Publicly Verifiable Private Credentials – a technique for privately signing Citizen Initiatives –

Marius C. Silaghi and Kishore R. Kattamuri
Florida Institute of Technology
msilaghi,kattamuk@fit.edu

January 26, 2005

Technical Report CS-2005-2

Abstract

Besides voting, another way of practicing democracy is by signing popular/citizen initiatives or optional referendums of certain types. We address the problems related to the electronic remote signing of support for referendums and popular initiatives (e-referendums). We propose, *publicly verifiable private credentials (PVPCs)*, a type of certified pseudonym digital signatures for which one can publicly prove that only eligible users (belonging to a specified group) got them and that no user got two of them. They can be used to sign citizen initiatives. Our technique generates PVPCs that can be publicly proven to belong to an unknown permutation of the eligible users (proving the aforementioned property). We also argue that e-referendum systems can achieve more success than remote e-voting (being more robust to the main weaknesses of the SERVE project, namely denial-of-service, "Man in the middle", and virus attacks). In particular we provide a technique to reduce the risk of exposure to virus attacks by incorporating manually generated passwords into computer-generated random numbers.

1 Introduction

Besides voting, another way of practicing democracy is by signing popular/citizen initiatives or optional referendums of certain types. Citizen initiatives allow for the inclusion of constitutional or statutory proposals on the ballot if enough signatures are collected in support of the proposal. Some optional (abrogative) referendums are based on a similar collection of signatures, but are aimed at rejecting recently issued legislation [6, 9, 14, 7]. We address the problems related to the electronic remote signing of support for referendums and popular initiatives (e-referendums). We also argue that such systems can achieve more success than remote e-voting (being more robust to the main weaknesses of the SERVE project, namely denial-of-service, "Man in the middle", and virus attacks [10]).

Most citizen initiative systems offer some amount of privacy to the signers. E.g., the Swiss constitution stipulates that after the signature gathering stage for a citizen initiative (referendum) is closed, the access to archives containing names of the signers is very restricted [7]. A similar level of privacy can be easily offered for e-referendums with simple digital signature schemes (abstaining from publishing the name of the signers [1]). Unable to access archives, citizens have to trust the administration for a correct count of the signatures (for e-voting this already led to many debates and research effort, see [12] and the IEEE P1583 Standard).

Our motivation stems from the conjecture that the limitations of the degree of privacy and of the public verifiability of current citizen initiative systems is due to past technical limitations. It follows that many citizens will be interested in an improved privacy and verifiability.

We propose, *publicly verifiable private credentials (PVPCs)*, a type of certified pseudonym digital signatures for which one can publicly prove that only eligible users (belonging to a specified group) got them and that no user got two of them. They can be used to sign citizen initiatives. Our technique generates PVPCs that can be publicly proven to correspond to a permutation of the eligible users (proving the aforementioned property).

2 Background

Cryptographic blind signatures allow to obtain an electronic certificate without revealing the relation certificate-owner. Assume Bob wants Alice (authority) to digitally sign a message, m (e.g., a document), certifying it. However Bob does not want Alice to learn anything about m. Blind digital

signatures based on RSA work as follows:

1. Alice's RSA public/secret keys are (n, e) and (n, d).
2. Bob generates a random number r, $gcd(r, n) = 1$, and sends $x = (r^e m) \bmod n$ to Alice.
3. Alice digitally signs x, sending $t = x^d \bmod n$ to Bob.
4. Bob gets the signature s of m, $s = m^d \bmod n$, by computing $s = r^{-1} t \bmod n$. Everybody can verify s, verifying that $m = s^e \bmod n$.

Private credentials [8] allows somebody to prove that he/she has certain characteristics (e.g. is member of a certain organization, has a certain age) without revealing anything else. These credentials can be offered by an authority and can be implemented as an extension of blind digital signatures. The verifiers need to trust that the authority does not introduce false credentials. See [4, 5, 11, 8, 15] for other pseudonym techniques.

3 Publicly Verifiable Private Credentials

PVPCs use another extension to blind signatures for proving that a user belongs to a group G. Any third party can verify that a given credential belongs to an authorized person (member of G) and that each such person has exactly one pseudonym.

Let the authority A publish a public key for blind signatures, P_A, and keep the corresponding secret key S_A. First, each member C_i of $G = \{C_1, ..., C_N\}$ generates a digital signature key pair (P_i,S_i) and a pseudonym digital signature key pair (P'_i,S'_i,$S_A(P'_i)$). P_i and S_i are registered with A. To get the pseudonym digital signature, C_i generates a secret digital signature key pair (P'_i,S'_i) and asks the authority to blindly sign P'_i, getting $S_A(P'_i)$. Then, C_i sends (P'_i,$S_A(P'_i)$) to A using an anonymous channel (Chaumian mix-net [3] in a version called PVPC_0). The authority builds a list with all received pseudonym public keys in some order, $L = [P'_{k_1}, ..., P'_{k_N}]$, and asks all members of G to sign $Hash(L)$. Each member C_i of G verifies that P'_i is in L, and sends $S_i(Hash(L))$ to A, to certify that his pseudonym is listed. If only members of G sign and if the number of signers equals the number of pseudonyms, any third party can be convinced that no false participant was introduced.

To ask a blind signature of its pseudonym public key, C_i sends the request digitally signed with the public key of his digital signature. As an

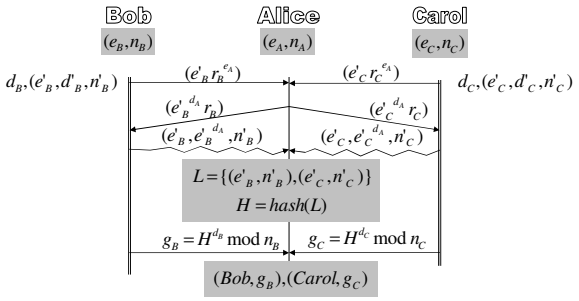

Figure 1: PVPC generation.

example, the following steps create RSA-based PVPCs for a group G with two members.

1. Alice is the authority. Bob and Carol are the only members of the group G. Alice has RSA key (e_A, d_A, n_A), while Bob and Carol have (e_B, d_B, n_B) and (e_C, d_C, n_C).

2. Bob and Carol, each secretly generate a key pair: (e'_B, d'_B, n'_B) and (e'_C, d'_C, n'_C), respectively.

3. Bob and Carol get from Alice $({e'_B}^{d_A} \bmod n_A)$ and $({e'_C}^{d_A} \bmod n_A)$, respectively, using the blind signature scheme.

4. Using an anonymous channel, Bob and Carol send Alice their pseudonyms: $(e'_B, {e'_B}^{d_A}, n'_B)$ and $(e'_C, {e'_C}^{d_A}, n'_C)$, respectively.

5. Alice computes $L = [\langle e'_B, n'_B \rangle, \langle e'_C, n'_C \rangle]$ (or $L = [\langle e'_C, n'_C \rangle, \langle e'_B, n'_B \rangle]$) and $H = Hash(L)$ and publishes $\langle L, H \rangle$.

6. Bob and Carol verify that their pseudonyms are in L and that $H = Hash(L)$. Then, they send Alice $g_B = H^{d_B} \bmod n_B$ respectively $g_C = H^{d_C} \bmod n_C$. Alice publishes (Bob,g_B) and (Carol,g_C).

7. Verifier Victor gets L, H, g_B, g_C from Alice and checks that $H = Hash(L)$, $H = g_B^{e_B} \bmod n_B$, and $H = g_C^{e_C} \bmod n_C$. Victor accepts any message signed with a public key in L as coming from a member of G.

PVPC_0 (PVPC generation with mix-nets based on volunteers [3, 2]) is not particularly robust due to possible problems with mix-nets [2]. It can

also be disrupted by members of G falsely claiming that their signature is not in L. However, it is easy to obtain a robust version using a small modification, obtaining PVPC_1.

In PVPC_1 the pseudonyms can be sent to Alice using mix-nets similar to Merritt's election protocol [13] rather than the original mix-net [3]. The shuffling agents are the members of G and their correctness can be publicly verified. Each user can detect the shuffler that loses his pseudonym while shufflers can prove their correctness. Liars and non-cooperating citizens can then be removed from G.

If we do not intend to hide the relation pseudonym-citizen from the authority, then citizens can ask the authority, A, to digitally sign their pseudonym, instead of asking A to blindly sign it. Then, the anonymous channel is no longer needed and a normal channel can be used instead. This version will be refered to as PVPC_2.

The system is scalable if the eligible users are grouped and verified separately for each circumscription, rather than having to verify a whole state. The only information leak is the totals for each circumscriptions, and this is acceptable and common for current manual systems.

PVPC-based e-referendums can be used in parallel with manual signature collection systems. However, users of PVPCs cannot sign manually and vice-versa. Periodically (since some citizens die, lose/acquire rights), the setup procedure is repeated. If somebody learns the secret key of a user and misuses it, the user can disable his/her PVPC and wait for a renewal to get a new credential or register for manual signatures. This results in a temporary loss of rights, risk that may be found acceptable by many users.

4 Robust remote e-referendums

E-referendums are more robust to denial-of-service attacks than e-voting because they need not complete on a single day. Denial-of-service attacks are not successful for long periods of time.

The only way virus attacks can break the integrity of PVPC-based e-referendums is by manipulating users to generate identical pseudonyms. To resist this, it is sufficient that each random number used for generating pseudonyms (i.e., e_C, q_C, p_C) has certain bits input manually by the user. The user is asked for a password that will be used to replace some of the significant bytes (but not the most significant byte) of the random number. The user will then be able to check that his password is present in the randomly generated keys, and detect an attack if the password does not

appear.

Man in the middle attacks can modify messages, but e-referendums become robust if citizens are allowed to correct previous decisions. E-referendums can allow users to check at any moment the way in which their signature was counted and to re-submit/withdraw their signature in case a problem is detected.

5 Conclusions

References

[1] J.-L. Abbet. Projet pilote neuchatêlois, 2004.

[2] T. Atkinson and M.-C. Silaghi. Reply-pay and hansdhaking for incentives with anonymizer servers. Technical Report TR-FIT-16/2004, Florida Institute of Technology, Melbourne, FL, November 2004.

[3] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Com. of ACM*, 24(2):84–88, 1981.

[4] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, October 1985.

[5] L. Chen. Access with pseudonyms. In *Cryptography: Policy and Algorithms*, number 1029 in LNCS, pages 232–243, 1995.

[6] E. Greenwood. Focus on ..- citizen initiatives. http://focus.at.org/direct-democracy/citizen-initiatives, 2004.

[7] C. Helvetica. Popular initiative. http://www.admin.ch/ch/e/pore/index3.html, 2004.

[8] Z.-K. S. Inc. Private credentials. http://www.zeroknowledge.com/media/credsnew.pdf, 2000.

[9] P. K. Jameson and M. Hosack. Citizen initiative in Florida: An analysis of Florida's constitutional initiative process, issues, and statutory intitiative alternatives. *Law Review*, 23(2), 1996. http://www.law.fsu.edu/journals/lawreview/issues/232/ jameson.html.

[10] D. Jefferson, A. D. Rubin, B. Simons, and D. Wagner. A security analysis of the secure electronic registration and voting experiment (serve). New York Times (http://www.servesecurityreport.org), January 2004.

[11] A. Lysyanskaya, R. Rivest, and A. Sahai. Pseudonym systems. In *Selected Areas in Cryptography, SAC*, number 1758 in LNCS, pages 184–200, 1999.

[12] R. Mercuri. Explanation of voter-verified ballot systems. *ACM Software Engineering Notes (SIGSOFT)*, 27(5), September 2002. http://www.notablesoftware.com.

[13] M. Merritt. *Cryptographic Protocols*. PhD thesis, Georgia Inst. of Tech., Feb 1983.

[14] Oregon. Elections - initiative, referendum & referral. http://www.sos.state.or.us/elections/other.info/irr.htm, April 2004.

[15] R. Samuels and E. Hawco. Untraceable nym creation on the freedom 2.0 network. http://www.zeroknowledge.com/media/Freedom-NymCreation.pdf, 2000.