Bifurcated Signatures: Folding the Accountability vs. Anonymity Dilemma into a Single Private Signing Scheme

Benoît Libert[1,2], Khoa Nguyen[3], Thomas Peters[4], and Moti Yung[5]

[1] CNRS, Laboratoire LIP, France
[2] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, Inria, UCBL), France
[3] Nanyang Technological University, SPMS, Singapore
[4] FNRS and UCLouvain (ICTEAM), Belgium
[5] Google and Columbia University, USA

Abstract. Over the development of modern cryptography, often, alternative cryptographic schemes are developed to achieve goals that in some important respect are orthogonal. Thus, we have to choose either a scheme which achieves the first goal and not the second, or vice versa. This results in two types of schemes that compete with each other. In the basic area of user privacy, specifically in anonymous (multi-use credentials) signing, such an orthogonality exists between anonymity and accountability.
The conceptual contribution of this work is to reverse the above orthogonality by design, which essentially typifies the last 25 years or so, and to suggest an alternative methodology where the opposed properties are carefully folded into a single scheme. The schemes will support both opposing properties simultaneously in a bifurcated fashion, where:

- First, based on rich semantics expressed over the message's context and content, the user, etc., the relevant property is applied point-wise per message operation depending on a predicate; and
- Secondly, at the same time, the schemes provide what we call "branch-hiding;" namely, the resulting calculated value hides from outsiders which property has actually been locally applied.

Specifically, we precisely define and give the first construction and security proof of a "Bifurcated Anonymous Signature" (BiAS): A scheme which supports either absolute anonymity or anonymity with accountability, based on a specific contextual predicate, while being branch-hiding. This novel signing scheme has numerous applications not easily implementable or not considered before, especially because: (i) the conditional traceability does *not* rely on a trusted authority as it is (non-interactively) encapsulated into signatures; and (ii) signers *know* the predicate value and can make a conscious choice at each signing time.
Technically, we realize BiAS from homomorphic commitments for a general family of predicates that can be represented by bounded-depth circuits. Our construction is generic and can be instantiated in the standard model from lattices and, more efficiently, from bilinear maps. In particular, the signature length is independent of the circuit size when we use commitments with suitable efficiency properties.

Keywords. New primitive, privacy, anonymity, accountability, group signatures, conditional traceability, predicate-based privacy.

1 Introduction

Properties provided by cryptographic primitives (such as confidentiality and anonymity) generate a natural tension between the requirements of individual users (such as privacy and other rights), and those of society (such as safety and individual accountability). This fact has created a very rigid positioning of cryptosystems: Designs that serve individual needs, and those which serve societal concerns. A classical example of the above rigidity is the scenario of anonymous signing. On the one hand, there are group signatures [18], central privacy tools allowing users to anonymously sign messages in the name of a population of users they belong to. In order to keep users accountable for their actions, group signatures involve a trusted opening authority (OA) which is called upon when needed only, and is endowed with some privileged information allowing it to trace any signature back to its author. This accountability mechanism, therefore, in these cases, revokes the anonymity of that user. On the other hand, ring signatures [45] and related primitives [30,39,19,11] allow users to sign whatever they like in the name of a population while retaining unconditional anonymity.

In light of this over quarter-of-a-century old situation, we claim that for many applications, in fact, group and ring signatures fall short of providing an appropriate tradeoff between anonymity and accountability that would be sufficiently fair for, both, signers and authorities. Privacy-aware signers naturally want to protect their privacy as much as possible. At the same time, authorities aim to ensure that signers of all problematic signatures can be caught. What we argue in this work is that in many real-life situations, the ability to trace a signature or not should actually depend on the content and context of the message, and should be provided programmatically by the primitive rather than being supported by a one sided mechanism which is part of the primitive specification.

Consider the scenario where each signature authenticates an anonymous financial transaction associated with a hidden amount of money and between users in different countries that should only be known to the payer and the payee (this can be done by employing additional cryptographic mechanisms, e.g., the amount and countries are encrypted or committed to, as in the privacy-preserving cryptocurrency system Monero [41]). For money-laundering detection purposes, the authorities would like to make sure that transactions with amounts above a certain threshold between two specific countries can be traced. On the other hand, to satisfy privacy-aware users, the system should also provide absolute anonymity for transactions amounts below the traceability threshold or within the same country, say, or for any other messages of harmless content. Importantly, for privacy reasons as well (e.g., keeping statistics of the financial transactions hidden), the public should not be able to determine whether a given signed transaction, corresponds to a traceable or to an untraceable type.

As another example, imagine that visitors of a digital library are required to register and sign before reviewing specific e-books. The ability/ inability to identify signers should naturally depend on whether the books in question are totally benign (e.g., comics, essays containing controversial but inoffensive political opinions, etc.) or potentially harmful (like chemistry books explaining

how to produce bombs, any form of advocacy of hatred, etc.).

As yet another example, note that service providers often ask users to attest to personal attributes, for example, to guarantee the veracity of answers to questions like "Have you been to one of these countries in the last 6 months?", "Are you above 18?", etc. In this case we argue that while suspicious online activities and alert-raising messages should be traceable by some warranted authority, regular well-behaved typical users should not have to reveal their history and information to service providers that verify their signatures.

The above examples motivate the design of a new anonymous signature primitive where the ability to trace a signature back to its source is determined by a predicate that depends on the signed message and the user's credential, and where the traceability property of a signature is hidden from the general public. Such schemes are highly desirable, so that they can support the above scenarios, since they provide a fair privacy-preserving non-rigid setting which users and authorities both have strong incentives to deploy. However, to our knowledge, a non-rigid conditional setting, and such signatures have never been considered so far. We call this type of schemes that allow this on the fly flexibility "Bifurcated Cryptosystems."

1.1 Our Contributions

We introduce the study of cryptographic primitives that provide tradeoffs between competing requirements like end-to-end privacy and accountability, by suggesting *bifurcated anonymous signatures* (BiAS). In short, BiAS schemes are anonymous signature schemes allowing to bifurcate into absolute anonymity and identity escrow at the signing time, where computing signatures is associated with a predicate P. They enable unconditional anonymity when the predicate $P(M, \mathsf{id}, w)$ evaluates to 1 on input of the message M, the user's identity id and some secret context-dependent piece of information w which we call a witness. At the same time, signatures should be traceable by an authority whenever $P(M, \mathsf{id}, w) = 0$.

As a first major requirement, BiAS must be *branch-hiding*: verifiers as well as the issuing credential authority should be unable to figure out whether a signature is traceable or not.

In our BiAS primitive, whether a signature is traceable or not depends on the content which is being signed and the user's identity. Since users know the predicate and its value before signing, they know beforehand when they will be subject to tracing and they can make an educated decision as to whether they can afford to sign a specific message or not. At the same time, *only* the tracing authority will be able to learn whether the signatures are traceable or not. The users are assured that if their signature is not traceable, no one, even if the authority's keys are available, will be able to trace them. In fact, let us emphasize that this is an unconditional anonymity property, which is of high importance in some applications, such as the case of journalists signing an article unfavorable to the local regime, in a place where their life is in danger upon eventual identification.

As a natural second requirement, we pair the notion of branch-hiding with

the notion of *branch-soundness*. Branch-soundness prevents users from generating untraceable signatures when the signer should have been identified or vice versa. Said otherwise, no signers can fool the system even with the help of the authorities and be able to produce a signature of which the traceability does not respect the predicate.

We first give precise syntax and security definitions for the BiAS primitive. The guarantee offered by this notion allows us to extend the notions of traceability, non-frameability, and anonymity, borrowed from ordinary group signatures to our more general predicate-based primitive.

Secondly, we provide a generic BiAS realization where predicates may consist of polynomial-size circuits of a priori bounded depth. As building blocks, our construction relies on the homomorphic equivocal commitment (HEC) primitive defined by Katsumata *et al.* [27], dual-mode non-interactive zero-knowledge (NIZK) arguments [24,42,26], and a variant of the $\mathcal{R}$-lossy encryption primitive of Boyle *et al.* [14]. Our constructions are instantiable in the standard model for arbitrary polynomial-size Boolean circuits under the Learning-With-Errors (LWE) assumption [44]. For Boolean formulas (equivalently, NC^1 circuits), more efficient instantiations are possible under falsifiable assumptions in groups endowed with a bilinear map. In both cases, our schemes enjoy the property that the signature size only depends on the maximal circuit depth, and not on its size. The signature size is dominated by $O((\log N + |w|) \cdot \lambda^c)$ bits (where N is the group size and c is a constant) committing to the witness w and the user's identity together with NIZK arguments showing that the ciphertexts were properly generated.

1.2 Technical Overview

Defining Security. Our security model puts forth the notions of branch-hiding and branch-soundness for our bifurcated anonymous signature (BiAS) primitive. We advocate, more generally, that it is the first instance of a new fundamental notion of bifurcated cryptosystems (balancing based on a predicate in one scheme, both, user concerns and public safety issues). Further, to capture anonymity we extend the CCA-like notion of unlinkability of signatures which can be now produced from different predicate values. We call the resulting notion anonymity "in the traceable case" which implies the branch-hiding property of BiAS. We augment this anonymity notion with the anonymity "in the non-traceable case" where all the signatures are generated from a predicate value equals to 1, i.e., from the branch leading to unconditional anonymity. A BiAS is, then, called fully anonymous if it fulfills both anonymity notions. Branch-hiding and full anonymity, primarily, take care of privacy of BiAS.

To prevent misuse of the BiAS functionality, we build on two security notions from the Kiayias-Yung model [29] of group signatures. First, the security against *mis-identification attacks* (a.k.a. traceability) which requires that, even if the adversary can introduce users under its control in the group of signers, it cannot produce a signature that traces outside the dishonest coalition. Second, the notion of security against *framing attacks* which implies that honest users can never be falsely accused of having signed messages, even if the whole system conspires

against them. However, extending these security notions is not straightforward or immediate in our model, since we have to detect whether a given signature contradicts one of these properties even, if that signature is untraceable. Indeed, to build a reduction in a security proof, for instance, we have to figure out if a given untraceable signature has been generated honestly by a legitimate signer or if it is a forgery. However, being able to do so, in fact, seems to contradict statistical (unconditional) anonymity.

To circumvent the apparent incompatibility between privacy and security, we rely on our branch-soundness notion. It is a two-stage definition which first defines an extractable mode of the scheme only useful for the sake of proving security: it generates parameters of the scheme allowing to extract the identity and the witness behind *any* valid signatures. Such an extraction allows evaluating the predicate a posteriori given any signature. As a second stage, we require that the (real) tracing algorithm can be indistinguishably emulated from the (ideal) extractable mode, even when the authorities' keys are exposed. A BiAS satisfying branch-soundness thus ensures the hardness of "cheating" with the predicate, even for corrupt authorities. The reason is that it implies the infeasibility of producing signatures that: (i) can be traced while the context allows retaining statistical anonymity, i.e., $P(M, \mathsf{id}, w) = 1$, and conversely (ii) cannot be traced while the context allows retaining identity escrow, i.e., $P(M, \mathsf{id}, w) = 0$. We stress that the indistinguishability in branch-soundness cannot be statistical since, otherwise, untraceable signatures would no longer be statistically anonymous. Based on the branch-soundness notion, we can now extend the security notion of [29] from the (ideal) extractable mode of the BiAS.

Underlying primitives. Our construction for bounded-depth circuits is based on combining a number of primitives. First, it is built on the homomorphic equivocable commitments (HEC) of Katsumata *et al.* [27]. An HEC is a commitment scheme that allows committing to a message $\vec{x}$ using random coins R in such a way that anyone can publicly evaluate a circuit C over the commitment com to obtain an evaluated commitment $\mathsf{com}_{ev} = \mathsf{Eval}(C, \mathsf{com}_{ev})$ to $C(\vec{x})$. Using the pair $(\vec{x}, R)$, the committer can internally run a private evaluation algorithm over $(\vec{x}, R)$ in order to compute a proof π which will convince a verifier that com_{ev} is a commitment to $C(\vec{x})$. The primitive is instantiatable for all circuits via the fully homomorphic commitments of Gorbunov, Vaikuntanathan and Wichs [23], which in turn, is built on the FHE scheme of Gentry, Sahai and Waters (GSW) [22]. Katsumata *et al.* [27] also gave a construction for log-depth circuits under pairing-related assumptions [27]. In order to combine statistical anonymity in non-tracing mode and extractability in the security proofs, we employ a dual-mode NIZK argument, which either provides statistical zero-knowledge and computational soundness or vice versa, depending on the distribution of the common reference string. In the public verifiability setting, dual-mode NIZK is known to exist under standard assumptions in pairing-friendly groups, as shown by Groth, Ostrovsky and Sahai [24]. Peikert and Shiehian [42] (inspired by the earlier work of Canetti *et al.* [17]) recently gave constructions under the Learning-With-Errors (LWE) assumption. In order to smoothly interact with HEC schemes, the dual-mode

NIZK system makes use of dual-mode commitments [24], where the commitment key can be tuned to give either statistically hiding or extractable commitments.

Construction. At a high level, our construction proceeds as follows. When a user joins the group, he generates a fresh public key for a digital signature $\mathsf{pk}_{\mathsf{id}}$ and obtains from the group manager (GM) a membership certificate $\mathsf{cert}_{\mathsf{id}}$ consisting of the GM's signature on the pair $(\mathsf{id}, \mathsf{pk}_{\mathsf{id}})$, where id is the user's identity. In order to sign a message w.r.t. the predicate P, a group member computes an HEC commitment $\mathsf{com}_{(\mathsf{id},w)}$ to the witness w and his identity id. At the same time, the signer computes a dual-mode commitment to (id, w), which is configured to be statistically hiding in the real scheme. The signer then considers the message-dependent circuit $C_M(.,.)$ which evaluates $C_M(\mathsf{id}, w) = P(M, \mathsf{id}, w)$ on input of (id, w). He runs the private HEC evaluation algorithm to compute a proof $\pi_{C,M}$ that $\mathsf{com}_{ev} = \mathsf{Eval}(C_M, \mathsf{com}_{ev})$ is really a commitment to $C_M(\mathsf{id}, w)$. It finally computes a public key encryption $\mathsf{ct}_{\mathsf{id}} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, (1 - C_M(\mathsf{id}, w)) \cdot \mathsf{id})$ of the product $(1 - C_M(\mathsf{id}, w)) \cdot \mathsf{id}$, so that $\mathsf{ct}_{\mathsf{id}}$ encrypts 0 when unconditional anonymity is enabled (i.e., when $C_M(\mathsf{id}, w) = 1$) and id otherwise. A dual-mode NIZK argument then allows arguing that all steps were properly carried out.

When the tracing capability is enabled (namely, when $C_M(\mathsf{id}, w) = 0$), we need to rely on a special kind of dual-mode commitment to prove computational anonymity in the CCA sense (i.e., when the adversary has access to a signature opening oracle). Specifically, we need to program the commitment key to make commitments extractable in all signature opening queries and statistically hiding in the challenge phase. This is achieved using tag-based commitments, where each commitment is computed under a tag-dependent commitment key that either provides statistically hiding or extractable commitments, depending on the tag. In order to instantiate these dual-mode tag-based commitments, we use a recent variant [34] of the $\mathcal{R}$-lossy encryption of Boyle *et al.* [14], for which we give a DDH-based realization (as well as an LWE-based realization adapted from [34]). Using this $\mathcal{R}$-lossy encryption to instantiate the dual-mode commitment component, we can make it statistically hiding in the challenge phase (for a specific tag corresponding to a one-time verification key $\mathsf{VK}^{\star}$) and statistically extractable for all other tags $\mathsf{VK} \neq \mathsf{VK}^{\star}$. This allows us to proceed with a sequence of hybrid games where, instead of answering opening queries by decrypting the ciphertext $\mathsf{ct}_{\mathsf{id}}$, we can extract the committed (id, w). This can be seen as applying the two-key paradigm of Naor and Yung [40] for CCA2-secure encryption.

The above construction crucially relies on HEC to avoid the signature length from depending on the circuit size. If we were to give up the circuit-size independent property, constructions would be possible from any non-interactive statistically hiding commitment.

Open Questions. Naturally, our work, being in a new area (with some new technical and definitional challenges as described above), leaves many open problems which we believe to be interesting. Conceptually, one can ask how bifurcated cryptography applies elsewhere. Technically, the first problem that comes to mind is finding practically efficient instantiations from lattice assumptions (even in the random oracle model) as our LWE-based construction is only meant to be a first

feasibility result. In particular, it would be interesting to provide more efficient lattice-based solutions using, e.g., the Fiat-Shamir-with-abort method [37,10] or the techniques recently suggested in [49]. The second open problem is to determine the extent to which more specific predicate families can be realized more efficiently (with or without random oracles) and under different assumptions. While the work of Katsumata *et al.* [27] implies a pairing-based construction for Boolean formulas, it is only known to achieve circuit-size independence under a q-type (although falsifiable) assumption. A sufficient condition to avoid relying on variable-size assumptions in the pairing setting would be to build an HEC scheme based on simple assumptions, where the size of partial openings does not depend on the circuit size. Finally, while the rest of the paper will concentrate on BiAS, exploring further primitives providing point-wise predicate-based built-in privacy or confidentiality bifurcated tradeoffs seems like a new area for broader investigations.

1.3 Related Work

The rigid anonymity vs. accountability situation indeed generated much discomfort in the community. Group signatures traditionally allow opening authorities to identify the author of any signature. As advocated by Sakai *et al.* [46], it may be desirable to prevent the OA from seeing the entire signature history of all group members. Restricting the power of the opening authorities is a challenging research direction, where a few steps have been taken. For example, traceable signatures [28], group signatures with message-dependent opening [46], and accountable tracing signatures [33] can all be viewed as group signatures with restricted opening authorities. However, the OA can still freely break the anonymity of some subset of signatures without the user's agreement. These primitives offer more privacy to users than ordinary group signatures, but not at a level that privacy-sensitive users would hope for.

Ring signatures [45,6] confer everlasting anonymity to group members. They depart from group signatures in that signers are not required to register in the system and signers have complete freedom on the list of their ring-mates at each signature. Compared to group signatures, they stand at the opposite extreme of the spectrum as they do not provide any accountability at all. Linkable ring signature [36], traceable ring signatures [20], as well as k-time anonymous authentication systems [47] only introduce a weak form of accountability in ring signatures as users only lose anonymity to some extent: for example, if they issue two or more signatures for some message, their signatures become linkable but they can still create one controversial signature and disappear from the system without being caught.

Accountable tracing signatures (ATS) [32,33] take a different approach to balance accountability and anonymity, by allowing the two extreme settings of ring signatures and group signatures to co-exist. In ATS schemes, a given user is either always unconditionally anonymous or always traceable, based on a decision made by the authority when the user joins the system. In addition, users are never notified about their traceability status. In our setting and use-cases, the

ability to trace or not should depend point-wise on the message and not only on group members' identities. Moreover, we deliberately aim at leaving users some control on when and under which circumstances they want to accept traceability.

Accountable ring signatures (ARS) [48,9] provide another kind of tradeoff where anonymity and traceability can live together. Xu and Yung [48] consider a threshold opening mechanism where no single opener is given the entire power. The ARS model of Bootle *et al.* [9] also provides some flexibility in the choice of tracing authorities as signers are allowed to choose which opener they trust without necessarily leaving the full tracing capability to a pre-determined authority. On the other hand, neither of these models [48,9] provides full expressiveness as to which messages can be signed with unconditional anonymity and which ones should always be traceable.

Bangerter *et al.* [2] considered an informal framework allowing to monitor the release of certified data. Their (interactive) model fully trusts the opening authority to only disclose users' data when specific conditions are met. In contrast, BiAS does not trust the opener when de-anonymization conditions are not fulfilled and even requires unconditional anonymity in this case.

Boyen and Delerablée [12] introduced a variant of group signatures allowing group members to flexibly and expressively choose a subgroup wherein they hide their identity. Our goals are orthogonal to theirs since their model always allows tracing authorities to identify signers whereas we accurately control the conditions under which the signer's identity can be uncovered.

Garms and Lehmann [21] put forth the concept of convertably linkable signatures (CLS) which are group signatures where a "converter" can blindly relate a bunch of signatures to some randomized pseudonyms. To convert the given signatures into linkable ones, another authority first blinds the signatures in order to mitigate the power of the converter. However, the converter is actually an opening authority that can always trace a given signature as long as it was not blinded. At any time, CLS thus "only" provide computational anonymity.

Attribute-based signatures (ABS) [38] allow a signer to sign a message while simultaneously showing possession of credentials satisfying a public predicate. Policy-based signatures (PBS) [3] are signature schemes where users obtain a policy-based signing key (associated with some predicate P) from some authority, which allows them to sign exactly those messages M for which $P(M) = 1$. ABS and PBS address different problems than our BiAS primitive in that they provide fine-grained control over "who can sign" and "which messages can be signed at all", respectively. As such, they do not give users control over which signatures can be traced (with user knowledge of it). Our BiAS functionality departs from PBS [3] in that predicates are not associated with keys but with signed messages and may vary across signatures. In terms of generic implications, PBS were shown [3] to imply digital signatures, NIZK [7] and CCA-secure encryption [43]. However, they are not known to imply homomorphic equivocable commitments with circuit-size independent verification, and the relationship between PBS and BiAS thus remains unclear. In particular, we do not see any obvious way to obtain BiAS realizations by generically using a PBS.

Functional signatures [13] differ from conditionally traceable signatures in that, in the same spirit as PBS, they accurately control which messages can be signed. In contrast, BiAS controls which signatures can be traced.

In the context of anonymous compact e-cash [15], Camenisch *et al.* [16] were bothered by the anonymity vs. accountability issue in a specific scenario. They considered a conditional anonymity setting which restricts transactions to be untraceable only when they do not exceed a specific amount. In their model, the threshold amount is fixed for each merchant over multiple transactions. If a user performs a number of transactions with total values exceeding a threshold, he can be traced based on public records. On the other hand, if the total amount remains under the threshold, the traceable authority is unable to extract the user's identity. However, the anonymity remains computational even in that case. In contrast, our BiAS primitive is a generic add-on mechanism, and ensures the statistical anonymity of signers as long as the predicate equals 1, e.g., the amount does not reach a fixed threshold.

In summary, there was a lot of discomfort with the existing dichotomy between anonymity and accountability. This has produced a large number of interesting and useful cases and solutions. However, a bifurcated solution with choice at the user's hand folded into a single scheme, where the choice is driven by a local predicate which further remains undetected by others, has not been considered. This new BiAS system, which gives the user the best possible (i.e., unconditional) anonymity when permitted, in fact, constitutes the most private and the most versatile solution for the problem of balancing user's vs. society's needs within anonymous signing scenarios.

2 Preliminaries

2.1 $\mathcal{R}$-Lossy Public-Key Encryption

Boyle *et al.* [14] formalized the notion of $\mathcal{R}$-lossy encryption. The primitive is a tag-based encryption scheme [31] where the tag space $\mathcal{T}$ is partitioned into *injective* tags and *lossy* tags. When ciphertexts are generated under an injective tag, the decryption algorithm recovers the underlying plaintext. On lossy tags, the ciphertext statistically hides the underlying plaintext. In $\mathcal{R}$-lossy PKE schemes, the tag space is partitioned according to a binary relation $\mathcal{R} \subseteq \mathcal{K} \times \mathcal{T}$. The key generation algorithm inputs an initialization value $K \in \mathcal{K}$ and partitions $\mathcal{T}$ in such a way that injective tags $t \in \mathcal{T}$ are exactly those for which $(K, t) \in \mathcal{R}$ (i.e., all tags t for which $(K, t) \notin \mathcal{R}$ are lossy).

Libert *et al.* [34] considered a flavor of $\mathcal{R}$-lossy PKE schemes with two distinct key generation algorithms and equivocable lossy ciphertexts. For our purposes, we do not need to equivocate lossy ciphertexts but we still need two distinct key generation algorithms. Looking ahead, our proof of anonymity (in Lemma 4), requires to switch from a setting where all tags are lossy to a setting where only one tag is lossy. Also, proving other security notions requires to move from the "all lossy" setting to the "all injective" setting in the proof of Theorem 1.

Definition 1. *Let $\mathcal{R} \subseteq \mathcal{K}_\lambda \times \mathcal{T}_\lambda$ be an efficiently computable binary relation. An $\mathcal{R}$-lossy PKE scheme with efficient opening is a 5-uple of PPT algorithms* $(\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *such that:*

Parameter generation: *On input of a security parameter λ, a desired length of initialization values $L \in \mathsf{poly}(\lambda)$ and a lower bound $B \in \mathsf{poly}(\lambda)$ on the message length,* $\mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$ *outputs public parameters Γ that specify a tag space $\mathcal{T}$, a space of initialization values $\mathcal{K}$, a public key space $\mathcal{PK}$ and a secret key space $\mathcal{SK}$.*

Key generation: *For an initialization value $K \in \mathcal{K}$ and public parameters Γ, algorithm* $\mathsf{Keygen}(\Gamma, K)$ *outputs an injective public key $\mathsf{pk} \in \mathcal{PK}$ and a decryption key $\mathsf{sk} \in \mathcal{SK}$. The public key specifies a ciphertext space* CtSp *and a randomness space R^{RLE}.*

Lossy Key generation: *Given an initialization value $K \in \mathcal{K}$ and public parameters Γ, the lossy key generation algorithm* $\mathsf{LKeygen}(\Gamma, K)$ *outputs a lossy public key $\mathsf{pk} \in \mathcal{PK}$ and a lossy secret key $\mathsf{sk} \in \mathcal{SK}$.*

Decryption under injective tags: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, any $K \in \mathcal{K}$, any $t \in \mathcal{T}$ such that $(K, t) \in \mathcal{R}$, and any $\mathsf{Msg} \in \mathsf{MsgSp}$, we have*

$$\Pr\left[\exists r \in R^{\mathsf{RLE}} : \mathsf{Decrypt}\big(\mathsf{sk}, t, \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}; r)\big) \neq \mathsf{Msg}\right] < \nu(\lambda) \ ,$$

for some negligible function $\nu(\lambda)$, where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$ and the probability is taken over the randomness of Keygen.

Indistinguishability: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, the key generation algorithms* $\mathsf{LKeygen}$ *and* Keygen *satisfy the following:*

(i) For any $K \in \mathcal{K}$, the distributions $D_{\text{inj}} = \{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}(\Gamma, K)\}$ and $D_{\text{loss}} = \{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$ are computationally indistinguishable.

(ii) For any initialization values $K, K' \in \mathcal{K}$, the two distributions $\{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$ and $\{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{LKeygen}(\Gamma, K')\}$ are statistically indistinguishable.

Lossiness: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, any initialization value $K \in \mathcal{K}$ and tag $t \in \mathcal{T}$ such that $(K, t) \notin \mathcal{R}$, any $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$, and any $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$, the following distributions are statistically close:*

$$\{C \mid C \leftarrow \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_0)\} \quad \approx_s \quad \{C \mid C \leftarrow \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_1)\}.$$

For any $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$, the above holds for any tag t.

We will use an $\mathcal{R}$-lossy encryption scheme for the inequality relation.

Definition 2. *Let $\mathcal{K} = \{0,1\}^L$ and $\mathcal{T} = \{0,1\}^L \setminus \{\mathbf{0}^L\}$, for some $L \in \mathsf{poly}(\lambda)$. The* **inequality relation** *$\mathcal{R}_{\mathsf{NEQ}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$ is the relation for which we have $\mathcal{R}_{\mathsf{NEQ}}(K, t) = 1$ if and only if $K \neq t$.*

We note that, since we exclude the all-zeroes string $\mathbf{0}^L$ from $\mathcal{T}$, running Keygen on input of the initialization value $K = \mathbf{0}^L$ produces a key pk that is injective for *all* tags. In contrast $\mathsf{LKeygen}$ produces keys that are lossy for any tag and any initialization value.

We now give an $\mathcal{R}_{\mathsf{NEQ}}$-Lossy PKE realization under the Decision Diffie-Hellman (DDH) assumption (MDDH generalization is obvious). In the full version, we also provide a construction from the LWE assumption.

An $\mathcal{R}_{\mathsf{NEQ}}$-Lossy PKE Scheme from DDH. The construction below is inspired from [35], which is itself inspired from Groth-Sahai commitments [25] to scalars. We first recall the definition of the DDH problem.

Definition 3. *In a cyclic group $\mathbb{G}$ of prime order p, the* **Decision Diffie-Hellman Problem** *(DDH) in $\mathbb{G}$, is to distinguish the distributions (g, g^a, g^b, g^{ab}) and (g^a, g^b, g^c), with $a, b, c \hookleftarrow \mathbb{Z}_p$. The* **Decision Diffie-Hellman** *assumption is the intractability of DDH for any PPT algorithm $\mathcal{D}$.*

$\mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$: Define $\mathcal{K} = \{0,1\}^L$, and $\mathcal{T} = \{0,1\}^L \setminus \{\mathbf{0}^L\}$. Define public parameters as $\Gamma = (1^\lambda, 1^L, 1^B)$.

$\mathsf{Keygen}(\Gamma, K)$: On input of Γ and $K \in \mathcal{K}$, generate a key pair as follows.

1. Choose a cyclic group $\mathbb{G}$ or prime order $p > 2^\lambda$ with a generator $g \hookleftarrow U(\mathbb{G})$. Choose $\alpha \hookleftarrow U(\mathbb{Z}_p)$ and compute $h = g^\alpha$. Define $\vec{g}_0 = (g, h) \in \mathbb{G}^2$ and $\vec{g} = \vec{g}_0^{\,\beta} \cdot (1, g) \in \mathbb{G}^2$, where $\beta \hookleftarrow U(\mathbb{Z}_p)$.
2. Pick $\gamma \hookleftarrow U(\mathbb{Z}_p)$ and compute $\vec{u} = \vec{g}_0^{\,\gamma} \cdot \vec{g}^{-K} \in \mathbb{G}^2$, where $K \in \{0,1\}^L$ is interpreted as an element of $\mathbb{Z}_p$.

Define $R^{\mathsf{RLE}} = \mathbb{Z}_p^B$ and output $\mathsf{sk} = (\alpha, K)$ as well as $\mathsf{pk} := \left(\mathbb{G}, \vec{g}_0, \vec{g}, \vec{u}\right)$.

$\mathsf{LKeygen}(\Gamma, K)$: This algorithm is identical to Keygen with the difference that, at step 1, it computes $\vec{g}$ as $\vec{g} = \vec{g}_0^{\,\beta} \in \mathbb{G}^2$, where $\beta \hookleftarrow U(\mathbb{Z}_p)$. It defines $R^{\mathsf{RLE}} = \mathbb{Z}_p^B$ and outputs $\mathsf{sk} = (\alpha, K)$ as well as $\mathsf{pk} := \left(\mathbb{G}, \vec{g}_0, \vec{g}, \vec{u}\right)$.

$\mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg})$: To encrypt $\mathsf{Msg} \in \{0,1\}^B$, interpret the tag $t \in \mathcal{T}$ as an element of $\mathbb{Z}_p$. For each index $i \in [B]$, pick $r_i \hookleftarrow U(\mathbb{Z}_p^*)$ and compute $\mathsf{ct}_i = \left(\vec{u} \cdot \vec{g}^t\right)^{\mathsf{Msg}[i]} \cdot \vec{g}_0^{\,r_i} \in \mathbb{G}^2$. Then, output $\mathsf{ct} = (\mathsf{ct}_1, \ldots, \mathsf{ct}_B) \in \mathbb{G}^{2B}$.

$\mathsf{Decrypt}(\mathsf{sk}, t, \mathsf{ct})$: Given $\mathsf{sk} = (\alpha, K)$ and $t \in \{0,1\}^L$, interpret t as an element of $\mathbb{Z}_p$. If $t = K$, return $\perp$. Otherwise, for each $i \in [B]$, do the following:

1. Parse ct_i as $(\mathsf{ct}_{i,1}, \mathsf{ct}_{i,2}) \in \mathbb{G}^2$
2. Set $\mathsf{Msg}[i] = 0$ if $\mathsf{ct}_{i,2} = \mathsf{ct}_{i,1}^\alpha$ and $\mathsf{Msg}[i] = 1$ otherwise.

Output $\mathsf{Msg} \in \{0,1\}^B$.

The proof of Lemma 1 is straightforward. The first indistinguishability property follows immediately from the semantic security of ElGamal and the observation that Keygen and $\mathsf{LKeygen}$ only differ in the distribution of $\vec{g}$. The second indistinguishability property follows from the fact that, for any $K \in \mathcal{K}$ and any public key pk generated by $\mathsf{LKeygen}$, $\vec{u} \in \mathbb{G}^2$ is uniformly distributed in the subspace

spanned by $\vec{g}_0$. The same holds for any ciphertext encrypted under an injective key for the lossy tag $t = K$ or a lossy key for any tag. The construction readily extends to rely on the k-linear assumption [8] for $k > 1$.

Lemma 1. *The above construction is an $\mathcal{R}_{\mathsf{NEQ}}$-lossy public-key encryption scheme under the* DDH *assumption.*

2.2 Homomorphic Equivocal Commitments

We now recall the definition of homomorphic equivocal commitment, as formalized by Katsumata *et al.* [27].

Definition 4. *A HEC scheme with message space $\mathcal{X}$, randomness space R^{HEC} and randomness distribution $\mathcal{D}^{\mathsf{HEC}}$ over R^{HEC} for a circuit class $\mathcal{C} = \{C : \mathcal{X} \to \{0,1\}\}$ is a tuple of PPT algorithms* $\mathsf{HEC} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval}^{in}, \mathsf{Eval}^{out}, \mathsf{Verify})$ *with the following specifications:*

Setup(1^λ)**:** *Inputs a security parameter 1^λ and outputs public parameters* pp*, an evaluation key* ek *and a master secret key* msk*.*

Commit$(\mathsf{pp}, \vec{x}, r)$**:** *Takes as input public parameters* pp*, a message $\vec{x} \in \mathcal{X}$ and randomness $r \in R^{\mathsf{HEC}}$. It outputs a commitment* com*. When r is omitted from the notation* $\mathsf{Commit}(\mathsf{pp}, \vec{x})$*, we man that r is sampled from $\mathcal{D}^{\mathsf{HEC}}$.*

Open$(\mathsf{msk}, \vec{x}, r, \vec{x}')$**:** *Takes as input a master secret key* msk*, messages $\vec{x}, \vec{x}' \in \mathcal{X}$, and randomness $r \in R^{\mathsf{HEC}}$. It outputs fake randomness $r' \in R^{\mathsf{HEC}}$.*

Eval$^{in}(\mathsf{ek}, C, \vec{x}, r)$**:** *The inner evaluation algorithm inputs a key* ek*, a circuit $C \in \mathcal{C}$, a message $\vec{x}$ and randomness $r \in R^{\mathsf{HEC}}$. It outputs a proof π.*

Eval$^{out}(\mathsf{ek}, C, \mathsf{com})$**:** *The outer evaluation algorithm is a deterministic algorithm that inputs an evaluation key* ek*, a circuit $C \in \mathcal{C}$ and a commitment* com*. It outputs an evaluated commitment* com_{ev}*.*

Verify$(\mathsf{pp}, \mathsf{com}_{ev}, z, \pi)$**:** *The verification algorithm takes as input public parameters* pp*, an evaluated commitment* com_{ev}*, a message $z \in \{0,1\}$, and a proof π. It outputs 0 or 1.*

In addition, it should satisfy the following properties:

Evaluation correctness: For all $\lambda \in \mathbb{N}$, all $(\mathsf{pp}, \mathsf{ek}, \mathsf{msk}) \leftarrow \mathsf{HEC.Setup}(1^\lambda)$, any input $\vec{x} \in \mathcal{X}$, any randomness $r \in R^{\mathsf{HEC}}$, and any circuit $C \in \mathcal{C}$, if $\mathsf{com} = \mathsf{HEC.Commit}(\mathsf{pp}, \vec{x}; r)$, $\pi \leftarrow \mathsf{HEC.Eval}^{in}(\mathsf{msk}, C, \vec{x}, r)$, and

$$\mathsf{com}_{ev} = \mathsf{HEC.Eval}^{out}(\mathsf{ek}, C, \mathsf{com}),$$

then $\Pr[\mathsf{HEC.Verify}(\mathsf{pp}, \mathsf{com}_{ev}, C(\vec{x}), \pi) = 1] \geq 1 - \nu(\lambda)$, for some function $\nu(\lambda) \in \mathsf{negl}(\lambda)$.

Distributional equivalence of Open: For all $\lambda \in \mathbb{N}$, any $(\mathsf{pp}, \mathsf{ek}, \mathsf{msk}) \leftarrow \mathsf{HEC.Setup}(1^\lambda)$, any $\vec{x}, \bar{\vec{x}} \in \mathcal{X}$, randomness $r, r' \hookleftarrow \mathcal{D}^{\mathsf{HEC}}$, the distributions $\{(\mathsf{pp}, \mathsf{ek}, \mathsf{msk}, \vec{x}, r, \mathsf{com}) \mid r \hookleftarrow \mathcal{D}^{\mathsf{HEC}},\ \mathsf{com} = \mathsf{HEC.Commit}(\mathsf{pp}, \vec{x}; r)\}$ and

$$\{(\mathsf{pp}, \mathsf{ek}, \mathsf{msk}, \vec{x}, r' = \mathsf{HEC.Open}(\mathsf{msk}, \bar{\vec{x}}, \bar{r}, \vec{x}), \mathsf{com}') \mid \bar{r} \hookleftarrow \mathcal{D}^{\mathsf{HEC}},\ \mathsf{com}' = \mathsf{HEC.Commit}(\mathsf{pp}, \bar{\vec{x}}; \bar{r})\}$$

are statistically close.

Computational binding on evaluated commitments: For any PPT adversary $\mathcal{A}$, we have

$$\Pr\Big[\mathsf{HEC.Verify}(\mathsf{pp}, \mathsf{com}_{ev}, z^\star, \pi^\star) = 1 \ \wedge \ z^\star \neq C(\vec{x}) \Big| \begin{array}{r} (\mathsf{pp}, \mathsf{ek}, \mathsf{msk}) \leftarrow \mathsf{HEC.Setup}(1^\lambda), \\ (\vec{x}, r, C, z^\star, \pi^\star) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{ek}), \\ \mathsf{com} = \mathsf{HEC.Commit}(\mathsf{pp}, \vec{x}; r), \\ \mathsf{com}_{ev} = \mathsf{HEC.Eval}^{out}(\mathsf{ek}, C, \mathsf{com}) \end{array} \Big] \in \mathsf{negl}(\lambda)$$

Efficient committing: There exists a polynomial $\mathsf{poly}(\lambda)$ such that, for all $(\mathsf{pp}, \mathsf{ek}, \mathsf{msk}) \leftarrow \mathsf{HEC.Setup}(1^\lambda)$, $\vec{x} \in \mathcal{X}$ and $r \in R^{\mathsf{HEC}}$, the running time of $\mathsf{com} = \mathsf{HEC.Commit}(\mathsf{pp}, \vec{x}; r)$ is bounded by $|\vec{x}| \cdot \mathsf{poly}(\lambda)$.

Efficient verification: There exists a polynomial $\mathsf{poly}(\lambda)$ such that, for all $(\mathsf{pp}, \mathsf{ek}, \mathsf{msk}) \leftarrow \mathsf{HEC.Setup}(1^\lambda)$ and any $\vec{x} \in \mathcal{X}$, $r \in R^{\mathsf{HEC}}$, $C \in \mathcal{C}$ and $z \in \{0,1\}$, if $\mathsf{com} = \mathsf{HEC.Commit}(\mathsf{pp}, \vec{x}; r)$, $\pi \leftarrow \mathsf{HEC.Eval}^{in}(\mathsf{msk}, C, \vec{x}, r)$ and $\mathsf{com}_{ev} = \mathsf{HEC.Eval}^{out}(\mathsf{ek}, C, \mathsf{com})$, then $|\pi|, |\mathsf{com}_{ev}| \leq \mathsf{poly}(\lambda)$ and the running time of $\mathsf{HEC.Verify}(\mathsf{pp}, \mathsf{com}_{ev}, z, \pi)$ is at most $\mathsf{poly}(\lambda)$ (which does not depend on $|C|$).

Context hiding: There exists a PPT simulator $\mathsf{HEC.ProofSim}$ such that, for all $\lambda \in \mathbb{N}$, $(\mathsf{pp}, \mathsf{ek}, \mathsf{msk}) \leftarrow \mathsf{HEC.Setup}(1^\lambda)$, $\vec{x} \in \mathcal{X}$, $C \in \mathcal{C}$, $r \in R^{\mathsf{HEC}}$ and $\mathsf{com} = \mathsf{HEC.Commit}(\mathsf{pp}, \vec{x}; r)$, the distribution $\{\pi \leftarrow \mathsf{HEC.Eval}^{in}(\mathsf{msk}, C, \vec{x}, R)\}$ is statistically indistinguishable from $\{\pi' \leftarrow \mathsf{HEC.ProofSim}(\mathsf{msk}, \mathsf{com}, C, C(\vec{x}))\}$.

We note that the distributional equivalence of Open implies that the commitment is statistically hiding. Here, we only need the statistically hiding property and we do not rely on equivocation. We do not explicitly rely on the context hiding property either since partial openings π produced by Open will be part of witnesses in a NIZK argument. On the other hand, we will exploit the efficient verification property to achieve circuit-size independence in terms of signature size.[6]

3 Bifurcated Anonymous Signatures

This section formalizes the primitive of bifurcated anonymous signature (BiAS). BiAS is the first general signature primitive reconciling statistical anonymity and accountability in front of dishonest authorities in a single scheme. Nevertheless, our model and syntax are inspired by those in the context of dynamic group signatures given by Kiayias and Yung [29] —who extended the work of Bellare, Micciancio and Warinschi [4] on static group signatures.

3.1 Syntax

Like in [29], we consider dynamically growing groups. The syntax includes an interactive protocol which allows users to be enrolled as new members of the

[6] As pointed out in [27, Remark 3.3], it is straightforward to build an HEC without the context-hiding and efficient verification properties, using any statistically hiding commitment.

group at any time. Analogously to the similar model of Bellare, Shi and Zhang [5], the Kiayias-Yung (KY) model assumes an interactive *join* protocol whereby a prospective user becomes a group member by interacting with the group manager responsible to issuing credentials. This protocol provides the user with a membership certificate, cert_i, and a membership secret, sec_i.

In the syntax below, we define a space $\mathcal{ID}$ of user identifiers as well as a space of witnesses $\mathcal{W}$. For any message-identity-witness triple $(M, \mathsf{id}, w) \in \mathcal{M} \times \mathcal{ID} \times \mathcal{W}$, we adopt the convention that $P(M, \mathsf{id}, w) = 0$ whenever the user of identity id was only allowed to sign M using the witness w while being subject to tracing, by the opening authority. In contrast, having $P(M, \mathsf{id}, w) = 1$ allows the user to use the witness w to generate a signature on M while retaining unconditional anonymity. For generality and more applicability, a BiAS defines a family $\mathcal{P}$ of authorized *public* predicates P which are needed to verify signatures.

Definition 5 (Bifurcated Anonymous Signature). *A* bifurcated anonymous signature *(BiAS) scheme for a predicate family $\mathcal{P}$ consists of the following algorithms or protocols.*

Setup$(1^\lambda, 1^N)$**:** *given a security parameter λ and a maximal number of group members $N \in \mathsf{poly}(\lambda) \cap \mathbb{N}$, this algorithm is run by a trusted party to generate a group public key $\mathcal{Y}$ associated to the predicate family $\mathcal{P}_\lambda$, which specifies a message space $\mathcal{M}$, a space of user identifiers $\mathcal{ID}$, and a witness space $\mathcal{W}$. It also outputs the group manager's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$. Each key is given to the appropriate authority while $\mathcal{Y}$ is made public. The algorithm also initializes a public state St comprising a set data structure $St_{\mathsf{users}} = \emptyset$ and a string data structure $St_{\mathsf{trans}} = \epsilon$. From now on we assume λ is implicit.*

Join: *is an* interactive *protocol between the group manager GM and a user $\mathcal{U}$ who gets a unique identifier $\mathsf{id} \in \mathcal{ID}$ (both are responsible for enforcing the uniqueness of the identifier). The protocol involves two interactive Turing machines $\mathsf{J}_{\mathsf{user}}$ and J_{GM} that both take $\mathcal{Y}$ as input. The execution, denoted as $[\mathsf{J}_{\mathsf{user}}(\mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, St)]$, ends with user $\mathcal{U}_{\mathsf{id}} := \mathcal{U}$ obtaining a membership secret $\mathsf{sec}_{\mathsf{id}}$, that no one else knows, and a membership certificate $\mathsf{cert}_{\mathsf{id}}$. If the protocol is successful, the group manager updates the public state St by setting $St_{\mathsf{users}} := St_{\mathsf{users}} \cup \{\mathsf{id}\}$ as well as $St_{\mathsf{trans}} := St_{\mathsf{trans}} || \langle \mathsf{id}, \mathsf{transcript}_{\mathsf{id}} \rangle$.*

Sign($\mathsf{id}, \mathsf{cert}_{\mathsf{id}}, \mathsf{sec}_{\mathsf{id}}, M, w, P$**):** *given an identifier $\mathsf{id} \in \mathcal{ID}$, a membership certificate $\mathsf{cert}_{\mathsf{id}}$, a membership secret $\mathsf{sec}_{\mathsf{id}}$, a message $M \in \mathcal{M}$, a witness $w \in \mathcal{W}$, and a predicate $P \in \mathcal{P}$, this* probabilistic *algorithm outputs a signature σ.*

Verify($\mathcal{Y}, M, \sigma, P$**):** *given a message $M \in \mathcal{M}$, a signature σ, a predicate $P \in \mathcal{P}$ and a group public key $\mathcal{Y}$, this* deterministic *algorithm returns either 0 or 1.*

Open($\mathcal{Y}, \mathcal{S}_{\mathsf{OA}}, M, \sigma, P, St$**):** *takes as input the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$, a message $M \in \mathcal{M}$, a signature σ w.r.t. $\mathcal{Y}$ and a predicate $P \in \mathcal{P}$ as well as the public state St. It outputs $\mathsf{id} \in St_{\mathsf{users}} \cup \{\perp\}$, which is the identity of a group member or a symbol indicating anonymity.*

Correctness basically requires that, if all parties *honestly* run the protocols, all algorithms are correct with respect to their specification described as above.

Correctness of Bifurcated Anonymous Signatures. Following the Kiayias-Yung terminology [29], we say that a public state St is *valid* if it can be reached from $St = (\emptyset, \epsilon)$ by a Turing machine having oracle access to J_{GM}. Also, a state St' is said to *extend* another state St if it is within reach from St. Moreover, we write $\mathsf{cert}_{\mathsf{id}} \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_{\mathsf{id}}$ to mean that there exists coin tosses ϖ for J_{GM} and $\mathsf{J}_{\mathsf{user}}$ such that, for some valid public state St', the execution of the interactive protocol $[\mathsf{J}_{\mathsf{user}}(\mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, St')](\varpi)$ provides $\mathsf{J}_{\mathsf{user}}$ with $\langle \mathsf{id}, \mathsf{sec}_{\mathsf{id}}, \mathsf{cert}_{\mathsf{id}} \rangle$.

Definition 6 (Correctness). *A BiAS scheme is correct if the following conditions are all satisfied for any* $(St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$*:*

(1) In a valid state St*,* $|St_{\mathsf{users}}| = |St_{\mathsf{trans}}|$ *always holds and two distinct entries of* St_{trans} *always contain certificates with distinct* id*.*
(2) If $[\mathsf{J}_{\mathsf{user}}(\mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, St)]$ *is run by two honest parties following the protocol and* $\langle \mathsf{id}, \mathsf{cert}_{\mathsf{id}}, \mathsf{sec}_{\mathsf{id}} \rangle$ *is obtained by* $\mathsf{J}_{\mathsf{user}}$*, then we have* $\mathsf{cert}_{\mathsf{id}} \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_{\mathsf{id}}$*.*
(3) For each $\langle \mathsf{id}, \mathsf{cert}_{\mathsf{id}}, \mathsf{sec}_{\mathsf{id}} \rangle$ *such that* $\mathsf{cert}_{\mathsf{id}} \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_{\mathsf{id}}$*, any message* $M \in \mathcal{M}$*, any witness* $w \in \mathcal{W}$ *and any predicate* $P \in \mathcal{P}$*, we have*

$$\mathsf{Verify}\big(\mathcal{Y}, M, \mathsf{Sign}(\mathsf{id}, \mathsf{cert}_{\mathsf{id}}, \mathsf{sec}_{\mathsf{id}}, M, w, P), P\big) = 1.$$

(4) For any $M \notin \mathcal{M}$ *or any* $P \notin \mathcal{P}$*, and any* σ*, we have* $\mathsf{Verify}(\mathcal{Y}, M, \sigma, P) = 0$*.*
(5) $\mathsf{Open}(\mathcal{Y}, \mathcal{S}_{\mathsf{OA}}, M, \sigma, P, St) \in \mathcal{ID} \cup \{\perp\}$ *as long as* $\mathsf{Verify}(\mathcal{Y}, M, \sigma, P) = 1$*.*
(6) For any outcome $\langle \mathsf{id}, \mathsf{cert}_{\mathsf{id}}, \mathsf{sec}_{\mathsf{id}} \rangle$ *of* $[\mathsf{J}_{\mathsf{user}}(.,.), \mathsf{J}_{\mathsf{GM}}(., St, ., .)]$*, for some valid* St*, any predicate* $P \in \mathcal{P}$*, any message* $M \in \mathcal{M}$*, any witness* $w \in \mathcal{W}$ *and* $\sigma \leftarrow \mathsf{Sign}(\mathsf{id}, \mathsf{cert}_{\mathsf{id}}, \mathsf{sec}_{\mathsf{id}}, M, w, P)$*, with overwhelming probability:*
(a) if $P(M, \mathsf{id}, w) = 0$*, then* $\mathsf{Open}(\mathcal{Y}, \mathcal{S}_{\mathsf{OA}}, M, \sigma, P, St) = \mathsf{id}$*;*
(b) if $P(M, \mathsf{id}, w) = 1$*, then* $\mathsf{Open}(\mathcal{Y}, \mathcal{S}_{\mathsf{OA}}, M, \sigma, P, St) = \perp$*.*

We formalize security properties via experiments where the adversary interacts with a stateful interface $\mathcal{I}$ that maintains the following variables:

- $\mathsf{state}_{\mathcal{I}}$: is a data structure representing the state of the interface as the adversary invokes the various oracles available in the attack games. It is initialized as $\mathsf{state}_{\mathcal{I}} = (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$. It includes the (initially empty) set St_{users} of group members and a dynamically growing database St_{trans} storing the transcripts of previously executed join protocols.
- $n = |St_{\mathsf{users}}| \leq N$ denotes the current cardinality of the group.
- Sigs: is a database of honestly generated signatures created by the signing oracle. Each entry consists of a tuple $(\mathsf{id}, M, w, \sigma, P)$ indicating that message M was signed by user id with respect to the witness w and the predicate P.
- U^a: is an initially empty set of users that are introduced by the adversary in the system in an execution of the join protocol.
- U^b: is an initially empty set of honest users introduced in the system by the adversary acting as a dishonest group manager. For these users, the adversary obtains the transcript of $[\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}]$ but not the user's membership secret.

In attack games, adversaries are granted access to the following oracles:

- Q_{pub}, Q_{keyGM} and Q_{keyOA}: when these oracles are invoked, the interface looks up $\mathsf{state}_{\mathcal{I}}$ and returns the group public key $\mathcal{Y}$, the GM's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ respectively. Once the oracle Q_{keyGM} (resp. Q_{keyOA}) is invoked, it updates the initially empty key state $St_{\mathsf{GM}} \leftarrow \{\mathcal{S}_{\mathsf{GM}}\}$ (resp. $St_{\mathsf{OA}} \leftarrow \{\mathcal{S}_{\mathsf{OA}}\}$).
- $Q_{\mathsf{a\text{-}join}}$: allows the adversary to introduce users under its control in the group. On behalf of the GM, the interface runs J_{GM} in interaction with the $\mathsf{J}_{\mathsf{user}}$-executing adversary who plays the role of the prospective user in the join protocol. At the beginning of $\mathsf{J}_{\mathsf{user}}$, the user chooses an identifier id and the interface aborts if id was previously assigned to a different user in U^a. If this protocol successfully ends, the interface updates St by inserting the new user id in both sets St_{users} and U^a. It also sets $St_{\mathsf{trans}} := St_{\mathsf{trans}}||\langle \mathsf{id}, \mathsf{transcript}_{\mathsf{id}} \rangle$.
- $Q_{\mathsf{b\text{-}join}}$: allows the adversary, acting as a corrupted group manager, to introduce new honest group members of its choice. The interface triggers an execution of $[\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}]$ and runs $\mathsf{J}_{\mathsf{user}}$ in interaction with the adversary who runs J_{GM}. If the protocol successfully completes, the interface adds user id to St_{users} and U^b and sets $St_{\mathsf{trans}} := St_{\mathsf{trans}}||\langle \mathsf{id}, \mathsf{transcript}_{\mathsf{id}} \rangle$. It stores the membership certificate $\mathsf{cert}_{\mathsf{id}}$ and the membership secret $\mathsf{sec}_{\mathsf{id}}$ in a *private* part of $\mathsf{state}_{\mathcal{I}}$.
- Q_{sig}: given a tule (M, w, P) and an identifier id, the interface returns $\perp$ if $\mathsf{id} \notin U^b$. Otherwise, the private area of $\mathsf{state}_{\mathcal{I}}$ must contain a certificate $\mathsf{cert}_{\mathsf{id}}$ and a membership secret $\mathsf{sec}_{\mathsf{id}}$. The interface outputs a signature σ on behalf of user id and also updates $\mathsf{Sigs} \leftarrow \mathsf{Sigs}||(\mathsf{id}, M, w, \sigma, P)$.
- Q_{open}: when this oracle is invoked on input of a valid triple (M, σ, P), the interface runs algorithm Open using the current state St. When S is a set of tuples of the form (M, σ, P), $Q_{\mathsf{open}}^{\neg S}$ denotes a restricted oracle that only applies the opening algorithm to tuples (M, σ, P) which are not in S.
- Q_{read} and Q_{write}: are used by the adversary to read and write the content of St. At each invocation, Q_{read} outputs the state St of the interface. By using Q_{write}, the adversary can modify St at will as long as it does not invalidate St: for example, the adversary is allowed to create dummy users as long as it does not re-use already existing certificates.

In the random oracle model we implicitly assume that all the BiAS algorithms and protocols have access to the random oracle.

3.2 Branch-Hiding and Privacy

Branch-Hiding. The notion of branch-hiding captures the infeasibility, even for a corrupt group manager, to decide whether a user signs a message M for a given predicate P while enabling traceability or not. In particular, $P(M, \mathsf{id}, w)$ remains computationally hidden. Said otherwise, signatures do not betray any potential intent of a user to remain untraceable or accept traceability. The formal description is given in the full version as we require a stronger privacy notion.

Full Anonymity. The notion of anonymity is formalized via two games parametrized by a bit d and involving a two-stage adversary. The first stage is called play

stage and allows the adversary $\mathcal{A}$ to modify $\mathsf{state}_{\mathcal{I}}$ via Q_{write}-queries and open arbitrary signatures by probing Q_{open}. When the play stage ends, $\mathcal{A}$ chooses a message-predicate pair $(M^\star, P^\star)$ as well as two 4-ules $(\mathsf{id}_0^\star, w_0^\star, \mathsf{sec}_0^\star, \mathsf{cert}_0^\star)$ and $(\mathsf{id}_1^\star, w_1^\star, \mathsf{sec}_1^\star, \mathsf{cert}_1^\star)$, both containing a valid membership certificate and a corresponding membership secret. Then, depending on $d \in \{0,1\}$, the adversary is given a challenge signature $\sigma^\star$ computed using $(\mathsf{id}_d^\star, w_d^\star, \mathsf{sec}_d^\star, \mathsf{cert}_d^\star)$ with the task of eventually guessing the bit $d \in \{0,1\}$. Before doing so, it is allowed further oracle queries throughout the second stage, called guess stage, but is restricted not to query Q_{open} for $(M^\star, \sigma^\star, P^\star)$. We note that the adversary is allowed to choose $(\mathsf{id}_0^\star, \mathsf{sec}_0^\star, \mathsf{cert}_0^\star)$ and $(\mathsf{id}_1^\star, \mathsf{sec}_1^\star, \mathsf{cert}_1^\star)$ such that $P^\star(M^\star, \mathsf{id}_0^\star, w_0^\star) \neq P^\star(M^\star, \mathsf{id}_1^\star, w_1^\star)$. Our definition of anonymity thus reflects the inability of a verifier to distinguish signatures that are traceable from those that are not. To strengthen the model, the definition even allows the adversary to corrupt the opening authority as long as $P^\star(M^\star, \mathsf{id}_0^\star, w_0^\star) = 1 = P^\star(M^\star, \mathsf{id}_1^\star, w_1^\star)$. In such a non-traceable case, we require that the indistinguishability is statistically independent of the bit d. We elaborate more on this adversarial complexity just after.

Definition 7. *A BiAS is fully anonymous if it satisfies the next conditions:*

Traceable case *For any* PPT *adversary* $\mathcal{A}$ *the following advantage is negligible.*

$$\mathbf{Adv}^{\text{anon}}_{\mathcal{A},N}(\lambda) := \left|\Pr\left[\mathbf{Exp}^{\text{anon-}1}_{\mathcal{A},N}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{anon-}0}_{\mathcal{A},N}(\lambda) = 1\right]\right|$$

Non-traceable case *For any (unbounded) adversary involved in* $\mathbf{Exp}^{\text{anon-}d}_{\mathcal{A},N}$ *and* $\mathbf{Exp}^{\text{anon-ntr-}d}_{\mathcal{A},N}$ *(defined in Fig. 1), the following advantage is negligible.*

$$\mathbf{Adv}^{\text{anon-ntr}}_{\mathcal{A},N}(\lambda) := \left|\Pr\left[\mathbf{Exp}^{\text{anon-ntr-}1}_{\mathcal{A},N}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{anon-ntr-}0}_{\mathcal{A},N}(\lambda) = 1\right]\right|$$

1 $\mathsf{state}_{\mathcal{I}} := (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$;
2 $\big(aux, M^\star, w_0^\star, w_1^\star, (\mathsf{id}_0^\star, \mathsf{sec}_0^\star, \mathsf{cert}_0^\star), (\mathsf{id}_1^\star, \mathsf{sec}_1^\star, \mathsf{cert}_1^\star), P^\star\big)$
$\leftarrow \mathcal{A}(\mathsf{play};\, Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{open}}, Q_{\mathsf{read}}, Q_{\mathsf{write}}, Q_{\mathsf{keyOA}})$;
3 **if** $\neg\big(\mathsf{cert}_{\mathsf{id}_0^\star} \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_{\mathsf{id}_0^\star}\big) \vee \neg\big(\mathsf{cert}_{\mathsf{id}_1^\star} \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_{\mathsf{id}_1^\star}\big)$ **then return** 0;
4 **if** $\big(M^\star \notin \mathcal{M}\big) \vee \big(w_0^\star, w_1^\star \notin \mathcal{W}\big) \vee \big(P^\star \notin \mathcal{P}\big)$ **then return** 0;
5 $\sigma_d \leftarrow \mathsf{Sign}(\mathcal{Y}, \mathsf{id}_d^\star, \mathsf{cert}_d^\star, \mathsf{sec}_d^\star, M^\star, w_d^\star, P^\star)$;
6 $d' \leftarrow \mathcal{A}(\mathsf{guess};\, \sigma_d, aux, Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{open}}^{\neg\{(M^\star, \sigma_d, P^\star)\}}, Q_{\mathsf{read}}, Q_{\mathsf{write}}, Q_{\mathsf{keyOA}})$;
7 **if** $\big(P^\star(M^\star, \mathsf{id}_0^\star, w_0^\star) = 1\big) \wedge \big(P^\star(M^\star, \mathsf{id}_1^\star, w_1^\star) = 1\big)$ **then return** d';
8 **if** $\big(St_{\mathsf{OA}} = \emptyset\big)$ **then return** d';
9 **return** 0;

Fig. 1: Experiment $\mathbf{Exp}^{\text{anon-}d}_{\mathcal{A},N}(\lambda)$ (*resp.* $\mathbf{Exp}^{\text{anon-ntr-}d}_{\mathcal{A},N}(\lambda)$) excluding the dotted (*resp.* solid) box.

The anonymity definition has two parts: a first one that captures the (CCA) unlinkability against all entities but the OA regardless of the predicate value; and a second one which captures the unlinkability even against the OA when the predicate evaluates to 1. Clearly, the first case can never be statistical if the predicate is not constantly equal to 1. However, while the requirement of the second case could only have been computational, we stress that having two cases has nothing to do with the running time of the adversary.

The reason we are requiring statistical anonymity is because we advocate the need to enhance the privacy branch in the context of anonymous signatures. When the predicate equals 1, the signer should have full confidence in his anonymity. Allowing computational anonymity leaves room for a potential backdoor in the system, which could be exploitable in an unexpected way in some applications.

In the full version of this paper, we suggest an even stronger notion of anonymity, called unsubversive anonymity, in the non-traceable case. This notion allows for adversarially-generated authorities' keys. Since it only seems achievable in the random oracle model, we do not include it in the general BiAS model and leave the design of a BiAS achieving it for future research.

3.3 Branch-Soundness and Security

Defining strong unforgeability-related notions requires being able to check whether an adversary fools the underlying predicate value embedded into signatures. However, checking such a relation needs extracting meaningful information even from (statistically) non-traceable signatures. To circumvent this apparent conflicting requirements we first define the branch-soundness notion which sets an indistinguishable extractable mode even if all the keys are exposed. It also captures the inability of any efficient adversary to produce valid signatures in the extractable mode that contradict the openability of signatures in the real mode. Equipped with a setting where identities and witnesses are extractable "all the time" we can turn to other security notions.

Branch-Soundness. To be able to extract (id, w) from *any* valid signatures, we introduce an indistinguishable setting allowing such extractions for the purpose of testing the underlying predicate value $P(M, \mathsf{id}, w)$. As long as signatures are traceable, we require id to be consistent with the outcome of Open.

Definition 8. *A BiAS scheme satisfies the* branch-soundness *property if there is a pair of efficient algorithms with the following specifications:*

SimSetup$(1^\lambda, 1^N)$**:** *given a security parameter λ and a maximal number of users $N \in \mathrm{poly}(\lambda) \cap \mathbb{N}$, this algorithm generates a group public key $\mathcal{Y}$, the group manager's secret key $\mathcal{S}_{\mathsf{GM}}$, the opening authority's secret key $\mathcal{S}_{\mathsf{OA}}$ as well as an extraction trapdoor τ_{ext}. The algorithm also initializes a public state $St = (St_{\mathsf{users}}, St_{\mathsf{trans}}) := (\emptyset, \epsilon)$ as in* Setup*;*

Extract$(\mathcal{Y}, \tau_{\mathsf{ext}}, M, \sigma, P, St)$**:** *inputs a valid message-signature pair (M, σ) w.r.t. $\mathcal{Y}$ and a predicate $P \in \mathcal{P}$, the extraction trapdoor τ_{ext} as well as the public state St. It outputs an identity $\mathsf{id} \in \mathcal{ID}$ and a witness $w \in \mathcal{W}$.*

In addition, these algorithms must satisfy the following notions.

Extractable correctness: *for any* $(St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}, \tau_{\mathsf{ext}}) \leftarrow \mathsf{SimSetup}(1^\lambda, 1^N)$*, for any outcome* $\langle \mathsf{id}, \mathsf{cert}_{\mathsf{id}}, \mathsf{sec}_{\mathsf{id}} \rangle$ *such that* $\mathsf{cert}_{\mathsf{id}} \rightleftharpoons_{\mathcal{Y}} \mathsf{sec}_{\mathsf{id}}$*, any message* M*, any witness* w*, and any predicate* $P \in \mathcal{P}$*: if* $\sigma \leftarrow \mathsf{Sign}(\mathsf{id}, \mathsf{cert}_{\mathsf{id}}, \mathsf{sec}_{\mathsf{id}}, M, w, P)$ *and* $(\mathsf{id}', w') \leftarrow \mathsf{Extract}(\mathcal{Y}, \tau_{\mathsf{ext}}, M, \sigma, P, St)$*, then* $(\mathsf{id}, w) = (\mathsf{id}', w')$ *with overwhelming probability.*

Extractable soundness: *For any* PPT *adversary* $\mathcal{A}$ *involved in the experiments defined in Fig.2, the following advantage function must be negligible:*

$$\mathbf{Adv}^{\text{ext-s}}_{\mathcal{A}}(\lambda) = \Big| \Pr\big[\mathbf{Exp}^{\text{real}}_{\mathcal{A}}(\lambda) = 1\big] - \Pr\big[\mathbf{Exp}^{\text{ext}}_{\mathcal{A}}(\lambda) = 1\big] \Big|.$$

1 $\boxed{(St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)}$ $\boxed{(St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}, \tau_{\mathsf{ext}}) \leftarrow \mathsf{SimSetup}(1^\lambda, 1^N)}$;

2 $(M, \sigma, P, st) \leftarrow \mathcal{A}(St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}})$;

3 **if** $\mathsf{Verify}(\mathcal{Y}, M, \sigma, P) = 0$ **then return** 0;

4 $\boxed{\mathsf{id}^\star \leftarrow \mathsf{Open}(\mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, M, \sigma, P, St)}$ $\boxed{(\mathsf{id}, w) \leftarrow \mathsf{Extract}(\tau_{\mathsf{ext}}, \mathcal{Y}, M, \sigma, P, St)}$;

5 **if** $\boxed{P(\mathsf{id}, M, w) = 1}$ **then** $\boxed{\mathsf{id}^\star \leftarrow \perp}$ **else** $\boxed{\mathsf{id}^\star \leftarrow \mathsf{id}}$;

6 **return** $\mathcal{A}(st, \mathsf{id}^\star)$;

Fig. 2: Experiment $\mathbf{Exp}^{\text{real}}_{\mathcal{A}}(\lambda)$ (resp. $\mathbf{Exp}^{\text{ext}}_{\mathcal{A}}(\lambda)$) excluding the dotted (resp. solid) boxes.

In the random oracle model, Item 2 of Fig. 2 is modified as follows:

2' $\boxed{(M, \sigma, P, st) \leftarrow \mathcal{A}^{H_0}(St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}})}$ $\boxed{(M, \sigma, P, st) \leftarrow \mathcal{A}^{H_1}(St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}})}$;

Here, H_0 and H_1 are random oracles which privately evaluate and return the digests of given inputs. In the real setup, the BiAS algorithms have access to H_0 whereas, in the extractable setup, they have access to H_1.

We stress that all secret keys but the extraction trapdoor are given to the distinguisher/adversary. This is necessary because we need the extractable properties even in presence of dishonest authorities. In the extractable setting, we require Extract to output a potential identifier $\mathsf{id} \in \mathcal{ID}$ and a witness $w \in \mathcal{W}$ with overwhelming probability, even on adversarially-chosen verifying signatures and when both authorities are corrupted. This extractable mode makes it possible to compute the predicate in a meaningful way. Further, the extractable soundness property implies the hardness of computing a valid signature that traces to some user id for some predicate P although this predicate would have allowed user id to sign the message with statistical anonymity. While Extract is consistent with Open, we still do not have the complementary property of the hardness of computing a valid signature that cannot be traced although the tracing operation should have been possible. Indeed, if Open identifies a signature as non-traceable, we still have no clue about the meaning of the identity-witness pair produced by Extract on adversarially generated valid signatures that are not honestly generated (as otherwise, extractable-correctness implies the match with the actual pair)

Security Against Misidentification Attacks (a.k.a. traceability). In a misidentification attack, the adversary can corrupt the opening authority using the Q_{keyOA} oracle and introduce malicious users in U^a via $Q_{\mathsf{a\text{-}join}}$-queries. It aims at producing a valid signature $\sigma^\star$ that does not open to any adversarially-controlled user.

Definition 9. *A BiAS scheme is secure against* misidentification attacks *if it is branch-sound and, for any* PPT *adversary* $\mathcal{A}$ *involved in experiment* $\mathbf{Exp}^{\text{mis-id}}_{\mathcal{A},N}$ *(as defined in Fig.3), we have:* $\mathbf{Adv}^{\text{mis-id}}_{\mathcal{A},N}(\lambda) = \Pr\left[\mathbf{Exp}^{\text{mis-id}}_{\mathcal{A},N}(\lambda) = 1\right] \in \mathsf{negl}(\lambda)$.

1 $\mathsf{state}_{\mathcal{I}} := (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}})$; $(St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}, \tau_{\mathsf{ext}}) \leftarrow \mathsf{SimSetup}(1^\lambda, 1^N)$;
2 $(M, \sigma, P) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{a\text{-}join}}, Q_{\mathsf{read}}, Q_{\mathsf{write}}, Q_{\mathsf{keyOA}})$;
3 **if** $\mathsf{Verify}(\mathcal{Y}, M, \sigma, P) = 0$ **then return** 0;
4 $(\mathsf{id}, w) \leftarrow \mathsf{Extract}(\tau_{\mathsf{ext}}, \mathcal{Y}, M, \sigma, P, St)$;
5 **if** $\mathsf{id} \in \mathcal{ID} \setminus U^a$ **then return** 1;
6 **return** 0;

Fig. 3: Experiment $\mathbf{Exp}^{\text{mis-id}}_{\mathcal{A},N}(\lambda)$.

The winning condition is also checkable without the extractor if we rather define $\mathsf{id}^\star \leftarrow \mathsf{Open}(\mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, M, \sigma, P, St)$ in the experiment, as long as $\mathsf{id}^\star \neq \perp$ in the winning condition of line 5. In that case, the analogue security with the real setup is implied by the extractable soundness property. Nevertheless, in the extractable mode the definition also captures the unforgeability of anonymous signatures, i.e. those which would have made Open to return $\mathsf{id}^\star = \perp$ at line 5, if the extracted id does not correspond to a corrupt user when the group manager remains honest.

Non-Frameability. Framing attacks consider the case where the entire system is colluding against some honest user. The adversary can corrupt the group manager as well as the opening authority (via oracles Q_{keyGM} and Q_{keyOA}, respectively). It can also introduce honest group members (via $Q_{\mathsf{b\text{-}join}}$-queries), observe the system while these users sign messages and create dummy users using Q_{write}. The adversary aims at framing an honest group member. Moreover, the adversary is also deemed successful if it is able to create a non-traceable valid signature which could have been created by an honest user but who never computed it: even a corrupted group manager is unable to compute a non-traceable signature using the identity of an honest user. For example, if the predicate of a BiAS only allows some users to compute perfectly anonymous signatures, it is infeasible to compute such signatures without corrupting at least one of these users. The definition follows the indistinguishable approach of security against misidentification attacks.

Definition 10. *A BiAS scheme is secure against* framing attacks *if it satisfies branch-soundness and, for any* PPT *adversary* $\mathcal{A}$ *involved in experiment* $\mathbf{Exp}^{\text{fra}}_{\mathcal{A},N}$ *(as defined in Fig.4), we have:* $\mathbf{Adv}^{\text{fra}}_{\mathcal{A},N}(\lambda) = \Pr\left[\mathbf{Exp}^{\text{fra}}_{\mathcal{A},N}(\lambda) = 1\right] \in \mathsf{negl}(\lambda)$.

$\mathsf{state}_{\mathcal{I}} := (St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}})$; $(St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}, \tau_{\mathsf{ext}}) \leftarrow \mathsf{SimSetup}(1^{\lambda}, 1^{N})$;
$(M^{\star}, \sigma^{\star}, P^{\star}) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{keyOA}}, Q_{\mathsf{b\text{-}join}}, Q_{\mathsf{sig}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$;
if $\mathsf{Verify}(\mathcal{Y}, M^{\star}, \sigma^{\star}, P^{\star}) = 0$ **then return** 0;
$(\mathsf{id}, w) \leftarrow \mathsf{Extract}(\tau_{\mathsf{ext}}, \mathcal{Y}, M^{\star}, \sigma^{\star}, P^{\star}, St)$;
if $\big(\mathsf{id} \in U^{b}\big) \wedge (\mathsf{id}, M^{\star}, w, \sigma^{\star}, P^{\star}) \notin \mathsf{Sigs}$ **then return** 1;
return 0;

Fig. 4: Experiment $\mathbf{Exp}^{\mathrm{fra}}_{\mathcal{A},N}(\lambda)$

Let $\mathsf{id}^{\star} = \mathsf{Open}(\mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, M^{\star}, \sigma^{\star}, P^{\star}, St)$ in the framing experiment. Then, we can derive two winning conditions depending on whether $\mathsf{id}^{\star} \in \mathcal{ID}$ or $\mathsf{id}^{\star} = \perp$. In the former case, the branch-soundness tells us that $\mathsf{id}^{\star} = \mathsf{id}$. This traceable case is thus the analogue of the usual framing attack of KY in group signature transposed to our BiAS primitive. In the latter case, the signature $\sigma^{\star}$ of a successful adversary is deemed non-traceable, but it would have been created on behalf of an honest signer with identifier id who never produced it. This further justifies the need of all these security notions as we now have the complementary property discussed after Definition 8: a branch-sound BiAS scheme whose extracting algorithm returns independent identity-witness pairs given non-honest valid signatures cannot be secure against framing attacks.

Finally, we note that a signature does not only authenticate the message M, but it also binds the predicate value as well as the hidden (id, w) to M. The framing resistance also guarantees that the signature itself is not malleable as the winning condition is akin to the "strong"-unforgeability notion of standard signatures. This requirement is actually necessary since, in order to achieve anonymity in the "CCA sense", we need to prevent signatures from being malleable.

Discussion on the witness. Our model does not assume any property of the witness. At first glance, it may seem strange to apparently let the users choose their witnesses arbitrarily at the signature generation time. This syntactic choice makes BiAS more flexible to be combined with other building blocks. For instance, the witness w may already be committed in an external commitment, i.e. outside our syntax, and bound by the application. Additional zero-knowledge proofs between w, the context, and the BiAS scheme are of course possible, which might prevent the user from choosing w freely.

In a money-laundering prevention application, a signer has no incentive in authenticating a transaction for a big amount of money w if he does not want to pay such an amount. Therefore, even if $P(M, \mathsf{id}, w)$ may vary when w varies at each transaction, the context prevents the user with identity id from fixing w in an arbitrary way. We thus leave it to the applications to define their own rules on the w's and the desire and the way to keep their level of secrecy.

4 Generic Construction

We provide a generic construction of BiAS for an arbitrary predicate family $\mathcal{P} : \{0,1\}^* \to \{0,1\}$. Our construction relies on the following building blocks:

- An $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme $\Pi^{\mathsf{RLE}} = (\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$;
- An ordinary lossy PKE scheme $\Pi^{\mathrm{lpke}} = (\mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ where the message space has size at least N and forms an additive group;
- A digital signature scheme $\Pi^{\mathrm{sig}} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Verify})$ with signature space $\mathcal{S}$ and public key space $\mathcal{VK}$;
- A one-time signature $\Pi^{\mathrm{ots}} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Verify})$;
- A homomorphic equivocal commitment scheme $\mathsf{HEC} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval}^{in}, \mathsf{Eval}^{out}, \mathsf{Verify})$, where Commit samples its random coins from a distribution $\mathcal{D}^{\mathsf{HEC}}$ over a randomness space R^{HEC};
- A dual-mode statistical NIZK argument system $\mathsf{NIZK} = (\mathsf{Setup}, \mathsf{ExtSetup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{Sim}, \mathsf{Extract})$, as defined in the full version.

Since an $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme implies a standard lossy PKE scheme, the only ingredients we need are an $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE system, a digital signature, a homomorphic equivocable commitment and a dual-mode NIZK argument.

For our purposes, it is sufficient to use an HEC scheme without the context-hiding property since we combine it with NIZK arguments where its partial openings serve as witnesses.[7] By using an HEC with the efficient verification property, we can make the signature length independent of the circuit size. Katsumata *et al.* [27] gave such a pairing-based HEC construction under a q-type assumption for NC^1 circuits. In the lattice setting, the fully homomorphic commitments of Gorbunov *et al.* [23] provide efficient verification (for bounded-depth circuits) under the Short Integer Solution [1] assumption, as recalled in the full version. At the expense of a signature length depending on the circuit size, the construction can be simplified to use any statistically hiding commitment instead of an efficiently verifiable HEC. However, we aim at avoiding the circuit-size dependency.

Intuitively, the construction encrypts the group member's identity id and the witness w using an HEC *and* simultaneously encrypts them into $\mathsf{ct}_{(\mathsf{id},w)}$ using the $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme, which realizes either a statistically hiding or an extractable commitment to (id, w). Our proofs of anonymity require statistically-hiding commitments (as in the real scheme). In our proofs of security against mis-identification attacks and framing attacks, we will switch $\mathsf{ct}_{(\mathsf{id},w)}$ to its extractable mode because we need to be able to extract the underlying w and id.

In the signing algorithm, the group member next computes an evaluated HEC commitment com_{ev} of the predicate evaluation $C_M(\mathsf{id}, w)$ by homomorphically computing over $\mathsf{com}_{(\mathsf{id},w)}$ (note that com_{ev} need not be included in the signature since the verifier can recompute if from $\mathsf{com}_{(\mathsf{id},w)}$). Then, the signer computes a ciphertext $\mathsf{ct}_{\mathsf{id}}$ that verifiably encrypts a product $(1 - C_M(\mathsf{id}, w)) \cdot \mathsf{id}$ of his

[7] A context-hiding construction can still improve the efficiency by outputting partial openings in the clear in each signature.

identity id and the logical NOT of $C_M(\mathsf{id}, w)$. When the predicate evaluates to $C_M(\mathsf{id}, w) = 0$, $\mathsf{ct}_{\mathsf{id}}$ is distributed as a lossy encryption[8] of id. When $C_M(\mathsf{id}, w) = 1$, $\mathsf{ct}_{\mathsf{id}}$ is completely independent of the signer's identity as it encrypts $\mathbf{0}^{|\mathsf{id}|}$.

Setup$(1^\lambda, 1^N, \mathcal{P}_d)$**:** Given a security parameter λ, a predicate family $\mathcal{P}_d$ modeled by circuits of depth $d = d(\lambda)$ and the maximal number of group members $N = 2^\ell \in \mathsf{poly}(\lambda)$, do the following.

1. Generate a key pair $(\mathsf{pk}_{sig}, \mathsf{sk}_{sig}) \leftarrow \Pi^{\mathrm{sig}}.\mathsf{Kg}(1^\lambda)$ for the signature scheme. We assume that each public key has bitlength $\ell_{sig} \in \mathsf{poly}(\lambda)$.
2. Run $(\mathsf{pp}, \mathsf{ek}, \mathsf{msk}) \leftarrow \mathsf{HEC.Setup}(1^\lambda)$ to generate parameters for the homomorphic equivocal commitment, together with an evaluation key ek and a master key msk.
3. Choose a one-time signature scheme $\Pi^{\mathrm{ots}} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Verify})$ with verification key space $\{0,1\}^L$, for some $L \in \mathsf{poly}(\lambda)$.
4. Choose public parameters $\Gamma \leftarrow \Pi^{\mathsf{RLE}}.\mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$ for an $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme with tag space $\mathcal{K} = \mathcal{T} = \{0,1\}^L$ and message length $B = \ell + \ell_w$, where $\ell_w \in \mathsf{poly}(\lambda)$ is the bitlength of witnesses from the witness space $\mathcal{W} = \{0,1\}^{\ell_w}$. Then, generate lossy keys $(\mathsf{pk}_{\mathsf{RLE}}, \mathsf{sk}_{\mathsf{RLE}}) \leftarrow \Pi^{\mathsf{RLE}}.\mathsf{LKeygen}(\Gamma, \mathbf{0}^L)$ for the initialization value $K = \mathbf{0}^L$.
5. Generate an injective key pair $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \Pi^{\mathrm{lpke}}.\mathsf{Keygen}(1^\lambda)$ for the standard lossy PKE scheme.
6. Generate a common reference string ρ from $(\rho, \zeta) \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$ for a dual-mode NIZK argument in its statistical ZK mode.

The algorithm outputs $\big(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}\big)$, where the group public key is as

$$\mathcal{Y} := \big(\rho, \mathsf{pk}_{sig}, (\mathsf{pp}, \mathsf{ek}), (\Gamma, \mathsf{pk}_{\mathsf{RLE}}), \mathsf{pk}_e\big),$$

the opening authority's private key is $\mathcal{S}_{\mathsf{OA}} := \mathsf{sk}_e$ and the private key of the group manager consists of $\mathcal{S}_{\mathsf{GM}} := \mathsf{sk}_{sig}$. $\mathcal{Y}$ implicitly initializes St.

Join$^{(\mathrm{GM},\mathcal{U}_{\mathsf{id}})}$**:** the group manager and the prospective user $\mathcal{U}_{\mathsf{id}}$ run the following interactive protocol $[\mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})]$:

1. User $\mathcal{U}_{\mathsf{id}}$ generates a key pair $(\mathsf{sk}_{\mathsf{id}}, \mathsf{pk}_{\mathsf{id}}) \leftarrow \Pi^{\mathrm{sig}}.\mathsf{Kg}(1^\lambda)$ and sends the public key $\mathsf{pk}_{\mathsf{id}}$ together with his identity $\mathsf{id} \in \{0,1\}^\ell \setminus \{\mathbf{0}^\ell\}$ and an ordinary signature $sig_{\mathsf{id}} \leftarrow \Pi^{\mathrm{sig}}.\mathsf{Sign}(\mathsf{usk}[\mathsf{id}], (\mathsf{id}, \mathsf{pk}_{\mathsf{id}}))$ to GM.
2. J_{GM} verifies that: $\mathsf{id} \neq \mathbf{0}^\ell$; id was not previously used by a registered user; sig_{id} is a valid signature on $(\mathsf{id}, \mathsf{pk}_{\mathsf{id}})$ w.r.t. $\mathsf{upk}[\mathsf{id}]$. It aborts if this is not the case. Otherwise, it computes $\mathsf{cert}_{\mathsf{id}} \leftarrow \Pi^{\mathrm{sig}}.\mathsf{Sign}(\mathsf{sk}_{sig}, (\mathsf{id}, \mathsf{pk}_{\mathsf{id}}))$ as a signature on the message $(\mathsf{id}, \mathsf{pk}_{\mathsf{id}})$. The membership certificate $\mathsf{cert}_{\mathsf{id}}$ is sent to $\mathcal{U}_{\mathsf{id}}$. Then, $\mathsf{J}_{\mathsf{user}}$ verifies that $\Pi^{\mathrm{sig}}.\mathsf{Verify}(\mathsf{pk}_{sig}, (\mathsf{id}, \mathsf{pk}_{\mathsf{id}}), \mathsf{cert}_{\mathsf{id}}) = 1$. If this condition is not satisfied, $\mathsf{J}_{\mathsf{user}}$ aborts. Otherwise, $\mathsf{J}_{\mathsf{user}}$ defines the membership certificate as $\mathsf{cert}_{\mathsf{id}}$. The membership secret $\mathsf{sec}_{\mathsf{id}}$ is defined to be $\mathsf{sec}_{\mathsf{id}} = \mathsf{sk}_{\mathsf{id}}$. J_{GM} stores $\mathsf{transcript}_{\mathsf{id}} = (\mathsf{id}, \mathsf{pk}_{\mathsf{id}}, \mathsf{cert}_{\mathsf{id}}, \mathsf{upk}[\mathsf{id}], sig_{\mathsf{id}})$ in the database St_{trans} of joining transcripts.

[8] It is possible to compute $\mathsf{ct}_{\mathsf{id}}$ using an ordinary (i.e., non-lossy) PKE scheme but it requires to rely on the simulation-soundness of NIZK in the proof of Lemma 4.

Sign$(\mathsf{id}, \mathsf{cert}_{\mathsf{id}}, \mathsf{sec}_{\mathsf{id}}, M, w, P)$**:** To sign a message $M \in \{0,1\}^{\ell_m}$ using the witness $w = w[1] \dots w[\ell_w] \in \{0,1\}^{\ell_w}$ w.r.t. the predicate $P \in \mathcal{P}_d$, let $C_M : \{0,1\}^{\ell_w} \times \{0,1\}^{\ell} \to \{0,1\}$ be the message-dependent Boolean circuit of depth $\leq d$ that evaluates $P(M, \mathsf{id}, w)$ on input of $(w[1], \dots, w[\ell_w], \mathsf{id}[1], \dots, \mathsf{id}[\ell])$.

1. Generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \Pi^{\mathrm{ots}}.\mathsf{Kg}(1^{\lambda})$.
2. Choose $r_{\mathsf{id},w} \hookleftarrow R^{\mathsf{RLE}}$ in the randomness space of Π^{RLE} and encrypt the identity-witness pair $(\mathsf{id}, w) \in \{0,1\}^{\ell+\ell_w}$ as an $\mathcal{R}_{\mathsf{NEQ}}$-lossy encryption
$$\mathsf{ct}_{(\mathsf{id},w)} = \Pi^{\mathsf{RLE}}.\mathsf{Encrypt}(\mathsf{pk}_{\mathsf{RLE}}, \mathsf{VK}, (\mathsf{id}, w); r_w) \tag{1}$$
under the tag $\mathsf{VK} \in \{0,1\}^{L}$.
3. Sample random coins $r^{\mathrm{hec}} \hookleftarrow \mathcal{D}^{\mathsf{HEC}}$ and compute a commitment
$$\mathsf{com}_{(\mathsf{id},w)} = \mathsf{HEC}.\mathsf{Commit}(\mathsf{pp}, \mathsf{ek}, (\mathsf{id}, w); r^{\mathrm{hec}}). \tag{2}$$
4. Using the homomorphic evaluation algorithm of HEC, compute
$$\begin{aligned} \pi_{C,M} &\leftarrow \mathsf{HEC}.\mathsf{Eval}^{in}\big(\mathsf{ek}, C_M, (\mathsf{id}, w), r^{\mathrm{hec}}\big) \\ \mathsf{com}_{ev} &= \mathsf{HEC}.\mathsf{Eval}^{out}\big(\mathsf{ek}, C_M, \mathsf{com}_{(\mathsf{id},w)}\big). \end{aligned}$$
5. Choose $r^{\mathrm{lpke}} \hookleftarrow R^{\mathrm{lpke}}$ and compute
$$\mathsf{ct}_{\mathsf{id}} = \Pi^{\mathrm{lpke}}.\mathsf{Encrypt}\big(\mathsf{pk}_e, (1 - c_{ev}) \cdot \mathsf{id}; r^{\mathrm{lpke}}\big), \tag{3}$$
where $c_{ev} = C_M(w_1, \dots, w_{\ell_w}, \mathsf{id}_1, \dots, \mathsf{id}_{\ell}) \in \{0,1\}$.
6. Generate $\sigma \leftarrow \Pi^{\mathrm{sig}}.\mathsf{Sign}(\mathsf{sk}_{\mathsf{id}}, (M, P, \mathsf{ct}_{(\mathsf{id},w)}))$ as a signature on the message $(M, P, \mathsf{ct}_{(\mathsf{id},w)})$.
7. Generate a NIZK argument $\vec{\pi} \leftarrow \mathsf{NIZK}.\mathsf{Prove}(\rho, \vec{x}, \vec{w})$ for the statement $\vec{x}$ that there exists a witnesses $\vec{w}$ comprised of $(\mathsf{id}, w) \in \{0,1\}^{\ell+\ell_w}$, $(\mathsf{pk}_{\mathsf{id}}, \mathsf{cert}_{\mathsf{id}}, \sigma) \in \mathcal{VK} \times \mathcal{S} \times \mathcal{S}$, $r_{\mathsf{id},w} \in R^{\mathsf{RLE}}$, $r^{\mathrm{hec}} \in R^{\mathsf{HEC}}$, $r^{\mathrm{lpke}} \in R^{\mathrm{lpke}}$, $c_{ev} \in \{0,1\}$ and $\pi_{C,M}$, which satisfy the relations (1)-(3) as well as
$$\begin{aligned} \Pi^{\mathrm{sig}}.\mathsf{Verify}(\mathsf{pk}_{sig}, (\mathsf{id}, \mathsf{pk}_{\mathsf{id}}), \mathsf{cert}_{\mathsf{id}}) &= 1 \\ \Pi^{\mathrm{sig}}.\mathsf{Verify}(\mathsf{pk}_{\mathsf{id}}, (M, P, \mathsf{ct}_{(\mathsf{id},w)}), \sigma) &= 1 \\ \mathsf{HEC}.\mathsf{Verify}(\mathsf{pp}, \mathsf{com}_{ev}, c_{ev}, \pi_{C,M}) &= 1. \end{aligned} \tag{4}$$
8. Compute $sig \leftarrow \Pi^{\mathrm{ots}}.\mathsf{Sign}(\mathsf{SK}, (\mathsf{ct}_{(\mathsf{id},w)}, \mathsf{com}_{(\mathsf{id},w)}, \mathsf{ct}_{\mathsf{id}}, \vec{\pi}))$.

Return the signature
$$\Sigma = \big(\mathsf{VK}, (\mathsf{ct}_{(\mathsf{id},w)}, \mathsf{com}_{(\mathsf{id},w)}, \mathsf{ct}_{\mathsf{id}}, \vec{\pi}), sig\big) \tag{5}$$

Verify$(\mathcal{Y}, M, \Sigma, P)$**:** Parse Σ as above. Return 1 if and only if: (i) sig is a valid one-time signature on $(\mathsf{ct}_{(\mathsf{id},w)}, \mathsf{com}_{(\mathsf{id},w)}, C_{\mathsf{id}}, \vec{\pi})$ for the verification key VK; (ii) The NIZK argument $\vec{\pi}$ properly verifies for the commitment com_{ev} publicly obtained as $\mathsf{com}_{ev} = \mathsf{HEC}.\mathsf{Eval}^{out}\big(\mathsf{ek}, C_M, \mathsf{com}_{(\mathsf{id},w)}\big)$.

Open$(\mathcal{Y}, \mathcal{S}_{\mathsf{OA}}, M, \Sigma, P, St)$**:** Given the opener's secret key $\mathcal{S}_{\mathsf{OA}} := \mathsf{sk}_e$, parse the signature Σ as in (5). Compute $t_{\mathsf{id}} = \Pi^{\mathrm{lpke}}.\mathsf{Decrypt}(\mathsf{sk}_e, \mathsf{ct}_{\mathsf{id}})$. If $t_{\mathsf{id}} = \mathbf{0}^\ell$, return $\perp$. Otherwise, check if the string $t_{\mathsf{id}} \in \{0,1\}^\ell$ appears in a record $(t_{\mathsf{id}}, \mathsf{transcript}_{\mathsf{id}} = (t_{\mathsf{id}}, \mathsf{pk}_{\mathsf{id}}, \mathsf{cert}_{\mathsf{id}}, \mathsf{upk}[\mathsf{id}], sig_{\mathsf{id}}))$ of St_{trans}. If it does, output $\mathsf{id} = t_{\mathsf{id}} \in \{0,1\}^\ell$ (and, optionally, $\mathsf{upk}[\mathsf{id}]$). Otherwise, output $\perp$.

In the full version, we provide details on instantiations from lattices and bilinear maps. The lattice-based construction is only a feasibility result based on generic NIZK for NP statements [42]. In the case of NC^1 circuits, the scheme can be instantiated with Groth-Sahai proofs [25] to provide much shorter signatures than using the Groth-Ostrovsky-Sahai techniques [24].

4.1 Branch-Soundness and Security

To prove security under our definitions, we first consider the following SimSetup and Extract algorithms associated to our BiAS construction.

SimSetup$(1^\lambda, 1^N, \mathcal{P}_d)$**:** This algorithm is exactly as $\mathsf{Setup}(1^\lambda, 1^N, \mathcal{P}_d)$ except that steps 4 and 6 are modified in the following way:

4. Choose public parameters $\Gamma \leftarrow \Pi^{\mathsf{RLE}}.\mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$ for an $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme with tag space $\mathcal{K} = \mathcal{T} = \{0,1\}^L$ and message length $B = \ell + \ell_w$, where $\ell_w \in \mathsf{poly}(\lambda)$ is the bitlength of witnesses from the witness space $\mathcal{W} = \{0,1\}^{\ell_w}$. Then, generate *injective* keys $(\mathsf{pk}_{\mathsf{RLE}}, \mathsf{sk}_{\mathsf{RLE}}) \leftarrow \Pi^{\mathsf{RLE}}.\mathsf{Keygen}(\Gamma, \mathbf{0}^L)$ for the initialization value $K = \mathbf{0}^L$.
6. Generate a common reference string ρ from $(\rho, \xi) \leftarrow \mathsf{NIZK.ExtSetup}(1^\lambda)$ for an extractable (and thus statistically sound) NIZK proof system.

The algorithm returns the same output as Setup, together with an extraction trapdoor $\tau_{\mathsf{ext}} = (\mathsf{sk}_{\mathsf{RLE}}, \xi)$, where ξ is the extraction trapdoor of NIZK.

Extract$(\mathcal{Y}, \tau_{\mathsf{ext}}, M, \Sigma, P, St)$**:** Write Σ as $\big(\mathsf{VK}, (\mathsf{ct}_{(\mathsf{id},w)}, \mathsf{com}_{(\mathsf{id},w)}, \mathsf{ct}_{\mathsf{id}}, \vec{\pi}), sig\big)$ and return $\perp$ if its components do not parse properly. Otherwise, use $\mathsf{sk}_{\mathsf{RLE}}$ to decrypt the $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE ciphertexts $\mathsf{ct}_{(\mathsf{id},w)}$ (recall that the NEQ relations makes all tags injective on a public key produced by Keygen for the initialization value $K = \mathbf{0}^\ell$). If any decryption fails, return $\perp$. Otherwise, output $w \in \{0,1\}^{\ell_w}$ and $\mathsf{id} \in \{0,1\}^\ell$.

The security properties of the NIZK argument system ensure that the common reference strings ρ produced by NIZK.Setup and NIZK.ExtSetup are computationally indistinguishable. Moreover, in the $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme, the public keys produced by LKeygen and Keygen are computationally indistinguishable as well.

Next, we will show that this extractable BiAS satisfies the extractable soundness notion unless the adversary can break the (statistical) soundness of the proof $\vec{\pi}$ included in a valid signature Σ.

Theorem 1. *The scheme satisfies branch-soundness if: (i)* Π^{RLE} *is a secure* $\mathcal{R}_{\mathsf{NEQ}}$*-lossy PKE scheme; (ii)* NIZK *is a dual-mode NIZK argument system (i.e., its statistically sound and statistically ZK modes are computationally indistinguishable); (iii)* HEC *is computationally binding for evaluated commitments.*

Proof. To prove the result, we consider a sequence of games. In each game, we call W_i the event that the challenger outputs 1.

Game 0**:** This is the real experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{real}}(\lambda)$, where the adversary $\mathcal{A}$ is given $(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}})$, where $(St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$. The adversary outputs a tuple (M, Σ, P, st), where $\Sigma = \big(\mathsf{VK}, (\mathsf{ct}_{(\mathsf{id},w)}, \mathsf{com}_{(\mathsf{id},w)}, \mathsf{ct}_{\mathsf{id}}, \vec{\pi}), sig\big)$. If Σ does not verify, the challenger outputs 0. Otherwise, it runs Open to obtain $\mathsf{id}^\star \in \{0,1\}^\ell$ and feeds $\mathcal{A}$ with $\mathsf{id}^\star$. Then, the challenger outputs whatever $\mathcal{A}$ outputs. By definition, $\Pr[W_0] = \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{real}}(\lambda) = 1]$.

Game 1**:** This game is identical to Game 0 except that, at step 4 of the Setup algorithm, the challenger computes $(\mathsf{pk}_{\mathsf{RLE}}, \mathsf{sk}_{\mathsf{RLE}}) \leftarrow \Pi^{\mathsf{RLE}}.\mathsf{Keygen}(\Gamma, \mathbf{0}^L)$ instead of $(\mathsf{pk}_{\mathsf{RLE}}, \mathsf{sk}_{\mathsf{RLE}}) \leftarrow \Pi^{\mathsf{RLE}}.\mathsf{LKeygen}(\Gamma, \mathbf{0}^L)$. By the first indistinguishability property of Π^{RLE}, we have $|\Pr[W_1] - \Pr[W_0]| \in \mathsf{negl}(\lambda)$.

Game 2**:** This game is like Game 1 except that, at step 6 of the Setup algorithm, the challenger generates $(\rho, \xi) \leftarrow \mathsf{NIZK.ExtSetup}(1^\lambda)$ instead of $(\rho, \zeta) \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$ and keeps the extraction trapdoor $\tau_{\mathsf{ext}} = (\mathsf{sk}_{\mathsf{RLE}}, \xi)$ to itself. By the dual-mode property of NIZK, the CRSes produced by $\mathsf{NIZK.Setup}$ and $\mathsf{NIZK.ExtSetup}$ have computationally indistinguishable distributions, thus ensuring that $|\Pr[W_2] - \Pr[W_1]| \in \mathsf{negl}(\lambda)$ for any PPT adversary $\mathcal{A}$.

Game 3**:** In this game, the challenger makes use of the trapdoor $\tau_{\mathsf{ext}} = (\mathsf{sk}_{\mathsf{RLE}}, \xi)$. When $\mathcal{A}$ outputs a tuple (M, Σ, P, st), the challenger parses the signature Σ as $\big(\mathsf{VK}, (\mathsf{ct}_{(\mathsf{id},w)}, \mathsf{com}_{(\mathsf{id},w)}, \mathsf{ct}_{\mathsf{id}}, \vec{\pi}), sig\big)$ and uses $\mathsf{sk}_{\mathsf{RLE}}$ to extract $(\mathsf{id}^\dagger, w^\dagger)$. From the NIZK proof $\vec{\pi}$, it uses ξ to extract the witnesses $(\mathsf{id}, w) \in \{0,1\}^{\ell_w + \ell}$, $(\mathsf{pk}_{\mathsf{id}}, \mathsf{cert}_{\mathsf{id}}, \sigma) \in \mathcal{VK} \times \mathcal{S} \times \mathcal{S}$, $r_{\mathsf{id},w} \in R^{\mathsf{RLE}}$, $r^{\mathrm{hec}} \in R^{\mathsf{HEC}}$, $r^{\mathrm{lpke}} \in R^{\mathrm{lpke}}$, $c_{ev} \in \{0,1\}$ and $\pi_{C,M}$. Then, the challenger halts and outputs a random bit if $c_{ev} \neq C_M(w_1, \ldots, w_{\ell_w}, \mathsf{id}[1], \ldots, \mathsf{id}[\ell])$.

We claim that $|\Pr[W_3] - \Pr[W_2]| \in \mathsf{negl}(\lambda)$ as the two games only differ when $\mathcal{A}$ breaks the computational binding property of HEC for evaluated commitments. Indeed, by the statistical soundness of NIZK on a CRS ρ produced by $\mathsf{NIZK.ExtSetup}$, we have $(\mathsf{id}, w) = (\mathsf{id}^\dagger, w^\dagger)$ and extracted witnesses satisfy the relations (1)-(3). In particular, we have $\mathsf{com}_{(\mathsf{id},w)} = \mathsf{HEC.Commit}(\mathsf{pp}, \mathsf{ek}, (\mathsf{id}, w); r^{\mathrm{hec}})$ and the extracted $c_{ev} \in \{0,1\}$, $\pi_{C,M}$ satisfy $\mathsf{HEC.Verify}(\mathsf{pp}, \mathsf{com}_{ev}, c_{ev}, \pi_{C,M}) = 1$, where $\mathsf{com}_{ev} = \mathsf{HEC.Eval}^{out}\big(\mathsf{ek}, C_M, \mathsf{com}_{(\mathsf{id},w)}\big)$. It is easy to see that Game_3 only differs from Game_2 when the extracted $\pi_{C,M}$ differs from

$$\bar{\pi}_{C,M} \leftarrow \mathsf{HEC.Eval}^{in}\big(\mathsf{ek}, C_M, (\mathsf{id}, w), r^{\mathrm{hec}}\big),$$

which is the value that would satisfy $\mathsf{HEC.Verify}(\mathsf{pp}, \mathsf{com}_{ev}, C_M(w, \mathsf{id}), \bar{\pi}_{C,M}) = 1$. Hence, if $|\Pr[W_3] - \Pr[W_2]|$ is noticeable, the challenger can break the binding property of HEC by outputting $\big((\mathsf{id}, w), r^{\mathrm{hec}}, C_M, c_{ev}, \pi_{C,M}\big)$.

Game 4**:** This game is identical to Game 3 with the difference that, after having extracted (id, w), the challenger computes $C_M(w, \mathsf{id}) \in \{0,1\}$, which is identical to the extracted $c_{ev} \in \{0,1\}$ unless the failure event of Game 3 occurs. If $C_M(w, \mathsf{id}) = 0$, it overwrites $\mathsf{id}^\star \leftarrow \mathsf{Open}(\mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, M, \sigma, P, St)$

with $\mathsf{id}^\star = \mathsf{id}$, which was extracted from $\mathsf{ct}_{(\mathsf{id},w)}$. If $C_M(w,\mathsf{id}) = 1$, it sets $\mathsf{id}^\star = \perp$. In both cases, it feeds $\mathcal{A}$ with $\mathsf{id}^\star$ and returns whatever $\mathcal{A}$ outputs in reaction. This change does not modify the output distribution of $\mathcal{A}$ because, as long as $c_{ev} = C_M(w,\mathsf{id})$, the statistical soundness of $\vec{\pi}$ ensures that $\mathsf{ct}_{\mathsf{id}} = \Pi^{\mathrm{lpke}}.\mathsf{Encrypt}\big(\mathsf{pk}_e, (1 - C_M(w,\mathsf{id})) \cdot \mathsf{id}; r^{\mathrm{lpke}}\big)$, where r^{lpke} and id are extracted from $\vec{\pi}$. Hence, unless $\mathcal{A}$ breaks the statistical soundness of $\vec{\pi}$, Game 4 eventually returns $\mathsf{id} = \perp$ or $\mathsf{id} = \mathsf{id}^\star$ to $\mathcal{A}$ whenever Game 3 does.

Game 5: This game is like Game 4 but we remove the restriction introduced in Game 3. Namely, the challenger does no longer replace $\mathcal{A}$'s output by a random bit when the witnesses $c_{ev} \in \{0,1\}$, $(\mathsf{id}, w) \in \{0,1\}^{\ell_w + \ell}$ extracted from $\vec{\pi}$ are such that $c_{ev} \neq C_M(w_1, \ldots, w_{\ell_w}, \mathsf{id}[1], \ldots, \mathsf{id}[\ell])$. The same arguments as those in the transition between the first two games show that $|\Pr[W_5] - \Pr[W_4]| \in \mathsf{negl}(\lambda)$ so long as HEC is computationally binding.

We conclude the proof by noting that Game 5 is identical to $\mathbf{Exp}^{\mathrm{ext}}_{\mathcal{A}}(\lambda)$, so that we have $|\Pr\big[\mathbf{Exp}^{\mathrm{real}}_{\mathcal{A}}(\lambda) = 1\big] - \Pr\big[\mathbf{Exp}^{\mathrm{ext}}_{\mathcal{A}}(\lambda) = 1\big]| = [\Pr[W_0] - \Pr[W_5]|$. □

Security Against Mis-Identification and Framing Attacks.

Lemma 2. *The scheme is secure against misidentification attacks if: (i) Π^{sig} is existentially unforgeable under chosen-message attacks; (ii) The NIZK argument is computationally sound.* (The proof is given in the full version.)

Lemma 3. *The scheme is secure against framing attacks provided: (i) Π^{sig} is strongly unforgeable under chosen-message attacks; (ii) The NIZK argument is computationally sound.* (The proof is given in the full version.)

4.2 Branch-Hiding and Privacy

The branch-hiding property follows from the full anonymity of our scheme.

Theorem 2. *The scheme provides full anonymity if: Π^{RLE} and Π^{lpke} are secure $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE and standard lossy PKE schemes, respectively; (ii) NIZK is a computationally sound NIZK argument; (iii) Π^{ots} is strongly unforgeable.*

To prove Theorem 2, we separately consider the tracing and non-tracing modes. Lemma 4 first considers the former case where the adversary does not corrupt the opening authority. Lemma 5 shows that even an unbounded adversary is unable to distinguish group members' signatures in non-tracing mode.

Lemma 4. *The scheme provides anonymity in tracing mode assuming that: (i) Π^{RLE} is a secure $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme; (ii) Π^{lpke} is a standard lossy PKE scheme; (ii) The NIZK argument system provides soundness; (iii) Π^{ots} is strongly unforgeable.* (The proof is given in the full version.)

Lemma 5. *The scheme provides statistical anonymity in non-tracing mode.*

Proof. Recall that experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{anon-ntr-}d}(\lambda)$ allows the adversary to obtain a challenge for the non-tracing mode. Namely, it is allowed to corrupt the opening authority and obtain $\mathcal{S}_{\mathsf{OA}}$ as long as, in the challenge phase, it chooses a pair $(M^\star, P^\star)$ and two tuples $(\mathsf{id}_0^\star, w_0^\star, \mathsf{sec}_0^\star, \mathsf{cert}_0^\star)$ and $(\mathsf{id}_1^\star, w_1^\star, \mathsf{sec}_1^\star, \mathsf{cert}_1^\star)$, such that $P^\star(M^\star, \mathsf{id}_0^\star, w_0^\star) = P^\star(M^\star, \mathsf{id}_1^\star, w_1^\star) = 1$. In this scenario, we will prove that, even after having obtained $\mathcal{S}_{\mathsf{OA}}$, an unbounded adversary $\mathcal{A}$ remains unable to infer anything about the bit $d \in \{0,1\}$ used by the challenger to compute the signature $\Sigma^\star = \big(\mathsf{VK}^\star, (\mathsf{ct}_{(\mathsf{id},w)}^\star, \mathsf{com}_{(\mathsf{id},w)}^\star, \mathsf{ct}_{\mathsf{id}}^\star, \vec{\pi}^\star), sig^\star\big)$ using $(\mathsf{id}_d^\star, w_d^\star, \mathsf{sec}_d^\star, \mathsf{cert}_d^\star)$.

To this end, we consider two statistically indistinguishable games. The first one is the real game whereas the second one appeals to the statistical honest-verifier zero-knowledge simulator of the argument system.

$\mathbf{Game}^{(d)}$ 0**:** This is the real game, which is as in the proof of Lemma 4.

$\mathbf{Game}^{(d)}$ 1**:** This game is like $\mathbf{Game}^{(d)}$ 0 except that, in the challenge signature $\Sigma^\star$, we use the simulation trapdoor ζ generated from $\mathsf{NIZK.Setup}$ and the statistical NIZK simulator $\mathsf{NIZK.Sim}$ to generate $\vec{\pi}^\star$. Owing to the statistical ZK property of NIZK, the simulated $\vec{\pi}^\star$ is statistically close to a real $\vec{\pi}^\star$ that would be generated using the witnesses. Moreover, it is statistically independent of the witnesses used to compute $\mathsf{ct}_{(\mathsf{id},w)}^\star$, $\mathsf{com}_{(\mathsf{id},w)}^\star$ and $\mathsf{ct}_{\mathsf{id}}^\star$.

In $\mathbf{Game}^{(d)}$ 1, we note that, when $C_{M^\star}(w_1^\star, \ldots, w_{\ell_w}^\star, \mathsf{id}_1^\star, \ldots, \mathsf{id}_\ell^\star) = 1$, the ciphertext $\mathsf{ct}_{\mathsf{id}}^\star$ is of the form $\mathsf{ct}_{\mathsf{id}}^\star = \Pi^{\text{lpke}}.\mathsf{Encrypt}\big(\mathsf{pk}_e, \mathbf{0}^\ell; r^{\text{lpke}^\star}\big)$, where $r^{\text{lpke}^\star} \hookleftarrow R^{\text{lpke}}$, so that $\mathsf{ct}_{\mathsf{id}}^\star$ is independent of $d \in \{0,1\}$ although pk_e is an injective public key. Moreover, $\mathsf{ct}_{(\mathsf{id},w)}^\star$, $\mathsf{com}_{(\mathsf{id},w)}^\star$ statistically hide the underlying pair (id, w) since, by definition, the homomorphic equivocable commitment $\mathsf{com}_{(\mathsf{id},w)}^\star$ is statistically hiding and the $\mathcal{R}_{\mathsf{NEQ}}$-lossy encryption $\mathsf{ct}_{(\mathsf{id},w)}^\star$ is computed under a lossy key produced by $\mathsf{LKeygen}$. □

Acknowledgements

Part of this research was funded by the French ANR ALAMBIC project (ANR-16-CE39-0006). This work was also supported in part by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701). Khoa Nguyen was supported in part by the Gopalakrishnan - NTU PPF 2018, by A*STAR, Singapore under research grant SERC A19E3b0099, and by Vietnam National University HoChiMinh City (VNU-HCM) under grant number NCM2019-18-01. Thomas Peters is a research associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS).

References

1. M. Ajtai. Generating hard instances of lattice problems. In *STOC*, 1996.
2. E. Bangerter, J. Camenisch, and A. Lysyanskaya. A cryptographic framework for the controlled release of certified data. In *Security Protocols*, 2004.
3. M. Bellare and G. Fuchsbauer. Policy-based signatures. In *PKC*, 2014.

4. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Eurocrypt*, 2003.
5. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA*, 2005.
6. A. Bender, J. Katz, and R. Morselli. Ring Signatures: Stronger Definitions, and Constructions without Random Oracles. *J. Cryptology*, 22(1):114–138, 2009.
7. M. Blum, M. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *STOC*, 1988.
8. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Crypto*, 2004.
9. J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, J. Groth, and C. Petit. Short accountable ring signatures based on DDH. In *ESORICS*, 2015.
10. J. Bootle, V. Lyubashevsky, and G. Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In *Crypto*, 2019.
11. X. Boyen. Mesh signatures. In *Eurocrypt*, 2007.
12. X. Boyen and C. Delerabée. Expressive subgroup signatures. In *SCN*, 2008.
13. E. Boyle, S. Goldwasser, and I. Ivan. Functional signatures and pseudorandom functions. In *PKC*, 2014.
14. E. Boyle, G. Segev, and D. Wichs. Fully leakage-resilient signatures. In *Eurocrypt*, 2011.
15. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *Eurocrypt*, 2005.
16. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Balancing accountability and privacy using e-cash. In *SCN*, 2006.
17. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. Rothblum, R. Rothblum, and D. Wichs. Fiat-Shamir: From practice to theory. In *STOC*, 2019.
18. D. Chaum and E. Van Heyst. Group signatures. In *Eurocrypt*, 1991.
19. Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous identification in ad-hoc groups. In *Eurocrypt*, 2004.
20. E. Fujisaki and K. Suzuki. Traceable ring signature. In *PKC*, 2007.
21. L. Garms and A. Lehmann. Group signatures with selective linkability. In *PKC*, 2019.
22. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Crypto*, 2013.
23. S. Gorbunov, V. Vaikuntanathan, and D. Wichs. Leveled fully homomorphic signatures from standard lattices. In *STOC*, 2015.
24. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero-knowledge for NP. In *Eurocrypt*, 2006.
25. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt*, 2008.
26. D. Hofheinz and B. Ursu. Dual-mode NIZKs from obfuscation. In *Asiacrypt*, 2019.
27. S. Katsumata, R. Nishimaki, S. Yamada, and T. Yamakawa. Exploring constructions of compact nizks from various assumptions. In *Crypto*, 2019.
28. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *Eurocrypt*, 2004.
29. A. Kiayias and M. Yung. Secure scalable group signature with dynamic joins and separable authorities. *Int. Journal of Security and Networks*, 1(1):24–45, 2006.
30. J. Kilian and E. Petrank. Identity escrow. In *Crypto*, 1998.
31. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, 2006.
32. M. Kohlweiss and I. Miers. Accountable tracing signatures. *IACR Cryptology ePrint Archive*, 2014:824, 2014.

33. M. Kohlweiss and I. Miers. Accountable metadata-hiding escrow: A group signature case study. In *PoPETs*, 2015.
34. B. Libert, K. Nguyen, A. Passelègue, and R. Titiu. Simulation-sound arguments for LWE and applications to KDM-CCA2 security. In *Asiacrypt*, 2020.
35. B. Libert and M. Yung. Non-interactive CCA-secure threshold cryptosystems with adaptive security: New framework and constructions. In *TCC*, 2012.
36. J. Liu, V. Wei, and D. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *ACISP*, 2004.
37. V. Lyubashevsky. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In *Asiacrypt*, 2009.
38. H. Maji, M. Prabhakaran, and M. Rosulek. Attribute-based signatures. In *CT-RSA*, 2011.
39. M. Naor. Deniable ring authentication. In *Crypto*, 2002.
40. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, 1990.
41. S. Noether and A. Mackenzie. Ring confidential transactions. *Ledger*, 1:1–18, 2016.
42. C. Peikert and S. Shiehian. Non-interactive zero knowledge for NP from (plain) Learning With Errors. In *Crypto*, 2019.
43. C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Crypto*, 1991.
44. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.
45. R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Asiacrypt*, 2001.
46. Y. Sakai, K. Emura, G. Hanaoka, Y. Kawai, T. Matsuda, and K. Omote. Group signatures with message-dependent opening. In *Pairing*, 2012.
47. I. Teranishi, J. Furukawa, and K. Sako. k-times anonymous authentication (extended abstract). In *Asiacrypt*, 2004.
48. S. Xu and M. Yung. Accountable ring signatures: A smart card approach. In *CARDIS*, 2004.
49. R. Yang, M.-H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *Crypto*, 2019.