Voting with designated verifier signature-like protocol

Emmanuel Dall'Olio and Olivier Markowitch

Université Libre de Bruxelles
Bd du Triomphe CP212, 1050 Brussels, Belgium

{edalloli,omarkow}@ulb.ac.be

Abstract. We propose in this paper a new voting scheme where the voter, while receiving a receipt for his/her vote allowing further contestations, cannot use it to reveal the vote to other unspecified entities, and therefore cannot be forced to do so. The scheme is based on the concept of strong designated verifier signature scheme, and is very cheap in both computation and data communication, regarding to other works on the subject.
Keywords. Electronic vote, security protocol, designated verifier digital signatures

1 Introduction

In these days, electronic voting is becoming a reality in many cases, e.g. in companies, national ballot, etc. These mechanisms may take different forms, from automated voting system to vote through networks. In practice, the electronic vote by the mean of a network is nowadays ensured by a unique trusted third party, which is a real weakness as soon as its trustability becomes controversial.

Moreover, a problem arises when trying to combine voting privacy with the ability for the voter to check the correctness of his own voting by the mean of a receipt. Most of the time, being in possession of a voting receipt implies that a dishonest third party may possibly force the voter to reveal his/her vote. To avoid this, some works [BT94,HS00,LK00,MBC01] propose receipt-free voting protocols but in that case, the main problem becomes the difficulty for the voter to be sure his/her vote was taken into account. In [Oka97] this problem is handled, but at the price of an important amount of data transmissions and computations which are, in comparison with ours, multiplied by a factor representing a large number of authorities. Besides, [LK00] propose another scheme needing a large quantity of computations and which, above all, requires the voters to trust the polling office of behaving honestly. This seems to us an unrealistic assumption.

This paper proposes a new voting scheme where the voter, while receiving a receipt for his/her vote, cannot use it to reveal the vote to someone else. This

feature allows the voters to verify their own vote, but also to complain if necessary, while it forbids an attacker to threaten or buy them, in order to control their vote. The scheme is based on the concept of strong designated verifier signature scheme.

A strong designated verifier signature scheme, such as [JSI96,RST01,SKM03], provides that a message is signed in such a way that only one person, the designated verifier will be able to check its validity. Moreover this designated verifier will not be able to convince other entities that the signature is a valid one, even by revealing his own secret key.

By using designated-verifier schemes to create a voting receipt, the voter, who is the designated verifier, is both able to check his vote and unable to convince anyone else with his receipt. In [HS00], Hirt and Sako introduced a designated-verifier based voting scheme, using a multiple permutation over all possible votes. In this paper, we build a more natural voting scheme, using the strong designated verifier signature introduced in [SKM03]. In comparison with the protocol of Hirt and Sako, our scheme is nearer of the traditional voting system, and moreover cheaper, since our system only requires a constant number of encryptions and data transmissions for each vote, where they need, for each vote, a number of these operations which is proportional to both the number of possible votes and the number of authorities.

In this scope, as in many voting schemes, we introduce a trusted judge. He has the same verifying power as the voter, and can be used in case of conflict between the voter and the polling office.

The paper is organized as follows. In the next section we introduce the notations used in the rest of the paper, and we describe the voting scheme. Therefore we analyse our scheme in the third section, and we conclude in the fourth and last section.

2 The voting scheme

In this section, we describe our voting protocol during which the voter, after being identified by an identification authority, submits a vote (yes/no, names, etc.) to a known polling office P, under the authority of a judge J, which we will suppose unique, though his work could easily be dispatched. We suppose that a public key infrastructure exists.

2.1 Notations

We will use the following notation:

- V : the voter
- J : the judge

– I : the identification authority
– P : the polling (voting) office
– $\sigma_A()$: a blind signature produced by the entity A
– $blind()$: a blinding function related to $\sigma_A()$
– $unblind()$: the corresponding unblinding function
– v : the concatenation of a vote and a random number
– $A \xrightarrow{m} B$: the transmission of a message m from A to B
– $A \overset{\text{Auth.}}{\longleftrightarrow} B$: the mutual entity authentication between A and B
– $E_A()$: asymmetric cipher intended to A

We now divide our voting scheme in four sub-protocols: initialization, voters identification, vote and publication, that are described hereafter.

2.2 Initialization

Let p be a large prime, q a prime factor of $p-1$ and g a generator of $\mathbb{Z}_p^*$ of order q. The public information are p, q and g; x_P, x_J are respectively the private key of the voting office and the judge's one. The corresponding public keys are calculated as $y_P = g^{x_P} \bmod p$ and $y_J = g^{x_J} \bmod p$.

The voter tells the judge in which polling office he is voting. The judge answers by sending $(x_V, g_V, y_{P,V}, c_V)$ that will be used in the verifying procedure. Formally, we have:

V J

$\xrightarrow{E_J(y_P)}$

checks that y_P designates a valid polling office
randomly chooses two values associated with the voter: d_V and k_V
$x_V = x_J(d_V)^{-1} \bmod q$
$g_V = g^{d_V} \bmod p$
$y_{P,V} = y_P^{d_V} \bmod p$
$c_V = y_J^{k_V}$

$\xleftarrow{E_V(x_V, g_V, y_{P,V}, c_V)}$

The judge only has to remember d_V and k_V for each voter[1].

2.3 Voters identification

We have to make sure that the vote remains secret, even from the voting authorities. To achieve this goal, at the beginning of the vote, each voter V obtains,

[1] In order to save storing space for the judge, k_V can be choosen in such a way that it can be deduced from d_V by the judge

as described below, a random number signed from an identification authority I, which classically has to be distinct from the polling office. As for the judge, the role of I could be divided into several parts and handled by different identification authorities.

First of all, the authority needs to make sure the voter has not already got its signed random number. This can be done easily if the identification authority keeps a list of all the already registered voters. When the voter wants to vote, he starts a mutual authentication protocol (for instance [BGH+93]) with I. Then the identification authority checks in its list if this voter already appears. If not, the voter is allowed to submit a blinded random value r, using a blinding operator of a blind signature scheme such as described in [Cha83], and sends this blinded value to the identification authority. If it is the first time that a voter authenticates itself, the identification authority will sign it. Once he receives the blinded signed message, the voter unblinds the signature and obtains the identification authority's signature on the original random value.

$$
\begin{array}{lcl}
V & & I \\
 & \overset{\text{Auth.}}{\longleftrightarrow} & \\
\text{chooses a random } r & & \\
 & \overset{blind(r)}{\longrightarrow} & \\
 & & \text{computes } \varsigma = \sigma_i(blind(r)) \\
 & \overset{E_V(\varsigma)}{\longleftarrow} & \\
\text{computes } unblind(\varsigma) & &
\end{array}
$$

The signature on r is, once unblinded, used as a random and untraceable identification number for the voter.

2.4 Vote

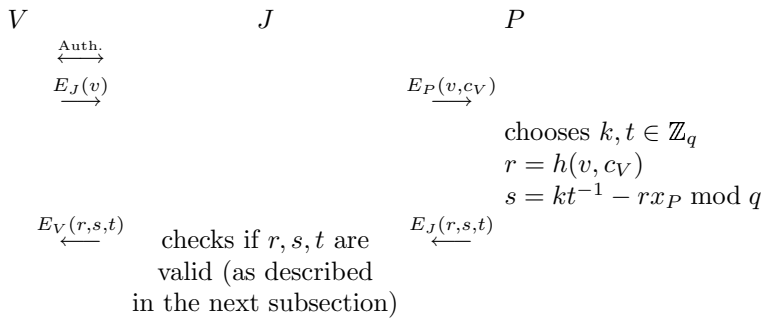
The voter concatenates his vote with its signed random number and sends this combination v to the polling office, who checks if he has already voted. This can be done easily by keeping a list of the random numbers already used. If the voter has not already voted, the polling office replies with a designated verifier receipt, computed in the following way:

$$
\begin{array}{lcl}
V & & P \\
 & \overset{E_P(v,c_V)}{\longrightarrow} & \\
 & & \text{chooses } k, t \in \mathbb{Z}_q \\
 & & r = h(v, c_V) \\
 & & s = kt^{-1} - rx_P \bmod q \\
 & \overset{E_V(r,s,t)}{\longleftarrow} &
\end{array}
$$

Using his secret key x_V and the three other informations g_V, $y_{P,V}$ and c_V provided earlier by the judge, the voter checks if the receipt matches his vote by verifying if

1. $(g_V^s y_{P,V}^r)^{tx_V} \bmod p = c_V$
2. $h(v, c_V) = r$

If the receipt is incorrect, the voter can cancel his vote by asking the judge to vote on his name, after being mutually authenticated with him. The loss of the voter's anonymity for the judge is not a problem as the judge is considered honest. This is achieved by the following steps:



If the receipt provided by the polling office is incorrect, the judge contacts this polling office to resolve the problem, possibly using appropriate legal procedures.

2.5 Results publication and possible complaints

Before closing the vote, a delay is left to the voters in order to complain if the transmitted receipt was incorrect. This delay is supposed long enough to prevent course problems. The polling office then publishes a list of all the votes, each associated with the corresponding random number. It is then straightforward for the voter to check if his vote was changed after the reception of the polling office's receipt, and to contact the judge if it is the case.

If his vote was corrupted or lost, the voter authenticates with the judge and sends his receipt (r, s, t) to him. The judge thens checks the receipt, using his secret key x_J, by verifying:

1. whether $h(v, (g^s y_P^r)^{tx_J} \bmod p) = r$, to be sure that the receipt is correctly formed. The correspondence between the voter and the judge's verification appears clearly:

$$\begin{aligned}(g_V^s y_{P,V}^r)^{tx_V} \bmod p &= (g^{d_V s} y_P^{d_V r})^{tx_J d_V^{-1}} \bmod p \\ &= (g^s y_P^r)^{tx_J} \bmod p\end{aligned}$$

2. that the equation $y_J^{st} = c_V y_P^{-rx_J t} \bmod p$ holds, where $c_V = y_J^{k_V}$ can be recomputed by the judge in order to convince himself that the receipt has not been forged by the voter. Indeed, if the voter has not complained after he got the receipt during the vote phase, then this equality will hold unless the voter is trying to cheat.

If the receipt provided by the voter is correct and does not match the published vote, the judge accepts the voter complaint and contacts the polling office to resolve the problem, possibly using appropriate legal procedures.

3 Analysis

In this section, we show that our scheme is secure against impersonalization; respects anonymity, confidentiality and impossibility to forge a vote or to vote more than once; and prevents the polling office of cheating.

First of all, the impersonalization is prevented during the identification phase thanks to the mutual authentication between V and I. Moreover, during the vote, the designated verifier protocol forbids, via the resistance against forging [SKM03], any intruder to take the place of any party.

Anonymity is a consequence of the signature provided by the identification authority. Since the signature was achieved blindly, I cannot link the voter to his confidential identification number, nor can anyone else. Since the voter always authenticates himself by the way of this number, even if the polling office and the identification authority associate together, they cannot break the voter's anonimity.

Vote confidentiality is insured by the fact that our scheme is based on a designated verifier signature protocol. In case of coercion, the voter can generate a c and a corresponding receipt by himself in the following way: he chooses α, β in $\mathbb{Z}_q$, and computes:

$$\begin{aligned} c &= g_V^{\alpha} y_{P,V}^{\beta} \bmod p \\ r &= h(v, c) \\ \ell &= \beta r^{-1} \bmod q \\ s &= \alpha \ell^{-1} \bmod q \\ t &= \ell x_V^{-1} \bmod q \end{aligned}$$

Anyone possessing the voter's secret key x_V can straightforwardly check that $h(v, (g_V^s y_{P,V}^r)^{tx_V} \bmod p) = r$:

$$\begin{aligned} h(v, (g_V^s y_{P,V}^r)^{tx_V} \bmod p) &= h(v, (g_V^s y_{P,V}^r)^{\ell} \bmod p) \\ &= h(v, g_V^{s\ell} y_{P,V}^{r\ell} \bmod p) \\ &= h(v, g_V^{\alpha} y_{P,V}^{\beta} \bmod p) \\ &= h(v, c) = r \end{aligned}$$

Therefore, an attacker, even with the private key of the voter, cannot still make the difference between a true receipt and the receipt generated by the voter, and can hence draw no conclusion at all from any receipt he gets. Moreover the voter cannot mislead the judge in the same way, since the judge checks if $y_J^{st} = cy_P^{-rx_J t} \bmod p$:

$$\begin{aligned} y_J^{st} &= y_J^{(kt^{-1} - rx_P)t} \bmod p \\ &= y_J^{k - rx_P t} \bmod p \\ &= y_J^k y_J^{-rx_P t} \bmod p \\ &= cg^{-rx_J x_P t} \bmod p \\ &= cy_P^{-rx_J t} \bmod p \end{aligned}$$

On the other hand, the same checking applied to a fake receipt leads to the following result, where all of the calculations are computed modulo p:

$$\begin{array}{lll} y_J^{st} & \stackrel{?}{=} & cy_P^{-rx_J t} \\ \| & & \| \\ g^{x_J st} & \stackrel{?}{=} & g_V^{\alpha} y_{P,V}^{\beta} y_P^{-rx_J t} \\ \| & & \| \\ g^{x_J \alpha \ell^{-1} \ell x_V^{-1}} & \stackrel{?}{=} & g^{\alpha d_V} g^{d_V \beta x_P} g^{-rx_J x_P t} \\ \| & & \| \\ g^{x_J \alpha x_V^{-1}} & \stackrel{?}{=} & g^{\alpha d_V + d_V \beta x_P - rx_J x_P t} \end{array}$$

which implies that, to trick the judge, one has to choose α and β such that: $x_J \alpha x_V^{-1} = \alpha d_V + d_V \beta x_P - rx_J x_P t \bmod q$, which cannot be done with probability more than $1/q$, since x_J, x_P and d_V are unknown for the voter.

Forging a vote is prevented, first of all, thanks to the security of the identification protocol used in the initialization. Indeed, if the voter is unable to identify himself, he never receives his identification number, necessary to start the voting procedure. Moreover, forging the designated verifier signature needs to obtain the keys or to solve a discrete logarithm, as seen in [SKM03].

In the same way, unless obtaining two different identification numbers, which is impossible if the identification authority behaves correctly and remembers who is already identified, a voter cannot vote more than once, since the identification numbers are stored on this purpose.

The polling office has two ways of cheating: either he changes the vote and sends a receipt corresponding to the original vote, or he changes the vote and sends a receipt for the fake vote. In the first case, the voter has no way to detect the fraud during the vote, but he can notice the problem when the results are published, and complain at that moment. The second case is avoided by the fact that the voter can ask the judge to vote in his place. We can note that, since the polling office cannot successfully cheat in this second way, it is reasonable to assume that only a marginal part of the vote will ever be done by the judge. It is therefore pointless to discuss the problem of an overcharging of judge.

In the improbable case of a coalition of several voters and polling offices, trying to flood the judge during the vote phase, by asking him a large amount of vote requests, we suppose that he has the power to postpone the end of the voting and possibly ask the voters to vote again, in order to handle every request without risking a buffer overflow.

4 Conclusion

We have introduced a voting system based on a strong designated verifier scheme, which allows the voter, and only him, to check his voting, and possibly complain about it. The strength of our method lies in the fact that no attacker can manipulate the voter. Besides, a trusted judge can help to acknowledge a vote, or point out that a vote has been miscounted. Moreover, our scheme is secure against impersonalization, dishonest voters or polling office, and ensures anonymity. Incidentally, its cost is very cheap comparing to other related works. Finally, thanks to similarity between our scheme and a classical voting system, the changes needed to use our protocol in the framework of calls for tenders are very light.

References

[BGH+93] Ray Bird, Inder S. Gopal, Amir Herzberg, Philippe A. Janson, Shay Kutten, Refik Molva, and Moti Yung. Systematic design of a family of attack-resistant authentication protocols. *IEEE Journal on Selected Areas in Communications*, 11(5):679–693, 1993.

[BT94] Josh Cohen Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections. In *26th ACM Symposium on the Theory of Computing*, pages 544–553. ACM Press, 1994.

[Cha83] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology Proceedings of Crypto 82*, pages 199–203. D. Chaum, R.L. Rivest, and A.T. Sherman (Eds.), Plenum, 1983.

[FS87] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — Crypto '86*, pages 186–194, New York, 1987. Springer-Verlag.

[HS00] Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. In *Proceedings of Eurocrypt 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 539–556. Springer-Verlag, 2000.

[JSI96] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In *Proceedings of Eurocrypt 1996*, volume 1070 of *Lecture Notes in Computer Science*, pages 143–154. Springer-Verlag, 1996.

[LK00] Byoungcheon Lee and Kwangjo Kim. Receipt-free electronic voting through collaboration of voter and honest verifier. In *Korea-Japan Joint Workshop on Information Security and Cryptology*, pages 101–108, 2000.

[MBC01] Emmanouil Magkos, Mike Burmester, and Vassilios Chrissikopoulos. Receipt-freeness in large-scale elections without untappable channels. In *First IFIP Conference on E-Commerce, E-Business and E-Government*, pages 683–694. Kluwer Academic Publishers, 2001.

[Oka97] Tatsuaki Okamoto. Receipt-free electronic voting schemes for large scale elections. In *5th International Security Protocols Workshop*, volume 1361, pages 25–35. Springer-Verlag, 1997.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *Proceedings of Asiacrypt 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–564. Springer-Verlag, 2001.

[SKM03] Shahrokh Saeednia, Steve Kremer, and Olivier Markowitch. Efficient designated verifier signature schemes. In *24st Symposium on Information Theory in the Benelux*. Werkgemeenschap Informatie en Communicatietheorie, 2003.