On coercion resistance in decentralized voting (Extended Abstract)

Pourandokht Behrouz, Panagiotis Grontas, Marianna Spyrakou

School of Electrical and Computer Engineering,
National Technical University of Athens
pbehrouz@gmail.com, pgrontas@corelab.ntua.gr, mar.spyrakou@gmail.com

Abstract. We investigate the problem of coercion resistance in decentralized voting scenarios. To overcome the barrier imposed by universal verifiability we create a new form of a private channel. Our solution builds on a new cryptographic primitive, Conditional Designated-Verifier Ring Signatures, that combines the anonymity provided by ring signatures with the controlled verifiability of strong designated verifier signatures. Coercion resistance can be achieved by making vote validity conditional to the use of the correct signing key, in a manner similar to using fake authentication credentials. We encapsulate this primitive in a voting protocol and discuss its implications.

Keywords: coercion resistance, decentralized voting, ring signatures, strong designated verifier signatures

1 Introduction

Electronic elections can be conducted like traditional elections or new ways to vote can be invented. While we are far from securing supervised voting [Ber+17], it is worth considering how can new technologies transform the election paradigm.

One of the most interesting methods made possible by remote electronic voting, *self-tallying* elections, were proposed in [KY02], where voters can conduct the elections themselves, without using or trusting tallying authorities. That initial idea received many revisions and improvements [Gro04], with the most efficient one being the Open Vote Network (OV-net)[HRZ10], which was implemented on top of the Ethereum blockchain in [MSH17]. However, smart contracts limitation restricted the number of voters to around forty. Recently, scalability in the OVT was improved [SGY20], at the expense of decentralization though. Instead of using smart contracts for self-tallying, [SGY20] delegates the vote counting functionality to an untrusted authority that performs it off-chain, but provides the computation trace, so that the result can be verified by everybody. This untrusted authority does not rely on private keys and as a result *any* entity with enough local processing power can play this role.

As far as security is concerned decentralized voting schemes should have the following basic properties [Kha+12]: *Perfect ballot secrecy:* In order to learn a

voter's choice all other participants must conspire. *Self-tallying:* All voters and interested third parties can tally the election result from published data. This property provides universal verifiability [Cor+16]. *Dispute-freeness:* The protocol avoids situations were one party (rightly) blames another for breaking the protocol, without providing evidence to support it. This property is related to accountability [KTV10]. *Fairness:* No party can deduce partial results before the voting period has ended. *Robustness:* The voting protocol and result computation cannot be blocked by a corrupted party.

Coercion resistance in decentralized voting is not well researched. The reason is that in such protocols, a coercer can be present during vote counting to '*help*' its victim '*correctly*' count the votes and at the same time make sure that his attack succeeded - i.e. the coerced voter followed his instructions. In fact, [Che+10] proves that universal verifiability cannot coexist with receipt freeness - a weaker form of coercion resistance - unless private channels are available. This however leaves open what can be achieved with private or anonymous channels.

The most practical framework to enable coercion resistance, was proposed in [JCJ05], according to which a coercion resistant scheme should offer not only receipt-freeness, but also defend against randomization, forced-abstention, and simulation attacks. To achieve this goal, [JCJ05] relies on the use of fake credentials for vote authorization. During a registration phase, each voter obtains an anonymous credential that must accompany her vote. Under coercion the voter must create a fake but indistinguishable credential, to temporarily fool the coercer that his commands were followed. However, during tallying only the votes corresponding to the registered credential will be taken into account. This must occur in a manner, not detectable by the coercer, while being publicly verifiable. The assumptions of the JCJ protocol comprise an anonymous channel during vote casting to thwart the forced abstention attack, a moment of privacy for the voter to cast her real vote and that the registration authority is not fully corrupted so that the credential does not leak to the coercer.

The JCJ coercion resistance framework has been applied to many protocols, with the most recent one being [Gro+18], where the coercion resistance functionality has been embedded in a primitive called *Publicly Auditable Conditional Blind Signatures* (PACBS). A previous version called *Conditional Blind Signatures* was defined in [ZGP17]. In more detail, a registration authority, creates a signature that embeds both credentials (registered and actually used) in a way that makes it valid if and only if the same token is used in both phases. Then the signature is checked by the designated verifier. Designated verifier signatures were proposed in [JSI96], as a method to enable a coerced voter to cheat a coercer. The signer (prover), instead of creating a proof that statement Θ is true, creates a proof for Θ or 'I know the private key of the verifier'. In *strong* designated verifier signatures, verifiability is not public and as a result the private key of the verifier should be used during verification. A simple way to create them is by encrypting the signature with the public key of the verifier. In [Gro+18] they are used in 'reverse mode': to notify the tallier if the vote should be counted or not. In order to avoid adversarial registration and tallying authorities evidence in

the form of non interactive Σ-protocols are provided. However, PACBS operates in a centralized environment, were authorization and tallying are conducted by two well defined authorities. A similar idea has been proposed in [PS17] using ring signatures and only a single tallying authority. However in that proposal the coercion resistance mechanism remains unspecified.

2 Our proposal

We propose a new cryptographic primitive, *Conditional Designated-Verifier Ring Signatures* (CDVRS), that extends PACBS in the decentralized setting. A designated verifier must still exist, however it is not required that there is an authority that always assumes this role. We also describe how CDVRS can be used to provide coercion resistance in decentralized voting. Due to space limits, we provide a unified description of CDVRS and the voting protocol. In the full version of this work, we plan to separately and formally define and analyse them.

2.1 Setup

Our scheme operates in a group $\mathbb{G}$ of order q, where the DDH assumption is supposed to hold. Instead of a registration procedure, we assume that each voter has a credential consisting of a private part and its public counterpart. This is a common assumption in all self-tallying schemes. In particular, we consider n voters with private keys $\{\mathsf{sk}_i = x_i\}_{i=1}^n \in \mathbb{Z}_q$ and corresponding public keys $\{\mathsf{pk}_i = y_i = g^{x_i}\}_{i=1}^n \in \mathbb{G}$. Votes are encoded as group elements. We also assume two random oracles $\mathsf{H}_G, \mathsf{H}_q$ that map binary strings to $\mathbb{G}, \mathbb{Z}_q$, respectively.

Voters are arranged in rings and the votes of each ring are counted by a tallier who acts as the designated verifier for the ring. A sortition mechanism, like in [Gil+17] can be used to assign voters to rings and select the tallier at random. Alternatively, in the case where the protocol is executed over a Bitcoin [Nak09] like blockchain the proof of work mechanism can be used. More specifically the participants locally run an algorithm, until their output matches some characteristics (e.g. number of zeros) of the proof of work target. These mechanism have the goal to deter participants from conspiring to create rings and select a designated verifier and will be explored in the full version of this work.

2.2 Voting

During vote casting, the voter decides on her choice m and signs it using CDVRS. If the voter is under coercion she does not use her regular private key, but a randomly selected one. In her moment of privacy (akin to [JCJ05]) she uses her regular private key. As a result, a vote that is accompanied by an invalid signature is considered coerced and therefore not counted. Signature verification is not public, but tied to a specific verifier identified by a key.

We now describe the signing and verifying algorithms of CDVRS, which are a combination of the schemes in [LWW04; SWP04; PS17]. The ring L, where

the voter belongs, consists of n_L public keys, namely $L = \{y_i\}_{i=1}^{n_L}$. The signer's index in the ring is π. We denote the designated verifier's private key by x_D with corresponding public key $y_D = g^{x_D}$.

Signing In order to generate the signature for message m, the signer invokes the $\mathsf{Sign}_{\mathsf{sk}_\pi, L, \mathsf{pk}_D}$ functionality which consists of the following steps:

- The signer computes $h := \mathsf{H}_{\mathbb{G}}(L)$ and $\hat{y} := h^{x_\pi}$ if the signature should be valid and $\hat{y} := h^x$ for $x \leftarrow_{\$} \mathbb{Z}_q$ if not (coercion).
- The ring part of the signature is in fact a proof of the statement $\mathsf{DLOG}_g(y_\pi) = \mathsf{DLOG}_h(\hat{y})$ OR $y_\pi \in L$, which is Chaum - Pedersen Σ-protocol [CP92]. The voter follows [CP92] for y_π and simulates the rest of the proofs.
- The signer picks $u \leftarrow_{\$} \mathbb{Z}_q$ and computes:

$$c_{\pi+1} := \mathsf{H}_q(L, \hat{y}, y_D, g^u, h^u, m)$$

- For $i \in \{\pi+1, ..., n_L, 1, ..., \pi-1\}$, the signer picks $s_i \leftarrow_{\$} \mathbb{Z}_q$ and computes:

$$\begin{aligned} T_i &:= g^{s_i} y_i^{c_i} \\ Z_i &:= h^{s_i} \hat{y}^{c_i} \\ K_i &:= y_D^{s_i} \\ c_{i+1} &:= \mathsf{H}_q(L, \hat{y}, y_D, T_i, Z_i, m) \end{aligned}$$

- Finally, the voter computes $s_\pi := u - c_\pi x_\pi$ if she is not coerced or $s_\pi := u - c_\pi x$ if she is, and also computes $T_\pi := g^{s_\pi} y_\pi^{c_\pi}$, $Z_\pi := h^{s_\pi} \hat{y}^{c_\pi}$, $K_\pi := y_D^{s_\pi}$.
- The signature is $\mathsf{Sig}_L(m) := (c_1, \{T_i\}_{i=1}^{n_L}, \{Z_i\}_{i=1}^{n_L}, \{K_i\}_{i=1}^{n_L}, \hat{y})$

Verifying To verify the signature $\mathsf{Sig}_L(m) = (c_1, \{T_i\}_{i=1}^{n_L}, \{Z_i\}_{i=1}^{n_L}, \{K_i\}_{i=1}^{n_L})$ the designated verifier invokes the $\mathsf{Vf}_{L, \mathsf{pk}_D}$ algorithm which:

- Recomputes $h := \mathsf{H}_{\mathbb{G}}(L)$.
- For all group members indexed by $i \in [n_L]$ it computes:

$$c_{i+1} := \mathsf{H}_q(L, \hat{y}, y_D, T_i, Z_i, m)$$

- The signature verifies if the following relations hold:

$$\begin{aligned} c_1 &= c_{n+1} \\ (T_i y_i^{-c_i})^{x_D} &= K_i \quad \forall i \in [n_L] \end{aligned}$$

Note that the ring part of the signature can be independently and publicly verified. As a result, the coercer can learn if the signature belongs to the ring (i.e. the correct private key has been used). This is a shortcoming, but it can be dealt with, by converting the signature to a strong designated verifier signature through encryption of T_i, Z_i with the public key of the designated verifier. The same applies to the pseudoidentity $\hat{y}$. As a result the signature becomes $\mathsf{Sig}_L(m) := (c_1, \{\mathsf{Enc}_{y_D}(T_i)\}_{i=1}^{n_L}, \{\mathsf{Enc}_{y_D}(Z_i)\}_{i=1}^{n_L}, \{K_i\}_{i=1}^{n_L}, \mathsf{Enc}_{y_D}(\hat{y}))$

2.3 Tallying

Subsequently the voter sends the pair $m, \mathsf{Sig}_L(m)$ to the designated tallier. It decrypts and validates the signatures and counts the votes corresponding to the ones that are valid. Apparently, they post their partial tallies, and everybody can merge the results and create the final tally. This can be done, publicly in a manner similar to [HRZ10; SGY20], by disclosing the computation steps.

After each tallier has it's list of valid votes, she chooses a different private key x for each entry of her list and publishes the public keys $y = g^x$ to an authenticated public channel (e.g. an Ethereum smart contract). Let $Y = \{y_i\}_{i=1}^k$ be the list of all public key that correspond to all valid votes of all talliers. Each tallier computes h_i for each valid vote he received:

$$h_i = \frac{\prod_{j\in\{1,\cdots,i-1\}} y_j}{\prod_{j\in\{i+1,\cdots,k\}} y_j}$$

And encrypts each vote $v_i \in \{0, 1\}$ using h_i:

$$b_i = h_i^{x_i} \cdot g^{v_i}$$

Finally, she publishes b_i along with a proof of the correct construction of h_i and a proof that $v_i \in \{0, 1\}$ as in [HRZ10].
Self Tallying: Everyone can compute the result of the elections by computing V:

$$V = \prod_{i\in[k]} b_i = \prod_{i\in[k]} h_i^{x_i} \cdot g^{v_i} = g^{\sum_{i\in[k]} v_i}$$

And then compute the discrete logarithm $v = \log_g(V)$, which can be done in reasonable time for the scale of a typical election.

In future work, we plan to define the threshold version of CDVRS to restrict the power of the talliers.

3 Security issues and future work

This first draft leaves much to be desired. We now detail security issues and assumptions we identify with this simple description.

Firstly, the security model of the CDVRS must be formally defined and its security properties proved. More importantly the actions of the designated verifier must be made auditable. We stress again that the reason a designated verifier is required, is to implement the private channels required by [Che+10] so that coercion resistance is not impossible. If the conditional signature is publicly verifiable then the coercer can himself check if it is valid, or not and hence if the vote is counted. As a result his attack succeeds with certainty. However this has the effect that the designated tallier should be trusted, which makes the scheme unverifiable. For instance the verifier could disregard the signature completely and decide to count the vote or not, like in [ZGP17]. This can be achieved using

non interactive Σ-protocols as in [Gro+18], to accompany all computations for the verification of signatures. These prove that the verifier followed the protocol, without leaking the secret information of whether the signature was valid.

Then the voting protocol must be also formally defined and analyzed. To begin with, we must identify the minimal assumptions that are required for coercion resistance. For instance, we must assume that the coercer is not tracking the voter during the time the credentials were created. Such an assumption is sound, for the following reasons: Firstly, if the coercer always controls the voter, then the coercer essentially becomes the voter. Besides such an attack has a limited scale. Secondly, the registration phase can take place independently of the election, so coercion at that time makes no sense, as the issue of the election is not decided yet. In a first attack, the coercer could require that the voter discloses her private credential so that he compares the public counterpart on his own in order to find out if he is deceived or not. This can be achieved by creating the signatures through a trusted tamper - resistant hardware module, so that the coercer cannot have access to the private key. Such tamper resistant hardware modules have been used in many supervised e-voting schemes [HW14] as well as in a blockchain based scheme [Dim19]. In order to notify this module about whether the signature it must generate should be valid, an authentication mechanism such as panic passwords from Selections [CH11], could be used. When the voter sets up the token, she provides the standard password and a set of different passwords that indicate coercion. If the voter provides the standard password, authentication is considered successful and the signature is generated according to the protocol. If one of the registered panic password is provided, then the module knows that the coercer is watching the voter and generates an invalid signature, using a random private credential, without altering the user experience, so that this case is not be distinguishable from normal user interactions. Finally, in all other cases the module considers the authentication process unsuccessful and requests the user to re-type the password. Another issue with coercion resistance might be the choice of talliers. If the voter is unlucky and her vote is sent to a counter corrupted by the coercer, then incoercibility is lost for that particular voter. As we mentioned, coercion resistance is not a property claimed for decentralized voting schemes. Except for the impossibility result of [Che+10], another reason for this is that such schemes were usually applied to boardroom voting scenarios, where there are only a few voters, which know each other. This makes coercion easier to achieve.

Vote privacy is secured by the anonymity provided by ring signatures. The voter identity is hidden inside the ring. As a result, the vote could be unencrypted which allows for versatility when it comes to the social choice function that decides the winner. Another feature of ring signatures, is linkability, which protects the system from double voting. If the voter reuses the same credential the signatures become linked, and the vote should not be counted by the tallier. However, we have sacrificed their public verifiability for coercion resistance, double voting will be made known only to the designated verifier, who must provide proofs for not counting a vote. Furthermore a malicious voter can send two votes

to different verifiers and thus manage to double vote. Special care could be given to thwart this attack.

Finally, in order to implement the above scheme a realization of a broadcast channel is required. The most promising candidate for this is the Ethereum blockchain, where voting schemes have also been implemented. It is an open question if the primitives described in this paper, could be made compatible with the restrictions posed by smart contracts.

All these issues will be explored in detail in future works.

References

[CP92] David Chaum and Torben P. Pedersen. "Wallet Databases with Observers". In: *CRYPTO '92*. Vol. 740. LNCS. Springer, 1992, pp. 89–105.

[JSI96] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. "Designated Verifier Proofs and Their Applications". In: *EUROCRYPT '96*. Vol. 1070. LNCS. Springer, 1996, pp. 143–154.

[KY02] Aggelos Kiayias and Moti Yung. "Self-tallying Elections and Perfect Ballot Secrecy". In: LNCS 2274 (2002), pp. 141–158.

[Gro04] Jens Groth. "Efficient Maximal Privacy in Boardroom Voting and Anonymous Broadcast". In: *FC 2004*. Vol. 3110. LNCS. Springer, 2004, pp. 90–104.

[LWW04] Joseph K Liu, Victor K Wei, and Duncan S Wong. "Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract)". In: *LNCS* 3108.7200005 (2004), pp. 325–335. ISSN: 03029743.

[SWP04] Ron Steinfeld, Huaxiong Wang, and Josef Pieprzyk. "Efficient Extension of Standard Schnorr/RSA Signatures into Universal Designated-Verifier Signatures". In: *Public Key Cryptography*. Vol. 2947. LNCS. Springer, 2004, pp. 86–100.

[JCJ05] 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*. ACM, 2005, pp. 61–70.

[Nak09] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: *Cryptography Mailing list at https://metzdowd.com* (2009).

[Che+10] Benoît Chevallier-Mames et al. "On Some Incompatible Properties of Voting Schemes". In: *Towards Trustworthy Elections, New Directions in Electronic Voting*. Vol. 6000. LNCS. Springer, 2010, pp. 191–199.

[HRZ10] Feng Hao, Peter Y. A. Ryan, and Piotr Zielinski. "Anonymous voting by two-round public discussion". In: *IET Information Security* 4.2 (2010), pp. 62–67.

[KTV10] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. "Accountability: definition and relationship to verifiability". In: *ACM CCS 2010*. ACM, 2010, pp. 526–535.

[CH11] Jeremy Clark and Urs Hengartner. "Selections: Internet Voting with Over-the-Shoulder Coercion-Resistance". In: *FC 2011*. Vol. 7035. LNCS. Springer, 2011, pp. 47–61.

[Kha+12] Dalia Khader et al. "A Fair and Robust Voting System by Broadcast". In: *5th International Conference on Electronic Voting 2012, (EVOTE 2012)*. Vol. P-205. LNI. GI, 2012, pp. 285–299.

[HW14] Sven Heiberg and Jan Willemson. "Verifiable internet voting in Estonia". In: *6th International Conference on Electronic Voting, EVOTE 2014*. IEEE, 2014, pp. 1–8.

[Cor+16] Véronique Cortier et al. "SoK: Verifiability Notions for E-Voting Protocols". In: *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*. IEEE Computer Society, 2016, pp. 779–798.

[Ber+17] Matthew Bernhard et al. "Public Evidence from Secret Ballots". In: *E-VOTE-ID*. Vol. 10615. LNCS. Springer, 2017, pp. 84–109.

[Gil+17] Yossi Gilad et al. "Algorand: Scaling Byzantine Agreements for Cryptocurrencies". In: *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*. ACM, 2017, pp. 51–68.

[MSH17] Patrick McCorry, Siamak F. Shahandashti, and Feng Hao. "A Smart Contract for Boardroom Voting with Maximum Voter Privacy". In: *FC 2017*. Vol. 10322. LNCS. Springer, 2017, pp. 357–375.

[PS17] Stefan Patachi and Carsten Schürmann. "Eos a Universal Verifiable and Coercion Resistant Voting Protocol". In: *Second International Joint Conference, E-Vote-ID 2017*. Vol. 10615. LNCS. Springer, 2017, pp. 210–227.

[ZGP17] Alexandros Zacharakis, Panagiotis Grontas, and Aris Pagourtzis. "Conditional Blind Signatures". In: *Short version presented in 7th International Conference on Algebraic Informatics - CAI 2017* 2017 (2017). IACR eprint report 2017/682, http://eprint.iacr.org/2017/682, p. 682. URL: `http://eprint.iacr.org/2017/682`.

[Gro+18] Panagiotis Grontas et al. "Towards Everlasting Privacy and Efficient Coercion Resistance in Remote Electronic Voting". In: *FC 2018 Workshops, BITCOIN, VOTING, and WTSC, Revised Selected Papers*. Vol. 10958. LNCS. Springer, 2018, pp. 210–231.

[Dim19] Tassos Dimitriou. *Efficient, Coercion-free and Universally Verifiable Blockchain-based Voting*. Cryptology ePrint Archive, Report 2019/1406. `https://eprint.iacr.org/2019/1406`. 2019.

[SGY20] Mohamed Seifelnasr, Hisham S. Galal, and Amr M. Youssef. "Scalable Open-Vote Network on Ethereum". In: *IACR Cryptology ePrint Archive* 2020 (2020), p. 33.