Collusion Resistant Broadcast and Trace from Positional Witness Encryption

Rishab Goyal[*] Satyanarayana Vusirikala[†] Brent Waters[‡]

Abstract

An emerging trend is for researchers to identify cryptography primitives for which feasibility was first established under obfuscation and then move the realization to a different setting. In this work we explore a new such avenue — to move obfuscation-based cryptography to the assumption of (positional) witness encryption. Our goal is to develop techniques and tools, which we will dub "witness encryption friendly" primitives and use these to develop a methodology for building advanced cryptography from positional witness encryption.

We take a bottom up approach and pursue our general agenda by attacking the specific problem of building collusion-resistant broadcast systems with tracing from positional witness encryption. We achieve a system where the size of ciphertexts, public key and private key are polynomial in the security parameter λ and independent of the number of users N in the broadcast system. Currently, systems with such parameters are only known from indistinguishability obfuscation.

1 Introduction

Over the past five years the introduction of candidate indistinguishability obfuscation schemes [GGH+13b] has produced a dramatic shift in the community's view of which cryptographic primitives are plausibly achievable. Starting with [SW14] there have been several works [SW14, BZ14, GGHR14, BP15, KLW15, CLTV15, HJK+16, GPS16, BPW16] that leverage the power of indistiguishability obfuscation [BGI+01, BGI+12] to give new solutions for problems ranging from deniable encryption to showing the hardness of finding Nash equilibrium.

An emerging trend is for researchers to identify cryptography primitives for which feasibility was first established under obfuscation and then move the realization to a different setting. For example, several works [BP15, KW16, AP16, GKW17b, GKW17a, WZ17] proposed solutions under the Learning with Errors [Reg05] (LWE) assumption of primitives (or impossibility results) that to that point were known only under indistinguishability obfuscation. The motivation for this movement is that LWE is considered a standard assumption with connections to certain problems on lattices, while current indistinguishability obfuscation constructions are based on much newer multilinear map candidates. In a different line of researchers [GPSZ17, LZ17] have shown how to base applications such as realizing trapdoor permutations and the hardness of Nash equilibrium from functional encryption. While subexponentially hard functional encryption is known to imply indistinguishability obfuscation [AJ15, BV15, AJS15], this direction is motivated by building these primitives with only a polynomial loss in the reductions coupled with prospect of functional encryption schemes realized from the polynomial hardness of standard assumptions.

In this work we explore a new such avenue — to move obfuscation-based cryptography to the assumption of (positional) witness encryption [GGSW13, GLW14]. Recall that in a witness encryption scheme, say for SAT, an encryption algorithm takes in a message m along with a boolean formula ϕ that operates an

[*] University of Texas at Austin. Email: rgoyal@cs.utexas.edu.

[†] University of Texas at Austin. Email: satya@cs.utexas.edu.

[‡] University of Texas at Austin. Email: bwaters@cs.utexas.edu. Supported by CNS-1414082, DARPA SafeWare, Microsoft Faculty Fellowship, and Packard Foundation Fellowship.

n bit input w producing a ciphertext ct. A decryptor can recover the message m from ct if it knows a w such that $\phi(w) = 1$. If no such w exists, then the message is computationally hidden. In addition to serving as its own application, witness encryption is known to give rise to primitives such as identity-based encryption [Sha85, BF01] and attribute-based encryption [SW05].

A natural question is why push for moving cryptography from indistinguishability obfuscation to positional witness encryption when current constructions for both rely on multilinear maps [GGH13a, CLT13, GGH15, CLT15]. The justification (like in [GPSZ17, LZ17]) relies on some projection to the future. Since witness encryption is a less powerful primitive than indistinguishability obfuscation, it is believed that the community will likely arrive at a standard assumption solution earlier. This conjecture is supported by some heuristic evidence:

- The work of [GLW14] showing provably secure positional witness encryption from simple multilinear map assumptions came earlier than and was simpler than the later work [GLSW15] which gave a similar result for obfuscation.
- Recently, it was shown [BJK$^+$17] that attribute-based encryption gives rise to a non-trivial form of witness encryption. This might lead to further advances in witness encryption which would not necessarily translate to general obfuscation.
- Recently, the concept of lockable obfuscation [GKW17a, WZ17] was proposed and shown to be realizable under the LWE assumption. Like witness encryption this is a general class of obfuscation, but is more restricted than indistinguishability obfuscation.
- Very recently, Chen et al. [CVW18] gave a new candidate for witness encryption (albeit not positional witness encryption) inspired by [GGH15] multilinear encodings. An important feature of their candidate is that it directly encodes read-once branching program representations of the associated CNF formulae, thereby avoiding attacks such as input-mixing and more. Since read-once branching programs are much less expressive than general branching programs, this also points towards reaching the goal of witness encryption before obfuscation.

In addition, we expect future solutions to witness encryption to be practically more efficient than full blown indistinguishability obfuscation.

Our goal is to develop techniques and tools, which we will dub "witness encryption friendly" primitives[1], and use these to develop a methodology for building advanced cryptography from positional witness encryption. While we don't expect to move all or even "most" of obfuscation-based cryptography to positional witness encryption, we believe that a long term effort could yield a number of applications which are comparable to those achieved from the aforementioned efforts on building from functional encryption [GPSZ17, LZ17] or lockable obfuscation [GKW17a, WZ17].

We will take a bottom-up approach and pursue our general agenda by attacking specific problems that are not known from witness encryption. To that end in this work we study building collusion-resistant broadcast systems with tracing from positional witness encryption. Our goal is to achieve where the size of ciphertexts, public key and private key are polynomial in the security parameter λ and independent of the number of users N in the broadcast system.[2] Below we provide an overview of prior work, present our new results, toolkit of "witness encryption friendly" primitives, and the techniques that allow us to achieve the above goals.

1.1 Overview

Broadcast Encryption with Tracing. Broadcast Encryption was introduced by Fiat and Naor [FN94]. A broadcast encryption scheme, like a standard public key encryption scheme, consists of three algorithms

[1] This is intended to mirror the term "iO friendly" used elsewhere in the literature.

[2] Following prior broadcast encryption literature we will not count a description S of the recipients of a ciphertext toward the ciphertext overhead.

— setup, encryption and decryption. The setup algorithm outputs a public key and N secret keys, where N represents the number of users given as an input. Using the encryption algorithm, a sender can encrypt a message such that the corresponding ciphertext can only be decrypted by the "qualified" users $S \subseteq [N]$.[3] Here the set S is given as input to the encryption algorithm. The decryption algorithm is self-explanatory. For security it is required that no set of colluding users can decrypt a ciphertext if none of them are *qualified*.

Suppose that a set of users S_1 collude to create a decoding box D which is capable of decrypting ciphertexts intended for some (possibly different) set of users S_2 with some non-negligible probability. A broadcast system which provides tracing capabilities allows extraction of a non-empty set T (from the box D) such that $T \subseteq S_1$, i.e. contains at least one colluding user but none outside of it. Such broadcast systems are referred to as Trace and Revoke systems in the folklore [NP00, NNL01]. However, we chose to refer to it as Broadcast and Trace system as it is more appropriate. They have an additional tracing algorithm which given only the oracle access to the box D can perform this traitor extraction.

Broadcast and Trace via Augmented Broadcast Encryption (AugBE). Boneh and Waters (BW) [BW06a] built the first fully collusion resistant Broadcast and Trace scheme with sub-linear (in N) ciphertext size. They also provided a framework for building Broadcast and Trace schemes by introducing an intermediate primitive called augmented broadcast encryption (AugBE). We follow the same approach in this work and therefore we elaborate on it now.

An AugBE scheme, as the name suggests, is a broadcast encryption scheme with an augmented encryption functionality. Similar to a standard broadcast encryption scheme it consists of setup, encryption and decryption algorithms. In an AugBE system, the encryption algorithm also receives a "cutoff" index $i \in [N+1]$, in addition to a set $S \subseteq [N]$, as an input. This cutoff index affects the decryptability of the ciphertext in such a way that the resultant ciphertext can only be decrypted by the users $S' = S \setminus [i-1]$, i.e. users whose indices are as large as i and belong to the set S are now labelled as *qualified*. BW defined two security properties for an AugBE system — index hiding and message hiding security. The first security property (index hiding) states that an encryption of m under set S to index i is indistinguishable from an encryption of m under set S to index $i+1$, if either $i \notin S$ (even when the adversary has all the secret keys), or the adversary does not have the i^{th} key. The second property (message hiding) states that an encryption of m_0 under set S to index $N+1$ is indistinguishable from an encryption of m_1 under set S to index $N+1$, even when the adversary is given all N secret keys.

BW argued that if an AugBE scheme satisfies these two properties, then that is sufficient for constructing a Broadcast and Trace (BT) scheme. In their transformation, the BT setup and decryption algorithm are identical to their AugBE counterparts. For encryption, a sender runs the AugBE encryption algorithm with the cutoff index value set to be 1. The tracing algorithm runs AugBE encryption varying the value of cutoff index. Given a decoder box D and target set S, the tracing algorithm encrypts random messages under set S to every index $i = 1$ to $N+1$, and estimates (for each index i) the probability D decrypts correctly. Suppose the probability decoder D is successful, i.e. decrypts standard (index 1) ciphertexts correctly, is at least ϵ. By message hiding property, we know that D can not have non-negligible success probability when run on ciphertexts encrypting to index $N+1$. This implies that there must exist an index $i^* \in [N]$ such that the decoder's success probability in decrypting index i^* ciphertexts is at least $\approx \epsilon/N$ more than in decrypting index i^*+1 ciphertexts. Every cutoff index i where there is a gap in the estimated success probabilities for index i and $i+1$, the tracing algorithm adds that user i to the set of traitors. The main idea here is that if an index $i \notin S$ or the adversary does not have the key for user i, then by index hiding security it should not be able to distinguish between index i and $i+1$ ciphertexts.

Although the above transformation seems to work (at least intuitively), we would like to point out that the proof provided in [BW06a] was inaccurate. Very briefly, the problem lies in the fact that there is a "semantic gap" between the definitions of BT and AugBE schemes. The issue is that in a BT system an adversary outputs a box which performs some decoding/decryption operations, whereas in an AugBE system the adversary plays a distinguishing game. At first, it seems like one could use the decoder box to decrypt the ciphertext and use its output for distinguishing. The problem is that decoder might work incorrectly

[3] Here *qualified* could alternatively be interpreted as "non-revoked".

sometime and it would affect the success probability of the reduction algorithm. Similar issues were observed by Goyal, Koppula and Waters [GKW18] in the context of (non-broadcast) traitor tracing. They resolved the issue by upgrading the security requirements from the underlying intermediate primitives to match the decoder-based security notions required for traitor tracing. In this work we fix the proof of security for the BW transformation showing that it does lead to a secure BT scheme.[4] More details are provided later in Section 3.

Our Results and Prior Work. Our main result are new collusion-resistant Broadcast and Trace schemes from positional witness encryption where the size of ciphertexts, public key and private key are polynomial in the security parameter λ and independent of the number of users N[5]. Currently, systems with such parameters are only known from indistinguishability obfuscation [NWZ16]. If we drop the tracing requirement, that is consider only broadcast encryption, there are constructions based on multilinear maps [BWZ14] and iO [BZ14]. If we drop the revocation requirement, that is consider only traitor tracing, schemes with such parameters are known based on iO [BZ14]. In bilinear groups we can achieve short ciphertexts [BGW05, GW09], but with longer keys if we drop the tracing requirement. Additionally, we have solutions [BW06a] with ciphertexts that grow proportionally to $\sqrt{N}$ if we keep it. Very recently, Goyal, Koppula and Waters [GKW18] gave a polylog traitor tracing scheme from the LWE assumption. However, their system does not have the capability to broadcast to arbitrary sets.

We further develop a toolkit of certain simpler primitives such that these could be used in conjunction with positional witness encryption in similar vein to how we have *iO friendly* primitives to support applications of iO. Our BT scheme is secure assuming the existence of positional witness encryption and these simpler primitives. We provide numerous instantiations of these primitives from a wide variety of standard assumptions such as LWE, RSA and decision linear over bilinear groups. Now we describe our techniques and main ideas to build a Broadcast and Trace system.

Building Augmented Broadcast Encryption from Positional Witness Encryption. The main building block used in our construction is a positional witness encryption (PWE) scheme. In a PWE scheme, the encryption algorithm also takes as input a *cutoff* index $i \in \{0, \ldots, 2^n\}$ where n is the bit length of witnesses on which the corresponding boolean formula (witness relation) ϕ operates. A decryptor can recover the message m from ct if it knows a w such that $\phi(w) = 1$ and $w \geq i$.[6] For security it has two properties — message hiding and index hiding. First, message hiding states that a message encrypted for index 2^n (i.e., the last index) is hidden irrespective of the boolean formula used. Second, index hiding states that an encryption of m under formula ϕ for index i is indistinguishable from an encryption of m under ϕ to index $i+1$, if $\phi(i) = 0$.

We now provide an outline of our AugBE construction. Let us start with a simple idea. Suppose during setup, the algorithm samples a key pair for a standard signature scheme. Next, the secret key for i^{th} user consists of a signature σ_i on message i and the public key simply corresponds to the verification key vk. To encrypt a message m under set S and index i, the encryptor runs the PWE encryption algorithm on message m for index $i \,||\, 0^\ell$ and formula $\phi_{\mathsf{vk},S}$, where $\phi_{\mathsf{vk},S}(j, \sigma) = 1$ iff '$j \in S$' and 'σ is a valid signature on j under vk'. Here ℓ denotes the length of the signatures. For decryption a user simply runs the PWE decryption with its index and signature as the witness. Correctness of this scheme follows directly. However, this scheme is clearly not compact since the set S is embedded in the formula $\phi_{\mathsf{vk},S}$ and since the size of PWE ciphertexts could arbitrarily (but polynomially) depend on the size of the formula, thus the overall AugBE scheme could be highly inefficient. In a few words the problem is that we are implementing a trivial set membership check which breaks compactness.

To get around this problem we will use an alternate set membership check. Our idea is to embed only a succinct commitment to the set S in the formula ϕ such that there exists proofs of membership in S that grow at most logarithmically with the number of users N. Clearly such a primitive would resolve the inefficiency

[4] Here we only consider BT schemes with public traceability.

[5] Here we assume that number of users N is at most $\mathsf{poly}(\lambda)$

[6] Here comparisons between bit-strings is performed by interpreting each bit-string as non-negative integer.

problem. One possible execution of this idea is via a Merkle hash tree.[7] Let $\mathbb{I}_S$ represent the N-bit indicator string corresponding to set S, i.e. i^{th} bit of $\mathbb{I}_S$ is 1 iff $i \in S$. We modify the encryption procedure as follows — first compute a hash h of string $\mathbb{I}_S$; next run the PWE encryption algorithm on message m for index $i \,||\, 0^\ell \,||\, 0^k$ and formula $\phi_{\mathsf{vk},h,N}$, where $\phi_{\mathsf{vk},h,N}(j, \sigma, \pi) = 1$ iff '$j \leq N$', 'π is a valid proof membership for index j w.r.t. hash h' and 'σ is a valid signature on j under vk'. Here proof π simply corresponds to the pre-images in the hash tree along the path from the root h to the leaf node containing the j^{th} bit, and k denotes the length of proof π. The decryption is then performed analogously where the decryptor computes the membership proof by hashing $\mathbb{I}_S$ and using the appropriate leaf-to-root path as a proof. This seems to resolve the succinctness problem as the size of the ciphertext is independent of the number of users. Also, at least intuitively, it seems that the scheme should satisfy both index hiding and message hiding security properties. The intuition is that since $\phi_{\mathsf{vk},h,N}$ is not satisfied by any witness larger than $(N+1) \,||\, 0^\ell \,||\, 0^k$, by using security of PWE we can argue message hiding security for the above scheme.[8] For arguing index hiding security we would hope to use the fact that if $i \notin S$, or if the adversary does not receive the key for i^{th} user, then the adversary does not know of any witnesses of the form $i \,||\, \{0,1\}^\ell \,||\, \{0,1\}^k$ and thus we could use PWE index hiding security. In the first case (i.e., $i \notin S$) hardness of computing witnesses should follow from collision resistance of the hash function, and in the second scenario it should follow from unforgeability of the signature scheme. However, there is a problem here. Although we could argue that witnesses are hard-to-compute while proving index hiding for AugBE, this won't be sufficient overall as for applying PWE index hiding security as it is necessary that there does *not* exist *any* witness of the form $i \,||\, \{0,1\}^\ell \,||\, \{0,1\}^k$. Thus, unless the underlying PWE scheme provides some strong notion of extractable security, it is not clear how to prove security of the above construction.[9]

To this end, we develop a toolkit of certain simpler primitives, which aid us in proving our construction to be secure. Our motivation here is that using such primitives, we could somehow indistinguishably switch between instances/formulae which have hard-to-compute witnesses to instances/formulae which do not have any witnesses (in some particular pre-specified range). Thus this would enable applicability of the index hiding security property of PWE scheme in the corresponding proof. Below we elaborate on two such primitives — all-but-one signatures and somewhere perfectly binding hash functions (a primitive similar to somewhere statistically binding hash functions described in [HW15, OPWW15]).[10]

A Toolkit for Witness Encryption. The first primitive we consider is a special type of signature scheme called all-but-one (ABO) signatures. These are just like standard signatures, except the setup algorithm has a special "punctured" mode in which it takes a message m^* as an additional input and outputs a pair of signing and verification key $(\mathsf{sk}, \mathsf{vk})$ such that there does not exist any signature that gets verified for message m^*. In other words, the verification algorithm on inputs vk and m^* rejects *every* signature σ. Now instead of unforgeability-type security, we only require that an adversary should not be able to distinguish verification keys that are output by *punctured* setup with message m^* from those output by normal setup, even when given access to the signing oracle.[11] We note that the notion of ABO signatures is motivated by constrained signatures [BZ14] and splittable signatures [KLW15], but is much weaker than both of those. In this work, we also provide new constructions of ABO signatures from a wide variety of standard assumptions. Next we

[7] The idea of using Merkle hash tree for efficiently committing to large sets has also been previously used in works such as [ABG+13, Zha16].

[8] The proof will involve an exponential number of hybrids. This is because for applying message hiding security property of PWE the index used must be $2^{\lambda+\ell+k}$ (i.e., the last index), therefore we need to use index hiding security to go from index $(N+1) \,||\, 0^\ell \,||\, 0^k$ to $2^{\lambda+\ell+k}$ which takes an exponential number of hybrid steps. Here the exact ordering of witness components, i.e. i, σ, π, is very important for the proof to go through. We can only use the security of PWE scheme if index i is leading term and corresponds to the most significant bits.

[9] Although the notion of witness encryption with extractable security has been well studied [GKP+13, GGHW14], extractability in the case of positional witness encryption is rather non-trivial to define due to the fact that PWE already requires index hiding to hold for all indices.

[10] We would like to point out that our techniques of relaxing extractably-secure assumptions to more standard indistinguishability-based assumptions are in part inspired by analogous results in the regime of moving from differing-inputs obfuscation (diO) to indistinguishability obfuscation (iO) [HW15, NWZ16, CDG+17].

[11] The adversary is not allowed to query the oracle on message m^* to allow trivial distinguishing attacks.

discuss the second primitive we use, and later we will circle back to the new ABO signature constructions we provide.

The next primitive we employ is a somewhere perfectly binding (SPB) hash function [HW15, OPWW15]. An SPB hash consists of four algorithms — setup, hash, open and verify. The setup algorithm is used to sample a hash key hk, and has two modes (akin to ABO signatures) — normal and "binding". In the *binding* mode it takes an index i as an additional input, and it ensures that the corresponding hash function H_{hk} is perfectly binding for the i^{th} message position (i.e., the hash value completely determines the i^{th} bit of the pre-image). Additionally, SPB hashes have a local opening property which states that for any message m, any index $i \leq |m|$ and hash $h = H_{\mathsf{hk}}(m)$, one could create a *short* proof π proving that the message's i^{th} bit is $m[i]$ and it hashes to h.[12] Such proofs could be verified by running the verification algorithm which also take as input the hash key, hash value and a position. For security it is required that an adversary should not be able to distinguish between hash keys that are output by *binding* setup and those output by normal setup.

Next we show that if we use ABO signatures and SPB hash functions in the previously described AugBE construction then we can prove its security using positional witness encryption.

Completing AugBE Construction. As discussed earlier, ABO signature scheme and an SPB hash function enable us to indistinguishably turn instances with hard-to-compute witnesses into instances which have no witnesses (in a particular range). Therefore, by simply using an ABO signature scheme and an SPB hash function in our AugBE construction, we can also prove index hiding property of our construction. The construction is identical to the one described before, except that checking membership of index j will now be done by SPB verification algorithm as follows — 'π proves that there exists a string x such that $x[j] = 1$ and $H_{\mathsf{hk}}(x) = h$'. The proof of AugBE message hiding stays the same as $\phi_{\mathsf{vk},\mathsf{hk},h,N}$ is not satisfied by any witness larger than $(N+1) \,||\, 0^{\ell} \,||\, 0^{k}$. The AugBE index hiding proof is divided in two parts. Let i be the challenge index, S the challenge set and $S_{\mathcal{A}}$ the set of keys in adversary's possession. We know that either $i \notin S$ or $i \notin S_{\mathcal{A}}$. Consider the following cases.

- $i \notin S_{\mathcal{A}}$: The idea here is that since the adversary does not have key for user i, thus we could instead generate the $(\mathsf{sk}, \mathsf{vk})$ key pair by running *punctured* setup for message i. From adversary's perspective this can not be distinguished with non-negligible probability by ABO security. And now, since the verification key vk no longer accepts any signature σ for message i, we get $\phi_{\mathsf{vk},\mathsf{hk},h,N}(w) = 0$ for all $i \,||\, 0^{\ell} \,||\, 0^{k} \leq w < (i+1) \,||\, 0^{\ell} \,||\, 0^{k}$. As a result, we could use PWE index hiding security to switch from index i AugBE ciphertexts to index $i+1$ ciphertexts. Finally, we could *un-puncture* the key vk to complete the proof.

- $i \notin S$: The proof is very similar to the one described above. The only modification will be that instead of puncturing the verification key at index i, we *bind* the hash key for position i. The intuition is that since the i^{th} bit of string $\mathbb{I}_S$ is zero (as $i \notin S$), thus if the hash key hk was (perfectly) binding at position i then there will not exist any proof π that proves that there exists a string x such that $H_{\mathsf{hk}}(\mathbb{I}_S) = H_{\mathsf{hk}}(x)$ the i^{th} bit of x is 1. Thus, as before $\phi_{\mathsf{vk},\mathsf{hk},h,N}(w) = 0$ for all indices in that range and we can apply PWE index hiding security.

At a high level, the proposed paradigm is to first use the developed toolkit to turn formulae with *hard-to-compute* satisfying inputs into formulae with only *range-restricted* satisfying inputs, then use PWE security to cut through the range of *inactive* inputs, and finally switch back to original formulae using our toolkit. We believe that such a methodology will find more applications especially in bringing more primitives based on obfuscation to the assumption of (positional) witness encryption. Finally, we talk about the new ABO signature constructions that we provide.

[12] Technically one could visualize the proof π as only proving that the i^{th} bit of pre-image is $m[i]$. The fact that it also proves that the message hashes to $H_{\mathsf{hk}}(m)$ is just due to the structure of the proof.

ABO Signatures from Standard Assumptions. In this work we give two new pathways to build ABO signatures. First, we show that an ABO signature scheme can be generically built from any verifiable random function (VRF) [MRV99] and a perfectly-binding (non-interactive) commitment scheme. Second, we show that any identity-based encryption (IBE) scheme [Sha85, BF01], that is anonymous [BBDP01] as well as allows efficient key verifiability, also leads to an ABO signature scheme. VRFs can be based on a wide variety of assumptions such as decision-linear over bilinear maps as well as RSA-like assumptions [MRV99, HJ16] and perfectly-binding (non-interactive) commitment schemes can be based on assumptions such as DDH, LWE and LPN [GHKW17] and perfectly injective OWFs. IBE schemes with such verifiability and anonymity properties can be based on simple assumptions over bilinear maps as well as LWE [BW06b, SKOS09, ABB10, LSJ+11, GKW17a, WZ17]. Thus this leads to new constructions of ABO signatures. We also point out that ABO signatures can be built from constrained signatures [BZ14] and splittable signatures [KLW15] which have been constructed under iO and OWFs. Constrained signatures have also been constructed from non-interactive witness indistinguishable proofs and perfectly binding commitments [BZ14].

We now briefly highlight the main ideas to build these from VRFs. A VRF is like a pseudorandom function (PRF) in which the secret key holder can also prove correctness and uniqueness of PRF evaluation. Concretely, using the secret key sk, it could efficiently evaluate the function $F_{\mathsf{sk}}(\cdot)$ on any input x as well as generate a proof π of the statement $y = F_{\mathsf{sk}}(x)$. An ABO signing key will simply correspond to the VRF secret key sk, and the ABO verification key will contain the VRF verification key vk as well as a commitment COM. Here COM commits to 0 during standard setup, whereas during punctured setup (with message x^*) COM commits to 1 where the random coins used are $F_{\mathsf{sk}}(x^*)$. A signature σ for any message x will simply correspond to its function evaluation $y = F_{\mathsf{sk}}(x)$ as well as corresponding proof π. While verifying a message-signature pair $x, (y, \pi)$ w.r.t. key $(\mathsf{vk}, \mathsf{COM})$, the verifier checks two things — (1) π proves that y is a correct evaluation on input x, and (2) COM does not match the commitment of bit 1 obtained using y as randomness. Clearly this scheme satisfies the ABO scheme correctness condition if the underlying commitment scheme is perfectly binding as in case of normal setup, condition (2) will never be satisfied. Both our ABO constructions are provided later in Section 5.

Lastly, one might think that the full power of ABO signatures is not needed to build the above Broadcast and Trace system. Instead a restricted version where the message space is fixed to be $\{1, 2, \ldots, N\}$ might suffice. It turns out that such a restricted ABO signature scheme can be directly constructed from any SPB Hash function and length doubling pseudo-random generator (PRG). The idea is to sample an SPB hash key hk, random λ bit strings s_i for each message $i \in [N]$ during setup. The verification key consists of the hash key hk and a hash value h, where h is computed as the SPB hash on the set $\{t_i = \mathsf{PRG}(s_i)\}_i$. The signature on message i consists of (s_i, π_i) where π_i is the SPB hash opening of hash h on index i. The verification procedure first checks correctness of the hash proof π_i, and then also checks that $\mathsf{PRG}(s_i)$ is i^{th} block value. For punctured setup at index i^*, the algorithm changes the following — 1) it samples SPB hash hk to be binding at index i^*, 2) it samples t_{i^*} uniformly at random from $\{0,1\}^{2\lambda}$. With all-but-negligible probability, t_{i^*} will lie outside the range space of PRG, therefore no valid signature for i^* would exist under punctured setup.

However, such an ABO scheme can only be used to build a Broadcast and Trace system in which the numbers of users is a-priori (and polynomially) bounded. A more desirable setting would be where the number of users that can be supported is exponential (i.e., unbounded), while allowing the encryptor to choose any polynomial sized (a-priori unbounded) subset of users to broadcast to. Such a Broadcast and Trace system would still require the full power of ABO signatures, thus we stick to the more general setting.

2 Preliminaries

Notations For a probability distribution D, we denote by $x \leftarrow D$ that x is sampled according to D. If S is a set, $y \leftarrow S$ denotes that y is sampled from S according to the uniform distribution on S. We use $[m, n]$ to denote the set of contiguous integers $\{m, \ldots, n\}$ for some $m, n \in \mathbb{Z}$. For simplicity, we simply use $[n]$ to denote the set $[1, n] = \{1, \ldots, n\}$ for any $n \geq 1$. We sometimes slightly abuse notation and refer to bit strings in $\{0,1\}^\ell$ by integers, where the left most bit of $x \in \{0,1\}^\ell$ is considered as the most significant bit. For any set S, we denote the size of the set of $|S|$. We denote security parameter by λ in the rest of the paper. For any bit string t, we denote that $\mathsf{int}(t)$ as the integer representation of string t.

2.1 Positional Witness Encryption

In this section, we formally define Positional Witness Encryption (PWE) [GLW14] and list its correctness and security properties. The encryption system is defined for an NP language L and a message space $\{\mathcal{M}_\lambda\}_\lambda$. Let $R(\cdot,\cdot)$ be the witness relation corresponding to L i.e., for any string $x \in \{0,1\}^*$, $x \in L$ iff $\exists w \in \{0,1\}^{n(|x|)}$ s.t. $R(x,w) = 1$, where $n(|x|)$ is the witness length of instance x. For simplicty of notation, we hereby denote $n = n(|x|)$. A party can encrypt a message m with an instance x and index ind. Another party can decrypt the ciphertext using a witness w to the instance x such that $R(x,w) = 1$ and $w \geq \mathsf{ind}$. Given a string $w \in \{0,1\}^n$, we sometimes slightly abuse notation and also refer to w as an integer. Formally, the encryption system contains two procedures defined as follows.

- $\mathsf{Encrypt}(1^\lambda, x, m, \mathsf{ind}) \to \mathsf{ct}$. The encryption algorithm takes as input a security parameter 1^λ, an instance $x \in \{0,1\}^*$, a message m, an index $\mathsf{ind} \in [0, 2^n]$ and outputs a ciphertext ct.
- $\mathsf{Decrypt}(w, \mathsf{ct}) \to m$. The decryption algorithm takes as input a witness $w \in [0, 2^n - 1]$, a ciphertext ct and outputs either a message m or $\perp$.

Correctness. We say that a PWE scheme is correct if for every $\lambda \in \mathbb{N}$, any instance $x \in \{0,1\}^*$, any message $m \in \mathcal{M}_\lambda$, any witness $w \in [0, 2^n - 1]$, any position index $\mathsf{ind} \in [0, 2^n]$ such that $R(x,w) = 1$ and $w \geq \mathsf{ind}$, and $\mathsf{ct} \leftarrow \mathsf{Encrypt}(1^\lambda, x, m, \mathsf{ind})$, we have

$$\Pr\Big[\mathsf{Decrypt}(w, \mathsf{ct}) = m\Big] = 1$$

Security. A positional witness encryption scheme should satisfy 2 security properties: *message indistinguishability* and *position indistinguishability* defined as follows.

Definition 2.1 (Message Indistinguishability)**.** A PWE scheme for a language L is message indistinguishability secure if for any stateful PPT adversary $\mathcal{A}$, there exists a negligible function negl$(\cdot)$ such that for every $\lambda \in \mathbb{N}$, we have

$$\Pr\left[\mathcal{A}(\mathsf{ct}) = b \ : \ \begin{array}{c} (x, m_0, m_1) \leftarrow \mathcal{A}(1^\lambda); \\ b \leftarrow \{0,1\}; \mathsf{ct} \leftarrow \mathsf{Encrypt}(1^\lambda, x, m_b, 2^n) \end{array}\right] \leq \frac{1}{2} + \mathrm{negl}(\lambda).$$

Note that the above property needs to be satisfied even for instances $x \in L$.

Definition 2.2 (Position Indistinguishability)**.** A PWE scheme for a language L with witness relation $R(\cdot,\cdot)$ is position indistinguishability secure if for every stateful PPT adversary $\mathcal{A}$, there exists a negligible function negl$(\cdot)$ such that for every $\lambda \in \mathbb{N}$, we have

$$\Pr\left[\mathcal{A}(\mathsf{ct}) = b \ : \ \begin{array}{c} (x, m, \mathsf{ind}) \leftarrow \mathcal{A}(1^\lambda); \\ b \leftarrow \{0,1\}; \mathsf{ct} \leftarrow \mathsf{Encrypt}(1^\lambda, x, m, \mathsf{ind} + b) \end{array}\right] \leq \frac{1}{2} + \mathrm{negl}(\lambda).$$

where the adversary $\mathcal{A}$ is restricted to produce a challenge (x, m, ind) such that $R(x, \mathsf{ind}) = 0$.

2.2 All-But-One Signatures

In this section, we define all-but-one (ABO) signatures and describe its security properties. An ABO signature scheme is similar to a public key signature scheme, except that it offers an additional setup algorithm which outputs a verification key punctured at input message. We note that the notion is motivated by and is weaker than splittable signatures [KLW15] and constrained signatures [BZ14]. Both the primitives offer an additional split algorithm that can generate constrained secret key and constrained verification key. Formally, we define an ABO scheme with respect to message space $\{0,1\}^{n(\lambda)}$ and signature space $\{0,1\}^{\ell(\lambda)}$ for some polynomials $n(\cdot)$ and $\ell(\cdot)$ as follows.

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{sk}, \mathsf{vk})$. The setup algorithm takes as input a security parameter λ and outputs a signing key sk and a verification key vk.
- $\mathsf{Setup\text{-}Punc}(1^\lambda, m^*) \to (\mathsf{sk}, \mathsf{vk})$. The punctured setup algorithm takes as input a security parameter λ and a message m^*. It outputs a signing key sk and a verification key vk.
- $\mathsf{Sign}(\mathsf{sk}, m) \to \sigma$. The signing algorithm takes as input a signing key sk, a message m and outputs a signature σ.
- $\mathsf{Verify}(\mathsf{vk}, m, \sigma) \to 0/1$. The verification algorithm is a *deterministic* algorithm that takes as input a (possibly punctured) verification key vk, a message m and a signature σ. It outputs either 0 or 1.

Correctness. We say that an ABO signature scheme is correct if

1. Correctness of Setup: For any security parameter $\lambda \in \mathbb{N}$, $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup}(1^\lambda)$, any $m \in \{0,1\}^{n(\lambda)}$, $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$, we have $\mathsf{Verify}(\mathsf{vk}, m, \sigma) = 1$.
2. Correctness of Punctured Setup: For any security parameter $\lambda \in \mathbb{N}$, any message $m^* \in \{0,1\}^{n(\lambda)}$, $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup\text{-}Punc}(1^\lambda, m^*)$ and any $\sigma \in \{0,1\}^{\ell(\lambda)}$, we have $\mathsf{Verify}(\mathsf{vk}, m^*, \sigma) = 0$.

Security. We now define the required security property for all-but-one signatures.

Definition 2.3 (VK Indistinguishability)**.** An all-but-one signature scheme $\mathrm{S} = (\mathsf{Setup}, \mathsf{Setup\text{-}Punc}, \mathsf{Sign}, \mathsf{Verify})$ is said to be VK indistinguishable if for any stateful PPT adversary $\mathcal{A}$, there exists a negligible function $\mathrm{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$, the following holds.

$$\Pr\left[\mathcal{A}^{\mathsf{Sign}(\mathsf{sk}_b,\cdot)}(\mathsf{vk}_b) = b \ : \ \begin{array}{c} m^* \leftarrow \mathcal{A}(1^\lambda); (\mathsf{sk}_0, \mathsf{vk}_0) \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathsf{sk}_1, \mathsf{vk}_1) \leftarrow \mathsf{Setup\text{-}Punc}(1^\lambda, m^*); b \leftarrow \{0,1\} \end{array}\right] \leq \frac{1}{2} + \mathrm{negl}(\lambda).$$

where the adversary is not allowed to make signature query on message m^*.

Note that by a hybrid argument we can prove that if an all-but-one signature scheme satisfies VK Indistinguishability property, it also satisfies the standard security notion of selective unforgeability.

2.3 Somewhere Perfectly Binding Hash Function

In this section, we define somewhere perfectly binding hash function, which is similar to somewhere statistically binding hash function defined in [HW15, OPWW15] and describe its correctness and security properties.

- $\mathsf{Setup}(1^\lambda, L) \to \mathsf{hk}$. The setup algorithm takes as input a security parameters λ, a message length $L < 2^\lambda$ and outputs a public hashing key hk.
- $\mathsf{Setup\text{-}Bind}(1^\lambda, L, \mathsf{ind}) \to \mathsf{hk}$. The binding setup algorithm takes as input a security parameters λ, message length $L < 2^\lambda$, an index $\mathsf{ind} \leq L$ and outputs a public hashing key hk.
- $\mathsf{Hash}(\mathsf{hk}, m) \to h$. The hash function takes as input a hash key hk, a message $m \in \{0,1\}^L$ and outputs a hash h.
- $\mathsf{Open}(\mathsf{hk}, m, \mathsf{ind}) \to \pi$. This opening algorithm takes as input a hash key hk, a message $m \in \{0,1\}^L$, an index $\mathsf{ind} \leq L$ and outputs a proof π.
- $\mathsf{Verify}(\mathsf{hk}, h, \mathsf{ind}, b, \pi) \to 0/1$. The verification algorithm is a *deterministic* algorithm takes as input a hash key hk, a hash value h, an index $\mathsf{ind} \leq L$, a bit b, a proof π and outputs 0 or 1.

Correctness of Opening. We say that an SPB hash scheme is correct if the following conditions hold.

- For every security parameter $\lambda \in \mathbb{N}$, any message length $L < 2^\lambda$, any message $m \in \{0,1\}^L$, every index $\mathsf{ind} \leq L$, $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, L)$, $h = \mathsf{Hash}(\mathsf{hk}, m)$, $\pi \leftarrow \mathsf{Open}(\mathsf{hk}, m, \mathsf{ind})$, we have
$$\mathsf{Verify}(\mathsf{hk}, h, \mathsf{ind}, m[\mathsf{ind}], \pi) = 1$$
- For every security parameter $\lambda \in \mathbb{N}$, any message length $L < 2^\lambda$, any message $m \in \{0,1\}^L$, every pair of indices $\mathsf{ind}, \mathsf{ind}' \leq L$, $\mathsf{hk} \leftarrow \mathsf{Setup\text{-}Bind}(1^\lambda, L, \mathsf{ind}')$, $h = \mathsf{Hash}(\mathsf{hk}, m)$, $\pi \leftarrow \mathsf{Open}(\mathsf{hk}, m, \mathsf{ind})$, we have
$$\mathsf{Verify}(\mathsf{hk}, h, \mathsf{ind}, m[\mathsf{ind}], \pi) = 1$$

Security. An SPB Hash function need to satisfy 2 security properties - Index Hiding and Somewhere Perfectly Binding w.r.t Opening.

Definition 2.4 (Index Hiding)**.** An SPB Hash function is said to have index hiding property if for any stateful PPT adversary A, there exists a negligible function $\text{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$, the following holds.
$$\Pr\left[\mathcal{A}(\mathsf{hk}_b) = b \ : \ \begin{array}{c} (L, \mathsf{ind}) \leftarrow \mathcal{A}(1^\lambda); \mathsf{hk}_0 \leftarrow \mathsf{Setup}(1^\lambda, L) \\ \mathsf{hk}_1 \leftarrow \mathsf{Setup\text{-}Bind}(1^\lambda, L, \mathsf{ind}); b \leftarrow \{0,1\} \end{array}\right] \leq \frac{1}{2} + \text{negl}(\lambda).$$

Definition 2.5 (Somewhere Perfectly Binding w.r.t. Opening)**.** We say that a hash key hk is binding w.r.t. opening at index ind if for every hash h, there does not exist proofs π and π' such that
$$\mathsf{Verify}(\mathsf{hk}, h, \mathsf{ind}, 0, \pi) = \mathsf{Verify}(\mathsf{hk}, h, \mathsf{ind}, 1, \pi') = 1$$

We say that a hash family is somewhere perfectly binding w.r.t opening if for any security parameter $\lambda \in \mathbb{N}$, any message length $L < 2^\lambda$, and any index $\mathsf{ind} \leq L$, we have
$$\Pr\left[\mathsf{hk} \text{ is binding w.r.t opening at index } \mathsf{ind} : \mathsf{hk} \leftarrow \mathsf{Setup\text{-}Bind}(1^\lambda, L, \mathsf{ind})\right] = 1.$$

We note that such hash functions have been constructed in [HW15, OPWW15] from assumptions such as LWE, DDH and DCR.

2.4 Verifiable and Anonymous Identity Based Encryption

In this section, we define verifiable and anonymous identity-based encryption (VAIBE) and describe its correctness and security properties. This is an identity based encryption system with 2 additional features. First, the ciphertext does not reveal any information about the identity used in encrypting the message. Second, there is a deterministic verification algorithm which can be used to verify if the given secret key corresponds to the given identity. Formally, the encryption system for message space $\{\mathcal{M}_\lambda\}_\lambda$, identity space $\{\mathcal{I}_\lambda\}_\lambda$ and key space $\{\mathcal{K}_\lambda\}_\lambda$ is defined as follows.

- $\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{mpk}, \mathsf{msk})$. The setup algorithm takes as input a security parameter λ. It outputs a master public key mpk and a master secret key msk.
- $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}) \rightarrow (\mathsf{sk}_{\mathsf{id}}, \pi)$. The key generation algorithm takes as input a master secret key msk and an identity id. It outputs a secret key $\mathsf{sk}_{\mathsf{id}}$ and a (possibly empty) proof π.
- $\mathsf{Encrypt}(\mathsf{mpk}, \mathsf{id}, m) \rightarrow \mathsf{ct}$. The encryption algorithm takes as input a master public key mpk, an identity id, a message m and outputs a ciphertext ct.
- $\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}}, \mathsf{ct}) \rightarrow m/\perp$. The decryption algorithm takes as input a secret key $\mathsf{sk}_{\mathsf{id}}$, a ciphertext ct and outputs a message m or $\perp$.
- $\mathsf{Verify}(\mathsf{mpk}, \mathsf{id}, \mathsf{sk}_{\mathsf{id}}, \pi) \rightarrow 0/1$. The verification algorithm is a deterministic algorithm that takes as input a master public key mpk, an identity id, a secret key $\mathsf{sk}_{\mathsf{id}}$, a proof π and outputs either 0 or 1.

Correctness. A VAIBE scheme is said to be correct if for every security parameter $\lambda \in \mathbb{N}$, every identity $\mathsf{id} \in \mathcal{I}_\lambda$, every message $m \in \mathcal{M}_\lambda$, every $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{sk}_{\mathsf{id}}, \pi) \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$ and $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{id}, m)$, we have

- Correctness of encryption: $\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}}, \mathsf{ct}) = m$.
- Correctness of verification: $\mathsf{Verify}(\mathsf{mpk}, \mathsf{id}, \mathsf{sk}_{\mathsf{id}}, \pi) = 1$.

Security. A VAIBE scheme is said to be secure if it satisfies the following 3 security properties.

Definition 2.6 (IND-CPA security)**.** A VAIBE scheme is said to be selective IND-CPA secure if for every stateful PPT adversary $\mathcal{A}$, there exists a negligible function $\mathrm{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$, we have

$$\Pr\left[\mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{mpk}, \mathsf{ct}) = b \;:\; \begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda); \\ (\mathsf{id}^*, m_0, m_1) \leftarrow \mathcal{A}(1^\lambda); b \leftarrow \{0,1\}; \\ \mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{id}^*, m_b) \end{array}\right] \leq 1/2 + \mathrm{negl}(\lambda).$$

where the adversary $\mathcal{A}$ is not allowed to make key generation query on identity id^*.

Definition 2.7 (Anonymous IBE)**.** A VAIBE scheme is said to be selective IND-ANON secure if for every stateful PPT adversary $\mathcal{A}$, there exists a negligible function $\mathrm{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$, we have

$$\Pr\left[\mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{mpk}, \mathsf{ct}) = b \;:\; \begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda); \\ (\mathsf{id}_0, \mathsf{id}_1, m) \leftarrow \mathcal{A}(1^\lambda); b \leftarrow \{0,1\}; \\ \mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{id}_b, m) \end{array}\right] \leq 1/2 + \mathrm{negl}(\lambda).$$

where the adversary $\mathcal{A}$ is not allowed to make key generation queries on identities id_0 and id_1.

Definition 2.8 (Soundness of Verifiability)**.** For every security parameter $\lambda \in \mathbb{N}$, every identity $\mathsf{id} \in \mathcal{I}_\lambda$, every $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, every secret key $\mathsf{sk} \in \mathcal{K}_\lambda$, we need

$$\exists \pi \text{ s.t. } \mathsf{Verify}(\mathsf{mpk}, \mathsf{id}, \mathsf{sk}, \pi) = 1 \implies \forall m \in \mathcal{M}_\lambda, \mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \mathsf{id}, m), \text{ we have } \mathsf{Decrypt}(\mathsf{sk}, \mathsf{ct}) = m.$$

2.5 Verifiable Random Function

In this section, we define verifiable random function (VRF) [MRV99] and describe its correctness and security properties. VRFs are similar to PRFs with an additional feature: VRF evaluation algorithm additionally outputs a proof using which the output of the evaluation algorithm can be verified. Formally, VRFs with input domain $\{\mathcal{X}_\lambda\}_\lambda$ and output domain $\{\mathcal{Y}_\lambda\}_\lambda$ are defined as follows.

- $\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{sk}, \mathsf{vk})$. The setup algorithm takes as input a security parameter 1^λ. It outputs a secret key sk and a verification key vk.
- $\mathsf{Eval}(\mathsf{sk}, x) \rightarrow (y, \pi)$. The evaluation algorithm takes as input a secret key sk and a message x. It outputs a value y and a proof π.
- $\mathsf{Verify}(\mathsf{vk}, x, y, \pi) \rightarrow 0/1$. The verification algorithm takes as input a verification key vk, a message x in input domain, a value y in output domain, a proof π and outputs either 0 or 1.

Correctness. A VRF scheme is said to be correct if for every security parameter $\lambda \in \mathbb{N}$, every $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup}(1^\lambda)$, every message $x \in \mathcal{X}_\lambda$ and $(y, \pi) \leftarrow \mathsf{Eval}(\mathsf{sk}, x)$, we have

$$\mathsf{Verify}(\mathsf{vk}, x, y, \pi) = 1.$$

Security. A VRF is said to be secure if it satisifies the following 2 security properties.

Definition 2.9 (Selective Pseudorandomness)**.** For every stateful PPT Adversary $\mathcal{A}$, there exists a negligible function negl$(\cdot)$ such that, for every $\lambda \in \mathbb{N}$, we have

$$\Pr\left[\mathcal{A}^{\mathsf{Eval}(\mathsf{sk},\cdot)}(\mathsf{vk}, y_b) = b \ : \ \begin{array}{c} (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup}(1^\lambda); b \leftarrow \{0,1\}; \\ x^* \leftarrow \mathcal{A}(1^\lambda); \\ (y_0, \pi) \leftarrow \mathsf{Eval}(\mathsf{sk}, x^*); y_1 \leftarrow \mathcal{Y}_\lambda \end{array}\right] \leq 1/2 + \mathrm{negl}(\lambda).$$

where the adversary is not allowed to make evaluation query on input x^*.

Definition 2.10 (Unique Provability)**.** For every $\lambda \in \mathbb{N}$ and every tuple $(\mathsf{vk}, x, y_1, \pi_1, y_2, \pi_2)$, where $x \in \mathcal{X}_\lambda, y_1, y_2 \in \mathcal{Y}_\lambda, y_1 \neq y_2$, the following must hold for at least one of $i \in \{1, 2\}$.

$$\mathsf{Verify}(\mathsf{vk}, x, y_i, \pi_i) = 0.$$

Although existing schemes achieve the unique provability property for even maliciously generated verification keys, we note that for the purpose of this paper, it is sufficient for a VRF scheme to satisfy the property only for verification keys generated by the setup algorithm.

2.6 Perfectly Binding Commitments

In this section, we define perfectly binding computationally hiding commitments (PB-CH Coms) and describe its security properties. This primitive can be constructed from *injective* one-way functions. Formally, a commitment scheme with randomness space $\{\mathcal{R}_\lambda\}_\lambda$ and commitment space $\{\mathcal{C}_\lambda\}_\lambda$ is defined as follows.

- $\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{pp}$. The setup algorithm takes as input a security parameter λ and outputs public parameters pp.
- $\mathsf{Commit}(\mathsf{pp}, b; r) \rightarrow c$. The commit algorithm is a randomized algorithm that takes as input the public parameters pp, a bit b to be committed, random coins r and outputs a commitment c.
- $\mathsf{Verify}(\mathsf{pp}, b, c, \pi) \rightarrow 0/1$. The verification algorithm takes as input the public parameters pp, a bit b, a commitment c and an opening π. It outputs either 0 or 1.

For simplicity, we assume that the opening for a commitment is simply the randomness used during the commitment phase. As a result, we do not have a separate 'reveal' algorithm. Below we formally define perfectly binding and computationally hiding requirements.

Definition 2.11. (Perfect Correctness) A commitment scheme is said to be correct if for all $\lambda \in \mathbb{N}$, every public parameter $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, any bit $b \in \{0, 1\}$ and randomness $r \in \mathcal{R}_\lambda$, we have

$$\text{if } c = \mathsf{Commit}(\mathsf{pp}, b; r), \text{ then } \mathsf{Verify}(\mathsf{pp}, b, c, r) = 1.$$

Definition 2.12. (PB-CH Commitments) A pair of polynomial time algorithms $(\mathsf{Commit}, \mathsf{Verify})$ is a perfectly binding computationally hiding (PB-CH) commitment scheme if it satisfies the following conditions:

- (Perfect Binding) For every $\lambda \in \mathbb{N}$, every public parameter $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, every $(c, r_0, r_1) \in \{\mathcal{C}_\lambda \times \mathcal{R}_\lambda \times \mathcal{R}_\lambda\}$, the following holds for at least one $b \in \{0, 1\}$:

$$\mathsf{Verify}(\mathsf{pp}, b, c, r_b) = 0.$$

- (Computationally Hiding) For every PPT adversary $\mathcal{A}$, there exists a negligible function negl$(\cdot)$ such that for every $\lambda \in \mathbb{N}$, we have

$$\Pr\left[\mathcal{A}(\mathsf{pp}, c) = b \ : \ \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), b \leftarrow \{0,1\}, r \leftarrow \mathcal{R}_\lambda, c \leftarrow \mathsf{Commit}(\mathsf{pp}, b; r)\ \right] \leq \frac{1}{2} + \mathrm{negl}(\lambda).$$

3 Revisiting Broadcast and Trace System

3.1 Broadcast and Trace System

In this section, we formally define Broadcast and Trace system and describe its security properties. The security definition is motivated by a recent work by Goyal et al. [GKRW17] which points out problems with previously proposed notions of traitor tracing and proposes an indistinguishability based security definiton for the primitive.

- $\mathsf{Setup}(1^\lambda, 1^N) \to (\mathsf{pk}, \{\mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_N\})$. The setup algorithm takes as input a security parameter λ and number of users N. It outputs a public key pk, and secret keys for N users $\{\mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_N\}$.
- $\mathsf{Encrypt}(\mathsf{pk}, S, m) \to \mathsf{ct}$. The encryption algorithm takes as input public key pk, a set $S \subseteq [N]$ of users, a message m and outputs a ciphertext ct.
- $\mathsf{Decrypt}(i, \mathsf{sk}_i, \mathsf{pk}, S, \mathsf{ct}) \to m/\perp$. The decryption algorithm takes as input an index $i \in [N]$, secret key of i^{th} user, public key pk, a set of users $S \subseteq [N]$, a ciphertext ct and outputs either a message m or $\perp$.
- $\mathsf{Trace}^D(\mathsf{pk}, S_D, m_0, m_1, 1^{1/\epsilon}) \to S^*$. The tracing algorithm takes as input a public key pk, a set of users S_D, two messages m_0, m_1 and parameter $\epsilon < 1$. The algorithm has a black-box access to the decoder D and outputs a set of indices $S^* \subseteq [N]$.

Correctness. The Broadcast and Trace system is said to be correct if for every $\lambda \in \mathbb{N}$, any number of users $N \in \mathbb{N}$, every subset of users $S \subseteq [N]$, every message $m \in \mathcal{M}_\lambda$, every user $i \in S$, $(\mathsf{pk}, \{\mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_N\}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$ and $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S, m)$, we have

$$\mathsf{Decrypt}(i, \mathsf{sk}_i, \mathsf{pk}, S, \mathsf{ct}) = m.$$

Security. Intuitively, the system is said to be secure if it is IND-CPA secure and if no poly-time adversary can produce a decoder that can fool the tracing algorithm. We formally define both of these properties below.

Definition 3.1 (IND-CPA security)**.** We say that a Broadcast and Trace scheme is IND-CPA secure if for every stateful PPT adversary $\mathcal{A}$, there exists a negligible function negl$(\cdot)$ such that for all $\lambda \in \mathbb{N}$, the following holds

$$\Pr\left[\mathcal{A}^{O(\cdot)}(\mathsf{ct}) = b \;:\; \begin{array}{c} 1^N \leftarrow \mathcal{A}(1^\lambda); (\mathsf{pk}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_N)) \leftarrow \mathsf{Setup}(1^\lambda, 1^N); \\ (S', m_0, m_1) \leftarrow \mathcal{A}^{O(\cdot)}(\mathsf{pk}); b \leftarrow \{0,1\}; \mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S', m_b) \end{array}\right] \leq \frac{1}{2} + \mathrm{negl}(\lambda).$$

Here, $O(\cdot)$ is an oracle that has $\{\mathsf{sk}_i\}_{i\in[N]}$ hardwired, takes as input an index $i \in [n]$ and outputs sk_i. Let the set of indices queried by the adversary to the oracle be $S \subseteq [N]$. Then the adversary is restricted to output the challenge set S' such that $S' \subseteq [N] \setminus S$.

Definition 3.2 (IND-secure Traitor Tracing)**.** Let $(\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Trace})$ be a Broadcast and Trace scheme. For any non-negligible function $\epsilon(\cdot)$ and stateful PPT adversary $\mathcal{A}$, consider the experiment $\mathsf{Expt\text{-}BT}_{\mathcal{A},\epsilon}(\lambda)$ defined as follows.

In order to define the security of tracing mechanism, we define the following events and probabilities as a function of security parameter λ.

- $\mathsf{Good\text{-}Dec}_{\mathcal{A},\epsilon} : \Pr[D(\mathsf{ct}) = b : b \leftarrow \{0,1\}, \mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S_D, m_b)] \geq 1/2 + \epsilon(\lambda)$
  $\mathsf{Pr\text{-}Good\text{-}Dec}_{\mathcal{A},\epsilon}(\lambda) = \Pr[\mathsf{Good\text{-}Dec}_{\mathcal{A},\epsilon}]$
- $\mathsf{Correct\text{-}Tr}_{\mathcal{A},\epsilon} : |S^*| > 0, S^* \subseteq S \cap S_D$
  $\mathsf{Pr\text{-}Correct\text{-}Tr}_{\mathcal{A},\epsilon}(\lambda) = \Pr[\mathsf{Correct\text{-}Tr}_{\mathcal{A},\epsilon}]$

Experiment $\mathsf{Expt\text{-}BT}_{\mathcal{A},\epsilon}(\lambda)$

- $1^N \leftarrow \mathcal{A}(1^\lambda)$.
- $(\mathsf{pk}, (\mathsf{sk}_1, \dots, \mathsf{sk}_N)) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$.
- $(D, S_D, m_0, m_1) \leftarrow \mathcal{A}^{O(\cdot)}(\mathsf{pk})$.
- $S^* \leftarrow \mathsf{Trace}^D(\mathsf{pk}, S_D, m_0, m_1, 1^{1/\epsilon(\lambda)})$.

Here, $O(\cdot)$ is an oracle that has $\{\mathsf{sk}_i\}_{i\in[N]}$ hardwired, takes as input an index $i \in [N]$ and outputs sk_i. Let S be the set of indices queried by $\mathcal{A}$.

Figure 1: Experiment $\mathsf{Expt\text{-}BT}$

- $\mathsf{False\text{-}Tr}_{\mathcal{A},\epsilon} : S^* \not\subseteq S \cap S_D$
  $\mathsf{Pr\text{-}False\text{-}Tr}_{\mathcal{A},\epsilon}(\lambda) = \Pr[\mathsf{False\text{-}Tr}_{\mathcal{A},\epsilon}]$

The Broadcast and Trace scheme is said to have Ind-secure tracing mechanism if for every stateful PPT adversary $\mathcal{A}$, polynomial $q(\cdot)$ and non-negligible function $\epsilon(\cdot)$, there exists negligible functions $\mathrm{negl}_1(\cdot)$ and $\mathrm{negl}_2(\cdot)$ such that for all $\lambda \in \mathbb{N}$ satisfying $\epsilon(\lambda) > 1/q(\lambda)$, $\mathsf{Pr\text{-}Correct\text{-}Tr}_{\mathcal{A},\epsilon}(\lambda) \geq \mathsf{Pr\text{-}Good\text{-}Dec}_{\mathcal{A},\epsilon}(\lambda) - \mathrm{negl}_1(\lambda)$ and $\mathsf{Pr\text{-}False\text{-}Tr}_{\mathcal{A},\epsilon}(\lambda) \leq \mathrm{negl}_2(\lambda)$.

3.2 Augmented Broadcast Encryption

In this section, we define Augmented Broadcast Encryption (AugBE) and its security properties.

- $\mathsf{Setup}(1^\lambda, 1^N) \rightarrow (\mathsf{pk}, \{\mathsf{sk}_1, \dots, \mathsf{sk}_N\})$. The setup algorithm takes as input security parameter λ and number of users N. It outputs a public key pk and secret keys $\{\mathsf{sk}_1, \dots, \mathsf{sk}_N\}$, where sk_i is the secret key for user i.

- $\mathsf{Encrypt}(\mathsf{pk}, S, m, \mathsf{ind}) \rightarrow \mathsf{ct}$. The encryption algorithm takes as input public key pk, a set of users $S \subseteq [N]$, a message m, and an index $\mathsf{ind} \in [N+1]$. It outputs a ciphertext ct.

- $\mathsf{Decrypt}(i, \mathsf{sk}_i, \mathsf{pk}, S, \mathsf{ct}) \rightarrow m/\perp$. The decryption algorithm takes as input an index i, secret key for i^{th} user sk_i, public key pk, a set of users $S \subseteq [N]$, a cipheretxt ct and outputs a message m or $\perp$.

Correctness. An AugBE scheme is said to be correct if for every security parameter $\lambda \in \mathbb{N}$, any number of users $N \in \mathbb{N}$, any message $m \in \mathcal{M}_\lambda$, any subset of users $S \subseteq [N]$, any index $\mathsf{ind} \in [N]$, any $i \in S \cap \{\mathsf{ind}, \mathsf{ind}+1, \dots, N\}$, $(\mathsf{pk}, \{\mathsf{sk}_1, \mathsf{sk}_2, \dots, \mathsf{sk}_N\}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$ and $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S, m, \mathsf{ind})$, we have

$$\mathsf{Decrypt}(i, \mathsf{sk}_i, \mathsf{pk}, S, \mathsf{ct}) = m.$$

Security. We need AugBE to satisfy 2 security properties. The first is *message hiding* property which states that no PPT adversary can distinguish between encryptions of m_0 and m_1 encrypted using the last index $N+1$. The second is *index hiding* property which states that ciphertexts encrypted to index ind do not reveal any non-trivial information about the index. We formally define the security properties below.

Definition 3.3 (Message Hiding)**.** We say that an AugBE scheme satisfies message hiding property if for every stateful PPT adversary $\mathcal{A}$, there exists a negligible function $\mathrm{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$, the following holds

$$\Pr\left[\mathcal{A}^{O(\cdot)}(\mathsf{ct}) = b \ : \ \begin{array}{c} 1^N \leftarrow \mathcal{A}(1^\lambda); (\mathsf{msk}, \mathsf{pk}, (\mathsf{sk}_1, \dots, \mathsf{sk}_N)) \leftarrow \mathsf{Setup}(1^\lambda, 1^N); \\ (S', m_0, m_1) \leftarrow \mathcal{A}^{O(\cdot)}(\mathsf{pk}); b \leftarrow \{0,1\}; \mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S', m_b, N+1) \end{array}\right] \leq \frac{1}{2} + \mathrm{negl}(\lambda).$$

Here, $O(\cdot)$ is an oracle that has $\{\mathsf{sk}_i\}_{i\in[N]}$ hardwired, takes as input an index $i \in [N]$ and outputs sk_i.

Definition 3.4 (Index Hiding)**.** We say that an AugBE scheme satisfies index hiding property if for every stateful PPT adversary $\mathcal{A}$, there exists a negligible function negl$(\cdot)$ such that for every $\lambda \in \mathbb{N}$, the following holds,

$$\Pr\left[\mathcal{A}^{O(\cdot)}(\mathsf{ct}) = b \;:\; \begin{array}{c} (1^N, \mathsf{ind}) \leftarrow \mathcal{A}(1^\lambda); (\mathsf{msk}, \mathsf{pk}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_N)) \leftarrow \mathsf{Setup}(1^\lambda, 1^N); \\ (S', m) \leftarrow \mathcal{A}^{O(\cdot)}(\mathsf{pk}); b \leftarrow \{0,1\}; \mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S', m, \mathsf{ind} + b) \end{array}\right] \leq \frac{1}{2} + \mathrm{negl}(\lambda).$$

Here, $O(\cdot)$ is an oracle that has $\{\mathsf{sk}_i\}_{i \in [N]}$ hardwired, takes as input an index $i \in [N]$ and outputs sk_i. Let the set of keys queried by the adversary be S. We restrict the adversary to satisfy $\mathsf{ind} \notin S' \vee \mathsf{ind} \notin S$.

3.3 Broadcast and Trace from AugBE

In this section, we construct a Broadcast and Trace system assuming we have an AugBE scheme. The construction is same as [BW06a], but we modify their security proof as per indistinguishability based definition of Broadcast and Trace. The construction proceeds as follows.

- $\mathsf{Setup}_{\mathsf{BT}}(1^\lambda, 1^N) : \mathsf{Setup}_{\mathsf{AugBE}}(1^\lambda, 1^N)$
- $\mathsf{Encrypt}_{\mathsf{BT}}(\mathsf{pk}, S, m) : \mathsf{Encrypt}_{\mathsf{AugBE}}(\mathsf{pk}, S, m, 1)$
- $\mathsf{Decrypt}_{\mathsf{BT}}(i, \mathsf{sk}_i, \mathsf{pk}, S, \mathsf{ct}) : \mathsf{Decrypt}_{\mathsf{AugBE}}(i, \mathsf{sk}_i, \mathsf{pk}, S, \mathsf{ct})$
- $\mathsf{Trace}_{\mathsf{BT}}(\mathsf{pk}, S_D, m_0, m_1, 1^{1/\epsilon})$:

```
For index $i = 1$ to $N + 1$:
    Set count $= 0$
    For step $= 1$ to $T$ : $(T = 8\lambda(N/\epsilon)^2)$
        Sample $b \leftarrow \{0, 1\}$
        $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{AugBE}}(\mathsf{pk}, S, m_b, i)$
        if $D(\mathsf{ct}) = b$ then count $=$ count $+ 1$
    Set $\hat{p}_i = \frac{\mathsf{count}}{T}$
Output $\{i : i \leq N, \hat{p}_i - \hat{p}_{i+1} \geq \frac{\epsilon}{4N}\}$.
```

The correctness of the above scheme follows from the correctness of the underlying AugBE scheme. We now prove that the above scheme is a secure Broadcast and Trace system assuming that the underlying AugBE scheme has message hiding and index hiding properties. We first prove IND-CPA security of the construction.

3.3.1 IND-CPA security

Theorem 3.1. Assuming that the AugBE scheme has message hiding and index hiding properties, the Broadcast and Trace system described above is IND-CPA secure.

Proof. The IND-CPA security of the construction is already proved in [BW06a]. So, we only present the proof at a high level. The proof proceeds using a sequence of hybrids defined as follows. Game 1 is equivalent to IND-CPA game. Game i is similar to the IND-CPA game except that the challenger encrypts the challenge message using index i.

Game i ($i \in [N+1]$).

- *Setup Phase.* The adversary $\mathcal{A}$ sends the number of users 1^N to the challenger. The challenger samples $(\mathsf{pk}, \{\mathsf{sk}_1, \ldots, \mathsf{sk}_N\}) \leftarrow \mathsf{Setup}_{\mathsf{AugBE}}(1^\lambda, 1^N)$ and sends public key pk to the adversary $\mathcal{A}$.
- *Pre-Challenge Query Phase.* The adversary then adaptively queries for a subset of the secret keys. For each query j, the challenger responds with secret key sk_j.

- *Challenge Phase.* The adversary then sends a set $S' \subseteq [N]$ and messages m_0, m_1 to the challenger. The challenger samples a bit $b \leftarrow \{0,1\}$, $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{AugBE}}(\mathsf{pk}, S', m_b, i)$ and sends ct to the adversary.
- *Post-Challenge Query Phase.* This is identical to Pre-Challenge Query Phase.
- *Output Phase* The adversary sends a bit b' to the challenger. The adversary wins if $b' = b$.

Let the set of key queries made by the adversary be S. The adversary is restricted to produce challenge s.t. $S' \cap S = \emptyset$. For any PPT Adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in game $\mathsf{Game}\ i$ is defined as $\mathsf{Adv}_i^{\mathcal{A}}(\lambda) = \Pr[\mathcal{A} \text{ wins }] - 1/2$. We prove that the advantage of any PPT adversary $\mathcal{A}$ in $\mathsf{Game}\ i$ is at most negligible in security parameter. Claim 3.1 establishes that the difference of the adversary's advantage between each adjacent game is at most negligible in the security parameter. Finally, Claim 3.2 we show that if any adversary wins in the last game, then it wins message hiding game against AugBE challenger as well.

Claim 3.1. Assuming index hiding property of AugBE, for every stateful PPT adversary $\mathcal{A}$, $i \in [N]$, there exists a negligible function $\text{negl}(\cdot)$ such that $\mathsf{Adv}_i^{\mathcal{A}}(\lambda) - \mathsf{Adv}_{i+1}^{\mathcal{A}}(\lambda) \leq \text{negl}(\lambda)$.

Claim 3.2. Assuming message hiding property of AugBE, for every stateful PPT adversary $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that $\mathsf{Adv}_{N+1}^{\mathcal{A}} \leq \text{negl}(\lambda)$.

Note that using the above two claims and triangle inequality, for any stateful PPT adversary $\mathcal{A}$, we have $\mathsf{Adv}_1^{\mathcal{A}}(\lambda) \leq \text{negl}(\lambda)$ for some negligible function $\text{negl}(\cdot)$. ■

3.3.2 Correctness of Tracing

We now prove that no stateful PPT adversary can fool the tracing mechanism of the above scheme. The following analysis is based on the analysis provided in [GKW18], which analyzes traitor tracing construction from private linear broadcast encryption (PLBE) [BSW06]. We would like to point that in [GKW18], the authors introduced the notion of decoder-based PLBE, proved that 1-query PLBE implies decoder-based PLBE and decoder-based PLBE implies traitor tracing. In this paper, we prove that AugBE implies Broadcast and Trace using similar techniques, but without introducing an intermediate primitive.

False Trace Probability. We prove that the above tracing algorithm does not falsely accuse any user. Specifically, no stateful PPT adversary can output a decoder such that the tracing algorithm when executed on the decoder falsely outputs an index that is not queried by the adversary with non-negligible probability. Formally, we prove the following theorem.

Theorem 3.2. For every stateful PPT adversary $\mathcal{A}$, polynomial $q(\cdot)$ and non-negligible function $\epsilon(\cdot)$, there exists a negligible function $\text{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$ satisfying $\epsilon(\lambda) \geq 1/q(\lambda)$,

$$\mathsf{Pr\text{-}False\text{-}Tr}_{\mathcal{A},\epsilon}(\lambda) \leq \text{negl}(\lambda)$$

Proof. Consider any stateful PPT adversary $\mathcal{A}$ in the tracing game described in Definition 3.2. It outputs a decoder D, a set S_D and a pair of messages m_0, m_1. Let S be the set of keys queried by $\mathcal{A}$. For $1 \leq i \leq N+1$, let us define $p_i = \Pr\left[D(\mathsf{ct}) = b : b \leftarrow \{0,1\}, \mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{AugBE}}(\mathsf{pk}, S_D, m_b, i)\right]$. For $1 \leq i \leq N$, let us define the events $\mathsf{Diff\text{-}Adv}_{i,\epsilon}^D : p_i - p_{i+1} \geq \frac{\epsilon}{8N}$ and $\mathsf{Diff\text{-}Adv}_{\epsilon}^D : \vee_{k \notin S \cap S_D} \mathsf{Diff\text{-}Adv}_{k,\epsilon}^D$. Note that

$$\begin{aligned}
\mathsf{Pr\text{-}False\text{-}Tr}_{\mathcal{A},\epsilon}(\lambda) &\leq \Pr[\mathsf{False\text{-}Tr}_{\mathcal{A},\epsilon} | \overline{\mathsf{Diff\text{-}Adv}_{\epsilon}^D}] + \Pr[\mathsf{False\text{-}Tr}_{\mathcal{A},\epsilon} \wedge \mathsf{Diff\text{-}Adv}_{\epsilon}^D] \\
&\leq \Pr[\mathsf{False\text{-}Tr}_{\mathcal{A},\epsilon} | \overline{\mathsf{Diff\text{-}Adv}_{\epsilon}^D}] + \Pr[\mathsf{Diff\text{-}Adv}_{\epsilon}^D] \\
&\leq \Pr[\mathsf{False\text{-}Tr}_{\mathcal{A},\epsilon} | \overline{\mathsf{Diff\text{-}Adv}_{\epsilon}^D}] + \sum_{i \in [N]} \Pr[i \notin S \cap S_D \wedge \mathsf{Diff\text{-}Adv}_{i,\epsilon}^D]
\end{aligned}$$

We hereby show that each of the terms in the expression is upper bounded by a negligible function.

Lemma 3.1. For every stateful PPT adversary $\mathcal{A}$, polynomial $q(\cdot)$ and non-negligible function $\epsilon(\cdot)$, there exists a negligible function $\text{negl}_1(\cdot)$ such that for all $\lambda \in \mathbb{N}$ satisfying $\epsilon(\lambda) \geq 1/q(\lambda)$,

$$\Pr[\mathsf{False\text{-}Tr}_{\mathcal{A},\epsilon} | \overline{\mathsf{Diff\text{-}Adv}^{D}_{\epsilon}}] \leq \text{negl}_1(\lambda)$$

Proof. We are given that $\wedge_{i \notin S \cap S_D} p_i - p_{i+1} < \epsilon/8N$ and we would like to prove that $\Pr[\vee_{i \notin S \cap S_D} \hat{p}_i - \hat{p}_{i+1} \geq \epsilon/4N] \leq \text{negl}_1(\lambda)$. Let us compute $\Pr[\hat{p}_i - \hat{p}_{i+1} \geq \epsilon/4N]$ for some $i \notin S \cap S_D$. The tracing algorithm iteratively samples $b \leftarrow \{0,1\}$, $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{AugBE}}(\mathsf{pk}, S, m_b, i)$ and checks if $D(\mathsf{ct}) = b$. Let $X_{i,j}$ be an indicator random variable which takes value 1 if the check succeeds in the j^{th} iteration. Let $Z_{i,j} = X_{i,j} - X_{i+1,j}$. We know that, $\forall i, j$, $\hat{p}_i = \frac{1}{T} \sum_{j=1}^{T} X_{i,j}, E[X_{i,j}] = p_i$ and $\mu_i = E[Z_{i,j}] = p_i - p_{i+1}$. Since $Z_{i,j}$s are independent samples, by applying the chernoff bound, we get $\Pr[\frac{1}{T} \sum_j Z_{i,j} \geq 2 \cdot \frac{\epsilon}{8N}] \leq \Pr[\frac{1}{T} \sum_j Z_{i,j} \geq 2 \cdot \mu_i] \leq 2^{-O(\lambda)}$. Using this, we can say that for every $i \notin S \cap S_D$, $\Pr[i \in S^* | \overline{\mathsf{Diff\text{-}Adv}^{D}_{\epsilon}}] \leq 2^{-O(\lambda)}$, where S^* is the output of the tracing algorithm. Using union bound, we obtain

$$\Pr[\mathsf{False\text{-}Tr}_{\mathcal{A},\epsilon} | \overline{\mathsf{Diff\text{-}Adv}^{D}_{\epsilon}}] \leq N \cdot 2^{-O(\lambda)} = \text{negl}_1(\lambda)$$

■

Lemma 3.2. Assuming index hiding property of AugBE, for every PPT adversary $\mathcal{A}$, polynomial $q(\cdot)$ and non-negligible function $\epsilon(\cdot)$, there exists a negligible function $\text{negl}_2(\cdot)$ such that for all $\lambda \in \mathbb{N}$ satisfying $\epsilon(\lambda) \geq 1/q(\lambda)$ and $i \in [N]$,

$$\Pr[i \notin S \cap S_D \wedge \mathsf{Diff\text{-}Adv}^{D}_{i,\epsilon}] \leq \text{negl}_2(\lambda)$$

Proof. Suppose there exists a PPT adversary $\mathcal{A}$, polynomial $q(\lambda)$ and non-negligible functions $\epsilon(\cdot), \delta(\cdot)$ such that for every $\lambda \in \mathbb{N}$ satisfying $\epsilon(\lambda) \geq \frac{1}{q(\lambda)}$, there exists an $i^* \in [N]$ such that $\Pr[i^* \notin S \cap S_D \wedge \mathsf{Diff\text{-}Adv}^{D}_{i^*,\epsilon}] \geq \delta(\lambda)$. We use this adversary $\mathcal{A}$ to build a reduction algorithm $\mathcal{B}$ that can break index hiding property of the underlying AugBE scheme.

The reduction algorithm $\mathcal{B}$ receives number of users 1^N from $\mathcal{A}$ and chooses a random $i \leftarrow [N]$. $\mathcal{B}$ then sends 1^N to the challenger of index hiding game at index i. The challenger samples public key and secret keys of AugBE scheme and sends the public key to $\mathcal{B}$, which forwards the public key to $\mathcal{A}$. $\mathcal{A}$ then adaptively queries $\mathcal{B}$ for secret keys. If $\mathcal{A}$ queries for i, then $\mathcal{B}$ outputs a uniform random bit and aborts. If $\mathcal{A}$ queries for $j \neq i$, then $\mathcal{B}$ forwards the query to $\mathcal{C}$. The challenger responds with the corresponding secret key to $\mathcal{B}$, which forwards the secret key to $\mathcal{A}$. After all queries, $\mathcal{A}$ sends a decoding box D, messages m_0, m_1 and set S_D to $\mathcal{B}$. Let the set of key queries made by $\mathcal{A}$ be S. If $i \in S \cap S_D$, $\mathcal{B}$ outputs a uniformly random bit and aborts. If $i \notin S \cap S_D$, $\mathcal{B}$ continues playing the game and chooses a random bit $\gamma \leftarrow \{0,1\}$ and sends S_D, m_γ to $\mathcal{C}$. Note that, $\mathcal{B}$ acts as a valid index hiding game adversary to $\mathcal{C}$. The challenger chooses a random bit α and responds with $\mathsf{ct}_1 \leftarrow \mathsf{Encrypt}_{\mathsf{AugBE}}(\mathsf{pk}, S_D, m_\gamma, i + \alpha)$. $\mathcal{B}$ then chooses a random bit β and computes $\mathsf{ct}_2 \leftarrow \mathsf{Encrypt}_{\mathsf{AugBE}}(\mathsf{pk}, S_D, m_\gamma, i + \beta)$. $\mathcal{B}$ outputs β if $D(\mathsf{ct}_1) = D(\mathsf{ct}_2)$ and outputs $1 - \beta$ otherwise. The reduction algorithm wins if its output is equal to α.

Let's analyze the probability that $\mathcal{B}$ wins in the index hiding game when $\mathcal{B}$ does not abort ($i \notin S \cap S_D$). Let $p_{j,b} = \Pr[D(\mathsf{ct}) = b : \mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{AugBE}}(\mathsf{pk}, S_D, m_b, j)]$, where the probability is taken over the coin tosses of the decoder D and the encryption algorithm. Note that $p_j = (p_{j,0} + p_{j,1})/2$. Let E^D be the event that $\mathcal{B}$ wins the index hiding game, E^D_b be the event that $\mathcal{B}$ wins when it choses to send m_b to the challenger. Therefore,

$$\begin{aligned}
\Pr[E^D_b] &= 1/4(\Pr[E^D_b | \alpha = 0, \beta = 0] + \Pr[E^D_b | \alpha = 0, \beta = 1] + \Pr[E^D_b | \alpha = 1, \beta = 0] + \Pr[E^D_b | \alpha = 1, \beta = 1]) \\
&= 1/4\Big(p^2_{i,b} + (1 - p_{i,b})^2 + 2(p_{i,b}(1 - p_{i+1,b}) + (1 - p_{i,b})p_{i+1,b}) + p^2_{i+1,b} + (1 - p_{i+1,b})^2\Big) \\
&= 1/2 + 1/2(p_{i,b} - p_{i+1,b})^2
\end{aligned} \tag{1}$$

By our assumption,

$$\Pr[i^* \notin S \cap S_D \wedge \mathsf{Diff\text{-}Adv}^{D}_{i^*,\epsilon}] \geq \delta(\lambda)$$

This implies,

$$\Pr[i = i^* \wedge i^* \notin S \cap S_D \wedge \mathsf{Diff\text{-}Adv}^{D}_{i^*,\epsilon}] \geq \delta(\lambda)/N$$

Let the event $i = i^* \wedge i^* \notin S \cap S_D \wedge \mathsf{Diff\text{-}Adv}^{D}_{i^*,\epsilon}$ be denoted by F. When the event F occurs, $p_i - p_{i+1} \geq \epsilon(\lambda)/8N$, and therefore $\exists b'$ s.t. $p_{i,b'} - p_{i+1,b'} \geq \epsilon(\lambda)/8N$. Irrespective of whether F occurs, $\Pr[E^D_b] \geq 1/2$ for $b \in \{0,1\}$. Note that,

$$\begin{aligned}\Pr[E^D_{b'}] &= \Pr[E^D_{b'}|F] \cdot \Pr[F] + \Pr[E^D_{b'}|\overline{F}] \cdot \Pr[\overline{F}] \\ &\geq \Big(\frac{1}{2} + \frac{1}{2}\Big(\frac{\epsilon(\lambda)}{8N}\Big)^2\Big)\frac{\delta(\lambda)}{N} + \frac{1}{2} \cdot \Big(1 - \frac{\delta(\lambda)}{N}\Big)\end{aligned} \tag{2}$$

$$\begin{aligned}\Pr[E^D] &= 1/2 \Pr[E^D_{b'}] + 1/2 \Pr[E^D_{1-b'}] \\ &\geq \frac{1}{2}\Big(\frac{1}{2} + \frac{\epsilon(\lambda)^2\delta(\lambda)}{128N^3}\Big) + \frac{1}{2} \cdot \frac{1}{2}\end{aligned} \tag{3}$$

As $S \neq [N]$, the probability that $\mathcal{B}$ doesn't abort is at least $1/N$. Therefore, the algorithm $\mathcal{B}$ breaks index hiding property of the underlying AugBE scheme with a non-negligible advantage.

■

From the above 2 lemmas, it follows that false tracing proabability $\mathsf{Pr\text{-}False\text{-}Tr}_{\mathcal{A},\epsilon}(\lambda) \leq \mathrm{negl}_1(\lambda) + N \cdot \mathrm{negl}_2(\lambda) = \mathrm{negl}(\lambda)$. ■

Correct Trace Probability. We prove that whenever an adversary produces a good decoder, the tracing algorithm correctly traces at least one of the keys queried by the adversary with all but negligible probability. Formally, we prove the following theorem.

Theorem 3.3. For every stateful PPT adversary $\mathcal{A}$, polynomial $q(\cdot)$ and non-negligible function $\epsilon(\cdot)$, there exists a negligible function $\mathrm{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$ satisfying $\epsilon(\lambda) \geq 1/q(\lambda)$,

$$\mathsf{Pr\text{-}Correct\text{-}Tr}_{\mathcal{A},\epsilon}(\lambda) \geq \mathsf{Pr\text{-}Good\text{-}Dec}_{\mathcal{A},\epsilon}(\lambda) - \mathrm{negl}(\lambda)$$

Proof. Consider a stateful PPT adversary $\mathcal{A}$ of the tracing game described in Definition 3.2. It outputs a decoder D, a set S_D and a pair of messages m_0, m_1. Let $S^* \leftarrow \mathsf{Trace}^D(S_D, m_0, m_1, 1^{1/\epsilon})$. We first compute the probability that S^* is non-empty. If the event $\mathsf{Good\text{-}Dec}_{\mathcal{A},\epsilon}$ occurs, we have $p_1 \geq 1/2 + \epsilon$. We know that $p_{N+1} \leq 1/2 + \mathrm{negl}_2(\lambda)$ for some negligible function $\mathrm{negl}_2(\cdot)$ due to the message hiding property of the underlying AugBE scheme. Hence if $\mathsf{Good\text{-}Dec}_{\mathcal{A},\epsilon}$ occurs, the set $R = \{i : p_i - p_{i+1} \geq \frac{\epsilon}{2N}\}$ is non-empty. By chernoff bound, we obtain

$$\forall i \in R, \Pr\Big[\hat{p}_i - \hat{p}_{i+1} < \frac{\epsilon}{4N}\Big] < \mathrm{negl}_1(\lambda)$$

for some negligible function $\mathrm{negl}_1(\cdot)$. Hence if $\mathsf{Good\text{-}Dec}_{\mathcal{A},\epsilon}$ occurs, S^* is non-empty set with all but non-negligible probability i.e.,

$$\Pr[S^* = \emptyset | \mathsf{Good\text{-}Dec}_{\mathcal{A},\epsilon}] \leq \sum_{i \in [N]} \Pr\Big[\hat{p}_i - \hat{p}_{i+1} < \frac{\epsilon}{4N}\Big| i \in R\Big] \leq N \cdot \mathrm{negl}_1(\lambda).$$

This implies,

$$\begin{aligned}\Pr[S^* \neq \emptyset] &\geq \Pr[S^* \neq \emptyset \wedge \mathsf{Good\text{-}Dec}_{\mathcal{A},\epsilon}] \\ &\geq (1 - N \cdot \mathrm{negl}_1(\lambda)) \cdot \mathsf{Pr\text{-}Good\text{-}Dec}_{\mathcal{A},\epsilon}(\lambda) \\ &\geq \mathsf{Pr\text{-}Good\text{-}Dec}_{\mathcal{A},\epsilon}(\lambda) - \mathrm{negl}_3(\lambda)\end{aligned}$$

for some negligible function $\mathrm{negl}_3(\cdot)$. Combining this result with Theorem 3.2, we get $\mathsf{Pr\text{-}Correct\text{-}Tr}_{\mathcal{A},\epsilon}(\lambda) \geq \mathsf{Pr\text{-}Good\text{-}Dec}_{\mathcal{A},\epsilon}(\lambda) - \mathrm{negl}(\lambda)$. ■

4 Construction of Augmented Broadcast Encryption

In this section, we construct an augmented broadcast encryption (AugBE) scheme from positional witness encryption (PWE), somewhere perfectly binding hash (SPB hash) function and all-but-one (ABO) signatures. We also prove that the construction satisfies the message hiding and index hiding properties.

Let $\mathcal{ABO} = (\mathsf{Setup}_{\mathsf{ABO}}, \mathsf{Setup\text{-}Punc}_{\mathsf{ABO}}, \mathsf{Sign}_{\mathsf{ABO}}, \mathsf{Verify}_{\mathsf{ABO}})$ be an ABO signature scheme with message space $\{0,1\}^\lambda$, signature space $\{0,1\}^{k(\lambda)}$, secret key space $\{\mathcal{S}_\lambda\}_\lambda$ and verification key space $\{\mathcal{V}_\lambda\}_\lambda$. Let $\mathcal{SPB} = (\mathsf{Setup}_{\mathsf{SPB}}, \mathsf{Setup\text{-}Bind}_{\mathsf{SPB}}, \mathsf{Hash}_{\mathsf{SPB}}, \mathsf{Open}_{\mathsf{SPB}}, \mathsf{Verify}_{\mathsf{SPB}})$ be an SPB hash function with hash key space $\{\mathcal{K}_\lambda\}_\lambda$, hash space $\{\mathcal{H}_\lambda\}_\lambda$ and hash opening space $\{0,1\}^{\ell(\lambda)}$. For simplicity of notation, we hereby use $\ell = \ell(\lambda)$ and $k = k(\lambda)$. Let $\mathcal{PWE} = (\mathsf{Encrypt}_{\mathsf{PWE}}, \mathsf{Decrypt}_{\mathsf{PWE}})$ be a PWE scheme with message space $\{\mathcal{M}_\lambda\}_\lambda$ with respect to the following language $\mathcal{L}$. The language $\mathcal{L}$ contains instances of the form $(1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}) \in 1^\lambda \times \{0,1\}^\lambda \times \mathcal{H}_\lambda \times \mathcal{K}_\lambda \times \mathcal{V}_\lambda$, where $\lambda \in \mathbb{N}$, with the following witness relation $\mathcal{R}$:

$$(1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}) \in \mathcal{L} \iff \begin{array}{c} \exists (i, \sigma, \pi) \in \{0,1\}^\lambda \times \{0,1\}^k \times \{0,1\}^\ell \text{ s.t.} \\ 1 \le i \le N \ \wedge\ \mathsf{Verify}_{\mathsf{ABO}}(\mathsf{vk}, i, \sigma) = 1 \ \wedge\ \mathsf{Verify}_{\mathsf{SPB}}(\mathsf{hk}, h, i, 1, \pi) = 1. \end{array}$$

Note that the above witness relation $\mathcal{R}$ is well defined as $\mathsf{Verify}_{\mathsf{ABO}}$ and $\mathsf{Verify}_{\mathsf{SPB}}$ are deterministic algorithms. We construct an AugBE scheme $\mathcal{AUGBE} = (\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{Decrypt})$ with message space $\{\mathcal{M}_\lambda\}_\lambda$. We sometimes slightly abuse notation and denote the values in $\{0,1\}^z$ (for $z \in \mathbb{N}$) by integers. For any set $S \subseteq [N]$, let $\mathbb{I}_S$ be a bit vector of length N, where the i^{th} element $\mathbb{I}_S(i)$ is defined as

$$\mathbb{I}_S(i) = \begin{cases} 1 & \text{if } i \in S, \\ 0 & \text{otherwise} \end{cases}$$

.

- $\mathsf{Setup}(1^\lambda, 1^N)$: Sample $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Setup}_{\mathsf{ABO}}(1^\lambda)$ and $\mathsf{hk} \leftarrow \mathsf{Setup}_{\mathsf{SPB}}(1^\lambda, N)$. Compute signatures $\{\sigma_i \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, i) : 1 \le i \le N\}$. Output $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$, and secret keys $\{\mathsf{sk}_i = \sigma_i : 1 \le i \le N\}$.
- $\mathsf{Encrypt}(\mathsf{pk}, S, m, \mathsf{ind})$: Let $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$. Compute SPB hash on $\mathbb{I}_S$ i.e., compute the hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$. Then encrypt the message m with PWE scheme using the instance $\mathsf{inst} = (1^\lambda, N, h, \mathsf{hk}, \mathsf{vk})$ and index $\mathsf{ind}||0^{k+\ell}$, i.e., computes $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}\ (\mathsf{inst}, m, \mathsf{ind}||0^{k+\ell})$.
- $\mathsf{Decrypt}(i, \mathsf{sk}_i, \mathsf{pk}, S, \mathsf{ct})$: Let $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$. Compute hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$ and proof $\pi_i \leftarrow \mathsf{Open}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S, i)$. Then decrypt the ciphertext using the witness $w = i||\mathsf{sk}_i||\pi_i$ i.e., output message $m \leftarrow \mathsf{Decrypt}_{\mathsf{PWE}}(w = i||\mathsf{sk}_i||\pi_i, \mathsf{ct})$.

Note that the correctness properties of $\mathcal{SPB}$ hash and $\mathcal{ABO}$ signature schemes imply that $w = i||\mathsf{sk}_i||\pi_i$ is a valid witness to the instance $\mathsf{inst} = (1^\lambda, N, h, \mathsf{hk}, \mathsf{vk})$ (i.e., $R(x, w) = 1$). This along with the correctness of $\mathcal{PWE}$ scheme imply the correctness of the above scheme.

Note that the length of the hash key hk is $\mathsf{poly}_1(\lambda, \log N)$ and verification key vk is $\mathsf{poly}_2(\lambda)$ for some polynomials $\mathsf{poly}_1(\cdot)$ and $\mathsf{poly}_2(\cdot)$. This implies lengths of instance inst and witness $w = i||\mathsf{sk}_i||\pi_i$ are at most $\mathsf{poly}_3(\lambda, \log N)$ for some polynomial $\mathsf{poly}_3(\cdot)$. Therefore, the ciphertext length of the above scheme is also at most $\mathsf{poly}(\lambda, \log N)$ for some polynomial $\mathsf{poly}(\cdot)$. In the following subsections, we prove that the above AugBE construction satisfies message hiding and index hiding properties. Formally, we prove the following theorem.

Theorem 4.1. If $\mathcal{PWE}$ is a sub-exponentially secure PWE scheme as per Definitions 2.1 and 2.2, $\mathcal{ABO}$ is a secure ABO signature scheme as per Definition 2.3 and $\mathcal{SPB}$ is a secure SPB hash function as per Definitions 2.4 and 2.5, then $\mathcal{AUGBE}$ is a secure AugBE scheme as per Definitions 3.3 and 3.4.

4.1 Message Hiding

In this subsection, we prove the message hiding property of the above scheme assuming sub-exponential security of $\mathcal{PWE}$ scheme. For any instance $(1^\lambda, N, h, \mathsf{hk}, \mathsf{vk})$, let $r(\lambda)$ be the length of witnesses accepted by the witness relation $\mathcal{R}$, i.e., $r(\lambda) = \lambda + k(\lambda) + \ell(\lambda)$. For simplicity of notation, we ignore the parameters and simply denote it by r. We first describe the following games that help us in proving the property.

$\mathsf{Game}\ N+1||0^{k+\ell}$. This game is same as the AugBE message hiding game.

1. *Setup Phase.* The adversary $\mathcal{A}$ sends the number of users 1^N to the challenger. The challenger samples the keys $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Setup}_{\mathsf{ABO}}(1^\lambda)$, hash key $\mathsf{hk} \leftarrow \mathsf{Setup}_{\mathsf{SPB}}(1^\lambda, N)$ and signatures $\{\sigma_i \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, i) : 1 \leq i \leq N\}$. It then sends the public key $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$.

2. *Pre-Challenge Query Phase.* The adversary then adaptively queries for secret keys. For each query j, the challenger responds with the secret key $\mathsf{sk}_j = \sigma_j$.

3. *Challenge Phase.* The adversary then sends a pair of messages m_0, m_1 and a set $S \subseteq [N]$ to the challenger. The challenger samples a bit $b \leftarrow \{0,1\}$ and computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$. It then samples ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}\ (x = (1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}), m_b, \mathsf{int}(N+1||0^{k+\ell}))$ and responds with ct.

4. *Post-Challenge Query Phase.* This is identical to Pre-Challenge Query Phase.

5. *Output Phase.* The adversary sends a bit b' to the challenger. The adversary wins if $b' = b$.

$\mathsf{Game}\ y\ (N+1||0^{k+\ell} < y \leq 2^r)$. This game is similar to $\mathsf{Game}\ N+1||0^{k+\ell}$, except that the challenger encrypts the challenge message using index y instead of index $\mathsf{int}(N+1||0^{k+\ell})$.

3. *Challenge Phase.* The adversary then sends a pair of messages m_0, m_1 and a set $S \subseteq [N]$ to the challenger. The challenger samples a bit $b \leftarrow \{0,1\}$ and computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$. It then samples ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}\ (x = (1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}), m_b, y)$ and responds with ct.

For any stateful PPT adversary $\mathcal{A}$, we define the advantage of the adversary in $\mathsf{Game}\ x$ as $\mathsf{Adv}_x^{\mathcal{A}}(\lambda) = \Pr[\mathcal{A} \text{ wins}] - 1/2$. We prove that the advantage of any PPT adversary $\mathcal{A}$ in $\mathsf{Game}\ N+1||0^{k+\ell}$ is negligible in security parameter. For any stateful PPT adversary $\mathcal{B}$ and $\lambda \in \mathbb{N}$, let $\mathsf{AdvPosInd}^{\mathcal{B}}(\lambda)$ denote the advantage of $\mathcal{B}$ in position indistinguishability game and $\mathsf{AdvMsgInd}^{\mathcal{B}}(\lambda)$ denote the advantage of $\mathcal{B}$ in message indistinguishability game of $\mathcal{PWE}$ scheme. For any $\lambda \in \mathbb{N}$, let $\mathsf{AdvPosInd}(\lambda) = \sup_{\mathrm{PPT}\ \mathcal{B}} \mathsf{AdvPosInd}^{\mathcal{B}}(\lambda)$ and $\mathsf{AdvMsgInd}(\lambda) = \sup_{\mathrm{PPT}\ \mathcal{B}} \mathsf{AdvMsgInd}^{\mathcal{B}}(\lambda)$. We now establish using the following lemma that the difference of the adversary's advantage between each adjacent game is at most $2 \cdot \mathsf{AdvPosInd}(\lambda)$. Finally we show that if any adversary wins in the last game, then it wins message indistinguishability game against PWE challenger as well.

Claim 4.1. For every y s.t. $N+1||0^{k+\ell} \leq y \leq 2^r - 1$, every PPT adversary $\mathcal{A}$ and $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}_y^{\mathcal{A}}(\lambda) - \mathsf{Adv}_{y+1}^{\mathcal{A}}(\lambda) \leq 2 \cdot \mathsf{AdvPosInd}(\lambda)$.

Proof. Consider any y s.t. $N+1||0^{k+\ell} \leq y \leq 2^r - 1$, any PPT adversary $\mathcal{A}$ and $\lambda \in \mathbb{N}$. We build a PPT algorithm $\mathcal{B}$ which uses $\mathcal{A}$ and has advantage $(\mathsf{Adv}_y^{\mathcal{A}}(\lambda) - \mathsf{Adv}_{y+1}^{\mathcal{A}}(\lambda))/2$ in the position indistinguishability game of the $\mathcal{PWE}$ scheme. The reduction algorithm $\mathcal{B}$ proceeds as follows.

$\mathcal{A}$ first sends the number of users 1^N to $\mathcal{B}$. $\mathcal{B}$ then samples $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup}_{\mathsf{ABO}}(1^\lambda)$, $\mathsf{hk} \leftarrow \mathsf{Setup}_{\mathsf{SPB}}(1^\lambda, N)$, signatures $\{\sigma_j \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j) : 1 \leq j \leq N\}$ and sends the public key $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$. $\mathcal{A}$ then adaptively queries for secret keys. For each query j, $\mathcal{B}$ responds with the secret key $\mathsf{sk}_j = \sigma_j$. After query phase, $\mathcal{A}$ sends a challenge set S and a pair of messages m_0, m_1 to $\mathcal{B}$. $\mathcal{B}$ samples a bit $b \leftarrow \{0,1\}$ and computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$. It then sends the challenge instance $\mathsf{inst} = (1^\lambda, N, h, \mathsf{hk}, \mathsf{vk})$, challenge message m_b and challenge index y to the challenger $\mathcal{C}$ of position indistinguishability game. The challenger samples a bit $\beta \leftarrow \{0,1\}$ and responds with a ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}(\mathsf{inst}, m_b, y + \beta)$ to $\mathcal{B}$, which forwards it to $\mathcal{A}$. $\mathcal{A}$ further adaptively queries for secret keys. For each query j, $\mathcal{B}$ responds with the secret key $\mathsf{sk}_j = \sigma_j$. Finally, $\mathcal{A}$ sends a bit b' to $\mathcal{B}$. If $b' = b$, then $\mathcal{B}$ outputs 0 indicating its guess that the challenger encrypted m_b using index y. If $b' \neq b$, then $\mathcal{B}$ outputs 1 indicating its guess that the challenger encrypted m_b using index $y+1$.

We know that the index y cannot be a witness for the instance $(1^\lambda, N, h, \mathsf{hk}, \mathsf{vk})$ as $y \geq N+1||0^{k+\ell}$ (i.e., $y[1:\lambda] \geq N+1$). Therefore, the reduction algorithm $\mathcal{B}$ acts as a valid adversary in the position indistinguishability game. If $\beta = 0$, $\mathcal{B}$ simulates the view of $\mathsf{Game}\ y$ to $\mathcal{A}$ and $\Pr[b' = b] = 1/2 + \mathsf{Adv}_y^{\mathcal{A}}(\lambda)$. Otherwise, it simulates the view of $\mathsf{Game}\ y+1$ to $\mathcal{A}$ and $\Pr[b' = b] = 1/2 + \mathsf{Adv}_{y+1}^{\mathcal{A}}(\lambda)$. Therefore, the advantage of $\mathcal{B}$ in position indistinguishability game is given by $\mathsf{AdvPosInd}^{\mathcal{B}}(\lambda) = 1/2 \cdot \Pr[b' = b|\beta = 0] + 1/2 \cdot \Pr[b' \neq b|\beta = 1] - 1/2 = 1/2 \cdot (\mathsf{Adv}_y^{\mathcal{A}}(\lambda) - \mathsf{Adv}_{y+1}^{\mathcal{A}}(\lambda))$. Therefore, $\mathsf{Adv}_y^{\mathcal{A}}(\lambda) - \mathsf{Adv}_{y+1}^{\mathcal{A}}(\lambda) \leq 2 \cdot \mathsf{AdvPosInd}(\lambda)$. ■

Claim 4.2. For every stateful PPT adversary $\mathcal{A}$ and every $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}_{2^r}^{\mathcal{A}}(\lambda) \leq \mathsf{AdvMsgInd}(\lambda)$.

Proof. Consider any PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$. We build a PPT algorithm $\mathcal{B}$ which uses $\mathcal{A}$ and has advantage $\mathsf{Adv}_{2^r}^{\mathcal{A}}(\lambda)$ in message indistinguishability game of the $\mathcal{PWE}$ scheme. The reduction algorithm $\mathcal{B}$ proceeds as follows.

$\mathcal{A}$ first sends the number of users 1^N to $\mathcal{B}$. $\mathcal{B}$ then samples $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup}_{\mathsf{ABO}}(1^\lambda)$, hash key $\mathsf{hk} \leftarrow \mathsf{Setup}_{\mathsf{SPB}}(1^\lambda, N)$ and signatures $\{\sigma_j \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j) : 1 \leq j \leq N\}$. It then sends the public key $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$. $\mathcal{A}$ then adaptively queries for secret keys. For each query j, $\mathcal{B}$ responds with the secret key $\mathsf{sk}_j = \sigma_j$. After query phase, $\mathcal{A}$ sends a challenge set S and messages m_0, m_1 to $\mathcal{B}$. $\mathcal{B}$ computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$. It then sends challenge instance $\mathsf{inst} = (1^\lambda, N, h, \mathsf{hk}, \mathsf{vk})$ and challenge messages m_0, m_1 to message indistinguishability game challenger $\mathcal{C}$. The challenger samples a bit $b \leftarrow \{0, 1\}$ and responds with ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}(\mathsf{inst}, m_b, 2^r)$ to $\mathcal{B}$, which forwards ct to $\mathcal{A}$. $\mathcal{A}$ further adaptively queries for secret keys. For each query j, $\mathcal{B}$ responds with the secret key $\mathsf{sk}_j = \sigma_j$. Finally, $\mathcal{A}$ sends a bit b' to $\mathcal{B}$, which outputs b' as its guess in message indistinguishability game.

Clearly, $\mathcal{B}$ is a valid adversary of the message indistinguishability game, and also simulates the view of $\mathsf{Game}\ 2^r$ to $\mathcal{A}$. Note that advantage of $\mathcal{B}$ in message indistinguishability game is given by $\mathsf{AdvMsgInd}^{\mathcal{B}}(\lambda) = \mathsf{Adv}_{2^r}^{\mathcal{A}}(\lambda)$, and therefore $\mathsf{Adv}_{2^r}^{\mathcal{A}}(\lambda) \leq \mathsf{AdvMsgInd}(\lambda)$. ■

Note that by combining claims 4.1 and 4.2, the advantage of any PPT adversary $\mathcal{A}$ in AugBE message hiding game is $\mathsf{Adv}_{N+1||0^{k+\ell}}^{\mathcal{A}}(\lambda) = \sum_{y=N+1||0^{k+\ell}}^{2^r-1}(\mathsf{Adv}_y^{\mathcal{A}}(\lambda) - \mathsf{Adv}_{y+1}^{\mathcal{A}}(\lambda)) + \mathsf{Adv}_{2^r}^{\mathcal{A}}(\lambda) \leq 2 \cdot (2^\lambda - N) \cdot 2^{k+\ell} \cdot \mathsf{AdvPosInd}(\lambda) + \mathsf{AdvMsgInd}(\lambda)$. Using complexity leveraging, we demand that $\mathsf{AdvPosInd}(\lambda) \leq 2^{-(\lambda+k+\ell+1)} \cdot \mathrm{negl}(\lambda)$ for some negligible function $\mathrm{negl}(\cdot)$. At the instantiation level, the security parameter will be increased to match this condition.

4.2 Index Hiding

In this section, we prove the index hiding property of the above scheme. We first describe the following 2 games that help us in describing the lemma formally.

$\mathsf{Game}\ 0$. This game corresponds to AugBE index hiding game where the challenger always uses bit $b = 0$.

1. *Setup Phase.* The adversary $\mathcal{A}$ sends the number of users 1^N and index i s.t. $1 \leq i \leq N$ to the challenger. The challenger samples $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Setup}_{\mathsf{ABO}}(1^\lambda)$, hash key $\mathsf{hk} \leftarrow \mathsf{Setup}_{\mathsf{SPB}}(1^\lambda, N)$ and signatures $\{\sigma_j \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j) : 1 \leq j \leq N\}$ of the AugBE scheme. It then sends the public key $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$.

2. *Pre-Challenge Query Phase.* The adversary then adaptively queries for secret keys. For each query j, the challenger responds with the secret key $\mathsf{sk}_j = \sigma_j$.

3. *Challenge Phase.* The adversary then sends a message m and a set $S \subseteq [N]$ to the challenger. The challenger computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$ and responds with ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}$ $(x = (1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}), m, \mathsf{int}(i||0^{k+\ell}))$.

4. *Post-Challenge Query Phase.* This is identical to Pre-Challenge Query Phase.

5. *Output Phase.* The adversary sends a bit b' to the challenger.

Let the set of all secret keys queried by the adversary be S^*. The adversary is restricted to query such that $i \notin S \vee i \notin S^*$.

$\mathsf{Game}\ 3$. This game is similar to the first game, except that the challenger always uses bit $b = 1$.

3. *Challenge Phase.* The adversary then sends a message m and a set $S \subseteq [N]$ to the challenger. The challenger computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$ and responds with ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}(x = (1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}), m, \mathsf{int}(i + 1||0^{k+\ell}))$.

For any stateful PPT adversary $\mathcal{A}$, let the probability that $\mathcal{A}$ outputs 1 in $\mathsf{Game}\ y$ be $p_y^{\mathcal{A}}(\lambda)$. We denote the advantage of a PPT adversary $\mathcal{A}$ in distinguishing between any two games $\mathsf{Game}\ x$ and $\mathsf{Game}\ y$ by $\mathsf{Adv}_{x,y}^{\mathcal{A}}(\lambda) = |p_x^{\mathcal{A}}(\lambda) - p_y^{\mathcal{A}}(\lambda)|$.

Lemma 4.1. If $\mathcal{ABO}$ is a secure ABO signature scheme as per Definition 2.3, $\mathcal{SPB}$ is a secure SPB hash function as per Definitions 2.4 and 2.5, and $\mathcal{PWE}$ is a sub-exponentially secure PWE scheme as per Definition 2.2, for every stateful PPT Adversary $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that for every security parameter λ, $\mathsf{Adv}_{0,3}^{\mathcal{A}}(\lambda) \leq \text{negl}(\lambda)$.

Proof. We first classify the adversaries into the following 2 types.

- $\mathsf{Type}\ 1$ adversary: Restricted to generate set of key queries S^* and challenge set S s.t. $i \notin S$.
- $\mathsf{Type}\ 2$ adversary: Restricted to generate set of key queries S^* and challenge set S s.t. $i \in S \wedge i \notin S^*$.

We now prove Lemma 4.2 and 4.3 which together imply Lemma 4.1.

Lemma 4.2. If $\mathcal{SPB}$ is secure as per Definitions 2.4 and 2.5, and $\mathcal{PWE}$ is a sub-exponentially secure as per Definition 2.2, for every stateful $\mathsf{Type}\ 1$ PPT Adversary $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that for every security parameter λ, $\mathsf{Adv}_{0,3}^{\mathcal{A}}(\lambda) \leq \text{negl}(\lambda)$.

Proof. We prove the lemma using the following sequence of hybrids.

$\mathsf{Game}\ 1.t$ (for $0 \leq t < 2^{k+\ell}$): Here t is a bit string of length $k + \ell$. This game is similar to $\mathsf{Game}\ 0$ except that challenger samples SPB hash key using $\mathsf{Setup\text{-}Bind}$ and encrypts the challenge message using index $\mathsf{int}(i||0^{k+\ell}) + t$.

1. *Setup Phase.* The adversary $\mathcal{A}$ sends the number of users 1^N to the challenger. The challenger samples $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Setup}_{\mathsf{ABO}}(1^\lambda)$, $\mathsf{hk} \leftarrow \mathsf{Setup\text{-}Bind}_{\mathsf{SPB}}(1^\lambda, N, i)$ and signatures $\{\sigma_j \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j) : 1 \leq j \leq N\}$. It then sends the public key $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$.

3. *Challenge Phase.* The adversary then sends a message m and a set $S \subseteq [N]$ to the challenger. The challenger computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$ and responds with ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}((1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}), m, \mathsf{int}(i||0^{k+\ell}) + t)$.

$\mathsf{Game}\ 1.2^{k+\ell}$: This game is similar to $\mathsf{Game}\ 1.2^{k+\ell} - 1$ except that challenger encrypts the challenge message using index $\mathsf{int}(i + 1||0^{k+\ell})$.

1. *Setup Phase.* The adversary $\mathcal{A}$ sends the number of users 1^N to the challenger. The challenger samples $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Setup}_{\mathsf{ABO}}(1^\lambda)$, hash key $\mathsf{hk} \leftarrow \mathsf{Setup\text{-}Bind}_{\mathsf{SPB}}(1^\lambda, N, i)$ and signatures $\{\sigma_j \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j) : 1 \leq j \leq N\}$ of the AugBE scheme. It then sends the public key $\mathsf{pk} = (\mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$.

3. *Challenge Phase.* The adversary then sends a message m and a set $S \subseteq [N]$ to the challenger, which computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$ and responds with ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}((1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}), m, \mathsf{int}(i + 1||0^{k+\ell}))$.

For any PPT adversary $\mathcal{B}$ and $\lambda \in \mathbb{N}$, let $\mathsf{AdvSpbInd}^{\mathcal{B}}(\lambda)$ denote the advantage of $\mathcal{B}$ in index hiding game of $\mathcal{SPB}$ scheme and $\mathsf{AdvPosInd}^{\mathcal{B}}(\lambda)$ denote the advantage of $\mathcal{B}$ in position indistinguishability game of $\mathcal{PWE}$ scheme. For any $\lambda \in \mathbb{N}$, let $\mathsf{AdvPosInd}(\lambda) = \sup_{\text{PPT}\ \mathcal{B}} \mathsf{AdvPosInd}^{\mathcal{B}}(\lambda)$ and $\mathsf{AdvSpbInd}(\lambda) = \sup_{\text{PPT}\ \mathcal{B}} \mathsf{AdvSpbInd}^{\mathcal{B}}(\lambda)$. We prove Lemma 4.2 using the following sequence of claims.

Claim 4.3. For every Type 1 PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}^{\mathcal{A}}_{0,1.0}(\lambda) \leq 2 \cdot \mathsf{AdvSpbInd}(\lambda)$.

Proof. Consider any Type 1 PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$. We build a PPT algorithm $\mathcal{B}$ which uses $\mathcal{A}$ and has advantage $\mathsf{Adv}^{\mathcal{A}}_{0,1.0}(\lambda)/2$ in index hiding game of the $\mathcal{SPB}$ scheme. The reduction algorithm $\mathcal{B}$ proceeds as follows.

$\mathcal{A}$ first sends the number of users 1^N and an index i s.t. $1 \leq i \leq N$ to $\mathcal{B}$. $\mathcal{B}$ then sends (N, i) to index hiding game challenger $\mathcal{C}$. The challenger samples a bit $b \leftarrow \{0, 1\}$. If $b = 0$, it responds with $\mathsf{hk} \leftarrow \mathsf{Setup}_{\mathsf{SPB}}(1^\lambda, N)$. Otherwise, it responds with $\mathsf{hk} \leftarrow \mathsf{Setup\text{-}Bind}_{\mathsf{SPB}}(1^\lambda, N, i)$. $\mathcal{B}$ samples $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup}_{\mathsf{ABO}}(1^\lambda)$, signatures $\{\sigma_j \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j) : 1 \leq j \leq N\}$ and sends the public key $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$. $\mathcal{A}$ then adaptively queries for secret keys. For each query j, $\mathcal{B}$ responds with secret key $\mathsf{sk}_j = \sigma_j$. After query phase, $\mathcal{A}$ sends a challenge set S and a message m to $\mathcal{B}$. $\mathcal{B}$ aborts if $i \in S$. Otherwise, it computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$ and responds with ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}\ ((1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}),\ m, \mathsf{int}(i||0^{k+\ell}))$. $\mathcal{A}$ further adaptively queries for secret keys. For each query j, $\mathcal{B}$ responds with secret key $\mathsf{sk}_j = \sigma_j$. Finally, $\mathcal{A}$ sends a bit b' to $\mathcal{B}$, which outputs b' as its guess in the index hiding game.

As $\mathcal{A}$ is a Type 1 adversary, note that $i \notin S$ and $\mathcal{B}$ does not abort. Note that if $b = 0$, $\mathcal{B}$ simulates the view of Game 0 to $\mathcal{A}$ and $\Pr[b' = 1] = p^{\mathcal{A}}_0(\lambda)$. Otherwise, it simulates the view of Game 1.0 to $\mathcal{A}$ and $\Pr[b' = 1] = p^{\mathcal{A}}_{1.0}(\lambda)$. This implies, the advantage of $\mathcal{B}$ in the index hiding game is given by $\mathsf{AdvSpbInd}^{\mathcal{B}}(\lambda) = |1/2 \cdot \Pr[b' = 0|b = 0] + 1/2 \cdot \Pr[b' = 1|b = 1] - 1/2| = \mathsf{Adv}^{\mathcal{A}}_{0,1.0}(\lambda)/2$. Therefore, $\mathsf{Adv}^{\mathcal{A}}_{0,1.0}(\lambda) \leq 2 \cdot \mathsf{AdvSpbInd}(\lambda)$. ■

Claim 4.4. Assuming $\mathcal{SPB}$ is somewhere perfectly binding w.r.t. opening, for any $0 \leq t \leq 2^{k+\ell} - 1$, any stateful Type 1 PPT Adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}^{\mathcal{A}}_{1.t,1.t+1}(\lambda) \leq 2 \cdot \mathsf{AdvPosInd}(\lambda)$.

Proof. Consider any t s.t. $0 \leq t \leq 2^{k+\ell} - 1$, any Type 1 PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$. Assuming $\mathcal{SPB}$ is somewhere perfectly binding w.r.t. opening, we build a PPT algorithm $\mathcal{B}$ which uses $\mathcal{A}$ and has advantage $\mathsf{Adv}^{\mathcal{A}}_{1.t,1.t+1}(\lambda)/2$ in position indistinguishability game of the $\mathcal{PWE}$ scheme. The reduction algorithm $\mathcal{B}$ proceeds as follows.

$\mathcal{A}$ first sends the number of users 1^N and an index i s.t. $1 \leq i \leq N$ to $\mathcal{B}$. $\mathcal{B}$ then samples $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup}_{\mathsf{ABO}}(1^\lambda)$, hash key $\mathsf{hk} \leftarrow \mathsf{Setup\text{-}Bind}_{\mathsf{SPB}}(1^\lambda, N, i)$ and signatures $\{\sigma_j \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j) : 1 \leq j \leq N\}$. It then sends the public key $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$. $\mathcal{A}$ then adaptively queries for secret keys. For each query j, $\mathcal{B}$ responds with secret key $\mathsf{sk}_j = \sigma_j$. After query phase, $\mathcal{A}$ sends a challenge set S and a message m to $\mathcal{B}$. $\mathcal{B}$ aborts if $i \in S$. Otherwise, it computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$, and sends the challenge instance $\mathsf{inst} = (1^\lambda, N, h, \mathsf{hk}, \mathsf{vk})$, challenge message m and challenge index $\mathsf{int}(i||0^{k+\ell}) + t$ to the position indistinguishability game challenger $\mathcal{C}$. The challenger samples a bit $\beta \leftarrow \{0, 1\}$ and responds with a ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}(\mathsf{inst}, m, \mathsf{int}(i||0^{k+\ell}) + t + \beta)$[13] to $\mathcal{B}$. $\mathcal{B}$ forwards the ciphertext to $\mathcal{A}$. $\mathcal{A}$ further adaptively queries for secret keys. For each query j, $\mathcal{B}$ responds with secret key $\mathsf{sk}_j = \sigma_j$. Finally, $\mathcal{A}$ sends a bit b' to $\mathcal{B}$, which outputs b' as its guess in the position indistinguishability game.

As $\mathcal{A}$ is a Type 1 adversary, note that $\mathbb{I}_S(i) = 0$ and $\mathcal{B}$ does not abort. We know that, $\Pr[\mathsf{hk}$ is binding w.r.t. opening at index $i] = 1$. This implies that there does not exist a proof π such that $\mathsf{Verify}_{\mathsf{SPB}}(\mathsf{hk}, h, i, 1, \pi) = 1$ and $\mathsf{int}(i||0^{\kappa+\ell}) + t$ cannot be a witness of the instance $(1^\lambda, N, h, \mathsf{hk}, \mathsf{vk})$. Therefore, $\mathcal{B}$ acts as a valid adversary of the position indistinguishability game. If $\beta = 0$, $\mathcal{B}$ simulates the view of Game $1.t$ to $\mathcal{A}$ and $\Pr[b' = 1] = p^{\mathcal{A}}_{1.t}(\lambda)$. Otherwise, it simulates the view of Game $1.t + 1$ to $\mathcal{A}$ and $\Pr[b' = 1] = p^{\mathcal{A}}_{1.t+1}(\lambda)$. This implies, the advantage of $\mathcal{B}$ in the position indistinguishability game is given by $\mathsf{AdvPosInd}^{\mathcal{B}}(\lambda) = |1/2 \cdot \Pr[b' = 0|\beta = 0] + 1/2 \cdot \Pr[b' = 1|\beta = 1] - 1/2| = \mathsf{Adv}^{\mathcal{A}}_{1.t,1.t+1}(\lambda)/2$. Therefore, $\mathsf{Adv}^{\mathcal{A}}_{1.t,1.t+1}(\lambda) \leq 2 \cdot \mathsf{AdvPosInd}(\lambda)$. ■

Claim 4.5. For every Type 1 PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}^{\mathcal{A}}_{1.2^{k+\ell},3}(\lambda) \leq 2 \cdot \mathsf{AdvSpbInd}(\lambda)$.

Proof. Consider any Type 1 PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$. We build a PPT algorithm $\mathcal{B}$ which uses $\mathcal{A}$ and has advantage $\mathsf{Adv}^{\mathcal{A}}_{1.2^{k+\ell},3}(\lambda)/2$ in the index hiding game of the $\mathcal{SPB}$ scheme. We ignore the description of algorithm $\mathcal{B}$ as it proceeds similar to proof of Claim 4.3.

[13] Note that index $\mathsf{int}(i||0^{k+\ell}) + 2^{k+\ell}$ is same as $\mathsf{int}(i + 1||0^{k+\ell})$.

■

Note that by triangle inequality and combining Claims 4.3, 4.4, and 4.5, the advantage of any Type 1 PPT adversary $\mathcal{A}$ in AugBE index hiding game is $\mathsf{Adv}^{\mathcal{A}}_{0,3}(\lambda) \leq \mathsf{Adv}^{\mathcal{A}}_{0,1.0} + \sum_{t=0}^{2^{k+\ell}-1} \mathsf{Adv}^{\mathcal{A}}_{1.t,1.t+1} + \mathsf{Adv}^{\mathcal{A}}_{1.2^{k+\ell},3} \leq 4 \cdot \mathsf{AdvSpbInd}(\lambda) + 2 \cdot 2^{k+\ell} \cdot \mathsf{AdvPosInd}(\lambda)$. Using complexity leveraging, we demand that $\mathsf{AdvPosInd}(\lambda) \leq 2^{-(k+\ell+1)} \cdot \mathrm{negl}(\lambda)$ for some negligible function $\mathrm{negl}(\cdot)$. At the instantiation level, the security parameter will be increased to match these conditions.

■

Lemma 4.3. If $\mathcal{ABO}$ is a secure ABO signature scheme as per Definition 2.3 and $\mathcal{PWE}$ is a sub-exponentially secure as per Definition 2.2, for every stateful Type 2 PPT Adversary $\mathcal{A}$, there exists a negligible function $\mathrm{negl}(\cdot)$ such that for every security parameter λ, $\mathsf{Adv}^{\mathcal{A}}_{0,3}(\lambda) \leq \mathrm{negl}(\lambda)$.

Proof. We prove the lemma using the following sequence of hybrids.

$\mathsf{Game}\ 2.t$ (for $0 \leq t < 2^{k+\ell}$): Here t is a bit string of length $k+\ell$. This game is similar to $\mathsf{Game}\ 0$ except, the challenger samples ABO signature verification key using $\mathsf{Setup\text{-}Punc}$ algorithm and encrypts challenge message using index $\mathsf{int}(i||0^{k+\ell}) + t$.

1. *Setup Phase.* The adversary $\mathcal{A}$ sends the number of users 1^N to the challenger. The challenger samples $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Setup\text{-}Punc}_{\mathsf{ABO}}(1^\lambda, i)$, $\mathsf{hk} \leftarrow \mathsf{Setup}_{\mathsf{SPB}}(1^\lambda, N, i)$ and signatures $\{\sigma_j \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j) : 1 \leq j \leq N\}$. It then sends the public key $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$.

3. *Challenge Phase.* The adversary then sends a message m and a set $S \subseteq [N]$ to the challenger. The challenger computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$ and responds with ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}((1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}), m, \mathsf{int}(i||0^{k+\ell}) + t)$.

$\mathsf{Game}\ 2.2^{k+\ell}$: This game is similar to $\mathsf{Game}\ 2.2^{k+\ell} - 1$ except that the challenger encrypts the challenge message using index $\mathsf{int}(i + 1||0^{k+\ell})$.

1. *Setup Phase.* The adversary $\mathcal{A}$ sends the number of users 1^N to the challenger. The challenger samples $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Setup\text{-}Punc}_{\mathsf{ABO}}(1^\lambda, i)$, hash key $\mathsf{hk} \leftarrow \mathsf{Setup}_{\mathsf{SPB}}(1^\lambda, N, i)$ and signatures $\{\sigma_j \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j) : 1 \leq j \leq N\}$ of the AugBE scheme. It then sends the public key $\mathsf{pk} = (\mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$.

3. *Challenge Phase.* The adversary then sends a message m and a set $S \subseteq [N]$ to the challenger, which computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$ and responds with ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}((1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}), m, \mathsf{int}(i + 1||0^{k+\ell}))$.

For any PPT adversary $\mathcal{B}$ and $\lambda \in \mathbb{N}$, let $\mathsf{AdvAboInd}^{\mathcal{B}}(\lambda)$ denote the advantage of $\mathcal{B}$ in VK indistinguishability game of $\mathcal{ABO}$ scheme and $\mathsf{AdvPosInd}^{\mathcal{B}}(\lambda)$ denote the advantage of $\mathcal{B}$ in position indistinguishability game of $\mathcal{PWE}$ scheme. For any $\lambda \in \mathbb{N}$, let $\mathsf{AdvPosInd}(\lambda) = \sup_{\mathrm{PPT}\ \mathcal{B}} \mathsf{AdvPosInd}^{\mathcal{B}}(\lambda)$ and $\mathsf{AdvAboInd}(\lambda) = \sup_{\mathrm{PPT}\ \mathcal{B}} \mathsf{AdvAboInd}^{\mathcal{B}}(\lambda)$. We prove Lemma 4.3 using the following sequence of claims.

Claim 4.6. For every Type 2 PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}^{\mathcal{A}}_{0,2.0}(\lambda) \leq 2 \cdot \mathsf{AdvAboInd}(\lambda)$.

Proof. Consider any Type 2 PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$. We build a PPT algorithm $\mathcal{B}$ which uses $\mathcal{A}$ and has advantage $\mathsf{Adv}^{\mathcal{A}}_{0,2.0}(\lambda)$ in VK indistinguishability game of the $\mathcal{ABO}$ scheme. The reduction algorithm $\mathcal{B}$ proceeds as follows.

$\mathcal{A}$ first sends the number of users 1^N and an index i s.t. $1 \leq i \leq N$ to $\mathcal{B}$. $\mathcal{B}$ sends challenge message i to VK indistinguishability game challenger $\mathcal{C}$. The challenger samples a bit $b \leftarrow \{0, 1\}$. If $b = 0$, it samples $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup}_{\mathsf{ABO}}(1^\lambda)$. Otherwise, it samples $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup\text{-}Punc}_{\mathsf{ABO}}(1^\lambda, i)$. It then sends vk to $\mathcal{B}$. $\mathcal{B}$ samples $\mathsf{hk} \leftarrow \mathsf{Setup}_{\mathsf{SPB}}(1^\lambda, N)$ and sends the public key $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$. $\mathcal{A}$ then adaptively queries for secret keys. For each query j, $\mathcal{B}$ aborts if $j = i$. Otherwise, it forwards the query to $\mathcal{C}$, which responds with $\sigma \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j)$. $\mathcal{B}$ forwards the reply to $\mathcal{A}$. After query phase, $\mathcal{A}$ sends a challenge set S and a message m to $\mathcal{B}$. $\mathcal{B}$ computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$ and responds with ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}$ $(x = (1^\lambda, N, h, \mathsf{hk}, \mathsf{vk}), m, \mathsf{int}(i||0^{k+\ell}))$. $\mathcal{A}$ then adaptively queries for secret keys. For each query j, $\mathcal{B}$ aborts if $j = i$. Otherwise, it forwards the query to $\mathcal{C}$, which responds with $\sigma \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j)$. $\mathcal{B}$ forwards the reply to $\mathcal{A}$. Finally, $\mathcal{A}$ sends a bit b' to $\mathcal{B}$, which outputs b' as its guess in the VK indistinguishability game.

As $\mathcal{A}$ is a Type 2 adversary, it does not query for secret key sk_i and therefore, $\mathcal{B}$ does not abort and acts as a valid adversary of the VK indistinguishability game. If $b = 0$, then $\mathcal{B}$ simulates the view of Game 0 to $\mathcal{A}$ and $\Pr[b' = 1] = p_0^{\mathcal{A}}(\lambda)$. Otherwise, it simulates the view of Game 2.0 to $\mathcal{A}$ and $\Pr[b' = 1] = p_{2.0}^{\mathcal{A}}(\lambda)$. This implies, the advantage of $\mathcal{B}$ in the index hiding game is given by $\mathsf{AdvSpbInd}^{\mathcal{B}}(\lambda) = |1/2 \cdot \Pr[b' = 0|b = 0] + 1/2 \cdot \Pr[b' = 1|b = 1] - 1/2| = \mathsf{Adv}_{0,2.0}^{\mathcal{A}}(\lambda)/2$. Therefore, $\mathsf{Adv}_{0,2.0}^{\mathcal{A}}(\lambda) \leq 2 \cdot \mathsf{AdvAboInd}(\lambda)$. $\blacksquare$

Claim 4.7. For every t s.t. $0 \leq t \leq 2^{k+\ell} - 1$, every Type 2 PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}_{2.t,2.t+1}^{\mathcal{A}}(\lambda) \leq 2 \cdot \mathsf{AdvPosInd}(\lambda)$.

Proof. Consider any t s.t. $0 \leq t \leq 2^{k+\ell} - 1$, a Type 2 PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$. We build a PPT algorithm $\mathcal{B}$ which uses $\mathcal{A}$ and has advantage $\mathsf{Adv}_{2.t,2.t+1}^{\mathcal{A}}(\lambda)$ in position indistinguishability game of the $\mathcal{PWE}$ scheme. The reduction algorithm $\mathcal{B}$ proceeds as follows.

$\mathcal{A}$ first sends the number of users 1^N and an index i s.t. $1 \leq i \leq N$ to $\mathcal{B}$. $\mathcal{B}$ then samples $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup\text{-}Punc}_{\mathsf{ABO}}(1^\lambda, i)$, hash key $\mathsf{hk} \leftarrow \mathsf{Setup}_{\mathsf{SPB}}(1^\lambda, N)$ and signatures $\{\sigma_j \leftarrow \mathsf{Sign}_{\mathsf{ABO}}(\mathsf{sk}, j) : 1 \leq j \leq N, j \neq i\}$. It then sends the public key $\mathsf{pk} = (1^\lambda, N, \mathsf{vk}, \mathsf{hk})$ to $\mathcal{A}$. $\mathcal{A}$ then adaptively queries for secret keys. For each query j, $\mathcal{B}$ aborts if $j = i$. Otherwise, it responds with the secret key $\mathsf{sk}_j = \sigma_j$. After query phase, $\mathcal{A}$ sends a challenge set S and a message m to $\mathcal{B}$. $\mathcal{B}$ computes hash $h = \mathsf{Hash}_{\mathsf{SPB}}(\mathsf{hk}, \mathbb{I}_S)$ and sends the challenge instance $\mathsf{inst} = (1^\lambda, N, h, \mathsf{hk}, \mathsf{vk})$, challenge message m and challenge index $\mathsf{int}(i||0^{k+\ell}) + t$ to the position indistinguishability game challenger $\mathcal{C}$. The challenger samples a bit $\beta \leftarrow \{0, 1\}$ and responds with a ciphertext $\mathsf{ct} \leftarrow \mathsf{Encrypt}_{\mathsf{PWE}}(\mathsf{inst}, m, \mathsf{int}(i||0^{k+\ell}) + t + \beta)$[14] to $\mathcal{B}$. $\mathcal{B}$ forwards the ciphertext to $\mathcal{A}$. $\mathcal{A}$ further adaptively queries for secret keys. For each query j, $\mathcal{B}$ aborts if $j = i$. Otherwise, it responds with the secret key $\mathsf{sk}_j = \sigma_j$. Finally, $\mathcal{A}$ sends a bit b' to $\mathcal{B}$, which outputs b' as its guess in the position indistinguishability game.

As $\mathcal{A}$ is a Type 2 adversary, it does not make key query on i and therefore, $\mathcal{B}$ does not abort. As vk is punctured at i, $\nexists \sigma$ s.t. $\mathsf{Verify}_{\mathsf{ABO}}(\mathsf{vk}, i, \sigma) = 1$. This implies $\mathsf{int}(i||0^{k+\ell}) + t$ cannot be a witness of the instance $(1^\lambda, N, h, \mathsf{hk}, \mathsf{vk})$ and therefore, $\mathcal{B}$ acts as a valid adversary of the position indistinguishability game. If $\beta = 0$, $\mathcal{B}$ simulates the view of Game $2.t$ to $\mathcal{A}$ and $\Pr[b' = 1] = p_{2.t}^{\mathcal{A}}(\lambda)$. Otherwise, it simulates the view of Game $2.t{+}1$ to $\mathcal{A}$ and $\Pr[b' = 1] = p_{2.t+1}^{\mathcal{A}}(\lambda)$. This implies, the advantage of $\mathcal{B}$ in the position indistinguishability game is given by $\mathsf{AdvPosInd}^{\mathcal{B}}(\lambda) = |1/2 \cdot \Pr[b' = 0|\beta = 0] + 1/2 \cdot \Pr[b' = 1|\beta = 1] - 1/2| = \mathsf{Adv}_{2.t,2.t+1}^{\mathcal{A}}(\lambda)/2$. Therefore, $\mathsf{Adv}_{2.t,2.t+1}^{\mathcal{A}}(\lambda) \leq 2 \cdot \mathsf{AdvPosInd}(\lambda)$. $\blacksquare$

Claim 4.8. For every Type 2 PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}_{2.2^{k+\ell},3}^{\mathcal{A}}(\lambda) \leq 2 \cdot \mathsf{AdvAboInd}(\lambda)$.

Proof. Consider any Type 2 PPT adversary $\mathcal{A}$ and any $\lambda \in \mathbb{N}$. We build a PPT algorithm $\mathcal{B}$ which uses $\mathcal{A}$ and has advantage $\mathsf{Adv}_{2.2^{k+\ell},3}^{\mathcal{A}}(\lambda)$ in VK indistinguishability game of the $\mathcal{ABO}$ scheme. We ignore the description of algorithm $\mathcal{B}$ as it proceeds similar to proof of Claim 4.6.

$\blacksquare$

Note that by combining triangle inequality and Claims 4.6, 4.7, and 4.8, the advantage of any Type 2 PPT adversary $\mathcal{A}$ in AugBE index hiding game is $\mathsf{Adv}_{0,3}^{\mathcal{A}}(\lambda) \leq \mathsf{Adv}_{0,2.0}^{\mathcal{A}} + \sum_{t=0}^{2^{k+\ell}-1} \mathsf{Adv}_{2.t,2.t+1}^{\mathcal{A}} + \mathsf{Adv}_{2.2^{k+\ell},3}^{\mathcal{A}} \leq 4 \cdot \mathsf{AdvAboInd}(\lambda) + 2 \cdot 2^{k+\ell} \cdot \mathsf{AdvPosInd}(\lambda)$. Using complexity leveraging, we demand that $\mathsf{AdvPosInd}(\lambda) \leq 2^{-(k+\ell+1)} \cdot \mathrm{negl}(\lambda)$ for some negligible function $\mathrm{negl}(\cdot)$. At the instantiation level, the security parameter will be increased to match this condition.

$\blacksquare$

[14] Note that index $\mathsf{int}(i||0^{k+\ell}) + 2^{k+\ell}$ is same as $\mathsf{int}(i + 1||0^{k+\ell})$.

Note that Lemma 4.1 follows by combining Lemmas 4.2 and 4.3 as any adversary $\mathcal{A}$ of AugBE index hiding game is of either Type 1 or Type 2. ■

5 All-but-one Signatures from Standard Assumptions

In this section, we present two new constructions for all-but-one (ABO) signatures from standard assumptions. The first construction is based on verifiable random functions (VRF) and perfectly-binding (non-interactive) commitment schemes. The second construction is based on verifiable and anonymous identity-based encryption (VAIBE). The first ABO scheme satisfies perfect correctness, where as the second scheme satisfies correctness with all but negligible probability. We would like to point that using the second ABO signature scheme to instantiate the AugBE construction described in Section 4 results in AugBE scheme without perfect correctness. We finally note that VRFs can be based on simple assumptions over bilinear maps as well as RSA-like assumptions [MRV99, HJ16], and perfectly binding commitments can be constructed from any injective OWF as well as based on assumptions such as DDH, LWE and LPN [GHKW17], and VAIBE can be based on simple assumptions over bilinear maps as well as LWE [BW06b, SKOS09, LSJ+11, ABB10].[15] Therefore, this leads to constructions of ABO signatures from a wide variety of standard assumptions listed above.

5.1 All-but-one Signatures from VRFs

Let $\mathcal{VRF} = (\mathsf{Setup}_{\mathsf{VRF}}, \mathsf{Eval}_{\mathsf{VRF}}, \mathsf{Verify}_{\mathsf{VRF}})$ be a verifiable random function (VRF) with input space $\{0,1\}^{i(\lambda)}$, output space $\{0,1\}^{o(\lambda)}$ and proof space $\{0,1\}^{p(\lambda)}$. Let $\mathcal{COM} = (\mathsf{Setup}_{\mathsf{COM}}, \mathsf{Commit}, \mathsf{Verify}_{\mathsf{COM}})$ be a perfectly binding computationally hiding commitment scheme with randomness space $\{0,1\}^{o(\lambda)}$ and commitment space $\{0,1\}^{k(\lambda)}$. We construct an ABO signature scheme $\mathcal{ABO} = (\mathsf{Setup}, \mathsf{Setup\text{-}Punc}, \mathsf{Sign}, \mathsf{Verify})$ on message space $\{0,1\}^{i(\lambda)}$ and signature space $\{0,1\}^{o(\lambda)+p(\lambda)}$ as follows. For the simplicity of notation, we hereby denote $i = i(\lambda)$, $o = o(\lambda)$, $p = p(\lambda)$ and $k = k(\lambda)$.

- $\mathsf{Setup}(1^\lambda)$. Sample $(\mathsf{sk}_{\mathsf{VRF}}, \mathsf{vk}_{\mathsf{VRF}}) \leftarrow \mathsf{Setup}_{\mathsf{VRF}}(1^\lambda)$ and $\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{COM}}(1^\lambda)$. Sample $y^* \leftarrow \{0,1\}^o$ and $\mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, 0; y^*)$. Output $\mathsf{sk} = \mathsf{sk}_{\mathsf{VRF}}$ and $\mathsf{vk} = (\mathsf{pp}, \mathsf{vk}_{\mathsf{VRF}}, \mathsf{cm})$.
- $\mathsf{Setup\text{-}Punc}(1^\lambda, m^*)$. Sample $(\mathsf{sk}_{\mathsf{VRF}}, \mathsf{vk}_{\mathsf{VRF}}) \leftarrow \mathsf{Setup}_{\mathsf{VRF}}(1^\lambda)$ and $\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{COM}}(1^\lambda)$. Sample $(y^*, \pi) \leftarrow \mathsf{Eval}_{\mathsf{VRF}}(\mathsf{sk}_{\mathsf{VRF}}, m^*)$ and $\mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, 1; y^*)$. Output $\mathsf{sk} = \mathsf{sk}_{\mathsf{VRF}}$ and $\mathsf{vk} = (\mathsf{pp}, \mathsf{vk}_{\mathsf{VRF}}, \mathsf{cm})$.
- $\mathsf{Sign}(\mathsf{sk}, m)$. Sample $(y, \pi) \leftarrow \mathsf{Eval}_{\mathsf{VRF}}(\mathsf{sk}, m)$. Output $\sigma = (y, \pi)$.
- $\mathsf{Verify}(\mathsf{vk}, m, \sigma)$. Let $\sigma = (y, \pi)$ and $\mathsf{vk} = (\mathsf{pp}, \mathsf{vk}_{\mathsf{VRF}}, \mathsf{cm})$. Output 1 iff $\mathsf{Verify}_{\mathsf{VRF}}(\mathsf{vk}_{\mathsf{VRF}}, m, y, \pi) = 1 \wedge \mathsf{Verify}_{\mathsf{COM}}(\mathsf{pp}, 1, \mathsf{cm}, y) = 0$.

We now prove that the $\mathcal{ABO}$ signature scheme satisfies the required correctness properties.

Correctness of Setup. Consider any $\lambda \in \mathbb{N}$, any $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup}(1^\lambda)$, any message $m \in \{0,1\}^i$, and signature $\sigma = (y, \pi) \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$. Let $\mathsf{sk} = \mathsf{sk}_{\mathsf{VRF}}$ and $\mathsf{vk} = (\mathsf{pp}, \mathsf{vk}_{\mathsf{VRF}}, \mathsf{cm})$. By the perfect binding property of $\mathcal{COM}$ scheme, we have $\mathsf{Verify}_{\mathsf{COM}}(\mathsf{pp}, 1, \mathsf{cm}, y) = 0$. By the correctness of $\mathcal{VRF}$ scheme, we have $\mathsf{Verify}_{\mathsf{VRF}}(\mathsf{vk}_{\mathsf{VRF}}, m, y, \pi) = 1$. Therefore, $\mathsf{Verify}(\mathsf{vk}, m, \sigma) = 1$.

Correctness of Punctured Setup. Consider any $\lambda \in \mathbb{N}$, any message $m^* \in \{0,1\}^i$, $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup\text{-}Punc}(1^\lambda, m^*)$. Let $\mathsf{sk} = \mathsf{sk}_{\mathsf{VRF}}$ and $\mathsf{vk} = (\mathsf{pp}, \mathsf{vk}_{\mathsf{VRF}}, \mathsf{cm})$. Consider any $\sigma = (y, \pi) \in \{0,1\}^o \times \{0,1\}^p$. We prove that $\mathsf{Verify}(\mathsf{vk}, m^*, \sigma) = 0$. We know that for any $(y^*, \pi^*) \leftarrow \mathsf{Eval}_{\mathsf{VRF}}(\mathsf{sk}_{\mathsf{VRF}}, m^*)$, we have $\mathsf{Verify}_{\mathsf{VRF}}(\mathsf{vk}_{\mathsf{VRF}}, m^*, y^*, \pi^*) = 1$. By the unique provability property of $\mathcal{VRF}$ scheme, we know that $\mathsf{Verify}_{\mathsf{VRF}}(\mathsf{vk}_{\mathsf{VRF}}, m^*, y, \pi) = 1$ only if $y = y^*$. But we know that, $\mathsf{Verify}_{\mathsf{COM}}(\mathsf{pp}, 1, \mathsf{cm}, y^*) = 1$. This implies, $\mathsf{Verify}_{\mathsf{VRF}}(\mathsf{vk}_{\mathsf{VRF}}, m^*, y, \pi) = 1$ and $\mathsf{Verify}_{\mathsf{COM}}(\mathsf{pp}, 1, \mathsf{cm}, y) = 0$ can not happen simultaneously. Therefore, $\mathsf{Verify}(\mathsf{vk}, m^*, \sigma) = 0$.

[15] We would like to point out that most existing IBE constructions based on LWE are already verifiable and they can be made anonymous by using the transformation from [GKW17a, WZ17].

VK Indistinguishability. Assuming that $\mathcal{VRF}$ satisfies pseudorandomness property and $\mathcal{COM}$ satisfies computational hiding property, we now prove that $\mathcal{ABO}$ satisfies VK indistinguishability property. We state the lemma formally using the following games.

$\mathsf{Game}\ 0$. This game is same as VK indistinguishability game, except that the challenger always executes punctured setup algorithm.

- *Setup Phase.* The adversary $\mathcal{A}$ sends a message m^* to challenger $\mathcal{C}$. The challenger $\mathcal{C}$ samples $(\mathsf{sk}_{\mathsf{VRF}}, \mathsf{vk}_{\mathsf{VRF}}) \leftarrow \mathsf{Setup}_{\mathsf{VRF}}(1^\lambda)$ and $\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{COM}}(1^\lambda)$. It then samples $(y^*, \pi^*) \leftarrow \mathsf{Eval}_{\mathsf{VRF}}(\mathsf{sk}_{\mathsf{VRF}}, m^*)$ and $\mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, 1; y^*)$. It sends $\mathsf{vk} = (\mathsf{pp}, \mathsf{vk}_{\mathsf{VRF}}, \mathsf{cm})$ to the adversary.
- *Query Phase.* The adversary adaptively queries for signatures to the challenger. For each query $m \neq m^*$, the challenger responds with $\mathsf{Eval}_{\mathsf{VRF}}(\mathsf{sk}_{\mathsf{VRF}}, m)$.
- *Output Phase.* Finally, the adversary sends a bit b' to the challenger.

$\mathsf{Game}\ 1$. This game is similar to $\mathsf{Game}\ 0$ except that the challenger samples value y^* randomly.

- *Setup Phase.* The adversary $\mathcal{A}$ sends a message m^* to challenger $\mathcal{C}$. The challenger $\mathcal{C}$ samples $(\mathsf{sk}_{\mathsf{VRF}}, \mathsf{vk}_{\mathsf{VRF}}) \leftarrow \mathsf{Setup}_{\mathsf{VRF}}(1^\lambda)$ and $\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{COM}}(1^\lambda)$. It then samples $y^* \leftarrow \{0,1\}^o$ and $\mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, 1; y^*)$. It sends $\mathsf{vk} = (\mathsf{pp}, \mathsf{vk}_{\mathsf{VRF}}, \mathsf{cm})$ to the adversary.

$\mathsf{Game}\ 2$. This game is same as VK indistinguishability game, except that the challenger always executes normal setup algorithm.

- *Setup Phase.* The adversary $\mathcal{A}$ sends a message m^* to challenger $\mathcal{C}$. The challenger $\mathcal{C}$ samples $(\mathsf{sk}_{\mathsf{VRF}}, \mathsf{vk}_{\mathsf{VRF}}) \leftarrow \mathsf{Setup}_{\mathsf{VRF}}(1^\lambda)$ and $\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{COM}}(1^\lambda)$. It then samples $y^* \leftarrow \{0,1\}^o$ and $\mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, 0; y^*)$. It sends $\mathsf{vk} = (\mathsf{pp}, \mathsf{vk}_{\mathsf{VRF}}, \mathsf{cm})$ to the adversary.

In all the above games, the adversary is not allowed to make a signature query on message m^*.

For any stateful PPT adversary $\mathcal{A}$, let the probability that $\mathcal{A}$ outputs 1 in $\mathsf{Game}\ y$ be $p_y^{\mathcal{A}}(\lambda)$. We denote the advantage of a PPT adversary $\mathcal{A}$ in distinguishing between any two games $\mathsf{Game}\ x$ and $\mathsf{Game}\ y$ by $\mathsf{Adv}_{x,y}^{\mathcal{A}}(\lambda) = |p_x^{\mathcal{A}}(\lambda) - p_y^{\mathcal{A}}(\lambda)|$. We prove that $\mathsf{Game}\ 0$ is computationally indistinguishable from $\mathsf{Game}\ 2$ using the following 2 claims.

Claim 5.1. Assuming that $\mathcal{VRF}$ has selective pseudorandomness property, for every PPT algorithm $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}_{0,1}^{\mathcal{A}}(\lambda) \leq \text{negl}(\lambda)$.

Proof. Consider any PPT adversary $\mathcal{A}$. We build a PPT algorithm $\mathcal{B}$ that uses $\mathcal{A}$ and breaks pseudorandomness property of $\mathcal{VRF}$ with advantage $1/2 \cdot \mathsf{Adv}_{0,1}^{\mathcal{A}}(\lambda)$. The reduction algorithm $\mathcal{B}$ proceeds as follows.

The adversary $\mathcal{A}$ first sends a challenge message m^* to $\mathcal{B}$, which it forwards to the VRF pseudorandomness game challenger $\mathcal{C}$. $\mathcal{C}$ samples $(\mathsf{sk}_{\mathsf{VRF}}, \mathsf{vk}_{\mathsf{VRF}}) \leftarrow \mathsf{Setup}_{\mathsf{VRF}}(1^\lambda)$ and a bit $b \leftarrow \{0,1\}$. If $b = 0$, $\mathcal{C}$ samples $(y^*, \pi^*) \leftarrow \mathsf{Eval}_{\mathsf{VRF}}(\mathsf{sk}_{\mathsf{VRF}}, m^*)$. If $b = 1$, it samples $y^* \leftarrow \{0,1\}^o$. The challenger then sends verification key $\mathsf{vk}_{\mathsf{VRF}}$ and y^* to $\mathcal{B}$. $\mathcal{B}$ samples $\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{COM}}(1^\lambda)$, computes $\mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, 1; y^*)$ and sends ABO verification key $\mathsf{vk} = (\mathsf{pp}, \mathsf{vk}_{\mathsf{VRF}}, \mathsf{cm})$ to $\mathcal{A}$. The adversary adaptively queries for signatures to $\mathcal{B}$. For each query $m \neq m^*$, $\mathcal{B}$ forwards the query to $\mathcal{C}$, which responds with $(y, \pi) \leftarrow \mathsf{Eval}_{\mathsf{VRF}}(\mathsf{sk}_{\mathsf{VRF}}, m)$. $\mathcal{B}$ then forwards the signature (y, π) to $\mathcal{A}$. Finally, $\mathcal{A}$ sends a bit b' to $\mathcal{B}$, which outputs b' as its guess in the VRF pseudorandomness game.

Note that $\mathcal{A}$ is not allowed to make signature query on m^*. Therefore, $\mathcal{B}$ does not make any VRF evaluation query on m^*, and acts as a valid adversary in VRF pseudorandomness game. We observe that if $b = 0$, then $\mathcal{B}$ simulates the view of $\mathsf{Game}\ 0$ to $\mathcal{A}$ and $\Pr[b' = 1] = p_0^{\mathcal{A}}(\lambda)$. Otherwise, it simulates the view of $\mathsf{Game}\ 1$ to $\mathcal{A}$ and $\Pr[b' = 1] = p_1^{\mathcal{A}}(\lambda)$. This implies, the advantage of $\mathcal{B}$ is given by $\Pr[b' = b] = |1/2 \cdot \Pr[b' = 0 | b = 0] + 1/2 \cdot \Pr[b' = 1 | b = 1] - 1/2| = 1/2 \cdot \mathsf{Adv}_{0,1}^{\mathcal{A}}(\lambda)$. Therefore, if $\mathcal{A}$ has non-negligible advantage in distinguishing between $\mathsf{Game}\ 0$ and $\mathsf{Game}\ 1$, then $\mathcal{B}$ can break pseudorandomness property of $\mathcal{VRF}$ scheme. ■

Claim 5.2. Assuming that $\mathcal{COM}$ has computational hiding property, for every PPT algorithm $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}^{\mathcal{A}}_{1,2}(\lambda) \leq \text{negl}(\lambda)$.

Proof. Consider any PPT adversary $\mathcal{A}$. We build a PPT algorithm $\mathcal{B}$ that uses $\mathcal{A}$ and breaks computational hiding property of $\mathcal{COM}$ with advantage $1/2 \cdot \mathsf{Adv}^{\mathcal{A}}_{1,2}(\lambda)$. The reduction algorithm $\mathcal{B}$ proceeds as follows.

The computational hiding game challenger $\mathcal{C}$ first samples $\mathsf{pp} \leftarrow \mathsf{Setup}_{\mathsf{COM}}(1^\lambda)$, a bit $b \leftarrow \{0,1\}$, $y \leftarrow \{0,1\}^o$ and $\mathsf{cm} \leftarrow \mathsf{Commit}(\mathsf{pp}, b; y)$. It then sends $(\mathsf{pp}, \mathsf{cm})$ to $\mathcal{B}$. The adversary $\mathcal{A}$ then sends a challenge message m^* to $\mathcal{B}$. $\mathcal{B}$ samples $(\mathsf{sk}_{\mathsf{VRF}}, \mathsf{vk}_{\mathsf{VRF}}) \leftarrow \mathsf{Setup}_{\mathsf{VRF}}(1^\lambda)$ and sends ABO verification key $\mathsf{vk} = (\mathsf{pp}, \mathsf{vk}_{\mathsf{VRF}}, \mathsf{cm})$ to $\mathcal{A}$. The adversary adaptively queries for signatures to $\mathcal{B}$. For each query m, $\mathcal{B}$ samples $(y, \pi) \leftarrow \mathsf{Eval}_{\mathsf{VRF}}(\mathsf{sk}_{\mathsf{VRF}}, m)$ and responds with signature (y, π). Finally, $\mathcal{A}$ sends a bit b' to $\mathcal{B}$, which outputs b' as its guess in the computational hiding game.

We observe that if $b = 0$, then $\mathcal{B}$ simulates the view of Game 2 to $\mathcal{A}$ and $\Pr[b' = 1] = p^{\mathcal{A}}_2(\lambda)$. Otherwise, it simulates the view of Game 1 to $\mathcal{A}$ and $\Pr[b' = 1] = p^{\mathcal{A}}_1(\lambda)$. This implies, the advantage of $\mathcal{B}$ is given by $\Pr[b' = b] = |1/2 \cdot \Pr[b' = 0 | b = 0] + 1/2 \cdot \Pr[b' = 1 | b = 1] - 1/2| = 1/2 \cdot \mathsf{Adv}^{\mathcal{A}}_{1,2}(\lambda)$. Therefore, if $\mathcal{A}$ has non-negligible advantage in distinguishing between Game 1 and Game 2, then $\mathcal{B}$ can break computational hiding property of $\mathcal{COM}$ scheme.

■

Note that by combining Claims 5.1 and 5.2, for every PPT adversary $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}^{\mathcal{A}}_{0,2}(\lambda) \leq \text{negl}(\lambda)$.

5.2 All-but-one Signatures from VAIBE

In this section, we construct all-but-one (ABO) signatures from verifiable and anonymous identity based encryption system (VAIBE). Let $\mathcal{VAIBE} = (\mathsf{Setup}_{\mathsf{VAIBE}}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Verify}_{\mathsf{VAIBE}})$ be a VAIBE scheme for message space $\{0,1\}^{m(\lambda)}$, ciphertext space $\{0,1\}^{c(\lambda)}$, secret key space $\{0,1\}^{k(\lambda)}$, identity space $\{0,1\}^{i(\lambda)}$ and proof space $\{0,1\}^{r(\lambda)}$. We construct an ABO signature scheme $\mathcal{ABO} = (\mathsf{Setup}, \mathsf{Setup\text{-}Punc}, \mathsf{Sign}, \mathsf{Verify})$ for message space $\{0,1\}^{i(\lambda)} \setminus \{0^{i(\lambda)}\}$ and signature space $\{0,1\}^{k(\lambda)+r(\lambda)}$ i.e., for every $\lambda \in \mathbb{N}$, identity $0^{i(\lambda)}$ is not supported by the signature scheme. Let $\mathcal{I}_\lambda = \{0,1\}^{i(\lambda)} \setminus \{0^{i(\lambda)}\}$. For simplicity of notation, we hereby denote $m = m(\lambda), c = c(\lambda), k = k(\lambda), i = i(\lambda)$ and $p = p(\lambda)$. Also, we hereby refer to messages in $\mathcal{ABO}$ scheme by identities in $\mathcal{VAIBE}$ scheme. Formally, the construction proceeds as follows.

- $\mathsf{Setup}(1^\lambda)$. Sample VAIBE keys $(\mathsf{mpk}_{\mathsf{VAIBE}}, \mathsf{msk}_{\mathsf{VAIBE}}) \leftarrow \mathsf{Setup}_{\mathsf{VAIBE}}(1^\lambda)$. Sample a random message $x \leftarrow \{0,1\}^m$ and compute ciphertext $t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}_{\mathsf{VAIBE}}, 0^i, x)$. Output secret key $\mathsf{sk} = \mathsf{msk}_{\mathsf{VAIBE}}$ and verification key $\mathsf{vk} = (x, \mathsf{mpk}_{\mathsf{VAIBE}}, t)$.
- $\mathsf{Setup\text{-}Punc}(1^\lambda, \mathsf{id}^*)$. Sample VAIBE keys $(\mathsf{mpk}_{\mathsf{VAIBE}}, \mathsf{msk}_{\mathsf{VAIBE}}) \leftarrow \mathsf{Setup}_{\mathsf{VAIBE}}(1^\lambda)$. Choose a random message $x \leftarrow \{0,1\}^m$. Encrypt the message x using identity id^* i.e., compute ciphertext $t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}_{\mathsf{VAIBE}}, \mathsf{id}^*, x)$. Output secret key $\mathsf{sk} = \mathsf{msk}_{\mathsf{VAIBE}}$ and verification key $\mathsf{vk} = (x, \mathsf{mpk}_{\mathsf{VAIBE}}, t)$.
- $\mathsf{Sign}(\mathsf{sk}, \mathsf{id})$. Sample $(\mathsf{sk}_{\mathsf{id}}, \pi) \leftarrow \mathsf{KeyGen}(\mathsf{sk}, \mathsf{id})$. Output signature $\sigma = (\mathsf{sk}_{\mathsf{id}}, \pi)$.
- $\mathsf{Verify}(\mathsf{vk}, \mathsf{id}, \sigma)$. Let $\sigma = (\mathsf{sk}', \pi)$ and $\mathsf{vk} = (x, \mathsf{mpk}, t)$. Output 1 iff $\mathsf{Verify}_{\mathsf{VAIBE}}(\mathsf{mpk}, \mathsf{id}, \mathsf{sk}', \pi) = 1 \wedge x \neq \mathsf{Decrypt}(\mathsf{sk}', t)$.

We note that the $\mathcal{ABO}$ scheme does not achieve perfect correctness [16]. We now prove that the $\mathcal{ABO}$ scheme satisfies the required correctness properties with all but negligible probability.

[16] Using this ABO scheme in our AugBE construction results in an AugBE scheme without perfect correctness.

Correctness of Setup.

Claim 5.3. There exists a negligible function $\mathrm{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$ and any identity $\mathsf{id} \in \mathcal{I}_\lambda$, we have

$$\Pr\left[\mathsf{Verify}(\mathsf{vk}, \mathsf{id}, \sigma) = 0 \;:\; \begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}_{\mathsf{VAIBE}}(1^\lambda), x_0 \leftarrow \{0,1\}^m, t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, 0^i, x_0) \\ \mathsf{vk} \leftarrow (x_0, \mathsf{mpk}, t), \sigma = (\mathsf{sk}_{\mathsf{id}}, \pi) \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}) \end{array}\right] \leq \frac{1}{2^m} + \mathrm{negl}(\lambda).$$

Proof. Suppose there exists a non-negligible function $\delta(\cdot)$ such that, for every $\lambda \in \mathbb{N}$, there exists an identity $\mathsf{id}'_\lambda \in \mathcal{I}_\lambda$ such that,

$$\Pr\left[\mathsf{Verify}(\mathsf{vk}, \mathsf{id}'_\lambda, \sigma) = 0 \;:\; \begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}_{\mathsf{VAIBE}}(1^\lambda), x_0 \leftarrow \{0,1\}^m, t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, 0^i, x_0) \\ \mathsf{vk} \leftarrow (x_0, \mathsf{mpk}, t), \sigma = (\mathsf{sk}_{\mathsf{id}'_\lambda}, \pi) \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}'_\lambda) \end{array}\right] > \frac{1}{2^m} + \delta(\lambda).$$

By the correctness of $\mathcal{VAIBE}$ scheme, we know that $\mathsf{Verify}_{\mathsf{VAIBE}}(\mathsf{mpk}, \mathsf{id}'_\lambda, \mathsf{sk}_{\mathsf{id}'_\lambda}, \pi) = 1$. This implies,

$$\Pr\left[\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}'_\lambda}, t) = x_0 \;:\; \begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}_{\mathsf{VAIBE}}(1^\lambda), x_0 \leftarrow \{0,1\}^m \\ t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, 0^i, x_0), \sigma = (\mathsf{sk}_{\mathsf{id}'_\lambda}, \pi) \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}'_\lambda) \end{array}\right] > \frac{1}{2^m} + \delta(\lambda). \tag{4}$$

For any fixed $x_0 \in \{0,1\}^m$, let

$$p_{x_0} = \Pr\left[\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}'_\lambda}, t) = x_0 \;:\; \begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}_{\mathsf{VAIBE}}(1^\lambda), x_1 \leftarrow \{0,1\}^m \\ t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, 0^i, x_1), (\mathsf{sk}_{\mathsf{id}'_\lambda}, \pi) \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}'_\lambda) \end{array}\right].$$

We know that $\sum_{x_0} p_{x_0} = 1$. This implies,

$$\Pr\left[\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}'_\lambda}, t) = x_0 \;:\; \begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}_{\mathsf{VAIBE}}(1^\lambda), x_1 \leftarrow \{0,1\}^m, x_0 \leftarrow \{0,1\}^m \\ t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, 0^i, x_1), (\mathsf{sk}_{\mathsf{id}'_\lambda}, \pi) \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}'_\lambda) \end{array}\right] = \frac{1}{2^m}. \tag{5}$$

We build a non-uniform PPT adversary $\mathcal{A}$ that breaks IND-CPA security of $\mathcal{VAIBE}$ scheme. The algorithm proceeds as follows. Assume the adversary is given id'_λ as a non-uniform advice. $\mathcal{A}$ first samples two random messages $x_0 \leftarrow \{0,1\}^m, x_1 \leftarrow \{0,1\}^m$ and sends challenge messages (x_0, x_1) and challenge identity 0^i to VAIBE IND-CPA challenger $\mathcal{C}$. $\mathcal{C}$ samples VAIBE keys $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}_{\mathsf{VAIBE}}(1^\lambda)$, a bit $b \leftarrow \{0,1\}$, and computes ciphertext $t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, 0^i, x_b)$. $\mathcal{C}$ sends public key mpk and challenge response t to $\mathcal{A}$. The adversary then makes a key query on index id'_λ to the challenger, which responds with $(\mathsf{sk}_{\mathsf{id}'_\lambda}, \pi) \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}'_\lambda)$. $\mathcal{A}$ outputs 1 if $\mathsf{Decrypt}(\mathsf{sk}_{\mathsf{id}'_\lambda}, t) = x_0$ and outputs 0 otherwise.

By equation 4, if $b = 0$, $\mathcal{A}$ outputs 1 with probability greater than $\frac{1}{2^m} + \delta(\lambda)$. By equation 5, if $b = 1$, $\mathcal{A}$ outputs 1 with probability $\frac{1}{2^m}$. This implies that the advantage of $\mathcal{A}$ in the IND-CPA game is at least $1/2 \cdot \delta(\lambda)$. ■

Correctness of Punctured Setup.

Claim 5.4. For all $\lambda \in \mathbb{N}$, any identity $\mathsf{id}^* \in \mathcal{I}_\lambda$, any keys $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Setup\text{-}Punc}(1^\lambda, \mathsf{id}^*)$, any $\sigma \leftarrow \{0,1\}^{k+r}$, we have $\mathsf{Verify}(\mathsf{vk}, \mathsf{id}^*, \sigma) = 0$.

Proof. Let $\mathsf{vk} = (x, \mathsf{mpk}, t)$ and $\sigma = (\mathsf{sk}', \pi)$. From the soundness of verifiability property of $\mathcal{VAIBE}$ scheme, we know that if $\mathsf{Verify}_{\mathsf{VAIBE}}(\mathsf{mpk}, \mathsf{id}^*, \mathsf{sk}', \pi) = 1$, then $\mathsf{Decrypt}(\mathsf{sk}', t) = x$. Therefore, $\mathsf{Verify}(\mathsf{vk}, \mathsf{id}^*, \sigma) = 0$. ■

5.2.1 VK Indistinguishability

We now prove that $\mathcal{ABO}$ scheme satisfies the VK indistinguishability property. We first define the VK indistinguishability game below.

Game VK-IND.

- *Setup Phase.* The adversary sends a challenge id^* to the challenger. The challenger samples $(\mathsf{mpk}_{\mathsf{VAIBE}},\mathsf{msk}_{\mathsf{VAIBE}}) \leftarrow \mathsf{Setup}_{\mathsf{VAIBE}}(1^\lambda)$, a message $x \leftarrow \{0,1\}^m$ and a random bit $b \leftarrow \{0,1\}$. If $b = 0$, it samples $t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}_{\mathsf{VAIBE}}, 0^i, x)$. If $b = 1$, it samples $t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}_{\mathsf{VAIBE}}, \mathsf{id}^*, x)$. It then sends verification key $\mathsf{vk} = (x, \mathsf{mpk}_{\mathsf{VAIBE}}, t)$ to the adversary.

- *Query Phase.* The adversary adaptively queries for signatures to the challenger. For each query id, the challenger computes $(\mathsf{sk}_{\mathsf{id}}, \pi) \leftarrow \mathsf{KeyGen}(\mathsf{msk}_{\mathsf{VAIBE}}, \mathsf{id})$ and responds with signature $\sigma = (\mathsf{sk}_{\mathsf{id}}, \pi)$.

- *Output Phase.* The adversary sends a bit b' to the challenger. The adversary wins if $b' = b$.

Here, $\mathcal{A}$ is not allowed to make signature query on id^*. For any stateful PPT adversary $\mathcal{A}$, we define the advantage of the adversary $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{VK\text{-}IND}}(\lambda) = |\Pr[\mathcal{A} \text{ wins}] - 1/2|$.

Lemma 5.1. Assuming that $\mathcal{VAIBE}$ scheme is IND-ANON secure as per Definition 2.7, for every PPT adversary $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that for every $\lambda \in \mathbb{N}$, we have $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{VK\text{-}IND}}(\lambda) \leq \text{negl}(\lambda)$.

Proof. Suppose there exists an adversary $\mathcal{A}$ and a non-negligible function $\delta(\cdot)$ such that $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{VK\text{-}IND}}(\lambda) > \delta(\lambda)$ for every $\lambda \in \mathbb{N}$. We describe a reduction algorithm $\mathcal{B}$ that uses $\mathcal{A}$ and breaks anonymous IBE property of $\mathcal{VAIBE}$ scheme. The algorithm $\mathcal{B}$ proceeds as follows.

The adversary $\mathcal{A}$ sends an identity $\mathsf{id}^* \in \mathcal{I}_\lambda$ to $\mathcal{B}$. The reduction algorithm $\mathcal{B}$ samples a message $x \leftarrow \{0,1\}^m$, and sends challenge identities $(0^i, \mathsf{id}^*)$ and challenge message x to IND-ANON game challenger $\mathcal{C}$. The challenger samples $(\mathsf{mpk}_{\mathsf{VAIBE}}, \mathsf{msk}_{\mathsf{VAIBE}}) \leftarrow \mathsf{Setup}_{\mathsf{VAIBE}}(1^\lambda)$, a random bit $b \leftarrow \{0,1\}$, and responds with either $(\mathsf{mpk}_{\mathsf{VAIBE}}, t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}_{\mathsf{VAIBE}}, 0^i, x))$ if $b = 0$ or with $(\mathsf{mpk}_{\mathsf{VAIBE}}, t \leftarrow \mathsf{Encrypt}(\mathsf{mpk}_{\mathsf{VAIBE}}, \mathsf{id}^*, x))$ if $b = 1$. $\mathcal{B}$ sends the ABO verification key $(x, \mathsf{mpk}_{\mathsf{VAIBE}}, t)$ to the adversary $\mathcal{A}$. The adversary adaptively queries for signatures to $\mathcal{B}$. For each query id, $\mathcal{B}$ aborts if $\mathsf{id} = \mathsf{id}^*$. Otherwise, it makes a secret key query on id to $\mathcal{C}$. The challenger responds with $(\mathsf{sk}_{\mathsf{id}}, \pi) \leftarrow \mathsf{KeyGen}(\mathsf{msk}_{\mathsf{VAIBE}}, \mathsf{id})$, which $\mathcal{B}$ forwards to the adversary. Finally, $\mathcal{A}$ sends a bit b' to $\mathcal{B}$, which outputs b' as its guess in IND-ANON game.

As $\mathcal{A}$ does not make signature query on id^*, $\mathcal{B}$ does not make secret key queries on $0^i, \mathsf{id}^*$. Therefore, $\mathcal{B}$ acts as a valid adversary in the anonymous IBE property game. In addition, $\mathcal{B}$ simulates the view of Game VK-IND to $\mathcal{A}$. Therefore, $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{VK\text{-}IND}}(\lambda) > \delta(\lambda)$ and the advantage of $\mathcal{B}$ in the IND-ANON game is at least $\delta(\lambda)$. ■

Acknowledgement

We thank the anonymous reviewers of PKC 2019 for helpful feedback, especially for pointing out the connection between SPB hashes and (a weakening of) ABO signatures.

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (h)ibe in the standard model. In *Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'10, pages 553–572, Berlin, Heidelberg, 2010. Springer-Verlag.

[ABG$^+$13] Prabhanjan Ananth, Dan Boneh, Sanjam Garg, Amit Sahai, and Mark Zhandry. Differing-inputs obfuscation and applications. Cryptology ePrint Archive, Report 2013/689, 2013.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In *Advances in Cryptology - CRYPTO 2015*, pages 308–326, 2015.

[AJS15] Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Achieving compactness generically: Indistinguishability obfuscation from non-compact functional encryption. *IACR Cryptology ePrint Archive*, 2015.

[AP16] Navid Alamati and Chris Peikert. Three's compromised too: Circular insecurity for any cycle length from (ring-)lwe. In *Advances in Cryptology - CRYPTO 2016*, pages 659–680, 2016.

[BBDP01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 566–582. Springer, 2001.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil Pairing. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, 2001.

[BGI$^+$01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *Advances in Cryptology - CRYPTO 2001*, pages 1–18, 2001.

[BGI$^+$12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6, 2012.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, pages 258–275, 2005.

[BJK$^+$17] Zvika Brakerski, Aayush Jain, Ilan Komargodski, Alain Passelegue, and Daniel Wichs. Non-trivial witness encryption and null-io from standard assumptions. Cryptology ePrint Archive, Report 2017/874, 2017. https://eprint.iacr.org/2017/874.

[BP15] Nir Bitansky and Omer Paneth. Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In *Theory of Cryptography*, pages 401–427. Springer Berlin Heidelberg, 2015.

[BPW16] Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In *Theory of Cryptography - 13th International Conference, TCC 2016-A*, pages 474–502, 2016.

[BSW06] Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *EUROCRYPT*, pages 573–592, 2006.

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*, pages 171–190, 2015.

[BW06a] Dan Boneh and Brent Waters. A fully collusion resistant broadcast, trace, and revoke system. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, pages 211–220, 2006.

[BW06b] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *Advances in Cryptology - CRYPTO 2006*, pages 290–307, 2006.

[BWZ14] Dan Boneh, Brent Waters, and Mark Zhandry. Low overhead broadcast encryption from multilinear maps. In *Advances in Cryptology - CRYPTO 2014*, pages 206–223, 2014.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *Advances in Cryptology - CRYPTO 2014*, pages 480–499, 2014.

[CDG$^+$17] Chongwon Cho, Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Antigoni Polychroniadou. Laconic oblivious transfer and its applications. In *Annual International Cryptology Conference*, 2017.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *Advances in Cryptology - CRYPTO 2013*, pages 476–493, 2013.

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In *Advances in Cryptology - CRYPTO 2015*, 2015.

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015*, pages 468–497, 2015.

[CVW18] Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. Ggh15 beyond permutation branching programs: Proofs, attacks, and candidates. In *CRYPTO*, 2018.

[FN94] Amos Fiat and Moni Naor. Broadcast encryption. In *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '93, pages 480–491, 1994.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, 2013.

[GGH$^+$13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indstinguishability obfuscation and functional encryption for all circuits. In *FOCS*, 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *TCC*, 2015.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014*, pages 74–94, 2014.

[GGHW14] Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In *Advances in Cryptology - CRYPTO 2014*, pages 518–535, 2014.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *STOC*, 2013.

[GHKW17] Rishab Goyal, Susan Hohenberger, Venkata Koppula, and Brent Waters. A generic approach to constructing and proving verifiable random functions. In *Theory of Cryptography - 15th International Conference, TCC 2017*, pages 537–566, 2017.

[GKP$^+$13] Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, , and Nickolai Zeldovich. How to run turing machines on encrypted data. In *CRYPTO*, 2013.

[GKRW17] Rishab Goyal, Venkata Koppula, Andrew Russell, and Brent Waters. Risky traitor tracing and new differential privacy negative results. Cryptology ePrint Archive, Report 2017/1117, 2017. https://eprint.iacr.org/2017/1117.

[GKW17a] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 612–621, 2017.

[GKW17b] Rishab Goyal, Venkata Koppula, and Brent Waters. Separating semantic and circular security for symmetric-key bit encryption from the learning with errors assumption. In *EUROCRYPT*, 2017.

[GKW18] Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In *STOC*, 2018.

[GLSW15] Craig Gentry, Allison Bishop Lewko, Amit Sahai, and Brent Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*, pages 151–170, 2015.

[GLW14] Craig Gentry, Allison B. Lewko, and Brent Waters. Witness encryption from instance independent assumptions. In *Advances in Cryptology - CRYPTO 2014*, pages 426–443, 2014.

[GPS16] Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. Revisiting the cryptographic hardness of finding a nash equilibrium. In *Advances in Cryptology - CRYPTO 2016*, pages 579–604, 2016.

[GPSZ17] Sanjam Garg, Omkant Pandey, Akshayaram Srinivasan, and Mark Zhandry. Breaking the sub-exponential barrier in obfustopia. In *Advances in Cryptology - EUROCRYPT 2017*, pages 156–181, 2017.

[GW09] Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In *Advances in Cryptology - EUROCRYPT 2009*, pages 171–188, 2009.

[HJ16] Dennis Hofheinz and Tibor Jager. Verifiable random functions from standard assumptions. In *Theory of Cryptography - 13th International Conference, TCC 2016-A*, 2016.

[HJK$^+$16] Dennis Hofheinz, Tibor Jager, Dakshita Khurana, Amit Sahai, Brent Waters, and Mark Zhandry. How to generate and use universal samplers. In *Advances in Cryptology - ASIACRYPT 2016*, pages 715–744, 2016.

[HW15] Pavel Hubácek and Daniel Wichs. On the communication complexity of secure function evaluation with long output. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015*, pages 163–172, 2015.

[KLW15] Venkata Koppula, Allison Bishop Lewko, and Brent Waters. Indistinguishability obfuscation for turing machines with unbounded memory. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015*, pages 419–428, 2015.

[KW16] Venkata Koppula and Brent Waters. Circular security counterexamples for arbitrary length cycles from LWE. In *CRYPTO*, 2016.

[LSJ$^+$11] Song Luo, Qingni Shen, Yongming Jin, Yu Chen, Zhong Chen, and Sihan Qing. A variant of boyen-waters anonymous IBE scheme. In *Information and Communications Security - 13th International Conference, ICICS 2011*, pages 42–56, 2011.

[LZ17] Qipeng Liu and Mark Zhandry. Decomposable obfuscation: A framework for building applications of obfuscation from polynomial hardness. In *Proceedings of TCC 2017*, 2017.

[MRV99] Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *In Proc. 40th IEEE Symposium on Foundations of Computer Science (FOCS*, pages 120–130. IEEE, 1999.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In *Advances in Cryptology - CRYPTO 2001*, pages 41–62, 2001.

[NP00] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In *Financial Cryptography, 4th International Conference, FC 2000*, pages 1–20, 2000.

[NWZ16] Ryo Nishimaki, Daniel Wichs, and Mark Zhandry. Anonymous traitor tracing: How to embed arbitrary information in a key. In *Advances in Cryptology - EUROCRYPT 2016*, pages 388–419, 2016.

[OPWW15] Tatsuaki Okamoto, Krzysztof Pietrzak, Brent Waters, and Daniel Wichs. New realizations of somewhere statistically binding hashing and positional accumulators. In *Advances in Cryptology - ASIACRYPT 2015*, pages 121–145, 2015.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, 2005*, pages 84–93, 2005.

[Sha85] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[SKOS09] Jae Hong Seo, Tetsutaro Kobayashi, Miyako Ohkubo, and Koutarou Suzuki. Anonymous hierarchical identity-based encryption with constant size ciphertexts. In *Public Key Cryptography - PKC 2009*, pages 215–234, 2009.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 475–484, 2014.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pages 600–611, 2017.

[Zha16] Mark Zhandry. How to avoid obfuscation using witness prfs. In *Theory of Cryptography Conference*, pages 421–448. Springer, 2016.