Securing Abe's Mix-Net Against Malicious Verifiers via Witness Indistinguishability

Elette Boyle[1(✉)], Saleet Klein[2(✉)], Alon Rosen[1(✉)], and Gil Segev[3(✉)]

[1] IDC Herzliya, Herzliya, Israel
eboyle@alum.mit.edu, alon.rosen@idc.ac.il
[2] MIT, Cambridge, USA
saleet@csail.mit.edu
[3] Hebrew University, Jerusalem, Israel
segev@cs.huji.ac.il

Abstract. We show that the simple and appealing unconditionally sound mix-net due to Abe (Asiacrypt'99) can be augmented to further guarantee anonymity against malicious verifiers.

As our main contribution, we demonstrate how anonymity can be attained, even if most sub-protocols of a mix-net are merely witness indistinguishable (WI). We instantiate our framework with two variants of Abe's mix-net. In the first variant, ElGamal ciphertexts are replaced by an alternative, yet comparably efficient, "lossy" encryption scheme. In the second variant, new "dummy" vote ciphertexts are injected prior to the mixing process, and then removed.

Our techniques center on new methods to introduce additional witnesses to the sub-protocols within the proof of security. This, in turn, enables us to leverage the WI guarantees against malicious verifiers. In our first instantiation, these witnesses follow somewhat naturally from

E. Boyle—Supported by ISF grant 1861/16, AFOSR Award FA9550-17-1-0069, and ERC starting grant 307952.

S. Klein—This work was done in part while visiting at the IDC Herzliya FACT Center, supported by ERC starting grant 307952. Additionally supported by an Akamai Presidential Fellowship, by the NSF MACS - CNS-1413920, by the DARPA IBM - W911NF-15-C-0236, by the SIMONS Investigator award Agreement Dated 6-5-12, and by the MISTI MIT-Israel Seed Fund.

A. Rosen—Supported by ISF grant no. 1255/12, NSF-BSF Cyber Security and Privacy grant no. 2014/632, by the ERC under the EU's Seventh Framework Programme (FP/2007-2013) ERC Grant Agreement n. 307952, and by the MISTI MIT-Israel Seed Fund.

G. Segev—Supported by the European Union's 7th Framework Program (FP7) via a Marie Curie Career Integration Grant (Grant No. 618094), by the European Union's Horizon 2020 Framework Program (H2020) via an ERC Grant (Grant No. 714253), by the Israel Science Foundation (Grant No. 483/13), by the Israeli Centers of Research Excellence (I-CORE) Program (Center No. 4/11), by the US-Israel Binational Science Foundation (Grant No. 2014/632), and by a Google Faculty Research Award.

D. Catalano and R. De Prisco (Eds.): SCN 2018, LNCS 11035, pp. 274–291, 2018.
https://doi.org/10.1007/978-3-319-98113-0_15

the lossiness of the encryption scheme, whereas in our second instantiation they follow from leveraging combinatorial properties of the Beneš-network. These approaches may be of independent interest.

Finally, we demonstrate cases in Abe's original mix-net (without modification) where only one witness exists, such that if the WI proof leaks information on the (single) witness in these cases, then the system will not be anonymous against malicious verifiers.

1 Introduction

A mix-net, introduced by Chaum [Cha81], is a means to provide anonymity for a set of users. It has become a central tool for electronic voting, in which each voter submits an encrypted vote and the mix-net outputs the same set of votes in randomized order. Mix-nets have also found applications in other areas, including anonymous web browsing [GGMM97], payment systems [JM98], and as a building block for secure multi-party computation [JJ00].

In some cases, for instance for electronic voting, the mix-net is required to be *verifiable*. That is, the mixing process should be accompanied by a proof that does not violate anonymity (traditionally, zero-knowledge), and at the same time convinces that the set of votes (alternatively, the vote tally) was preserved following the mixing process (soundness). Much work has been devoted to optimizing the running times of protocols, resulting in highly efficient solutions (e.g., [Nef01, GI08, Wik09, TW10, BG12]). At the same time, the strive for efficiency has almost always required assuming that verifying parties act honestly.

While there exist relatively simple methods for enforcing honest verifier behavior, very often the verifier ends up being replaced with some concrete "challenge-generating" hash function that is modeled as a random oracle. This transformation (known as the *Fiat-Shamir transform* [FS86]) only provides heuristic guarantees for anonymity, as any concrete instantiation of a hash function is far from behaving randomly (and consequently is far from emulating the behavior of an honest verifier). Moreover, there is indication that when applied to *computationally sound* protocols (which include all known mix-nets with sub-linear verification) it may result in loss of soundness [GK03].

The primary reason known efficient solutions require assuming honest verifiers is that they achieve anonymity by requiring underlying protocols to be *zero-knowledge* (ZK). In some sense this is an overkill, since it may be possible to guarantee anonymity of the overall system even if some of its building blocks do not satisfy such a strong security notion. One prime example is given by Feige and Shamir, who demonstrated how to construct 4-round ZK arguments for NP by invoking sub-protocols that satisfy the notion of *witness indistinguishability* (WI) [FS90]. In contrast to ZK, WI protocols are only required to hide which of the (possibly many) NP-witnesses is used in the protocol execution. This weaker notion gives rise to very simple and consequently efficient constructions, secure even against *malicious* verifiers and sound even against *computationally unbounded* provers.

1.1 This Work

The goal of this work is to explore the possibility of constructing a simple mix-net that is secure against malicious verifiers and in addition is unconditionally sound. This would in particular mean that when applying the Fiat-Shamir transform to the proofs in the mix-net, anonymity would provably be guaranteed for *any choice* of a hash function. While soundness would still be heuristic, unconditional soundness of the protocols makes them less susceptible to theoretical doubts cast on the Fiat-Shamir transform in the case of certain computationally sound protocols [GK03].

Towards this end, we aim for a relaxed indistinguishability-based notion of anonymity, which is weaker than zero-knowledge and yet guarantees the privacy of voters in the system. We demonstrate how indistinguishability-based anonymity of an entire mix-net system can be attained, even if most of the underlying sub-protocols are merely WI. At the core of our analysis are new techniques for guaranteeing the existence of multiple witnesses in NP-verification relations upon which the soundness of mix-nets is based.

We instantiate our ideas with a very simple and appealing Beneš-network based construction due to Abe [Abe99, AH01]. While this construction does not match the sublinear verification efficiency of later mix-nets in the literature (verification time is quasi-linear in the number of voters), it does enjoy a number of desirable features, most notably high parallelizability. In addition, proving and verifying consists of invoking standard and widely used proofs of knowledge, making the mix-net easy to understand and implement.

Abe's mix-net was originally shown to be anonymous assuming honest verifiers, and specifically based on the honest-verifier ZK property of the underlying proofs of knowledge. In the case of a *malicious* verifier, these sub-protocols are known only to be witness indistinguishable; alas, this guarantees nothing in cases where there is a single witness. Moreover, (as we show) in Abe's mix-net, cases in which only one witness exists *cannot be ruled out*, and if indeed leakage on the single witness occurs in these situations we demonstrate that the system is *not anonymous*.

1.2 Our Results

We propose two different methods for modifying Abe's original proposal that result in a verifiable mix-net anonymous against malicious verifiers and sound against computationally unbounded provers. Both methods require only minor changes to Abe's original protocol:

Lossy Abe mix-net: This encryption is identical to Abe's original proposal, with the only difference being that plain ElGamal encryption is replaced with an alternative, yet comparably efficient, encryption scheme with the property that public-keys can be sampled using a "lossy" mode (this mode is only invoked in the analysis). When sampled with lossy public-keys, encrypted ciphertexts do not carry any information about the plaintext. (The same

property can also be satisfied by the Goldwasser-Micali QR-based, Paillier's DCR-based and Regev's LWE-based, encryption schemes.)

Injected Abe mix-net: This method consists of running the original Abe mix-net with additional dummy ciphertexts that are injected to the system for the purpose of proving D-WI without having to modify and/or assume anything about the encryption scheme in use (beyond it being re-randomizable). The analysis of this construction relies on combinatorial properties of the Beneš-network, and may turn out to be relevant elsewhere.

These modifications correspond to two approaches for introducing additional witnesses to the sub-protocols of the mix-net verification: In the first, the extra witnesses follow from the lossiness of the encryption scheme, and in the second they follow by leveraging combinatorial properties of the Beneš-network.

In both cases, we show that the entire transcript of the mix-net system satisfies the following natural anonymity property (in the style of [NSK04]): *for any choice of votes and any two permutations on the votes, the corresponding views of an adversary are computationally indistinguishable.*

We allow the adversary to control all but one of the mix-servers, an arbitrary subset of the voters, subset of the decryption servers, and the verifier. If the adversary controls a subset of the voters, then our definition quantifies over any two permutations that are consistent on the votes that it controls. Note that this anonymity notion completely hides information about which honest voter placed which vote from the collective set of honest votes (which is necessarily revealed by the shuffled output).

Theorem. *The Lossy and Injected Abe mix-nets are anonymous against malicious verifiers.*

Our result assumes the availability of a non-malleable (more precisely, plaintext aware) encryption scheme, under which the ciphertexts are encrypted, and an efficient secure (simulatable) multi-party protocol for threshold decryption of the ciphertexts. The latter building blocks can be constructed in an efficient manner, even if participating parties are malicious, and moreover are routinely assumed available in the cryptographic voting literature.

In a precise strong sense, the modifications introduced in the lossy and injected versions of Abe's mix-net are necessary for achieving anonymity in the case of a malicious verifier.

1.3 Technical Overview

In what follows, we provide further background on verifiable mix-nets and Abe's mix-net construction, and then describe the main technical ideas behind our results.

Verifiable Mix-Nets. Ideally, a mix-net is a protocol that completely breaks the link between a user and the vote she has submitted. This remains true even

if a subset of users share their votes with each other with the purpose of "de-anonymizing" votes of users outside their coalition.

In principle, if one is only interested in the tally, then a simple way to protect anonymity of individual voters would be to output the tally $\sum_i v_i$. However, specifically designing a protocol to meet this functionality limits its applicability in case one is interested in alternative tallying mechanisms. Further, such tallying solutions relying on homomorphic encryption either limit the message size or require the use of relatively complicated zero-knowledge procedures for proving the submitted vote encryptions are well formatted.

Mix-Net Phases. The operating assumption underlying most known mix-net constructions is that some vote-encryption mechanism is in place, resulting in a list $c_1, \ldots, c_n$ of ciphertexts where $c_i = \mathsf{Enc}_{\mathsf{pk}}(v_i; r_i)$ is an encryption of v_i with randomness r_i, under a public key pk that corresponds to a certain polling station. The output of the mix-net is a shuffled list of plaintexts $v_{\sigma(1)}, \ldots, v_{\sigma(n)}$, and we want a mix-net that hides σ even if malicious entities were involved in the mixing phase, the input phase, and the verification phase.

The public key pk is jointly generated and certified in a distributed manner by a set of trustees, so that no individual entity (or even any sufficiently small coalition of entities) is able to decrypt its corresponding ciphertexts. The assumption is that a large subset of the trustees acts as prescribed by the set-up protocol. We note that such an assumption is standard in the literature, and it does not necessitate the generation of a common reference string, at least not in its most general form.

Given such a setup, most known verifiable mix-net constructions can be conceptually decomposed to the following three stages:

Submit Ciphertexts: Each of the n users publishes their own ciphertext c_i on an authenticated bulletin board. For simplicity it is convenient to assume that the encryption is "non-malleable" (in fact, plaintext aware), which guarantees that voters cannot make their own vote depend on others'.

Verifiably Mix: Ciphertexts $c_1, \ldots, c_n$ are:

- re-randomized (i.e. $\mathsf{Enc}_{\mathsf{pk}}(v_i; r_i)$ is mapped to $\mathsf{Enc}_{\mathsf{pk}}(v_i; s_i)$ for random and independent s_i) and then
- randomly shuffled to obtain ciphertexts $c'_{\pi(i)} = \mathsf{Enc}_{\mathsf{pk}}(v_i; s_i)$, where π is randomly sampled from S_n.

In addition, the mixing party provides a proof that the set of plaintexts underlying the output ciphertexts $c'_1, \ldots, c'_n$ equals the original set of submitted votes underlying $c_1, \ldots, c_n$.

Decrypt: The ciphertexts $c'_1, \ldots, c'_n$ are collectively decrypted by means of a secure distributed protocol.

In terms of complexity, the Submit Ciphertexts and Decrypt stages can be implemented in time $O(n)$. Moreover, by using lightweight protocols for threshold decryption, the Decrypt phase can be implemented in a zero-knowledge fashion without paying much penalty in terms of efficiency. In light of this, much of the

literature (including the present work) focuses on optimizing the efficiency of the Verifiably Mix stage.

A naïve implementation would require work proportional to $O(n^2)$ (by proving consistency of individual input-output ciphertext pairs). Remarkably, it has been shown how to achieve perfect ZK with verification time as little as $o(n)$ (see [Nef01, GI08, Wik09, Gro10, TW10, BG12] to name a few). As we mentioned above, in many cases this comes at the price of the assumption that the prover is computationally bounded and that verification is performed as prescribed.

Abe's Mix-Net. Abe presented [Abe99, AH01] a simple mix-net construction which performs the Verifiably Mix stage on user ciphertexts via a sequence of pairwise ciphertext rerandomize-and-swap operations, as dictated by a *Beneš permutation network*. A d-dimensional Beneš network is a "butterfly" switching network on $n = 2^d$ inputs, consisting of $(2d - 1)$ levels of $\frac{n}{2}$ switch gates. Given any permutation $\pi \in S_n$, this permutation can be implemented via some (efficiently determined) choice of the control bits for each of the switch gates, where 0 at a gate indicates its input are output in order and 1 indicates its inputs are swapped.

In Abe's mix-net construction, the mixing entity samples a random permutation $\pi \leftarrow S_n$, and identifies a corresponding choice of Beneš control bits. Then, implementing and proving the validity of the overall n-input mix reduces to the same task on each of the $O(n \log n)$ individual switch gates in its Beneš representation. Namely, the overall proof is simply a collection of independent proofs that an individual rerandomize-and-switch gate operation preserved the plaintext values underlying its input ciphertexts.

For many common encryption schemes, this simple statement structure yields lightweight proofs of knowledge. For example, for ElGamal encryption (as considered by Abe), such a proof can be attained with 3 rounds by combining the Chaum-Pedersen protocol [CP92], which proves the equality of two discrete logarithms, with the protocol used in [CDS94], which proves two statements connected by OR, overall costing about four times as much computation as a single Chaum-Pedersen protocol execution.

However, lightweight protocols of this kind (inherently) provide only witness indistinguishability guarantees and/or *honest verifier* zero knowledge. Because of this, the mix-net of Abe was only proved to possess these properties as well.

Techniques and Ideas. To prove anonymity of our constructions, we must prove for any vector of votes $\boldsymbol{v} = (v_1, \ldots, v_n)$, and any permutation π of the honest parties, that the view of a (possibly *malicious*) verifier in the mix-net proof of correctness executed on votes $(v_1, \ldots, v_n)$ is indistinguishable from the analogous view on initial votes $(v_{\pi(1)}, \ldots, v_{\pi(n)})$. That is, intuitively, the verifier cannot distinguish which of the honest votes came from which honest party.

The semantic security of the encryption scheme directly allows us to "swap out" the starting honest-party vote encryptions themselves. So the core task is showing that interaction with an honest mix-server proving proper execution of

random permutation σ on encryptions of $(v_1, \ldots, v_n)$ is indistinguishable from an analogous proof executing $\sigma \circ \pi^{-1}$ on encryptions of $(v_{\pi(1)}, \ldots, v_{\pi(n)})$. The main difficulty in doing so arises for adversaries who have partial control of the votes: specifically, when the adversary controls a subset $\{v_i\}_{i \in A}$ of votes for some arbitrary $A \subseteq [n]$ of his choice.

Recall that Abe's construction is composed of a collection of underlying proofs of knowledge, where each individual sub-protocol is WI. Consider the proof for a single switch gate. To leverage the WI property, we must arrive to a state where the corresponding gate-validity statement $((c_1, c_2), (c'_1, c'_2))$ has at least two witnesses. This aligns precisely with the case in which the two input ciphertexts (c_1, c_2) of the gate have the *same underlying plaintext*. In such case, one could have reached the output ciphertexts (c'_1, c'_2) either by simply rerandomizing directly, or by swapping first and then rerandomizing (with different randomness); conversely, if the input plaintexts differ then by the correctness of the encryption scheme there is a unique witness.

Now, suppose we are in the case of a gate where both input ciphertexts c_1, c_2 correspond to encrypted votes of *honest* users. Then although the underlying votes of the two users may disagree, by relying on the semantic security of the encryption scheme, we can argue that the adversary cannot distinguish this state from the one in which the votes *do* agree. Once in this modified version of the world, we can invoke the WI guarantee to argue that the proof hides the identity of the swap bit. A similar approach can further take care of the situation where a single input ciphertext to a gate is controlled by the adversary (by changing the honest ciphertext to agree with the adversary's fixed vote).

What poses an issue is when *both* input ciphertexts to a gate are under adversarial control. The adversary can then force the gate to have a single witness, by choosing different plaintext votes. (Note we cannot hope to invoke semantic security arguments as above, as the adversary generates the ciphertexts himself). In such a case, for all that is known, the underlying protocol may very well leak the control bit of this gate. Interestingly, we demonstrate that such leakage, while directly regarding only corrupt-party ciphertexts, would be fatal to anonymity of *honest* parties in Abe's mix-net (see full version on eprint[1]).

We address this issue via two alternative proposed modifications to Abe's protocol.

Using a lossy encryption scheme. In the first variant, we instantiate the encryption scheme within Abe's mix-net with a DDH-based *lossy* encryption scheme that admits a similar underlying WI gate-consistency proof. A lossy encryption scheme has the property that standard key generation is indistinguishable from a "lossy" version, such that encryption under a lossy key pk completely loses all information about the message. In particular, for a lossy pk, for *any* pair of ciphertexts c, c' (not necessarily formed by encrypting the same message), there exists a choice of re-randomization that takes c to c'. This means for a lossy key that for *any* switch gate tuple (c_1, c_2, c'_1, c'_2), there

[1] https://eprint.iacr.org/2017/544.

necessarily exist two witnesses.

The proof of anonymity then follows from four simple steps. First, the public key is replaced by a lossy version. Then, once we are under a lossy pk, we can directly use the WI of the underlying gate protocols to switch (gate by gate) from a Beneš representation of a starting permutation π to the representation of any other permutation σ. Additionally, by the guaranteed hiding, we can switch the plaintexts of honest users' votes to an arbitrary shuffle amongst themselves. Once we attain the desired permutation and plaintext settings, we simply return back to a standard (non-lossy) pk.

Injecting and removing "dummy" votes. In the second variant, we consider an arbitrary rerandomizable public-key encryption scheme (e.g., standard ElGamal), and instead modify the Abe mix protocol at a higher level. Interestingly, the design approach leverages the combinatorial structure of the Beneš network, without modifying the underlying building block proofs of knowledge (for which it is not known how to prove an analogous property). The new mixing procedure begins by generating and injecting n "dummy" votes (i.e., encryptions of a fixed non-vote message $\perp$) into the list of n real encrypted votes. Abe's mix phase is performed (without modification) on the combined list of $2n$ ciphertexts (injecting the $\perp$ ciphertexts into the even-indexed positions). Then, Abe's Decrypt protocol is performed on *all* $2n$ resulting ciphertexts, and the $\perp$ plaintexts are identified and removed. Verification consists of Abe's standard verification, plus a process for verifying that $\perp$ ciphertexts were properly injected and removed in each mix step. We remark that this modification of injecting non-adversarial ciphertexts into the even-indexed positions does not directly preclude gates within the Beneš execution whose input ciphertexts are both under adversarial control; indeed, this remains quite likely to occur in many locations within later levels of the Beneš network. However, leveraging the combinatorial Beneš structure, we prove that the power we gain by ensuring the *first-level* gates do not have this problem, is sufficient to hide all control bits used within the Beneš network.

Our proof takes an inductive approach, on the dimension d (i.e., number of users $n = 2^d$) of the Beneš network. Ultimately, we design a carefully ordered sequence of hybrids which enables us to step from honest input votes $\boldsymbol{u}_{\mathsf{honest}}$ and permutation $\pi \in S_n$ to an arbitrary other choice $\boldsymbol{u}'_{\mathsf{honest}}, \sigma$. In essence, for each gate g in the Beneš network whose control bit we would like to flip, we: (1) switch the control bits of relevant first-level gates to ensure at least one non-adversarial ciphertext c_i becomes directed to gate g; (2) rely on semantic security to change the plaintext underlying c_i to agree with its neighboring ciphertext c_j into g; and then (3) use the WI to flip the control bit of gate g, now that we have forced the existence of 2 witnesses. This procedure is performed on gates in a particular order to ensure progress is made in each step, while leaving sufficient flexibility to enable that the step (1) redirection can be performed.

2 Indistinguishability-Based Anonymity of Mix-Nets

In this section we discuss the property of *anonymity* of a mix-net system, which is our main focus. Due to space limitations, in the full version (see footnote 1) we provide a complete definition of the standard syntax, correctness, and verifiability properties of a mix-net system $(\mathsf{Setup}, \mathsf{SubmitCipher}, \mathsf{VrfblyMix}, \mathsf{Decrypt}, (\mathcal{P}, \mathcal{V}))$ (as discussed informally in Sect. 1.3).

A wide range of anonymity notions have been considered within the mix-net literature, ranging from addressing specific anonymity attacks, to very strong notions of universally composable (UC) simulation.

In particular, the mix-net of Abe was proved in [Abe99, AH01, AI06] to satisfy the following anonymity notion: An efficient adversary who corrupts a subset of users, mix-servers, and decryption servers cannot gain noticeable advantage in predicting any single input-output pair $(i, j) \in [n]^2$ for which honest user i's encrypted plaintext is permuted to position j in the output. Note that this definition protects the anonymity of each user, but is weaker than more general indistinguishability and simulation definitions, in that it could potentially reveal correlations between users (e.g., that users 2 and 3 voted in the same fashion).

We consider a stronger indistinguishability-based notion of anonymity, in the flavor of [NSK04]. Intuitively, our definition requires that for any permutation on the honest users' votes, the resulting views of the mix-net protocol and verification—including the view of a *possibly corrupt* verifier—are indistinguishable.[2] Note that this implies the anonymity definition of Abe [Abe99, AH01, AI06], as a successful (i, j)-predicting adversary would serve as a successful distinguisher between views for permutations σ, σ' which disagree on user i.

We formalize this notion via a notion of *distributional WI* (D-WI), a strengthening of WI we introduce that is related to strong-WI [Gol01], but parametrized by specific pairs of distributions.

Distributional Witness Indistinguishabilty. For ease of reading, we will make use of the following shorthand notation for the distribution over the view of a (potentially malicious) verifier $\mathcal{V}$ within an interactive proof $(\mathcal{P}, \mathcal{V})$ for a given distribution over statements (and witnesses).

Notation 1 ($\mathsf{View}_{\mathcal{V}^*}[D_\lambda]$)**.** *Let $(\mathcal{P}, \mathcal{V})$ be an interactive proof for a relation R. For a given ensemble of distributions D_λ over statements, witnesses, and auxiliary input $\{(X_\lambda, W_\lambda, Z_\lambda)\}_{\lambda \in \mathbb{N}}$ for which $(X_\lambda, W_\lambda) \in \mathsf{R}$ and $|X_\lambda| \geq \lambda$, and PPT interactive machine $\mathcal{V}^*$, we define the distribution*

$$\mathsf{View}_{\mathcal{V}^*}[D_\lambda] := \{\langle \mathcal{P}(W_\lambda), \mathcal{V}^*(Z_\lambda)\rangle (X_\lambda) : (X_\lambda, W_\lambda, Z_\lambda) \leftarrow D_\lambda\}_{\lambda \in \mathbb{N}}.$$

[2] We remark, however, that [Abe99, AH01, AI06] directly consider non-malleability concerns, which we factor out and address separately; see Remark on Non-Malleability below. Note that Abe and Imai considered notions of anonymity against both static and *adaptive* adversaries [AI06]; however, anonymity of Abe's mix-net construction was proven only in the static setting [Abe99, AH01], and thus this is the notion we compare against.

Definition 1 (D-WI). *Let $(\mathcal{P}, \mathcal{V})$ be an interactive proof for a relation* R*, and let D_λ and D'_λ be two probability ensembles over statements, witnesses, and auxiliary inputs, as in Notation 1. We say that $(\mathcal{P}, \mathcal{V})$ is* distributional witness-indistinguishable (D-WI) with respect to D_λ, D'_λ *for relation* R *if for every PPT interactive machine $\mathcal{V}^*$, the following holds:* $\mathsf{View}_{\mathcal{V}^*}[D_\lambda] \overset{c}{\approx} \mathsf{View}_{\mathcal{V}^*}[D'_\lambda]$.

Mix-Net Anonymity. For a given mix-net protocol MixNet, adversarial entity A, and vector of honest user votes $(u_i)_{i \in \mathcal{U}_{\bar{A}}}$, the distribution $D_\lambda^{\mathsf{MixNet},\mathcal{A}}((u_i)_{i \in \mathcal{U}_{\bar{A}}})$ as given in Definition 2 denotes the induced distribution over statements, witnesses, and auxiliary input of correctness of the mix-net. Our notion of anonymity (Definition 3) requires the interactive proof system for correctness of the mix-net to be distributional witness indistinguishable (D-WI) with respect to any pair $D_\lambda^{\mathsf{MixNet},A}((u_i)_{i \in \mathcal{U}_{\bar{A}}})$ and $D_\lambda^{\mathsf{MixNet},A}((u_{\sigma(i)})_{i \in \mathcal{U}_{\bar{A}}})$, for any permutation σ on the ordering of the honest users.

Definition 2 ($D_\lambda^{\mathsf{MixNet},A}$ distribution). *Let* $\mathsf{MixNet} = (\mathsf{Setup}, \mathsf{SubmitCipher}, \mathsf{VrfblyMix}, \mathsf{Decrypt})$ *be a verifiable n-user m-server mix-net system with respect to a re-randomizable encryption scheme $\mathcal{E}$ over message space M*

Let $A = (\mathcal{U}_A, \mathcal{S}_A, \mathcal{A})$ be given, where $\mathcal{U}_A \subseteq [n]$, $\mathcal{S}_A \subset [m]$ are corrupted subsets of users and mix-servers, respectively, and $\mathcal{A}$ is an adversarial non-uniform PPT algorithm which has four modes $\mathsf{setup}, \mathsf{submit\ votes}, \mathsf{mix}$*, and* $\mathsf{decrypt}$ *with the syntax as below. We define the distribution $D_\lambda^{\mathsf{MixNet},A}$ as follows (we denote $\mathcal{U}_{\bar{A}} = [n] \setminus \mathcal{U}_A$ and $\mathcal{S}_{\bar{A}} = [m] \setminus \mathcal{S}_A$):*

$D_\lambda^{\mathsf{MixNet},A}((u_i)_{i \in \mathcal{U}_{\bar{A}}})$:

Input: For each honest user $i \in \mathcal{U}_{\bar{A}}$, a vote $u_i \in \{0, 1\}$.

- Let $\mathsf{state} := \emptyset$
- Sample $(\mathsf{pk}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_m), \mathsf{state}) \leftarrow \mathsf{Setup}^{\mathcal{A}(\text{“setup”}, 1^\lambda, \mathsf{state})}(1^\lambda)$,
  i.e., simulate Setup protocol execution on honest party input 1^λ and (oracle access to) adversarial next-message function $\mathcal{A}(\text{“setup”}, 1^\lambda, \mathsf{state})$, in each round with updated state. Output the induced values $\mathsf{pk}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_m)$, and updated state.
- For each $i = 1, \ldots, n$: // Submit votes (n users)
  if $i \in \mathcal{U}_A$
  then Sample $(c_i^0, z_i) \leftarrow \mathcal{A}(\text{“submit votes”}, \mathsf{pk}, i, \mathsf{state})$
  Update $\mathsf{state} := \{z_i\} \cup \mathsf{state}$
  else Sample $c_i^0 \leftarrow \mathsf{SubmitCipher}(\mathsf{pk}, u_i)$
- For $j = 1, \ldots, m$ do: // Mix phase (m mix servers)
  if $j \in \mathcal{S}_A$
  then Sample $(\boldsymbol{c}^j, w_j^\pi, z_j^\pi) \leftarrow \mathcal{A}(\text{“mix”}, \mathsf{pk}, j, \boldsymbol{c}^{j-1}, \mathsf{state})$
  Update $\mathsf{state} := \{z_j^\pi\} \cup \mathsf{state}$
  else Sample $\mathsf{rnd}_j \leftarrow \$$, and set $\boldsymbol{c}^j = \mathsf{Mix}_j(\mathsf{pk}, \boldsymbol{c}^{j-1}; \mathsf{rnd}_j)$
- Run $(\boldsymbol{v}, (w_j^{\mathsf{sk}})_{j \in \mathcal{S}_A}, \mathsf{state}) \leftarrow \mathsf{Decrypt}^{\mathcal{A}(\text{“decrypt”}, \mathsf{state})}(\boldsymbol{c}^m, (\mathsf{sk}_j)_{j \in \mathcal{S}_{\bar{A}}})$,
  i.e., simulate $\mathsf{Decrypt}$ protocol execution on input $(\boldsymbol{c}^m, \mathsf{sk}_j)$ for each honest mix-server j, and oracle access to adversarial next-message function

$\mathcal{A}(\text{“decrypt”}, \mathsf{state})$, in each round with updated state. Output the induced plaintext vector $\boldsymbol{v}$, adversarial witness information $(w_j^{\mathsf{sk}})_{j \in \mathcal{S}_A}$ for decryption, and updated state.

Output: $(X_\lambda, W_\lambda, Z_\lambda)$ where
- $X_\lambda = (\mathsf{pk}, \boldsymbol{c}^0, \boldsymbol{v})$
- $W_\lambda = ((\mathsf{rnd}_j)_{j \in \mathcal{S}_{\bar{A}}}, (w_j^\pi)_{j \in \mathcal{S}_A}, (\mathsf{sk}_j)_{j \in \mathcal{S}_{\bar{A}}}, (w_j^{\mathsf{sk}})_{j \in \mathcal{S}_A})$
- $Z_\lambda = (\mathsf{state})$

Definition 3 (*Anonymous* Mix-Net System). *We say that a verifiable n-user m-server mix-net system* MixNet *is* anonymous *if for every A (as in Definition 2), every choice of honest user votes $u_i \in \{0,1\}$ for $i \in \mathcal{U}_{\bar{A}}$, and every permutation σ over the honest users $\mathcal{U}_{\bar{A}}$ (i.e. $\sigma : \mathcal{U}_{\bar{A}} \hookrightarrow \mathcal{U}_{\bar{A}}$) the interactive proof system $(\mathcal{P}, \mathcal{V})$ for correctness of* MixNet *is D-WI with respect to the following two probability ensembles $D_\lambda = D_\lambda^{\mathsf{MixNet},A}((u_i)_{i \in \mathcal{U}_{\bar{A}}})$ and $D'_\lambda = D_\lambda^{\mathsf{MixNet},A}((u_{\sigma(i)})_{i \in \mathcal{U}_{\bar{A}}})$ where $D_\lambda^{\mathsf{MixNet},A}$ is as in Definition 2.*

3 Abe's Mix-Net with Lossy Encryption

For our first mix-net construction, we consider an implementation of Abe with a modified *lossy ElGamal* encryption scheme. In Sect. 3.1 we present the additional necessary building blocks, and in Sect. 3.2 we provide our construction.

3.1 Building Blocks for Lossy Abe

A *lossy encryption scheme* [PVW07] $(\mathsf{KeyGen}, \mathsf{KeyGen}_{\mathsf{loss}}, \mathsf{Enc}, \mathsf{Dec})$ is a PKE scheme which possesses an alternative "lossy mode" key generation algorithm $\mathsf{KeyGen}_{\mathsf{loss}}$, whose output pk is computationally indistinguishable from an honestly generated pk, but for which the encryption of a message m information theoretically hides m.

We make use of the following lossy variant of ElGamal.

Definition 4 (Lossy ElGamal [BHY09]). *The* lossy ElGamal encryption *scheme for message space $M = \{0,1\}$ is given by:*

- $\mathsf{KeyGen}(1^\lambda)$*: Generate the description of a cyclic group $\mathbb{G}$ of prime order q (with $\log_2 q \geq \lambda$) and generators g_0, g_1. Sample a random secret key $s \leftarrow [q-1]$ and compute $h_0 = g_0^s, h_1 = g_1^s$. Output $\mathsf{pk} = (\mathbb{G}, q, g_0, g_1, h_0, h_1)$ and $\mathsf{sk} = s$.*
- $\mathsf{KeyGen}_{\mathsf{loss}}(1^\lambda)$*: Generate the description of $\mathbb{G}$ and g_0, g_1 as above. Sample two random elements $s_0, s_1 \leftarrow [q-1]$, compute $h_0 = g_0^{s_0}, h_1 = g_1^{s_1}$. Output $\mathsf{pk} = (\mathbb{G}, q, g_0, g_1, h_0, h_1)$.*
- $\mathsf{Enc}_{\mathsf{pk}}(m)$*: Sample $r_0, r_1 \leftarrow [q-1]$. Output $(g_0^{r_0} g_1^{r_1}, h_0^{r_0} h_1^{r_1} \cdot g^m)$.*
- $\mathsf{Dec}_{\mathsf{sk}}(c = (a,b))$*: Compute $u := b \cdot a^{-s}$, and output $m \in \{0,1\}$ for which $u = g^m$.*
- $\mathsf{ReRand}_{\mathsf{pk}}(c = (a,b))$*: Choose random $r_0, r_1 \leftarrow [q-1]$, and output $c_{out} = (a \cdot g_0^{r_0} g_1^{r_1}, b \cdot h_0^{r_0} h_1^{r_1})$.*

Theorem 1 ([BHY09]). *Based on the Decisional Diffie-Hellman assumption, the Lossy ElGamal scheme (Definition 4) is a rerandomizable lossy PKE scheme.*

Note that ciphertexts are composed of two group elements, and conversely any pair of elements of $\mathbb{G}$ can be interpreted as a "valid" ciphertext under a given public key pk.

Proving correctness of the new switch gate can be achieved with WI via a similar approach as to standard ElGamal: Here, combining the protocol of Cramer *et al.* for proving OR [CDS94] instead with Okamoto's protocol [Oka93] for proving knowledge of Pedersen commitments (in the place of the Chaum-Pederson protocol [CP92] for proving equality of discrete logarithms). Further details of the resulting 3-round proof are given in the full version (see footnote 1).

3.2 Lossy Abe Mix-Net

Construction 1 (Lossy Abe Mix-Net). *We define the $n = 2^d$-user* lossy Abe mix-net system $\mathsf{MixNet}^{\mathsf{loss}}$ *to be identical to Abe's mix-net, with two exceptions:*

- *All mix-net procedures* $\mathsf{Setup}, \mathsf{SubmitCipher}, \mathsf{VrfblyMix}, \mathsf{Decrypt}$ *make use of the* Lossy ElGamal *algorithms* $\mathsf{KeyGen}, \mathsf{Enc}$, *and* ReRand *(Definition 4), in the place of ElGamal.*
- *Each gate-consistency proof execution* $(\mathcal{P}_{\mathsf{Gate}}, \mathcal{V}_{\mathsf{Gate}})$ *(which was specific to ElGamal) within Abe's* $(\mathcal{P}^{\mathsf{Abe}}_{\mathsf{Mix}}, \mathcal{V}^{\mathsf{Abe}}_{\mathsf{Mix}})$ *is replaced by a corresponding gate-consistency proof execution* $(\mathcal{P}^{\mathsf{loss}}_{\mathsf{Gate}}, \mathcal{V}^{\mathsf{loss}}_{\mathsf{Gate}})$ *for Lossy ElGamal, (this proof is formed as an OR (via Cramer* et al. *[CDS94]) of ANDs of Okamoto [Oka93]).*

Note that while we use Lossy ElGamal for concreteness, a similar approach could be taken using amenable lossy encryption schemes based on, e.g., quadratic residosity, Paillier, or LWE (see e.g., [BHY09,PW11,FGK+13]).

Theorem 2 (Lossy Abe is Anonymous). *The Lossy Abe Mix-Net, as described in Construction 1, is anonymous, as per Definition 3.*

Proof. Let $A = (\mathcal{U}_A, \mathcal{S}_A, \mathcal{A})$ be as in Definition 2, $u_i \in \{0,1\}$ for $i \in \mathcal{U}_{\bar{A}}$ a choice of honest user votes, and σ a permutation over the honest users $\mathcal{U}_{\bar{A}}$. We show that for any PPT interactive machine $\mathcal{V}^*$: $\mathsf{View}_{\mathcal{V}^*}[D^{\mathsf{MixNet},A}_\lambda((u_i)_{i\in\mathcal{U}_{\bar{A}}})] \overset{c}{\approx} \mathsf{View}_{\mathcal{V}^*}[D^{\mathsf{MixNet},A}_\lambda((u_{\sigma(i)})_{i\in\mathcal{U}_{\bar{A}}})]$, where $D^{\mathsf{MixNet},A}_\lambda$ is as in Definition 2), by a sequence of the hybrids which use also the following claim:

Claim (Multiple Witnesses). With overwhelming probability over the choice of a *lossy* key $\mathsf{pk}^{\mathsf{loss}} \leftarrow \mathsf{KeyGen}_{\mathsf{loss}}(1^\lambda)$, the following holds. For any ciphertexts x_0, x_1, y_0, y_1 in the support of $\mathsf{Enc}_{\mathsf{pk}^{\mathsf{loss}}}(\cdot)$, there exists $(\hat{r}_0, \hat{r}_1), (\tilde{r}_0, \tilde{r}_1)$ for which $y_b = \mathsf{ReRand}_{\mathsf{pk}^{\mathsf{loss}}}(x_b; \hat{r}_b)$ *and* $y_b = \mathsf{ReRand}_{\mathsf{pk}^{\mathsf{loss}}}(x_{1-b}; \tilde{r}_{1-b})$ for $b \in \{0,1\}$.

Proof. Follows by the equivalence of distributions $\mathsf{Enc}_{\mathsf{pk}^{\mathsf{loss}}}(m_0) \equiv \mathsf{Enc}_{\mathsf{pk}^{\mathsf{loss}}}(m_1)$ for all messages $m_0, m_1 \in M$ under a lossy key.

Recall the view of $\mathcal{V}^*$ consists of: honest user votes $(u_i)_{i\in\mathcal{U}_{\bar{A}}}$ (chosen by A), the view during the key setup phase $\mathsf{view}^{\mathsf{Setup}}$, the public key pk, secret shares $(\mathsf{sk}_j)_{j\in\mathcal{S}_A}$ of sk, vote ciphertexts of corrupt parties $(c_i^0)_{i\in\mathcal{U}_{\bar{A}}}$, the view of $\mathcal{V}^*$ within the mix phase $\left(\mathsf{view}^{\mathsf{Mix}_j}\right)_{j\in[m]}$, the view of $\mathcal{V}^*$ during the $\mathsf{Decrypt}$ joint decryption $\mathsf{view}^{\mathsf{Dec}}$, and the shuffled plaintext votes $\boldsymbol{v}$.

At a high level, the proof of Theorem 2 moves from

$$\mathsf{View}_{\mathcal{V}^*}[D_\lambda^{\mathsf{MixNet},A}((u_i)_{i\in\mathcal{U}_{\bar{A}}})] \text{ to } \mathsf{View}_{\mathcal{V}^*}[D_\lambda^{\mathsf{MixNet},A}((u_{\sigma(i)})_{i\in\mathcal{U}_{\bar{A}}})]$$

via the following sequence of hybrids. (1) First, $\mathsf{view}^{\mathsf{Setup}}$ and $\mathsf{view}^{\mathsf{Dec}}$ are replaced by simulated views, relying on zero knowledge simulation of the setup and joint decryption protocols. (2) The honest setup functionality is replaced by a modified one which samples a *lossy* system public key and outputs *random* secret key shares sk_j to the corrupt servers. (3) Using semantic security, the encryptions of honest user votes $(u_i)_{i\in\mathcal{U}_{\bar{A}}}$ are replaced by encryptions of the σ-permuted values $(u_{\sigma(i)})_{i\in\mathcal{U}_{\bar{A}}}$ (but the mix and decryption phases are still with respect to $(u_i)_{i\in\mathcal{U}_{\bar{A}}}$). (4) One uncorrupted mix-server modifies his permutation to "undo" the σ shuffle of honest votes. This step relies on the special-soundness property of the mix phase (in order to extract the permutations used by corrupt mix-servers), the WI of the gate-consistency proofs, and the existence of multiple witnesses for any ReRand-switch gate with respect to a lossy public key. (5) The setup procedure is returned to the honest (non-lossy) version. (6) Finally, the simulated $\mathsf{view}^{\mathsf{Setup}}, \mathsf{view}^{\mathsf{Dec}}$ are returned to the honestly generated versions.

4 Abe's Mix-Net with Injected Dummy Votes

We demonstrate that an alternative simple tweak to the Abe mix-net system with comparable efficiency preserves verifiability, and further guarantees anonymity against a malicious verifier. At a high level, our construction is identical to the Abe mix-net (*without* changing the encryption scheme) on $2n$ votes, where n "dummy" ciphertexts of $\perp$ are introduced and removed at the beginning and end of each mix-server mix phase. To verify that this process was followed honestly, the injected ciphertexts will be decrypted at the end along with the shuffled votes (in a carefully chosen order).

Construction 2 (Injected Abe Mix-Net). *The* injected Abe $n = 2^d$-user m-servers mix-net system *is identical to Abe's mix-net with two exceptions: (1)* $\mathsf{VrfblyMix}_{\mathsf{Abe}}^{\mathsf{inject}}(\mathsf{pk}, \boldsymbol{c}^0)$ *is a sequential algorithm with* m *iterations, where each iteration* $j \in [m]$ *is an execution of* $\mathsf{Mix}^{\mathsf{Inject}}$ *as given below (instead of* $\mathsf{Mix}^{\mathsf{Abe}}$*), and (2) the verification proof system* $(\mathcal{P}_{\mathsf{Abe}}^{\mathsf{inject}}, \mathcal{V}_{\mathsf{Abe}}^{\mathsf{inject}})$ *has* 4 *steps as described below (instead of* $(\mathcal{P}_{\mathsf{Abe}}, \mathcal{V}_{\mathsf{Abe}})$*).*

$\mathsf{Mix}^{\mathsf{Inject}}(\mathsf{pk}, \boldsymbol{c}^{j-1})$*: Let* $L = 2d - 1$*. Perform the following:*

1. *(Inject Fake Votes). Generate* n *encryptions of the message* $\perp$*, and insert them into the even positions of a new* $(N = 2n)$*-length vector* $\boldsymbol{C}^{j-1}$*, with the real input ciphertexts in the odd positions. That is, for every* $i \in [n]$*,*

$$C^{j-1}_{2i-1} := c^{j-1}_i, \qquad C^{j-1}_{2i} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\perp).$$

2. *(Choose Permutation). Sample a random permutation* $\pi_j \leftarrow S_n$ *on* n *elements, and let* $\pi^{\mathsf{new}}_j \in S_N$ *be the permutation on* N *elements that acts as* π *on the odd positions and as the identity on the evens. That is:*

$$\forall i \in [n]: \quad \pi^{\mathsf{new}}[2i-1] = 2 \cdot \pi[i] - 1, \qquad \pi^{\mathsf{new}}[2i] = 2i.$$

3. *(*AbeMix *on* $2n$ *Inputs): Execute* AbeMix *with input* $(\mathsf{pk}, \boldsymbol{C}^{j-1}, \pi^{\mathsf{new}}_j)$*. Let* $w^j = (\boldsymbol{C}^j, B_{\pi^{\mathsf{new}}_j}, \hat{R}^j_0, \hat{R}^j_1)$ *be the resulting output.*
4. *(Remove Fake Votes). Output the length-*n *vector* $\boldsymbol{c}^j$ *corresponding to the odd locations of* $\boldsymbol{C}^j$*. That is, output*

$$\forall i \in [n]: \quad c^j_i := C'_{2i-1}.$$

$(\mathcal{P}^{\mathsf{inject}}_{\mathsf{Abe}}, \mathcal{V}^{\mathsf{inject}}_{\mathsf{Abe}})$*: The interactive proof system* $(\mathcal{P}^{\mathsf{inject}}_{\mathsf{Abe}}, \mathcal{V}^{\mathsf{inject}}_{\mathsf{Abe}})$ *with common input* $(\mathsf{pk}, \boldsymbol{c}^0, \boldsymbol{v})$ *and witness* $(\mathsf{rnd}_j, \mathsf{sk}_j)_{j \in [m]}$ *is*

1. ***Submission of intermediate ciphertext vectors:*** *For every* $j \in [m]$*:* $\mathcal{P}$ *generates and sends* $\mathcal{V}$ *the input and output lists of ciphertexts* $(\boldsymbol{C}^{j-1}, \boldsymbol{C}^j)$ *where* $\boldsymbol{C}^{j-1}$ *is generated as in step 1 above, and* $\boldsymbol{C}^j$ *is the list of ciphertexts output from* AbeMix *in step 3 above.* $\mathcal{V}$ *verifies that the output ciphertexts in odd locations for each mix-server* $j-1$ *are identical to the corresponding input ciphertexts to mix-server* j*: i.e., for every* $j \in [m-1], i \in [n]$*:* $C^j_{2i-1} = C^{j+1}_{2i-1}$*. Additionally,* $\mathcal{V}$ *verifies that the first set of ciphertexts in odd locations agree with the submitted vote ciphertexts:* $c^0_i = C^0_{2i-1}$*, for every* $i \in [n]$*.*
2. ***Correctness Proof of*** $\mathsf{VrfblyMix}$*:* *For every* $j \in [m]$*, execute* $(\mathcal{P}^{\mathsf{Abe}}_{\mathsf{Mix}}, \mathcal{V}^{\mathsf{Abe}}_{\mathsf{Mix}})$ *with input* $(\mathsf{pk}, \boldsymbol{C}^{j-1}, \boldsymbol{C}^j)$ *and witness* $(\pi^{\mathsf{new}}, B_{\pi^{\mathsf{new}}_j}, \hat{R}^j_0, \hat{R}^j_1)$*.*
3. ***Correctness Proof of Injected Fake Votes:*** *Let* $\boldsymbol{v}^{\perp} = (\perp, \ldots, \perp)$ *be an* n*-dimension vector of the message* $\perp$*, and* $c^{j-1,\perp}$ *be the vector of* n *ciphertexts such that* $c^{j-1,\perp}_i = C^{j-1}_{2i}$ *for every* $i \in [n]$*. Execute* $(\mathcal{P}^{\mathsf{Abe}}_{\mathsf{Dec}}, \mathcal{V}^{\mathsf{Abe}}_{\mathsf{Dec}})$ *with input* $(\mathsf{pk}, \boldsymbol{c}^{j-1,\perp}, \boldsymbol{v}^{\perp})$*, using witness* $(\mathsf{sk}_j)_j \in [m]$*. If the prover is rejected in this step, the proof system terminates, and no further steps take place.*
4. ***Correctness Proof of*** $\mathsf{Decrypt}$*:* *Let* $\boldsymbol{c}^m$ *be a list of* n *ciphertexts such that* $c^m_i = C^m_{2i-1}$*. Execute* $(\mathcal{P}^{\mathsf{Abe}}_{\mathsf{Dec}}, \mathcal{V}^{\mathsf{Abe}}_{\mathsf{Dec}})$ *with input* $(\mathsf{pk}, \boldsymbol{c}^m, \boldsymbol{v})$ *and witness* $(\mathsf{sk}_j)_j \in [m]$*. If the prover is rejected in this step, or if for any* $i \in [n]$ *it holds that* $v_i = \perp$*, the proof system terminates, and no further steps take place.*

5. ***Correctness Proof of Removed Fake Votes:*** *Let* $\boldsymbol{v}^{\perp} = (\perp, \ldots, \perp)$ *be an* n*-dimension vector of the message* $\perp$*, and* $c^{j,\perp}$ *be the vector of* n *ciphertexts such that* $c_i^{j,\perp} = C_{2i}^{j}$ *for every* $i \in [n]$*. Execute* $(\mathcal{P}_{\mathsf{Dec}}^{\mathsf{Abe}}, \mathcal{V}_{\mathsf{Dec}}^{\mathsf{Abe}})$ *with input* $(\mathsf{pk}, \boldsymbol{c}^{j,\perp}, \boldsymbol{v}^{\perp})$*, using witness* $(\mathsf{sk}_j)_j \in [m]$*. If the prover is rejected in this step, the proof system terminates, and no further steps take place.*

Overall $(\mathcal{P}_{\mathsf{Abe}}^{\mathsf{inject}}, \mathcal{V}_{\mathsf{Abe}}^{\mathsf{inject}})$ proves that: (1) the submitted user ciphertexts are properly copied into the odd positions of the first mix input vector, and for every mix server j the ciphertexts in the odd locations of its input ciphertext vector are the same as those in the output of server $j-1$;[3] (2) every mix server permuted its input vector to its output vector; (3) the injected ciphertexts (in even positions) of each mix server are encryptions of $\perp$; (4) the final ciphertexts in the odd locations indeed decrypt to $\boldsymbol{v}$; and (5) the final ciphertexts in the even locations decrypt to $\perp$. Altogether, this ensures that the final vector $\boldsymbol{v}$ is indeed the permutation of the votes underlying $\boldsymbol{c}^0$. That is, soundness holds.

Theorem 3 (Injected Abe Mix-Net is Anonymous). *The Injected Abe Mix-Net, as described in Construction 2, is anonymous (as per Definition 3).*

The proof uses the following core lemma, focusing on the proof of a single mix-phase. It states that for an honest mix-server who indeed injects ciphertexts of $\perp$ in even positions, then the view of a malicious verifier during the proof of correctness of the corresponding mix-phase is indistinguishable for *any* pair of implemented permutations which fix the even-location positions (but operate arbitrary $\pi_0, \pi \in S_n$ on the odd-location positions).

Lemma 1 (Replacing Permutation in Mix). *For every (adversarial) non-uniform PPT* $\mathcal{A}$*, and every two permutations* $\pi_0, \pi_1 \in S_n$*, the interactive proof system* $(\mathcal{P}_{\mathsf{Mix}}^{\mathsf{Abe}}, \mathcal{V}_{\mathsf{Mix}}^{\mathsf{Abe}})$ *(for correctness of Abe mixing) for the relation* $\mathsf{R}_{\mathsf{Mix}}$ *in Abe Mix-net satisfies distributional witness-indistinguishability (D-WI) with respect to the following two distribution ensembles* $D_\lambda = D_\lambda^{\mathsf{Mix},\mathcal{A}}(\pi_0)$ *and* $D'_\lambda = D_\lambda^{\mathsf{Mix},\mathcal{A}}(\pi_1)$ *where* $D_\lambda^{\mathsf{Mix},\mathcal{A}}$ *is as in Definition 5 described below.*

Definition 5 ($D_\lambda^{\mathsf{Mix},\mathcal{A}}$). *For any (adversarial) non-uniform PPT algorithm* $\mathcal{A}$*, and security parameter* $\lambda \in \mathbb{N}$*, we define the following distribution* $D_\lambda^{\mathsf{Mix},\mathcal{A}}$ *as follows:*
$D_\lambda^{\mathsf{Mix},\mathcal{A}}(\pi)$*:*

Input: *Permutation* $\pi \in S_n$
- *Sample* $(\mathsf{pk}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_m)) \leftarrow \mathsf{Setup}(1^\lambda)$
- *For every* $i \in [n]$*: Obtain* $c_i, z_i \leftarrow \mathcal{A}(\mathsf{pk}, i)$
- *For every* $i \in [n]$*: Set* $C_{2i-1} := c_i$ *and* $C_{2i} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\mathsf{Enc}_{\mathsf{pk}}(\perp)$
- *Let* π^{new} *be such that* $\forall i \in [n]$*:*
  $\pi^{\mathsf{new}}[2i-1] = 2 \cdot \pi[i] - 1, \quad \pi^{\mathsf{new}}[2i] = 2i.$

[3] Note that any pair of group elements can be interpreted as a "valid" ElGamal ciphertext under the public key pk.

– *Execute* AbeMix *(step 2 in Abe's Mix, on 2n votes):*
$(\boldsymbol{C'}, B_{\pi^{\mathsf{new}}}, \hat{R}_0, \hat{R}_1) \leftarrow \mathsf{AbeMix}(\mathsf{pk}, \boldsymbol{C}, \pi^{\mathsf{new}})$

Output: $(X_\lambda = (\mathsf{pk}, \boldsymbol{C}, \boldsymbol{C'}), W_\lambda = (B_{\pi^{\mathsf{new}}}, \hat{R}_0, \hat{R}_1), Z_\lambda = (z_1, \ldots, z_n))$

Proof. We change from the Beneš switch gate settings of $\pi_0^{\mathsf{new}} \in S_{2n}$ to those of π_1^{new} one gate at a time, in a particular order. This is achieved by a sequence of steps of the following two forms: (a) For any honest ciphertext (i.e., encrypting $\perp$), we can change the plaintext, by semantic security. (b) For any gate whose input ciphertexts encrypt the same plaintext (i.e., 2 witnesses to the switch gate), we can flip the switch bit from b to $1 - b$, by WI.

The order of gates is as follows.

We first target the last (output) level of the Beneš network, changing from the corresponding last-level bits of π_0^{new} to those of π_1^{new}. Since the mix-server is honest, in each even output position $2i$ in the last level is the (rerandomized) $\perp$ ciphertext that originated in input position $2i$. Using step type (a) (i.e., semantic security), convert each ciphertext $2i$ to encrypt the same value as its output-gate neighbor $2i-1$. This can be done by rerandomizing the neighbor ciphertext and using this as the original injected "$\perp$" $2i$th ciphertext. Note that changing the plaintext does not affect the permutation, meaning the same pairs of ciphertexts will appear together in the last level gates. Then, given the plaintext switch, we have that every gate in the final level has a pair of ciphertexts of the *same* plaintext. Then using step type (b) (i.e., WI), we may change each gate to agree with the Beneš settings for π_1^{new}.

Next we target the gates in the upper sub-BenešÀgain we will use the power of the honest mix-server controlled dummy "$\perp$" ciphertexts to change from the corresponding permutation bits of π_0^{new} to those of π_1^{new}. First, we "direct" all the $\perp$ ciphertexts up to enter this sub-network by (temporarily) changing the switch settings of the *first* (input) level of the Beneš: Using (a) change all $\perp$ ciphertexts $2i$ to encrypt the same value as their *input*-gate neighbor $2i-1$, using (b) change all first-level gates to switch value 1, so that all ciphertexts entering the upper sub-Beneš are dummy, and then using (a) change them all back to encryptions of $\perp$. At this point, all gates in the upper sub-Beneš satisfy the conditions of step (b) (namely, all ciphertexts encrypt the same plaintext $\perp$), which means they can be changed one by one to agree with the Beneš settings for π_1^{new}.

Finally, the gates in the lower sub-Beneš and in the first-level (input) gates are changed in an analogous fashion.

Given Lemma 1, the proof of anonymity follows essentially the same structure as in the case of the Lossy Abe Mix-net (where previously an analogous statement held by the multiple-witness guarantee of the lossy public key combined with WI). We note that the order of the executions of $(\mathcal{P}_{\mathsf{Dec}}^{\mathsf{Abe}}, \mathcal{V}_{\mathsf{Dec}}^{\mathsf{Abe}})$ (i.e., first the injected even-position ciphertexts, then the final shuffled user votes, then the post-shuffle even-position ciphertexts) is important in order to ensure that we can properly simulate the execution of these executions (i.e., Hybrid 1 in the Lossy Abe proof) without information on users' votes.

References

[Abe99] Abe, M.: Mix-networks on permutation networks. In: Lam, K.-Y., Okamoto, E., Xing, C. (eds.) ASIACRYPT 1999. LNCS, vol. 1716, pp. 258–273. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-540-48000-6_21

[AH01] Abe, M., Hoshino, F.: Remarks on mix-network based on permutation networks. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 317–324. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44586-2_23

[AI06] Abe, M., Imai, H.: Flaws in robust optimistic mix-nets and stronger security notions. IEICE Trans. **89**–**A**(1), 99–105 (2006)

[BG12] Bayer, S., Groth, J.: Efficient zero-knowledge argument for correctness of a shuffle. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 263–280. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_17

[BHY09] Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_1

[CDS94] Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48658-5_19

[Cha81] Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2), 84–88 (1981)

[CP92] Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-48071-4_7

[FGK+13] Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. J. Cryptol. **26**(1), 39–74 (2013)

[FS86] Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12

[FS90] Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, Baltimore, Maryland, USA, 13–17 May 1990, pp. 416–426 (1990)

[GGMM97] Gabber, E., Gibbons, P.B., Matias, Y., Mayer, A.: How to make personalized web browsing simple, secure, and anonymous. In: Hirschfeld, R. (ed.) FC 1997. LNCS, vol. 1318, pp. 17–31. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63594-7_64

[GI08] Groth, J., Ishai, Y.: Sub-linear zero-knowledge argument for correctness of a shuffle. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 379–396. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_22

[GK03] Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS 2003), Cambridge, MA, USA, 11–14 October 2003, pp. 102–113 (2003)

[Gol01] Goldreich, O.: The Foundations of Cryptography - Volume 1, Basic Techniques. Cambridge University Press, Cambridge (2001)

[Gro10] Groth, J.: A verifiable secret shuffle of homomorphic encryptions. J. Cryptol. **23**(4), 546–579 (2010)

[JJ00] Jakobsson, M., Juels, A.: Mix and match: secure function evaluation via ciphertexts. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 162–177. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44448-3_13

[JM98] Jacobson, M., M'Raïhi, D.: Mix-based electronic payments. In: Tavares, S., Meijer, H. (eds.) SAC 1998. LNCS, vol. 1556, pp. 157–173. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48892-8_13

[Nef01] Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: Proceedings of the 8th ACM Conference on Computer and Communications Security, CCS 2001, Philadelphia, Pennsylvania, USA, 6–8 November 2001, pp. 116–125 (2001)

[NSK04] Nguyen, L., Safavi-Naini, R., Kurosawa, K.: Verifiable shuffles: a formal model and a paillier-based efficient construction with provable security. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 61–75. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24852-1_5

[Oka93] Okamoto, T.: On the relationship among cryptographic physical assumptions. In: Ng, K.W., Raghavan, P., Balasubramanian, N.V., Chin, F.Y.L. (eds.) ISAAC 1993. LNCS, vol. 762, pp. 369–378. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57568-5_268

[PVW07] Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. IACR Cryptology ePrint Archive 2007:348 (2007)

[PW11] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. SIAM J. Comput. **40**(6), 1803–1844 (2011)

[TW10] Terelius, B., Wikström, D.: Proofs of restricted shuffles. In: Bernstein, D.J., Lange, T. (eds.) AFRICACRYPT 2010. LNCS, vol. 6055, pp. 100–113. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12678-9_7

[Wik09] Wikström, D.: A commitment-consistent proof of a shuffle. In: Boyd, C., González Nieto, J. (eds.) ACISP 2009. LNCS, vol. 5594, pp. 407–421. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02620-1_28