Technical University of Vienna

# CryptoVote 0.1 - Seminar on Cryptography

Mathias Wolf[*]

Vienna, April 24, 2018

This article explains the core functionality of the freely available distributed ledger protocol CryptoVote. CryptoVote is a protocol that allows the creation and execution of fully decentralized election processes by combining the blockchain with elaborated cryptography.

## 1 Introduction

One major idea behind cryptocurrencies that formally encouraged hundreds of smart and creative minds to work with this early technology was the idea of introducing a new monetary system. A monetary system, where no one could be excluded or be banned from it. A Monetary system, where the majority of its users control the core protocol and and define the rules. Finally a monetary system knowing no country borders, suitable for our nowadays interconnected worlds trading place. One of the cryptocurrencies that still appears to carry this ideas is 'Monero' which originated from a protocol called CryptoNote written by Nicolas van Saberhagen in October 2013 [6]. This CryptoNote 2.0 paper is basically the foundation of this project and hence the name giver. With deep appreciation and gratitude to all the programmers and mathematicians on whose shoulders we stand on.

## 2 The CryptoVote Technology

CryptoVote at its current state of development is a simple blockchain based peer-to-peer protocol. Instead of using consensus to agree on a certain token distribution state, it aims to guarantee a consensus on the state of a voting. As base signature algorithm

[*] mailto:e01315079@student.tuwien.ac.at

we use the Eduards Digital Signature Algorithm (EdDSA) which was developed and implemented by D.J. Bernstein et al [3]. As in the CryptoNote paper we use the same terminology defining $H_s$ as a cryptographic hash function $H_s : \{0,1\}^* \mapsto \mathbb{F}_q$ where $q$ is the prime oder of the Field $\mathbb{F}_q$. The function $E$ describes the curve, and $E(\mathbb{F}_q)$ is the set of all points the curve reaches within $\mathbb{F}_q$. $G$ is a basepoint of the curve. The deterministic hash function $H_p$ guarantees a mapping onto a curve point. Also we introduce our own notion of a reference pointer. If $A$ is some arbitrary data, $\overline{A}$ is a unique short representation that can be used for signing or referencing $A$.

The requirements on a voting system to be a serious alternative to current centralized voting procedures are:

1. Authenticity and forgery resistance of votes

2. Guaranteed anonymity of the voter s.t. no one can tell, if one participated in a voting or not.

3. Non-Excludability of any authorized voter.

4. No duplication and multiple voting.

5. Hiding the state of a voting until the voting period is over.

1) follows from the discrete logarithm problem which is known to be very hard. For 2), anonymity is created by using Linkable Spontaneous Anonymous Group signatures introduced by Liu et.al [5] and optimized by Adam Back [1]. In 3), the 'Non-Excludability' results mainly from the anonymity criteria and the distribution of interests among miningpools. Point 4) is achieved by attaching a key-image $I = xH_p(P)$ to each signature of a vote-transaction, where $P$ is the public and $x$ the corresponding signers private key. Point 5) is achieved by using a slightly modified version of the single address (or stealth address) as in [6]. The sender generates a secret $r$ and sets his voting output to $P = H_s(rA)G$ where $A$ is the candidates address. He then attaches $R = rG$ to his transaction. If the candidates secret $a$ is revealed, where $A = aG$, each node then can resolve $P$ since $aR = arG = rA$, and therefore compute the votes actual destination.

## 2.1 Decision Making Algorithm

The decision making algorithm is essential for a distributed ledger to ensure integrity, correctness and a certain degree of finality. At this state of the CryptoVote project, only proof-of-work(PoW) and proof-of-authority (PoA) based consensus mechanisms have been taken into consideration so far. The current implementation runs on PoW. PoW yields the advantage of decentralized 'autonomous' and 'unstoppable' protocol progression. Yet the miners can affect the outcome of a vote by selectively picking and rejecting votes. Using stealth-addresses reduces the miners ability to do so, up to a certain degree. Since they don't know where a vote goes to, until the secrets of the candidates are revealed. PoA protects the system from strong attackers trying to fork the chain under the assumption that the authority itself is not malicious. Also the electricity consumption of the entire protocol then is negligible small in comparison to PoW based systems. The authorities ability for selective mining and hence influencing voting-results can be reduced up to a degree similar to the PoW scheme.

## 2.2 Transaction Types

The CryptoVote Protocol has validation rules for four different types of transactions, that need to be included into the chain in their following listed order to describe a complete voting-process:

1. **Votingset (VS)**: A transaction containing a set of public keys $\{P_i\}_i$ where $P_i = p_i G$ and $P_i \in E(\mathbb{F}_q)$.

2. **Create Vote (CV)**: A transaction to set up a new voting. The key parameters are:
   - Candidates: Set of public keys representing all possible voting candidates.
   - Authorized Voters: Set of one or more $\overline{\mathbf{VS}}_i$ to specify all allowed voters. Each $\mathbf{VS}_i$ needs to be on chain for **CV** to be valid.
   - Reveal Needed: a flag to specify weather the votes onto this **CV** are expected to be linkable or unlinkable. In case of linkable, anyone can spot at any time the state of the voting. Votes on candidates that are not listed can be rejected then. In case of activated unlinkability, a later revealing **RV** is needed to complete the voting and to enable the chain participants to compute the voting result. Votes on unlisted candidates can't be rejected during the voting period then.

3. **Vote (V)**: The key parameters are:
   - Reference to a **CV** stored in the public ledger.
   - The destination address **DA**. If **CV** needs no revealing, the destination address is one of the specified candidates of the **CV**.
   - LSAG Signature of $H_s(\mathbf{DA}|\overline{\mathbf{CV}})$ to proof the ownership of a $p_i$ within **VS** referenced by the **CV**.
   - Reveal element $R = rG$, where $r$ is a secret created by the voter. Needed to resolve **DA**, after the voting period has passed (eg. **RV** has been committed to the chain).

4. **Reveal Vote (RV)**:A transaction referencing a **CV**, containing the secrets of one or all Candidates to enable public resolving of the unlinkability.

Having votingssets **VS** which can be referenced by various **CV**'s in different constellations yields the advantage of re-usability.

# 3 Security

The current CryptoVote protocol does not have a token system implemented to reward miners. Yet the incentives model shifts under the assumption that the willingness to participate constructively for a miner is likely to be reduced onto his/her votes he/she cares about. This leads to a highly variable network load. The incentives to disturb the system change also. A possible attack scenario emerges from the systems non-finality. E.g. once the **RV** has been published to the network, the attacker knows where each vote goes to and could start building a chain where he only includes votes he prefers. At the end of his chain he includes **RV** again and commits his result. If the 'constructive' miners stop to soon after the voting has been revealed, the attacker could easily make his chain the leading chain after some time, and thereby alter the results. A possible solution to this problem could be the inclusion of a blockhash to the **RV** transaction, that appeared $N$-blocks after the **CV** has been included. This introduces partial finality, if the nodes consider the blockchain unforkable up until **RV** is included. This would give the ones who are expected to reveal a voting a strong element to influence the systems consensus. Ideally the workload on the entire protocol is high enough s.t. Bitcoins security assumptions hold. Then an attacker could only alter voting outcomes with exponentially decreasing depth into the chain as long as he has $< 50\%$ of the entire computing-power. The partial-finality introduced before then would be unnecessary again.
The security model for the PoA decision making algorithm depends mainly on the authorities impartiality of each votings outcome. Since the authority is always able to reject a **RV** transaction if it does not 'like' the results, the PoA model breaks here. The users could overcome the manipulating authority and continue with a newly selected authority [1]. Currently ringsignatures are made with each member of an entire **VS**, to create untraceability. This might lead to unnecessary big signatures. Defining a lower boundary of including a least 7 co-signers (as its currently done in Monero) should be fine. Also measures against IP tracking needs to be implemented later on, to avoid device and endpoint based identification of voters via network traffic analysis. This could be done by using 'Kovri' [4] in later states of developmentt.

Further severe issues so far are: The creator of a votingset can create valid signatures for each one of the voters he included in his set since he knows the voters secret. The creator of a votingset can reconstruct who participated in a voting for the same reason and therefore can check each keyimage appended to a voters signature.

# 4 Outlooks and Ideas

**VS** need a trust-less setup algorithm s.t. only the voters know their secret. **VS** should be editable without the necessity of rebroadcasting unchanged parts.

The creator of the **CV** should be able to formulate constraints like the requirement of having reached a certain age or origin of the authorized voting participants, without

[1] I assume that such dynamic PoA protocols are likely to become as complex and tricky as current proof of stake implementations are, hence I put them aside for now

enforcing them to reveal themselves [2]. A **CV** could easily be designed in a way s.t. each voter gets $V_+$ voting tokens he can spend onto the parties of his favor and $V_-$ negative tokens he can use to discredit unfavored parties in a discrete or analogous manner. For example: Bob has $5V_+$ and $1V_-$. He sends 3.2 to party A, 1.8 to party B and removes 1 point from party C. A **CV** could allow to make first, second, third etc. choices s.t if the first party does not reach a certain threshold, the vote is then automatically passed to the next choice. This could empower democracy since then one can vote on an outsider party, without having the risk of wasting his/her vote. The prevention of spam either requires PoA or transactions fees. This fees could be introduced in form of PoW for each transaction that depends on the transactions size and type. Yet this would be a very weak spam protection.

## 5 Conclusion

We introduced a blockchain based voting system, and named a few of its pros and cons. We explained how cryptography can be used to fulfill the basic prerequisites a voting-system needs to provide and we gave an outlook onto possible extensions of the CryptoVote protocol. Lets conclude by saying that blockchain tech is exciting and promising as we are still far from knowing all of its fields of application that might turn out to be truly useful. However the author personally appreciates the idea of using consensus systems for decentralizing and democratizing voting, money and knowledge. Lets find out how this could be done.

## References

[1] Adam Back. Ring signature efficiency. https://bitcointalk.org/index.php?topic=972541.msg10619684#msg10
2015. [Online; accessed 19-April-2018].

[2] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 459–474. IEEE, 2014.

[3] Daniel J Bernstein. Curve25519: new diffie-hellman speed records. In *International Workshop on Public Key Cryptography*, pages 207–228. Springer, 2006.

[4] Monero devs. The kovri i2p router project. https://getkovri.org/, 2017. [Online; accessed 19-April-2018].

[2] This sounds like it could be done with Bulletproofs or zkSNARKS [2], but therefore I still have a long way to go. But I truly like the idea of replacing votingsets that define certain properties explicitly like: Votingset $XYZ$ contains 'Chileans' that were born 'before Salvador Allende retired', with a zero-knowledge-proof to the **CV**, proofing that one meets all voting prerequisites. The **CV** then only defines which 'authorization oracles' it considers to be trustworthy.

[5] Joseph K Liu, Victor K Wei, and Duncan S Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In *Information Security and Privacy*, pages 325–335. Springer, 2004.

[6] Nicolas van Saberhagen. Crypto note v 2.0. *CryptoNote. org.[Online]*, 17(10), 2013.