JCJ with Improved Verifiability Guarantees

Peter Browne Rønne

SnT, University of Luxembourg, Luxembourg City, Luxembourg
peter.roenne@uni.lu

Abstract. Using an additional zero-knowledge proof we improve the verifiability guarantees for the JCJ e-voting protocol [4], meaning that neither the Registration nor the Tally Tellers can collude to break verifiability.

1 Introduction

In their seminal paper Juels, Catalano and Jakobsson [4] investigated possible coercion attacks in e-voting and suggested a coercion-resistant protocol. This protocol was later implemented as Civitas [3] with several specifications, in particular for the registration procedure. Many papers have investigated the security properties of this protocol and several modifications and alternatives have been suggested, see e.g. [5] for an overview. The scheme is in its core quite simple: The voters have secret credentials provided to them by a set of Registration Tellers. When casting a ballot, the voter append her encrypted credential. A coerced voter can simply provide the coercer with a fake credential. Ballots containing invalid credentials are discarded in the tally by the use of plaintext equivalence tests done by a (threshold) set of Tally Tellers.

One problem of JCJ/Civitas is that it has verifiability trust in the Registration and Tally Tellers. If all the Registration Tellers collude, or a threshold set of the Tally Tellers, they know the credentials being used by the voters and can cast valid ballots on behalf of any voter. Depending on the verifiability definition, JCJ thus does not satisfy eligibility verifiability, see e.g. [8]. In JCJ this is especially troublesome since the adversary can also use this to change the votes of honest voters, e.g. if the update policy is last vote counts, the adversary will cast his choice as late as possible overruling the already cast honest vote.[1] This does not invalidate the security promises of Civitas which assumes no such collusion of Tellers. However this could, arguably should, be worrisome to a voter,

[1] This might be detected by an alert voter since JCJ/Civitas chooses to perform the weeding of votes before mixing (which in turn might be troublesome in dynamic coercion scenarios). In general, the percentage of voters actually doing such security checks are generally low, and even if a voter reports such an event it would be disputable since the voter could have cast the extra votes herself. A robust blaming seems to endanger coercion-resistance and is necessary in order to avoid voters maliciously denouncing a valid election after the tally has been announced. Thus it is better to prevent this situation from happening at all.

who does not know if the election has been manipulated by colluding Tellers or perhaps a hacker capable of attacking a few central security points.

In this note we remove the verifiability trust in the Tellers, completely within the setup of JCJ/Civitas, and with the mild price of a longer zero-knowledge proof for the vote casting part and a simple change in the registration procedure.

The idea is very simple, Civitas assumes (as also suggested in JCJ) that each voter has a designated verifier key. We let the credential depend on this key such that only the voter knows the discrete logarithm of the credential. Ballots are then cast with a proof of knowledge of this discrete logarithm. Now, even if the Tellers collude, they can get to know the credentials, but under assumption of hardness of the discrete logarithm problem they cannot use this to cast a valid vote. To our knowledge this idea has not been reported earlier.

2 Protocol Structure

To keep this note short we will assume that the reader is familiar with Civitas [3] and only display the differences. The main participants are the voters, V_i, the Registration Tellers, RT_j, and the Tally Tellers, TT_k. The cryptography is based on a DDH secure group of prime order q and generator g. Let $\mathsf{enc}(v; \mathsf{K}_{\mathrm{TT}})$ denote ElGamal encryption in this group and K_{TT} be the (threshold) public key of the Tally Tellers. We also assume that the voters are provided with an infrastructure of designated verifier keys $\mathsf{dvk}_i = g^{x_i}$.

2.1 Registration

The registration is quite similar to Civitas. For each eligible voter V_i, each Registration Teller RT_j picks randomly $c_{ij} \in \mathbb{Z}_q$ and publishes $\mathsf{enc}(g^{c_{ij}}; \mathsf{K}_{\mathrm{TT}})$ on the Bulletin Board in a row marked for voter V_i. From RT_j voter V_i gets c_{ij} and a zero-knowledge proof designated to dvk_i of correct encryption of $g^{c_{ij}}$. For each voter the ciphertexts of the credential shares are multiplied together and further multiplied with $\mathsf{enc}(\mathsf{dvk}_i; \mathsf{K}_{\mathrm{TT}})$ encrypted with trivial randomness. By the homomorphic property of ElGamal this gives an encryption of the voter credential $C_i = g^{c_i} := g^{\sum_j c_{ij} + x_i}$. The difference to Civitas is that the voter gets c_{ij} instead of $g^{c_{ij}}$ and the extra multiplication with dvk_i (or some public key of the voter).[2]

[2] We have here followed Civitas closely, but we could also construct the keys g^{x_i} during the registration interactively. The important part is that the registration authorities do not know x_i and the voter does, as in a designated verifier key infrastructure. One can also use erasure of data at the end of registration as is suggested in JCJ as an alternative to designated verifier proofs. In both cases an interactive proof of knowledge of x_i should be given during registration where we anyway assume no coercion (another possibility is to keep the term g^{x_i} under encryption and split for each Teller and the proofs changed accordingly). This is in order to stop a coercer from determining g^{x_i} before registration without divulging the secret key x_i to the voter, which in turn could create receipts and also allow forced abstention attacks.

If the voter is coerced, she chooses at random an alternative value $c'_i \in \mathbb{Z}_q$ and shows $g^{c'_i}$ as her credential to the coercer. The proofs can be faked with her secret designated verifier key and this key could even be revealed to the coercer, at least after registration.

2.2 Vote Casting

Like in JCJ/Civitas a ballot is cast via an anonymous channel to the Bulletin Board. Given a credential C and a vote choice v, the ballot has the form

$$(\, \mathsf{enc}(C; \mathsf{K}_{\mathrm{TT}}),\, \mathsf{enc}(v; \mathsf{K}_{\mathrm{TT}}),\, \pi_1,\, \pi_2,\, \pi_3 \,)\ .$$

Here π_1, π_2, π_3 are non-interactive zero-knowledge proofs (NIZKPs). Just like in Civitas, π_1 is a proof that the ciphertext for the vote is well-formed, and π_2 is a proof that C and v are simultaneously known which is done by demonstrating knowledge of the random coins used in the encryptions. This proof prevents the ballot from being malleable. The proof π_3 is our novel part and is a proof of knowledge of the discrete logarithm of the encrypted credential. To specify the proof let $\mathsf{enc}(C; \mathsf{K}_{\mathrm{TT}}) := (a, b)$. The proof is now done by choosing a random s and publishing

$$a^s, b^s, C^s, \mathsf{DLK}(a^s, a), \mathsf{DLE}(a^s, a, b^s, b), \mathsf{DLK}(C^s, g), \mathsf{DLE}(a^s, g, b^s/C^s, \mathsf{K}_{\mathrm{TT}})$$

where $\mathsf{DLK}(a, g)$ is a NIZKP of knowledge of the discrete logarithm of a relative to g and $\mathsf{DLE}(a, g, b, g')$ is a NIZKP of equality of the discrete logarithm of a and b with respect to generators g and g'. These are efficiently implemented by using the Fiat-Shamir transformation on Schnorr [7] and Chaum-Pedersen proofs [2]. The first part of π_3 shows that we lift a, b to the same known power and then we show that C^s is the plaintext of this encryption, and finally that we know its discrete logarithm. Since s is known, this shows that we know $\log_g C$. This gives soundness. Zero-knowledge follows from the employed NIZKPs and DDH for the extra elements, given that the adversary does not know the randomness in the encryption and the secret key of K_{TT}, in which case he would anyway know C. Full proofs are postponed for a long version of this note.

To avoid having malleability in the proofs we include both encryptions in the hashes of the Fiat-Shamir heuristic, see also [1]. If this is also done for π_1, we can drop the proof π_2 and in this case the extra overhead compared to Civitas is just 3 hashes and 7 exponentiations.

Finally, the tally can be done like in JCJ/Civitas.

3 Security & Comments

We see that it is no longer possible to cast valid ballots unless you know the discrete logarithm of the credential. The Registration Tellers do not know this even if they are all colluding, unless they somehow know the secret key of dvk_i, in which case they anyway could attack verifiability during registration. The

Tally Tellers, if colluding, could decrypt the credentials, but do not know the discrete logarithms. Such misbehavior can now give rise to privacy or possibly coercion attacks, but no longer endangers verifiability. We emphasize that this scheme does not improve on coercion-resistance, but only the verifiability guarantees, and it has the same assumptions as JCJ/Civitas for coercion-resistance. In JCJ this is done by assuming that the registration phase proceeds without any corruption, and will also require that the two set of tellers are not colluding, see further [4].

Note that more user-friendly versions of the protocol e.g. using hardware with pin-codes [6,5] are still possible if adapted. In the future it would be important to examine the security of the protocol in detail, cast the cryptography in the setting of bilinear maps to remove the random oracle assumption, and further develop the protocol e.g. using secret registration (towards the Tellers) for better and everlasting privacy.

Acknowledgements. We would like to thank Peter Y.A. Ryan, Vincenzo Iovino and Steve Kremer for discussions, and the reviewers for comments and suggestions. Further, Peter B. Rønne is supported by the INTER-Sequoia project from Fonds National de la Recherche, Luxembourg, which is joint with the ANR project SEQUOIA ANR-14-CE28-0030-01.

References

1. David Bernhard, Olivier Pereira, and Bogdan Warinschi. How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 626–643. Springer, 2012.
2. David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105. Springer, 1992.
3. Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: A secure voting system. In *In IEEE Symposium on Security and Privacy*, 2008.
4. Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society, WPES 2005, Alexandria, VA, USA, November 7, 2005*, pages 61–70, 2005.
5. Stephan Neumann, Christian Feier, Melanie Volkamer, and Reto E. Koenig. Towards A practical JCJ / civitas implementation. *IACR Cryptology ePrint Archive*, 2013:464, 2013.
6. Stephan Neumann and Melanie Volkamer. Civitas and the real world: Problems and solutions from a practical point of view. In *Seventh International Conference on Availability, Reliability and Security, Prague, ARES 2012, Czech Republic, August 20-24, 2012*, pages 180–185. IEEE Computer Society, 2012.

7. Claus-Peter Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.
8. Ben Smyth, Steven Frink, and Michael R. Clarkson. Computational election verifiability: Definitions and an analysis of helios and JCJ. *IACR Cryptology ePrint Archive*, 2015:233, 2015.