., (v) listed above (see [8] for the details). Note also that other signature schemes can be used in the proxy signature generation, with keys (x_P, y_P), provided that these schemes use keys of the form (x, y), with $y = g^x$; for example, ElGamal signature scheme or DSS.

3 Some Distributed Protocols in a General Framework

In [15], Stinson and Strobl propose a distributed version of Schnorr's signature scheme, which is proved to be as secure as the original signature scheme. This proposal is based on verifiable secret sharing schemes and on the joint generation of a random secret value.

We will consider a framework which is more general than the threshold one. That is, those subsets of players authorized to perform some specific actions, such as the recovery of a secret or the signature of a message, as well as those subsets of dishonest players that the system is able to tolerate, will not be necessarily defined according to their cardinality.

So we will adapt to this general framework the previous (threshold) proposals for verifiable secret sharing [10], the joint generation of a random secret [5] and threshold Schnorr's signature scheme [15].

3.1 Verifiable Secret Sharing

In a *secret sharing scheme*, a dealer distributes shares of a secret value among a set of players $\mathcal{P} = \{1, \ldots, n\}$ in such a way that only authorized subsets of players (those in the so-called *access structure*, denoted by $\Gamma \subset 2^{\mathcal{P}}$) can recover the secret value from their shares, whereas non-authorized subsets do not obtain any information about the secret (unconditional security). The structure Γ must be *monotone increasing*, that is, if $A_1 \in \Gamma$ and $A_1 \subset A_2$, then $A_2 \in \Gamma$.

Secret sharing schemes were introduced independently by Shamir [13] and Blakley [1] in 1979. Shamir proposed a well-known *threshold* scheme, in which the authorized subsets are those with more than t members (t is the threshold). Other works propose schemes realizing more general access structures; for example, *vector space secret sharing schemes* [2] are often used. An access structure Γ can be realized by such a scheme if, for some positive integer t and some vector space $E = K^t$ over a finite field K (in our context, it will be $K = \mathbb{Z}_q$), there exists a function

$$\psi : \mathcal{P} \cup \{D\} \longrightarrow E$$

such that $A \in \Gamma$ if and only if the vector $\psi(D)$ can be expressed as a linear combination of the vectors in the set $\psi(A) = \{\psi(i) | i \in A\}$. If Γ can be defined in this way, we say that Γ is a *vector space access structure*; then we can construct a secret sharing scheme for Γ with set of secrets $\mathbb{Z}_q$: given a secret value $k \in \mathbb{Z}_q$, the dealer takes a random element $\mathbf{v} \in E = (\mathbb{Z}_q)^t$, such that $\mathbf{v} \cdot \psi(D) = k$. The share of a participant $i \in \mathcal{P}$ is $s_i = \mathbf{v} \cdot \psi(i) \in \mathbb{Z}_q$. Let A be an authorized subset, $A \in \Gamma$; then, $\psi(D) = \sum_{i \in A} c_i^A \psi(i)$, for some $c_i^A \in \mathbb{Z}_q$. In order to recover the secret, the players of A compute

$$\sum_{i \in A} c_i^A s_i \;=\; \sum_{i \in A} c_i^A \mathbf{v} \cdot \psi(i) \;=\; \mathbf{v} \cdot \sum_{i \in A} c_i^A \psi(i) \;=\; \mathbf{v} \cdot \psi(D) \;=\; k \mod q$$

Shamir threshold secret sharing scheme with threshold t is a particular case of vector space schemes, taking $\psi(D) = (1, 0, \ldots, 0)$ and $\psi(i) = (1, i, i^2, \ldots, i^{t-1})$.

Linear secret sharing schemes can be seen as vector space secret sharing schemes in which each player can have associated more than one vector. They were introduced by Simmons, Jackson and Martin [14], who proved that any access structure can be realized by a linear secret sharing scheme, although in general the construction they proposed results in an inefficient secret sharing scheme. These schemes have been considered under other names such as geometric secret sharing schemes or monotone span programs. In our work, we will consider any possible access structure, so we will know that there exists a linear secret sharing scheme realizing this structure. However, we will suppose for simplicity that this scheme is a vector space one.

A variation of these schemes are *verifiable secret sharing schemes*, which prevent the dealer and the players from cheating; each participant can check if his share is consistent with the shared secret. The two most used verifiable secret sharing schemes are the proposals of Pedersen [10] and Feldman [3]. Here we present a modification of the (threshold) verifiable secret sharing scheme proposed in [10]. We consider any access structure Γ. Furthermore, we must take into account which subsets of dishonest players can be tolerated by the system. Those subsets form the *adversary structure* $\mathcal{A} \subset 2^{\mathcal{P}}$, which must be *monotone decreasing*: if $B_1 \in \mathcal{A}$ is tolerated and $B_2 \subset B_1$, then $B_2 \in \mathcal{A}$ is also tolerated.

The situation is modelized by an *active adversary* who can corrupt, at the beginning of the protocol, all players of some subset $R \in \mathcal{A}$. During the execution of the protocol, the adversary controls the behavior of these players, deciding at each moment which players of R follow the protocol correctly and which ones lie, but the adversary cannot change the subset R in $\mathcal{A}$ that he has chosen at the beginning (we say that it is a *static* adversary). An obvious requirement is that the adversary cannot obtain the secret from the shares of the participants that he has corrupted, so the condition $\Gamma \cap \mathcal{A} = \emptyset$ must be satisfied.

In the threshold case, the structures $\Gamma = \{A \in 2^{\mathcal{P}} \;:\; |A| \geq t\}$ and $\mathcal{A} = \{B \in 2^{\mathcal{P}} \;:\; |B| < t\}$ have been usually considered. We are going to consider any possible structures Γ and $\mathcal{A}$

satisfying $\Gamma \cap \mathcal{A} = \emptyset$, and so we will use general linear secret sharing schemes (for simplicity, vector space ones) instead of threshold secret sharing schemes.

As before, q and p are large primes such that $q|p-1$. Let g and h be generators of a multiplicative subgroup of $\mathbb{Z}_p^*$ with order q. The set of players is $\mathcal{P} = \{1, \ldots, n\}$, and the access structure $\Gamma \subset 2^{\mathcal{P}}$ is defined by the function $\psi : \mathcal{P} \cup \{D\} \longrightarrow (\mathbb{Z}_q)^t$. If the dealer wants to share the secret $k \in \mathbb{Z}_q$, in a verifiable way, he does the following:

1. Choose two random vectors in $(\mathbb{Z}_q)^t$:

$$\mathbf{v} = (v^{(1)}, \ldots, v^{(t)}) \quad , \quad \mathbf{w} = (w^{(1)}, \ldots, w^{(t)})$$

   such that $\mathbf{v} \cdot \psi(D) = k$.

2. Compute $(s_i, s'_i) = (\mathbf{v} \cdot \psi(i), \mathbf{w} \cdot \psi(i)) \in (\mathbb{Z}_q)^2$ and send the pair (s_i, s'_i) to player i, for $1 \leq i \leq n$.

3. Broadcast the public commitments $C_m = g^{v^{(m)}} h^{w^{(m)}} \in \mathbb{Z}_p^*$, for $1 \leq m \leq t$.

Each player i verifies that

$$g^{s_i} h^{s'_i} = \prod_{m=1}^{t} (C_m)^{\psi(i)^{(m)}} \tag{1}$$

where $\psi(i)^{(m)}$ denotes the m-th component of vector $\psi(i)$. If this equality does not hold, player i broadcasts a complaint against the dealer.

For each complaint from a player i, the dealer broadcasts the values $(s_i, s'_i) = (\mathbf{v} \cdot \psi(i), \mathbf{w} \cdot \psi(i))$ satisfying equation (1). The dealer is rejected if he receives complaints from players of a subset that is not in the adversary structure $\mathcal{A}$, or if he answers a complaint with values that do not satisfy equation (1). Otherwise, the dealer is accepted.

This verifiable secret sharing scheme is computationally secure, assuming that the discrete logarithm problem in the group generated by g is hard (the proof is almost the same as that in [10] for the threshold case).

3.2 Robust Joint Generation of a Random Secret Value

In this work, and roughly speaking, a distributed protocol is said to be *robust* if it always produces a correct output, even in the presence of some tolerated subset of dishonest players.

In [5] Gennaro, Jarecki, Krawczyk and Rabin use Pedersen's verifiable secret sharing scheme to design a protocol in which players in a set $\mathcal{P} = \{1, \ldots, n\}$ jointly generate a public key $y = g^x$ and shares of the corresponding secret key x, in such a way that t or more players can recover this secret key (threshold access structure). The idea is the following: each player i plays the role of a dealer and shares a random value k_i among the players. The secret key x will be the sum of some of these values.

We explain here the more general version considering any access structure $\Gamma \subset 2^{\mathcal{P}}$ (realizable, for simplicity, by a vector space scheme defined by a function ψ) and any adversary structure $\mathcal{A}$ satisfying some security and robustness conditions. If we want this protocol to be robust, we must make sure that, when we detect a dishonest subset of players in $\mathcal{A}$ and reject them from the protocol, an authorized subset in Γ still remains among the non-rejected players; this authorized subset of honest players can go on executing the protocol. That is, for any subset $R \in \mathcal{A}$, it must be $\mathcal{P} - R \in \Gamma$, or equivalently, $\mathcal{A}^c \subset \Gamma$, where $\mathcal{A}^c = \{\mathcal{P} - R \ : \ R \in \mathcal{A}\}$.

Combining this condition with the unforgeability condition $\Gamma \cap \mathcal{A} = \emptyset$, we have in particular that the structures $\mathcal{A}$ and Γ must satisfy the following condition: for all subset $R \in \mathcal{A}$ it is necessary $\mathcal{P} - R \notin \mathcal{A}$. We say that such a monotone decreasing structure $\mathcal{A}$ is $\mathcal{Q}^2$ in $\mathcal{P}$. Note that in the threshold case, this $\mathcal{Q}^2$ condition is equivalent to $n \geq 2t + 1$.

The protocol is as follows:

1. Each player i executes Pedersen's verifiable secret sharing scheme playing the role of a dealer. That is, he chooses two random vectors $\mathbf{v_i} = (v_i^{(1)}, \ldots, v_i^{(t)})$ and $\mathbf{w_i} = (w_i^{(1)}, \ldots, w_i^{(t)})$, in $(\mathbb{Z}_q)^t$, where $\mathbf{v_i} \cdot \psi(D) = k_i$ is the random secret distributed by player i, and sends to player j the pair $(s_{ij}, s'_{ij}) = (\mathbf{v_i} \cdot \psi(j), \mathbf{w_i} \cdot \psi(j))$, for $1 \leq j \leq n$. The public commitments are $C_{im} = g^{v_i^{(m)}} h^{w_i^{(m)}}$, for $1 \leq m \leq t$.

2. At step 1, players who cheat are detected and rejected. We define $F_0 = \{i|$ player i is not rejected at step $1\}$. Since $\mathcal{A}^c \subset \Gamma$, we have that $F_0 \in \Gamma$. Furthermore, for all players $i \in F_0$ that pass this phase, there are valid shares s_{ij} corresponding to players j that form an authorized subset. Each player $j \in \mathcal{P}$ computes his share of the total secret as $x_j = \sum_{i \in F_0} s_{ij}$ (the total secret will be $x = \sum_{i \in F_0} k_i \ \in \mathbb{Z}_q$).

3. Now they want to compute the value $y = g^x = \prod_{i \in F_0} g^{k_i} \in \mathbb{Z}_p^*$. They use Feldman's verifiable secret sharing scheme (see [3] for the original threshold version):

   3.1. Each player $i \in F_0$ broadcasts $A_{im} = g^{v_i^{(m)}}$, for $1 \leq m \leq t$.

   3.2. Each player j verifies the values broadcast by all the other players in F_0. That is, for each $i \in F_0$, player j checks that

   $$g^{s_{ij}} = \prod_{m=1}^{t} (A_{im})^{\psi(j)^{(m)}} \tag{2}$$

   If this verification is false, player j complains against i broadcasting the pair (s_{ij}, s'_{ij}) that satisfies verification at step 1 (Pedersen's scheme, equation (1) in Section 3.1), but does not satisfy equation (2).

   3.3. For players i who received some valid complaint at step 3.2, the other players j run the reconstruction phase of Pedersen's scheme to recover a vector $\tilde{\mathbf{v}}_\mathbf{i} = (\tilde{v}_i^{(1)}, \ldots, \tilde{v}_i^{(t)})$ such that $\tilde{\mathbf{v}}_\mathbf{i} \cdot \psi(j) = s_{ij}$, for all these players j (depending on the case, they will recover exactly $\tilde{\mathbf{v}}_\mathbf{i} = \mathbf{v_i}$, but this is not necessary). They can also recover the value k_i; this can be done because there are valid shares s_{ij} satisfying equation (1) at step 1 (Pedersen's scheme), corresponding to players j that form an authorized subset. All players in F_0 can compute, therefore, the correct value g^{k_i}. From the vector $\tilde{\mathbf{v}}_\mathbf{i}$, the correct commitment values $A_{im} = g^{\tilde{v}_i^{(m)}}$ can also be computed.

   Then the public key $y = g^x$ can be obtained by any participant in the following way:

$$y = \prod_{i \in F_0} g^{k_i} = \prod_{i \in F_0} g^{\mathbf{v_i} \cdot \psi(D)} = \prod_{i \in F_0} \prod_{m=1}^{t} g^{v_i^{(m)} \psi(D)^{(m)}} = \prod_{i \in F_0} \prod_{m=1}^{t} (A_{im})^{\psi(D)^{(m)}}$$

After the execution of this protocol, we have the public key $y = g^x$, where $x = \sum_{i \in F_0} k_i$ is the corresponding secret key, and $x_j = \sum_{i \in F_0} s_{ij} = (\sum_{i \in F_0} \mathbf{v_i}) \cdot \psi(j) = \mathbf{v} \cdot \psi(j)$ is the share of player j corresponding to the secret x, where $\mathbf{v} = (v^{(1)}, \ldots, v^{(t)})$, with $v^{(m)} = \sum_{i \in F_0} v_i^{(m)}$. Besides, the final commitment values $A_m = g^{v^{(m)}}$ can be easily computed as $A_m = \prod_{i \in F_0} A_{im}$, for $1 \leq m \leq t$.

We note all these facts (parameters and outputs of the protocol) with the following expression:

$$(x_1, \ldots, x_n) \quad \overset{(\mathcal{P}, \Gamma, \mathcal{A})}{\longleftrightarrow} \quad ((x, y), \{A_m\}_{1 \leq m \leq t}, F_0)$$

The security and robustness of this protocol can be proved analogously to the proof in [5] (which corresponds to the threshold case $n \geq 2t + 1$).

3.3 Stinson and Strobl Distributed Schnorr's Signature Scheme

Now we will explain the proposal of Stinson and Strobl [15] for distributing Schnorr's signature scheme. They consider threshold structures; that is, the system can tolerate the presence of less than t dishonest players, whereas any subset of at least t honest players can compute a valid signature. But they remark that the protocol can be adapted to run with other structures, using a general linear (verifiable) secret sharing scheme instead of the threshold secret sharing scheme (and its verifiable variants) of Shamir.

We now explain the scheme in [15] adapted to the case of any access structure Γ and adversary structure $\mathcal{A}$, such that $\Gamma \cap \mathcal{A} = \emptyset$ and $\mathcal{A}^c \subset \Gamma$ (the justification for these combinatorial requirements is the same as in Section 3.2). We assume again that Γ is a vector space access structure defined by a function ψ. The protocol has three parts.

Key generation: players in $\mathcal{P} = \{1, \ldots, n\}$ use the protocol explained in Section 3.2 to jointly generate shares of a secret key and the corresponding public key. The output will be:

$$(x_1, \ldots, x_n) \quad \overset{(\mathcal{P},\Gamma,\mathcal{A})}{\longleftrightarrow} \quad ((x, y), \{A_m\}_{1 \le m \le t}, F_0)$$

Signature generation: let H be a collision-free hash function, and M the message to be signed. If an authorized subset $F_1 \in \Gamma$, $F_1 \subset F_0$ wants to sign M, they do the following:

1. Players in F_1 run again the joint generation protocol of Section 3.2, with output

   $$(k_1, \ldots, k_n) \quad \overset{(\mathcal{P},\Gamma,\mathcal{A})}{\longleftrightarrow} \quad ((k, r), \{C_m\}_{1 \le m \le f}, F_2)$$

   where k is a random secret shared value in $\mathbb{Z}_q$ and $r = g^k$ is public, and $F_2 \subset F_1$.

2. Each player $i \in F_2$ broadcasts

   $$\gamma_i = k_i + H(M, r)x_i$$

3. Each player $j \in F_2$ verifies, for all $i \in F_2$, that

   $$g^{\gamma_i} \;=\; \prod_{m=1}^{t} (C_m)^{\psi(i)^{(m)}} [(A_m)^{\psi(i)^{(m)}}]^{H(M,r)}$$

   Define $F_3 = \{i |$ player i is not detected to be cheating at step $3\}$.

4. Each player $i \in F_3$ computes $s = k + H(M, r)x \mod q$, in the following way: since $\mathcal{A}^c \subset \Gamma$, we have that $F_3 \in \Gamma$, so there exist public coefficients $\{\lambda_j^{F_3}\}_{j \in F_3}$ in $\mathbb{Z}_q$ such that $\sum_{j \in F_3} \lambda_j^{F_3} \psi(j) = \psi(D)$. Then, each player $i \in F_3$ computes

   $$s = \sum_{j \in F_3} \lambda_j^{F_3} \gamma_j$$

The signature for the message M is the pair (r, s).

Verification: the verification phase is the same as in Schnorr's signature scheme; that is, the recipient cannot distinguish if the signature has been generated in a distributed way or not. The recipient checks that

$$g^s = r y^{H(M,r)}$$

Notation: we will use the expression

$$DistSchnSig(\mathcal{P}, \Gamma, \mathcal{A}, M, y, \{x_i\}_{i \in \mathcal{P}}, \{A_m\}_{1 \le m \le f}) \;=\; (r, s)$$

to refer to an execution of the signature generation phase, in which players of a set $\mathcal{P}$, with authorized subsets in the access structure Γ and tolerated subsets of dishonest players in the adversary structure $\mathcal{A}$, jointly generate a Schnorr's signature (r, s) on a message M, using the public key y, shares $(x_1, \ldots, x_n)$ of the secret key x, and commitment values $A_m = g^{v^{(m)}}$ for the components $v^{(m)}$ of the vector that in fact distributes the shares of x.

Security of the protocol. In [15], this distributed signature scheme is proved to be as secure as Schnorr's signature scheme. The idea of the proof is the following: they prove that the protocol is *simulatable*; that is, given an adversary against the scheme, there exists an algorithm which outputs values that are computationally indistinguishable from the values that the adversary views during a real execution of the protocol. Then, assuming that this adversary against the distributed scheme is successful in forging a signature under a chosen message attack, both this fact and the simulability of the distributed protocol can be used to construct an adversary against the original Schnorr's scheme, which is also successful in forging a signature under a chosen message attack. But in the random oracle model, this is equivalent to solving the discrete logarithm problem [11], so they can conclude that the distributed version of Schnorr's signature scheme has this same level of security, in the random oracle model (see [15] for the complete proof).

The protocol is also robust, if $\mathcal{A}^c \subset \Gamma$. This is due to the fact that there is always a subset in Γ that passes all the verification tests, and so players of this subset can finish the protocol correctly.

4 Fully Distributed Proxy Signatures

In this section, we propose a distributed proxy signature scheme based on the proxy signature scheme of Lee et al. [8] and on the idea of the distributed Schnorr's signature scheme of Stinson and Strobl [15], explained above.

Distributed protocols have two main advantages with respect to individual ones: an increase of the *security*, because now more than one party must be corrupted in order to obtain a secret key, for example; and an increase of the *reliability*, because the protocol can be executed even if some parties are non-working at that moment for some reason.

There are various proposals of distributed (threshold) proxy signature schemes. Zhang's proposal [16] is not strongly unforgeable, because the original signer can impersonate the proxy signer. Kim et al. [7] also proposed a threshold version of their proxy signature scheme. Hwang, Lin and Lu [6] adapt the threshold scheme of Kim et al. to the case in which the verifier of the proxy signature must be able to identify which concrete players in the proxy entity have signed the message. All these schemes distribute only the power of the proxy signer that signs messages on behalf of the original signer. Why not also distribute the original signer, and in this way increase the security and reliability of the full scheme?

Our proxy signature scheme is the first that is fully distributed, in the sense that we distribute both the original and the proxy signer. We consider general structures for the authorized subsets and for the tolerated subsets of dishonest players. Finally, our scheme is based on the proxy signature scheme of Lee et al. [8], and so the original signer entity does not need to include explicitly his identity, nor the identity of the proxy signer in the warrant information that it signs.

4.1 The Scenario

We must think of entities A and B as sets of players $A = \{P_1, \ldots, P_{n_A}\}$ and $B = \{Q_1, \ldots, Q_{n_B}\}$. We consider general monotone increasing access structures $\Gamma_A \subset 2^A$ and $\Gamma_B \subset 2^B$ in these sets. Furthermore, the system will tolerate the presence of some coalitions of dishonest players, those in the adversary structures $\mathcal{A}_A \subset 2^A$ and $\mathcal{A}_B \subset 2^B$, which must be monotone decreasing; that is, the scheme will be unforgeable even if some players in A and some players in B are corrupted and exchange their secret information, provided $\Gamma_A \cap \mathcal{A}_A = \emptyset$ and $\Gamma_B \cap \mathcal{A}_B = \emptyset$, of course. Finally, we require $\mathcal{A}_A^c \subset \Gamma_A$ and $\mathcal{A}_B^c \subset \Gamma_B$, in order to give robustness to the scheme, in the same way as in Sections 3.2 and 3.3.

We assume, for simplicity, that there exists a function $\psi_A : \{D\} \cup A \longrightarrow (\mathbb{Z}_q)^{t_A}$, for some positive integer t_A, such that a subset $J_A \subset A$ is in Γ_A if and only if $\psi_A(D) \in \langle \psi_A(j) \rangle_{P_j \in J_A}$, and the same for the structure Γ_B with a certain positive integer t_B and a certain function ψ_B.

Any subset of A whose honest players form a subset in Γ_A can delegate A's signing capability, and any subset of B whose honest players form a subset in Γ_B can sign a message on behalf of entity A.

4.2 Our proposal

The protocol that we present has four parts:

Generation of the entities' keys

Players in A jointly generate a public key and shares of the corresponding secret key, using the protocol in Section 3.2. Players in B do the same. The result is:

$$(x_{A,1}, \ldots, x_{A,n_A}) \overset{(A,\Gamma_A,\mathcal{A}_A)}{\longleftrightarrow} ((x_A, y_A), \{A_m\}_{1 \le m \le t_A}, F_{0,A})$$

$$(x_{B,1}, \ldots, x_{B,n_B}) \overset{(B,\Gamma_B,\mathcal{A}_B)}{\longleftrightarrow} ((x_B, y_B), \{B_\ell\}_{1 \le \ell \le t_B}, F_{0,B})$$

Distributed generation of the proxy key

In this phase, players in entity A sign a warrant information M_{ω_A}, using the first part of the distributed Schnorr's signature scheme explained in Section 3.3. However, they do not obtain the explicit signature, but shares of it (thus preventing the possibility of one dishonest participant in A sending this secret signature to a dishonest participant in entity B). Then they send some information to players in entity B. Each player in B then computes, from this information, his share of the proxy key, which will later be used to generate a proxy signature in a distributed way. This subprotocol is as follows.

1. Players in A execute the first step in the signature generation phase of the distributed Schnorr's signature scheme explained in Section 3.3. That is, they run the joint generation protocol of Section 3.2, with output

$$(k_{A,1}, \ldots, k_{A,n_A}) \overset{(A,\Gamma_A,\mathcal{A}_A)}{\longleftrightarrow} ((k_A, r_A), \{C_m\}_{1 \le m \le t_A}, F_{1,A})$$

   The values $r_A = g^{k_A}$ and M_{ω_A} are made public.

2. Each player $P_i \in F_{1,A}$ computes his share of the value $s_A = k_A + x_A H(M_{\omega_A}, r_A) \bmod q$ as

$$\gamma_i = k_{A,i} + H(M_{\omega_A}, r_A) x_{A,i} \mod q$$

3. Each player $P_i \in F_{1,A}$ distributes the value γ_i, verifiably among the players in entity B, in such a way that any subset in Γ_B can recover this value. He uses Feldman's scheme [3]; that is, P_i chooses a random vector $\mathbf{v_i} = (v_i^{(1)}, \ldots, v_i^{(t_B)})$ in $\mathbb{Z}_q^{t_B}$ such that $\mathbf{v_i} \cdot \psi_B(D) = \gamma_i$, he makes public the commitment values $D_{i\ell} = g^{v_i^{(\ell)}}$, for $1 \le \ell \le t_B$, and sends to each player $Q_j \in B$ the share $s_{ij} = \mathbf{v_i} \cdot \psi_B(Q_j)$.

4. In some way (we do not explain the details here), the correct commitments $\{A_m\}_{1 \le m \le t_A}$ and $\{C_m\}_{1 \le m \le t_A}$ corresponding to the sharing of the secret values x_A and k_A, respectively, must be publicly revealed to all players in entity B. Then each player $Q_j \in B$ checks, for any received share s_{ij}, that

$$\prod_{\ell=1}^{t_B} (D_{i\ell})^{\psi_B(D)^{(\ell)}} = \prod_{m=1}^{t_A} (C_m)^{\psi_A(P_i)^{(m)}} [(A_m)^{\psi_A(P_i)^{(m)}}]^{H(M_{\omega_A}, r_A)}$$

and that

$$g^{s_{ij}} = \prod_{\ell=1}^{t_B} (D_{i\ell})^{\psi_B(Q_j)^{(\ell)}}$$

If either of these two checks fails, Q_j broadcast a complaint against P_i. If P_i receives complaints from players that form a subset of B that is not in $\mathcal{A}_B$, then he is rejected. Let $F_{2,A}$ be the subset of players in A that pass this verification phase. Since $\mathcal{A}_A^c \subset \Gamma_A$, we have that $F_{2,A} \in \Gamma_A$.

5. Players of B publicly fix coefficients $\{\lambda_i^{F_{2,A}}\}_{P_i \in F_{2,A}}$ in $\mathbb{Z}_q$ such that $\psi_A(D) = \sum_{P_i \in F_{2,A}} \lambda_i^{F_{2,A}} \psi_A(P_i)$. Then the equality $\sum_{P_i \in F_{2,A}} \lambda_i^{F_{2,A}} \gamma_i = s_A$ holds, and each player $Q_j \in B$ uses these fixed coefficients to compute his share of the value s_A as

$$s_{A,j} = \sum_{P_i \in F_{2,A}} \lambda_i^{F_{2,A}} s_{ij} \mod q \ .$$

In effect, if $J_B \in \Gamma_B$, there exists coefficients $\{\lambda_j^{J_B}\}_{Q_j \in J_B}$ in $\mathbb{Z}_q$ such that $\psi_B(D) = \sum_{Q_j \in J_B} \lambda_j^{J_B} \psi_B(Q_j) \mod q$. Then it is not difficult to see that $\sum_{Q_j \in J_B} \lambda_j^{J_B} s_{A,j} = s_A \mod q$, and that $\{s_{A,j}\}_{Q_j \in B}$ is a perfect sharing of the secret s_A, according to the access structure Γ_B.

6. Each player $Q_j \in B$ computes $x_{P,j} = x_{B,j} + s_{A,j} \mod q$ as his share of the secret proxy key $x_P = x_B + s_A \mod q$. The public proxy key is computed as $y_P = g^{x_P} = y_B r_A y_A^{H(M_{\omega_A}, r_A)} \mod p$.

Note that the vector that in fact shares the secret value s_A among the participants of B is

$$\mathbf{v} = \sum_{P_i \in F_{2,A}} \lambda_i^{F_{2,A}} \mathbf{v_i} = (v^{(1)}, \ldots, v^{(t_B)}) \ \ ,$$

where $v^{(\ell)} = \sum_{P_i \in F_{2,A}} \lambda_i^{F_{2,A}} v_i^{(\ell)}$, for $1 \leq \ell \leq t_B$. Therefore, the commitment values V_ℓ corresponding to the components $v^{(\ell)}$ of this vector $\mathbf{v}$ can be publicly computed from the commitments $D_{i\ell}$ of the components $v_i^{(\ell)}$ of the vectors $\mathbf{v_i}$, for $P_i \in F_{2,A}$ as follows:

$$V_\ell = g^{v^{(\ell)}} = g^{\sum\limits_{P_i \in F_{2,A}} \lambda_i^{F_{2,A}} v_i^{(\ell)}} = \prod_{P_i \in F_{2,A}} (g^{v_i^{(\ell)}})^{\lambda_i^{F_{2,A}}} = \prod_{P_i \in F_{2,A}} (D_{i\ell})^{\lambda_i^{F_{2,A}}}$$

Finally, the commitments corresponding to the components of the vector that shares the secret proxy key $x_P = x_B + s_A \mod q$ will be $U_\ell = B_\ell V_\ell$, for $1 \leq \ell \leq t_B$.

Note also that another possible strategy is to have an authority that receives the shares γ_i from players in A, computes the secret value s_A from these shares, and redistributes shares of s_A among players in B. This solution reduces the total number of communications of the scheme, but it has some drawbacks: the authority must be fully trusted and reliable (opposite to the philosophy of this work), and a bottleneck in the system is possible.

Distributed generation of a proxy signature

If the players of entity B want to sign a message M conforming to M_{ω_A} on behalf of entity A, they execute

$$DistSchnSig(B, \Gamma_B, \mathcal{A}_B, M, y_P, \{x_{P,j}\}_{j \in B}, \{U_\ell\}_{1 \leq \ell \leq t_B}) \ = \ (r_P, s_P)$$

The proxy signature is the tuple $(M, r_P, s_P, M_{\omega_A}, r_A)$.

Verification

The recipient of a proxy signature can verify its validity by checking that

$$g^{s_P} = r_P (y_B r_A y_A^{H(M_{\omega_A}, r_A)})^{H(M, r_P)}$$

4.3 Security and Robustness of the Scheme

The security of our distributed proxy signature scheme stems from the security requirements that are satisfied by the proxy signature scheme of Lee et al. [8], and from the existential unforgeability of the distributed Schnorr's signature scheme under chosen message attacks, in the random oracle model [15]. Roughly speaking, if an algorithm could forge a new distributed proxy signature after some executions of our scheme (in which the forger algorithm *views* all the public information and the secret information of a tolerated subset of dishonest players), then we could construct from it another algorithm that would forge a distributed Schnorr's signature; and this is computationally infeasible, in the random oracle model.

Thus, if the conditions $\Gamma_A \cap \mathcal{A}_A = \emptyset$ and $\Gamma_B \cap \mathcal{A}_B = \emptyset$ hold, we can state that any subset of $\mathcal{A}_A$ does not obtain any information that allows it to delegate A's signing capability to a proxy entity; and any subset of $\mathcal{A}_B$ does not obtain any information that allows it to sign a message on behalf of an original signer entity A (strong distributed unforgeability). Moreover, the distributed proxy signature scheme satisfies the requirements of verifiability, strong identifiability, strong undeniability and prevention of misuse (see Section 2).

Steps 3 and 4 in the distributed proxy key generation phase are a variation of Feldman's verifiable secret sharing scheme (which is computationally secure, see [3]). In these steps, players in B detect dishonest players $P_i \in F_{1,A}$ who want to share an incorrect $\tilde{\gamma}_i$ among players in B or who want to give them shares $\tilde{s}_{ij}$ which are inconsistent with the correct γ_i.

Since we impose $\mathcal{A}_A^c \subset \Gamma_A$ and $\mathcal{A}_B^c \subset \Gamma_B$, the scheme is robust: an authorized subset always remains in the set of non rejected players and can execute each step of the protocol.

Note that, even in the case where the players of a subset $R_A \in \mathcal{A}_A$ and the players of a subset $R_B \in \mathcal{A}_B$ are corrupted at the same time by the same adversary, the scheme is unforgeable and robust.

5 Conclusion and Open Problems

In this paper we propose a secure and fully distributed proxy signature scheme. We consider a framework which is more general than the threshold one, in the sense that the authorized subsets and the tolerated subsets of dishonest players are not necessarily defined according to their cardinality. We state the combinatorial conditions that these structures must satisfy if we want our scheme to be unforgeable and robust. The scheme is based on the results of [8] and [15], and inherits its security from the security of these two previous works. All these properties, especially the fact that we distribute not only the power of the proxy signer, but also the original signer ability to delegate his signing capability, make our scheme more complete than the previous proposals of threshold proxy signature schemes ([16, 7, 6]).

Distributing protocols is a way of achieving security and reliability, so our scheme can be used in a framework in which entities wish to prevent external attacks or dishonest actions from their own members. For example, we might imagine a company in which a department wants to delegate its signing capability to a proxy department of the same company. These departments are formed by many members, and it is dangerous to give all the power of a department to a single member. Our work allows this company to be secure so there is no possibility of irregularity in the functioning of the company, even in the presence of some dishonest members in each department. Besides, we consider general access structures (not only the threshold ones) in the departments; that is, the members do not all have the same power or influence within the department. We also consider general adversary structures; that is, members do not all have the same susceptibility to be corrupted.

Some problems remain open in the area of proxy signatures. Up to now, all the proposed schemes are based on Schnorr's signature scheme; therefore the keys of all the users are in the same group and the security parameters must be the same for each user. This may sometimes be undesirable, so it would be very interesting to find proxy signature schemes based on other signature schemes in which this situation does not arise (for example, RSA); this would appear to be a hard problem to solve.

With respect to distributed proxy signature schemes, other signature schemes based on the discrete logarithm problem can be used, such as DSS [4]. But this scheme makes use of the called *problem of the multiplication*, which has an efficient solution only in the threshold case, if an active adversary is considered. So it will be very interesting to find a way of solving the problem of the multiplication in the case of more general structures.

Finally, the number of communications between the participants in our fully distributed scheme is quite large, but this fact is in part inherited from the cost of the joint generation of a random secret value. Furthermore, communications between entities A and B must be performed only once. However, perhaps other fully distributed proxy signature schemes can be designed to overcome this drawback.

References

[1] G.R. Blakley. Safeguarding cryptographic keys. *Proc. of the National Computer Conf., American Fed. of Information.* Processing Societies Proceedings **48** p. 313-317 (1979).

[2] E.F. Brickell. Some ideal secret sharing schemes. *J.Combin. Math. and Combin. Comput.* **9** p. 105-113 (1989).

[3] P. Feldman. A practical scheme for non-interactive verifiable secret sharing. *Proc. of the 28th IEEE Symp. on the Found. of Computer Science.* IEEE Press, p. 427-437 (1987).

[4] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin. Robust Threshold DSS Signatures. *Advances in Cryptology-Eurocrypt'96*, LNCS 1070, Springer-Verlag, p. 354-371 (1996).

[5] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Advances in Cryptology-Eurocrypt'99*, LNCS 1592, Springer-Verlag, p. 295-310 (1999).

[6] M. Hwang, I. Lin and E.J. Lu. A secure nonrepudiable threshold proxy signature scheme with known signers. *International Journal of Informatica*, vol. 11, no. 2, p. 1-8, (2000).

[7] S. Kim, S. Park and D. Won. Proxy signatures, revisited. *Proc. of International Conference on Information and Communications Security (ICISC'97)* p. 223-232 (1997).

[8] B. Lee, H. Kim and K. Kim. Strong proxy signature and its applications. *The 2001 Symposium on Cryptography and Information Security (SCIS 2001)* (2001).

[9] M. Mambo, K. Usuda and E. Okamoto. Proxy signatures: Delegation of the power to sign messages. *IEICE Trans. Fundamentals* Vol. E79-A, No. **9**, p. 1338-1353 (1996).

[10] T.P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. *Advances in Crypt.-CRYPTO'91*, LNCS 576, Springer-Verlag, p. 129-140 (1991).

[11] D. Pointcheval and J. Stern. Security proofs for signature schemes. *Advances in Cryptology-Eurocrypt'96*, LNCS 1070, Springer-Verlag, p. 387-398 (1996).

[12] C.P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology* Vol. **4**, p. 161-174 (1991).

[13] A. Shamir. How to share a secret. *Com. of the ACM* No. **22** p.612-613 (1979).

[14] G. J. Simmons, W. Jackson and K. Martin. The geometry of secret sharing schemes. *Bulletin of the ICA* **1** p.71-88 (1991).

[15] D.R. Stinson and R. Strobl. Provably secure distributed Schnorr signatures and a (t, n) threshold scheme for implicit certificates. *Sixth Australasian Conference on Information Security and Privacy (ACISP 2001)* LNCS 2119, Springer-Verlag, p. 417-434, (2001).

[16] K. Zhang. Threshold proxy signature scheme. *1997 Information Security Workshop, Japan* p. 191-197 (1997).