Coercion-Freeness in E-voting via Multi-Party Designated Verifier Schemes

Jérôme Dossogne[1], Frédéric Lafitte[2], Olivier Markowitch[1]

[1]Computer Science Department, Université Libre de Bruxelles,
Bld. du Triomphe – CP 212, 1050 Brussels, Belgium
{jdossogn | Olivier.Markowitch@ulb.ac.be}

[2]Department of Mathematics, Royal Military Academy,
30 Renaissancelaan, 1000 Brussels, Belgium
Frederic.Lafitte@rma.ac.be,

Abstract: In this paper we present how multi-party designated verifier signatures can be used as generic solution to provide coercion-freeness in electronic voting schemes. We illustrate the concept of multi-party designated verifier signatures with an enhanced version of Ghodosi and Pieprzyk [GP06]'s threshold signature scheme. The proposed scheme is efficient, secure, allows distributed computations of the signature on the ballot receipt, and can be parameterized to set a threshold on the number of required signers. The security of the designated verifier property is evaluated using the simulation paradigm [Gol00] based on the security analysis of [GHKR08]. Unlike previously provable schemes, ours is ideal, i.e. the bit-length of each secret key share is bounded by the bit-length of the RSA modulus.

1 Introduction

Electronic voting is now a reality for national ballots (e.g. during the 2003-2004 referenda in Switzerland, some voters near Geneva were able to cast binding vote electronically [Sen04]; in Estonia, in 2009 more than 100 000 people voted through Internet for the local municipal elections; and the Estonian Parliament has recently opened the door for mobile phones to be used to authenticate voters in its 2011 election [Ric]), companies (e.g. it is common in shareholder elections in the United States to allow most voters to cast ballots via a web browser [Pro]), universities (e.g. to elect student representatives [Ass09]). Internet-based voting is a broadening trend [WV10]. The existing mechanisms of e-voting take different forms, from automated voting system to voting through networks. Recurring arguments are that electronic voting encourages a higher voter turnout and should make the counting of the ballots faster and more accurate. Whether using such technology in those contexts is a good choice or not is out of the scope of this paper. However, it is certain that electronic voting is a reality nowadays. Therefore, it is now mandatory to propose and to implement the technology to support essential e-voting systems requirements. For example, several properties are mandatory for a useful electronic voting system, such as ensuring the robustness of the system, the verifiability (i.e. ballots are published on a public bulletin board in a way that allow voters to verify the result of the election process), the anonymity of the voter, and

being coercion-free (e.g. Voteauction offered US citizens the chance to sell their presidential vote to the highest bidder during the Presidential Elections 2000, Al Gore vs. G.W. Bush [BKS+]). A number of contributions have described different ways to achieve robustness and verifiable electronic voting [DM10]. Problems arise when trying to combine voters' privacy with the ability for voters to check the correctness of their own votes by means of a receipt. Indeed, on the basis of such a receipt, a dishonest third-party could possibly force or encourage a voter to reveal his vote.

To avoid this weakness, some solutions [LK00] propose receipt-free voting protocols, but they are not problem-free. Some of these protocols can prevent the voters from being able to check whether their votes were counted, or they make it near impossible to report problems using evidence of the vote. Several schemes have been proposed to manage this problem, either by assuming that the voters must simply trust the polling office to behave honestly [LK00] or by paying more for data transmissions and computations overheads [HS00].

In a recent work, Juels et al. [JCJ05] and Backes [BHM08] present four different properties related to coercion resistance: receipt-freeness, immunity to simulation attacks, immunity to forced-abstention attacks, and immunity to randomization attacks. Essentially, coercion-freeness states that a coercer cannot force a voter to cast a certain vote or provide a receipt that would certify her vote. Intuitively, a protocol guarantees receipt-freeness if a voter does not gain any information that can be used to prove to a coercer that she voted in a certain way.

In this paper, while we intend to provide the voter with a receipt, we respect these four properties related to coercion resistance. However, our aim is to provide a receipt to the voter that he could use in court in case of conflict with the polling office. Nevertheless, we provide also the voter with the means to create his own receipts that are indistinguishable from a genuine receipt for an attacker but that cannot be used in a court since only the judge can distinguish between a valid receipt and one forged by the user.

The use of designated verifier signatures (DVS) by the polling office to sign the receipt, with the voter as designated verifier, is suitable to achieve such a feature [DM09a, DM09b, OMD04]. Jakobsson, Sako, Impagliazzo [JSI96] and Chaum [Cha96] introduced the notion of designated verifier signatures in order to strengthen the concept of undeniable signatures in Chaum and van Antwerpen [CV90]; their particular aim was to prevent blackmailing and mafia attacks [DGB87]. A valid designated verifier signature is such that it convinces only a specified recipient, while other entities would not be able to deduce anything about the validity of the presented signature. This can be achieved if the designated verifier of a signature s is able to produce a signature s′ intended for himself that is indistinguishable from s.

Furthermore, DVS can be generalized to allow multiple verifiers and are called Multi-DVS (MDVS) in such cases [SHCL08]. MDVS can be created based on ring signatures [LV04]; without encryption, based on [BGLS03]'s pairing-based ring signature [Lag07]; and on identity, based on [Cho08] a multi-signature extension of Hess's ID-based signature [Hes02] and Schnorr signature. MDVS suits e-voting very well since both the voter and a judge should be able to verify a signature created on a receipt at a polling office.
Multi-signer DVS (MSDVS) and their strong version MSSDVS [ZZZ08] are respectively a form of DVS where multiple signers are involved for a single designated verifier.

1.1 Our contribution

The aim of this paper is to introduce voting schemes in which each voter receives a receipt of his vote that cannot be used to reveal the vote to anyone except a judge. Therefore, such voting schemes, while they deter a coercer who might want to buy the votes, should allow the voters to verify his or her own vote but also to complain if necessary.

We propose a generic solution that relies on (w - 1,w)-threshold signature schemes and that allows coercion-freeness. Introduced in 1987 by Desmedt [Des88], a (t,w)-threshold signature scheme is a signature scheme where at least t participants out of w chosen entities have to cooperate using their own share of a common secret key in order to produce a valid signature. An attractive feature of most threshold schemes is that the shared key does not have to be known or reconstructed by the participants to produce the signature. Furthermore, there is no constraint on the number of participants that is needed in the verification process; therefore anyone should be able to verify the validity of the signature.

Based on a (w - 1,w)-threshold signature scheme, since any set of w - 1 out of the w participants can produce the signature, schemes can be created so that no one can deduce which one of the w - 1 participants participated in the signature generation. Hence all of the w participants can simultaneously deny their own implication in the signature generation. In such cases, everyone knows that only one of them would be honest when denying his or her implication; this provides us with the desired ambiguity.

Our objective, called source hiding and defined in [Lag07], is to transmit a receipt, r, for a ballot, b, from the polling office, P, to the voter, V, who cast b, that cannot be used by an attacker, A, to figure out the true content of b. We achieve this by creating a signature σ that can be produced either by P or by V, therefore, A can be sure that V did not create r to protect himself from A's coercion. At the same time, we want V to be able to ask a judge, J, to help him in case P did try to cheat him. This can only be achieved if r can serve as evidence for J, i.e. J can distinguish whether r was created by P or by V. In our construction, this is achieved by asking J to contribute to the signature creation, thus J would know whether the signature was created by V or by P.

MDVS is defined by [LSMP07] as a generic term for VS where "the signature is intended for n verifiers, n > 1". MSSDVS [ZZZ08], on the other hand, are DVS where multiple signers are involved. Since our construction's intent and purpose is to consider implicitly the signer J as verifier as well as V, and since both J and P are signers, it respects both properties based on those definitions1. [ZZZ08] illustrate the definition with a scheme based on bilinear pairing, whereas we will present a scheme based on RSA-PFDH [Cor02]. To avoid possible confusion with MDVS and MSDVS, we introduce the idea of multi-party designated verifier signatures (MPDVS).

Intuitively, we define tripartite multi-party designated verifier signatures in the following way: let P(A,B,C) be a protocol for Alice (A) to prove, with the help of Colin (C), the truth of the statement Ω to Bob (B). We say that Bob is a multi-party designated verifier if he can produce, with the help of Colin, identically distributed transcripts that are indistinguishable from those of P(A,B,C). This definition can be generalised to the multi-party case if we consider Colin as a set of co-signers called witnesses.

Multi-party designated verifier signatures are well suited for electronic voting schemes since those schemes can require an adjudicator to solve conflicts between the voter and the polling office and, as such, are tripartite by nature. If a voter systematically produces the indistinguishable transcripts every time he votes, an attacker who intercepts him after the voting procedure would not be able to know which of the receipts is the one corresponding to the real vote.

We illustrate our solution with an efficient, flexible multi-party designated verifier signature that is based on the threshold signature scheme of Ghodosi and Pieprzyk [GP06] and chosen for its simplicity and efficiency. We enhanced the scheme to make its security provable in the standard model while remaining ideal, i.e., the shared signing key's size is bounded by the size of an RSA modulus. At the same time, the proposed design facilitates distributed implementations of the computations and sets a threshold on the number of required signers.

The paper is organised as follows: In section 2 we present the notations, the adversarial model, and the security requirement for MPDVS schemes. In section 3 we describe an ideal and secure threshold RSA-PFDH signature scheme and use it to create a MPDVS scheme suitable for e-voting. In section 4 we analyse the security of that MPDVS and of the underlying threshold signature scheme. We conclude in section 5.

[1] The way Multi-DVS are defined and formalised imposes that "the participants …have to generate a shared RSA key"[LV04], "in identity-based cryptosystem, it also produces a master secret key (MSK), kept in secret by PKG (private key generator)"[Cho08]. This is not required in our primitive.

2 Model

2.1 Notation

The set of *w* participants (users) is denoted by $U = \{u_1,\ldots,u_w\}$, where

u_1 is the polling office
u_2 is the voter
$u_3, \ldots, u_w$ are the witnesses

We also consider a trusted key generation server, denoted KGS. $A_u(x) = y$ means that the randomized algorithm A is run by user $u \in U \cup \{KGS\}$ and produces the output $y \in \{0,1\}^*$ on input $x \in \{0,1\}^*$.

$S \subset U$ is the set of signers. We define $S_i \stackrel{def}{=} U \setminus \{u_i\}$ as the set of users that signs a message for the designated verifier u_i. In particular, we use the sets S_1 and S_2.

We write "$u_i \rightarrow u_j : m$" to denote that message m is sent from u_i to u_j via an authentic channel (tamper-resistant and authenticated).

$\sigma_{m,i}$ denotes the (partial) signature of user i on message m, $m_1|m_2$ is the concatenation of m_1 and m_2, $|m|$ is the bit-length of m and $m_1 \oplus m_2$ is the result of a bitwise XOR (exclusive disjunction) between m_1 and m_2.

Finally, since in our case $\sigma_{m,S1} = \sigma_{m,S2}$, indicating which S did sign is irrelevant, therefore we use σ_m to denote the usual RSA signature on message m. That is, $\sigma_m = m^d \bmod n$ where $ed = 1 \bmod \varphi(n)$ and $n = pq$. The prime numbers p,q are such that both their bit-lengths are approximately equal to the security parameter η.

2.2 Generic Description of MPDVS Schemes

A DVS scheme in which u_1 issues a signature for the designated verifier u_2 with help from witnesses $W = \{u_3,\ldots,u_w\}$ is defined as a set of five probabilistic polynomial time algorithms:
$\text{Setup}_{KGS}(\eta)$: Inputting security parameter η generates a master public key (MPK) and a master secret key (MSK). The MPK is transmitted to each user $u_i \in U$.

$\text{KeyGen}_{KGS}(\text{MPK},\text{MSK})$: Using the master parameters, this algorithm generates the pair (vk_i,sk_i) for each participant $u_i \in U$ with vk_i as the public verification key and sk_i as the secret signing key.

$Sign_{u1,W}(m,sk_1,sk_3,\ldots,sk_w)$: This is a distributed process where u_1 and $W = \{u_3 \ldots u_w\}$ collaborate in order to sign message m for the designated verifier u_2.

$Sim_{u2,W}(m,sk_2,sk_3,\ldots,sk_w)$: This is a distributed process where u_2 and $W = \{u_3 \ldots u_w\}$ collaborate in order to sign message m for the designated verifier u_1. This algorithm generates a dummy signature that is indistinguishable from the signature returned by algorithm *Sign*.

$Vrfy(\sigma_m,m,MPK)$: Anyone can use this algorithm to check whether σ_m is a valid signature on m.

2.3 Security Requirements

The polling office u_1 signs the ballot sent by the voter u_2 with witnesses $u_3 \ldots, u_w$. This signature is like a receipt that all users can verify but that is only convincing to the voter (designated verifier): his ability to produce the same receipt makes it unconvincing for users that did not participate in the protocol.

Let's consider an active adversary who, before the execution of the protocol, is able to corrupt a fixed subset of at most $k < t$ users. By corrupting user u_i, the adversary learns the secret key sk_i.

The security definitions we use are taken from [LWB05] and adapted to our multi-party setting. DVS schemes are required to satisfy unforgeability and non-transferability as defined below:

- **Unforgeability:** If a signature is valid, then either u_1 or u_2 participated in its computation. This means that the threshold t must higher than the number of witnesses, otherwise the witnesses alone would be able to forge a signature.

- **Non-transferability:** When given a valid signature σ_m, it is infeasible to tell which users participated in its computation. In particular, it is infeasible to tell whether u_1 or u_2 participated.

In addition to these two properties, [LWB05] observes that some DVS schemes have the property of delegatability, which can lead to undesired situations for some applications. According to [LWB05], a DVS scheme is delegatable if the signer is able to reveal information other than her secret key (a function of that secret $y = f_i(sk_i) \neq sk_i$) that allows the attacker to produce a valid signature with regard to a single designated verifier. According to this definition, our scheme is non-delegatable. Indeed, the only information that the signer u_i could reveal, and that would allow the attacker to create such a signature, is her secret key sk_i. In this case, and contrary to [LWB05], non-delegatability follows from unforgeability.

3 Multi-party Designated Verifier Signature Scheme

3.1 The Ideal and Secure (t,w)-threshold RSA-PFDH Scheme

Our designated verifier scheme is based on Ghodosi and Pieprzyk's threshold signature scheme [GP06], which itself relies on Shamir's threshold cryptosystem [Sha79]. We adapted the scheme in order to provide a security analysis as strong as [Sho00, GHKR08], which is stronger than [GP06]. However, we maintain the same performance. Essentially, when creating shares of the secret d, our scheme uses y, a prime number close to n, as a modulus, whereas [GP06]'s scheme uses n. Also, instead of using basic RSA [Cor01], we use RSA-PFDH [Cor02], i.e., the signature is not computed based on the original message *msg* but on $m = H(r|msg)$ where H is collision-resistant one-way hash function and r a random value of B bits[2].

The scheme considers an RSA secret key d that is shared between $w > 2$ potential signers, whereas the corresponding RSA public key (e,n) remains private. See [Ber08] for various optimizations and recommendations regarding the choice of the parameters when implementing.
Each participant receives one share such that,

- any set of t - 1 < w shares or less, reveal no information about the secret d
- any set of t shares allows for the efficient reconstruction of d

This method, based on polynomial interpolation, is rather simple. Given any field K, a polynomial f(x) ∈ K[x] is chosen at random with a degree t - 1 and a constant term d. Next, each user $i \in U$ receives f(i) ∈ K as a share. Since each user knows a point in the polynomial, any of t users can interpolate f(x) and thus recover the secret d = f(0).

In more detail, our scheme uses the field $\mathbb{Z}_y$, with y being the closest prime to n such that $\varphi(n) < y$. Coefficients $a_1,\ldots,a_{t-1}$ are chosen randomly in $\mathbb{Z}_y$ ($a_{t-1} \neq 0$), which yields the polynomial

$$f(x) = d + \sum_{j=1}^{t-1} a_j x^j \bmod y \qquad (1)$$

If each user has an integer $i \in U$ as his or her identity and receives the share f(i) mod y, then given any number of t points $S = \{i_1,\ldots,i_t\}$, the polynomial f(x) can be interpolated based on its Lagrange form:

$$f(x) = \sum_{j=1}^{t} L_S(x, i_j) f(i_j) \bmod y \qquad (2)$$

[2] Again, see [Ber08] for the importance of H, r, and B. For instance, H prevents existential forgery and "large choices of B are often conjectured to make non-generic attacks, attacks that pay attention to the hash function H, more difficult"[Ber08]. However, none of the two enlarge the original message (msg) space and thus neither diminishes the success rate of exhaustive search.

where the Lagrange coefficients $L_S(\cdot,\cdot)$ are given by

$$L_S(\alpha,\beta) = \prod_{\gamma \in S \setminus \{\beta\}} \frac{\alpha-\gamma}{\beta-\gamma} \bmod y \qquad (3)$$

Now, each participant owns a share f(i) mod y and outputs the partial signature

$$\sigma_{m,i} = m^{f(i) \bmod y} \bmod n \qquad (4)$$

Then the altered signature $\sigma'_{m,S} = m^{d+k_S y}$ is computed by combining the partial signatures:

$$\sigma'_{m,S} = \prod_{i \in S} \sigma_{m,i}^{L_S(0,i)} \bmod n \qquad (5)$$

the RSA signature can then be obtained by removing the term $k_S y$ in the exponent of $\sigma'_{m,S}$:

$$\sigma = \sigma'_{m,S} m^{k_S y} \bmod n \qquad (6)$$

with a pre-computed $k_S = \left(d - \sum_{i \in S} L_S(0,i) f(i)\right) / y$.

3.2 The (w - 1, w)-threshold scheme

There are three types of participants: (1) The designated verifier, (2) the signer, and (3) the contributors and witnesses to the signature creation. Both the signer and the contributors will be creating a signature that the designated verifier will be able to verify. Applied to electronic voting, these participants are respectively the voter (u_2), the polling office (u_1), and the adjudicators/witnesses ($u_3,\ldots,u_w$). The witnesses are the contributors. They are trusted to cooperate with the signer (u_1 or u_2) by signing the messages they receive and by keeping their own private signing key secret.

In [GP06] the secret key would be split twice, once for each possible set of w - 1 signatories. In our scheme, the secret key is split once into w shares. k_{Sz} is computed twice, once for each set S_z with $z \in \{1,2\}$[3], where S_z denotes a set of w - 1 signatories. S_1 is the set of signatories containing the voter and all the witnesses, and S_2 is the set of signatories containing the polling office and all the witnesses. The explanations for f(x), the shares f(i), k_{S1}, and k_{S2} can be found in section 3.1.

[3] If w = 3, it is possible to imagine $z \in \{1, 2, 3\}$ since *V and P* can generate a signature without the help of the only *W*. However, this seems to have no useful application in the case of electronic voting since their interests are opposite.

It is of course possible to compute k_{Si} for each of the w subsets of w - 1 participants (out of the w potential participants), but it seems of no use when applied to e-voting, since all the other subsets would ask both the voter and the polling office to contribute to the signature. This would not contribute to the signer ambiguity concerning the two parties since both would be required to co-sign.

3.3 Instantiation of the Model

Setup $_{KGS}(\eta)$: Entering the security parameter η will generate RSA parameters MPK = (n,e,y), MSK = d.

KeyGen $_{KGS}$(MPK,MSK) : based on the RSA parameters, transmit the pair of keys (vk_i,sk_i) to user u_i where

$$vk_i = (n, e, y) \quad \forall i \in \{1, \ldots, w\}$$

$$sk_i = \begin{cases} (f(1), k_{S_2}) & \text{if } i = 1 \\ (f(2), k_{S_1}) & \text{if } i = 2 \\ f(i) & \text{if } i \notin \{1, 2\} \end{cases}$$

$Sign_{u1,W}(m,sk_1,sk_3,\ldots,sk_w)$: This is a distributed process where u_1 and $W = \{u_3 \ldots u_w\}$ collaborate in order to sign message m for the designated verifier u_2:

1. $u_1 \rightarrow u_j$: m, with $j \in \{3,\ldots,w\}$
2. $u_j \rightarrow u_1$: $\sigma_{m,uj} = m^{sk}{}_j \pmod n$ with $j \in \{3,\ldots,w\}$
3. u_1 computes $\sigma'_{m,S2} = m^{f(1)} \cdot \prod_{j=3}^{w} \sigma_{m,uj} = \sigma m^{k}{}_{S2}{}^{y} \pmod n$
4. u_1 issues signature $\sigma = \sigma'_{m,S2} m^{-k}{}_{S2}{}^{y} \pmod n$

$Sim_{u2,W}(m,sk_2,\ldots,sk_w)$: This algorithm generates a dummy signature that is indistinguishable from (in this case, identical to) the original signature returned by the algorithm *Sign*.

1. $u_2 \rightarrow u_j$: m, with $j \in \{3,\ldots,w\}$
2. $u_j \rightarrow u_2$: $\sigma_{m,uj} = m^{sk}{}_j \pmod n$ with $j \in \{3,\ldots,w\}$
3. u_2 computes $\sigma'_{m,S1} = m^{f(2)} \cdot \prod_{j=3}^{w} \sigma_{m,uj} = \sigma m^{k}{}_{S1}{}^{y} \pmod n$
4. u_2 issues signature $\sigma = \sigma'_{m,S1} m^{-k}{}_{S1}{}^{y} \pmod n$

Vrfy(σ,m,mpk) Anybody can use this algorithm to check whether σ is a valid signature on m, i.e. whether $\sigma^e = m \bmod n$.

3.4 Efficiency

This scheme is ideal. The signing-key size is bounded by the size of an RSA modulus. The signature's size is independent of the number of verifiers. In addition to the computation of a classical RSA signature by each participant, combining the w - 1 partial signatures requires only w - 1 modular multiplications. The verification process remains the same as a classical RSA-PFDH signature verification.

With y^+ and y^- as the closest prime integers to n such that $\varphi(n) < y^- < n < y^+$, if $y = y^-$ then the scheme is ideal, since each $|sk_i|$ is smaller or equal to $|n|$. However, since we know that $\varphi(n) < y^-$, this reveals some information on $\varphi(n)$. This loss of security could be avoided by choosing $y = y^+$ which produces a scheme very close to the ideal but could prevent the use of existing implementations with a fixed size for the integers.

When considering [LSMP07]'s definition of strength, where a DVS is strong if the secret key of the designated verifier is required to execute the verification algorithm, it follows that creating an MPSDVS from this threshold scheme is trivial. Indeed, the key e does not have to be public but could very well be distributed only to the designated verifier as part of his secret key. By doing so, only the designated verifier would be able to verify the designated signature using his secret key as an input to the verification algorithm.

3.5 Confidentiality

The purpose of a digital signature is not to provide confidentiality on the signed message, i.e., the purpose is not to prevent someone from recovering the message from the signature. However, this still looks like a desirable trait with regard to the witnesses and of course an external attacker.
As mentioned in section 3.1, m = H(r|msg). However a small message space could allow an adversary to perform an exhaustive search in order to determine the value of msg. In such a case, the issuer could choose m = H(r ⊕ msg) where |r| is kept secret by the issuer and is long enough to prevent such a brute force attack (possibly $|r| \gg |msg|$). The issuer also has to commit to this value by publishing H(r).
While r is revealed to W in case of conflict with the polling office, it does not leak any useful information since msg would be revealed at the same time.

4 Security

The signature-hiding property requires that the signature issued by the set of signers S_1 is indistinguishable from the signature issued by the set of signers S_2. In our case, this property is achieved since it holds that $\sigma_{m,S1} = \sigma_{m,S2} = \sigma_m$.

This section focuses on the unforgeability of the signature. The analysis is based on the simulation proof in [GHKR08].

4.1 Security against an external opponent

Let's imagine that an adversary corrupts a set of k participants, denoted $B = \{u_{i1},\ldots,u_{ik}\} \subset U$, learning all their secret information but unable to control their behaviour. That is, all users are assumed to follow the protocol.

By corrupting both u_1 and u_2, the adversary would learn both k_{S1} and k_{S2}. These values give no more information about d when taken together than when taken separately. Moreover, given our application to voting, if an attacker corrupts both the voter and the polling office, then there is little interest in securing the protocol. Therefore, the unforgeability of our scheme depends only on the security of the underlying threshold signature scheme.

As in [GHKR08], we show that the adversary, in a chosen message scenario, is unable to gain more information about the missing share than the information given by the signature σ_m itself. For this, we describe a simulator that, given only what the adversary knows, is able to generate a view of the protocol that is indistinguishable from the actual view.

Unlike previous schemes (e.g. [GHKR08, Sho00]), the Lagrange coefficients involved in our protocol can be directly evaluated, since they are computed over the field $\mathbb{Z}_y$. This makes the simulation proof much easier.

Given the simulated shares $f(i_1),\ldots,f(i_k)$ and the final signature σ_m, the simulator can directly generate a value for the missing partial signature $\sigma_{m,k+1}$ that satisfies equations (5) and (6). This can be done by interpolating $f(i_{k+1})$ in the exponent, based on the set of points $\tilde{B} = \{0,i_1,\ldots,i_k\}$, since the signature σ_m can be seen as the "partial signature" $m^{f(0)}$ of "user" 0:

$$\begin{aligned}\sigma_{m,i_{k+1}} &= m^{\sum_{j\in\tilde{B}} L_S(i_{k+1},j)f(j)} \\ &= m^{L_S(i_{k+1},0)f(0)} \prod_{j\in\tilde{B}\setminus\{0\}} m^{L_S(i_{k+1},j)f(j)} \\ &= \sigma_m^{L_S(i_{k+1},0)} \prod_{j\in\tilde{B}\setminus\{0\}} m^{L_S(i_{k+1},j)f(j)}\end{aligned}$$

The term $m^{-k_{Si}y}$, $i \in \{1,2\}$, which is required to satisfy equation (6), is simply obtained by dividing σ_m through σ'_m: $m^{-k_{S_i}y} = \sigma_m/\sigma'_m = m^d/m^{d+k_{S_i}y} \bmod n$

Therefore, the adversary is unable to gain the information about the share of the honest user necessary to forge the signature of a previously unsigned message.

4.2 Security against a dishonest participant

Even if corrupted participants do not follow the protocol, the scheme is still required to be robust. Unlike the previous subsection, this analysis takes into account the application to voting, where a distinction is made between participants according to their roles.

Dishonest Dealer

A dishonest dealer can distribute bogus shares of the key, resulting in a failure of the signature process. Moreover, the dealer could claim that the problem is due to a dishonest participant.

Protection against a dishonest dealer can also be achieved using the partial signature verification scheme described in [GRJK07], in which the dealer is required to publish the values $g^d, g^{a_1}, \ldots, g^{a_k}$ where $g \in_R \mathbb{Z}_n^*$ has a high order and $a_1, \ldots, a_k$ are the coefficients of polynomial f. Thus, participant u_i can make sure the received share f(i) is correct by verifying that

$$g^{f(i)} = g^d \prod_{j=1}^{k} g^{a_j i^j} \bmod n$$

Dishonest signers

Dishonest witnesses that output incorrect partial signatures can be detected using the verification scheme of [GRJK07]. The users are required to output the verification value gf(i) together with their partial signature σmf(i). In order to verify that the partial signature is correct, ui is asked to return xf(i) from the input x = gamb where a and b are chosen at random. Then one is able to verify that the following equality holds.

$$x^{f(i)} = \left(g^{f(i)}\right)^a \sigma_{m,i}^b \bmod n$$

It might happen that the polling office refuses to transmit the signature σ_m in exchange for the voter's ballot. It is shown in [PG99] that this problem of fair exchange cannot be solved without including an additional trusted party.

Regarding forced abstention attacks, note that in the complete scheme, a single corrupt witness should not be able to reveal whether or not a voter voted. The easiest approach would be to associate the secret share with an anonymous identity (by the use of credentials [JCJ05]) instead of the voter's real identity.

Finally, notice that the witnesses could be selected so that they have highly conflicting interests to decrease the likelihood that a coalition could form. For instance, a council involving all parties and members of the voting community (even including voters[4]) could be chosen to form the set of witnesses. With the possibility to detect malicious behavior as discussed above, it is less likely that a party would run the risk of deviating from the protocol's instructions.

5 Conclusion

The contributions of this work are threefold.

First, we showed how to provide coercion freeness from any MSDVS in e-voting (including MSSDVS, MPDVS and MPSDVS) by using them to sign the receipt created to provide verifiability.

Second, we described how to create a MPDVS and MPSDVS from any (t,w)-threshold signature by instantiating the scheme as a (w - 1,w)-threshold one.

Finally, we proposed a secure and ideal threshold RSA signature by enhancing [GP06]'s scheme and proving its security under standard assumption with a proof inspired by [Sho00, GHKR08]'s security proof. Although the scheme is ideal, due to its threshold nature, it implies an unavoidable cost in communications.

By doing so, we present a generic solution that helps create coercion-freeness in electronic voting schemes based on threshold signature schemes. We illustrate our point with an efficient, ideal, and secure threshold scheme. Compared to previous proposals, our scheme is both secure and efficient. It also leads to an easy distribution of the computations, since the partial signatures can be computed simultaneously by each participant. The scheme requires the participation of a (set of) contributor(s) to generate the desired signatures. In the framework of electronic voting, the contributor is a set of witnesses/adjudicators who help settle the possible conflicts that can occur between the polling office and the voter. Therefore, if the receipt or the signature provided by the polling office is incorrect, the voter contacts the adjudicator (the contributor) and collaborates with him or her to verify the validity of the signature together. If it appears that the voter is honest, the adjudicator can contact the polling office to resolve the problem using legal procedures when appropriate.

The number of witnesses, t- 1, can be adjusted to decrease the required trust in each of them, i.e., more distinct witnesses, each selected for their conflicting interest with the others, would have to collaborate to cheat.

[4] To reach such a high level of citizen participation, a good idea might be to divide the census in constituencies where each voter is a witness for the rest of the constituency or, as we prefer, to allow citizen to participate but to choose randomly for which constituency he will be allowed to be witness.

The scheme we present can easily be used in existing protocols based on RSA signatures in order to convert these signatures into multi-party designated verifier signatures (the existing keys can be reused as well as most of the existing software.) The scheme is being implemented in conjunction with other Internet voting and security enhancement techniques and methodology [DM11] such as Mental Booths [DL11], TreeCounting [DM10], credentials [JCJ05], or re-encryption mixnets with randomized partial checking [CH11] to provide, resistance against side-channel attacks, over-the-shoulder coercion-resistance, practical verifiability, and anonymity respectively. The implementation is available on the author's website.

Bibliography

[Ass09] Assemblée Générale des étudiants de Louvain : Election étudiante à l'ULC : une première en Belgique, 2009.

[Ber08] Bernstein, D.: RSA signatures and Rabin-Williams signatures: the state of the art, 2008.

[BG02] Boneh D.; Golle P.: Almost entirely correct mixing with applications to voting. Proc.CCS '02, pp. 68–77, 2002. ACM Press.

[BGLS03] Boneh D. et al.: Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In EUROCRYPT, pp. 416–432, 2003.

[BHM08] Backes, M.; Hritcu, C.; Maffei, M: Automated verification of remote electronic voting protocols in the applied pi-calculus. 21th IEEE Symposium on Computer Security, pp. 195–209, 2008. IEEE Computer Society.

[BKS+] Baumgartner, J. et al.: Vote-auction.net.

[CH11] Clark, J.; Hengartner, U.: Internet Voting with Over-the-Shoulder Coercion-Resistance. FC 2011, vol. 2011, pp. 1–25, 2011.

[Cha96] Chaum, D.: Private signature and proof systems, 1996.

[Cho08] Chow, D.: Multi-Designated Verifiers Signatures Revisited. IJNS, 7(3):348–357, 2008.

[Cor01] Coron, J-S.: Cryptanalysis and Security Proofs for Public-key Schemes. PhD thesis, 2001.

[Cor02] Coron, J-S.: Optimal Security Proofs for PSS and other Signature Schemes. EUROCRYPT 2002, 2332:272–287, 2002.

[CV90] Chaum, D.; Van Antwerpen, H.: Undeniable signatures. Crypto'90, LNCS vol. 435, pp. 212–216. Springer, 1990.

[Des88] Desmedt, Y.: Society and group oriented cryptography: A new concept. Crypto'87, LNCS vol. 293, 120–127, 1988. Springer.

[DGB87] Desmedt, Y.; Goutier, C.; Bengio, S.: Special Uses and Abuses of the Fiat-Shamir Passport Protocol. Crypto '87, LNCS vol. 293, pp. 21–39, 1987. Springer.

[DL11] Dossogne, J.; Lafitte, F.: Mental Voting Booths, NordSec 2011, LNCS, 2011. Springer.

[DM09a] Dossogne, J.; Markowitch, O.: A Tripartite Strong Designated Verifier Scheme Based On Threshold RSA Signatures. SAM 2009, pp. 314–317, 2009. CSREA Press.

[DM09b] Dossogne, J.; Markowitch, O.: Voting With a Tripartite Designated Verifier Scheme Based On Threshold RSA Signatures. WIC09, vol. 1, pp. 113–118, 2009.

[DM10] Dossogne, J.; Markowitch, O.: E-voting : Individual verifiability of public boards made more achievable. WICSITB2010, pp. 5–10, 2010.

[DM11] Dossogne, J.; Medeiros, S. : Enhancing Cryptographic Code Against Side Channel Cryptanalysis with Aspects. WOSIS 2011, pp. 39–48, 2011. SciTePress.

[GHKR08] Gennaro, R.; et al.: Threshold RSA for Dynamic and Ad-Hoc Groups. EUROCRYPT'08, vol. 2008, pp. 88–107, 2008.
[Gol00] Goldreich, O.: Modern cryptography, probabilistic proofs and pseudorandomness, vol. 17 of Algorithms and Combinatorics. Springer, 2000.
[GP06] Ghodosi, H.; Pieprzyk, J.:. An Ideal and Robust Threshold RSA. VIETCRYPT 2006, vol. 4341 of LNCS, pp. 312–321, 2006. Springer.
[GRJK07] Gennaro, R.; et al.: Robust and Efficient Sharing of RSA Functions. J. Cryptology, 20(3):393, 2007.
[Hes02] Hess, F.: Efficient Identity Based Signature Schemes Based on Pairings. SAC'02, LNCS vol. 2595, pp. 310–324, 2002. Springer.
[HS00] Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. Eurocrypt'00), LNCS vol. 1807, pp. 539–556. Springer, 2000.
[JCJ05] Juels, A.; Catalano, D.; Jakobsson, M.: Coercion-resistant electronic elections. WPES'05, pp. 61–70, 2005. ACM Press.
[JSI96] Jakobsson, M.; Sako, K.; Impagliazzo, R.: Designated verifier proofs and their applications. Eurocrypt'96, LNCS vol. 1070, pp. 143–154. Springer, 1996.
[Lag07] Laguillaumie, F.: Multi-designated verifiers signatures: anonymity without encryption. IPL, 102(2-3):127–132, April 2007.
[LK00] Lee, B.; Kim, K.: Receipt-free electronic voting through collaboration of voter and honest verifier. JW-ISC2000, pp. 101–108, 2000.
[LSMP07] Li, Y.; et al.: Designated Verifier Signature: Definition, Framework and New Constructions. UIC'07, LNCS vol. 4611, pp. 1191–1200. Springer, 2007.
[LV04] Laguillaumie, F.; Vergnaud, D.: Multi-designated Verifiers Signatures. ICICS'04, vol. 3269 of LNCS, pp. 495–507, 2004. Springer.
[LWB05] Lipmaa, H.; Wang, G.; Bao, F.: Designated verifier signature schemes: Attacks, new security notions and a new construction. ICALP'05, LNCS vol. 3580, pp. 459–471, 2005. Springer.
[OMD04] Dall Olio, E.; Markowitch, O.:. Voting with designated verifier signature-like protocol. IADIS'04, pp. 295–301, 2004. Iadis Press.
[PG99] Pagnia, H.; Gärtner, F.: On the impossibility of fair exchange without a trusted third party. Tech. Rep., Darmstadt University of Technology, 1999.
[Pro] Proxyvote.com. Shareholder election website.
[Ric] Ricknäs, M.: Estonia to Use Mobile Phones to Simpify E-voting.
[Sha79] Shamir, A.: How to share a secret. Communications of the ACM, 22(11):612–613, 1979.
[SHCL08] Seo, S.; et al.: Identity-based universal designated multi-verifiers signature schemes. CSI, 30(5):288–295, July 2008.
[Sho00] Shoup, V.: Practical Threshold Signatures. EUROCRYPT'00, LNCS vol. 1807, pp. 207–220, 2000. Springer.
[WV10] Weldemariam, K.; Villafiorita, A.: A Survey: Electronic Voting Development and Trends. EVOTE2010, 2010.
[ZZZ08] Zhang, Y.;, Zhang, J.; Zhang. Y.: Multi-signers Strong Designated Verifier Signature Scheme. SNPD'08, pp. 324–328, 2008. IEEE Computer Society.