Improvements in Everlasting Privacy: Efficient and Secure Zero Knowledge Proofs*

Thomas Haines and Clementine Gritti

NTNU, Trondheim, Norway (thomas.haines),(clementine.gritti)@ntnu.no

Abstract. Verifiable electronic voting promises to ensure the correctness of elections even in the presence of a corrupt authority, while providing strong privacy guarantees. However, few practical systems with end-to-end verifiability are expected to offer long term privacy, let alone guarantee it. Since good guarantees of privacy are essential to the democratic process, good guarantees of everlasting privacy must be a major goal of secure online voting systems. Various currently proposed solutions rely on unusual constructions whose security has not been established. Further, the cost of verifying the zero knowledge proofs of other solutions has only been partially analysed. Our work builds upon Moran and Naor's solution—and its extensions, applications and generalisations—to present a scheme which is additively homomorphic, efficient to verify, and rests upon well studied assumptions.

Keywords: Voting · Everlasting Privacy · Zero Knowledge Proofs.

1 Introduction

Electronic voting schemes have been studied extensively and ongoing research has developed schemes with increasingly strong privacy and integrity guarantees. However, at present the literature has few solutions which are simultaneously efficient, practical, and ensure the ongoing—also called everlasting—privacy of elections. By practical we mean solutions which are easy to deploy securely. Much of the existing literature relies on trusted setup or complicated recovery procedures which reduce the trustworthiness of the election.

Many schemes have sketched how to do elections with everlasting privacy. The constructions tend to use perfectly hiding commitment schemes and public key encryption; this is made verifiable by use of Zero Knowledge Proofs (ZKPs) for correct encryption and correct shuffling of ballots. At present, one of the most common commitment schemes used is not proven secure [26]. A possible method of mixing has been suggested but the security proof is missing [12]. Further, the suggested method of mixing is not sufficiently practical. The importance of everlasting privacy has been widely recognised and prior works present constructions with competing efficiency.

* This is the full version of a paper to appear at E-Vote-ID 2019. The authors acknowledge support from the Luxembourg National Research Fund (FNR) and the Research Council of Norway for the joint project SURCVS.

We want an electronic voting system with everlasting privacy, which is also efficient to run. We introduce the following mechanisms that will enable us to design such a solution, namely Pedersen commitments [28], Sigma Protocols [10,13] and mix-nets [8].

1.1 Background

A Pedersen commitment [28] is an informational-theoretic hiding and computational binding commitment scheme. It provides privacy regardless of the computational power of the adversary but its binding property reduces to the Discrete Logarithm (DLOG) problem. Pedersen commitments are popular in electronic voting schemes because the binding property is only relevant during the course of the election, but privacy should be assured even after the election.

Multiparty computation [35] allows the secure evaluation of a function without leaking anything more about the inputs than can be derived from the result and the inputs previously known to the adversary. ZKPs [19] are a powerful technique which allows proving the correctness of a statement without leaking any other information. The application of both multiparty computation and ZKPs to voting is obvious and commonly mentioned [12,13]. However, the general strategies for both techniques are too computationally intensive in most real elections. Hence there are tailored solutions (such as those we present here) which take advantage of the particularities of elections to construct more efficient solutions.

Sigma Protocols [10,13] are a class of protocols known to be secure under composition. They tend to be more efficient than zero knowledge protocols. A protocol of the correct form is proved to be a Sigma Protocol by showing it satisfies the following properties: *completeness*, capturing that the protocol will succeed when both parties are honest; *special soundness*, referring to the inability of the adversary to generate proofs without knowing a witness; and *honest verifier zero knowledge*, emphasising that the proof leaks negligible information.

Mix-nets were first proposed by Chaum [8], as a way to provide privacy. In the context of verifiable electronic voting mix-nets are also required to be verifiable. This is achieved by proving the correctness of the shuffle using a ZKP, of which two techniques are dominant; namely those of Bayer and Groth [4] and that of Terelius and Wikström [32]. Both techniques are general in nature and tend to be optimised for the particularities of the system in which they are used.

A verifiable mix-net is not sufficient to provide privacy in an electronic voting scheme. As Cortier et al [9] demonstrated, a property called ballot independence or, more powerfully, non-malleability, is often required to achieve privacy. Bernhard et al [5] showed that the Enc+PoK construction converts a IND-CPA homomorphic encryption scheme into one which is voting-friendly when a straight-line extractor exists [18]. We suggest the simpler solution suggested by [11,14] of using unique identifiers in the challenge, or challenge generation if using the Fiat-Shamir heuristic, to provide ballot independence. For such a solution to be efficient there must exist efficient zero-knowledge proofs that the ballot is correctly formed, which is precisely one of our contributions.

1.2 Related Work

Much of the everlasting privacy literature relies on and builds upon Moran and Naor's work [26], which was modified as an extension to the web-based voting Helios scheme [17]. This kind of extension reduces privacy attacks on the system (from an external adversary) to information theoretic security rather than computational. Hence, no future breakthrough in computation power, mathematics, or large-scale quantum computers will put the voters' privacy at risk. While there are schemes which provide information theoretic maximal privacy (That is, the adversary learns nothing more about the honest voters' input than it learned from the result and its own input) these are impractical for most real elections. Moran and Naor's scheme and many others, including ours, have at least one (sometimes threshold of) authorities against which privacy holds only computationally. Overwhelmingly everlasting privacy schemes rely on commitments and blinded values and this work is no exception.[1]

Unfortunately, the bulk of this work relies on primitives which are somewhat unusual. Since Moran and Naor, a Pedersen commitment variant is often used but its security appears never to have been rigorously established. Indeed, there is much literature which states that Pedersen commitments and Sigma Protocols are generally required to be defined in a prime order group, which this variant is not, meaning its security should be rigorously established [3,7,29]. We denote, in the paper, the combination of Paillier encryption [27] and Pedersen commitments [28], pioneered by Moran and Naor, as the MN encryption scheme.

Arapinis *et al.* [2] recently showed in ProVerif, an automatic cryptographic protocol verifier, that various constructions achieve everlasting privacy, some of these solutions lose verifiability properties in exchange for everlasting privacy but are highly practical in those situations where these verifiability properties are not important. Cuvelier *et al.* [12] systematised much of the research by showing how certain types of primitives can be securely combined. They also present an elegant scheme called PPATC based on Abe *et al.*'s [1] commitment scheme on bilinear pairings, which they show has efficient encryption on the order of 40 times faster then existing methods. The efficiency is due to the elliptic curves which are more secure relative to their size than problems based on factorisation.

However, Cuvelier *et al.* [12] do not account for the verification complexity. We show that Moran-Naor suggestion of Paillier encryption and Pedersen commitments—refereed as PPATP in [12]—is at least as fast to verify as PPATC when using the Sigma Protocol and mix-net we will detail later. Further, the MN system supports homomorphic tallying where PPATC does not which is a significant advantage in some situations. We note that Cuvelier *et al.* [12] do sketch the same Sigma Protocol for correct encryption in their paper that we later present, but provide no proof. We also note that recent work of Hazay *et al.* [23], has made threshold key generation in Paillier practical as with PPATC.

[1] Another set of schemes uses some form of anonymous signature (ring, group, linkable) and an anonymous channel [22,24,25] which achieves everlasting privacy cleanly, but the existence of such a channel is problematic to realise. [34]

Many of the existing solutions—except Cuvelier *et al.* [12]—are unsatisfactory in one of two ways. They complicate practical issues, by detecting issues after they have occurred rather than using ZKPs initially. Alternatively, they rely on cut-and-choose based ZKPs rather than Sigma proofs, resulting in an increase in computation and communication of about six orders of magnitude.

There are efficient mix-nets for both Paillier ciphertexts and Pedersen commitments (e.g., Moran and Naor highlight Groth's mix-net working for Paillier encryption scheme [20]). However, mixing the commitments and ciphertexts separately significantly complicates the election process and weakens security. Cuvelier *et al.* note that the general construction of Wikström [33] can be applied but do not prove the required Sigma Protocol. Further, this construction is significantly slower than the optimised constructions popular in electronic voting.

1.3 Contributions

Our main contribution is to rigorously establish various efficient and practical variants of known primitives which are particularly well suited for use in electronic voting. Specifically, our contributions are:

- We present the Sigma Protocol for re-encryption of the MN cryptosystem; we also provide the proof for this Sigma Protocol and for the protocol for correct encryption [12] of the MN cryptosystem;
- We provide the first proof of security for the existing modified Pedersen commitment of semi-prime order;
- We present an efficient variant of ballot mixing;
- We give an analysis of verification efficiency of MN cryptosystem and compare with PPATC, showing MN is as fast to verify when using the mix-net and Sigma Protocols from above.

When Moran and Naor first introduced the MN cryptosystem they said "although more efficient (zero knowledge) protocols exist for these applications, for the purpose of this paper we concentrate on simplicity and ease of understanding" [26]. Unfortunately in the decade since the follow up work has continued to rely on cut-and-choose [6,17]; and, has found updating the existing zero knowledge work to the requirements of the MN cryptosystem more difficult than Moran and Naor expected. Our contribution finally closes this gap by providing efficient proofs for encryption, re-encryption and shuffling.

Since various electronic voting systems, both in-booth and online, have already been presented based on these primitives [12,17,26], our contribution immediately implies several efficient and secure verifiable voting schemes.

1.4 Road Map

In the next section, we provide the notations and definitions useful for the comprehension of the paper. In Section 3, we present our security proof for the modified Perdersen commitment scheme [26]. In Section 4, we describe our new Sigma Protocol for re-encryption, and give the security proofs for the latter as

well for the Sigma Protocol for encryption [12]. In Section 5, we depict our verifiable mix-net, improving the efficiency of the general construction proposed in [33]. In Section 6, we analyse and compare the efficiency of our solution with the similar work of Cuvelier *et al.* [12]. We conclude our paper in the last section.

2 Preliminaries and Building Blocks

Due to lack of space, we let the readers refer to [19] for zero knowledge notions, and specifically to [13] for Sigma Protocols, and to [8] for mix-nets.

Notations

Arithmetic Natural numbers are denoted by $\mathbb{N}$ and integers by $\mathbb{Z}$. The ring of integers modulo n is denoted $\mathbb{Z}_n$, and its multiplicative group $\mathbb{Z}_n^*$. Let M denote a square matrix of order N from $\mathbb{Z}_n^{N*N}$. Let $\mathbf{v}$ be a vector of length N from $\mathbb{Z}_n^N$. Let $\langle \mathbf{v}, \mathbf{v'} \rangle = \sum_{i=1}^{N} v_i v'_i$ denote the inner product.

Miscellaneous We let κ denote the main security parameter.We denote by $negl(\kappa)$ any function for which for every constant c and for all sufficiently large κ it holds that $negl(\kappa) < \kappa^{-c}$. We denote 1^κ the string of 1s of length κ, the unary representation of the security parameter. Given a finite set S, $s \leftarrow_r S$ means a uniformly random assignment of an element in S to the variable s. We also use this notation for algorithms which use random coins.

Polynomial-Time Algorithms A Polynomial-Time Algorithm (PPT) , or equivalently an efficient, algorithm is a probabilistic algorithm running in time polynomial in its input size.

Experiments and Advantage An experiment is a game played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. Successes is defined as $Succ^*(\mathcal{A}, \circ) = Pr[Exp^*_{\mathcal{A}}(\circ) = 1]$. The experiment is denoted Exp^{*-b}, where $\mathcal{A}$ must distinguish between two variants of the experiment, and the advantage is defined as $Adv^*(\mathcal{A}, \circ) = Pr[Exp^{*-1}_{\mathcal{A}}(\circ) = 1] - Pr[Exp^{*-0}_{\mathcal{A}}(\circ) = 1]$. In other words, Adv with reference to an experiment denotes the adversary's ability to win with advantage over a random guess.

Relationships A relationship $\mathcal{R}_*(\circ)(\diamond)$ is a subset of the Cartesian product of the sets $\circ$ and $\diamond$. We denote by $\mathcal{R}_1 \vee \mathcal{R}_2$ the relationship consisting of the pairs $((x_1, x_2), w)$ s.t. $(x_1, w) \in \mathcal{R}_1$ or $(x_2, w) \in \mathcal{R}_2$. Let $\mathcal{R}_1 \wedge \mathcal{R}_2$ be the relationship consisting of the pairs $((x_1, x_2), w)$ s.t. $(x_1, w) \in \mathcal{R}_1$ and $(x_2, w) \in \mathcal{R}_2$.

Definition 1. *The (t, ϵ)-DLOG assumption holds in group $\mathbb{G}$ if no t-time algorithm $\mathcal{A}$ has $Succ^{dlog}(\mathcal{A}, \mathbb{G}, g) > \epsilon$ in $Exp^{dlog}_{\mathcal{A}}(\mathbb{G}, g)$ (Fig. 1). For simplicity we will often drop the t and ϵ and refer to the DLOG assumption in $\mathbb{G}$.*

$Exp^{dlog}_{\mathcal{A}}(\mathbb{G}, g)$

$x \leftarrow_r \mathbb{Z}_q$
$X \leftarrow g^x$
$x' \leftarrow \mathcal{A}(\mathbb{G}, g, X)$
if $x' = x$ **then return** 1
else return 0

Fig. 1. DLOG experiment

Discrete Logarithm Assumption Given primes p, q and $n = pq$, where $kn+1$ is also prime, for $k \in \mathbb{N}$. Let $\mathbb{G}_n$ denote the group of order n mod $\mathbb{Z}^*_{kn+1}$ and let $\mathbb{G}_p$, $\mathbb{G}_q$ denote the groups of order p and q respectively mod $\mathbb{Z}^*_{kn+1}$. $\mathbb{G}_p$ and $\mathbb{G}_q$ are called *Schnorr groups*. The prime p must be large enough to prevent specific attacks on this class of cyclic group, such as Index Calculus. The prime q must be large enough to prevent generic attacks that apply to all cyclic groups. A rough concrete suggestion might be q of length 256 bits and p of length 2048 bits. The Discrete Logarithm (DLOG) assumption, definition 1, is believed to hold for the set of Schnorr groups.

Commitment Scheme

Definition 2. *A homomorphic commitment scheme Π is a triple of PPT algorithms $(\Pi.\mathsf{Setup}, \Pi.\mathsf{Com}, \Pi.\mathsf{Open})$, s.t.:*
- The Setup *algorithm for a given group $\mathbb{G}$ defines a set of valid Commit Keys $\mathcal{CK}$ from which one is uniformly selected: $CK \in \mathcal{CK} \leftarrow_r \Pi.\mathsf{Setup}(\mathbb{G})$.*
- A given Commit Key CK defines a message space $\mathcal{M}_{CK}$, randomness space $\mathcal{R}_{CK}$, commitment space $\mathcal{C}_{CK}$, and opening space $\mathcal{D}_{CK}$. The Com *algorithm takes these as domain and co-domain: $\forall m \in \mathcal{M}_{CK}, \forall r \in \mathcal{R}_{CK}$, $(c \in \mathcal{C}_{CK}, d \in \mathcal{D}_{CK}) \leftarrow \Pi.\mathsf{Com}_{CK}(m, r)$.*
- The Open *algorithm takes a commitment $c \in \mathcal{C}_{CK}$ and opening $d \in \mathcal{D}_{CK}$ and returns either a message $m \in M_{CK}$ or null $\perp$: $\Pi.\mathsf{Open}_{CK}(c \in \mathcal{C}_{CK}, d \in \mathcal{D}_{CK}) \rightarrow m \in \mathcal{M}_{CK}$ or $\perp$.*

We adopt multiplicative notation for the commitment space and additive notation for the message space since we will mainly focus on Pedersen style commitments. However, provided both spaces are in fact groups, under certain operations the notation is unimportant.

Correctness: $\forall CK \in \mathcal{CK}, \forall m \in \mathcal{M}_{CK}, \forall r \in \mathcal{R}_{CK}$, we have $\Pi.\mathsf{Open}_{CK}(\Pi.\mathsf{Com}_{CK}(m, r)) = m$.

Homomorphism: $\forall CK \in \mathcal{CK}$, $\forall m_1, m_2 \in \mathcal{M}_{CK}$, $\forall r_1, r_2 \in \mathcal{R}_{CK}$, we have $\Pi.\mathsf{Com}_{CK}(m_1, r_1) * \Pi.\mathsf{Com}_{CK}(m_2, r_2) = \Pi.\mathsf{Com}_{CK}(m_1 + m_2, r_1 + r_2)$. The homomorphic property implies the ability to re-randomise commitments: let the ReRand algorithm be defined as $\Pi.\mathsf{ReRand}_{CK}(c \in \mathcal{CK}_{CK}, r \in \mathcal{R}_{CK}) = c * \Pi.\mathsf{Com}_{CK}(1, r)$.

Definition 3. *Perfectly hiding property of a commitment scheme: Given a group $\mathbb{G}$, a commitment scheme Π is perfectly hiding if for any adversary $\mathcal{A}$, it holds that $Adv^{hiding}(\mathcal{A}, \Pi, \mathbb{G}) = Pr[Exp^{hiding-1}_{\mathcal{A}}(\Pi, \mathbb{G})] - Pr[Exp^{hiding-0}_{\mathcal{A}}(\Pi, \mathbb{G})] = 0$ (Fig. 2).*

$Exp^{hiding-b}_{\mathcal{A}}(\Pi, \mathbb{G})$

$CK \leftarrow_r \Pi.\mathsf{Setup}(\mathbb{G})$
$(m_0, m_1, \alpha) \leftarrow_r \mathcal{A}(CK)$
$r \leftarrow_r \mathcal{R}_{CK}$
$(c, d) \leftarrow \Pi.\mathsf{Com}_{CK}(m_b, r)$
$b' \leftarrow_r \mathcal{A}(CK, c, \alpha)$

Fig. 2. Hiding experiments

Definition 4. *Binding property of a commitment scheme: Given a group $\mathbb{G}$, a commitment scheme Π is (t, ϵ) binding if no t-time algorithm $\mathcal{A}$ has $Succ^{binding}(\mathcal{A}, \Pi, \mathbb{G}) > \epsilon$ in $Exp^{binding}_{\mathcal{A}}(\Pi, \mathbb{G})$ (Fig. 3). For simplicity we will often drop t and ϵ and refer to Π as binding.*

$Exp^{binding}_{\mathcal{A}}(\Pi, \mathbb{G})$

$CK \leftarrow_r \Pi.\mathsf{Setup}(\mathbb{G})$

$(c, d, d') \leftarrow_r \mathcal{A}(CK)$

$m \leftarrow \Pi.\mathsf{Open}_{CK}(c, d)$

$m' \leftarrow \Pi.\mathsf{Open}_{CK}(c, d')$

if $m \neq m'$ **return** 1

else return 0

Fig. 3. Binding experiment

Public Key Encryption Scheme

Definition 5. *A homomorphic public key encryption scheme Σ is a triple of PPT algorithms $(\Sigma.\mathsf{KeyGen}, \Sigma.\mathsf{Enc}, \Sigma.\mathsf{Dec})$, s.t.:*
- The KeyGen *algorithm defines a set of valid key pairs $(\mathcal{PK}, \mathcal{SK})$ from which one is uniformly selected: $(PK \in \mathcal{PK}, SK \in \mathcal{SK}) \leftarrow_r \Sigma.\mathsf{KeyGen}(1^k)$.*
- A given public key PK defines a message space $\mathcal{M}_{PK}$, randomness space $\mathcal{R}_{PK}$, and ciphertext space $\mathcal{C}_{PK}$. The Enc *algorithm takes these as domain and co-domain: $\forall PK \in \mathcal{PK}, \forall m \in \mathcal{M}_{PK}, \forall r \in \mathcal{R}_{PK},\ CT \in \mathcal{C}_{PK} \leftarrow \Sigma.\mathsf{Enc}_{PK}(m, r)$.*
- The Dec *algorithm takes a ciphertext $CT \in \mathcal{C}_{PK}$ and $SK \in \mathcal{SK}$ and returns either a message $m \in \mathcal{M}_{PK}$ or null $\perp$: $\forall CT \in \mathcal{C}_{PK}, \Sigma.\mathsf{Dec}_{SK}(c) \rightarrow m \in \mathcal{M}_{PK}$ or $\perp$.*

Correctness: $\forall (PK \in \mathcal{PK}, SK \in \mathcal{SK}) \leftarrow_r \Sigma.\mathsf{KeyGen}(1^k), \forall m \in \mathcal{M}_{PK}, \forall r \in \mathcal{R}_{PK}$, we have $\Sigma.\mathsf{Dec}_{SK}(\Sigma.\mathsf{Enc}_{PK}(m, r)) = m$.

Homomorphism: $\forall PK \in \mathcal{PK}, \forall m_1, m_2 \in \mathcal{M}_{PK}, \forall r_1, r_2 \in \mathcal{R}_{PK}$, we have $\Sigma.\mathsf{Enc}_{PK}(m_1 + m_2, r_1 + r_2) = \Sigma.\mathsf{Enc}_{PK}(m_1, r_1) * \Sigma.\mathsf{Enc}_{PK}(m_2, r_2)$.

$Exp^{ind-cpa-0}_{\mathcal{A}}(1^\kappa)$	$Exp^{ind-cpa-1}_{\mathcal{A}}(1^\kappa)$
$(PK, SK) \leftarrow_r \Pi.Setup(1^k)$	$(PK, SK) \leftarrow_r \Pi.Setup(1^k)$
$(m_0, m_1, \alpha) \leftarrow_r \mathcal{A}(PK)$	$(m_0, m_1, \alpha) \leftarrow_r \mathcal{A}(PK)$
$r \leftarrow_r \mathcal{R}_{PK}$	$r \leftarrow_r \mathcal{R}_{PK}$
$c \leftarrow \Pi.Enc_{PK}(m_0, r)$	$c \leftarrow \Pi.Enc_{PK}(m_1, r)$
$b \leftarrow_r \mathcal{A}(PK, c, \alpha)$	$b \leftarrow_r \mathcal{A}(PK, c, \alpha)$

Definition 6. *IND-CPA security property of a public key encryption scheme: For a given κ, we say a public key encryption scheme Π is (t, ϵ) IND-CPA secure if no t-time algorithm $\mathcal{A}$ has advantage at least ϵ in $Adv^{ind-cpa}(\mathcal{A}, \kappa)$ (Fig. **??**). For simplicity we will often drop t and ϵ and refer to Π as being IND-CPA secure.*

Informally, a game between a challenger and a adversary against IND-CPA security is run as follows. The adversary receives the public key from the challenger (and the latter keeps the secret key) and is given access to a key generation oracle. After queries to that oracle, the adversary generates two messages of equal length. The challenger decides, randomly, to encrypt one of them. The adversary tries to guess which of the messages was encrypted and wins the game if the guess is correct. A public key encryption scheme is said to be IND-CPA secure if an adversary has negligible advantage in winning the above game.

2.1 Modified Pedersen Commitment Scheme

As we have already noted starting with Moran and Naor [26], Pedersen commitments of semi-prime order have become a significant building block for voting schemes with everlasting privacy. The construction proposed in [26] was to take two safe primes p, q (i.e. to be of the form $2p + 1$ for p prime), let $n = pq$ and work in the subgroup of order n of $\mathbb{Z}^*_{4n+1}$ where $4n + 1$ is also prime.

The modified Pedersen commitment scheme Π is the triple of PPT algorithms $(\Pi.\mathsf{Setup}, \Pi.\mathsf{Com}, \Pi.\mathsf{Open})$, s.t.:
- $CK \leftarrow \Pi.\mathsf{Setup}(\mathbb{G})$ s.t. $CK = \{\mathbb{G}, g, h\}$. Given a group $\mathbb{G}$ of semi-prime order n, let g be any generator of $\mathbb{G}$ and choose $h \leftarrow_r \mathbb{G}$ (with overwhelming probability h will be a generator).
- A given Commit Key $CK = \{\mathbb{G}, g, h\}$ defines the message space $\mathcal{M}_{CK} = \mathbb{Z}_n$, randomness space $\mathcal{R}_{CK} = \mathbb{Z}_n$, commitment space $\mathcal{C}_{CK} = \mathbb{G}_n$, and opening space $\mathcal{D}_{CK} = (\mathbb{Z}_n, \mathbb{Z}_n)$. The $\Pi.\mathsf{Com}_{CK}$ algorithm takes $m \in \mathbb{Z}_n, r \in \mathbb{Z}_n$ and sets $c = g^r h^m$ and $d = (m, r)$.
- The $\Pi.\mathsf{Open}_{CK}$ algorithm takes a commitment $c \in \mathbb{G}_n$ and opening $d \in (m \in \mathbb{Z}_n, r \in \mathbb{Z}_n)$. If $c = g^r h^m$ return m else return $\perp$.

2.2 A Commitment Consistent Encryption System

The encryption scheme suggested by Moran and Naor [26] is a particular kind of encryption system specialised for everlasting privacy, and commonly used in verifiable electronic voting [16,17]. The standard suggestion, which we describe below, is to use Pedersen commitments of semi-prime order and the generalised Paillier cryptosystem. This notation–while slightly unusual–is useful because it enables the direct application of various existing results, particularly those in the area of mix-nets, as we shall see later. For convenience, we shall refer to this system as the MN cryptosystem.

We now describe MN encryption scheme. Let $\Sigma = (\Sigma.\mathsf{KegGen}, \Sigma.\mathsf{Enc}, \Sigma.\mathsf{Dec})$ denote a public key encryption scheme. Specifically let $\Sigma.\mathsf{KeyGen}$ be the key generation function of the (generalised) Paillier cryptosystem [27,15] producing $PK = (n)$ and $SK = (d)$, where $n = pq$ is a RSA modulus and d is the lowest common multiple of $p - 1$ and $q - 1$. Choose k s.t. $kn + 1$ is prime, and let g, h be random generators of subgroup of order n in $\mathbb{Z}^*_{kn+1}$, denoted $\mathbb{G}_n$. We denote the ciphertext space $\mathcal{C}_{PK} = \mathbb{G}_n \times \mathbb{Z}^*_{n^2} \times \mathbb{Z}^*_{n^2}$, the message space $\mathcal{M}_{PK} = \mathbb{Z}_n$, and the randomness space $\mathcal{R}_{PK} = \mathbb{Z}_n \times \mathbb{Z}^*_n \times \mathbb{Z}^*_n$.

We quickly explain the encryption process. Let $\Sigma.\mathsf{Enc}_{PK}(m \in \mathbb{Z}_n, (r \in \mathbb{Z}_n, r' \in \mathbb{Z}_n^*, r'' \in \mathbb{Z}_n^*))$ produce $CT = (c, ct_1, ct_2) = (g^r h^m \bmod kn+1, (1+n)^m r'^n \bmod n^2, (1+n)^r r''^n \bmod n^2)$. That is we encode the message m in a Pedersen commitment hidden by the randomness r, and we encrypt the opening to this commitment in two Paillier ciphertexts. Let $\Sigma.\mathsf{Dec}_{SK}(CT = (c, ct_1, ct_2))$ be the decryption function. First use the Paillier decryption function to retrieve m, r from ct_1, ct_2 respectively, then if $c = g^r h^m$ the result is m else $\perp$.

We first make the observation that the Σ scheme is additively homomorphic, that is $\Sigma.\mathsf{Enc}_{PK}(m_0, (r_0, r'_0, r''_0)) * \Sigma.\mathsf{Enc}_{PK}(m_1, (r_1, r'_1, r''_1)) = \Sigma.\mathsf{Enc}_{pk}(m_0 + m_1, (r_0 + r_1, r'_0 * r'_1, r''_0 * r''_1))$. Secondly, that there is a shuffle friendly map [33]: given $CT = (c, ct_1, ct_2)$ and $r = (r_0, r_1, r_2)$, $c' = c * g^{r_0}, ct'_1 = ct_1 * r_1^n, ct'_2 = ct_2 * (1+n)^{r_0} r_2^n$. We denote this map by $(\phi_{PK}(CT, r) = \mathcal{C}_{PK} \times \mathcal{R}_{PK} \to \mathcal{C}_{PK})$. The existence of this map is necessary to apply Wikström's general mix-net construction to the cryptosystem [33].

In addition, we preserve the property of Paillier encryption and Pedersen commitments that given a ciphertext $CT = \Sigma.\mathsf{Enc}_{PK}(m_0 \in \mathcal{M}_{pk}, (r, r', r'') \in \mathcal{R}_{pk})$ and a message m_1 it is easy to compute $CT^{m_1} = \Sigma.\mathsf{Enc}_{pk}(m_0 * m_1; (r * m_1, r'^{m_1}, r''^{m_1}))$. In this case the exact effect on the randomness is a combination of multiplication and exponentiation. Lastly, since the Pailler variant we use is the variant of Damgård et al [15], threshold decryption is also available.

3 Security Proof for the Modified Pedersen Commitment Scheme

The sketch of the security proof for the commitment scheme in [26] lacks sufficient detail to be of use in establishing the security of the commitment. Since the group n is not of prime order, given a tuple (m, r, m', r') if $GCD(|m - m'|, n) \neq 1$ and $GCD(|r - r'|, n) \neq 1$ then the sketched reduction to the DLOG problem fails. While it is not particularly surprising that the DLOG problem holds in a group whose order contains a large prime factor, it is important to show that this is indeed true and furthermore does not break any other part of the system. A correct reduction is hence needed. Moreover, we do not require the primes to be safe and thus consider a subgroup of order n of $\mathbb{Z}_{kn+1}^*$ for an integer k. Therefore, the above commitment scheme can be extended to the general case with integers k, n such that $kn + 1$ is prime. We now present the security proof of the generalization of the modified Pedersen commitment scheme.

Proposition 1. *The modified Pedersen commitment scheme Π is a homomorphic perfectly hiding commitment scheme.*

Proof. The correctness of the scheme follows immediately from the definitions of $\Pi.\mathsf{Com}$ and $\Pi.\mathsf{Open}$. The perfect hiding property of the scheme follows in the same way as normal Pedersen commitment schemes: for any two messages m_0, m_1 and commitment c there exist two unique random coins r_0, r_1 s.t. $c = g^{r_0} h^{m_0}$ and $c = g^{r_1} h^{m_1}$, and since the random coins are taken uniformly, the commitment provides no information about which message was committed to.

The key to understanding the next part on the binding property is to recall that for a cyclic group of semi-prime order $n = pq$, there are exactly two non-trivial subgroups: one is of order p and the other q. If we let $\mathbb{G}$ be the subgroup of $\mathbb{Z}^*_{kn+1}$ of order n, where $kn+1$ is prime, then the two non-trivial subgroups are two Schnorr groups. The reduction we present in the next paragraph reduces the binding property of the modified Pedersen commitment to the DLOG problem in the two Schnorr groups, which we label $\mathbb{G}_p$ and $\mathbb{G}_q$.

To show that the scheme is binding, we present a reduction in two parts. First, we show that for any t-time adversary $\mathcal{A}$ against the modified Pedersen commitment scheme Π with $Succ^{binding}(\mathcal{A}, \Pi, \mathbb{G}) = \epsilon$, we can construct an algorithm which–given a DLOG problem in $\mathbb{G}_p$, and another in $\mathbb{G}_q$–outputs the answer to at least one with probability ϵ. Then having observed against which of the two groups the better success rate is achieved, we construct an adversary against the DLOG problem in that group which succeeds with probability at least $\frac{\epsilon}{2}$. This suffices to show that the binding property of the commitment scheme cannot be broken with probability more than twice that of the DLOG problem in the weakest of the two underlying Schnorr groups $\mathbb{G}_p$ and $\mathbb{G}_q$.

Given instances of the discrete log problem in $\mathbb{G}_p(g_p, h_p)$, $\mathbb{G}_q(g_q, h_q)$ we can simulate the honest setup in Binding game as $G_n(g_n, h_n)$. Let $g_n = g_p * g_q$ and $h_n = h_p * h_q$ note that since both discrete problem instances are random and the groups are isomorphic the new commitment key is also uniformly random and hence a perfect simulation of honest setup. The challenger takes the two subgroups of $\mathbb{G}_n$ and a DLOG problem in each. It combines these to construct the commitment key which it gives to the adversary. Since g_p and g_q are generators of their respective groups $\mathbb{G}_p$ and $\mathbb{G}_q$, if h_p and h_q are random elements (as they are in the DLOG experiment) than this is indistinguishable from the honest run. We show that if there exists a successful adversary A that breaks the binding property of the commitments than their exists an adversary which breaks the discrete log problem in at least one of the two Schnorr groups with non-negligible probability. The successful adversary $\mathcal{A}(\mathbb{G}, g, h)$ outputs $(c, (m, r), (m', r'))$ s.t. $m \neq m'$. If $GCD(|m - m'|, n) = 1$ or $GCD(|r - r'|, n) = 1$ then we extract $\alpha = dlog_g h$ as normal with Pedersen commitments and calculate $dlog_{g_p} h_p = \alpha \mod p$ and $dlog_{g_q} h_q = \alpha \mod q$. If this is not the case, then w.l.o.g. $GCD(|r - r'|, n) = GCD(|m - m'|, n) = p$ and hence there exists unique $\delta, \gamma \in Z_q$ s.t. $\delta p = \alpha\gamma p \mod n$ and hence $\alpha = \frac{\delta}{\gamma} \mod q$. By the Chinese remainder theorem $\alpha \mod q = dlog_{g_q} h_q$ and we successfully answer that.

□

3.1 Observations

In our use case, the Schnorr groups will have a value q of at least 1024 bits and a value k at least as large so the binding problem would have comparable security to 128 bit symmetric encryption (barring any breakthroughs in cryptanalysis or large scale quantum computers).

Our solution is not only provably secure (under reasonable assumptions) but also more general with the setting $kn + 1$, with $k \in \mathbb{N}$, rather than $4n + 1$. The

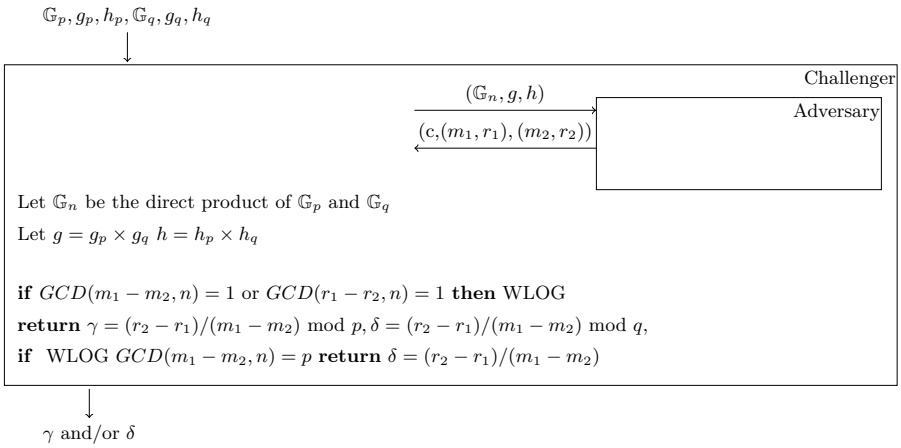

Fig. 4. Reduction from binding to discrete log

homomorphism of the scheme follows immediately from the group properties and the isomorphism of $\mathbb{Z}_n$ and $\mathbb{G}_n$.

4 Security Proofs for Sigma Protocols

We present two Sigma Protocols, one for correct encryption from [12] and a new protocol for correct re-encryption; we believe that proofs of both Sigma Protocols have never been published before. These proofs allow the realisation of an electronic voting scheme that is secure (compared to without ZKPs) and highly efficient (compared to the cut-and-choose solutions currently in the literature).

4.1 Sigma Protocol for Correct Encryption

The following Sigma Protocol for correct encryption was proposed by Cuvelier *et al.* [12], though they omit the proof. Such a protocol is used to prove that given a ciphertext, one knows the inputs and uses them to generate that ciphertext.

Given $CT = (c = g^r h^m \bmod kn + 1, ct_1 = (1+n)^m r'^n \bmod n^2, ct_2 = (1+n)^r r''^n \bmod n^2)$, we show that we know $m \in \mathbb{Z}_n$ and $(r \in \mathbb{Z}_n, r' \in Z_n^*, r'' \in \mathbb{Z}_n^*)$:

1. Let t_1, t_2 be random elements in $\mathbb{Z}_n$ and t_3, t_4 be random elements in $\mathbb{Z}_n^*$. The prover computes $\alpha = g^{t_1} h^{t_2} \bmod kn + 1, \beta = (1+n)^{t_2} t_3^n \bmod n^2, \gamma = (1+n)^{t_1} t_4^n \bmod n^2$ and sends them to the verifier.
2. The verifier sends a challenge ξ chosen at random in $\mathbb{Z}_n$.
3. The prover computes $s_1 = t_1 + \xi r \bmod n, s_2 = t_2 + \xi m \bmod n, s_3 = t_3 * r'^{\xi} \bmod n, s_4 = t_4 * r''^{\xi} \bmod n$, and sends these to the verifier.
4. The verifier accepts if $\alpha c^{\xi} = g^{s_1} h^{s_2} \bmod kn + 1, \beta c t_1^{\xi} = (1+n)^{s_2} s_3^n \bmod n^2, \gamma c t_2^{\xi} = (1+n)^{s_1} s_4^n \bmod n^2$.

The transcript (with the elements exchanged between the prover and the verifier) is $(\alpha \in \mathbb{G}_n, \beta, \gamma \in \mathbb{Z}_{n^2}^*, \xi, s_1, s_2 \in \mathbb{Z}_n, s_3, s_4 \in \mathbb{Z}_n^*)$.

Security Proof

Proposition 2. *The above protocol has perfect completeness, special soundness, and honest verifier zero knowledge and is hence a Sigma Protocol.*

Proof. We now show that **Sigma for Correct Encryption** protocol is correct. For convenience, we let E denote here the Paillier encryption function, and then argue based on the homomorphism of Paillier.

$$\begin{aligned} \alpha c^{\xi} &\stackrel{?}{=} g^{s_1} h^{s_2} \\ g^{t_1} h^{t_2} g^{r\xi} h^{m\xi} &\stackrel{?}{=} g^{t_1+\xi r} h^{t_2+\xi m} \\ g^{t_1+r\xi} h^{t_2+m\xi} &= g^{t_1+r\xi} h^{t_2+m\xi} \end{aligned}$$

$$\begin{aligned} \beta c t_1^{\xi} &\stackrel{?}{=} (1+n)^{s_2} s_3^n \\ (1+n)^{t_2} t_3^n (1+n)^{m\xi} r'^{n\xi} &\stackrel{?}{=} (1+n)^{t_2+m\xi} {t_3}^n r'^{n\xi} \\ E(t_2; t_3) E(m\xi; r'^{\xi}) &\stackrel{?}{=} E(t2 + m\xi; 1) E(0, t_3 r'^{\xi}) \\ E(t_2 + m\xi; t_3 r'^{\xi}) &= E(t2 + m\xi; t_3 r'^{\xi}) \end{aligned}$$

$$\begin{aligned} \gamma c t_2^{\xi} &\stackrel{?}{=} (1+n)^{s_1} s_4^n \\ (1+n)^{t_1} t_4^n (1+n)^{r\xi} r''^{n\xi} &\stackrel{?}{=} (1+n)^{t_1+r\xi} t_4 r''^{n\xi} \\ E(t_1; t_4) E(r\xi; r''^{\xi}) &\stackrel{?}{=} E(t_1 + r\xi; 1) E(0, t_4 r''^{\xi}) \\ E(t_1 + r\xi; t_4 r''^{\xi}) &= E(t_1 + r\xi; t_4 r''^{\xi}) \end{aligned}$$

We detail the extractor and simulator.

Special Soundness Given two accepting transcripts $(\alpha, \beta, \gamma, \xi, s_1, s_2, s_3, s_4)$ and $(\alpha, \beta, \gamma, \xi', s_1', s_2', s_3', s_4')$, we show that $r = \frac{s_1 - s_1'}{\xi - \xi'}, m = \frac{s_2 - s_2'}{\xi - \xi'}, r' = (s_3/s_3')^{\frac{1}{\xi - \xi'}}$, $r'' = (s_4/s_4')^{\frac{1}{\xi - \xi'}}$ must be valid given that two transcripts accept. The difference $\xi - \xi'$ has no inverse with negligible probability.

$$\begin{aligned} c &= \frac{c^e \alpha}{c^{e'} \alpha}^{\frac{1}{e-e'}} \\ c &= \frac{g^{s_1} h^{s_2}}{g^{s_1'} h^{s_2'}}^{\frac{1}{e-e'}} \\ c &= g^{\frac{s_1 - s_1'}{e-e'}} h^{\frac{s_2 - s_2'}{e-e'}} \end{aligned}$$

$$ct_1 = \frac{ct_1^e \beta^{\frac{1}{e-e'}}}{ct_1^{e'} \beta}$$
$$ct_1 = \frac{(1+n)^{s_2} s_3^n}{(1+n)^{s_2'} s_3'^n}^{\frac{1}{e-e'}}$$
$$ct_1 = (1+n)^{\frac{s_2-s_2'}{e-e'}} \left(\frac{s_3}{s_3'}^{\frac{1}{e-e'}}\right)^n$$

$$ct_2 = \frac{ct_2^e \gamma^{\frac{1}{e-e'}}}{ct_2^{e'} \gamma}$$
$$ct_2 = \frac{(1+n)^{s_1} s_4^n}{(1+n)^{s_1'} s_4'^n}^{\frac{1}{e-e'}}$$
$$ct_2 = (1+n)^{\frac{s_1-s_1'}{e-e'}} \left(\frac{s_4}{s_4'}^{\frac{1}{e-e'}}\right)^n$$

To calculate $r' = (s_3/s_3')^{\frac{1}{\xi-\xi'}}, r'' = (s_4/s_4')^{\frac{1}{\xi-\xi'}}$, we use our knowledge of the message in ct_1 and ct_2, extracted from s_1 and s_2, and the homomorphic property of Paillier encryption to create $ct_1' = r'^n$ and $ct_2' = r''^n$. We can directly apply the technique from Damgård *et al.* [15] to extract r' and r'' from the elements s_3, s_3', s_4, s_4'.

Honest Verifier Zero Knowledge Consider a transcript $(\alpha, \beta, \gamma, \xi, s_1, s_2, s_3, s_4)$. In the honest run, t_1, t_2 are random elements in $\mathbb{Z}_n$, t_3, t_4 in $\mathbb{Z}_n^*$ and ξ in $\mathbb{Z}_n$. To simulate, choose s_1, s_2 from $\mathbb{Z}_n$, s_3, s_4 from $\mathbb{Z}_n^*$ and ξ at random from $\mathbb{Z}_n$. Set $\alpha = c^{-\xi} g^{s_1} h^{s_2}$, $\beta = ct_1^{-\xi}(1+n)^{s_2} s_3^n$, $\gamma = ct_2^{-\xi}(1+n)^{s_1} s_4^n$, that is a perfect simulation. Moreover, the elements β, γ are uniformly random in the honest run, and the tuple $(\alpha, s_1, s_2, s_3, s_4)$ is uniquely determined by (ξ, β, γ). In the simulation, the elements s_1, s_2, s_3, s_4 are chosen uniformly at random and consequently β, γ are uniformly at random for fixed elements ξ, c, ct_1, ct_2. □

4.2 Sigma Protocol for Correct Re-Encryption

We introduce the following Sigma Protocol for correct re-encryption. It is used to prove that given a pair of ciphertexts, the second is a re-encryption of the first.

Given $CT = (c, ct_1, ct_2), CT' = (c' = c * g^{r_0} \bmod kn+1, ct_1' = ct_1 * r_1^n \bmod n^2, ct_2' = ct_2 * (1+n)^{r_0} r_2^n \bmod n^2)$, we show that we know $(r_0 \in \mathbb{Z}_n, r_1 \in \mathbb{Z}_n^*, r_2 \in \mathbb{Z}_n^*)$:

1) Let t_1 be a random element in $\mathbb{Z}_n$ and t_2, t_3 be random elements in $\mathbb{Z}_n^*$. The prover computes $\alpha = g^{t_1} \bmod kn+1, \beta = t_2^n \bmod n^2, \gamma = (1+n)^{t_1} t_3^n \bmod n^2$ and sends them to the verifier.

2) The verifier sends a challenge ξ chosen at random in $\mathbb{Z}_n$.

3) The prover computes $s_1 = t_1 + \xi r_0 \bmod n, s_2 = t_2 * r_1^{\xi} \bmod n, s_3 = t_3 * r_2^{\xi} \bmod n$, and sends these to the verifier.

4) The verifier accepts if $\alpha(c'/c)^{\xi} = g^{s_1}, \beta(ct_1'/ct_1)^{\xi} = s_2^n, \gamma(ct_2'/ct_2)^{\xi} = (1+n)^{s_1}s_3^n$.
The transcript (with the elements exchanged between the prover and the verifier) is $(\alpha \in \mathbb{G}_n, \beta, \gamma \in \mathbb{Z}_{n^2}^*, \xi, s_1 \in \mathbb{Z}_n, s_2, s_3 \in \mathbb{Z}_n^*)$.

Security Proof

Proposition 3. *The above protocol has perfect completeness, special soundness, and honest verifier zero knowledge and is hence a Sigma Protocol for correct re-encryption.*

Proof. Completeness follows trivially and is omitted.

Special Soundness Given two accepting transcripts $(\alpha, \beta, \gamma, \xi, s_1, s_2, s_3)$ and $(\alpha, \beta, \gamma, \xi', s_1', s_2', s_3')$, we show that $r_0 = \frac{s_1 - s_1'}{\xi - \xi'}, r_1 = (s_2/s_2')^{\frac{1}{\xi - \xi'}}, r_2 = (s_3/s_3')^{\frac{1}{\xi - \xi'}}$ must be valid given that two transcripts accept. The difference $\xi - \xi'$ has no inverse with negligible probability.

$$(c'/c) = \frac{(c'/c)^{\xi}\alpha}{(c'/c)^{\xi'}\alpha}^{\frac{1}{\xi - \xi'}}$$
$$(c'/c) = \frac{g^{s_1}}{g^{s_1'}}^{\frac{1}{\xi - \xi'}}$$
$$(c'/c) = g^{\frac{s_1 - s_1'}{\xi - \xi'}}$$

$$(ct_1'/ct_1) = \frac{(ct_1'/ct_1)^{\xi}\beta}{(ct_1'/ct_1)^{\xi'}\beta}^{\frac{1}{\xi - \xi'}}$$
$$(ct_1'/ct_1) = \frac{s_2^n}{s_2'^n}^{\frac{1}{\xi - \xi'}}$$
$$(ct_1'/ct_1) = \frac{s_2}{s_2'}^{\frac{1}{\xi - \xi'}^n}$$

$$(ct_2'/ct_2) = \frac{(ct_2'/ct_2)^{\xi}\gamma}{(ct_2'/ct_2)^{\xi'}\gamma}^{\frac{1}{\xi - \xi'}}$$
$$(ct_2'/ct_2) = \frac{(1+n)^{s_1}s_3^n}{(1+n)^{s_1'}s_3'^n}^{\frac{1}{\xi - \xi'}}$$
$$(ct_2'/ct_2) = (1+n)^{\frac{s_1 - s_1'}{\xi - \xi'}}\frac{s_3}{s_3'}^{\frac{1}{\xi - \xi'}^n}$$

Honest Verifier Zero Knowledge In the honest run, t_1 is chosen at random from $\mathbb{Z}_n$, t_2, t_3 from $\mathbb{Z}_n^*$ and ξ from $\mathbb{Z}_n$. To simulate, we instead choose s_1, s_2, s_3, ξ at random and set $\alpha = g^{s_1}(c_1'/c_1)^{-\xi}, \beta = s_2^n(c_2'/c_2)^{-\xi}, \gamma = (1+n)^{s_1}s_3^n(c_3'/c_3)^{-\xi}$. We get the same distribution in both cases.

□

5 A New Efficient Verifiable Mix-Net

Verifiable mixing is an important building block for almost all verifiable voting systems. Given a vector of ciphertexts with known relationships to the voters, mixing allows this link to be broken without allowing ballot modification or substitution. Let $L_0 = (c_{0,1}, ..., c_{0,N})$ be the input ciphertexts, the jth mixer chooses a random permutation π and $r_{j,i} \in \mathcal{R}_{PK}$ and sets $c_{j,i} = c_{j-1,\pi(i)} E_{PK}(1, r_{j,\pi(i)})$ and publishes $L_j = (c_{j,1}, ..., c_{j,N})$. The output of the final mix L_k can then be decrypted and, provided the encryption system is secure, the relationship between L_0 and the plaintexts can not be feasibly derived without the assistance of all mixers.

Wikström's general result [33] shows that verifiable mixing is possible for all cryptosystems on which a homomorphic map exists and an overwhelmingly complete Sigma Protocol is known for re-encryption. However, this generic construction gives an 8-round proof, while a more optimised instance is desirable for practicality. We can take advantage of special properties from our solution and derive a secure 4-round proof. We emphasise that the existence of a suitable, and somewhat practical, mix-net for the described construction is implied by known results as a result of the re-encryption Sigma Protocol we just presented. Nevertheless, to further increase practicality and scalability we present a more efficient version. We illustrate a verifiable ballot mixing process in Fig. 5 with three mixers.

Formally we operate two mixes, one on the public board and on the secret (private) board. At each step the authorities check that the two versions of the Pedersen commitments match. See figure 5 for an example with three mixers. Our suggestion is similar to Demirel *et al.* [17], however– because of our earlier contributions–our system is actually shown to be secure and far more computationally efficient.

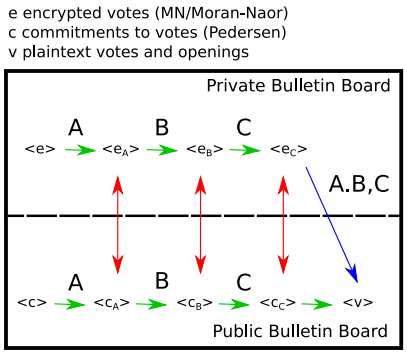

Fig. 5. Mixing with three authorities

We now present our more efficient mixers. While there are crucial differences (for instance, the composite group order), our optimisations and accompanying proofs are similar to those for the optimised ElGamal version which is presented and proven by Terelius *et al.* [30]. We first detail the mix-net for the public board, see Algorithm 1, and then the mix-net for the private board, see Algorithm 2. We recall that π is permutation function induced by the permutation matrix M and ϕ is the re-encryption map defined in Subsection 2.2. We use $\bar{1}$ to denote the all one vector.

We technically need a generalised Pedersen commitment for the mix. Such a generalised commitment takes a vector of messages and commitments to the entire vector in one commitment. We refer the reader to Wikström's treatment

in [33]. That the security of the generalised commitment also reduces to the discrete log problem is well known, and we omit the details. We define $\mathcal{R}_{com}$ to be the relation consisting of pairs of tuples of the form commitment key CK, commitment $\mathbf{c}$, two distinct messages M, M' and two associated randomness vectors $\mathbf{r}$ and $\mathbf{r}'$ s.t. $\mathbf{c} = \mathit{\Pi}.\mathsf{Com}_{CK}(M, \mathbf{r}) = \mathit{\Pi}.\mathsf{Com}_{CK}(M', \mathbf{r}')$. We also define $\mathcal{R}_{\pi}$ to be the relation consisting of pairs of tuples of the form commitment key CK, commitment $\mathbf{c}$, message M and associated randomness vector $\mathbf{r}$ s.t. M is a permutation matrix and $\mathbf{r} = \mathit{\Pi}.\mathsf{Com}_{CK}(M, \mathbf{r})$. Let $\mathcal{R}^{shuf}_{\phi_{PK}}$ be the relation consisting of pairs of tuples of the form public key PK, two vectors of ciphertexts $\mathbf{CT} = (ct_1, \cdots, ct_n)$ and $\mathbf{CT}' = (ct'_1, \cdots, ct'_n)$ and a permutation π and randomness vector $\mathbf{r} = (r_1, \cdots, r_n)$ such that $ct'_i = \phi_{PK}(ct_{\pi(i)}, r_{\pi(i)})$ for all $i \in [1, N]$. Let $\mathcal{R}^{shuf}_{rerand_{CK}}$ to be the relation consisting of pairs of tuples of the form commit key CK, two commitment vectors $\mathbf{c} = (c_1, \cdots, c_n)$ and $\mathbf{c}' = (c'_1, \cdots, c'_n)$, a permutation π and randomness vector $\mathbf{r} = (r_1, \cdots, r_n)$ such that $c'_i = \mathit{\Pi}.\mathsf{ReRand}_{CK}(c_{\pi(i)}, r_{\pi(i)})$.

Algorithm 1: Proof of Shuffle on Public Board

Common Input: Commitment parameters $g, h, h_1, ..., h_N \in \mathbb{G}_n$, two Pedersen commitments $\mathbf{e} = (e_1, ..., e_N) \in \mathbb{G}_n^N$ and $\mathbf{e}' = (e'_1, ..., e'_N) \in \mathbb{G}_n^N$, and a permutation matrix commitment $\mathbf{c} = (c_1, ..., c_N)$.

Private Input : Permutation matrix $M = (m_{i,j}) \in \mathbb{Z}_n^{N \times N}$, randomness $\mathbf{r} = (r_1, ..., r_N) \in \mathbb{Z}_n^N$ s.t. $c_j = g^{r_j} \prod_{i=1}^N h_i^{m_{j,i}}$, and randomness $\mathbf{r}' = (r'_1, ..., r'_N) \in \mathbb{Z}_n^N$ s.t. $e'_i = e_{\pi(i)} g^{r'_{\pi(i)}}$ for $i, j \in [1, N]$.

1 $\mathcal{V}$ chooses $\mathbf{u} = (u_1, ..., u_N) \in \mathbb{Z}_n^N$ randomly and hands $\mathbf{u}$ to $\mathcal{P}$.

2 $\mathcal{P}$ defines $\mathbf{u}' = (u'_1, ..., u'_N) = M\mathbf{u}$ and then chooses $\hat{\mathbf{r}} = (\hat{r}_1, ..., \hat{r}_N)$,$\hat{\mathbf{w}} = (\hat{w}_1, ..., \hat{w}_N)$,$\mathbf{w}' = (w'_1, ..., w'_N) \in \mathbb{Z}_n^N$, and $w_1, w_2, w_3, w_4 \in \mathbb{Z}_n$. $\mathcal{P}$ then defines $\bar{r} = \langle \bar{1}, \mathbf{r} \rangle$, $\tilde{r} = \langle \mathbf{r}, \mathbf{u} \rangle$, $\hat{r} = \sum_{i=1}^N \hat{r}_i \prod_{j=i+1}^N u'_j$ and $r' = \langle \mathbf{r}', \mathbf{u} \rangle$. $\mathcal{P}$ hands to $\mathcal{V}$, where we set $\hat{c}_0 = h$ and $i \in [1, N]$,

$$\hat{c}_i = g^{\hat{r}_i} \hat{c}_{i-1}^{u'_i} \quad t_1 = g^{w_1} \quad t_2 = g^{w_2} \quad t_3 = g^{w_3} \prod_{i=1}^N h_i^{w'_i} \quad t_4 = g^{-w_4} \prod_{i=1}^N (e'_i)^{w'_i} \quad \hat{t}_i = g^{\hat{w}_i} \hat{c}_{i-1}^{w'_i}$$

3 $\mathcal{V}$ chooses a challenge $\xi \in \mathbb{Z}_n$ at random and sends it to $\mathcal{P}$.

4 $\mathcal{P}$ then responds with:

$$s_1 = w_1 + \xi \cdot \bar{r} \qquad s_2 = w_2 + \xi \cdot \hat{r} \qquad s_3 = w_3 + \xi \cdot \tilde{r} \qquad s_4 = w_4 + \xi \cdot r'$$
$$\hat{s}_i = \hat{w}_i + \xi \cdot \hat{r}_i \qquad s'_i = w'_i + \xi \cdot u'_i$$

5 $\mathcal{V}$ accepts if and only if, for $i \in [1, N]$,

$$t_1 = (\prod_{i=1}^N c_i / \prod_{i=1}^N h_i)^{-\xi} g^{s_1} \quad t_2 = (\hat{c}_N / h^{\prod_{i=1}^N u_i})^{-\xi} g^{s_2} \quad t_3 = (\prod_{i=1}^N c_i^{u_i})^{-\xi} g^{s_3} \prod_{i=1}^N h_i^{s'_i}$$
$$t_4 = (\prod_{i=1}^N (e_i)^{u_i})^{-\xi} g^{s_4} \prod_{i=1}^N (e'_i)^{s'_i} \qquad \hat{t}_i = \hat{c}_i^{-\xi} g^{\hat{s}_i} \hat{c}_{i-1}^{s'_i}$$

Algorithm 2: Proof of Shuffle on Private Board

Common Input: Commitment parameters $g, h, h_1, ..., h_N \in \mathbb{G}_n$, two ciphertexts $\mathbf{e} = (e_1, ..., e_N) \in \mathcal{C}_{PK}$ and $\mathbf{e}' = (e'_1, ..., e'_n) \in \mathcal{C}_{PK}$, and a permutation matrix commitment $\mathbf{c} = (c_1, ..., c_N)$.

Private Input : Permutation matrix $M = (m_{i,j}) \in \mathbb{Z}_n^{N \times N}$, randomness $\mathbf{r} = (r_1, ..., r_N) \in \mathbb{Z}_n^N$ s.t. $c_j = g^{r_j} \prod_{i=1}^N h_i^{m_{j,i}}$, and randomness $\mathbf{r}' = (r'_1, ..., r'_N) \in \mathcal{R}_{pk}$ s.t. $e'_i = \phi_{PK}(e_{\pi(i)}, r'_{\pi(i)})$, for $i, j \in [1, N]$.

1 $\mathcal{V}$ chooses $\mathbf{u} = (u_1, ..., u_N) \in \mathbb{Z}_n^N$ randomly and hands $\mathbf{u}$ to $\mathcal{P}$.

2 $\mathcal{P}$ defines $\mathbf{u}' = (u'_1, ..., u'_N) = M\mathbf{u}$ and then chooses $\hat{\mathbf{r}} = (\hat{r}_1, ..., \hat{r}_N), \hat{\mathbf{w}} = (\hat{w}_1, ..., \hat{w}_N), \mathbf{w}' = (w'_1, ..., w'_N) \in \mathbb{Z}_n^N$, and $w_1, w_2, w_3, \in \mathbb{Z}_n$ and $w_4 \in \mathcal{R}_{PK}$. $\mathcal{P}$ defines $\bar{r} = \langle \bar{1}, \mathbf{r} \rangle$, $\tilde{r} = \langle \mathbf{r}, \mathbf{u} \rangle$, $\hat{r} = \sum_{i=1}^N \hat{r}_i \prod_{j=i+1}^N u'_j$ and $r' = (\sum_{i=1}^N r'_{i,0} u_i, \prod_{i=1}^N r'^{u_i}_{i,1}, \prod_{i=1}^N r'^{u_i}_{i,2})$. $\mathcal{P}$ hands to $\mathcal{V}$, where we set $\hat{c}_0 = h$ and $i \in [1, N]$,

$$\hat{c}_i = g^{\hat{r}_i} \hat{c}_{i-1}^{u'_i} \qquad t_1 = g^{w_1} \qquad t_2 = g^{w_2} \qquad t_3 = g^{w_3} \prod_{i=1}^N h_i^{w'_i}$$
$$t_4 = \Sigma.\mathsf{Enc}_{PK}(0, w_4) \prod_{i=1}^N e_i'^{w'_i} \qquad \hat{t}_i = g^{\hat{w}_i} \hat{c}_{i-1}^{w'_i}$$

3 $\mathcal{V}$ chooses a challenge $\xi \in \mathbb{Z}_n$ at random and sends it to $\mathcal{P}$.

4 $\mathcal{P}$ then responds with:

$$s_1 = w_1 + \xi \cdot \bar{r} \qquad s_2 = w_2 + \xi \cdot \hat{r} \qquad s_3 = w_3 + \xi \cdot \tilde{r} \qquad s_4 = w_4 - \xi \cdot r'$$
$$\hat{s}_i = \hat{w}_i + \xi \cdot \hat{r}_i \qquad s'_i = w'_i + \xi \cdot u'_i$$

5 $\mathcal{V}$ accepts if and only if, for $i \in [1, N]$,

$$t_1 = (\prod_{i=1}^N c_i / \prod_{i=1}^N h_i)^{-\xi} g^{s_1} \qquad t_2 = (\hat{c}_N / h^{\prod_{i=1}^N u_i})^{-\xi} g^{s_2} \qquad t_3 = (\prod_{i=1}^N c_i^{u_i})^{-\xi} g^{s_3} \prod_{i=1}^N h_i^{s'_i}$$
$$t_4 = (\prod_{i=1}^N (e_i)^{u_i})^{-\xi} \Sigma.\mathsf{Enc}_{PK}(0, s_4) \prod_{i=1}^N (e'_i)^{s'_i} \qquad \hat{t}_i = \hat{c}_i^{-\xi} g^{\hat{s}_i} \hat{c}_{i-1}^{s'_i}$$

Proposition 4. *Algorithm 1 is a perfectly complete, 4-round special soundness, and honest verifier zero knowledge of the relationship* $\mathcal{R}_{com} \vee (\mathcal{R}_{\pi} \wedge \mathcal{R}^{shuf}_{rerand_{CK}})$.

Proof. Proposition 4 (Algorithm 1) is closely related to many derivations of Wikström's mix-net, and also to Proposition 5 (Algorithm 2) and for this reason we omit the proof. □

Proposition 5. *Algorithm 2 is a perfectly complete, 4-round special soundness, and honest verifier zero knowledge of the relationship* $\mathcal{R}_{com} \vee (\mathcal{R}_{\pi} \wedge \mathcal{R}^{shuf}_{\phi_{PK}})$.

To prove that Algorithm 2 is a perfectly complete, 4-message special soundness, and statistical honest verifier zero-knowledge of the relationship $\mathcal{R}_{com} \vee (\mathcal{R}_{\pi} \wedge \mathcal{R}^{shuf}_{\phi_{pk}})$, one must demonstrate its completeness, special soundness extractor, and statistical honest verifier knowledge simulator. Many parts of the proof are identical to the ElGamal variant; we include verbatim those parts from the prior work of Haines [21].

Correctness

Proof. The correctness of the scheme follows from substituting the variables being verified with their definitions in the honest protocol. We provide the details for ease of reading.

Start with the verification equations:

$$t_1 = (\textstyle\prod_{i=1}^{N} c_i / \prod_{i=1}^{N} h_i)^{-\xi} g^{s_1} \qquad t_2 = (\hat{c}_N / h^{\prod_{i=1}^{N} u_i})^{-\xi} g^{s_2}$$

$$t_3 = (\textstyle\prod_{i=1}^{N} c_i^{u_i})^{-\xi} g^{s_3} \prod_{i=1}^{N} h_i^{s'_i}$$

$$t_4 = (\textstyle\prod_{i=1}^{N} (e_i)^{u_i})^{-\xi} \Sigma.\mathsf{Enc}_{PK}(0, s_4) \prod_{i=1}^{N} (e'_i)^{s'_i} \qquad \hat{t}_i = \hat{c}_i^{-\xi} g^{\hat{s}_i} \hat{c}_{i-1}^{s'_i}$$

Substitute the variables for their definitions (with some cancellation):

$$g^{w_1} = (g^{\bar{r}} \textstyle\prod_{i=1}^{N} h_i / \prod_{i=1}^{N} h_i)^{-\xi} g^{w_1 + \xi \cdot \bar{r}}$$

$$g^{w_2} = (g^{\hat{r}} h^{\prod_{i=1}^{N} u_i} / h^{\prod_{i=1}^{N} u_i})^{-\xi} g^{w_2 + \xi \cdot \hat{r}}$$

$$g^{w_3} \textstyle\prod_{i=1}^{N} h_i^{w'_i} = (g^{\tilde{r}} \prod_{i=1}^{N} h_i^{u'_i})^{-\xi} g^{w_3 + \xi \cdot \tilde{r}} \prod_{i=1}^{N} h_i^{w'_i + \xi \cdot u'_i}$$

$$E_{pk}(1, w_4) \textstyle\prod_{i=1}^{N} (e'_i)^{w'_i} = (\prod_{i=1}^{N} (e_i)^{u_i})^{-\xi} \Sigma.\mathsf{Enc}_{PK}(0, w_4 - \xi \cdot r') \prod_{i=1}^{N} (e'_i)^{w'_i + \xi \cdot u'_i}$$

$$g^{\hat{w}_i} \hat{c}_{i-1}^{w'_i} = g^{-\xi \cdot \hat{r}_i} \hat{c}_{i-1}^{-\xi \cdot u'_i} g^{\hat{w}_i + \xi \cdot \hat{r}_i} \hat{c}_{i-1}^{w'_i + \xi \cdot u'_i}$$

Further cancellation makes the equality obvious:

$$g^{w_1} = g^{w_1}$$

$$g^{w_2} = g^{w_2}$$

$$g^{w_3} \textstyle\prod_{i=1}^{N} h_i^{w'_i} = g^{w_3} \prod_{i=1}^{N} h_i^{w'_i}$$

$$\Sigma.\mathsf{Enc}_{PK}(0, w_4) \textstyle\prod_{i=1}^{N} (e'_i)^{w'_i} = \Sigma.\mathsf{Enc}_{PK}(0, w_4) \prod_{i=1}^{N} (e'_i)^{w'_i}$$

$$g^{\hat{w}_i} \hat{c}_{i-1}^{w'_i} = g^{\hat{w}_i} \hat{c}_{i-1}^{w'_i}$$

□

Zero-Knowledge

Proof. The special zero-knowledge simulator chooses $\hat{c}_1, ..., \hat{c}_N, \in \mathbb{G}_n$, $\mathbf{c} \in \mathbb{G}_n^N$ $\hat{s}, s', \mathbf{u} \in \mathbb{Z}_n^N$, and $s_1, s_2, s_3, c \in \mathbb{Z}_n$ and $s_4 = \mathcal{R}_{pk}$ randomly and defines $t_1, t_2, t_3, t_4, \hat{t}_i$ by the equations in step five. This is a perfect simulation.

To show that the simulated and real transcripts have the same statistical distribution we compare their terms as follows:

- $\mathbf{u} \in_R \mathbb{Z}_q^N$ in both.
- $\hat{c}_1, ..., \hat{c}_N \in_R \mathbb{G}_q$ in simulated and as $\hat{c}_i = g^{\hat{r}_i} \hat{c}_{i-1}^{u_i'}$ in the real transcript where $\hat{r}_i \in_R \mathbb{Z}_q$ which randomly distributes them in $\mathbb{G}_q$.
- $t_1, t_2, t_3, t_4, \hat{\mathbf{t}}$ and the corresponding $s_1, s_2, s_3, s_4, \hat{\mathbf{s}}, \mathbf{s}$ have a defined relation which depends on secrets and ws. Since the ws are randomly defined in an honest run and the $s_1, s_2, s_3, s_4, \hat{\mathbf{s}}, \mathbf{s}$ in the simulated, the elements are uniformly distributed in both, up to the defined relationship.
- The challenge c is uniformly distributed in both.

□

5.1 Soundeness

Proof. The extractor from two accepting transcripts

$$(\mathbf{c}, \mathbf{u}, \hat{\mathbf{c}}, t_1, t_2, t_3, t_4, \hat{\mathbf{t}}, \xi, s_1, s_2, s_3, s_4, \hat{\mathbf{s}}, \mathbf{s}')$$

$$(\mathbf{c}, \mathbf{u}, \hat{\mathbf{c}}, t_1, t_2, t_3, t_4, \hat{\mathbf{t}}, \xi^*, s_1^*, s_2^*, s_3^*, s_4^*, \hat{\mathbf{s}}^*, \mathbf{s}'^*)$$

with $\xi \neq \xi^*$ we define $\bar{r} = (s_1 - s_1^*)/(\xi - \xi^*), \hat{r} = (s_2 - s_2^*)/(\xi - \xi^*), \tilde{r} = (s_3 - s_3^*)/(\xi - \xi^*), r' = ((s_{4,1} - s_{4,1}^*)/(\xi - \xi^*), \frac{s_{4,2}}{s'_{4,2}} \frac{1}{\xi - \xi'}, \frac{s_{4,3}}{s'_{4,3}} \frac{1}{\xi - \xi'}), \hat{\mathbf{r}}' = ((\hat{\mathbf{s}}' - \hat{\mathbf{s}}'^*)/(\xi - \xi^*), \mathbf{u}' = (\mathbf{s}' - \mathbf{s}'^*)/(\xi - \xi^*)$, and show that

$$\prod_{j=1}^{N} c_j = C(\mathbf{1}, \bar{r}) \qquad \prod_{j=1}^{N} c_j^{u_j} = C(\boldsymbol{u}', \tilde{r}) \qquad \prod_{i=1}^{N} (e_i')^{u_i'} = \Sigma.\mathsf{Enc}_{PK}(0, r') \cdot \prod_{j=1}^{N} e_j^{u_j}$$

$$\hat{c}_i = g^{r_i} \hat{c}_{i-1}^{u_i'} \qquad \hat{c}_N = C(u, \hat{r})$$

The proof consists of simple algebraic transformations:

$$\left(\frac{(\prod_{j=1}^{N}\mathbf{c}_j)^{\xi}t_1}{(\prod_{j=1}^{N}\mathbf{c}_j)^{\xi^*}t_1}\right)^{\frac{1}{\xi-\xi^*}} = \prod_{j=1}^{N}\mathbf{c}_j \qquad \text{Tautology}$$

$$\left(\frac{g^{s_1}/(\prod_{i=1}^{N}h_i)^{-\xi}}{g^{s_1^*}/(\prod_{i=1}^{N}h_i)^{-\xi^*}}\right)^{\frac{1}{\xi-\xi^*}} = \prod_{j=1}^{N}\mathbf{c}_j \qquad \text{By the verification definition}$$

$$h^{\frac{s_1-s_1^*}{\xi-\xi^*}}\prod_{i=1}^{N}h_i = \prod_{j=1}^{N}\mathbf{c}_j \qquad \text{By algebraic manipulation}$$

$$EPC(\mathbf{1}, \frac{s_1-s_1^*}{\xi-\xi^*}) = \prod_{j=1}^{N}\mathbf{c}_j \qquad \text{By definition of EPC}$$

$$EPC(\mathbf{1}, \bar{r}) = \prod_{j=1}^{N}\mathbf{c}_j \qquad \text{By definition of } \bar{r}$$

$$\left(\frac{(\prod_{j=1}^{N}\mathbf{c}_j^{\mathbf{u}_j})^{\xi}t_3}{(\prod_{j=1}^{N}\mathbf{c}_j^{\mathbf{u}_j})^{\xi^*}t_3}\right)^{\frac{1}{\xi-\xi^*}} = \prod_{j=1}^{N}\mathbf{c}_j^{\mathbf{u}_j} \qquad \text{Tautology}$$

$$\left(\frac{g^{s_3}\prod_{i=1}^{N}h_i^{\mathbf{s}_i'}}{g^{s_3^*}\prod_{i=1}^{N}h_i^{\mathbf{s}_i'^*}}\right)^{\frac{1}{\xi-\xi^*}} = \prod_{j=1}^{N}\mathbf{c}_j^{\mathbf{u}_j} \qquad \text{By verification definition}$$

$$h^{\frac{s_3-s_3^*}{\xi-\xi^*}}\prod_{i=1}^{N}h_i^{\frac{\mathbf{s}_i'-\mathbf{s}_i'^*}{\xi-\xi^*}} = \prod_{j=1}^{N}\mathbf{c}_j^{\mathbf{u}_j} \qquad \text{By algebraic manipulation}$$

$$EPC(\frac{\mathbf{s}'-\mathbf{s}'^*}{\xi-\xi^*}, \frac{s_3-s_3^*}{\xi-\xi^*}) = \prod_{j=1}^{N}\mathbf{c}_j^{\mathbf{u}_j} \qquad \text{By definition of EPC}$$

$$EPC(\boldsymbol{u}', \tilde{r}) = \prod_{j=1}^{N}\mathbf{c}_j^{\mathbf{u}_j} \qquad \text{By definition of } \boldsymbol{u}' \text{ and } \tilde{r}$$

$$\left(\frac{(\prod_{i=1}^{N}(\mathbf{e}_i)^{\mathbf{u}_i})^{\xi}\mathbf{t}_4}{(\prod_{i=1}^{N}(\mathbf{e}_i)^{\mathbf{u}_i})^{\xi^*}\mathbf{t}_4}\right)^{\frac{1}{\xi-\xi^*}} = \prod_{i=1}^{N}\mathbf{e}_i{}^{\mathbf{u}_i} \qquad \text{Tautology}$$

$$\left(\frac{\prod_{i=1}^{N}(\mathbf{e}'_i)^{\mathbf{s}'_i}\mathbf{Enc}(0,\mathbf{s}_4)}{\prod_{i=1}^{N}(\mathbf{e}'_i)^{\mathbf{s}'^*_i}\mathbf{Enc}(0,\mathbf{s}^*_4)}\right)^{\frac{1}{\xi-\xi^*}} = \prod_{i=1}^{N}\mathbf{e}_i{}^{\mathbf{u}_i} \qquad \text{By verification definition}$$

$$\prod_{i=1}^{N}\mathbf{e}'_i{}^{\frac{\mathbf{s}'_i-\mathbf{s}'^*_i}{\xi-\xi^*}}\mathbf{Enc}(0,\frac{\mathbf{s}^*_4-\mathbf{s}_4}{\xi-\xi^*}) = \prod_{i=1}^{N}\mathbf{e}_i{}^{\mathbf{u}_i} \qquad \text{By algebraic manipulation}$$

$$\prod_{i=1}^{N}\mathbf{e}'_i{}^{\frac{\mathbf{s}'_i-\mathbf{s}'^*_i}{\xi-\xi^*}} = \mathbf{Enc}(0,\frac{\mathbf{s}_4-\mathbf{s}^*_4}{\xi-\xi^*})\prod_{i=1}^{N}\mathbf{e}_i{}^{\mathbf{u}_i} \qquad \text{By algebraic manipulation}$$

$$\prod_{i=1}^{N}\mathbf{e}'_i{}^{\mathbf{u}'_i} = \mathbf{Enc}_{pk}(0,\mathbf{r}^\star)\prod_{i=1}^{N}\mathbf{e}_i{}^{\mathbf{u}_i} \qquad \text{By definition of } r'_j \text{ and } \mathbf{u}'_i$$

Now, for each $i \in \{1,\ldots,N\}$

$$\left(\frac{\hat{c}_i^{c}\hat{\mathbf{t}}_i}{\hat{c}_i^{c^*}\hat{\mathbf{t}}_i}\right)^{\frac{1}{\xi-\xi^*}} = \hat{c}_i \qquad \text{Tautology}$$

$$\left(\frac{h^{\hat{\mathbf{s}}_i}\hat{c}_{i-1}^{\mathbf{s}'_i}}{h^{\hat{\mathbf{s}}^*_i}\hat{c}_{i-1}^{\mathbf{s}'^*_i}}\right)^{\frac{1}{\xi-\xi^*}} = \hat{c}_i \qquad \text{By verification definition}$$

$$h^{\frac{\hat{\mathbf{s}}_i-\hat{\mathbf{s}}^*_i}{\xi-\xi^*}}\hat{c}_{i-1}^{\frac{\mathbf{s}'_i-\mathbf{s}'^*_i}{\xi-\xi^*}} = \hat{c}_i \qquad \text{By algebraic manipulations}$$

$$PC_{h,\hat{c}_{i-1}}(\frac{\mathbf{s}'_i-\mathbf{s}'^*_i}{\xi-\xi^*},\frac{\hat{\mathbf{s}}_i-\hat{\mathbf{s}}^*_i}{\xi-\xi^*}) = \hat{c}_i \qquad \text{By algebraic manipulations}$$

$$PC_{h,\hat{c}_{i-1}}(\mathbf{u}'_i,\hat{\mathbf{r}}_i) = \hat{c}_i \qquad \text{By definition of } \mathbf{u}'_i \text{ and } \hat{\mathbf{r}}_i$$

$$\left(\frac{\hat{c}_N^{\xi}t_2}{\hat{c}_N^{\xi^*}t_2}\right)^{\frac{1}{\xi-\xi^*}} = \hat{c}_N \qquad \text{Tautology}$$

$$\left(\frac{(h_1^{\prod_{i=1}^{N}\mathbf{u}_i})^{\xi}g^{s_2}}{(h_1^{\prod_{i=1}^{N}\mathbf{u}_i})^{\xi^*}g^{s_2^*}}^{\frac{1}{\xi-\xi^*}}\right) = \hat{c}_N \qquad \text{By verification definition}$$

$$g^{\frac{s_2-s_2^*}{\xi-\xi^*}}h_1^{\prod_{i=1}^{N}\mathbf{u}_i} = \hat{c}_N \qquad \text{By algebraic manipulation}$$

$$PC(\prod_{i=1}^{N}\mathbf{u}_i,\frac{s_2-s_2^*}{\xi-\xi^*}) = \hat{c}_N \qquad \text{By algebraic manipulation}$$

$$PC_{g,h_1}(\prod_{i=1}^{N}\mathbf{u}_i,r^\diamond) = \hat{c}_N \qquad \text{By definition of } r^\diamond$$

Extended Extractor We now sketch the extended extractor which, for a given statement (see the common input in Algorithm 2), for n different witnesses extracted by the basic extractor, produces the witnesses to the main statement. Let the collective output of the basic extractors be denoted as $\bar{\mathbf{r}}, \mathbf{r}^{\diamond}, \tilde{\mathbf{r}}, \mathbf{r}^{\star} \in \mathbb{Z}_q^n$, and $\hat{R}, U' \in \mathbb{Z}_q^{N \times N}$ extracted from the primary challenges $U \in \mathbb{Z}_q^{N \times N}$. We denote by U_i the ith column of U which is the challenge vector from the ith run of the basic extractor, and by $U_{j,i}$ the j element of the challenge vector from the ith run of the basic extractor.

First note with overwhelming probability the set of U_is is linearly independent, concretely the probability is bounded by $\frac{q-2}{q}$. From linear independence, it follows that their exists $A \in \mathbb{Z}_q^{N \times N}$ such that UA_l is the lth standard unit vector in $\mathbb{Z}_q$ which we will denote by $\mathbb{I}_l$. A is the inverse of U. Clearly,

$$\mathbf{c}_l = \prod_{i=1}^{N} (\mathbf{c}^{UA_l})_i \qquad \text{since } UA_l \text{ is } \mathbb{I}_l \tag{1}$$

$$\mathbf{c}_l = \prod_{i=1}^{N} \mathbf{c}_i^{\sum_{j=1}^{N} U_{i,j} A_{j,l}} \qquad \text{by definition of } UA_l \tag{2}$$

$$\mathbf{c}_l = \prod_{i=1}^{N} \left(\prod_{j=1}^{N} \mathbf{c}^{U_{i,j} A_{j,l}} \right)_i \qquad \text{by algebraic manipulation} \tag{3}$$

$$\mathbf{c}_l = \prod_{j=1}^{N} \left(\left(\prod_{i=1}^{N} \mathbf{c}_i^{U_{i,j}} \right)^{A_{j,l}} \right) \qquad \text{by algebraic manipulation} \tag{4}$$

$$\mathbf{c}_l = \prod_{j=1}^{N} EPC(U'_j, \tilde{\mathbf{r}}_j)^{A_{j,l}} \qquad \text{by some algebraic manipulation and } \prod_{i=1}^{N} \mathbf{c}_i^{U_{i,j}} = EPC(U'_j, \tilde{\mathbf{r}}_j) \tag{5}$$

$$\mathbf{c}_l = \prod_{j=1}^{N} EPC(U'_j A_{j,l}, \tilde{\mathbf{r}}_j A_{j,l}) \qquad \text{by algebraic manipulation} \tag{6}$$

$$\mathbf{c}_l = EPC(\sum_{j=1}^{N} U'_j A_{j,l}, \langle \tilde{\mathbf{r}}, A_l \rangle) \qquad \text{by algebraic manipulation} \tag{7}$$

$$\mathbf{c}_l = EPC(U' A_l, \langle \tilde{\mathbf{r}}, A_l \rangle) \qquad \text{by algebraic manipulation} \tag{8}$$

Therefore, we can open $\mathbf{c}$ to the matrix M, where the lth column of M is $U'A_l$, with randomness $\langle \tilde{\mathbf{r}}, A_l \rangle$. In other words we open $\mathbf{c} = U'A$ using randomness $\tilde{\mathbf{r}}A$.

We expect M to be a permutation matrix, but if it is not, then one can find a witness to $\mathcal{R}_{com}$ (which, as has been mentioned, can only happen with negligible probability, under our security assumptions). We extract in two different ways depending on whether $M\mathbf{1} \neq \mathbf{1}$.

Option one If $M\mathbf{1} \neq \mathbf{1}$, then let $\mathbf{u}'' = M\mathbf{1}$ and note that

$$\mathbf{u}'' \neq \mathbf{1} \text{ and } EPC(\mathbf{1}, \bar{\mathbf{r}}_j) = \prod_{i=1}^{N} \mathbf{c}_i = \prod_{i=1}^{N} \mathbf{c}_i^{\mathbf{1}_i} = EPC(\mathbf{u}'', \tilde{\mathbf{r}}A)$$

in which case we found a witness breaking the commitment scheme.

Option two If $M\mathbf{1} = \mathbf{1}$, then recall Theorem 1 from "Proofs of Restricted Shuffles", which states that M is a permutation matrix if and only if $M\mathbf{1} = \mathbf{1}$ and $\prod_{i=1}^{N} \langle \mathbf{m}_i, \mathbf{x} \rangle - \prod_{i=1}^{N} \mathbf{x}_i = 0$. Since $M\mathbf{1} = \mathbf{1}$ and M is not a permutation matrix, then $\prod_{i=1}^{N} \langle \mathbf{m}_i, \mathbf{x} \rangle - \prod_{i=1}^{N} \mathbf{x}_i \neq 0$. The Schwartz–Zippel says that if you sample, a non-zero polynomial, at a random point the chance that it equals zero is negligible in the order of the underlying field; hence, with overwhelming probability there exists $j \in \{1, \ldots, N\}$ such $\prod_{i=1}^{N} \langle \mathbf{m}_i, U_j \rangle - \prod_{i=1}^{N} U_{i,j} \neq 0$. Since this is true with overwhelming probability, we require it to be true and rewind if this is not the case. (Strictly speaking we should take $N + 1$ extractions from the basic extractor, if we recover a different M we win, if we get the same M then U_{l+1} is actually independent of M and the lemma can be applied.)

Let $\mathbf{u}'' = MU_j$ and note that

$$\mathbf{u}'' \neq U'_j \qquad \text{Which must be true since } \prod_{i=1}^{N} U'_{i,j} = \prod_{i=1}^{N} U_{i,j} \neq \prod_{i=1}^{N} \mathbf{u}''_i$$

$\prod_{i=1}^{N} U'_j = \prod_{i=1}^{N} U_j$ follows from the base statements and $\prod_{i=1}^{N} U_j \neq \prod_{i=1}^{N} \mathbf{u}''$ by definition of $\mathbf{u}''$ and $\prod_{i=1}^{N} \langle \mathbf{m}_i, U_j \rangle - \prod_{i=1}^{N} U_{i,j} \neq 0$.

$$EPC(U'_j, \tilde{\mathbf{r}}_j) = \prod_{i=1}^{N} \mathbf{c}_i^{U_{i,j}} = EPC(\mathbf{u}'', \langle \tilde{\mathbf{r}}A, U_j \rangle)$$

This completes the proof that M is a permutation matrix or we have found a witness to $\mathcal{R}_{com}$.

The correctness of U' We now show that $U'_l = MU_l$ for all $l \in [1, N]$ or we can find a witnesses to $\mathcal{R}_{com}$. Let $\mathbf{u}'' = MU_l$ and by assumption $\mathbf{u}'' \neq U'_l$.

$$EPC(U'_l, \tilde{\mathbf{r}}_l) = \prod_{i=1}^{N} \mathbf{c}_i^{U_{i,l}} = EPC(\mathbf{u}'', \langle \tilde{\mathbf{r}}A, U_l \rangle)$$

Extracting the randomness We having shown that if M is not a permutation matrix we can extract a witness to $\mathcal{R}_{com}$. We now show that we can extract

$R \in \mathcal{R}_{pk}$ such that $\mathbf{e}'_i = \mathbf{e}_{\pi(i)} \Sigma.\mathsf{Enc}_{PK}(0, R_{\pi(i)})$.

$$\mathbf{e}_l = \prod_{i=1}^{N} (\mathbf{e}^{UA_l})_i \qquad \text{since } UA_l \text{ is } \mathbb{I}_l \tag{9}$$

$$\mathbf{e}_l = \prod_{i=1}^{N} \mathbf{e}_i^{\sum_{j=1}^{N} U_{i,j} A_{j,l}} \qquad \text{by definition of } UA_l \tag{10}$$

$$\mathbf{e}_l = \prod_{i=1}^{N} \left(\prod_{j=1}^{N} \mathbf{e}^{U_{i,j} A_{j,l}} \right)_i \qquad \text{by algebraic manipulation} \tag{11}$$

$$\mathbf{e}_l = \prod_{j=1}^{N} \left(\prod_{i=1}^{N} \mathbf{e}^{U_{i,j}} \right)^{A_{j,l}} \qquad \text{by algebraic manipulation} \tag{12}$$

$$\mathbf{e}_l = \prod_{j=1}^{N} \left(\prod_{i=1}^{N} \mathbf{e}_i'^{U'_{i,j}} \Sigma.\mathsf{Enc}(0, -\mathbf{r}_j^*) \right)^{A_{j,l}} \qquad \text{since } \prod_{i=1}^{N} \mathbf{e}_i^{U_{i,j}} = \prod_{i=1}^{N} \mathbf{e}_i'^{U'_{i,j}} \Sigma.\mathsf{Enc}_{pk}(0, -\mathbf{r}_j^\star) \tag{13}$$

$$\mathbf{e}_l = \prod_{j=1}^{N} \left(\prod_{i=1}^{N} \mathbf{e}_i'^{U'_{i,j} A_{j,l}} \Sigma.\mathsf{Enc}_{pk}(1, -\mathbf{r}_j^* A_{j,l}) \right) \qquad \text{by algebraic manipulation} \tag{14}$$

$$\mathbf{e}_l = \prod_{i=1}^{N} \mathbf{e}_i'^{\sum_{j=1}^{N} U'_{i,j} A_{j,l}} \Sigma.\mathsf{Enc}_{pk}(1, -\langle \mathbf{r}^*, A_l \rangle) \qquad \text{by algebraic manipulation} \tag{15}$$

$$\mathbf{e}_l = \prod_{i=1}^{N} (\mathbf{e}'^{U' A_l})_i \Sigma.\mathsf{Enc}_{pk}(1, -\langle \mathbf{r}^*, A_l \rangle) \qquad \text{by algebraic manipulation} \tag{16}$$

$$\mathbf{e}_l = \prod_{i=1}^{N} (\mathbf{e}'^{MUA_l})_i \Sigma.\mathsf{Enc}_{pk}(1, -\langle \mathbf{r}^*, A_l \rangle) \qquad \text{since U' = MU} \tag{17}$$

$$\mathbf{e}_l = \prod_{i=1}^{N} (\mathbf{e}'^{M\mathbb{I}_l})_i \mathbf{Enc}_{pk}(1, -\langle \mathbf{r}^*, A_l \rangle) \qquad \text{since } UA_l = \mathbb{I}_l \tag{18}$$

$$\mathbf{e}_l = \prod_{i=1}^{N} (\mathbf{e}'^{M_l})_i \mathbf{Enc}_{pk}(1, -\langle \mathbf{r}^*, A_l \rangle) \qquad \text{since } M\mathbb{I}_l = M_l \tag{19}$$

$$\mathbf{e}_l = \mathbf{e}'_{\pi_M^{-1}(l)} \mathbf{Enc}_{pk}(1, -\langle \mathbf{r}^*, A_l \rangle) \qquad \text{by definition of } \pi_M \tag{20}$$

$$\tag{21}$$

We have now shown that $\mathbf{ReEnc}_{pk}(\mathbf{e}_l, \langle \mathbf{r}_l^*, A_l \rangle) = \mathbf{e}'_{\pi^{-1}(l)}$; hence, $R_l = \langle \mathbf{r}_l^*, A_l \rangle$ which concludes the proof. □

We can use the known technique of (strong) Fiat-Shamir Heurestic to securely convert the interactive variant to a non-interactive variant.

6 Comparison and Analysis of Efficiency

We study the efficiency of our solution and compare it with Cuvelier *et al.*'s results [12]. In order to accurately confront both schemes, we adopt the similar

conventions to Cuvelier *et al.* The commitments used by PPATC scheme [12] require an elliptic curve with a type 3 pairing to function. Type 3 pairing is a pairing in which there exist no efficiently computable homomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ and where the Decisional Diffie-Hellmen is hard in both groups. We assume an embedding degree of 16 such that elements of $\mathbb{G}_T$ are of size p^{16}. We, also, associate a unit cost to the multiplication of two 256 bit integers. While Cuvelier *et al.* supposed quadratic growth in the length of the operands, we assume $\mathcal{O}(n^{1.5})$, which better reflects that many BigInteger libraries support the optimised multiplication algorithms. We target a security level equivalent to 2048 bits RSA modulus N. We select $\mathbb{G}_1$ to be taken on $\mathbb{F}_p$ for a 256 bits long prime p and $\mathbb{G}_2$ to be taken on $\mathbb{F}_{p^3}$. The size of the target group is then 4096 bits, and for simplicity we take pairing to cost 10 times the effort of a multiplication in $\mathbb{G}_1$, this seems to hold for most real implementations.

We count the number of operations in Cuvelier *et al.*'s scheme and our solution. Tables 1 and 2 show these numbers for both encryption and opening verification. Let $Exp_{\mathbb{Z}_X^*}$ denote the number of exponentiations in $\mathbb{Z}_X^*$, and $Mult_{\mathbb{G}_Y}$ the number of multiplications in $\mathbb{G}_Y$. *Pairing* is defined as the number of pairing operations.

Scheme	$Exp_{\mathbb{Z}^*_{kn+1}}$	$Exp_{\mathbb{Z}^*_{n^2}}$	$Mult_{\mathbb{G}_1}$	$Mult_{\mathbb{G}_2}$	Total cost
MN [26]	3.375	4	0	0	1024896 multiplications
PPATC [12]	0	0	9	4	114432 multiplications

Table 1. Total number of operations executed for encryption - Total cost is obtained according to the implementation setting.

Scheme	$Exp_{\mathbb{Z}^*_{kn+1}}$	$Exp_{\mathbb{Z}^*_{n^2}}$	$Mult_{\mathbb{G}_1}$	$Mult_{\mathbb{G}_2}$	$Pairing$	Total cost
MN [26]	1.125	0	0	0	0	79488 multiplications
PPATC [12]	0	0	1	0	3	119040 multiplications

Table 2. Total number of operations executed for opening verification - Total cost is obtained according to the implementation setting.

While PPATC remains faster for the encryption phase than MN scheme, the latter is 1.5 time faster for the verification phase than PPATC. In regards to mixing, which is of course a very substantial part of the verification cost, we have already shown how an optimised variant of Terelius and Wikström's approach [31] can be applied to MN cryptosystem.

Cuvelier *et al.* [12] suggested using Terelius and Wikström's approach as well. However, the efficiency of their general construction is poor compared to the optimised variants (especially when dealing with groups of composite order). The PPATC scheme of Cuvelier *et al.* is a highly elegant construction but contrary to expectations is not more efficient overall than our version of MN scheme [26]. Though, if the voting devices were unusually weak PPATC might still be preferred. In conclusion, while PPATC might still be preferred in some settings, in others where homomorphic properties are desired MN scheme with our optimised ZKPs are of comparable efficiency.

7 Conclusion

Ongoing privacy is fundamental for the proper functioning of elections but significant gaps remained. We fixed several of the outstanding issues. We showed that the modified Pedersen commitment is in fact secure and proved that the Sigma Protocols for correct encryption and correct re-encryption are safe to use. We also provided computational improvements to mixing and examined the feasibility of a secure deployment of our solution. In doing this, we help make everlasting privacy for homomorphic electronic voting a computationally feasible and rigorously secure reality. We show that this approach provides verification efficiency comparable to the most efficient non-homomorphic schemes.

References

1. Abe, M., Haralambiev, K., Ohkubo, M.: Group to group commitments do not shrink. In: EUROCRYPT. LNCS, vol. 7237, pp. 301–317. Springer (2012)
2. Arapinis, M., Cortier, V., Kremer, S., Ryan, M.: Practical everlasting privacy. In: POST. LNCS, vol. 7796, pp. 21–40. Springer (2013)
3. Bangerter, E., Camenisch, J., Krenn, S.: Efficiency limitations for Σ-protocols for group homomorphisms. In: TCC. Lecture Notes in Computer Science, vol. 5978, pp. 553–571. Springer (2010)
4. Bayer, S., Groth, J.: Efficient zero-knowledge argument for correctness of a shuffle. In: EUROCRYPT. LNCS, vol. 7237, pp. 263–280. Springer (2012)
5. Bernhard, D., Cortier, V., Pereira, O., Smyth, B., Warinschi, B.: Adapting helios for provable ballot privacy. In: Atluri, V., Díaz, C. (eds.) Computer Security - ESORICS 2011 - 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6879, pp. 335–354. Springer (2011). https://doi.org/10.1007978-3-642-23822-2_19, `http://dx.doi.org/10.1007/978-3-642-23822-2_19`
6. Buchmann, J.A., Demirel, D., van de Graaf, J.: Towards a publicly-verifiable mix-net providing everlasting privacy. In: Financial Cryptography. Lecture Notes in Computer Science, vol. 7859, pp. 197–204. Springer (2013)
7. Burmester, M.: A remark on the efficiency of identification schemes. In: EUROCRYPT. LNCS, vol. 473, pp. 493–495. Springer (1990)
8. Chaum, D.: Untraceable mail, return addresses and digital pseudonyms. Communications of the ACM **24(2)**, 84–88 (1981)
9. Cortier, V., Smyth, B.: Attacking and fixing helios: An analysis of ballot secrecy. Journal of Computer Security **21**(1), 89–148 (2013)
10. Cramer, R.: Modular design of secure yet practical cryptographic protocols. PhD thesis, Aula der Universiteit (1996)
11. Cramer, R., Gennaro, R., Schoenmakers, B.: A secure and optimally efficient multi-authority election scheme. European Transactions on Telecommunications **8**(5), 481–490 (1997). https://doi.org/10.1002/ett.4460080506, `http://dx.doi.org/10.1002/ett.4460080506`
12. Cuvelier, E., Pereira, O., Peters, T.: Election verifiability or ballot privacy: Do we need to choose? In: ESORICS. pp. 481–498. LNCS (2013)
13. Damgård, I.: On Σ-protocols. `http://www.daimi.au.dk/ivan/Sigma.pdf` (2010)
14. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In: Public Key Cryptography. Lecture Notes in Computer Science, vol. 1992, pp. 119–136. Springer (2001)

15. Damgård, I., Jurik, M., Nielsen, J.B.: A generalization of paillier's public-key system with applications to electronic voting. Int. J. Inf. Sec. **9**(6), 371–385 (2010)
16. Demirel, D., Henning, M., van de Graaf, J., Ryan, P.Y.A., Buchmann, J.A.: Prêt à voter providing everlasting privacy. In: VOTE-ID. LNCS, vol. 7985, pp. 156–175. Springer (2013)
17. Demirel, D., Van De Graaf, J., Araújo, R.: Improving helios with everlasting privacy towards the public. In: Electronic Voting Technology/Workshop on Trustworthy Elections. pp. 8–8. USENIX Ass. (2012)
18. Fischlin, M.: Communication-efficient non-interactive proofs of knowledge with online extractors. In: CRYPTO. Lecture Notes in Computer Science, vol. 3621, pp. 152–168. Springer (2005)
19. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Theory of Computing. pp. 291–304. ACM (1985)
20. Groth, J.: A verifiable secret shuffe of homomorphic encryptions. In: International Workshop on Public Key Cryptography. pp. 145–160. Springer (2003)
21. Haines, T.: A description and proof of a generalised and optimised variant of wikstr\" om's mixnet. arXiv preprint arXiv:1901.08371 (2019)
22. Haines, T., Boyen, X.: Votor: conceptually simple remote voting against tiny tyrants. In: Proceedings of the Australasian Computer Science Week Multiconference. p. 32. ACM (2016)
23. Hazay, C., Mikkelsen, G.L., Rabin, T., Toft, T., Nicolosi, A.A.: Efficient RSA key generation and threshold paillier in the two-party setting. J. Cryptology **32**(2), 265–323 (2019)
24. Locher, P., Haenni, R.: Receipt-free remote electronic elections with everlasting privacy. Annales des Télécommunications **71**(7-8), 323–336 (2016)
25. Locher, P., Haenni, R., Koenig, R.E.: Coercion-resistant internet voting with everlasting privacy. In: International Conference on Financial Cryptography and Data Security. pp. 161–175. Springer (2016)
26. Moran, T., Naor, M.: Split-ballot voting: Everlasting privacy with distributed trust. ACM Trans. Inf. Syst. Secur. **13**(2), 16:1–16:43 (2010)
27. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT. pp. 223–238. Springer (1999)
28. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: CRYPTO. LNCS, vol. 576, pp. 129–140. Springer (1991)
29. Shoup, V.: On the security of a practical identification scheme. J. Cryptology **12**(4), 247–260 (1999)
30. Terelius, B.: Some aspects of cryptographic protocols: with applications in electronic voting and digital watermarking. Ph.D. thesis, KTH Royal Institute of Technology (2015)
31. Terelius, B., Wikström, D.: Proofs of restricted shuffles. In: AFRICACRYPT. pp. 100–113. Springer (2010)
32. Terelius, B., Wikström, D.: Efficiency limitations of Σ-protocols for group homomorphisms revisited. In: SCN. Lecture Notes in Computer Science, vol. 7485, pp. 461–476. Springer (2012)
33. Wikström, D.: A commitment-consistent proof of a shuffle. In: Information Security and Privacy. pp. 407–421. Springer (2009)
34. Yang, N., Clark, J.: Practical governmental voting with unconditional integrity and privacy. In: Brenner, M., Rohloff, K., Bonneau, J., Miller, A., Ryan, P.Y.A., Teague, V., Bracciali, A., Sala, M., Pintore, F., Jakobsson, M. (eds.) Financial Cryptography and Data Security - FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected

Papers. Lecture Notes in Computer Science, vol. 10323, pp. 434–449. Springer (2017)
35. Yao, A.C.: Protocols for secure computations (extended abstract). In: FOCS. pp. 160–164. IEEE Computer Society (1982)