Hidden Identity-Based Signatures

Aggelos Kiayias [*] Hong-Sheng Zhou[*]

Abstract

This paper introduces Hidden Identity-based Signatures (Hidden-IBS), a type of digital signatures that provide mediated signer-anonymity on top of Shamir's Identity-based signatures. The motivation of our new signature primitive is to resolve an important issue with the kind of anonymity offered by "group signatures" where it is required that either the group membership list is *public* or that the opening authority is *dependent* on the group manager for its operation. Contrary to this, Hidden-IBS do not require the maintenance of a group membership list and they enable an opening authority that is totally independent of the group manager. As we argue this makes Hidden-IBS much more attractive than group signatures for a number of applications. In this paper, we provide a formal model of Hidden-IBS as well as two efficient constructions that realize the new primitive. Our elliptic curve construction that is based on the SDH/DLDH assumptions produces signatures that are merely half a Kbyte long and can be implemented very efficiently.

To demonstrate the power of the new primitive, we apply it to solve a problem of current onion-routing systems focusing on the Tor system in particular. Posting through Tor is currently blocked by sites such as Wikipedia due to the real concern that anonymous channels can be used to vandalize online content. By injecting a Hidden-IBS inside the header of an HTTP POST request and requiring the exit-policy of Tor to forward only properly signed POST requests, we demonstrate how sites like Wikipedia may allow anonymous posting while being ensured that the recovery of (say) the IP address of a vandal would be still possible through a dispute resolution system. Using our new Hidden-IBS primitive in this scenario allows to keep the listing of identities (e.g., IP addresses) of Tor users computationally hidden while maintaining an independent Opening Authority which would not have been possible with previous approaches.

Keywords. Anonymity and Privacy, Identity-based schemes, Digital Signatures, Onion-routing, Tor, Wikipedia.

1 Introduction

Anonymity and privacy is an issue of increasing concern in the Internet and the offering of services such as anonymous channels is an important aspect of the future Internet infrastructure if we want to retain fundamental rights such as free speech. Still, anonymous systems are plagued by the potential of misuse and any system that permits strong anonymity seems to be doomed to be of limited use in one sense or another. To see this point consider the recent example of Tor [Tor], an onion-routing system, and how Tor traffic is currently handled by Wikipedia [Wik]. While Wikipedia allows HTTP "GET requests" from Tor, it does not allow editing (i.e., HTTP "POST requests") since allowing such requests opens the possibility to malicious users to vandalize the content of the web-site (actually the Wikipedia suggests to disable privacy in Tor in order to publish to the web-site through the onion-router, see [Wik06]). For similar reasons, Tor's "exit policy" drops all SMTP packets (i.e., packets directed to port 25) to make sure that spammers do not take advantage of the anonymity offered by Tor.

The above two examples exemplify the fact that anonymous communication systems such as Tor *limit their scope* due to the potential of misuse. And it is conceivable that the increase of malicious activity

[*] University of Connecticut, Computer Science and Engineering, Storrs, CT, USA, {aggelos,hszhou}@cse.uconn.edu.

trafficking through anonymous communication networks (that includes the distribution of child pornography for example) will force such networks to become even more restricted in scope something that in turn will nullify the purpose they were built originally (to protect free speech and enable anonymous communication for legal uses).

Misusing anonymity is by no means a new idea: for example the work of [vSN92] shows how anonymous e-cash can be used to commit a perfect crime. For this reason primitives such as fair off-line cash [CMS96, FTY96] were proposed where it is possible for an authority to manage anonymity and reveal the identities of the entities behind a certain transaction given that certain conditions are satisfied. It should be stressed that the existence of such "anonymity mediation" systems are not restricting anonymity but rather *enhance it* since they make it possible to employ anonymous systems in cases where no such system may be allowed to exist (due to regulation and potential of misuse etc.).

Group signatures, introduced in [CvH91], and further studied in a number of works [CP94, Cam97, CS97, KP98, CM98, CM99, AT99, ACJT00, CL01, Son01, CL02a, AST02, BMW03, KY03, AdM03, TX03, KTY04, BBS04, CG04, FY04, CL04, NSN04, BS04, BSZ05, KY05a, KY05b, FI05, BW06, ACHdM05] constitute a tool that can be used to offer such mediated anonymity. Indeed, in a group signature it is possible for users to join the group and obtain a credential from the group manager (GM); subsequently, users can issue signatures that a verifier can identify as signatures originating from a group member but she cannot tell which member is issuing the signature. At the same time an opening authority (OA) is capable, given an "offending" signature, to recover a piece of information that leads to the identity of the signer.

However, as we notice in this work, if one tries to employ group signatures to mediate anonymity in an anonymous credential system, a fundamental problem arises:

The Anonymity Catch-22 of Group Signatures. In Heller's novel [Hel61] Catch-22 refers to a no-win situation; a certain setting where no matter what you do you lose. Here we argue that a similar "Catch-22" scenario occurs when one applies group signatures to mediate anonymity in an anonymous credential system.

To see the problem consider the following sequence of objectives: our primary goal is to (*i*) maximize anonymity and its scope; now given that perfect anonymity would be of limited scope, this implies that we need to: (*ii*) employ an opening authority; now, once the OA is allowed, one would want this entity to be managed properly and thus this brings forth: (*iii*) the OA should be separated from the GM (the registration service) and preferably be a "threshold entity" where many share-holders should be allowed to participate equally in the decision-making process of opening an offending signature.

Now recall the following: in *all* group signature schemes the OA is incapable of recovering the identity of the signer without comparing the information recovered from the signature to a *name directory* (essentially a group membership database that acts as PKI) that is maintained by the GM (this is even true in recent "identity-based" group signature [WYZ05]). With respect to the membership directory thus, it should be that either (*iv*-1) the group member directory is public knowledge, or (*iv*-2) the group member directory is kept secret by the GM. But if (*iv*-1) is true, our objective (*i*) is violated: publishing the list of users that take advantage of an anonymous service in most cases would be the most serious privacy violation possible! (indeed publishing the list of users that use an anonymous service maybe enough to incriminate them if someone wishes their persecution). On the other hand, if (*iv*-2) is true, objective (*iii*) is violated since the OA cannot open an offending signature without the help of the GM. This means that the GM can effectively produce a *denial of service* to any entity that requires the assistance of the OA and thus the OA cannot really guarantee to a service provider that it can open an offending signature. This in turn leads to the OA being less credible and may lead to service providers restricting the use of the anonymous system something that in turn hurts anonymity. Thus no matter how one deploys group signatures, privacy is being reduced.

Resolving this "Anonymity Catch-22" issue of group signatures requires a new signature primitive that we introduce in this work:

Our Contribution: Hidden Identity-Based Signatures. In this work we propose a new digital signature scheme that offers anonymity that can be mediated and is based on the concept of Identity-based signatures (IBS) [Sha84]. In a Hidden-IBS scheme, a signer obtains her signing key by communicating to an identity manager (IM) and negotiating her identity with IM. Given the secret-key the signer can produce signatures on a given message so that her identity is not revealed to the verifier. Still, the verifier is ensured of the fact that the identity negotiation has taken place between the signer and the IM and moreover that the signature *contains the name of the signer in enciphered form* and such name can be recovered by an opening authority.

Hidden-IBS resolve the Anonymity Catch-22 of group signatures since they allow the OA to recover the identity of the signer (i) without having to consult with the IM (which substitutes the GM in the Hidden-IBS setting) and (ii) without requiring the IM to publish a listing of users of the anonymous signatures. See Figure 1.

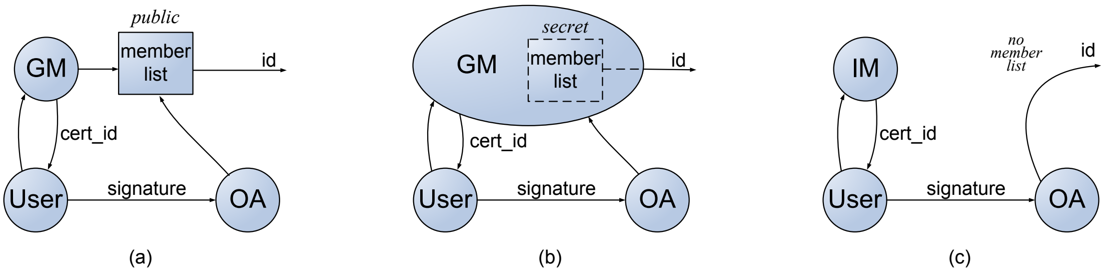

Figure 1: Comparison of the opening functionality between group signatures and Hidden-IBS: (a) group signature with public group membership list, (b) group signature with secret group-membership list, (c) Hidden-IBS.

We note that in a Hidden-IBS the identity of the signer may be equal to any piece of information that is considered acceptable under the policy of the IM, e.g., it can be the signer's e-mail address, the signer's IP address and so forth. Note that the IM and the signer may execute a multi-round protocol to establish the validity of the signer's identity (e.g., the IM may send a verification e-mail to the signer's e-mail account etc.).

In this work we present a formal model of Hidden-IBS, that captures two intuitive properties, misidentification-forgery and anonymity and also consider how the property of exculpability can also be achieved. We then present two constructions of Hidden-IBS, one over elliptic curve groups that is based on the Strong Diffie-Hellman assumption and the Decisional Linear Diffie-Hellman assumption that is merely 533 bytes long and one based on the Strong-RSA assumption and Decisional Composite Residuosity assumption that also achieves security against a malicious IM (called exculpability in the setting of group signatures).

Application to Onion Routing. We demonstrate how Hidden-IBS can be applied to onion-routing [GRS96] and in particular to the Tor system [Tor, DMS04] to allow mediation of anonymity and thus increase rather than limit the scope of such anonymous communication systems.

In Tor, each user transmits messages through a local Onion Proxy (OP) that allows a local host to build circuits hopping along a sequence of onion-routers (OR). Tor has an exit policy: packets that match certain conditions may be dropped (e.g., Tor drops packets directed to port 25 (SMTP) to prevent spammers from using Tor).

Our Hidden-IBS enhanced version of Tor is as follows: there will be certain types of traffic that the exit-policy of Tor will require to be signed with a Hidden-IBS. These may include HTTP POST requests directed to Wikipedia sites and traffic directed to the SMTP port (we stress that most of the traffic will be excluded from the requirement of being signed and thus the performance overhead of our extension would be low). Tor users that wish to use Tor for “sensitive traffic” will be given a list of IM’s and explain the conditions of usage as well as they will be informed of the identity of the OA (which ideally will be a threshold entity). The IM’s that will employed by Tor will require very little information from the users: in particular the identity of a signer can be simply *a verified e-mail address* or the signer’s *IP-address* (of course the identity information required by the IM can be calibrated accordingly). The OP of a user will detect that the user wishes to transmit something that according to the exit-policy must be signed and will redirect the user to obtain a Hidden-IBS secret-key that will allow the signing of the message. The Hidden-IBS will be injected into the packet itself (e.g., in the case of an HTTP POST we will use a special header field to contain the Hidden-IBS of the HTTP packet) and the signature will be verified by the Tor exit point. The ciphertext along with the necessary information to recover the identity of the user (if ever needed) will be posted in a public “Disputes&Grievances” database. The database will be designed in such a way so that it retains no publicly readable information about the identities of Tor users or the traffic they produce (only hashed packets and ciphertexts will be stored in the database). Still, the database will make it possible for any website that has received offending Tor traffic to submit a complaint to the OA that, if accepted, it will recover either the IP address or the e-mail of the culprit. Subsequently the identity may be blacklisted by the IM or receive negative points in a reputation system. Given that a Hidden-IBS signing credential would expire in short time periods the offender will have to face the outcome of his adverse behavior (lower reputation score with the IM, or being blacklisted etc.).

Organization. In Section 2 we introduce our model for the new primitive. Our basic Hidden-IBS construction in Section 3 is based on elliptic curve groups and pairing, and employs the digital signature of Boneh-Boyen (BB) signature which is based on the Strong Diffie-Hellman (SDH) assumption [BB04], and the Linear Encryption based on the Decision Linear Diffie-Hellman (DLDH) assumption [BBS04]. Given this construction we explain in more detail the way that our Hidden-IBS enhanced anonymous routing system is designed in Section 4. Wedescribe the problem of exculpability in the context of Hidden-IBS and provide an extended model to capture the setting of corrupted IM’s in Section 5. In our second construction presented in Section 6, we use Camenisch-Lysyanskaya (CL) signature which is based on the Strong RSA assumption [CL02b]. Also we use Paillier encryption based on the Decisional Composite Residuosity (DCR) assumption [Pai99]. We include the description of the cryptographic primitives we use in Section A.1 in the appendix.

2 Hidden-IBS: Modelling

2.1 Syntax

In this section, we give the definition of Hidden-IBS. First, we start with the syntax of the scheme. The parties are involved in the scheme include the identity manager IM, the open authority OA, the users U, and the verifiers V,

Definition 2.1. A *hidden identity-based signature* (Hidden-IBS) scheme is a digital signature scheme that consists of six polynomial-time algorithms $\langle \mathtt{Setup}, \mathtt{Reg}, \mathtt{Sign}, \mathtt{RegCheck}, \mathtt{Verify}, \mathtt{Open} \rangle$. The first three algorithms are probabilistic but the last three are not necessarily.

`Setup`: The `Setup` algorithm includes `SetupIM` and `SetupOA`. On input a security parameter, first the global system parameter is generated. Then on input a security parameter and the system parameter, the probabilistic algorithm `SetupIM` outputs the group verification key pk_{IM} and the signing key sk_{IM} for the identity manager, the probabilistic algorithm `SetupOA` outputs the public key pk_{OA} and the secret key sk_{OA} for the open authority. The `Setup` algorithm may include `SetupUser`; on the input a security parameter and the system parameter, outputs `id` for both the identity manager and the user.

`Reg`: A probabilistic algorithm that given an identity manager's verification key, an identity manager's signing key, a user's identity `id` outputs a membership certificate $\mathtt{cert}_{\mathtt{id}}$ for the identity `id`. We write $\mathtt{Reg}(pk_{\mathsf{IM}}; sk_{\mathsf{IM}}, \mathtt{id})$ to denote the registration algorithm.

`RegCheck`: An algorithm for user's checking the validity of the certificate for her identity with respect to an identity manager's public key. We denote the application of the registration checking algorithm as $\mathtt{RegCheck}(pk_{\mathsf{IM}}; \mathtt{id}, \mathtt{cert}_{\mathtt{id}}) \in \{0, 1\}$.

`Sign`: A probabilistic algorithm that given an identity manager's public key, an open authority's public key, a user's identity, a membership certificate on the user's identity, and a message m, outputs a signature for the message m. We write $\mathtt{Sign}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}, \mathtt{id}; \mathtt{cert}_{\mathtt{id}}, m)$ to denote the application of the signing algorithm.

`Verify`: An algorithm for establishing the validity of an alleged Hidden-IBS signature of a message with respect to an identity manager's verification key and an open authority's public key. If σ is a signature on a message m, then we have $\mathtt{Verify}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; m, \sigma) \in \{0, 1\}$.

`Open`: An algorithm that given a message, a valid Hidden-IBS signature on it, an identity manager's verification key, an open authority's public key, and an open authority secret key, determines the `id` directly. In particular $\mathtt{id} \leftarrow \mathtt{Open}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{OA}}, m, \sigma)$.

2.2 Correctness and Security

In this section we provide the definitions of correctness and security of a Hidden-IBS scheme. We begin with correctness and then define misidentification-forgery and anonymity.

Definition 2.2 (Correctness). The correctness of the Hidden-IBS include the registration correctness, the signing correctness, and the opening correctness.
Registration Correctness means that the IM issues only one valid membership certificate for each different `id`, which is defined as below,

$$\Pr\left[\begin{array}{l} (pk_{\mathsf{IM}}, sk_{\mathsf{IM}}) \leftarrow \mathtt{SetupIM}(1^\lambda); \\ (pk_{\mathsf{OA}}, sk_{\mathsf{OA}}) \leftarrow \mathtt{SetupOA}(1^\lambda); \\ \mathtt{cert}_{\mathtt{id}} \leftarrow \mathtt{Reg}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{IM}}, \mathtt{id}); \\ \qquad\qquad : \mathtt{RegCheck}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; \mathtt{id}, \mathtt{cert}_{\mathtt{id}}) = 1 \end{array}\right] = 1$$

Signing Correctness ensures that the correctness of the underlying signing and verification algorithms for any valid signing key.

$$\Pr\left[\begin{array}{l} (pk_{\mathsf{IM}}, sk_{\mathsf{IM}}) \leftarrow \mathtt{SetupIM}(1^\lambda); \\ (pk_{\mathsf{OA}}, sk_{\mathsf{OA}}) \leftarrow \mathtt{SetupOA}(1^\lambda); \\ \mathtt{cert}_{\mathtt{id}} \leftarrow \mathtt{Reg}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{IM}}, \mathtt{id}); \\ \mathtt{RegCheck}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; \mathtt{id}, \mathtt{cert}_{\mathtt{id}}) = 1; \\ \sigma \leftarrow \mathtt{Sign}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}, \mathtt{id}; \mathtt{cert}_{\mathtt{id}}, m) \\ \qquad\qquad : \mathtt{Verify}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; m, \sigma) = 1 \end{array}\right] = 1$$

Opening Correctness ensures that the `Open` algorithm can correctly identifies all signers from a valid signature, which is defined as below,

$$\Pr\left[\begin{array}{l}(pk_{\mathsf{IM}}, sk_{\mathsf{IM}}) \leftarrow \mathtt{SetupIM}(1^{\lambda});\\ (pk_{\mathsf{OA}}, sk_{\mathsf{OA}}) \leftarrow \mathtt{SetupOA}(1^{\lambda});\\ \mathtt{cert}_{\mathtt{id}} \leftarrow \mathtt{Reg}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{IM}}, \mathtt{id});\\ \mathtt{RegCheck}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; \mathtt{id}, \mathtt{cert}_{\mathtt{id}}) = 1;\\ \sigma \leftarrow \mathtt{Sign}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}, \mathtt{id}; \mathtt{cert}_{\mathtt{id}}, m);\\ \mathtt{Verify}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; m, \sigma) = 1\\ \qquad\qquad : \mathtt{Open}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{OA}}, m, \sigma) = \mathtt{id}\end{array}\right] = 1$$

Next, we proceed to the definition of security which is comprised of two different properties: misidentification-forgery where the attacker either manages to be not properly identified or forges a signature and anonymity where the attacker extracts some information about the signer's identity.

In a misidentification-forgery attack, the adversary is allowed to corrupt the registered honest users. Also the adversary is capable of corrupting the OA. The adversary is allowed to adaptively ask signing queries from the honest users. The adversary wins the game if it either produces a message-signature pair that foils the opening procedure (i.e. misidentification), or forges a message-signature which can be opened to an identity but the message has never been queried in the history of the user with the identity (i.e. forgery).

Definition 2.3 (Misidentification-Forgery). We say a Hidden-IBS scheme is against misidentification-forgery attacks if for any PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{misid}}(\lambda)$ is negligible in λ, where $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{misid}}(\lambda) = \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{misid}}(\lambda) = 1]$, where the experiment defined as in Figure 2.

```
RegOracle(id)
    If $\mathtt{id} \in CU \cup HU$ then return $\perp$;
    $\mathtt{cert}_{\mathtt{id}} \leftarrow \mathtt{Reg}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{IM}}, \mathtt{id})$;
    $MSG_{\mathtt{id}} \leftarrow \emptyset; HU \leftarrow HU \cup \{\mathtt{id}\}$;
    Return 1;

SignOracle($\mathtt{id}, \mathtt{cert}_{\mathtt{id}}, m$)
    If $\mathtt{id} \notin HU$ then return $\perp$;
    $\sigma \leftarrow \mathtt{Sign}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}, \mathtt{id}; \mathtt{cert}_{\mathtt{id}}, m)$;
    $MSG_{\mathtt{id}} \leftarrow MSG_{\mathtt{id}} \cup \{m\}$;
    Return $\sigma$;

CorruptUOracle(id)
    If $\mathtt{id} \notin HU$ then return $\perp$;
    $CU \leftarrow CU \cup \{\mathtt{id}\}; HU \leftarrow HU \backslash \{\mathtt{id}\}$
    Return $\mathtt{cert}_{\mathtt{id}}$;

CorruptOAOracle()
    Return $sk_{\mathsf{OA}}$;

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{misid}}(\lambda)$
    $(pk_{\mathsf{IM}}, sk_{\mathsf{IM}}) \leftarrow \mathtt{SetupIM}(1^{\lambda}); (pk_{\mathsf{OA}}, sk_{\mathsf{OA}}) \leftarrow \mathtt{SetupOA}(1^{\lambda}); HU \leftarrow \emptyset; CU \leftarrow \emptyset$
    $(m, \sigma) \leftarrow \mathcal{A}^{\mathsf{RegOracle}(), \mathsf{SignOracle}(), \mathsf{CorruptUOracle}(), \mathsf{CorruptOAOracle}()}(1^{\lambda}, pk_{\mathsf{IM}}, pk_{\mathsf{OA}})$
    If $\mathtt{Verify}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; m, \sigma) = 1 \wedge \mathtt{Open}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{OA}}, m, \sigma) = \perp$ then return 1;
    If $\mathtt{Verify}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; m, \sigma) = 1 \wedge \mathtt{Open}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{OA}}, m, \sigma) = \mathtt{id} \wedge m \notin MSG_{\mathtt{id}} \wedge \mathtt{id} \notin CU$
        then return 1;
    Return 0;
```

Figure 2: Experiment of misidentification-forgery

Next we define the anonymity notions. The CCA2-anonymity attack can be modelled as a CCA2 attack against the identity encryption of the Hidden-IBS, while the CPA-anonymity attack be modelled as a CPA attack.

Definition 2.4 (CCA2-Anonymity). We say a Hidden-IBS scheme is against anonymity attacks if for any PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{cca2-anon}}(\lambda)$ is negligible in λ, where $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{cca2-anon}}(\lambda) = \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{cca2-anon},1}(\lambda) = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{cca2-anon},0}(\lambda) = 1]$, where the experiment defined as in Figure 3.

Definition 2.5 (CPA-Anonymity). We say a Hidden-IBS scheme is against CPA-anonymity attacks if for any PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{cpa-anon}}(\lambda)$ is negligible in λ, where $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{cpa-anon}}(\lambda) = \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{cpa-anon},1}(\lambda) = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{cpa-anon},0}(\lambda) = 1]$, where the experiment defined as in Figure 4.

```
$\mathsf{OpenOracle}(m, \sigma)$
    If $\mathtt{Verify}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; m, \sigma) = 1$
        then return $\mathtt{Open}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{OA}}, m, \sigma)$
    Return $\perp$;
```

```
$\mathsf{CorruptIMOracle}()$
    Return $sk_{\mathsf{IM}}$;
```

```
Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{cca2-anon},b}(\lambda)$
    $(pk_{\mathsf{IM}}, sk_{\mathsf{IM}}) \leftarrow \mathtt{SetupIM}(1^\lambda)$;$(pk_{\mathsf{OA}}, sk_{\mathsf{OA}}) \leftarrow \mathtt{SetupOA}(1^\lambda)$;
    $(\mathtt{id}_0, \mathtt{id}_1, m) \leftarrow \mathcal{A}^{\mathsf{CorruptIMOracle}(),\mathsf{OpenOracle}()}(1^\lambda, pk_{\mathsf{IM}}, pk_{\mathsf{OA}})$;
    $\mathtt{cert}_{\mathtt{id}_0} \leftarrow \mathtt{Reg}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{IM}}, \mathtt{id}_0)$; $\mathtt{cert}_{\mathtt{id}_1} \leftarrow \mathtt{Reg}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{IM}}, \mathtt{id}_1)$;
    $\sigma \leftarrow \mathtt{Sign}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}, \mathtt{id}_b; \mathtt{cert}_{\mathtt{id}_b}, m)$;
    $b^* \leftarrow \mathcal{A}^{\mathsf{CorruptIMOracle}(),\mathsf{OpenOracle}^{\neg\sigma}()}(1^\lambda, pk_{\mathsf{IM}}, pk_{\mathsf{OA}})$;
    Return $b^*$;
```

Figure 3: Experiment of CCA2-anonymity. In the experiment above, $\mathsf{OpenOracle}^{\neg\sigma}()$ operates as the $\mathsf{OpenOracle}()$ with the restriction that it will return $\perp$ if the adversary submit σ as the signature to be opened.

```
Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{cpa-anon},b}(\lambda)$
    $(pk_{\mathsf{IM}}, sk_{\mathsf{IM}}) \leftarrow \mathtt{SetupIM}(1^\lambda)$;$(pk_{\mathsf{OA}}, sk_{\mathsf{OA}}) \leftarrow \mathtt{SetupOA}(1^\lambda)$;
    $(\mathtt{id}_0, \mathtt{id}_1, m) \leftarrow \mathcal{A}^{\mathsf{CorruptIMOracle}()}(1^\lambda, pk_{\mathsf{IM}}, pk_{\mathsf{OA}})$;
    $\mathtt{cert}_{\mathtt{id}_0} \leftarrow \mathtt{Reg}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{IM}}, \mathtt{id}_0)$; $\mathtt{cert}_{\mathtt{id}_1} \leftarrow \mathtt{Reg}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{IM}}, \mathtt{id}_1)$;
    $\sigma \leftarrow \mathtt{Sign}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}, \mathtt{id}_b; \mathtt{cert}_{\mathtt{id}_b}, m)$;
    $b^* \leftarrow \mathcal{A}^{\mathsf{CorruptIMOracle}()}(1^\lambda, pk_{\mathsf{IM}}, pk_{\mathsf{OA}})$;
    Return $b^*$;
```

Figure 4: Experiment of CPA-anonymity. In the experiment above, the $\mathsf{CorruptIMOracle}()$ used is same as that in the CCA2 version, and the $\mathsf{OpenOracle}()$ is not allowed.

Remark 2.6. Our modeling of the Hidden-IBS primitive is extending the basic modeling of identity-based signatures. As it was described in [BNN04] there is a relation between Identity-based signatures and Identity-based identification (IBI) schemes. In a similar way we can formalize Hidden-IBI schemes and in fact our two constructions would yield two Hidden-IBI schemes (that in fact would be provably secure without random oracles). The relation of Hidden-IBS and Hidden-IBI is parallel to the relation between group signatures and identity escrow [KP98].

3 Hidden-IBS: Construction

In this section we describe our first Hidden-IBS construction. It is geared towards producing short signatures and is suitable for relatively short identity strings (e.g., IP addresses of 32 bits). We first describe the scheme and then we prove that the scheme achieves the misidentification-forgery property and the CPA-anonymity property.

3.1 The Scheme

In our Hidden-IBS scheme, we let the IM use the Boneh-Boyen [BB04] signature to issue a certificate to each user identity. Once a user obtains the certificate from the IM, she can generate a Hidden-IBS signature for a message: the user uses Linear encryption [BBS04] to "embed" her identity which can be opened by the OA; the user forms the signature based on a proof of knowledge that ensures her identity, her certificate, and the relations between them are properly formed. We present the details below:

`Setup`. This procedure first generates the system parameters including the bilinear group parameter $\langle p, g, \widehat{g}, \mathbb{G}, \widehat{\mathbb{G}}, \psi, \mathbb{G}_T, e \rangle$, random element $\widehat{h} \xleftarrow{r} \widehat{\mathbb{G}} \backslash \{1\}$ and $h = \psi(\widehat{h})$, and a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_p$ which will be treated as a random oracle in the security proof. Then the algorithm $\mathtt{SetupIM}$ generates key pair $(pk_{\mathsf{IM}}, sk_{\mathsf{IM}})$: selects $x, y \overset{\mathtt{r}}{\leftarrow} \mathbb{Z}_p^*$ and compute $\widehat{X} = \widehat{g}^x$ and $\widehat{Y} = \widehat{g}^y$; sets $pk_{\mathsf{IM}} = \langle \widehat{X}, \widehat{Y} \rangle$, and sets $sk_{\mathsf{IM}} = \langle x, y \rangle$. The algorithm $\mathtt{SetupOA}$ generates key pair $(pk_{\mathsf{OA}}, sk_{\mathsf{OA}})$: selects $\widehat{w} \overset{\mathtt{r}}{\leftarrow} \widehat{\mathbb{G}} \backslash \{1\}$, selects $\delta, \xi \overset{\mathtt{r}}{\leftarrow} \mathbb{Z}_p^*$ and sets $\widehat{u}, \widehat{v} \in \widehat{\mathbb{G}}$ such that $\widehat{u}^{\zeta} = \widehat{v}^{\eta} = \widehat{w}$; sets $w = \psi(\widehat{w})$, $u = \psi(\widehat{u})$, $v = \psi(\widehat{v})$; note that $u^{\zeta} = v^{\eta} = w$ holds; sets $pk_{\mathsf{OA}} = \langle u, v, w, \widehat{u}, \widehat{v}, \widehat{w} \rangle$ and $sk_{\mathsf{OA}} = \langle \zeta, \eta \rangle$. Finally sets the public parameters for the Hidden-IBS as $\mathsf{pub} = \langle p, g, \widehat{g}, h, \widehat{h}, \mathbb{G}, \widehat{\mathbb{G}}, \psi, \mathbb{G}_T, e; \widehat{X}, \widehat{Y}; u, v, w, \widehat{u}, \widehat{v}, \widehat{w}; \mathcal{H} \rangle$. We still need to prescribe the form of the user identities: each identity is a short string with length ℓ. For example, it can be an IP address with $\ell = 32$ or a userid in a reputation system (e.g., using $\ell = 50$ we can allow 10 character long userids with 5 bits per character).

$\mathtt{Reg}$. In the registration algorithm, the user sends her identity $\mathtt{id}$ to the IM. The IM verifies that $\mathtt{id}$ is acceptable (e.g., not being used before or not blacklisted etc.). We note that the $\mathtt{id}$ can also be a product of a negotiation between the IM and the users. Then the IM generates a BB signature $\langle s, r \rangle$ for $\mathtt{id}$, where $s \leftarrow g^{\frac{1}{x + \mathtt{id} + yr}}$, $r \overset{\mathtt{r}}{\leftarrow} \mathbb{Z}_p$, and sends $\langle s, r \rangle$ to the user by a secure communication channel.

$\mathtt{RegCheck}$. Once receiving the signature $\langle s, r \rangle$ from the IM, the user verifies $e(s, \widehat{X}\widehat{g}^{\mathtt{id}}\widehat{Y}^r) = e(g, \widehat{g})$. The user sets her membership certificate to $\mathtt{cert}_{\mathtt{id}} = \langle s, r \rangle$.

$\mathtt{Sign}$. With a membership certificate $\mathtt{cert}_{\mathtt{id}} = \langle s, r \rangle$ in hand, a user can compute a Hidden-IBS signature σ for message m. We first develop a proof of knowledge in Figure 5, where the user proves her knowledge of $\mathtt{id}$ and $\mathtt{cert}_{\mathtt{id}}$, and proves that $\mathtt{cert}_{\mathtt{id}}$ is a BB signature of $\mathtt{id}$ from the IM. Then we transform the proof of knowledge into a signing algorithm by using the Fiat-Shamir heuristic.

Next we give a detailed description: (i) the user uses the Linear Encryption to encrypt $g^{\mathtt{id}}$ into $\langle U, V, \widehat{W} \rangle$ (Note that in the open algorithm below, the OA first computes $W = \psi(\widehat{W})$ and opens the ciphertext $\langle U, V, W \rangle$ into $g^{\mathtt{id}}$, then gets $\mathtt{id}$ by "brute force"; here the length of $\mathtt{id}$ is very short, e.g. a 32-bits IP address), where $U = u^k$, $V = v^l$, $\widehat{W} = \widehat{w}^{k+l}\widehat{g}^{\mathtt{id}}$, $k, l \overset{\mathtt{r}}{\leftarrow} \mathbb{Z}_p$; the user commits s into $S = g^{r_1}s$, commits r into $\widehat{R} = \widehat{g}^{r_2}\widehat{h}^{r_1}\widehat{Y}^r$, where $r_1, r_2 \overset{\mathtt{r}}{\leftarrow} \mathbb{Z}_p$. (ii) the user deploys a three move Σ-protocol to prove that she knows the underlying plaintext $g^{\mathtt{id}}$ of $\langle U, V, \widehat{W} \rangle$, and the underlying plaintexts s and r of ciphertexts S and $\widehat{R}$ respectively are a signature for $\mathtt{id}$ based on the BB verification key: compute $\delta_1 = r_1 k$, $\delta_2 = r_1 l$, $\delta_3 = r_1 r_2$, $\delta_4 = r_1^2$, $\delta_5 = r_1 r$; randomly select $\theta_{\mathtt{id}}, \theta_r, \theta_{r_1}, \theta_{r_2}, \theta_k, \theta_l \overset{\mathtt{r}}{\leftarrow} \mathbb{Z}_p$, $\theta_{\delta_1}, \theta_{\delta_2}, \theta_{\delta_3}, \theta_{\delta_4}, \theta_{\delta_5} \overset{\mathtt{r}}{\leftarrow} \mathbb{Z}_p$; and compute $B_1 = u^{-\theta_k}$, $B_2 = v^{-\theta_l}$, $B_3 = \widehat{w}^{-(\theta_k + \theta_l)}\widehat{g}^{-\theta_{\mathtt{id}}}$, $B_4 = \widehat{g}^{-\theta_{r_2}}\widehat{h}^{-\theta_{r_1}}\widehat{Y}^{-\theta_r}$, $B_5 = U^{-\theta_{r_1}}u^{\theta_{\delta_1}}$, $B_6 = V^{-\theta_{r_1}}v^{\theta_{\delta_2}}$, $B_7 = \widehat{R}^{-\theta_{r_1}}\widehat{g}^{\theta_{\delta_3}}\widehat{h}^{\theta_{\delta_4}}\widehat{Y}^{\theta_{\delta_5}}$, $B_8 = e(g, \widehat{X}\widehat{W}\widehat{R})^{\theta_{r_1}} e(S, \widehat{w})^{\theta_k + \theta_l} e(g, \widehat{w})^{-(\theta_{\delta_1} + \theta_{\delta_2})} e(S, \widehat{g})^{\theta_{r_2}}$ $e(g, \widehat{g})^{-\theta_{\delta_3}} e(S, \widehat{h})^{\theta_{r_1}} e(g, \widehat{h})^{-\theta_{\delta_4}}$; compute $c = \mathcal{H}(m||S||\widehat{R}||U||V||\widehat{W}||B_1||...||B_8)$; then compute $\xi_{\mathtt{id}} = \theta_{\mathtt{id}} + c \cdot \mathtt{id}$, $\xi_r = \theta_r + c \cdot r$, $\xi_{r_1} = \theta_{r_1} + c \cdot r_1$, $\xi_{r_2} = \theta_{r_2} + c \cdot r_2$, $\xi_k = \theta_k + c \cdot k$, $\xi_l = \theta_l + c \cdot l$, $\xi_{\delta_1} = \theta_{\delta_1} + c \cdot \delta_1$, $\xi_{\delta_2} = \theta_{\delta_2} + c \cdot \delta_2$, $\xi_{\delta_3} = \theta_{\delta_3} + c \cdot \delta_3$, $\xi_{\delta_4} = \theta_{\delta_4} + c \cdot \delta_4$, $\xi_{\delta_5} = \theta_{\delta_5} + c \cdot \delta_5$. Therefore, the generated signature for message m is the tuple: $\sigma = \langle S, \widehat{R}, U, V, \widehat{W}; c; \xi_{\mathtt{id}}, \xi_r, \xi_{r_1}, \xi_{r_2}, \xi_k, \xi_l, \xi_{\delta_1}, ..., \xi_{\delta_5} \rangle$.

$\mathtt{Verify}$. The verifier can verify a message-signature pair by checking the equation below:

$$\begin{aligned} & c =^? \mathcal{H}\big(m||S||\widehat{R}||U||V||\widehat{W} \\ & ||U^c u^{-\xi_k}||V^c v^{-\xi_l}||\widehat{W}^c \widehat{w}^{-(\xi_k + \xi_l)}\widehat{g}^{-\xi_{\mathtt{id}}}||\widehat{R}^c \widehat{g}^{-\xi_{r_2}}\widehat{h}^{-\xi_{r_1}}\widehat{Y}^{-\xi_r}||U^{-\xi_{r_1}}u^{\xi_{\delta_1}}||V^{-\xi_{r_1}}v^{\xi_{\delta_2}}||\widehat{R}^{-\xi_{r_1}}\widehat{g}^{\xi_{\delta_3}}\widehat{h}^{\xi_{\delta_4}}\widehat{Y}^{\xi_{\delta_5}} \\ & ||e(g, \widehat{X}\widehat{W}\widehat{R})^{\xi_{r_1}} e(S, \widehat{w})^{(\xi_k + \xi_l)} e(g, \widehat{w})^{-(\xi_{\delta_1} + \xi_{\delta_2})} e(S, \widehat{g})^{\xi_{r_2}} e(g, \widehat{g})^{-\xi_{\delta_3}} e(S, \widehat{h})^{\xi_{r_1}} e(g, \widehat{h})^{-\xi_{\delta_4}} (e(g, \widehat{g})/e(S, \widehat{X}\widehat{W}\widehat{R}))^c\big) \end{aligned}$$

$\mathtt{Open}$. Given a message-signature pair as described above, the OA first verifies the message-signature pair. Next the OA uses her secret key $sk_{\mathsf{OA}} = \langle \zeta, \eta \rangle$ to open ciphertext $\langle U, V, W \rangle$ into $g^{\mathtt{id}}$ where $W = \psi(\widehat{W})$; considering that the identity space is small, the OA recovers $\mathtt{id}$ from $g^{\mathtt{id}}$ (see below in performance for more details).

$$\texttt{pub} = \langle p, g, \widehat{g}, \mathbb{G}, \widehat{\mathbb{G}}, \psi, \mathbb{G}_T, e; h, \widehat{h}; \widehat{X}, \widehat{Y}; u, v, w, \widehat{u}, \widehat{v}, \widehat{w} \rangle$$

$\boxed{\texttt{User}}$ $\boxed{\texttt{Verifier}}$

$\texttt{id}, s, r$

$r_1, r_2, k, l \xleftarrow{\mathtt{r}} \mathbb{Z}_p,$
$S = g^{r_1}s,\ \widehat{R} = \widehat{g}^{r_2}\widehat{h}^{r_1}\widehat{Y}^{r},$
$\delta_1 = r_1k,\ \delta_2 = r_1l,$
$\delta_3 = r_1r_2,\ \delta_4 = r_1^2,\ \delta_5 = r_1r$
$U = u^k,\ V = v^l,\ \widehat{W} = \widehat{w}^{k+l}\widehat{g}^{\texttt{id}}$
$\theta_{\texttt{id}}, \theta_r, \theta_{r_1}, \theta_{r_2}, \theta_k, \theta_l \xleftarrow{\mathtt{r}} \mathbb{Z}_p,$
$\theta_{\delta_1}, \theta_{\delta_2}, \theta_{\delta_3}, \theta_{\delta_4}, \theta_{\delta_5} \xleftarrow{\mathtt{r}} \mathbb{Z}_p$
$B_1 = u^{-\theta_k},\ B_2 = v^{-\theta_l},$
$B_3 = \widehat{w}^{-(\theta_k+\theta_l)}\widehat{g}^{-\theta_{\texttt{id}}},$
$B_4 = \widehat{g}^{-\theta_{r_2}}\widehat{h}^{-\theta_{r_1}}\widehat{Y}^{-\theta_r},$
$B_5 = U^{-\theta_{r_1}}u^{\theta_{\delta_1}},\ B_6 = V^{-\theta_{r_1}}v^{\theta_{\delta_2}}$
$B_7 = \widehat{R}^{-\theta_{r_1}}\widehat{g}^{\theta_{\delta_3}}\widehat{h}^{\theta_{\delta_4}}\widehat{Y}^{\theta_{\delta_5}}$
$B_8 = e(g, \widehat{X}\widehat{W}\widehat{R})^{\theta_{r_1}}e(S, \widehat{w})^{\theta_k+\theta_l}\cdot$
$\quad e(g, \widehat{w})^{-(\theta_{\delta_1}+\theta_{\delta_2})}e(S, \widehat{g})^{\theta_{r_2}}\cdot$
$\quad e(g, \widehat{g})^{-\theta_{\delta_3}}e(S, \widehat{h})^{\theta_{r_1}}e(g, \widehat{h})^{-\theta_{\delta_4}}$

$$\xrightarrow{S,\widehat{R},U,V,\widehat{W};B_1,\ldots,B_8}$$

$c \xleftarrow{\mathtt{r}} \mathbb{Z}_p$

$$\xleftarrow{c}$$

$\xi_{\texttt{id}} = \theta_{\texttt{id}} + c\cdot\texttt{id},\ \xi_r = \theta_r + c\cdot r,$
$\xi_{r_1} = \theta_{r_1} + c\cdot r_1,\ \xi_{r_2} = \theta_{r_2} + c\cdot r_2$
$\xi_k = \theta_k + c\cdot k,\ \xi_l = \theta_l + c\cdot l$
$\xi_{\delta_1} = \theta_{\delta_1} + c\cdot\delta_1,\ \xi_{\delta_2} = \theta_{\delta_2} + c\cdot\delta_2$
$\xi_{\delta_3} = \theta_{\delta_3} + c\cdot\delta_3,\ \xi_{\delta_4} = \theta_{\delta_4} + c\cdot\delta_4$
$\xi_{\delta_5} = \theta_{\delta_5} + c\cdot\delta_5,$

$$\xrightarrow{\xi_{\texttt{id}},\xi_r,\xi_{r_1},\xi_{r_2},\xi_k,\xi_l,\xi_{\delta_1},\ldots,\xi_{\delta_5}}$$

$u^{\xi_k}B_1 =^? U^c,\ v^{\xi_l}B_2 =^? V^c$
$\widehat{w}^{\xi_k+\xi_l}\widehat{g}^{\xi_{\texttt{id}}}B_3 =^? \widehat{W}^c$
$\widehat{g}^{\xi_{r_2}}\widehat{h}^{\xi_{r_1}}\widehat{Y}^{\xi_r}B_4 =^? \widehat{R}^c$
$U^{\xi_{r_1}}u^{-\xi_{\delta_1}}B_5 =^? 1,\ V^{\xi_{r_1}}v^{-\xi_{\delta_2}}B_6 =^? 1$
$\widehat{R}^{\xi_{r_1}}\widehat{g}^{-\xi_{\delta_3}}\widehat{h}^{-\xi_{\delta_4}}\widehat{Y}^{-\xi_{\delta_5}}B_7 =^? 1$
$e(g, \widehat{X}\widehat{W}\widehat{R})^{-\xi_{r_1}}e(S, \widehat{w})^{-(\xi_k+\xi_l)}\cdot$
$\quad e(g, \widehat{w})^{(\xi_{\delta_1}+\xi_{\delta_2})}e(S, \widehat{g})^{-\xi_{r_2}}\cdot$
$\quad e(g, \widehat{g})^{\xi_{\delta_3}}e(S, \widehat{h})^{-\xi_{r_1}}e(g, \widehat{h})^{\xi_{\delta_4}}B_8$
$\quad =^? (e(g, \widehat{g})/e(S, \widehat{X}\widehat{W}\widehat{R}))^c$

Figure 5: The hidden identity-based identification protocol.

3.2 Efficiency of the Scheme

Signature Length. A generated Hidden-IBS signature includes 3 elements of $\mathbb{G}$, 2 elements of $\widehat{\mathbb{G}}$, and 12 elements of $\mathbb{Z}_p$. Using the families of curves described in [BLS04], we take p to be a 170-bit prime and use a group $\mathbb{G}$ where each element is 171 bits and a group $\widehat{\mathbb{G}}$ where each element 6×171bits, and the total signature length is 4605 bits or 576 bytes. The security is approximately the same as a standard 1024-bit RSA signature (that offers no anonymity whatsoever) and has length 128 bytes.

Performance. Consider that both signer and verifier can precompute the pairings $e(g, \widehat{w})$, $e(g, \widehat{g})$, and $e(g, \widehat{h})$; the signers can compute $e(S, \widehat{w}^{\theta_k + \theta_l} \widehat{g}^{\theta_{r_2}} \widehat{h}^{\theta_{r_1}})$ instead of $e(S, \widehat{w})^{\theta_k + \theta_l} e(S, \widehat{g})^{\theta_{r_2}} e(S, \widehat{h})^{\theta_{r_1}}$; and the verifier can compute $e(S, \widehat{w}^{\xi_k + \xi_l} \widehat{g}^{\xi_{r_2}} \widehat{h}^{\xi_{r_1}})$ instead of $e(S, \widehat{w})^{\xi_k + \xi_l} e(S, \widehat{g})^{\xi_{r_2}} e(S, \widehat{h})^{\xi_{r_1}}$, and compute $e(g^{-\xi_{r_1}} S^c, \widehat{X}\widehat{W}\widehat{R})$ instead of $e(g, \widehat{X}\widehat{W}\widehat{R})^{-\xi_{r_1}} e(S, \widehat{X}\widehat{W}\widehat{R})^c$. Thus generating a Hidden-IBS signature requires 14 multi-exponentiations (or exponentiation) and 2 pairing computations; and verifying a group signature requires 10 multi-exponentiations (or exponentiations) and 2 pairing computations. In opening, we first compute $W = \psi(\widehat{W})$, which takes roughly the same time as an exponentiation in group $\mathbb{G}$; then we decrypt ciphertext $\langle U, V, W \rangle$ into $g^{\mathtt{id}} = W U^{-\zeta} V^{-\eta}$, we need 1 multi-exponentiation. Then using for example Pollard's rho method [Pol75] the opening authority extracts `id` from $g^{\mathtt{id}}$ in $\tilde{O}(\sqrt{2^\ell})$ steps.

We note that we designed our scheme with a superpolynomial-time in the identity length ℓ opening algorithm for the sake of reducing the signature size. If a more efficient opening is desired and the signature length is of less importance, one can use our scheme in Section 6 that has a polynomial in ℓ opening operation.

3.3 Correctness and Security

Theorem 3.1 (Correctness). *The Hidden-IBS scheme of Section 3.1 is correct.*

Theorem 3.2 (Misidentification-Forgery). *In the random oracle model, the Hidden-IBS scheme of Section 3.1 satisfies the misidentification-forgery property if the SDH assumption holds.*

Theorem 3.3 (CPA-Anonymity). *In the random oracle model, the Hidden-IBS scheme of Section 3.1 is CPA-Anonymous if the DLDH assumption holds.*

Based on Theorem 3.1, Theorem 3.2 and Theorem 3.3, we have

Theorem 3.4. *The Hidden-IBS scheme of Section 3.1 is correct and secure satisfying misidentification-forgery and CPA-anonymity in the random oracle model under the SDH and the DLDH assumptions.*

4 Reducing Abuse in Anonymous Routing Systems

As mentioned in the introduction some internet services block certain types of traffic coming through anonymous routing systems in order to maintain the quality of their service (e.g., in the case of Wikipedia, POST requests coming from Tor are blocked to prevent vandalism). This practice stems from the fact that anonymous routing systems such as Tor have no built-in mechanisms to handle abusive users. In this section, we show how using our Hidden-IBS we can strengthen the Tor network with the capability to defend itself against such abusive users.

Our approach, outlined in Figure 6, adds three entities to the Tor network deployment: the Identity Manager (IM) of a Hidden-IBS, a Disputes&Grievances database and the Opening Authority (OA) of the Hidden-IBS. Our basic idea is to show how a service web-site that receives Tor traffic can complain about malicious requests (e.g., vandalism in the case of Wikipedia) and recover some information about the offending users. In this way the anonymous routing system offers a mechanism to prevent abusive users from taking advantage of anonymity and thus its services can be granted higher functionality by service providers. Our enhancement to Tor will be totally transparent to service web-sites that receive Tor traffic.

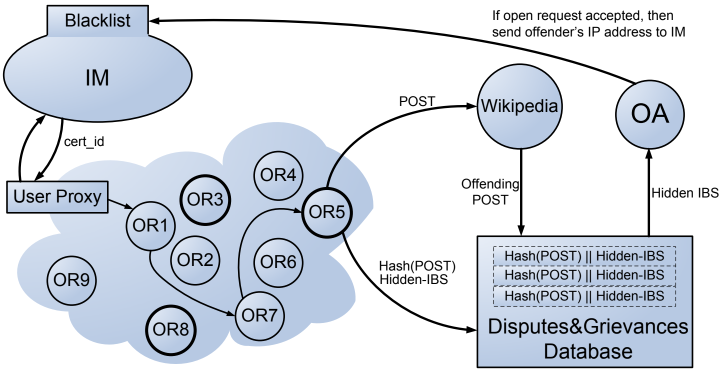

Figure 6: Enhancing the Tor network with a mechanism to defend against anonymity abuse using the Hidden-IBS primitive. Note that we use IP addresses as user identities in the figure but other types of identities can be used, e.g., userids of a reputation system.

More specifically now, the Hidden-IBS enhanced Tor works like this: certain packets generated by a Tor user are permitted through the Tor exit point only if they carry a Hidden-IBS. The Tor user's onion proxy (OP) catches this and assists the user to get the Hidden-IBS signing capability. Then any packet that needs to be signed is hashed and then signed. Tor exit points verify the Hidden-IBS signature on the hashed reconstructed packet and forward the packet (with the signature removed) to the web-site that the packet was directed while they write the hashed packet together with the signature to a Disputes&Grievances database. If any vandalism is caught by a service provider, the service-provider using the packet that was sent through Tor by the abusive user can retrieve the corresponding Hidden-IBS from the database and forward it to the OA along with a complaint report. Based on the properties of the Hidden-IBS scheme, the OA can open the signature and recover the identity of the abusive user. Subsequently the IM can be notified of the abusive user's identity and the user can be punished by being black-listed (or receiving a negative point in a reputation system). Below we describe in more details how we propose to deploy our Hidden-IBS enhanced Tor system for handling HTTP POST requests to Wikipedia. Note that all other traffic through Tor would be unaffected (i.e., it would not require a signature).

When the user first installs a Tor OP she can obtain a certificate cert_id for her identity id from the IM. The id that the user deposits to the IM can be the user's IP address or a long-lived userid in a reputation system. Subsequently whenever the user wants to send an HTTP POST the OP builds a route to a Tor exit point (in the figure, this route is OR1,OR7,OR5, and OR5 is the Tor exit point). When the user generates a POST request for a Wikipedia web-site the following things happen: (*i*) the user's browser passes the POST request, say post1 to the OP; (*ii*) the OP sanitizes post1 into post2 so that the header of post2 does not contain any unnecessary identity related information; (*iii*) the OP generates a random nonce and stored in a `Nonce` field into the header of post2, resulting to packet post3; (*iv*) the OP hashes post3 and signs the hash with the Hidden-IBS signing algorithm; (*v*) the OP creates a new field called `Signature` in the header of post3 and fills it with the generated signature; we call the modified post3 as post4; (*vi*) the OP forwards the post4 along the established circuit.

When a Tor exit point assembles a POST request such as post4 above, it parses the field `Signature` and obtains the Hidden-IBS signature; then it transforms post4 into post3 by throwing away the `Signature` field in the header and computes the hash value of post3 to verify the signature (using the public-key of the IM). Finally, if the signature verifies, the exit point forwards post3 to the Wikipedia web-site; at the same time it submits the hash value and the Hidden-IBS signature to the Disputes&Grievances database.

Wikipedia may now keep the POST request coming through a Tor exit point (or in fact only the hash of the request suffices). If a certain posting is found to be offensive or abusive the web-site may search for the corresponding Hidden-IBS signature into the Disputes&Grieances database (that will be indexed based on the hash of the post). Then, once the hidden-IBS is recovered it can be submitted to the opening authority (OA) along with a complaint report. The OA uses his secret key to open the Hidden-IBS and recover offender's identity (e.g., her IP address), and then sends this identity to the IM. The IM may blacklist this identity which may result in refusing future registration requests originating from the offender's IP address for example. Other strategies may be followed here by the IM, for example if the identity is a userid in a reputation system the user may receive a negative point.

Remark 4.1. Regarding the Dispute& Grievances database we make the following two observations: First, the database leaks no information about the identities of Tor users or the traffic they produce; indeed, only the hashed POST requests are stored together with the Hidden-IBS signatures that cryptographically hide the user identities. Second, the database size is quite manageable: indeed, using our construction from Section 3 and a 256-bit hash (e.g., SHA-256) we can store about 1.6 million pairs of hash and signature in 1Gbyte of storage. Given that only a small percentage of Tor traffic needs to be logged into the database (e.g., only POST requests) the size of the database is manageable by today's standards (e.g., with 100GB one can keep 160 million POST requests which is sufficient to maintain a database with long history).

Remark 4.2. Our solution is designed to be totally transparent to the service providers that receive Tor traffic. This is advantageous as it demonstrates the principle that Tor can manage the quality of its traffic by itself and provide mechanisms to catch misbehaving users. Still, service providers that are interested in allowing Tor traffic may sponsor the enhancement by providing storage for the Dispute& Grievances database for example.

Remark 4.3. The OA can be designed to be a distributed threshold entity where a voting decision-making procedure would be required to open the signature. In fact, the shareholders of the OA can be the population of all Tor users (with an appropriately low threshold) so in this way it would be possible for the users themselves to manage the anonymity revocation offered by the system they use. This would require a threshold variant of the encryption mechanism employed in our construction which is straightforward to build using a similar approach as [DF89, GJKR99].

Remark 4.4. In this section we used HTTP POST requests and Wikipedia as the motivating example. However it is straightforward to apply our Hidden-IBS enhancement to other types of Tor traffic or web-services. For example, we can require SMTP traffic to be signed and thus let it pass through Tor (while now it is blocked by the current Tor exit policy). Similarly Wikipedia is only one case of a web-site that faces the potential of vandalism through Tor; many other examples exist, e.g., Slashdot and they would benefit from the proposed architecture.

5 Hidden-IBS with Exculpability: Modelling

In our basic model for Hidden-IBS in Section 2, we assume the IM is honest. In that model it is evident that the IM has the capability to impersonate a user if it wishes but it is trusted not to do so. A similar problem can be observed in the primitive of group signatures where the relevant security property is called "exculpability" [ACJT00]. In a group signature scheme with exculpability the group manager is incapable of impersonating an existing user.

In this section we consider the exculpability property from the point of view of the Hidden-IBS primitive. We stress that this security property is not as important as in the case of group signatures since in Hidden-IBS there does not exist a public membership list and the overall identification performed by a Hidden-IBS is intended to be more “lightweight” compared to a group signature. Still there can be settings where it should be possible for a user to be able to deny an allegation that she is responsible for a signature and be able to prove instead that the IM tried to frame her.

To achieve the exculpability property, intuitively, based on the reasoning above, we should let the user have a secret associated to her signing capability which is not known by the IM. During user registration the user will submit a key corresponding to the secret that she only knows and the IM will embed the user’s key into the certificate he returns to the user. Subsequently, in order to issue a signature the user will have to employ her secret. On the other hand based on the hiding property of the commitment the IM will not be able to impersonate the user unless he produces another key to bind it to the user’s identity. When the OA opens a signature and accuses the user, the OA will also recover the key that was used as well. Thus the user can deny her involvement by presenting the key she used originally in her interaction with the IM. Upon the presentation of such evidence the OA will in turn accuse the IM instead.

In the remaining of the section, we modify our basic Hidden-IBS model to capture the new property.

Syntax. In a *Hidden-IBS scheme with exculpability*, the involved parties are same as that in the basic Hidden-IBS. Here the identity $\mathtt{id}$ has more complex structure though and consists of two parts, $\mathtt{name}$ and $\mathtt{key}$, where $\mathtt{name}$ can be the “name” part of the identity e.g., an IP address, or email address and $\mathtt{key}$ is the key that corresponds to a user’s secret $\mathtt{trapdoor}$. Given a certain identity $\mathtt{id}$ we will write $\mathtt{id.name}$ and $\mathtt{id.key}$ to refer to its two components respectively.

The $\mathtt{Setup}$ includes $\mathtt{SetupUser}$ in addition to $\mathtt{SetupIM}$, $\mathtt{SetupOA}$ which are defined as in the Hidden-IBS. Next we describe $\mathtt{SetupUser}$. On input a public parameter, the probabilistic algorithm outputs $\mathtt{key}$ and the corresponding user secret $\mathtt{trapdoor}$ based on some known relation. The $\mathtt{Reg}$ and $\mathtt{RegCheck}$ are same as those in the Hidden-IBS. Note that the involved identity is a pair of the form $\mathtt{id} = (\mathtt{name}, \mathtt{key})$. The $\mathtt{Sign}$ procedure will also involve the secret $\mathtt{trapdoor}$ data. We write $\mathtt{Sign}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}, \mathtt{id}; \mathtt{cert}_{\mathtt{id}}, \mathtt{trapdoor}, m)$ to denote the application of the signing algorithm, where $\mathtt{id} = (\mathtt{name}, \mathtt{key})$. The $\mathtt{Verify}$ and $\mathtt{Open}$ are syntactically the same as that in Hidden-IBS.

Correctness. The correctness of Hidden-IBS with exculpability includes the registration correctness, the signing correctness and the opening correctness: these properties are essentially identical to the properties as described in section Section 2.2 and are omitted. Note that in a Hidden-IBS scheme with exculpability a dispute resolution mechanism is available to the OA that is given two identities $\mathtt{id}, \mathtt{id}^*$, the first coming from the signature that the OA opens and the second coming from the user (cf. Figure 7). The OA proclaims the IM to be guilty or the user to be guilty depending on the output of the function $\mathtt{DisputeResult}$ that is defined in the same way for all schemes, see Figure 7. Note that if the user refuses to participate to the dispute resolution mechanism she should be considered guilty. On the other hand, the IM should not allow users with the same name register different keys.

Security. The misidentification-forgery and (CCA2,CPA)-anonymity notions are same as that in the basic Hidden-IBS (with the understanding that the identities are assumed to have the structure $\mathtt{id} = (\mathtt{name}, \mathtt{key})$). In addition to these two security properties we define a new security notion, exculpability, that captures the setting where the IM is corrupted.

In an exculpability attack the adversary is allowed to corrupt the IM and is capable of issuing signing credentials to honest users. The adversary is also capable of corrupting the OA (in the sense of obtaining the key - not in the sense of controlling the outcome of who is guilty). The adversary is allowed to adaptively ask signing queries from the honest users. The adversary wins the game if it manages to produce a signature that (i) opens to one of the honest users and (ii) the $\mathtt{DisputeResult}$ algorithm proclaims the user to be guilty despite the fact that the honest user participates in the dispute resolution procedure by providing her $\mathtt{id.key}$.

Definition 5.1 (Exculpability). We say a Hidden-IBS scheme is secure against exculpability attacks if for

```
$\mathsf{CreateOracle}(\mathtt{name})$
    If $\mathtt{name} \in HU$ then return $\bot$;
    $(\mathtt{key}, \mathtt{trapdoor}) \leftarrow \mathtt{SetupUser}(1^\lambda, pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; \mathtt{name})$;
    Save $\mathtt{id} = \langle \mathtt{name}, \mathtt{key} \rangle$;
    Return $\mathtt{key}$;

$\mathsf{SignOracle}(\mathtt{id}, \mathtt{cert}_{\mathtt{id}}, m)$
    If $\mathtt{id}.\mathtt{name} \notin HU$ then return $\bot$;
    $\sigma \leftarrow \mathtt{Sign}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}, \mathtt{id}; \mathtt{cert}_{\mathtt{id}}, m)$;
    $MSG_{\mathtt{id}} \leftarrow MSG_{\mathtt{id}} \cup \{m\}$;
    Return $\sigma$;

$\mathsf{CreateCheckOracle}(\mathtt{cert}_{\mathtt{id}})$
    If $\mathtt{RegCheck}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; \mathtt{id}, \mathtt{cert}_{\mathtt{id}}) \neq 1$ then return $\bot$;
    $HU \leftarrow HU \cup \{\mathtt{id}\}$
    Return 1;

$\mathsf{CorruptOAOracle}()$
    Return $sk_{\mathsf{OA}}$;

$\mathsf{CorruptIMOracle}()$
    Return $sk_{\mathsf{IM}}$;

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{exculp}}(\lambda)$
    $(pk_{\mathsf{IM}}, sk_{\mathsf{IM}}) \leftarrow \mathtt{SetupIM}(1^\lambda)$; $(pk_{\mathsf{OA}}, sk_{\mathsf{OA}}) \leftarrow \mathtt{SetupOA}(1^\lambda)$; $HU \leftarrow \emptyset$; $CU \leftarrow \emptyset$
    $(m, \sigma) \leftarrow \mathcal{A}^{\mathsf{CreateOracle}(),\mathsf{CreateCheckOracle}(),\mathsf{SignOracle}(),\mathsf{CorruptIMOracle}(),\mathsf{CorruptOAOracle}()}(1^\lambda, pk_{\mathsf{IM}}, pk_{\mathsf{OA}})$
    If $\mathtt{Verify}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; m, \sigma) = 1$
        $\wedge$ $\mathtt{Open}(pk_{\mathsf{IM}}, pk_{\mathsf{OA}}; sk_{\mathsf{OA}}, m, \sigma) = \mathtt{id}$ $\wedge$ $m \notin MSG_{\mathtt{id}}$ $\wedge$ $\mathtt{id}.\mathtt{name} \in HU$
        $\wedge$ $\mathtt{id}^* \in HU$ s.t. $\mathtt{id}.\mathtt{name} = \mathtt{id}^*.\mathtt{name}$
        $\wedge$ $\mathtt{DisputeResult}(\mathtt{id}, \mathtt{id}^*) =$"User is guilty" then return 1;
    Return 0;
```

The dispute resolution is defined as follows:

$$\mathtt{DisputeResult}(\mathtt{id}, \mathtt{id}^*) = \begin{cases} \text{“User is guilty”} & \mathtt{id}.\mathtt{key} = \mathtt{id}^*.\mathtt{key} \\ \text{“IM is guilty”} & \mathtt{id}.\mathtt{key} \neq \mathtt{id}^*.\mathtt{key} \end{cases}$$

Figure 7: Experiment of an exculpability attack

any PPT adversary $\mathcal{A}$, $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{exculp}}(\lambda)$ is negligible in λ, where $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{exculp}}(\lambda) = \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{exculp}}(\lambda) = 1]$, and the involved experiment defined as in Figure 7.

6 Hidden-IBS with Exculpability: Construction

Now we present an efficient construction which can achieve misidentification-forgery, CCA2-anonymity and also exculpability notions. In the user registration, the IM use CL signature [CL02b] to generate a certificate to user's identity. Note that the user's identity includes two components, name and key. The key is a RSA modulus, and only the user knows the trapdoor of the key which is two primes. When the user signs a message, she encrypts her identity by the CCA2-Paillier encryption [CS03] and sends the ciphertext to the verifier which can be used for the opening. Then, she proves to the verifier that she knows her identity, the certificate, and the relation between them. Furthermore, she proves that she knows the trapdoor which is corresponding to the key in the identity. We present the details below:

$\mathtt{Setup}$. The algorithm $\mathtt{SetupIM}$ generates key pair $(pk_{\mathsf{IM}}, sk_{\mathsf{IM}})$: first generates parameter for cyclic group $QR(n)$, i.e. $\langle n, p, q, p', q' \rangle$, where $n = pq$, $p = 2p' + 1$, $q = 2q' + 1$, p, q, p', q' are primes; randomly selects $a_0, a, b \overset{r}{\leftarrow} QR(n)$; sets $pk_{\mathsf{IM}} = \langle n, a_0, a, b \rangle$, and $sk_{\mathsf{IM}} = \langle p, q \rangle$. The algorithm $\mathtt{SetupOA}$ generates key pair $(pk_{\mathsf{OA}}, sk_{\mathsf{OA}})$: generate the Paillier encryption parameter $\langle N, G, P, Q, P', Q' \rangle$ where $N = PQ$, $P = 2P' + 1$, $Q = 2Q' + 1$; select $H_1, H_2, H_3 \in \langle G \rangle$ with $H_i = G^{\alpha_i}$, $\alpha_i \overset{r}{\leftarrow} \mathbb{Z}_{\lfloor N/4 \rfloor}$ for $i = 1, 2, 3$, and a hash-key hk for a universal one-way hash function family hash; sets $pk_{\mathsf{OA}} = \langle N, G, H_1, H_2, H_3, \mathsf{hk}, \mathsf{hash} \rangle$ and $sk_{\mathsf{OA}} = \langle \alpha_1, \alpha_2, \alpha_3 \rangle$. More system parameters are required: randomly selects $g, f_1, f_2, f_3, f_4, f_5, f_6 \overset{r}{\leftarrow} QR(n)$ for commitment; generates a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^{\lambda_0}$, which will be treated as a random oracle in the security proof. Finally set the public parameters for the Hidden-IBS as $\mathtt{pub} = \langle n, a_0, a, b; N, G, H_1, H_2, H_3, \mathsf{hash}, \mathsf{hk}; g, f_1, f_2, f_3, f_4, f_5, f_6, \mathcal{H} \rangle$. For each user, her identity $\mathtt{id}$ consists of two parts, name and key, i.e. $\mathtt{id} = (\mathtt{name}, \mathtt{key})$. The length of $\mathtt{key}$ is ℓ and the length of $\mathtt{name}$ is ℓ'. The algorithm $\texttt{SetupUser}$ generates key pair $(\texttt{key}, \texttt{trapdoor})$ for each user: generates RSA modulus $x = x_1x_2$ where x_1, x_2 are two primes with length $\ell/2$; sets $\texttt{key} = x$ and $\texttt{trapdoor} = \langle x_1, x_2\rangle$.

$\texttt{Reg}$. The user keep the trapdoor $\langle x_1, x_2\rangle$; sends the key x with her name $\texttt{name}$ to the IM. The IM verifies that $\texttt{name}$ has never been submitted before. Then the IM generates a CL signature $\langle v, e, s\rangle$ for identity $\texttt{id} = x + \texttt{name} \cdot 2^{\ell}$, where $v^e = a_0a^{x+\texttt{name}\cdot 2^{\ell}}b^s$ in $QR(n)$; and sends $\langle v, e, s\rangle$ to the user by a secure communication channel.

$\texttt{RegCheck}$. Once receiving the signature $\langle v, e, s\rangle$ from the IM, the user verifies $v^e = a_0a^{x+\texttt{name}\cdot 2^{\ell}}b^s$ in $QR(n)$. The user sets her membership certificate to $\texttt{cert}_{\texttt{id}} = \langle v, e, s\rangle$.

$\texttt{Sign}$. With a membership certificate $\texttt{cert}_{\texttt{id}} = \langle v, e, s\rangle$ in hand, a user can compute a Hidden-IBS signature σ for message m. We design the signing algorithm by applying the Fiat-Shamir heuristic on an proof of knowledge which is shown in Figure 8 in the appendix. In the proof of knowledge, the user proves her knowledge of the identity including the key x and the name $\texttt{name}$, and of the membership certificate on the identity from the IM. Also, the user proves her knowledge of the trapdoor $\langle x_1, x_2\rangle$ of the key x which can be used to prevent the exculpability from the IM.

Next, we give the details. We let $\ell_n, \ell_v, \ell_e, \ell_s, \ell_{\texttt{name}}, \ell_N$ denote the length of them. (i) The user uses the CCA2-Paillier encryption [CS03] to encrypt $\texttt{id} = x + \texttt{name} \cdot 2^{\ell}$ into $\langle C_1, C_2, C_3\rangle$: $d \overset{\texttt{r}}{\leftarrow} \pm\{0,1\}^{\ell_N - 2}$, $C_1 = G^d$ in $\mathbb{Z}^*_{N^2}$, $C_2 = H_1^d(1+N)^{x+\texttt{name}\cdot 2^{\ell}}$ in $\mathbb{Z}^*_{N^2}$, $C_3 = \mathrm{abs}\big(H_2H_3^{\mathsf{hash}(\mathsf{hk}, C_1, C_2)})^d\big)$ in $\mathbb{Z}^*_{N^2}$. Note that here $\mathrm{abs}(x) = x$ if $x \le N^2/2$ and $\mathrm{abs}(x) = N^2 - x$ otherwise. (ii) The user makes commitments for the membership certificate $\texttt{cert}_{\texttt{id}} = (v, e, s)$, and the key x and the corresponding trapdoor (x_1, x_2): randomly select $r_1, r_2, r_3 \overset{\texttt{r}}{\leftarrow} \pm\{0,1\}^{\ell_n - 2}$, compute $T_1 = g^{r_1}f_1^{x_1}$, $T_2 = g^{r_2}v$, and $T_3 = g^{r_3}f_1^xf_2^{x_2}f_3^{\texttt{name}}f_4^ef_5^sf_6^d$. (iii) The user deploys a three move Σ-protocol to prove her knowledge of the underlying plaintext $\texttt{id} = x + \texttt{name}\cdot 2^{\ell}$ of the Paillier ciphertext, the underlying plaintexts $v, e, s, x_1, x_2, d, x, \texttt{name}$ of the commitments, and the relation between (v, e, s) and $\texttt{id}$, the relation between x and (x_1, x_2): compute $r_4 = r_1x_2$ and $r_5 = r_2e$; randomly select $\theta_x \overset{\texttt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\mu'}$, $\theta_{x_1}, \theta_{x_2} \overset{\texttt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\mu''}$, $\theta_{\texttt{name}} \overset{\texttt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_{\texttt{name}}}$, $\theta_e \overset{\texttt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_e}$, $\theta_s \overset{\texttt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_s}$, $\theta_d \overset{\texttt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_d}$, $\theta_{r_1}, \theta_{r_2}, \theta_{r_3} \overset{\texttt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_n-2}$, $\theta_{r_4} \overset{\texttt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_n-2+\mu''}$, $\theta_{r_5} \overset{\texttt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_n-2+\ell_e}$; compute $B_1 = g^{-\theta_{r_1}}f_1^{-\theta_{x_1}}$, $B_2 = T_1^{-\theta_{x_2}}g^{\theta_{r_4}}f^{\theta_x}$, $B_3 = T_2^{-\theta_e}g^{\theta_{r_5}}a^{\theta_x+\theta_{\texttt{name}}\cdot 2^{\ell}}b^{\theta_s}$, $B_4 = g^{-\theta_{r_3}}f_1^{-\theta_x}f_2^{-\theta_{x_2}}f_3^{-\theta_{\texttt{name}}}f_4^{-\theta_e}f_5^{-\theta_s}f_6^{-\theta_d}$, $B_5 = G^{-\theta_d}$ in $\mathbb{Z}^*_{N^2}$, $B_6 = H_1^{-\theta_d}(1+N)^{-(\theta_x+\theta_{\texttt{name}}\cdot 2^{\ell})}$ in $\mathbb{Z}^*_{N^2}$, $B_7 = (H_2H_3^{\mathsf{hash}(\mathsf{hk}, C_1, C_2)})^{-2\theta_d}$ in $\mathbb{Z}^*_{N^2}$; compute $c = \mathcal{H}(m||T_1||T_2||T_3||C_1||C_2||C_3||B_1||...||B_7)$; then compute $\xi_x = \theta_x + c\cdot(x - 2^{\ell'})$, $\xi_{x_1} = \theta_{x_1} + c\cdot(x_1 - 2^{\ell''})$, $\xi_{x_2} = \theta_{x_2} + c\cdot(x_2 - 2^{\ell''})$, $\xi_e = \theta_e + c\cdot e$, $\xi_s = \theta_s + c\cdot s$, $\xi_d = \theta_d + c\cdot d$, $\xi_{r_1} = \theta_{r_1} + c\cdot r_1$, $\xi_{r_2} = \theta_{r_2} + c\cdot r_2$, $\xi_{r_3} = \theta_{r_3} + c\cdot r_3$, $\xi_{r_4} = \theta_{r_4} + c\cdot r_4$, $\xi_{r_5} = \theta_{r_5} + c\cdot r_5$. Therefore, the generated signature for message m is the tuple: $\sigma = \langle T_1, T_2, T_3, C_1, C_2, C_3; c; \xi_{\texttt{id}}, \xi_x, \xi_{x_1}, \xi_{x_2}, \xi_e, \xi_s, \xi_d, \xi_{r_1}, ..., \xi_{r_5}\rangle$.

$\texttt{Verify}$. The verifier can verify a message-signature pair by the following checks:

$$
\begin{aligned}
&\xi_x \overset{?}{\in} \pm\{0,1\}^{\lambda_0+\lambda_1+\mu'+1},\ \xi_{x_1}, \xi_{x_2} \overset{?}{\in} \pm\{0,1\}^{\lambda_0+\lambda_1+\mu''+1},\ C_1, C_2, C_3 \overset{?}{\in} \mathbb{Z}^*_{N^2},\ C_2 \overset{?}{\le} N^2/2,\\
&c \overset{?}{=} \mathcal{H}\big(m||T_1||T_2||T_3||C_1||C_2||C_3\\
&||(T_1)^cg^{-\xi_{r_1}}f_1^{-\xi_{x_1}}||T_1^{-\xi_{x_2}}g^{\xi_{r_4}}f_1^{\xi_x}||(a_0)^cT_2^{-\xi_e}g^{\xi_{r_5}}a^{(\xi_x+\xi_{\texttt{name}}\cdot 2^{\ell})}b^{\xi_s}\\
&||(T_3)^cg^{-\xi_{r_3}}f_1^{-\xi_x}f_2^{-\xi_{x_2}}f_3^{-\xi_{\texttt{name}}}f_4^{-\xi_e}f_5^{-\xi_s}f_6^{-\xi_d}\\
&||(C_1)^cG^{-\xi_d}||(C_2)^cH_1^{-\xi_d}(1+N)^{-(\xi_x+\xi_{\texttt{name}}\cdot 2^{\ell})}||(C_3)^{2c}(H_2H_3^{\mathsf{hash}(\mathsf{hk}, C_1, C_2)})^{-2\xi_d}\big)
\end{aligned}
$$

$\texttt{Open}$. Given a message-signature pair as described above, the OA first verifies the message-signature pair. Also the OA verifies the relation $C_3^2 = C_1^{2(\alpha_2+\alpha_3\mathsf{hash}(\mathsf{hk}, C_1, C_2))}$. Then the OA computes $\widehat{\texttt{id}} = C_2^2C_1^{-2\alpha_1}$, and $\texttt{id} = \big(\widehat{\texttt{id}} \cdot 2^{-1} \bmod N\big)/N$. Finally, the OA parsed $\texttt{id}$ into two parts, x and $\texttt{name}$.

Theorem 6.1. *Our Hidden-IBS is correct and secure satisfying misidentification-forgery, exculpability and CCA2-anonymity in the random oracle model under the Strong RSA, factoring and the DCR assumptions and the UOHF assumption respectively.*

References

[ACHdM05] Giuseppe Ateniese, Jan Camenisch, Susan Hohenberger, and Breno de Medeiros. Practical group signatures without random oracles. In *Cryptology ePrint Archive, Report 2005/385*, 2005. http://eprint.iacr.org/2005/385/.

[ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2000.

[AdM03] Giuseppe Ateniese and Breno de Medeiros. Efficient group signatures without trapdoors. In Chi-Sung Laih, editor, *ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 246–268. Springer, 2003.

[AST02] Giuseppe Ateniese, Dawn Xiaodong Song, and Gene Tsudik. Quasi-efficient revocation in group signatures. In Matt Blaze, editor, *Financial Cryptography 2002*, volume 2357 of *Lecture Notes in Computer Science*, pages 183–197. Springer, 2002.

[AT99] Giuseppe Ateniese and Gene Tsudik. Some open issues and new directions in group signatures. In Matthew K. Franklin, editor, *Financial Cryptography 1999*, volume 1648 of *Lecture Notes in Computer Science*, pages 196–211. Springer, 1999.

[BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2004.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. *J. Cryptology*, 17(4):297–319, 2004.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer, 2003.

[BNN04] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security proofs for identity-based identification and signature schemes. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 268–286. Springer, 2004. Full version at http://www-cse.ucsd.edu/users/mihir/papers/ibi.pdf.

[BS04] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors, *CCS 2004*, pages 168–177. ACM, 2004.

[BSZ05] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 136–153. Springer, 2005.

[BW06] Xavier Boyen and Brent Waters. Compact group signatures without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 427–444. Springer, 2006.

[Cam97] Jan Camenisch. Efficient and generalized group signatures. In *EUROCRYPT 1997*, pages 465–479, 1997.

[CG04] Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In Carlo Blundo and Stelvio Cimato, editors, *SCN 2004*, volume 3352 of *Lecture Notes in Computer Science*, pages 120–133. Springer, 2004.

[CL01] Jan Camenisch and Anna Lysyanskaya. An identity escrow scheme with appointed verifiers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 388–407. Springer, 2001.

[CL02a] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2002.

[CL02b] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 2002*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289. Springer, 2002.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72. Springer, 2004.

[CM98] Jan Camenisch and Markus Michels. A group signature scheme with improved efficiency. In Kazuo Ohta and Dingyi Pei, editors, *ASIACRYPT 1998*, volume 1514 of *Lecture Notes in Computer Science*, pages 160–174. Springer, 1998.

[CM99] Jan Camenisch and Markus Michels. Separability and efficiency for generic group signature schemes. In Michael J. Wiener, editor, *CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 413–430. Springer, 1999.

[CMS96] Jan Camenisch, Ueli M. Maurer, and Markus Stadler. Digital payment systems with passive anonymity-revoking trustees. In Elisa Bertino, Helmut Kurth, Giancarlo Martella, and Emilio Montolivo, editors, *ESORICS 1996*, volume 1146 of *Lecture Notes in Computer Science*, pages 33–43. Springer, 1996.

[CP94] Lidong Chen and Torben P. Pedersen. New group signature schemes (extended abstract). In *EUROCRYPT 1994*, pages 171–181, 1994.

[CS97] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (extended abstract). In Burton S. Kaliski Jr., editor, *CRYPTO 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 1997.

[CS03] Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144. Springer, 2003. Full version available at `http://shoup.net/papers/verenc.pdf`.

[CvH91] David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT 1991*, pages 257–265, 1991.

[DF89] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In Gilles Brassard, editor, *CRYPTO 1989*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315. Springer, 1989.

[DMS04] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. Tor: The second-generation onion router. In *USENIX Security Symposium 2004*, pages 303–320. USENIX, 2004.

[FI05] Jun Furukawa and Hideki Imai. An efficient group signature scheme from bilinear maps. In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP 2005*, volume 3574 of *Lecture Notes in Computer Science*, pages 455–467. Springer, 2005.

[FTY96] Yair Frankel, Yiannis Tsiounis, and Moti Yung. "Indirect Discourse Proof": achieving efficient fair off-line e-cash. In Kwangjo Kim and Tsutomu Matsumoto, editors, *ASIACRYPT 1996*, volume 1163 of *Lecture Notes in Computer Science*, pages 286–300. Springer, 1996.

[FY04] Jun Furukawa and Shoko Yonezawa. Group signatures with separate and distributed authorities. In Carlo Blundo and Stelvio Cimato, editors, *SCN 2004*, volume 3352 of *Lecture Notes in Computer Science*, pages 77–90. Springer, 2004.

[GJKR99] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. In *EUROCRYPT 1999*, pages 295–310, 1999.

[GRS96] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In Ross J. Anderson, editor, *Information Hiding 1996*, volume 1174 of *Lecture Notes in Computer Science*, pages 137–150. Springer, 1996.

[Hel61] Joseph L. Heller. *Catch-22*. Simon & Schuster, 1961.

[KP98] Joe Kilian and Erez Petrank. Identity escrow. In Hugo Krawczyk, editor, *CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 169–185. Springer, 1998.

[KTY04] Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Traceable signatures. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 571–589. Springer, 2004.

[KY03] Aggelos Kiayias and Moti Yung. Extracting group signatures from traitor tracing schemes. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 630–648. Springer, 2003.

[KY05a] Aggelos Kiayias and Moti Yung. Efficient secure group signatures with dynamic joins and keeping anonymity against group managers. In Ed Dawson and Serge Vaudenay, editors, *Mycrypt 2005*, volume 3715 of *Lecture Notes in Computer Science*, pages 151–170. Springer, 2005. Full version at http://eprint.iacr.org/2004/076/.

[KY05b] Aggelos Kiayias and Moti Yung. Group signatures with efficient concurrent join. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 198–214. Springer, 2005. Full version at http://eprint.iacr.org/2005/345/.

[NSN04] Lan Nguyen and Reihaneh Safavi-Naini. Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 372–386. Springer, 2004.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.

[Pol75] J.M. Pollard. A Monte Carlo method for factorization. *BIT*, 15:331–334, 1975.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO 1984*, pages 47–53, 1984.

[Son01] Dawn Xiaodong Song. Practical forward secure group signature schemes. In *CCS 2001*, pages 225–234, 2001.

[Tor] Tor. http://tor.eff.org/.

[TX03] Gene Tsudik and Shouhuai Xu. Accumulating composites and improved group signing. In Chi-Sung Laih, editor, *ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 269–286. Springer, 2003.

[vSN92] Sebastiaan H. von Solms and David Naccache. On blind signatures and perfect crimes. *Computers & Security*, 11(6):581–583, 1992.

[Wik] Wikipedia. http://wikipedia.org/.

[Wik06] Wikipedia. Advice to Tor users in China. May 2006. http://en.wikipedia.org/wiki/Wikipedia:Tor.

[WYZ05] Victor K. Wei, Tsz Hon Yuen, and Fangguo Zhang. Group signature where group manager, members and open authority are identity-based. In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP 2005*, volume 3574 of *Lecture Notes in Computer Science*, pages 468–480. Springer, 2005.

A Appendix

A.1 Preliminaries

Bilinear Groups. Let $\mathbb{G} = \langle g \rangle$ and $\widehat{\mathbb{G}} = \langle \widehat{g} \rangle$ be cyclic groups of prime order p where $\psi : \widehat{\mathbb{G}} \to \mathbb{G}$ is an isomorphism with $\psi(\widehat{g}) = g$ and $e : \mathbb{G} \times \widehat{\mathbb{G}} \to \mathbb{G}_T$ is a bilinear map, i.e., for all $(u, \widehat{u}) \in \mathbb{G} \times \widehat{\mathbb{G}}$ and $a, b \in \mathbb{Z}$, it holds that $e(u^a, \widehat{u}^b) = e(u, \widehat{u})^{ab}$ and e is non-trivial, i.e., $e(g, \widehat{g}) \neq 1$. Note that $|\mathbb{G}_T| = p$.

Linear Encryption. Boneh et al. [BBS04] proposed a variant of ElGamal encryption, called, Linear Encryption that is suitable for groups over which the DDH assumption fails. We call it LE for short.
Key Generation. The public key is a triple of generators $u, v, w \in \mathbb{G}$ and the secret key is the exponents $x, y \in \mathbb{Z}_p^*$ such that $u^x = v^y = w$.
Encryption. On input of a message $m \in \mathbb{G}$, choose random values $a, b \in \mathbb{Z}_p$, and output the triple $(u^a, v^b, w^{a+b}m)$.
Encryption. Given a ciphertext $\langle U, V, W \rangle$, by using the secret key x, y, we recover the plaintext m as follows $m = \frac{W}{U^x \cdot V^y}$.

The Linear encryption is based on the Decision Linear Diffie-Hellman assumption, which was first introduced by Boneh et al. [BBS04]. With $g \in \mathbb{G}$ as above, along with arbitrary generators u,v, and w of $\mathbb{G}$, consider the following problem:

Definition A.1 (Decision Linear Diffie-Hellman Problem in $\mathbb{G}$). Given $u, v, w, u^\alpha, v^\beta, w^\gamma \in \mathbb{G}$ as input, output 1 if $\alpha + \beta = \gamma$ and 0 otherwise.

It is believed that DLDH is a hard problem even in bilinear groups where DDH is easy. Now we define the advantage of an algorithm $\mathcal{A}$ in deciding the DLDH problem in $\mathbb{G}$ as

$$\mathsf{Adv}^{\mathcal{A}}_{\mathsf{DLDH}} = \left| \begin{array}{l} \Pr[1 \leftarrow \mathcal{A}(u, v, w, u^\alpha, v^\beta, w^{\alpha+\beta}) : u, v, w \in \mathbb{G}, \alpha, \beta \in \mathbb{Z}_p] \\ \quad - \Pr[1 \leftarrow \mathcal{A}(u, v, w, u^\alpha, v^\beta, \chi) : u, v, w, \chi, \in \mathbb{G}, \alpha, \beta \in \mathbb{Z}_p] \end{array} \right|$$

Assumption A.2 (Decision Linear Diffie-Hellman Assumption). We say that the Decision Linear Diffie-Hellman assumption holds in $\mathbb{G}$ if for all PPT algorithms $\mathcal{A}$ it holds that $\mathsf{Adv}^{\mathcal{A}}_{\mathsf{DLDH}}$ is negligible in the security parameter λ.

Boneh-Boyen Signature. Boneh and Boyen [BB04] propose a very efficient signature scheme secure in the standard model under the Strong Diffie-Hellman (SDH) assumption.
Key Generation. Let $\langle p, g, \widehat{g}, \mathbb{G}, \widehat{\mathbb{G}}, \psi, \mathbb{G}_T, e \rangle$ be bilinear groups parameter. Randomly select $x, y \xleftarrow{\mathtt{r}} \mathbb{Z}_p^*$ and compute $\widehat{X} \leftarrow \widehat{g}^x \in \widehat{\mathbb{G}}$ and $\widehat{Y} \leftarrow \widehat{g}^y \in \widehat{\mathbb{G}}$. Set public key $\langle p, g, \widehat{g}, \mathbb{G}, \widehat{\mathbb{G}}, \psi, \mathbb{G}_T, e; \widehat{X}, \widehat{Y} \rangle$ and secret key $\langle x, y \rangle$.
Signature Generation. On input $m \in \mathbb{Z}_p^*$, randomly selects $r \xleftarrow{\mathtt{r}} \mathbb{Z}_p^*$ such that $x + m + yr \not\equiv 0 \bmod p$; and compute $s = g^{\frac{1}{x+m+yr}} \in \mathbb{G}$. The signature for m is (s, r).
Signature Verification. Given public key $\langle p, g, \widehat{g}, \mathbb{G}, \widehat{\mathbb{G}}, \psi, \mathbb{G}_T, e; \widehat{X}, \widehat{Y} \rangle$, message m, and signature (s, r), check that $m, r \in \mathbb{Z}_p^*$, and $e(s, \widehat{X}\widehat{g}^m\widehat{Y}^r) = e(g, \widehat{g})$. If they hold, the verification is valid; otherwise invalid.

Definition A.3 (Strong Diffie-Hellman Assumption.). Given bilinear group parameter $\langle p, g, \widehat{g}, \mathbb{G}, \widehat{\mathbb{G}}, \psi, \mathbb{G}_T, e \rangle$, the q-SDH problem is defined as follows: given a $(q + 1)$-tuple $(g, \widehat{g}, \widehat{g}^x, \widehat{g}^{x^2}, \widehat{g}^{x^3}, ..., \widehat{g}^{x^q})$ as input, output a pair $(g^{\frac{1}{x+c}}, c)$ where $c \in \mathbb{Z}_p^*$. The q-SDH assumption suggests that any PPT algorithm solving the q-SDH problem has negligible success probability in secure parameter λ.

Paillier-Encryption. Paillier proposed a very efficient homomorphic encryption [Pai99]:
Key Generation. Let P and Q be random primes for which it holds $P \neq Q$, $|P| = |Q|$ and $\gcd(PQ, (P - 1)(Q - 1)) = 1$; let $N = PQ$, $\pi = \mathrm{lcm}(P - 1, Q - 1)$, $K = \pi^{-1} \bmod N$, and $G = (1 + N)$; the public key is $\langle N, G \rangle$ while the secret key is $\langle P, Q \rangle$.
Encryption. The plaintext set is $\mathbb{Z}_N$; given a plaintext m, choose a random $\zeta \in \mathbb{Z}_N^*$, and let the ciphertext be $C = G^m \zeta^N \bmod N^2$.
Decryption. Given a ciphertext C, let $K = \pi^{-1} \bmod N$ and now observe that $C^{\pi K} = G^{m \cdot \pi K} \cdot \zeta^{N \cdot \pi K} = G^{m \cdot \pi K \bmod N} \cdot \zeta^{N \cdot \pi K \bmod N\pi} = G^{m \bmod N} \cdot \zeta^{0 \bmod N\pi} = G^m = 1 + mN \bmod N^2$. Thus, it is possible to recover $m = \frac{(C^{\pi K} \bmod N^2) - 1}{N} \bmod N$.

The cryptosystem above has been proven semantically secure if and only if the Decisional Composite Residuosity (DCR) assumption [Pai99] is true. The advantage of an algorithm $\mathcal{A}$ in deciding the DCR problem is defined as follows:

$$\mathsf{Adv}^{\mathcal{A}}_{\mathsf{DCR}} = \left| \; \Pr[1 \leftarrow \mathcal{A}(z) : z \in \mathbb{Z}_{N^2}^*] - \Pr[1 \leftarrow \mathcal{A}(z) : z \in HR_{N^2}^N] \; \right|$$

where $HR_{N^2}^N$ is the subgroup of N-th residues modulo N^2.

Assumption A.4 (Decisional Composite Residuosity Assumption). We say that the DCR assumption holds in $\mathbb{G}$ if for all PPT algorithms $\mathcal{A}$ it holds that $\mathsf{Adv}_{\mathsf{DCR}}^{\mathcal{A}}$ is negligible in the security parameter λ.

Camenisch-Lysyanskaya Signature. Camenisch and Lysyanskaya [CL02a] proposes an efficient and multi-functional signature scheme that is EU-CMA under the Strong RSA assumption. Here we give a brief description of the scheme and the underlying assumption.

Key Generation. On input 1^λ, choose an RSA modulus $n = pq$, $p = 2p' + 1$, $q = 2q' + 1$ as a product of safe primes. Randomly choose $a_0, a, b \in QR(n)$. The public key is $\langle n, a_0, a, b\rangle$ and the secret key is $\langle p, q\rangle$.
Signature Generation. On input m, choose a random number e of length $\ell_e > \ell_m + 2$, and a random number s of length $\ell_s = \ell_n + \ell_m + \lambda$, where λ is a security parameter. Compute v such that $v^e = a_0 a^m b^s \bmod n$. The signature on message m consists of $\langle v, e, s\rangle$.
Signature Verification. Given public key $\langle n, a_0, a, b\rangle$ and message m, and signature $\langle v, e, s\rangle$, check that $v^e = a_0 a^m b^s \bmod n$, and check that $2^{\ell_e} > e > 2^{\ell_e - 1}$.

Definition A.5 (Strong RSA Assumption). Given a RSA composite n, and $x \in QR(n)$, it is infeasible to find $y \in \mathbb{Z}_n^*$ and $e > 1$ such that $y^e \equiv x \bmod n$ in time polynomial in parameter λ.

A.2 Proofs

A.2.1 Proof of Theorem 3.1

Proof. To prove the registration correctness, we need to verify a BB signature $\langle s = g^{\frac{1}{x+\mathtt{id}+yr}}, r\rangle$ based on the public key $\langle \widehat{X} = \widehat{g}^x, \widehat{Y} = \widehat{g}^y\rangle$:

$$e(s, \widehat{X}\widehat{g}^{\mathtt{id}}\widehat{Y}^r) = e(g^{\frac{1}{x+\mathtt{id}+yr}}, \widehat{g}^x\widehat{g}^{\mathtt{id}}(\widehat{g}^y)^r) = e(g^{\frac{1}{x+\mathtt{id}+yr}}, \widehat{g}^{x+\mathtt{id}+yr}) = e(g, \widehat{g})$$

Based on the registration correctness, next we prove the signing correctness. On input (m, σ) where $\sigma = \langle S, \widehat{R}, U, V, \widehat{W}; c; \xi_{\mathtt{id}}, \xi_r, \xi_{r_1}, \xi_{r_2}, \xi_k, \xi_l, \xi_{\delta_1}, ..., \xi_{\delta_5}\rangle$, we need to verify

$$\begin{aligned}
&c = \mathcal{H}\big(m||S||\widehat{R}||U||V||\widehat{W}\\
&||U^c u^{-\xi_k}||V^c v^{-\xi_l}||\widehat{W}^c\widehat{w}^{-(\xi_k+\xi_l)}\widehat{g}^{-\xi_{\mathtt{id}}}||\widehat{R}^c\widehat{g}^{-\xi_{r_2}}\widehat{h}^{-\xi_{r_1}}\widehat{Y}^{-\xi_r}||U^{-\xi_{r_1}}u^{\xi_{\delta_1}}||V^{-\xi_{r_1}}v^{\xi_{\delta_2}}||\widehat{R}^{-\xi_{r_1}}\widehat{g}^{\xi_{\delta_3}}\widehat{h}^{\xi_{\delta_4}}\widehat{Y}^{\xi_{\delta_5}}\\
&||e(g, \widehat{X}\widehat{W}\widehat{R})^{\xi_{r_1}}e(S, \widehat{w})^{(\xi_k+\xi_l)}e(g, \widehat{w})^{-(\xi_{\delta_1}+\xi_{\delta_1})}e(S, \widehat{g})^{\xi_{r_2}}e(g, \widehat{g})^{-\xi_{\delta_3}}e(S, \widehat{h})^{\xi_{r_1}}e(g, \widehat{h})^{-\xi_{\delta_4}}(e(g, \widehat{g})/e(S, \widehat{X}\widehat{W}\widehat{R}))^c\big)
\end{aligned}$$

In fact we just need to verify the equations: $B_1 = U^c u^{-\xi_k}$, $B_2 = V^c v^{-\xi_l}$, $B_3 = \widehat{W}^c\widehat{w}^{-(\xi_k+\xi_l)}$, $B_4 = \widehat{R}^c\widehat{g}^{-\xi_{r_2}}\widehat{h}^{-\xi_{r_1}}\widehat{Y}^{-\xi_r}$, $B_5 = U^{-\xi_{r_1}}u^{\xi_{\delta_1}}$, $B_6 = V^{-\xi_{r_1}}v^{\xi_{\delta_2}}$, $B_7 = \widehat{R}^{-\xi_{r_1}}\widehat{g}^{\xi_{\delta_3}}\widehat{h}^{\xi_{\delta_4}}\widehat{Y}^{\xi_{\delta_5}}$, $B_8 = e(g, \widehat{X}\widehat{W}\widehat{R})^{\xi_{r_1}}$ $e(S, \widehat{w})^{(\xi_k+\xi_l)}e(g, \widehat{w})^{-(\xi_{\delta_1}+\xi_{\delta_1})}e(S, \widehat{g})^{\xi_{r_2}}e(g, \widehat{g})^{-\xi_{\delta_3}}e(S, \widehat{h})^{\xi_{r_1}}e(g, \widehat{h})^{-\xi_{\delta_4}}(e(g, \widehat{g})/e(S, \widehat{X}\widehat{W}\widehat{R}))^c$. We put them in details:

$$\begin{aligned}
B_1 &= u^{-\theta_k} = u^{ck-\xi_k} = (u^k)^c u^{-\xi_k} = U^c u^{-\xi_k}\\
B_2 &= v^{-\theta_l} = v^{cl-\xi_l} = (v^l)^c v^{-\xi_l} = V^c v^{-\xi_l}\\
B_3 &= \widehat{w}^{-(\theta_k+\theta_l)}\widehat{g}^{-\theta_{\mathtt{id}}} = \widehat{w}^{ck-\xi_k+cl-\xi_l}\widehat{g}^{c\mathtt{id}-\xi_{\mathtt{id}}} = (\widehat{w}^{k+l}\widehat{g}^{\mathtt{id}})^c\widehat{w}^{-(\xi_k+\xi_l)}\widehat{g}^{-\xi_{\mathtt{id}}} = \widehat{W}^c\widehat{w}^{-(\xi_k+\xi_l)}\widehat{g}^{-\xi_{\mathtt{id}}},\\
B_4 &= \widehat{g}^{-\theta_{r_2}}\widehat{h}^{-\theta_{r_1}}\widehat{Y}^{-\theta_r} = \widehat{g}^{cr_2-\xi_{r_2}}\widehat{h}^{cr_1-\xi_{r_1}}\widehat{Y}^{cr-\xi_r} = (\widehat{g}^{r_2}\widehat{h}^{r_1}\widehat{Y}^r)^c\widehat{g}^{-\xi_{r_2}}\widehat{h}^{-\xi_{r_1}}\widehat{Y}^{-\xi_r} = \widehat{R}^c\widehat{g}^{-\xi_{r_2}}\widehat{h}^{-\xi_{r_1}}\widehat{Y}^{-\xi_r},\\
B_5 &= U^{-\theta_{r_1}}u^{\theta_{\delta_1}} = U^{c\cdot r_1-\xi_{r_1}}u^{-c\cdot\delta_1+\xi_{\delta_1}} = U^{-\xi_{r_1}}u^{\xi_{\delta_1}}(U^{r_1}u^{-\delta_1})^c = U^{-\xi_{r_1}}u^{\xi_{\delta_1}}(u^{r_1k}u^{-r_1k})^c = U^{-\xi_{r_1}}u^{\xi_{\delta_1}},\\
B_6 &= V^{-\theta_{r_1}}v^{\theta_{\delta_2}} = V^{c\cdot r_1-\xi_{r_1}}v^{-c\cdot\delta_2+\xi_{\delta_2}} = V^{-\xi_{r_1}}v^{\xi_{\delta_2}}(V^{r_1}v^{-\delta_2})^c = V^{-\xi_{r_1}}v^{\xi_{\delta_2}}(v^{r_1k}v^{-r_1k})^c = V^{-\xi_{r_1}}v^{\xi_{\delta_2}},\\
B_7 &= \widehat{R}^{-\theta_{r_1}}\widehat{g}^{\theta_{\delta_3}}\widehat{h}^{\theta_{\delta_4}}\widehat{Y}^{\theta_{\delta_5}} = \widehat{R}^{cr_1-\xi_{r_1}}\widehat{g}^{-c\delta_3+\xi_{\delta_3}}\widehat{h}^{-c\delta_4+\xi_{\delta_4}}\widehat{Y}^{-c\delta_5+\xi_{\delta_5}} = \widehat{R}^{-\xi_{r_1}}\widehat{g}^{\xi_{\delta_3}}\widehat{h}^{\xi_{\delta_4}}\widehat{Y}^{\xi_{\delta_5}}(\widehat{R}^{r_1}\widehat{g}^{-\delta_3}\widehat{h}^{-\delta_4}\widehat{Y}^{-\delta_5})^c\\
&= \widehat{R}^{-\xi_{r_1}}\widehat{g}^{\xi_{\delta_3}}\widehat{h}^{\xi_{\delta_4}}\widehat{Y}^{\xi_{\delta_5}}((\widehat{g}^{r_2}\widehat{h}^{r_1}\widehat{Y}^r)^{r_1}\widehat{g}^{-r_1r_2}\widehat{h}^{-r_1^2}\widehat{Y}^{-r_1r})^c = \widehat{R}^{-\xi_{r_1}}\widehat{g}^{\xi_{\delta_3}}\widehat{h}^{\xi_{\delta_4}}\widehat{Y}^{\xi_{\delta_5}},\\
B_8 &= e(g, \widehat{X}\widehat{W}\widehat{R})^{\theta_{r_1}}e(S, \widehat{w})^{\theta_k+\theta_l}e(g, \widehat{w})^{-(\theta_{\delta_1}+\theta_{\delta_1})}e(S, \widehat{g})^{\theta_{r_2}}e(g, \widehat{g})^{-\theta_{\delta_3}}e(S, \widehat{h})^{\theta_{r_1}}e(g, \widehat{h})^{-\theta_{\delta_4}}\\
&= e(g, \widehat{X}\widehat{W}\widehat{R})^{-cr_1+\xi_{r_1}}e(S, \widehat{w})^{-c(k+l)+(\xi_k+\xi_l)}e(g, \widehat{w})^{c(\delta_1+\delta_2)-(\xi_{\delta_1}+\xi_{\delta_2})}e(S, \widehat{g})^{-cr_2+\xi_{r_2}}e(g, \widehat{g})^{c\delta_3-\xi_{\delta_3}}\\
&\quad\; e(S, \widehat{h})^{-cr_1+\xi_{r_1}}e(g, \widehat{h})^{c\delta_4-\xi_{\delta_4}}\\
&= e(g, \widehat{X}\widehat{W}\widehat{R})^{\xi_{r_1}}e(S, \widehat{w})^{(\xi_k+\xi_l)}e(g, \widehat{w})^{-(\xi_{\delta_1}+\xi_{\delta_2})}e(S, \widehat{g})^{\xi_{r_2}}e(g, \widehat{g})^{-\xi_{\delta_3}}e(S, \widehat{h})^{\xi_{r_1}}e(g, \widehat{h})^{-\xi_{\delta_4}}
\end{aligned}$$

$$(e(g,\widehat{X}\widehat{W}\widehat{R})^{-r_1}e(S,\widehat{w})^{-(k+l)}e(g,\widehat{w})^{(\delta_1+\delta_2)}e(S,\widehat{g})^{-r_2}e(g,\widehat{g})^{\delta_3}e(S,\widehat{h})^{-r_1}e(g,\widehat{h})^{\delta_4})^c$$

Next, we just need to show

$$e(g,\widehat{X}\widehat{W}\widehat{R})^{-r_1}e(S,\widehat{w})^{-(k+l)}e(g,\widehat{w})^{(\delta_1+\delta_2)}e(S,\widehat{g})^{-r_2}e(g,\widehat{g})^{\delta_3}e(S,\widehat{h})^{-r_1}e(g,\widehat{h})^{\delta_4} = e(g,\widehat{g})/e(S,\widehat{X}\widehat{W}\widehat{R})$$

i.e. show

$$e(S,\widehat{X}\widehat{W}\widehat{R})e(g,\widehat{X}\widehat{W}\widehat{R})^{-r_1}e(S,\widehat{w})^{-(k+l)}e(g,\widehat{w})^{(\delta_1+\delta_2)}e(S,\widehat{g})^{-r_2}e(g,\widehat{g})^{\delta_3}e(S,\widehat{h})^{-r_1}e(g,\widehat{h})^{\delta_4} = e(g,\widehat{g})$$

Lefthand

$$\begin{aligned}
&= (e(S,\widehat{X}\widehat{W}\widehat{R})e(g,\widehat{X}\widehat{W}\widehat{R})^{-r_1})(e(S,\widehat{w})^{-(k+l)}e(g,\widehat{w})^{r_1(k+l)})(e(S,\widehat{g})^{-r_2}e(g,\widehat{g})^{r_1r_2})(e(S,\widehat{h})^{-r_1}e(g,\widehat{h})^{r_1^2})\\
&= e(Sg^{-r_1},\widehat{X}\widehat{W}\widehat{R})e(Sg^{-r_1},\widehat{w})^{-(k+l)}e(Sg^{-r_1},\widehat{g})^{-r_2}e(Sg^{-r_1},\widehat{h})^{-r_1}\\
&= e(s,\widehat{X}\widehat{W}\widehat{R})e(s,\widehat{w})^{-(k+l)}e(s,\widehat{g})^{-r_2}e(s,\widehat{h})^{-r_1}\\
&= e(s,\widehat{X}\widehat{W}\widehat{R}\widehat{w}^{-(k+l)}\widehat{g}^{-r_2}\widehat{h}^{-r_1})\\
&= e(s,\widehat{X}\widehat{g}^{\mathtt{id}}\widehat{Y}^r)\\
&= e(g^{\frac{1}{x+\mathtt{id}+yr}},\widehat{g}^x\widehat{g}^{\mathtt{id}}(\widehat{g}^y)^r)\\
&= e(g,\widehat{g})\\
&= \textbf{Righthand}
\end{aligned}$$

Based on the registration correctness and the signing correctness, now we prove the opening correctness. On input (m,σ) where $\sigma = \langle S,\widehat{R},U,V,\widehat{W};c;\xi_{\mathtt{id}},\xi_r,\xi_{r_1},\xi_{r_2},\xi_k,\xi_l,\xi_{\delta_1},...,\xi_{\delta_5}\rangle$, we compute $W=\psi(\widehat{W})$ and use OA's secret key $sk_{\mathsf{OA}} = \langle\zeta,\eta\rangle$ to decrypt the triple $\langle U,V,W\rangle$ to obtain $g^{\mathtt{id}} = \frac{W}{U^\zeta V^\eta}$. Then use brute force, we can compute `id` from $g^{\mathtt{id}}$. Note that here `id` is very short, so the brute force is feasible.

□

A.2.2 Proof of Theorem 3.2

Proof. In order to prove the theorem above, we prove that any misidentification-forgery attacker in the random oracle model against our Hidden-IBS can be transformed to an adaptive chosen message attacker in the standard model against the BB signature.

Let $\mathcal{A}$ be a misidentification adversary as specified in the misidentification attack game that has access to a random oracle. We will reduce this adversary to an adaptive chosen message adversary for the BB digital signature.

Let us describe an adversary $\mathcal{B}$ for the BB signature. First, the adversary is given the public key of the signature which is $\langle p,g,\widehat{g},\mathbb{G},\widehat{\mathbb{G}},\psi,\mathbb{G}_T,e,\widehat{X},\widehat{Y}\rangle$ as described in the generation of the public parameters. Now $\mathcal{B}$ using this information samples all remaining public parameters for the Hidden-IBS as in the `Setup` procedure, and the formed public parameters are `pub` $= \langle p,g,\widehat{g},\mathbb{G},\widehat{\mathbb{G}},\psi,\mathbb{G}_T,e;h,\widehat{h};\widehat{X},\widehat{Y};u,v,w,\widehat{u},\widehat{v},\widehat{w};\mathcal{H}\rangle$. Note that $\mathcal{B}$ knows the secret key of the OA. However he does not know the secret of the IM, i.e. $x=\log_{\widehat{g}}\widehat{X}$ and $y=\log_{\widehat{g}}\widehat{Y}$.

Before the simulation of $\mathcal{A}$, the BB-forger $\mathcal{B}$ initializes a table $\mathcal{H}$ for the simulation of the random oracle that is employed by $\mathcal{A}$. Now $\mathcal{B}$ starts the simulation of the adversary $\mathcal{A}$, and $\mathcal{B}$ has two modes.

- In mode 1, $\mathcal{B}$ is required to respond to the oracle queries below:
  - For the random oracle queries, $\mathcal{B}$ uses the table $\mathcal{H}$: if the query is already on the table, $\mathcal{B}$ returns the corresponding value; if not, $\mathcal{B}$ samples a random element in $\mathbb{Z}_p$, places it in the table, and returns it.
  - Consider that $\mathcal{B}$ does not know the IM secret key $\langle x,y\rangle$. To answer the registration oracle queries, $\mathcal{B}$ needs the help from the BB signing oracle. Be in detail, $\mathcal{B}$ receives `id` from $\mathcal{A}$; $\mathcal{B}$ forwards `id` to the BB-signing oracle; when $\mathcal{B}$ receives from the BB-signing oracle the response (s,r) such that $e(s,\widehat{X}\widehat{g}^{\mathtt{id}}\widehat{Y}^r)=e(g,\widehat{g})$, he records $\langle\mathtt{id};s,r\rangle$, and returns 1 to the $\mathcal{A}$.

- If $\mathcal{A}$ queries for corrupting a user with id, if the id has been registered, then $\mathcal{B}$ sends $\mathcal{A}$ the recorded $\langle s, r\rangle$, otherwise sends $\perp$.
- If $\mathcal{A}$ queries message m for a signature from a registered user with id, $\mathcal{B}$ can generate a valid $\sigma = \langle S, \widehat{R}, U, V, \widehat{W}; c; \xi_{\mathtt{id}}, \xi_r, \xi_{r_1}, \xi_{r_2}, \xi_k, \xi_l, \xi_{\delta_1}, ..., \xi_{\delta_5}\rangle$ as that in the signing algorithm because $\mathcal{B}$ knows the membership certificate (s, r) for the registered id. Then $\mathcal{B}$ returns such σ to $\mathcal{A}$.

After the queries, $\mathcal{A}$ will output a message-signature pair (m^*, σ^*), where $\sigma^* = \langle S^*, \widehat{R}^*, U^*, V^*, \widehat{W}^*; c^*; \xi^*_{\mathtt{id}}, \xi^*_r, \xi^*_{r_1}, \xi^*_{r_2}, \xi^*_k, \xi^*_l, \xi^*_{\delta_1}, ..., \xi^*_{\delta_5}\rangle$.

With probability ϵ_1, the generated pair can pass the verification and the signature cannot be opened to any existing identities, then we take the forking technique from Pointcheval and Stern by using a different challenge $\widetilde{c}^* \neq c^*$ for the same queries $(m^*, S^*, \widehat{R}^*, U^*, V^*, \widehat{W}^*)$ to the random oracle, and we can extract all witnesses. With the witnesses k^*, l^*, we can obtain $g^{\mathtt{id}^*} = W^* w^{-(k^*+l^*)}$ where $W^* = \psi(\widehat{W}^*)$, and further obtain id* by brute-force $g^{\mathtt{id}^*}$, where id* is only $\log\lambda$ short; with the witness r^*_1, we can obtain $s^* = S^* g^{-r^*_1}$. So now we have id* and its BB-signature (s^*, r^*). Consider the signature cannot be opened to any existing identities which means the id* has never been queried. Therefore, $\mathcal{B}$ can obtain a valid BB-forgery $\langle \mathtt{id}^*; s^*, r^*\rangle$.

If the generated pair can pass the verification, but the signature is opened to some registered but non-corrupted identity id*, and the message m^* has never been queried in the history of the user with id*, then $\mathcal{B}$ just terminates.

- In mode 2, $\mathcal{B}$ responses the oracle queries below in a different way:
  - For the random oracle queries, $\mathcal{B}$ uses the table $\mathcal{H}$ by the same way in mode 1: if the query is already on the table, $\mathcal{B}$ returns the corresponding value; if not, $\mathcal{B}$ samples a random element in $\mathbb{Z}_p$, places it in the table, and returns it.
  - $\mathcal{B}$ answers the registration oracle queries in a different way from that in the mode 1: when $\mathcal{B}$ receives id from $\mathcal{A}$; $\mathcal{B}$ always returns 1 to the $\mathcal{A}$.
  - If $\mathcal{A}$ queries for corrupting a user with id, then $\mathcal{B}$ asks the help from the BB signing oracle to generate a BB-signature (s, t) for id, and returns such (s, t) as the response to $\mathcal{A}$.
  - If $\mathcal{A}$ queries message m for a signature from a registered user with id, $\mathcal{B}$ cannot generate a valid signature σ like that in mode 1, because now he does not know the membership certificate (s, t) for id. However he can generate $\sigma = \langle S, \widehat{R}, U, V, \widehat{W}; c; \xi_{\mathtt{id}}, \xi_r, \xi_{r_1}, \xi_{r_2}, \xi_k, \xi_l, \xi_{\delta_1}, ..., \xi_{\delta_5}\rangle$ which can pass the verification by patching the random oracle at $(m||S||\widehat{R}||U||V||\widehat{W}||B_1||... ||B_8)$ to equal c, where $B_1, ...B_8$ can be reconstructed based on the signature and the public information. After this, $\mathcal{B}$ returns such σ to $\mathcal{A}$.

After the queries, $\mathcal{A}$ will output a message-signature pair (m^*, σ^*), where $\sigma^* = \langle S^*, \widehat{R}^*, U^*, V^*, \widehat{W}^*; c^*; \xi^*_{\mathtt{id}}, \xi^*_r, \xi^*_{r_1}, \xi^*_{r_2}, \xi^*_k, \xi^*_l, \xi^*_{\delta_1}, ..., \xi^*_{\delta_5}\rangle$.

With probability ϵ_2, the generated pair can pass the verification and the signature is opened to an honest id* which means the id* has appeared in the registration queries but not in the corruption queries, also m^* has never been queried in the signing queries history for the registered user with id, $\mathcal{A}$ will win and $\mathcal{A}$ is successful misidentification attacker. We again take the forking technique from Poincheval and Stern by using a different challenge $\widetilde{c}^* \neq c^*$ for the same queries $(m^*, S^*, \widehat{R}^*, U^*, V^*, \widehat{W}^*)$ for the random oracle, and we can extract all the witnesses. And from the witness, we obtain a BB message-signature pair $(\mathtt{id}^*; s^*, r^*)$. Note that id* has not appeared in the querying history of the BB signing oracle, because $\mathcal{B}$ only queries the BB signing oracle when a corruption query from $\mathcal{A}$ happens. Therefore, $\mathcal{B}$ can obtain a valid BB-forger $\langle \texttt{id}^*; s^*, r^* \rangle$.

If the generated pair can pass the verification, but the signature cannot be opened to any existing identities, then $\mathcal{B}$ just terminates.

Now $\mathcal{B}$ randomly choose mode 1 and mode 2, and the $\mathcal{A}$ cannot detect that which mode he is involved, so $\mathcal{B}$ has probability $\frac{1}{2}(\epsilon_1 + \epsilon_2)$ to obtain a successful BB-forgery. Consider $\mathcal{A}$ is a successful misidentification attacker, so ϵ_1 and ϵ_2 cannot be negligible at the same time. Therefore, $\mathcal{B}$ can obtain a successful BB-forgery with non-negligible probability which is against the SDH assumption.

□

A.2.3 Proof of Theorem 3.3

Proof. To prove the theorem above, we prove that any anonymity attacker against our Hidden-IBS in the random oracle model can be transformed to a CPA attacker against the Linear Encryption.

Let $\mathcal{A}$ be an anonymity attacker as specified in the CPA-anonymity attack game. Note that $\mathcal{A}$ has access to the random oracle $\mathcal{H}$. We show how to transform $\mathcal{A}$ into a CPA adversary $\mathcal{B}$ against the Linear Encryption.

First, $\mathcal{B}$ is given the public key of the Linear Encryption which is $\langle p, g, \widehat{g}, \mathbb{G}, \widehat{\mathbb{G}}, \psi, \mathbb{G}_T, e, u, v, w, \widehat{u}, \widehat{v}, \widehat{w} \rangle$ as described in the generation of the public parameters. Now $\mathcal{B}$ using this information samples all remaining public parameters for the Hidden-IBS as in the `Setup` procedure, and the formed public parameters are $\texttt{pub} = \langle p, g, \widehat{g}, \mathbb{G}, \widehat{\mathbb{G}}, \psi, \mathbb{G}_T, e; h, \widehat{h}; \widehat{X}, \widehat{Y}; u, v, w, \widehat{u}, \widehat{v}, \widehat{w}; \mathcal{H} \rangle$. Note that $\mathcal{B}$ knows the secret key of the IM. However he does not know the secret of the OA, i.e. ζ, η such that $u^\zeta = v^\eta = w$ (i.e. $\widehat{u}^\zeta = \widehat{v}^\eta = \widehat{w}$).

Before the simulation of $\mathcal{A}$, the BB-forger $\mathcal{B}$ initializes a table $\mathcal{H}$ for the simulation of the random oracle that is employed by $\mathcal{A}$. Now $\mathcal{B}$ starts the simulation of the adversary $\mathcal{A}$, and $\mathcal{B}$ is required to simulate the oracle queries below:

For the random oracle queries, $\mathcal{B}$ uses the table $\mathcal{H}$: if the query is already on the table, $\mathcal{B}$ returns the corresponding value; if not, $\mathcal{B}$ samples a random element in $\mathbb{Z}_p$, places it in the table, and returns it.

When $\mathcal{A}$ requests its CPA-anonymity challenge by providing two users' identities, $\texttt{id}_0$, $\texttt{id}_1$, and a message m. In turn $\mathcal{B}$ requests its indistinguishability challenge by providing $\widehat{M_0} = \widehat{g}^{\texttt{id}_0}$ and $\widehat{M_1} = \widehat{g}^{\texttt{id}_1}$. It is given a Linear encryption $\langle \widehat{U}, \widehat{V}, \widehat{W} \rangle$ of $\widehat{M_b} = \widehat{g}^{\texttt{id}_b}$, where bit b is chosen by the Linear Encryption challenger. The tuple $\langle U, V, W \rangle$ can be computed by using the isomorphism ψ.

$\mathcal{B}$ can respond with a valid $\sigma = \langle S, \widehat{R}, U, V, \widehat{W}; c; \xi_{\texttt{id}}, \xi_r, \xi_{r_1}, \xi_{r_2}, \xi_k, \xi_l, \xi_{\delta_1}, ..., \xi_{\delta_5} \rangle$. Here we need to patch the random oracle at $(m||S||\widehat{R}||U||V||\widehat{W}||B_1||...||B_8)$ to equal c, where $B_1, ..., B_8$ are computed from the σ and the public information.

Finally, $\mathcal{A}$ outputs a bit b^*. $\mathcal{B}$ returns b^* as the answer to its own challenge. It is easy to verify that $\mathcal{B}$ is a CPA adversary for the employed Linear Encryption, which is against the DLDH assumption. □

$$\texttt{pub} = \langle n, a_0, a, b; N, G, H_1, H_2, H_3, \mathsf{hash}, \mathsf{hk}; g, f_1, f_2, f_3, f_4, f_5, f_6 \rangle$$

$\boxed{\texttt{User}}$ $\qquad\qquad$ $\boxed{\texttt{Verifier}}$

$x, \texttt{name}; v, e, s; x_1, x_2$

$x = x_1 x_2$, $v^e = a_0 a^{x+\texttt{name}\cdot 2^\ell} b^s$

$r_1, r_2, r_3 \overset{\mathtt{r}}{\leftarrow} \pm\{0,1\}^{\ell_n - 2}$

$d \overset{\mathtt{r}}{\leftarrow} \pm\{0,1\}^{\ell_N - 2}$, $r_4 = r_1 x_2$, $r_5 = r_2 e$

$T_1 = g^{r_1} f_1^{x_1}$, $T_2 = g^{r_2} v$

$T_3 = g^{r_3} f_1^{x} f_2^{x_2} f_3^{\texttt{name}} f_4^{e} f_5^{s} f_6^{d}$

$C_1 = G^d$ (in $\mathbb{Z}_{N^2}^*$),

$C_2 = H_1^d (1+N)^{x+\texttt{name}\cdot 2^\ell}$ (in $\mathbb{Z}_{N^2}^*$),

$C_3 = \mathrm{abs}\big((H_2 H_3^{\mathsf{hash}(\mathsf{hk}, C_1, C_2)})^d\big)$ (in $\mathbb{Z}_{N^2}^*$),

$\theta_x \overset{\mathtt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\mu'}$

$\theta_{x_1}, \theta_{x_2} \overset{\mathtt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\mu''}$

$\theta_{\texttt{name}} \overset{\mathtt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_{\texttt{name}}}$

$\theta_e \overset{\mathtt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_e}$, $\theta_s \overset{\mathtt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_s}$

$\theta_d \overset{\mathtt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_d}$,

$\theta_{r_1}, \theta_{r_2}, \theta_{r_3} \overset{\mathtt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_n-2}$

$\theta_{r_4} \overset{\mathtt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_n-2+\mu''}$,

$\theta_{r_5} \overset{\mathtt{r}}{\leftarrow} \pm\{0,1\}^{\lambda_0+\lambda_1+\ell_n-2+\ell_e}$,

$B_1 = g^{-\theta_{r_1}} f_1^{-\theta_{x_1}}$, $B_2 = T_1^{-\theta_{x_2}} g^{\theta_{r_4}} f^{\theta_x}$

$B_3 = T_2^{-\theta_e} g^{\theta_{r_5}} a^{\theta_x + \theta_{\texttt{name}}\cdot 2^\ell} b^{\theta_s}$,

$B_4 = g^{-\theta_{r_3}} f_1^{-\theta_x} f_2^{-\theta_{x_2}} f_3^{-\theta_{\texttt{name}}} f_4^{-\theta_e} f_5^{-\theta_s} f_6^{-\theta_d}$

$B_5 = G^{-\theta_d}$ (in $\mathbb{Z}_{N^2}^*$),

$B_6 = H_1^{-\theta_d} (1+N)^{-(\theta_x + \theta_{\texttt{name}}\cdot 2^\ell)}$ (in $\mathbb{Z}_{N^2}^*$),

$B_7 = (H_2 H_3^{\mathsf{hash}(\mathsf{hk}, C_1, C_2)})^{-2\theta_d}$ (in $\mathbb{Z}_{N^2}^*$),

$$\xrightarrow[B_1, \ldots, B_7]{T_1, T_2, T_3, C_1, C_2, C_3;}$$

$c \overset{\mathtt{r}}{\leftarrow} \{0,1\}^{\lambda_0}$

$$\xleftarrow{\quad c \quad}$$

$\xi_{\texttt{name}} = \theta_{\texttt{name}} + c \cdot \texttt{name}$

$\xi_x = \theta_x + c \cdot (x - 2^{\ell'})$,

$\xi_{x_1} = \theta_{x_1} + c \cdot (x_1 - 2^{\ell''})$,

$\xi_{x_2} = \theta_{x_2} + c \cdot (x_2 - 2^{\ell''})$

$\xi_e = \theta_e + c \cdot e$, $\xi_s = \theta_s + c \cdot s$,

$\xi_d = \theta_d + c \cdot d$, $\xi_{r_1} = \theta_{r_1} + c \cdot r_1$,

$\xi_{r_2} = \theta_{r_2} + c \cdot r_2$, $\xi_{r_3} = \theta_{r_3} + c \cdot r_3$,

$\xi_{r_4} = \theta_{r_4} + c \cdot r_4$, $\xi_{r_5} = \theta_{r_5} + c \cdot r_5$

$$\xrightarrow[\xi_{r_1}, \ldots, \xi_{r_5}]{\xi_{\texttt{name}}, \xi_x, \xi_{x_1}, \xi_{x_2}, \xi_e, \xi_s, \xi_d,}$$

$\xi_x \in^? \pm\{0,1\}^{\lambda_0+\lambda_1+\mu'+1}$

$\xi_{x_1}, \xi_{x_2} \in^? \pm\{0,1\}^{\lambda_0+\lambda_1+\mu''+1}$

$C_1, C_2, C_3 \in^? \mathbb{Z}_{N^2}^*$, $C_2 \leq^? N^2/2$

$g^{\xi_{r_1}} f_1^{\xi_{x_1}} B_1 =^? (T_1)^c$,

$T_1^{\xi_{x_2}} g^{-\xi_{r_4}} f_1^{-\xi_x} B_2 =^? 1$

$T_2^{\xi_e} g^{-\xi_{r_5}} a^{-(\xi_x + \xi_{\texttt{name}}\cdot 2^\ell)} b^{-\xi_s} B_3 =^? (a_0)^c$,

$g^{\xi_{r_3}} f_1^{\xi_x} f_2^{\xi_{x_2}} f_3^{\xi_{\texttt{name}}} f_4^{\xi_e} f_5^{\xi_s} f_6^{\xi_d} B_4 =^? (T_3)^c$

$G^{\xi_d} B_5 =^? (C_1)^c$ (in $\mathbb{Z}_{N^2}^*$)

$H_1^{\xi_d} (1+N)^{\xi_x + \xi_{\texttt{name}}\cdot 2^\ell} B_6 =^? (C_2)^c$ (in $\mathbb{Z}_{N^2}^*$)

$(H_2 H_3^{\mathsf{hash}(\mathsf{hk}, C_1, C_2)})^{2\xi_d} B_7 =^? (C_3)^{2c}$ (in $\mathbb{Z}_{N^2}^*$)

Figure 8: The hidden identity-based identification protocol with exculpability.