Fully Anonymous Group Signature with Verifier-Local Revocation*

Ai Kitagawa† Yusuke Sakai† Keita Emura‡
Goichiro Hanaoka† Keisuke Tanaka§

January 17, 2021

Abstract

Group signature with verifier-local revocation (VLR-GS) is a special type of revocable group signature which enables a user to sign messages without referring to information regarding revoked users. Although there have been several proposals of VLR-GS schemes since the first scheme proposed by Boneh and Shacham [CCS 2004], all of these schemes only achieve a security notion called *selfless anonymity*, which is strictly weaker than the de facto standard security notion, *full anonymity*. Thus, for more than a decade, it has been an open problem whether a fully anonymous VLR-GS scheme can be achieved. In this paper, we give an affirmative answer to this problem. Concretely, we show the construction of a fully anonymous VLR-GS scheme from a digital signature scheme, a key-private public key encryption scheme, and a non-interactive zero-knowledge proof system. Also, we show that backward unlinkability, which ensures that even after a user is revoked, signatures produced by the user before the revocation remain anonymous, can be realized without additional building blocks. Although the size of group public key and signing key depend on the number of time periods, finally, we show that the size of these keys can be reduced by employing an identity-based encryption scheme.

Keywords: Group Signature, Verifier-Local Revocation, Full Anonymity, Backward Unlinkability

*An extended abstract appeared at SCN 2018 [23]. This is the full version.

†National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan. Email: {a.kitagawa, yusuke.sakai, hanaoka-goichiro}@aist.go.jp

‡National Institute of Information and Communications Technology (NICT), Tokyo, Japan. Email: k-emura@nict.go.jp

§Tokyo Institute of Technology, Tokyo, Japan. Email: keisuke@is.titech.ac.jp

1 Introduction

1.1 Background

Group Signature and Revocation. The notion of group signature was introduced by Chaum and van Heyst [13]. In a group signature scheme, a group manager called an issuer generates user signing keys by using the issuing key, and users can anonymously sign messages on behalf of the group with their own signing keys. However, in the case of disputes, a group manager called an opener can identify the signer from a signature.

Membership revocation is one of the most important research topics in group signatures, and has been widely investigated so far. Currently, there are two main approaches for realizing a group signature scheme with revocation functionality. The first approach is to periodically publish information related to the revoked users, and require both users and verifiers to use this when generating or verifying signatures [7, 3, 10, 20, 37, 15, 34, 32, 28, 29, 19, 39].[1] A scheme obtained by such an approach is sometimes inconvenient since users need to download the up-to-date information whenever signing. The second approach, group signature with verifier-local revocation (VLR-GS) [9] on which we focus in this paper, is free from this concern.

Group Signature with Verifier-local Revocation. The notion of VLR-GS was proposed by Boneh and Shacham [9]. After that, Nakanishi and Funabiki [35] extended the security notion for this type of scheme by considering backward unlinkability. The first scheme secure in the standard model was proposed by Libert and Vergnaud [31], and a lattice-based scheme was introduced by Langlois, Ling, Nguyen, and Wang [26] and Zhang, Liu, Hu, Zhang, and Jia [51].

In a VLR-GS scheme, verifiers need to download the up-to-date information of the revoked users to verify signatures but signers are not required to do so. That is, signers can generate signatures without any additional information of the revoked users. More precisely, a VLR-GS scheme operates as follows: a token (called a revocation token) is defined for each user, and the authority reveals this in a public list (called a revocation list) if the corresponding user is revoked. Namely, the revocation list contains the revocation tokens of the revoked users. A revocation token can be used to detect the signature generated by the corresponding user. Thus, a verifier can check whether the signer is revoked by using the revocation list. However, a signer can generate signatures using only his/her signing key, that is, he/she does not need to refer to the revocation list. Such a functionality is very attractive when it is difficult for users to periodically obtaining up-to-date information.

However, there is one drawback: all existing VLR-GS schemes have only been proved to satisfy a weak security notion called *selfless anonymity*, whereas several standard revocable group signature schemes (e.g., the schemes proposed by Libert, Peters, and Yung [28, 29]) satisfy a strong security notion called *full anonymity*. Specifically, there are trivial attacks against the full anonymity of almost all existing VLR-GS schemes. We provide more details of these two security notions in the next paragraph.

Full Anonymity vs. Selfless Anonymity. Full anonymity ensures that the signer's information cannot be extracted from a signature by an adversary with all user signing keys.[2] Selfless anonymity is a weaker security notion than full anonymity, and ensures the anonymity of a signature only against an adversary who does not possess the user signing key which was used in the generation of the corresponding signature.

From a practical point of view, a selfless-anonymous group signature scheme has two drawbacks: it is not resistant to the leakage of user signing keys and it might allow the issuer to identify the signer. More precisely, selfless anonymity does not ensure that the signer's information cannot be extracted from a signature by an adversary who has the signing key used to generate the signature. Therefore, once a signing key is exposed, the anonymity of the signatures generated by this signing key can no longer be guaranteed. Also, anonymity against the issuer cannot be ensured since he/she knows all user signing keys. Thus, selfless anonymity does not provide a security level strong enough for practical use, and full anonymity is recognized as one of the de facto standard security requirements of group signature

[1] In fully dynamic group signature schemes [3, 10, 32], the information is related to the current group members in addition to the revoked users.

[2] For simplicity, in this paper, we adopt the notion of full anonymity in the CPA-setting [7]. We remark that it is considered to be easy to upgrade to full anonymity in the CCA-setting by using standard techniques for acquiring CCA-security in a public key encryption scheme (for details, see **Remark 1** in Section 3.2).

	Building Blocks	Backward Anonymity	Size of gpk	Size of gsk
Scheme 1 (§3)	Digital Signature, PKE, NIZK	No	$O(1)$	$O(1)$
Scheme 2 (§4)	Digital Signature, PKE, NIZK	Yes	$O(T)$	$O(T)$
Scheme 3 (§5)	IBS, IBE, NIZK	Yes	$O(1)$	$O(1)$

* T: the number of time periods

Table 1: Our Fully Anonymous VLR-GS Schemes

(e.g., [7, 6, 20, 21, 25, 30, 38, 27, 16]). We remark that in VLR-GS the issuer has all revocation tokens and thus it can execute the implicit opening algorithm (which is defined later). Thus, we do not have to consider anonymity against the issuer unlike to other group signature schemes above. Nevertheless, full anonymity is important even in the VLR setting for considering signing key exposure above.

Although it is more desirable that group signatures satisfy full anonymity than selfless anonymity, it is more challenging to construct a fully anonymous group signature scheme since there is a big theoretical gap between selfless-anonymous group signature and fully anonymous group signature. In particular, Camenisch and Groth [11] showed that a selfless-anonymous group signature scheme can be constructed from a one-way function and a non-interactive zero-knowledge (NIZK) proof system. In contrast, several results [2, 40, 18] suggest that a public key encryption (PKE) scheme is an essential building block for constructing a fully anonymous group signature scheme. Therefore, it seems that the gap between selfless-anonymous group signature and fully anonymous group signature is the same as that between one-way function and PKE. Thus, it is an open problem whether a fully anonymous VLR-GS scheme can be achieved whereas selfless-anonymous VLR-GS schemes have already been proposed so far.

1.2 Our Contribution

In this paper, we give an affirmative answer to the above problem and give the first fully anonymous VLR-GS scheme. Concretely, we show three schemes summarized in Table 1. First, we construct a fully anonymous VLR-GS scheme from a digital signature scheme, a PKE scheme, and an NIZK proof system. Although the building blocks are essentially the same as those of a standard group signature scheme given by Bellare-Micciancio-Warinschi (BMW) [5], we additionally require the underlying PKE scheme to satisfy key privacy [4], which is essential to ensure that the VLR-GS scheme is fully anonymous. The first scheme shows a minimum requirement so far for achieving full anonymity. Second, we construct a fully anonymous VLR-GS scheme with backward unlinkability [35], which ensures that even after a user is revoked, signatures produced by the user before the revocation remain anonymous. The building blocks are the same as those of the first scheme. The second scheme shows that backward unlinkability can be realized without additional building blocks. Third, we construct a fully anonymous VLR-GS scheme with backward unlinkability with constant size group public key and signing key. We additionally employ a key-private identity-based encryption (IBE) scheme [3] as an additional building block. Since IBE is a stronger cryptographic primitive than PKE [8], the third scheme shows that we can reduce the size of these keys by employing such a strong tool. Although we also employ an identity-based signature (IBS) scheme, we remark that, from the feasibility point of view, IBS can be generically constructed from ordinary digital signature.

Differences from the Conference Version. An extended abstract appeared at SCN 2018 [23]. This is the full version. In this version, first we fixed a bug of Scheme 1 in [23] where it does not consider a strong unforgeability. That is, in the definition of traceability, m^*, which is a message of the forged signature, can be queried to the signing oracle. We consider the case by introducing a one-time signature (OTS) scheme. Second, we give a VLR-GS scheme which satisfies backward unlinkability (Scheme 2). Althugh we have insisted that such a scheme can be constructed from the same building blocks of Scheme 1 (at the expense of the public key size) in [23], we formally give the scheme in this version. Third, we give a new construction (Scheme 3). Although an IBE-based scheme has been given in [23], the size of gpk depends on T while that of Scheme 3 is constant. Finally, we give the detail of our cryptanalysis of the Perera-Koshiba scheme [41] while we gave a sketch of the attack in the conference version.

[3] Such a scheme is usually called anonymous IBE. However, we used the terminology key privacy (as in PKE) because we would like to use the word anonymity for indicating a security notion of group signatures.

1.3 Related Work

Perera and Koshiba [41, 48, 43, 45, 49, 47] proposed VLR-GS schemes which were claimed to be fully anonymous. However, in fact, these schemes do not satisfy full anonymity. In [43], all revocation tokens are signed by a secret signing key of the group manager, and the verification algorithm checks whether or not the signature of a revocation token is valid under the group manager verification key before verifying a group signature. They claimed that an adversary who knows all signing keys does not know the group manager signing key, and thus the adversary cannot produce a valid group signature even the adversary can generate revocation tokens from signing keys, and thus the scheme is fully anonymous. However, this attempt is wrong because one can ignore the verification process of revocation tokens and then check whether a signer has been revoked or not by checking the validity of group signature. In [33, 48, 49, 47, 45, 53], they separately generate signing keys and revocation tokens, and insisted that revocation tokens cannot be generated even all signing keys are exposed. The problem is that the signing algorithm takes a revocation token in addition to a signing key in their syntax (though in usual syntax of VLR-GS, the signing algorithm does not take a revocation token as input), and requiring a revocation token for signing means that a revocation token is regarded as a part of signing key. Thus, it is unnatural that an adversary can obtain all signing keys but cannot obtain revocation tokens. We remark that they have introduced the notion almost-full anonymity [44, 42, 46] that captures the above situation. So, the schemes in [33, 48, 49, 47, 45, 53] are also almost-fully anonymous.[4] For the scheme [41], we give a concrete attack in Section 6. In summary, their all schemes do not provide full anonymity, and no fully anonymous VLR-GS scheme has been proposed from the best of our knowledge.

1.4 Technical Overview

We will now give a technical overview of our constructions. Since we can obtain our scheme with backward unlinkability by extending our scheme without backward unlinkability, here, we only explain the construction of a VLR-GS scheme without backward unlinkability (Scheme 1). For details of the scheme with backward unlinkability, see Section 4.

Previous Approach. As mentioned above, all existing VLR-GS schemes satisfy only selfless anonymity. Specifically, there are trivial attacks against the full anonymity for most of the schemes [9, 35, 36, 50, 54, 12, 26] owing to their structure allowing the revocation token to be constructed from the corresponding user's signing key.[5] The Libert-Vergnaud scheme [31] is only exception, but these scheme has still only been proved to be selfless-anonymous. Recall that the revocation token can be used to detect signatures generated by the corresponding user. Thus, if the revocation token can be constructed from the corresponding signing key, an adversary holding all user signing keys can identify the signer from any signature by computing all users' revocation tokens. That is, a VLR-GS scheme with such a structure can never satisfy full anonymity. Therefore, if we attempt to achieve a fully anonymous VLR-GS scheme, we have to construct it from scratch.

Our Approach. Our construction mainly follows the construction of a group signature scheme proposed by Bellare, Micciancio, and Warinschi [5]. Then, we add revocation functionality by employing additional key pairs of a key-private PKE scheme [17, 14] for each user. Intuitively, a decryption key of the PKE scheme is used as a revocation token, and a signer computes a certain ciphertext using his encryption key

[4] Zhang et al. [52] showed that two Perera et al. schemes [48, 49] are not fully anonymous by giving concrete attacks. Moreover, they give an improved Stern-type protocol and claim that their group signature scheme is fully anonymous. However, since they follow the Perera et al.'s syntax, the signing algorithm takes a revocation token in addition to a signing key. Thus, their scheme is also almost fully anonymous.

[5] In the Wei-Liu scheme [50], additional revocation queries are required. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups with prime order p, and $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ be generators. For $\gamma, r_1, \ldots, r_T \in \mathbb{Z}_p$, $\omega = g_2^{\gamma}$ and $h_j = g_1^{r_j}$ for $j \in [1, T]$ are contained in gpk. For a user i, $(A_i, x_i) \in \mathbb{G}_1 \times \mathbb{Z}_p$ is a signing key, $\mathsf{grt}[i][j] = ((\omega g_2^{x_i})^{r_j}, h_j^{-x_i})$ is a revocation token at j. We describe the attack as follows. Let j^* be the target time period. Revoke two users i_0 and i_1 at j^* where they are not the challenge users. Obtain $(\omega g_2^{x_{i_0}})^{r_{j^*}}$ and $(\omega g_2^{x_{i_1}})^{r_{j^*}}$. Now x_{i_0} and x_{i_1} are obtained via the corruption oracle. Thus, compute $\{(\omega g_2^{x_{i_0}})^{r_{j^*}} / (\omega g_2^{x_{i_1}})^{r_{j^*}}\}^{1/(x_{i_0} - x_{i_1})} = \{(g_2^{x_{i_0} - x_{i_1}})^{r_{j^*}}\}^{1/(x_{i_0} - x_{i_1})} = g_2^{r_{j^*}}$. Then, compute $(\omega g_2^{x_{i_0}})^{r_{j^*}} / (g_2^{r_{j^*}})^{x_{i_0}} = \omega^{r_{j^*}}$. In the full anonymity setting, for the challenge user i_0^*, $x_{i_0^*}$ can be revealed. Thus, the revocation token $\mathsf{grt}[i_0^*][j^*]$ can be computed by $\omega^{r_{j^*}} (g_2^{r_{j^*}})^{x_{i_0^*}} = (\omega g_2^{x_{i_0^*}})^{r_{j^*}}$ and $h_{j^*}^{-x_{i_0^*}}$. This token can be used for distinguising whether the challenge signature is generated by the signing key of i_0^* or i_1^*.

of the PKE scheme as a part of a signature. A verifier can check whether a signature is generated by a revoked user by decrypting the ciphertext in the signature using all revocation tokens in the revocation list.

A more detailed explanation of our scheme is given in the following. In the BMW construction [5], each user possesses a certified key pair $(\mathsf{vk}_i, \mathsf{sk}_i)$ of a digital signature scheme. When signing a message m, the user i generates a signature σ on the message m using his/her signing key sk_i, and encrypts σ using the group manager's encryption key $\mathsf{ek}_{\mathsf{PKE}}$ of a PKE scheme to achieve anonymity. Let ct be this ciphertext. Moreover, the user produces an NIZK proof which proves that the series of procedures is honestly done and the signing key is certified. Thus, the signature in the BMW construction consists of a ciphertext ct and a proof.

In addition to the BMW construction, we employ additional key pairs of a PKE scheme to achieve verifier-local revocation functionality. More precisely, the group manager generates a key pair $(\mathsf{ek}_i, \mathsf{dk}_i)$ for each user i and certifies it, and sends only the encryption key ek_i as a part of the signing key to the user. Moreover, the manager sets the decryption key dk_i as the revocation token of the user i. Even if ek_i is revealed, dk_i is not revealed (if it is, the underlying PKE scheme is immediately broken). When signing a message m, the user i also computes a ciphertext $\widetilde{\mathsf{ct}}$ of the signature σ under the encryption key ek_i in addition to a ciphertext ct under $\mathsf{ek}_{\mathsf{PKE}}$. Then, the user produces an NIZK proof π which proves that the series of procedures is honestly done, and the signing key and the encryption key are certified. We remark that the user generates an OTS key pair $(\mathsf{vk}_{\mathrm{ots}}, \mathsf{sk}_{\mathrm{ots}})$ and generates σ on the message $\mathsf{vk}_{\mathrm{ots}}$ while it is a signature on m in the BMW construction. Finally, the user computes an OTS σ_{all} on $(\mathsf{vk}_{\mathrm{ots}}, \mathsf{ct}, \widetilde{\mathsf{ct}}, \pi)$. Adding the OTS part allows us to provide strong unforgeability which is captured by traceability. The signature in our scheme is a tuple of two ciphertexts, ct and $\widetilde{\mathsf{ct}}$, a proof π, the OTS verification key $\mathsf{vk}_{\mathrm{ots}}$, and a OTS σ_{all}. A verifier can check whether the signer is revoked by decrypting the underlying ciphertext $\widetilde{\mathsf{ct}}$ using all revocation tokens in the revocation list and checking that it can be decrypted by some revocation token.

Intuitively, our scheme satisfies full anonymity since the revocation token (i.e., the decryption key dk_i) cannot be computed from the corresponding signing key (i.e., the key which contains the encryption key ek_i) due to the security of the underlying PKE scheme. To implement this idea, the PKE scheme is also required to be key-private since the encryption key ek_i contained in the ciphertext $\widetilde{\mathsf{ct}}$ is associated with the signer i and may leak the identity of the signer.

2 Preliminaries

In this section, we define some notations and cryptographic primitives which we use in this paper.

Notations. $x \xleftarrow{\$} X$ denotes choosing an element from a finite set X uniformly at random. If A is a probabilistic algorithm, $y \leftarrow \mathsf{A}(x; r)$ denotes the operation of running A on an input x and a randomness r, and letting y be the output. When it is not necessary to specify the randomness, we omit it and simply write $y \leftarrow \mathsf{A}(x)$. If we describe the statement that the output of $\mathsf{A}(x)$ is y, then we denote $\mathsf{A}(x) = y$. If $\mathcal{O}$ is a function or an algorithm, $\mathsf{A}^{\mathcal{O}}$ denotes that A has oracle access to $\mathcal{O}$. λ denotes a security parameter. PPT stands for probabilistic polynomial time. A function $f(\lambda)$ is called negligible and denoted as $\mathsf{negl}(\lambda)$ if for any $c > 0$, there exists an integer Λ such that $f(\lambda) < \frac{1}{\lambda^c}$ for all $\lambda > \Lambda$.

2.1 Cryptographic Primitives

Digital Signature. A signature scheme $\mathcal{SIG}$ consists of three algorithms $(\mathsf{SIG.Gen}, \mathsf{SIG.Sign}, \mathsf{SIG.Verify})$. The $\mathsf{SIG.Gen}$ algorithm takes 1^λ as input and outputs a verification/signing key pair $(\mathsf{vk}, \mathsf{sk})$. The $\mathsf{SIG.Sign}$ algorithm takes sk and a message m as input, and outputs a signature σ. The $\mathsf{SIG.Verify}$ algorithm takes vk, m, and σ as input, and outputs either 1 or 0. We say that a signature scheme is correct if for all $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$ and all messages m, it holds that $\Pr[\mathsf{SIG.Verify}(\mathsf{vk}, (m, \sigma)) = 1 \mid \sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{sk}, m)] = 1$. In our construction, we use a signature scheme which satisfies existential unforgeability against chosen message attacks (EUF-CMA security). Let $\mathrm{Exp}^{unforge}_{\mathcal{SIG},\mathcal{A}}(\lambda)$ be the experiment given in Figure 1. We say that $\mathcal{SIG}$ is EUF-CMA secure if the advantage $\mathrm{Adv}^{unforge}_{\mathcal{SIG},\mathcal{A}}(\lambda) = \Pr[\mathrm{Exp}^{unforge}_{\mathcal{SIG},\mathcal{A}}(\lambda) = 1]$ is negligible for any PPT adversary $\mathcal{A}$.

$\mathrm{Exp}^{unforge}_{\mathcal{SIG},\mathcal{A}}(\lambda)$: $\mathsf{ML} \leftarrow \emptyset$; $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$; $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\cdot)}(\mathsf{vk})$
Return 1 if $\mathsf{SIG.Verify}(\mathsf{vk}, m^*, \sigma^*) = 1 \ \wedge \ m^* \notin \mathsf{ML}$
else return 0

Figure 1: This is the experiment used to define EUF-CMA security for a signature scheme $\mathcal{SIG}$. Here, the oracle Sign takes a message m, computes $\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{sk}, m)$, adds m to the list ML, and returns σ.

One-time signature (OTS) is simply defined that the number of signing query is restricted by one. For the sake of clarity, we denote an OTS scheme $\mathcal{OTS} = (\mathsf{OTS.Gen}, \mathsf{OTS.Sign}, \mathsf{OTS.Verify})$. Moreover, the list ML preserves (m, σ) and the adversary wins if $(m^*, \sigma^*) \notin \mathsf{ML}$. We say that $\mathcal{OTS}$ is strong EUF-CMA secure if the advantage is negligible for any PPT adversary $\mathcal{A}$.

Identity-Based Signature. An identity-based signature (IBS) scheme $\mathcal{IBS}$ consists of four algorithms $(\mathsf{IBS.Gen}, \mathsf{IBS.Ext}, \mathsf{IBS.Sign}, \mathsf{IBS.Verify})$. The $\mathsf{IBS.Gen}$ algorithm takes 1^λ as input and outputs a public parameter $\mathsf{params}_{\mathsf{IBS}}$ and a master secret key $\mathsf{msk}_{\mathsf{IBS}}$. The $\mathsf{IBS.Ext}$ algorithm takes $\mathsf{params}_{\mathsf{IBS}}$, $\mathsf{msk}_{\mathsf{IBS}}$, and an arbitrary string $\mathsf{ID} \in \{0,1\}^*$ as input, and outputs a signing key $\mathsf{sk}_{\mathsf{ID}}$. The $\mathsf{IBS.Sign}$ algorithm takes $\mathsf{params}_{\mathsf{IBS}}$, $\mathsf{sk}_{\mathsf{ID}}$, ID, and a message m as input, and outputs a signature σ. The $\mathsf{IBS.Verify}$ algorithm takes $\mathsf{params}_{\mathsf{IBS}}$, an identity ID, m, and σ as input, and outputs either 1 or 0. We say that an identity-based signature scheme is correct if for all identities ID and messages m, it holds that $\Pr[\mathsf{IBS.Verify}(\mathsf{params}_{\mathsf{IBS}}, \mathsf{ID}, (m, \sigma)) = 1 \mid (\mathsf{params}_{\mathsf{IBS}}, \mathsf{msk}_{\mathsf{IBS}}) \leftarrow \mathsf{IBS.Gen}(1^\lambda); \mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{IBS.Ext}(\mathsf{params}_{\mathsf{IBS}}, \mathsf{msk}_{\mathsf{IBS}}, \mathsf{ID}); \sigma \leftarrow \mathsf{IBS.Sign}(\mathsf{params}_{\mathsf{IBS}}, \mathsf{sk}_{\mathsf{ID}}, m)] = 1$. In our construction, we use an IBS scheme which satisfies existential unforgeability against chosen message attacks (EUF-CMA security). Let $\mathrm{Exp}^{unforge}_{\mathcal{IBS},\mathcal{A}}(\lambda)$ be the experiment given in Figure 2. We say that $\mathcal{IBS}$ is EUF-CMA secure if the advantage $\mathrm{Adv}^{unforge}_{\mathcal{IBS},\mathcal{A}}(\lambda) = \Pr[\mathrm{Exp}^{unforge}_{\mathcal{IBS},\mathcal{A}}(\lambda) = 1]$ is negligible for any PPT adversary $\mathcal{A}$.

$\mathrm{Exp}^{unforge}_{\mathcal{IBS},\mathcal{A}}(\lambda)$: $\mathsf{IDSet}, \mathsf{ML} \leftarrow \emptyset$; $(\mathsf{params}_{\mathsf{IBS}}, \mathsf{msk}_{\mathsf{IBS}}) \leftarrow \mathsf{IBS.Gen}(1^\lambda)$; $(\mathsf{ID}^*, m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{IBS.Ext}(\cdot),\mathsf{IBS.Sign}(\cdot,\cdot)}(\mathsf{params}_{\mathsf{IBS}})$
Return 1 if $\mathsf{IBS.Verify}(\mathsf{params}_{\mathsf{IBS}}, \mathsf{ID}^*, m^*, \sigma^*) = 1 \ \wedge \ \mathsf{ID}^* \notin \mathsf{IDSet} \ \wedge \ (\mathsf{ID}^*, m^*) \notin \mathsf{ML}$
else return 0

Figure 2: This is the experiment used to define EUF-CMA security for an identity-based signature scheme $\mathcal{IBS}$. Here, the oracle $\mathsf{IBS.Ext}$ takes an identity ID, computes $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{IBS.Ext}(\mathsf{params}_{\mathsf{IBS}}, \mathsf{msk}_{\mathsf{IBS}}, \mathsf{ID})$ if $\mathsf{sk}_{\mathsf{ID}}$ is undefined, adds ID to the list IDSet, and returns $\mathsf{sk}_{\mathsf{ID}}$. The oracle $\mathsf{IBS.Sign}$ takes an identity ID and a message m, computes $\mathsf{sk}_{\mathsf{ID}} \leftarrow \mathsf{IBS.Ext}(\mathsf{params}_{\mathsf{IBS}}, \mathsf{msk}_{\mathsf{IBS}}, \mathsf{ID})$ if $\mathsf{sk}_{\mathsf{ID}}$ is undefined, computes $\sigma \leftarrow \mathsf{IBS.Sign}(\mathsf{params}_{\mathsf{IBS}}, \mathsf{sk}_{\mathsf{ID}}, \mathsf{ID}, m)$, adds (ID, m) to the list ML, and returns σ.

Public Key Encryption. A public key encryption (PKE) scheme $\mathcal{PKE}$ consists of three algorithms $(\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$. The $\mathsf{PKE.Gen}$ algorithm takes 1^λ as input and outputs an encryption/decryption key pair $(\mathsf{ek}, \mathsf{dk})$. The $\mathsf{PKE.Enc}$ algorithm takes ek and a plaintext m as input, and outputs a ciphertext ct. In this paper, if necessary, we explicitly mention a randomness $r \in \mathcal{R}_{\mathrm{PKE}}$ used in the encryption and write $\mathsf{ct} \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}, m; r)$ where $\mathcal{R}_{\mathrm{PKE}}$ is the randomness space of $\mathcal{PKE}$. The $\mathsf{PKE.Dec}$ algorithm takes dk and ct as input, and outputs m. We say that a PKE scheme is correct if for all plaintexts m and all randomness r, it holds that $\Pr[m = \widetilde{m} \mid (\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda); \ \widetilde{m} \leftarrow \mathsf{PKE.Dec}(\mathsf{dk}, \mathsf{PKE.Enc}(\mathsf{ek}, m; r))] = 1$. Let $\mathrm{Exp}^{ind\text{-}cpa}_{\mathcal{PKE},\mathcal{A}}(\lambda)$ and $\mathrm{Exp}^{key\text{-}priv}_{\mathcal{PKE},\mathcal{A}}(\lambda)$ be the experiments given in Figure 3. We say that $\mathcal{PKE}$ is indistinguishable against chosen plaintext attacks (IND-CPA secure) if the advantage $\mathrm{Adv}^{ind\text{-}cpa}_{\mathcal{PKE},\mathcal{A}}(\lambda) = \left|\Pr[\mathrm{Exp}^{ind\text{-}cpa}_{\mathcal{PKE},\mathcal{A}}(\lambda) = 1] - 1/2\right|$ is negligible for any PPT adversary $\mathcal{A}$, and is key-private if the advantage $\mathrm{Adv}^{key\text{-}priv}_{\mathcal{PKE},\mathcal{A}}(\lambda) = \left|\Pr[\mathrm{Exp}^{key\text{-}priv}_{\mathcal{PKE},\mathcal{A}}(\lambda) = 1] - 1/2\right|$ is negligible for any PPT adversary $\mathcal{A}$.

Identity-Based Encryption. An identity-based encryption (IBE) scheme $\mathcal{IBE}$ consists of four algorithms $(\mathsf{IBE.Gen}, \mathsf{IBE.Ext}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$. The $\mathsf{IBE.Gen}$ algorithm takes 1^λ as input and outputs system parameters params and a master secret key msk. The $\mathsf{IBE.Ext}$ algorithm takes params, msk, and an arbitrary string $\mathsf{ID} \in \{0,1\}^*$ as input, and outputs a decryption key dk that is the corresponding decryption key with the public key ID. The $\mathsf{IBE.Enc}$ algorithm takes params, ID, and a plaintext m as input, and outputs a ciphertext ct. As the case of PKE schemes, if necessary, we explicitly mention a randomness $r \in \mathcal{R}_{\mathrm{IBE}}$ used in the encryption and write $\mathsf{ct} \leftarrow \mathsf{IBE.Enc}(\mathsf{params}, \mathsf{ID}, m; r)$ where $\mathcal{R}_{\mathrm{IBE}}$

$\mathrm{Exp}^{\textit{ind-cpa}}_{\mathcal{PKE},\mathcal{A}}(\lambda)$: $(\mathsf{ek},\mathsf{dk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$; $(\mathsf{st}, m_0, m_1) \leftarrow \mathcal{A}_1(\mathsf{ek})$
$b \xleftarrow{\$} \{0,1\}$; $\mathsf{ct}^* \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}, m_b)$; $\widetilde{b} \leftarrow \mathcal{A}_2(\mathsf{st}, \mathsf{ct}^*)$
Return 1 if $b = \widetilde{b}$, otherwise return 0

$\mathrm{Exp}^{\textit{key-priv}}_{\mathcal{PKE},\mathcal{A}}(\lambda)$: $(\mathsf{ek}_0,\mathsf{dk}_0) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$; $(\mathsf{ek}_1,\mathsf{dk}_1) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$
$(\mathsf{st}, m^*) \leftarrow \mathcal{A}_1(\mathsf{ek}_0, \mathsf{ek}_1)$
$b \xleftarrow{\$} \{0,1\}$; $\mathsf{ct}^* \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}_b, m^*)$; $\widetilde{b} \leftarrow \mathcal{A}_2(\mathsf{st}, \mathsf{ct}^*)$
Return 1 if $b = \widetilde{b}$, otherwise return 0

Figure 3: These are the experiments used to define IND-CPA security and key privacy for a PKE scheme $\mathcal{PKE}$. The adversary $\mathcal{A}$ is restricted to output m_0 and m_1 satisfying $|m_0| = |m_1|$.

is the randomness space of $\mathcal{IBE}$. The $\mathsf{IBE.Dec}$ algorithm takes params, dk, and ct as input, and outputs m. We say that an IBE scheme is correct if for all strings ID, all plaintexts m, and all randomness r, it holds that $\Pr[m = \widetilde{m} \mid (\mathsf{params},\mathsf{msk}) \leftarrow \mathsf{IBE.Gen}(1^\lambda);\ \mathsf{dk} \leftarrow \mathsf{IBE.Ext}(\mathsf{params},\mathsf{msk},\mathsf{ID});\ \widetilde{m} \leftarrow \mathsf{IBE.Dec}(\mathsf{dk}, \mathsf{IBE.Enc}(\mathsf{params},\mathsf{ID}, m; r))] = 1$. Let $\mathrm{Exp}^{\textit{ind-id-cpa}}_{\mathcal{IBE},\mathcal{A}}(\lambda)$ and $\mathrm{Exp}^{\textit{key-priv}}_{\mathcal{IBE},\mathcal{A}}(\lambda)$ be the experiments given in Figure 4. We say that $\mathcal{IBE}$ is indistinguishable against chosen plaintext attacks (IND-ID-CPA secure) if the advantage $\mathrm{Adv}^{\textit{ind-id-cpa}}_{\mathcal{IBE},\mathcal{A}}(\lambda) = \left|\Pr[\mathrm{Exp}^{\textit{ind-id-cpa}}_{\mathcal{IBE},\mathcal{A}}(\lambda) = 1] - 1/2\right|$ is negligible for any PPT adversary $\mathcal{A}$, and is key-private if the advantage $\mathrm{Adv}^{\textit{key-priv}}_{\mathcal{IBE},\mathcal{A}}(\lambda) = \left|\Pr[\mathrm{Exp}^{\textit{key-priv}}_{\mathcal{IBE},\mathcal{A}}(\lambda) = 1] - 1/2\right|$ is negligible for any PPT adversary $\mathcal{A}$.

$\mathrm{Exp}^{\textit{ind-id-cpa}}_{\mathcal{IBE},\mathcal{A}}(\lambda)$: $\mathsf{IDSet} \leftarrow \emptyset$; $(\mathsf{params},\mathsf{msk}) \leftarrow \mathsf{IBE.Gen}(1^\lambda)$
$(\mathsf{st}, \mathsf{ID}^*, m_0, m_1) \leftarrow \mathcal{A}_1^{\mathsf{Extract}(\cdot)}(\mathsf{params})$
If $\mathsf{ID}^* \in \mathsf{IDSet}$, return 0
$b \xleftarrow{\$} \{0,1\}$; $\mathsf{ct}^* \leftarrow \mathsf{IBE.Enc}(\mathsf{params}, \mathsf{ID}^*, m_b)$; $\widetilde{b} \leftarrow \mathcal{A}_2^{\mathsf{Extract}(\cdot)}(\mathsf{st}, \mathsf{ct}^*)$
Return 1 if $b = \widetilde{b}$, otherwise return 0

$\mathrm{Exp}^{\textit{key-priv}}_{\mathcal{IBE},\mathcal{A}}(\lambda)$: $\mathsf{IDSet} \leftarrow \emptyset$; $(\mathsf{params},\mathsf{msk}) \leftarrow \mathsf{IBE.Gen}(1^\lambda)$
$(\mathsf{st}, \mathsf{ID}_0, \mathsf{ID}_1, m^*) \leftarrow \mathcal{A}_1^{\mathsf{Extract}(\cdot)}(\mathsf{params})$
If $\{\mathsf{ID}_0, \mathsf{ID}_1\} \cap \mathsf{IDSet} \neq \emptyset$, return 0
$b \xleftarrow{\$} \{0,1\}$; $\mathsf{ct}^* \leftarrow \mathsf{IBE.Enc}(\mathsf{params}, \mathsf{ID}_b, m^*)$; $\widetilde{b} \leftarrow \mathcal{A}_2^{\mathsf{Extract}(\cdot)}(\mathsf{st}, \mathsf{ct}^*)$
Return 1 if $b = \widetilde{b}$, otherwise return 0

Figure 4: These are the experiments used to define IND-ID-CPA security and key privacy for an IBE scheme $\mathcal{IBE}$. Here, the oracle $\mathsf{Extract}$ takes ID, computes $\mathsf{dk} \leftarrow \mathsf{IBE.Ext}(\mathsf{params},\mathsf{msk},\mathsf{ID})$, adds ID to the list IDSet, and returns dk. We note that it is not allowed to query the identity ID^*, and the identities ID_0 and ID_1 to the $\mathsf{Extract}$ oracle in the experiment $\mathrm{Exp}^{\textit{ind-id-cpa}}_{\mathcal{IBE},\mathcal{A}}(\lambda)$ and $\mathrm{Exp}^{\textit{key-priv}}_{\mathcal{IBE},\mathcal{A}}(\lambda)$, respectively. The adversary $\mathcal{A}$ is restricted to output m_0 and m_1 satisfying $|m_0| = |m_1|$.

Non-interactive Zero-knowledge (NIZK) Proof. Let R_L be an efficiently computable binary relation. For a pair $(x, w) \in R_L$, we call x a statement and w a witness. Let L be the language consisting of statements in R_L. An NIZK proof system $\mathcal{P}_L$ for a language L consists of three algorithms ($\mathsf{ZK.Gen}$, $\mathsf{ZK.Prove}$, $\mathsf{ZK.Verify}$). The $\mathsf{ZK.Gen}$ algorithm takes 1^λ as input and returns a common reference string crs. The $\mathsf{ZK.Prove}$ algorithm takes crs, a statement x, and a witness w as input, and outputs a proof π. The $\mathsf{ZK.Verify}$ algorithm takes crs, x, and π as input, and outputs either 1 or 0. An NIZK proof system is required the following two conditions:

- **Completeness**: For all $(x, w) \in R_L$ and all $\mathsf{crs} \leftarrow \mathsf{ZK.Gen}(1^\lambda)$, $\Pr[\mathsf{ZK.Verify}(\mathsf{crs}, x, \pi) = 1 \mid \pi \leftarrow \mathsf{ZK.Prove}(\mathsf{crs}, x, w)] = 1$ holds.
- **Soundness**: For any PPT adversary $\mathcal{A}$, the advantage $\mathrm{Adv}^{\textit{sound}}_{\mathcal{P}_L,\mathcal{A}}(\lambda) = \Pr[x^* \notin L \wedge \mathsf{ZK.Verify}(\mathsf{crs}, x^*, \pi^*) = 1 \mid \mathsf{crs} \leftarrow \mathsf{ZK.Gen}(1^\lambda); (x^*, \pi^*) \leftarrow \mathcal{A}(\mathsf{crs})]$ is negligible.

Moreover, we say that $\mathcal{P}_L$ is zero-knowledge if for any PPT adversary $\mathcal{A}$ there exists a simulator $\mathcal{S} = (\mathtt{Sim}_1, \mathtt{Sim}_2)$ such that the advantage $\mathrm{Adv}^{zk}_{\mathcal{P}_L,\mathcal{A}}(\lambda) = \left|\Pr[\mathrm{Exp}^{\textit{proof}}_{\mathcal{P}_L,\mathcal{A}}(\lambda) = 1\ ] - \Pr[\ \mathrm{Exp}^{\textit{sim-proof}}_{\mathcal{P}_L,\mathcal{A}}(\lambda) = 1]\right|$ is negligible where the experiments $\mathrm{Exp}^{\text{proof}}_{\mathcal{P}_L,\mathcal{A}}(\lambda)$ and $\mathrm{Exp}^{\text{sim-proof}}_{\mathcal{P}_L,\mathcal{A}}(\lambda)$ are defined in Figure 5.

$\mathrm{Exp}^{proof}_{\mathcal{P}_L,\mathcal{A}}(\lambda)$:	$\mathsf{crs} \leftarrow \mathsf{ZK.Gen}(1^\lambda)$	$\mathrm{Exp}^{sim\text{-}proof}_{\mathcal{P}_L,\mathcal{A}}(\lambda)$:	$(\mathsf{crs}, \mathsf{td}) \leftarrow \mathtt{Sim}_1(1^\lambda)$
	$b \leftarrow \mathcal{A}^{\mathsf{Prove}(\cdot,\cdot)}(\mathsf{crs})$		$b \leftarrow \mathcal{A}^{\mathsf{SimProve}(\cdot,\cdot)}(\mathsf{crs})$
	Return b		Return b

Figure 5: These are the experiments used to define zero-knowledgeness for an NIZK proof system $\mathcal{P}_L$. Here, the oracle Prove takes (x, w), computes $\pi \leftarrow \mathsf{ZK.Prove}(\mathsf{crs}, x, w)$, and returns π. The oracle $\mathsf{SimProve}$ takes (x, w), computes $\pi \leftarrow \mathtt{Sim}_2(\mathsf{crs}, \mathsf{td}, x)$, and returns π. If $(x, w) \notin R_L$, then $\mathsf{SimProve}$ returns $\perp$.

2.2 Group Signature with Verifier-local Revocation

In this section, we review the syntax and the security requirements of group signature with verifier-local revocation (VLR-GS). We give the model of VLR-GS *with* backward unlinkability [35], which is extended from that of VLR-GS *without* backward unlinkability [9]. A VLR-GS scheme without backward unlinkability is a special case of that with backward unlinkability where the number of time periods is only one. A VLR-GS scheme $\mathcal{GS}$ consists of the following three algorithms $(\mathsf{GS.Gen}, \mathsf{GS.Sign}, \mathsf{GS.Verify})$.

$\mathsf{GS.Gen}$**:** The group key generation algorithm takes a security parameter 1^λ ($\lambda \in \mathbb{N}$), the number of users n, and the number of time periods T as input, and outputs a group public key gpk, a set of user signing keys $\mathbf{gsk} = \{\mathsf{gsk}[i]\}_i$, and a set of revocation tokens $\mathbf{grt} = \{\mathsf{grt}[i][j]\}_{ij}$. Here, $\mathsf{gsk}[i]$ and $\mathsf{grt}[i][j]$ denote the signing key of the user $i \in [1, n]$ and revocation token at the time period $j \in [1, T]$, respectively.

$\mathsf{GS.Sign}$**:** The signing algorithm takes gpk, time period j, $\mathsf{gsk}[i]$, and a message m as input, and outputs a signature Σ.

$\mathsf{GS.Verify}$**:** The verification algorithm takes gpk, j, a revocation list RL_j, m, and Σ as input, and outputs either 1 or 0. The list RL_j is defined as the set of the revocation tokens $\mathsf{RL}_j = \{\mathsf{grt}[i][j] \mid i \in \mathsf{RU}_j\}$ where RU_j is the set of the revoked users' identities at the time period j.

We assume that $\mathsf{RU}_j \subseteq \mathsf{RU}_{j+1}$ for $j \in [1, T-1]$, i.e., once a user is revoked at j, the user will be kept revoked from this time onward. In a VLR-GS scheme, the opening procedure can be done by using a set of revocation tokens $\mathbf{grt}$. More precisely, the implicit opening algorithm $\mathsf{GS.Open}$ can be defined as follows.

$\mathsf{GS.Open}$**:** The opening algorithm takes gpk, j, a set of revocation tokens $\mathbf{grt}$, m, and Σ as input, and executes the following procedures:

[Step 1] Set the revocation list $\mathsf{RL}_j = \emptyset$, and output $\perp$ if $\mathsf{GS.Verify}(\mathsf{gpk}, j, \mathsf{RL}_j, m, \Sigma) = 0$.

[Step 2] For $1 \le i \le n$, set the revocation list $\mathsf{RL}_j = \{\mathsf{grt}[i][j]\}$, and run $\mathsf{GS.Verify}(\mathsf{gpk}, j, \mathsf{RL}_j, m, \Sigma)$.

[Step 3] Let i be the index that the $\mathsf{GS.Verify}$ algorithm outputs 0 for the first time in Step 2. Then, output i. If there does not exist such an index, output $\perp$.

In the following, we define the security requirements, correctness, full anonymity, and traceability. Full anonymity is an extended notion of selfless anonymity [35].

Definition 2.1 (Correctness)**.** *Let $\mathcal{A}$ be an adversary for the correctness. We define the experiment $\mathrm{Exp}^{corr}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T)$ as follows.*

$$
\begin{aligned}
\mathrm{Exp}^{corr}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T): \ & (\mathsf{gpk}, \mathbf{gsk}, \mathbf{grt}) \leftarrow \mathsf{GS.Gen}(1^\lambda, n, T) \\
& (i^*, j^*, m^*, \mathsf{RU}^*) \leftarrow \mathcal{A}(\mathsf{gpk}) \\
& \text{If } i \in \mathsf{RU}^*, \text{ return } 0 \\
& \mathsf{RL}_{j^*} := \{\mathsf{grt}[i][j^*] \mid i \in \mathsf{RU}^*\};\ \Sigma^* \leftarrow \mathsf{GS.Sign}(\mathsf{gpk}, j^*, \mathsf{gsk}[i^*], m^*) \\
& \text{Return 1 if } \mathsf{GS.Verify}(\mathsf{gpk}, j^*, \mathsf{RL}_{j^*}, m^*, \Sigma^*) = 0, \text{ else return } 0
\end{aligned}
$$

We say that $\mathcal{GS}$ is correct if the advantage $\mathrm{Adv}^{corr}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T) = \Pr[\mathrm{Exp}^{corr}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T) = 1]$ is negligible for any PPT adversary $\mathcal{A}$.

Definition 2.2 (Full Anonymity). *Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary for full anonymity. We define the experiment $\mathrm{Exp}^{anon}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T)$ as follows.*

$$
\begin{aligned}
\mathrm{Exp}^{anon}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T):\ & \mathsf{RU}_j \leftarrow \emptyset;\ (\mathsf{gpk}, \mathbf{gsk}, \mathbf{grt}) \leftarrow \mathsf{GS.Gen}(1^\lambda, n) \\
& (\mathsf{st}, i_0, i_1, j^*, m^*) \leftarrow \mathcal{A}_1^{\mathsf{Revoke}(\cdot,\cdot)}(\mathsf{gpk}, \mathbf{gsk}) \\
& \text{If } i_0 \in \mathsf{RU}_{j^*} \vee i_1 \in \mathsf{RU}_{j^*},\ \text{return } 0 \\
& b \xleftarrow{\$} \{0,1\};\ \Sigma^* \leftarrow \mathsf{GS.Sign}(\mathsf{gpk}, j^*, \mathsf{gsk}[i_b], m^*) \\
& \widetilde{b} \leftarrow \mathcal{A}_2^{\mathsf{Revoke}(\cdot,\cdot)}(\mathsf{st}, \Sigma^*) \\
& \text{Return 1 if } b = \widetilde{b},\ \text{else return } 0
\end{aligned}
$$

Here, the oracle Revoke *takes $i \in [1, n]$ and $j \in [1, T]$, adds i to the list RU_j, and returns $\mathsf{grt}[i][j]$. We note that it is not allowed to query (i_0, j^*) and (i_1, j^*) to the* Revoke *oracle. We say that $\mathcal{GS}$ satisfies full anonymity if the advantage $\mathrm{Adv}^{anon}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T) = \left|\Pr[\mathrm{Exp}^{anon}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T) = 1] - 1/2\right|$ is negligible for any polynomial $n = n(\lambda)$ and $T = T(\lambda)$, and any PPT adversary $\mathcal{A}$.*

Definition 2.3 (Traceability). *Let $\mathcal{A}$ be an adversary for the traceability. We define the experiment $\mathrm{Exp}^{trace}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T)$ as follows.*

$$
\begin{aligned}
\mathrm{Exp}^{trace}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T):\ & \mathsf{CU} \leftarrow \emptyset;\ \mathsf{QL} \leftarrow \emptyset;\ (\mathsf{gpk}, \mathbf{gsk}, \mathbf{grt}) \leftarrow \mathsf{GS.Gen}(1^\lambda, n, T) \\
& (j^*, m^*, \Sigma^*, \mathsf{RU}^*) \leftarrow \mathcal{A}^{\mathsf{GS.Sign}(\cdot,\cdot,\cdot),\mathsf{Corrupt}(\cdot)}(\mathsf{gpk}, \mathbf{grt}) \\
& \mathsf{RL}^* := \{\mathsf{grt}[i][j^*] \mid i \in \mathsf{RU}^*\} \\
& i^* \leftarrow \mathsf{GS.Open}(\mathsf{gpk}, j^*, \mathbf{grt}, m^*, \Sigma^*) \\
& \text{Return 1 if all of the following holds, else return } 0 \\
& \quad \mathsf{GS.Verify}(\mathsf{gpk}, j^*, \mathsf{RL}^*, m^*, \Sigma^*) = 1 \\
& \quad i^* = \perp \ \vee\ i^* \notin \mathsf{CU} \ \vee\ i^* \in \mathsf{RU}^* \\
& \quad (i^*, j^*, m^*, \Sigma^*) \notin \mathsf{QL}
\end{aligned}
$$

Here, the oracle GS.Sign *takes (i, j, m), computes $\Sigma \leftarrow \mathsf{GS.Sign}(\mathsf{gpk}, j, \mathsf{gsk}[i], m)$, adds (i, j, m, Σ) to the list* QL*, and returns Σ. The oracle* Corrupt *takes $i \in [1, n]$, adds i to the list* CU*, and returns $\mathsf{gsk}[i]$. We say that $\mathcal{GS}$ satisfies traceability if the advantage $\mathrm{Adv}^{trace}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T) = \Pr[\mathrm{Exp}^{trace}_{\mathcal{GS},\mathcal{A}}(\lambda, n, T) = 1]$ is negligible for any polynomial $n = n(\lambda)$ and $T = T(\lambda)$, and any PPT adversary $\mathcal{A}$.*

Remark 1. Our scheme seems to be relatively easy to extend to CCA security. If a tag-based encryption scheme is deployed and an OTS verification key is regarded as a tag, the group signature scheme becomes non-malleable, that is, it satisfies CCA anonymity. Moreover, to achieve dynamic setting in the sense of the Bellare-Shi-Zhang (BSZ) model [6], a user also generates a part of signing key that the issuer does not know. Each technique is standard and widely used, for example, in the papers [29, 27].

3 A Fully Anonymous VLR-GS Scheme

In this section, we give a construction of a fully anonymous VLR-GS scheme (for $T = 1$). Concretely, we construct a VLR-GS scheme from a digital signature scheme, an OTS scheme, a key-private PKE scheme, and an NIZK proof system. Here, there is only one time period $j = 1$, thus, we do not specify the time period and omit it. Here, we assume that once $\mathsf{grt}[i]$ is contained in RU, $\mathsf{grt}[i]$ is not removed from RU.

As mentioned, all existing schemes [9, 35, 36, 50, 54, 12, 26, 31] only provide selfless anonymity regardless of whether or not the scheme has backward unlinkability. Specifically, there is an attack against the full anonymity for most of the schemes [9, 35, 36, 50, 54, 12, 26] due to their structure allowing the revocation token to be constructed from the user's signing key. Therefore, in order to achieve full anonymity, a VLR-GS scheme must not have such a structure provided the revocation token and signing key of the same user have some relation.

Intuitively, we achieve this by employing an encryption/decryption key pair of a PKE scheme as a part of the user signing key and the revocation token. In the following, we explain the proposed VLR-GS scheme without backward unlinkability in detail, which we call Scheme 1. Before describing the construction, we give the high-level idea of this scheme.

3.1 High Level Idea

Scheme 1 mainly follows the BMW construction [5], which allows us to construct a fully anonymous group signature scheme from a digital signature scheme, a PKE scheme, and an NIZK proof system. Now, we review the BMW construction.

In the BMW construction, the group manager possesses a key pair $(\mathsf{vk}_{\mathsf{SIG}}, \mathsf{sk}_{\mathsf{SIG}})$ of a digital signature scheme and a key pair $(\mathsf{ek}_{\mathsf{PKE}}, \mathsf{dk}_{\mathsf{PKE}})$ of a PKE scheme. Each user possesses a key pair $(\mathsf{vk}_i, \mathsf{sk}_i)$ of a digital signature scheme and its certificate cert_i given by the manager where cert_i is the signature of the verification key vk_i under the signing key $\mathsf{sk}_{\mathsf{SIG}}$. When a user i signs a message m, the user generates an internal signature σ on the message m using his signing key sk_i, and encrypts σ using $\mathsf{ek}_{\mathsf{PKE}}$ along with the verification key vk_i and the corresponding certificate cert_i. Let ct be this ciphertext. Moreover, the user produces an NIZK proof π which proves that the whole procedure is honestly done and the encrypted certificate cert_i is a valid signature on vk_i. Thus, the signature Σ in the BMW construction consists of a ciphertext ct and a proof π. The full anonymity is ensured by the IND-CPA security of the underlying PKE scheme and the zero-knowledgeness of the underlying NIZK proof system. The traceability is ensured by the EUF-CMA security of the underlying digital scheme and the soundness of the underlying NIZK proof system.

We add revocation functionality by introducing additional key pairs of a key-private PKE scheme to the BMW construction. In our construction, the manager generates an encryption/decryption key pair $(\mathsf{ek}_i, \mathsf{dk}_i)$ for each user i and sends only the encryption key ek_i as a part of the signing key to the user. In addition, the manager sets the decryption key dk_i as the revocation token of the user i. To certify that the key ek_i is generated for a user i by the manager, he also computes a signature cert_i on the message $\langle \mathsf{ek}_i, \mathsf{vk}_i \rangle$ under the signing key $\mathsf{sk}_{\mathsf{SIG}}$ as a certificate. Unlike the BMW construction, when signing a message m, a user i generates an OTS key pair $(\mathsf{vk}_{\mathrm{ots}}, \mathsf{sk}_{\mathrm{ots}}) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$ and generates an internal signature σ on the message $\mathsf{vk}_{\mathrm{ots}}$ using the signing key sk_i. The reason of this replacement is to achieve strong unforgeability which is captured by traceability in the VLR setting while only the usual unforgeability is required by traceability in the BMW model. As in the BMW model construction, the user encrypts σ, $\langle \mathsf{ek}_i, \mathsf{vk}_i \rangle$, and cert_i under $\mathsf{ek}_{\mathsf{PKE}}$.

Moreover, in our construction, the signer i generates a ciphertext $\widetilde{\mathsf{ct}}$ which is the encryption of the same plaintext $\langle \sigma, \mathsf{ek}_i, \mathsf{vk}_i, \mathsf{cert}_i \rangle$ as the ciphertext ct under the encryption key ek_i.[6] Then, the user produces an NIZK proof π which proves that the whole procedure is honestly done and cert_i is a valid signature on $\langle \mathsf{ek}_i, \mathsf{vk}_i \rangle$, in the case of the BMW construction. That is, the signature Σ in our construction consists of ciphertexts ct and $\widetilde{\mathsf{ct}}$, and a proof π. An OTS is also added for providing strong unforgeability. We remark that we do not have to introduce a tag-based PKE scheme since we do not consider CCA anonymity in this paper.

Our scheme does not have a structure allowing the revocation token to be computed from the corresponding signing key since it is hard to compute the decryption key dk_i even if knowing the corresponding encryption key ek_i because of the security of the underlying PKE scheme. The decryption key dk_i works as a revocation token as follows. If a user i is revoked, his/her revocation token $\mathsf{grt}[i] = \mathsf{dk}_i$ is listed in the revocation list RL. If a verifier checks whether the ciphertext $\widetilde{\mathsf{ct}}$ can be decrypted by each element in RL as the decryption key, the verifier can check whether the signer is a revoked user.

The security of our scheme can be discussed in almost the same way as the BMW construction. However, the underlying PKE scheme is required to be key-private in our construction since the ciphertext $\widetilde{\mathsf{ct}}$ is computed by the encryption key ek_i depending on the signer i. The full anonymity is ensured by the IND-CPA security and the key privacy of the underlying PKE scheme, and the zero-knowledgeness of the underlying NIZK proof system. The traceability is ensured by the EUF-CMA security of the underlying digital scheme and the soundness of the underlying NIZK proof system. Also, note that we can rule out the possibility that the ciphertext $\widetilde{\mathsf{ct}}$ decrypts to the same message σ under two different decription keys since the encryption key ek_i is bound by the verification key vk_i with the certificate cert_i. Therefore, we do not require the underlying PKE scheme to be robust [1].

[6] A reader might think that the ciphertext ct is redundant and it is enough that the ciphertext ct is replaced with the ciphertext $\widetilde{\mathsf{ct}}$. However, if so, it is difficult to reduce its traceability to the EUF-CMA security of the underlying digital signature scheme. More precisely, if an adversary uses an uncertified encryption key to generate $\widetilde{\mathsf{ct}}$, the reduction algorithm cannot extract a forgery of the digital signature scheme. Also, it is not necessary to encrypt the whole value $\langle \sigma, \mathsf{ek}_i, \mathsf{vk}_i, \mathsf{cert}_i \rangle$ in both ct and $\widetilde{\mathsf{ct}}$. Therefore, part of the value is encrypted in the ciphertexts in our scheme described in Section 3.2.

3.2 Description

Scheme 1 is given in Figure 6. We construct a VLR-GS scheme $\Pi_1 = (\mathsf{GS.Gen}, \mathsf{GS.Sign}, \mathsf{GS.Verify})$ from a digital signature scheme $\mathcal{SIG} = (\mathsf{SIG.Gen}, \mathsf{SIG.Sign}, \mathsf{SIG.Verify})$, an OTS scheme $\mathcal{OTS} = (\mathsf{OTS.Gen}, \mathsf{OTS.Sign}, \mathsf{OTS.Verify})$, a PKE scheme $\mathcal{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$, and an NIZK proof system $\mathcal{P}_L = (\mathsf{ZK.Gen}, \mathsf{ZK.Prove}, \mathsf{ZK.Verify})$. We say that a statement $x = \langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct} \rangle$ and a witness $w = \langle \mathsf{ek}_i, \mathsf{vk}_i, \mathsf{cert}_i, \sigma, r_1, r_2 \rangle$ satisfy the relation R_L if the following equations hold:

(a) $\widetilde{\mathsf{ct}} = \mathsf{PKE.Enc}(\mathsf{ek}_i, \sigma; r_1)$,

(b) $\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{ek}_{\mathsf{PKE}}, \langle \mathsf{ek}_i, \mathsf{vk}_i, \mathsf{cert}_i \rangle; r_2)$,

(c) $\mathsf{SIG.Verify}(\mathsf{vk}_{\mathsf{SIG}}, \langle \mathsf{ek}_i, \mathsf{vk}_i \rangle, \mathsf{cert}_i) = 1$,

(d) $\mathsf{SIG.Verify}(\mathsf{vk}_i, \mathsf{vk}_{\mathrm{ots}}, \sigma) = 1$.

Moreover, for a statement $x = \langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct} \rangle$, if there exists a witness that satisfies the above equations, then we say that the statement x belongs to the language L and denote it $x \in L$.

$\mathsf{GS.Gen}(1^\lambda, n)$:
  $\mathsf{crs} \leftarrow \mathsf{ZK.Gen}(1^\lambda)$; $(\mathsf{vk}_{\mathsf{SIG}}, \mathsf{sk}_{\mathsf{SIG}}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$
  $(\mathsf{ek}_{\mathsf{PKE}}, \mathsf{dk}_{\mathsf{PKE}}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$
  For $1 \le i \le n$:
    $(\mathsf{ek}_i, \mathsf{dk}_i) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$; $(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$
    $\mathsf{cert}_i \leftarrow \mathsf{SIG.Sign}(\mathsf{sk}_{\mathsf{SIG}}, \langle \mathsf{ek}_i, \mathsf{vk}_i \rangle)$; $\mathsf{grt}[i] \leftarrow (\mathsf{dk}_i, \mathsf{vk}_i)$
  $\mathsf{gpk} = (\mathsf{crs}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{ek}_{\mathsf{PKE}})$; $\mathsf{gsk}[i] = (\mathsf{ek}_i, \mathsf{vk}_i, \mathsf{sk}_i, \mathsf{cert}_i)$
  $\mathbf{gsk} = \{\mathsf{gsk}[i]\}_i$; $\mathbf{grt} = \{\mathsf{grt}[i]\}_i$
  Return $(\mathsf{gpk}, \mathbf{gsk}, \mathbf{grt})$

$\mathsf{GS.Sign}(\mathsf{gpk}, \mathsf{gsk}[i], m)$:
  $(\mathsf{vk}_{\mathrm{ots}}, \mathsf{sk}_{\mathrm{ots}}) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$
  $\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{sk}_i, \mathsf{vk}_{\mathrm{ots}})$
  $\widetilde{\mathsf{ct}} \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}_i, \sigma; r_1)$
  $\mathsf{ct} \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}_{\mathsf{PKE}}, \langle \mathsf{ek}_i, \mathsf{vk}_i, \mathsf{cert}_i \rangle; r_2)$
  $\pi \leftarrow \mathsf{ZK.Prove}(\mathsf{crs}, \langle \mathsf{gpk}, \widetilde{\mathsf{ct}}, \mathsf{ct} \rangle, \langle \mathsf{ek}_i, \mathsf{vk}_i, \mathsf{cert}_i, \sigma, r_1, r_2 \rangle)$
  $\sigma_{all} \leftarrow \mathsf{OTS.Sign}(\mathsf{sk}_{\mathrm{ots}}, \langle m, \mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi \rangle)$
  Return $\Sigma = (\mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi, \sigma_{all})$

$\mathsf{GS.Verify}(\mathsf{gpk}, \mathsf{RL}, m, \Sigma)$:
  If $\mathsf{OTS.Verify}(\mathsf{vk}_{\mathrm{ots}}, \langle m, \mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi \rangle, \sigma_{all}) = 0$, return 0
  If $\mathsf{ZK.Verify}(\mathsf{crs}, \langle \mathsf{gpk}, \widetilde{\mathsf{ct}}, \mathsf{ct} \rangle, \pi) = 0$, return 0
  For $(\mathsf{dk}, \mathsf{vk}) \in \mathsf{RL}$:
    If $\mathsf{SIG.Verify}(\mathsf{vk}, \mathsf{vk}_{\mathrm{ots}}, \mathsf{PKE.Dec}(\mathsf{dk}, \widetilde{\mathsf{ct}})) = 1$, return 0
  Return 1

Figure 6: Scheme 1: A VLR-GS Scheme without Backward Unlinkability

For the correctness of Scheme 1, the following theorem holds.

Theorem 3.1. *Scheme 1 is correct if the underlying OTS scheme $\mathcal{OTS}$ satisfies correctness, the underlying NIZK proof system $\mathcal{P}_L$ satisfies completeness, and the underlying digital signature scheme $\mathcal{SIG}$ satisfies EUF-CMA security.*

Proof. Let $\mathcal{A}$ be an adversary for the correctness of Π_1 and the output of $\mathcal{A}$ in the experiment $\mathrm{Exp}^{corr}_{\Pi_1,\mathcal{A}}(\lambda, n)$ be $(i^*, j^*, m^*, \mathsf{RU}^*)$. We note that now the number of time periods satisfies $T = 1$, then it holds that $j^* = 1$. Therefore, we do not specify the time period j^* as in the description of Scheme 1. If the experiment $\mathrm{Exp}^{corr}_{\Pi_1,\mathcal{A}}(\lambda, n)$ outputs 1, $\mathsf{GS.Verify}(\mathsf{gpk}, \mathsf{RL}, m^*, \Sigma^*) = 0$ and $i^* \notin \mathsf{RU}^*$ hold where $\mathsf{RL} = \{\mathsf{grt}[i] \mid i \in \mathsf{RU}^*\}$ and $\Sigma^* \leftarrow \mathsf{GS.Sign}(\mathsf{gpk}, \mathsf{gsk}[i^*], m^*)$. Let $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$.

From the definition of the $\mathsf{GS.Verify}$ algorithm, one of the event $\mathtt{E}_A$, the event $\mathtt{E}_B$, or the event $\mathtt{E}_C$ happens when $\mathsf{GS.Verify}(\mathsf{gpk}, \mathsf{RL}, m^*, \Sigma^*) = 0$ holds.

$\mathtt{E}_A$: $\mathsf{OTS.Verify}(\mathsf{vk}^*_{\mathrm{ots}}, \langle m^*, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^* \rangle, \sigma^*_{all}) = 0$ holds.

$\mathtt{E}_B$: $\mathsf{ZK.Verify}(\mathsf{crs}, \langle \mathsf{gpk}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle, \pi^*) = 0$ holds.

$\mathtt{E}_C$: For some $i \in \mathsf{RU}^*$, $\mathsf{SIG.Verify}(\mathsf{vk}_i, \mathsf{vk}^*_{\mathrm{ots}}, \mathsf{PKE.Dec}(\mathsf{dk}_i, \widetilde{\mathsf{ct}}^*)) = 1$ holds.

However, $\Pr[\mathtt{E}_A] = 0$ holds if $\mathcal{OTS}$ satisfies correctness, and $\Pr[\mathtt{E}_B] = 0$ holds if $\mathcal{P}_L$ satisfies completeness. Therefore, it holds that $\Pr[\mathrm{Exp}^{corr}_{\Pi_1,\mathcal{A}}(\lambda, n) = 1] = \Pr[\mathtt{E}_A \vee \mathtt{E}_B \vee \mathtt{E}_C] \leq \Pr[\mathtt{E}_A] + \Pr[\mathtt{E}_B] + \Pr[\mathtt{E}_C] = \Pr[\mathtt{E}_C]$.

We evaluate $\Pr[\mathtt{E}_C]$ by constructing an algorithm $\mathcal{B}$ that breaks the EUF-CMA security of the digital signature scheme $\mathcal{SIG}$. At the beginning of the game, $\mathcal{B}$ randomly chooses $\widehat{i} \in [1, n]$, and sets $\mathsf{vk}_{\widehat{i}} \leftarrow \mathsf{vk}$ where vk is the key given by the challenger of the EUF-CMA security game. $\mathcal{B}$ generates the rest of instance for the scheme Π_1 and sends $\mathsf{gpk} = (\mathsf{crs}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{ek}_{\mathsf{PKE}})$ to $\mathcal{A}$. For $\mathcal{A}$'s output $(i^*, m^*, \mathsf{RU}^*)$, $\mathcal{B}$ outputs $\perp$ if $\widehat{i} = i^*$. Otherwise, if $\widehat{i} \neq i^*$, $\mathcal{B}$ computes $\Sigma^* \leftarrow \mathsf{GS.Sign}(\mathsf{gpk}, \mathsf{gsk}[i^*], m^*)$. Then, $\mathcal{B}$ computes $\sigma^* \leftarrow \mathsf{PKE.Dec}(\mathsf{dk}_{i^*}, \widetilde{\mathsf{ct}}^*)$, and outputs (m^*, σ^*) as a forged signature.

When the event $\mathtt{E}_C$ happens, there exists at least one pair $(\mathsf{dk}_i, \mathsf{vk}_i) \in \mathsf{RL}$ such that $\mathsf{SIG.Verify}(\mathsf{vk}_i, m^*, \mathsf{PKE.Dec}(\mathsf{dk}_i, \widetilde{\mathsf{ct}})) = 1$ holds. Let I be the set of such indexes i and $\mathtt{Good}$ be the event that $\widehat{i} \in I$ holds where $\widehat{i}$ is the index chosen by $\mathcal{B}$ at the beginning of the game. Since the guess of $\widehat{i} \in [1, n]$ and the behavior of $\mathcal{A}$ are independent, we get $\Pr[\mathtt{E}_C \wedge \mathtt{Good}] = \Pr[\mathtt{E}_C] \cdot \Pr[\mathtt{Good}]$. When both events $\mathtt{E}_C$ and $\mathtt{Good}$ happen, it holds that $\mathsf{SIG.Verify}(\mathsf{vk}_{\widehat{i}}, \mathsf{vk}^*_{\mathrm{ots}}, \sigma^*) = 1$ where $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$ and $\sigma^* \leftarrow \mathsf{PKE.Dec}(\mathsf{dk}_{\widehat{i}}, \widetilde{\mathsf{ct}})$. Therefore, (m^*, σ^*) is a forgery of the digital signature scheme $\mathcal{SIG}$, and $\Pr[\mathtt{E}_C \wedge \mathtt{Good}] \leq \mathrm{Adv}^{unforge}_{\mathcal{SIG},\mathcal{B}}(\lambda)$ holds. Moreover, since $\widehat{i} \in [1, n]$ is randomly chosen, we get $\Pr[\mathtt{Good}] = 1/n$. Putting all together, we have $\mathrm{Adv}^{corr}_{\Pi_1,\mathcal{A}}(\lambda, n) = \Pr[\mathrm{Exp}^{corr}_{\Pi_1,\mathcal{A}}(\lambda, n) = 1] \leq \Pr[\mathtt{E}_C] = (1/\Pr[\mathtt{Good}]) \cdot \Pr[\mathtt{E}_C \wedge \mathtt{Good}] \leq n \cdot \mathrm{Adv}^{unforge}_{\mathcal{SIG},\mathcal{B}}(\lambda)$. Therefore, Π_1 is correct if the underlying OTS scheme $\mathcal{OTS}$ satisfies correctness, the NIZK proof system $\mathcal{P}_L$ satisfies completeness, and the digital signature scheme $\mathcal{SIG}$ satisfies EUF-CMA security. □

3.3 Security Analysis

Here, we discuss the security of Scheme 1. That is, we explain that Scheme 1 satisfies full anonymity and traceability defined in Section 2.2.

Full Anonymity. For a signature $\Sigma = (\mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi, \sigma_{all})$ of Scheme 1, the user's information is contained in the encryption key ek_i and the plaintext σ of the ciphertext $\widetilde{\mathsf{ct}}$, the plaintext $\langle \mathsf{ek}_i, \mathsf{vk}_i, \mathsf{cert}_i \rangle$ of the ciphertext ct, and the witness of the proof π. Intuitively, the information of the plaintexts σ and $\langle \mathsf{ek}_i, \mathsf{vk}_i, \mathsf{cert}_i \rangle$ is not revealed from the ciphertexts $\widetilde{\mathsf{ct}}$ and ct since the underlying PKE scheme is IND-CPA secure. Also, the information of the encryption key ek_i is not revealed from $\widetilde{\mathsf{ct}}$ by the key privacy of the underlying PKE scheme. Moreover, the information of the witness is not revealed from the proof π since the NIZK proof system $\mathcal{P}_L$ is zero-knowledge. Since the user ' s information is hidden from the adversary who has the corresponding signing key $(\mathsf{ek}_i, \mathsf{vk}_i, \mathsf{sk}_i, \mathsf{cert}_i)$, Scheme 1 satisfies full anonymity. Formally, the following theorem holds.

Theorem 3.2. *Scheme 1 satisfies full anonymity if the underlying NIZK proof system $\mathcal{P}_L$ satisfies zero-knowledgeness and the underlying PKE scheme $\mathcal{PKE}$ satisfies IND-CPA security and key privacy.*

Proof. Let $\mathcal{A}$ be an adversary for full anonymity of Π_1. We consider the following sequence of games. Let $\Pr[\mathtt{Suc}_\ell]$ denote the event that $\mathcal{A}$ succeeds in guessing the challenge bit in Game ℓ. Let b be the challenge bit, i_0 and i_1 be the challenge users, and m^* be the challenge message.

Game 0: This is the experiment $\mathrm{Exp}^{anon}_{\Pi_1,\mathcal{A}}(\lambda, n)$ itself. For simplicity, the challenge bit b is chosen at the beginning of the game. This change does not have an effect on the behavior of the adversary $\mathcal{A}$.

Game 1: This game is the same as Game 0, except that the common reference string crs in the group public key gpk, and a proof π^* in the challenge signature Σ^* are computed by using the simulator $\mathcal{S} = (\mathtt{Sim}_1, \mathtt{Sim}_2)$ of the NIZK proof system.

Game 2: In this game, we change the plaintext of the ciphertext ct^* in the challenge signature Σ^*. Concretely, the plaintext $0^{|\langle \mathsf{ek}_{i_b}, \mathsf{vk}_{i_b}, \mathsf{cert}_{i_b} \rangle|}$ is encrypted to the ciphertext ct^* instead of $\langle \mathsf{ek}_{i_b}, \mathsf{vk}_{i_b}, \mathsf{cert}_{i_b} \rangle$.

Game 3: In this game, we change the plaintext of the ciphertext $\widetilde{\mathsf{ct}}^*$ in the challenge signature Σ^*. Concretely, the plaintext $0^{|\sigma^*|}$ is encrypted to the ciphertext $\widetilde{\mathsf{ct}}^*$ instead of σ^* where $\sigma^* = \mathsf{SIG.Sign}(\mathsf{sk}_{i_b}, m^*)$.

Game 4: In this game, we change the encryption key of the ciphertext $\widetilde{\mathsf{ct}}^*$. Concretely, we use a random key ek^* to compute $\widetilde{\mathsf{ct}}^*$ instead of using the key ek_{i_b}.

For the advantage $\mathrm{Adv}^{anon}_{\Pi_1,\mathcal{A}}(\lambda, n)$,

$$\mathrm{Adv}^{anon}_{\Pi_1,\mathcal{A}}(\lambda, n) = \left|\Pr[\mathtt{Suc}_0] - \frac{1}{2}\right| \leq \sum_{\ell=0}^{3} \left|\Pr[\mathtt{Suc}_\ell] - \Pr[\mathtt{Suc}_{\ell+1}]\right| + \left|\Pr[\mathtt{Suc}_4] - \frac{1}{2}\right|$$

holds. Moreover, the following lemmas hold.

Lemma 3.1. *There exists a* PPT *algorithm* $\mathcal{B}_1$ *such that* $\left|\Pr[\mathtt{Suc}_0] - \Pr[\mathtt{Suc}_1]\right| = \mathrm{Adv}^{zk}_{\mathcal{P}_L,\mathcal{B}_1}(\lambda)$.

Proof of Lemma 3.1. Let $\mathcal{B}_1$ be an adversary for the zero-knowledgeness of $\mathcal{P}_L$. First, $\mathcal{B}_1$ chooses the challenge bit b, and receives the common reference string crs from the challenger. Next, $\mathcal{B}_1$ generates the rest of instance for the scheme Π_1, and sends $\mathsf{gpk} = (\mathsf{crs}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{ek}_{\mathsf{PKE}})$ and $\mathbf{gsk} = \{\mathsf{gsk}[i]\}$ to $\mathcal{A}$ where $\mathsf{gsk}[i] = (\mathsf{ek}_i, \mathsf{vk}_i, \mathsf{sk}_i, \mathsf{cert}_i)$. If $\mathcal{A}$ sends a query i to the Revoke oracle, $\mathcal{B}_1$ returns $\mathsf{grt}[i] = (\mathsf{dk}_i, \mathsf{vk}_i)$. For the challenge query (i_0, i_1, m^*), $\mathcal{B}_1$ computes the challenge signature Σ^* as follows:

1. Generate $(\mathsf{vk}^*_{\mathrm{ots}}, \mathsf{sk}^*_{\mathrm{ots}}) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$.
2. Compute $\sigma^* \leftarrow \mathsf{SIG.Sign}(\mathsf{sk}_{i_b}, \mathsf{vk}^*_{\mathrm{ots}})$.
3. Choose values r^*_1 and r^*_2 uniform randomly, and compute $\widetilde{\mathsf{ct}}^* \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}_{i_b}, \sigma^*; r^*_1)$ and $\mathsf{ct}^* \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}_{\mathsf{PKE}}, \langle \mathsf{ek}_{i_b}, \mathsf{vk}_{i_b}, \mathsf{cert}_{i_b}\rangle; r^*_2)$.
4. Set $x \leftarrow \langle \mathsf{gpk}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*\rangle$ and $w \leftarrow \langle \mathsf{ek}_{i_b}, \mathsf{vk}_{i_b}, \mathsf{cert}_{i_b}, \sigma^*, r^*_1, r^*_2\rangle$, and send (x, w) to the oracle of the NIZK proof system. Then, obtain a proof π.
5. Set $\pi^* \leftarrow \pi$, generate $\sigma^*_{all} \leftarrow \mathsf{OTS.Sign}(\mathsf{sk}^*_{\mathrm{ots}}, \langle m^*, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*\rangle)$, and send $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$ to $\mathcal{A}$ as the challenge signature.

Finally, when $\mathcal{A}$ terminates with $\widetilde{b}$, $\mathcal{B}_1$ outputs 1 if $b = \widetilde{b}$, and 0 otherwise. If crs is generated by the $\mathsf{ZK.Gen}$ algorithm and $\mathcal{B}_1$ accesses the Prove oracle, then $\mathcal{B}_1$ perfectly simulates Game 0 for $\mathcal{A}$. On the other hand, if crs is generated by using the simulator $\mathtt{Sim}_1$ and $\mathcal{B}_1$ accesses the $\mathsf{SimProve}$ oracle, then $\mathcal{B}_1$ perfectly simulates Game 1. Thus, since $\Pr[\mathrm{Exp}^{proof}_{\mathcal{P}_L,\mathcal{B}_1}(\lambda) = 1] = \Pr[\mathtt{Suc}_0]$ and $\Pr[\mathrm{Exp}^{sim\text{-}proof}_{\mathcal{P}_L,\mathcal{B}_1}(\lambda) = 1] = \Pr[\mathtt{Suc}_1]$ hold, we get $\mathrm{Adv}^{zk}_{\mathcal{P}_L,\mathcal{B}_1}(\lambda) = \left|\Pr[\mathtt{Suc}_0] - \Pr[\mathtt{Suc}_1]\right|$.

Lemma 3.2. *There exists a* PPT *algorithm* $\mathcal{B}_2$ *such that* $\left|\Pr[\mathtt{Suc}_1] - \Pr[\mathtt{Suc}_2]\right| = 2 \cdot \mathrm{Adv}^{ind\text{-}cpa}_{\mathcal{PKE},\mathcal{B}_2}(\lambda)$.

Proof of Lemma 3.2. Let $\mathcal{B}_2$ be an adversary for the IND-CPA security of $\mathcal{PKE}$ and β be the challenge bit in the IND-CPA security game. $\mathcal{B}_2$ chooses the challenge bit b, and receives the public key ek from the challenger. $\mathcal{B}_2$ sets $\mathsf{ek}_{\mathsf{PKE}} \leftarrow \mathsf{ek}$ and generates the common reference string crs by using the simulator $\mathtt{Sim}_1$ where $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathtt{Sim}_1(1^\lambda)$. Also, $\mathcal{B}_2$ generates the rest of instance for the scheme Π_1 by himself. Then, $\mathcal{B}_2$ sends $\mathsf{gpk} = (\mathsf{crs}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{ek}_{\mathsf{PKE}})$ and $\mathbf{gsk} = \{\mathsf{gsk}[i]\}$ to $\mathcal{A}$ where $\mathsf{gsk}[i] = (\mathsf{ek}_i, \mathsf{vk}_i, \mathsf{sk}_i, \mathsf{cert}_i)$. If $\mathcal{A}$ sends a query i to the Revoke oracle, then $\mathcal{B}_2$ returns $\mathsf{grt}[i] = (\mathsf{dk}_i, \mathsf{vk}_i)$. For the challenge query (i_0, i_1, m^*), $\mathcal{B}_1$ computes the challenge signature Σ^* as follows:

1. Generate $(\mathsf{vk}^*_{\mathrm{ots}}, \mathsf{sk}^*_{\mathrm{ots}}) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$.
2. Compute $\sigma^* \leftarrow \mathsf{SIG.Sign}(\mathsf{sk}_{i_b}, \mathsf{vk}^*_{\mathrm{ots}})$.
3. Choose a value r^*_1 uniform randomly, and compute $\widetilde{\mathsf{ct}}^* \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}_{i_b}, \sigma^*; r^*_1)$.
4. Set $M_0 \leftarrow 0^{|\langle \mathsf{ek}_{i_b}, \mathsf{vk}_{i_b}, \mathsf{cert}_{i_b}\rangle|}$ and $M_1 \leftarrow \langle \mathsf{ek}_{i_b}, \mathsf{vk}_{i_b}, \mathsf{cert}_{i_b}\rangle$. Then, send (M_0, M_1) to the challenger for the IND-CPA game and obtain a ciphertext ct^*.
5. Compute $\pi^* \leftarrow \mathtt{Sim}_2(\mathsf{crs}, \mathsf{td}, \langle \mathsf{gpk}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*\rangle)$ where td is the trapdoor generated by $\mathtt{Sim}_1$.
6. Generate $\sigma^*_{all} \leftarrow \mathsf{OTS.Sign}(\mathsf{sk}^*_{\mathrm{ots}}, \langle m^*, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*\rangle)$ and send $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$ to $\mathcal{A}$ as the challenge signature.

Finally, when $\mathcal{A}$ terminates with $\widetilde{b}$, $\mathcal{B}_2$ outputs $\widetilde{\beta} = 1$ if $b = \widetilde{b}$, and $\widetilde{\beta} = 0$ otherwise. If $\beta = 0$, ct^* is represented as $\mathsf{ct}^* = \mathsf{PKE.Enc}(\mathsf{ek}_{\mathsf{PKE}}, 0^{|\langle \mathsf{ek}_{i_b}, \mathsf{vk}_{i_b}, \mathsf{cert}_{i_b} \rangle|})$. Therefore, $\mathcal{B}_2$ perfectly simulates Game 2 if $\beta = 0$. On the other hand, if $\beta = 1$, ct^* is represented as $\mathsf{ct}^* = \mathsf{PKE.Enc}(\mathsf{ek}_{\mathsf{PKE}}, \langle \mathsf{ek}_{i_b}, \mathsf{vk}_{i_b}, \mathsf{cert}_{i_b} \rangle)$, and thus $\mathcal{B}_2$ perfectly simulates Game 1. Thus, we get $\mathrm{Adv}^{ind\text{-}cpa}_{\mathcal{PKE},\mathcal{B}_2}(\lambda) = \left|\Pr[\beta = \widetilde{\beta}] - \frac{1}{2}\right| = \frac{1}{2} \cdot \left|\Pr[\widetilde{\beta} = 1 | \beta = 1] - \Pr[\widetilde{\beta} = 1 | \beta = 0]\right| = \frac{1}{2} \cdot \left|\Pr[b = \widetilde{b} | \beta = 1] - \Pr[b = \widetilde{b} | \beta = 0]\right| = \frac{1}{2} \cdot \left|\Pr[\mathtt{Suc}_1] - \Pr[\mathtt{Suc}_2]\right|$. That is, it holds that $\left|\Pr[\mathtt{Suc}_1] - \Pr[\mathtt{Suc}_2]\right| = 2 \cdot \mathrm{Adv}^{ind\text{-}cpa}_{\mathcal{PKE},\mathcal{B}_2}(\lambda)$.

Lemma 3.3. *There exists a* PPT *algorithm* $\mathcal{B}_3$ *such that* $\left|\Pr[\mathtt{Suc}_2] - \Pr[\mathtt{Suc}_3]\right| = 2n \cdot \mathrm{Adv}^{ind\text{-}cpa}_{\mathcal{PKE},\mathcal{B}_3}(\lambda)$.

Proof of Lemma 3.3. Let $\mathcal{B}_3$ be an adversary for the IND-CPA security of $\mathcal{PKE}$ and β be the challenge bit in the IND-CPA security game. $\mathcal{B}_3$ chooses the challenge bit b, and receives the public key ek from the challenger. Moreover, $\mathcal{B}_3$ chooses an index $i^* \in [1, n]$ uniform randomly. Then, $\mathcal{B}_3$ sets $\mathsf{ek}_{i^*} \leftarrow \mathsf{ek}$ and generates the common reference string crs by using the simulator $\mathtt{Sim}_1$ where $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathtt{Sim}_1(1^\lambda)$. Also, $\mathcal{B}_3$ generates the rest of instance for the scheme Π by himself. Then, $\mathcal{B}_3$ sends $\mathsf{gpk} = (\mathsf{crs}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{ek}_{\mathsf{PKE}})$ and $\mathbf{gsk} = \{\mathsf{gsk}[i]\}$ to $\mathcal{A}$ where $\mathsf{gsk}[i] = (\mathsf{ek}_i, \mathsf{vk}_i, \mathsf{sk}_i, \mathsf{cert}_i)$. We remark that $\mathcal{B}_3$ cannot compute $\mathsf{grt}[i^*]$ since $\mathcal{B}_3$ does not know the decryption key dk_{i^*} corresponding to the encryption key ek_{i^*}. However, $\mathcal{B}_3$ can generate the i^*'s user signing key $\mathsf{gsk}[i^*] = (\mathsf{ek}_{i^*}, \mathsf{vk}_{i^*}, \mathsf{sk}_{i^*}, \mathsf{cert}_{i^*})$ without knowing the value dk_{i^*}. When $\mathcal{A}$ sends a query i to the Revoke oracle, then $\mathcal{B}_3$ returns $\mathsf{grt}[i] = (\mathsf{dk}_i, \mathsf{vk}_i)$ if $i \neq i^*$. Otherwise, if $i = i^*$, then $\mathcal{B}_3$ outputs a random bit $\widetilde{\beta}$. For the challenge (i_0, i_1, m^*), $\mathcal{B}_3$ computes the challenge signature Σ^* as follows:

1. If $i_b \neq i^*$, then output a random bit $\widetilde{\beta}$. If $i_b = i^*$, go to the next step.
2. Generate $(\mathsf{vk}^*_{\mathrm{ots}}, \mathsf{sk}^*_{\mathrm{ots}}) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$.
3. Compute $\sigma^* \leftarrow \mathsf{SIG.Sign}(\mathsf{sk}_{i_b}, \mathsf{vk}^*_{\mathrm{ots}})$.
4. Set $M_0 \leftarrow 0^{|\sigma^*|}$ and $M_1 \leftarrow \sigma^*$. Then, send (M_0, M_1) to the challenger for the IND-CPA game and receive a ciphertext $\widetilde{\mathsf{ct}}^*$.
5. Choose a value r^*_2 uniform randomly, and compute $\mathsf{ct}^* \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}_{\mathsf{PKE}}, 0^{|\langle \mathsf{ek}_{i_b}, \mathsf{vk}_{i_b}, \mathsf{cert}_{i_b} \rangle|}; r^*_2)$.
6. Compute $\pi^* \leftarrow \mathtt{Sim}_2(\mathsf{crs}, \mathsf{td}, \langle \mathsf{gpk}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle)$ where td is the trapdoor generated by $\mathtt{Sim}_1$.
7. Generate $\sigma^*_{all} \leftarrow \mathsf{OTS.Sign}(\mathsf{sk}^*_{\mathrm{ots}}, \langle m^*, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^* \rangle)$ and send $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$ to $\mathcal{A}$ as the challenge signature.

Finally, when $\mathcal{A}$ terminates with $\widetilde{b}$, $\mathcal{B}_3$ outputs $\widetilde{\beta} = 1$ if $b = \widetilde{b}$, and $\widetilde{\beta} = 0$ otherwise. If $i_b = i^*$ and $\beta = 0$, $\widetilde{\mathsf{ct}}^*$ is represented as $\widetilde{\mathsf{ct}}^* = \mathsf{PKE.Enc}(\mathsf{ek}_{i_b}, 0^{|\sigma^*|})$. Therefore, $\mathcal{B}_3$ perfectly simulates Game 3. On the other hand, if $i_b = i^*$ and $\beta = 1$, $\widetilde{\mathsf{ct}}^*$ is represented as $\widetilde{\mathsf{ct}}^* = \mathsf{PKE.Enc}(\mathsf{ek}_{i_b}, \sigma^*)$, and thus $\mathcal{B}_3$ perfectly simulates Game 2. Let $\mathtt{Good}$ be the event that $i_b = i^*$ holds where i^* is chosen by $\mathcal{B}_3$ at the beginning of the game. Since the guess of $i^* \in [1, n]$ and the behavior of $\mathcal{A}$ are independent with each other, $\Pr[\mathtt{Good}] = 1/n$ holds. Thus, we get $\left|\Pr[\mathtt{Suc}_2] - \Pr[\mathtt{Suc}_3]\right| = 2n \cdot \mathrm{Adv}^{ind\text{-}cpa}_{\mathcal{PKE},\mathcal{B}_3}(\lambda)$.

Lemma 3.4. *There exists a* PPT *algorithm* $\mathcal{B}_4$ *such that* $\left|\Pr[\mathtt{Suc}_3] - \Pr[\mathtt{Suc}_4]\right| = 2n \cdot \mathrm{Adv}^{key\text{-}priv}_{\mathcal{PKE},\mathcal{B}_4}(\lambda)$.

Proof of Lemma 3.4. Let $\mathcal{B}_4$ be an adversary for the key privacy of $\mathcal{PKE}$ and β be the challenge bit in the key privacy game. $\mathcal{B}_4$ chooses the challenge bit b, and receives two public keys ek and ek^* from the challenger. Moreover, $\mathcal{B}_4$ chooses the index $i^* \in [1, n]$ uniform randomly. Then, $\mathcal{B}_4$ sets $\mathsf{ek}_{i^*} \leftarrow \mathsf{ek}$ and generates the common reference string crs by using the simulator $\mathtt{Sim}_1$ where $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathtt{Sim}_1(1^\lambda)$. Also, $\mathcal{B}_4$ generates the rest of instance for the scheme Π_1 by himself. Then, $\mathcal{B}_4$ sends $\mathsf{gpk} = (\mathsf{crs}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{ek}_{\mathsf{PKE}})$ and $\mathbf{gsk} = \{\mathsf{gsk}[i]\}$ to $\mathcal{A}$ where $\mathsf{gsk}[i] = (\mathsf{ek}_i, \mathsf{vk}_i, \mathsf{sk}_i, \mathsf{cert}_i)$. We remark that $\mathcal{B}_4$ cannot compute $\mathsf{grt}[i^*]$ since $\mathcal{B}_4$ does not know the decryption key dk_{i^*} corresponding to the encryption key ek_{i^*}. However, $\mathcal{B}_4$ can generate the i^*'s user signing key $\mathsf{gsk}[i^*] = (\mathsf{ek}_{i^*}, \mathsf{vk}_{i^*}, \mathsf{sk}_{i^*}, \mathsf{cert}_{i^*})$ without knowing dk_{i^*}. When $\mathcal{A}$ sends a query i to the Revoke oracle, then $\mathcal{B}_4$ returns $\mathsf{grt}[i] = (\mathsf{dk}_i, \mathsf{vk}_i)$ if $i \neq i^*$. Otherwise, if $i = i^*$, then $\mathcal{B}_4$ outputs a random bit $\widetilde{\beta}$. For the challenge (i_0, i_1, m^*), $\mathcal{B}_4$ computes the challenge signature Σ^* as follows:

1. If $i_b \neq i^*$, then output a random bit $\widetilde{\beta}$. If $i_b = i^*$, go to the next step.

2. Generate $(\mathsf{vk}^*_{\mathrm{ots}}, \mathsf{sk}^*_{\mathrm{ots}}) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$.
3. Compute $\sigma^* \leftarrow \mathsf{SIG.Sign}(\mathsf{sk}_{i_b}, \mathsf{vk}^*_{\mathrm{ots}})$.
4. Set $M^* \leftarrow 0^{|\sigma^*|}$. Then, send M^* to the challenger for the key privacy game and receive $\widetilde{\mathsf{ct}}^*$.
5. Choose a value r^*_2 uniform randomly, and compute $\mathsf{ct}^* \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}_{\mathsf{PKE}}, 0^{|\langle \mathsf{ek}_{i_b}, \mathsf{vk}_{i_b}, \mathsf{cert}_{i_b}\rangle|}; r^*_2)$.
6. Compute $\pi^* \leftarrow \mathtt{Sim}_2(\mathsf{crs}, \mathsf{td}, \langle \mathsf{gpk}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*\rangle)$ where td is the trapdoor generated by $\mathtt{Sim}_1$.
7. Generate $\sigma^*_{all} \leftarrow \mathsf{OTS.Sign}(\mathsf{sk}^*_{\mathrm{ots}}, \langle m^*, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*\rangle)$ and send $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$ to $\mathcal{A}$ as the challenge signature.

Finally, when $\mathcal{A}$ terminates with $\widetilde{b}$, $\mathcal{B}_3$ outputs $\widetilde{\beta} = 1$ if $b = \widetilde{b}$, and $\widetilde{\beta} = 0$ otherwise. If $i_b = i^*$ and $\beta = 0$, $\widetilde{\mathsf{ct}}^*$ is represented as $\widetilde{\mathsf{ct}}^* = \mathsf{PKE.Enc}(\mathsf{ek}_{i_b}, 0^{|\sigma^*|})$, and thus $\mathcal{B}_4$ perfectly simulates Game 3. On the other hand, if $i_b = i^*$ and $\beta = 1$, $\widetilde{\mathsf{ct}}^*$ is represented as $\widetilde{\mathsf{ct}}^* = \mathsf{PKE.Enc}(\mathsf{ek}^*, 0^{|\sigma^*|})$, and thus $\mathcal{B}_4$ perfectly simulates Game 4. Let $\mathtt{Good}$ be the event that $i_b = i^*$ holds where i^* is chosen by $\mathcal{B}_4$ at the beginning of the game. Since the guess of $i^* \in [1, n]$ and the behavior of $\mathcal{A}$ are independent with each other, $\Pr[\mathtt{Good}] = 1/n$ holds. As in the same formula deformation in the proof of Lemma 3.3, we get $\left|\Pr[\mathtt{Suc}_3] - \Pr[\mathtt{Suc}_4]\right| = 2n \cdot \mathrm{Adv}^{\textit{key-priv}}_{\mathcal{PKE},\mathcal{B}_4}(\lambda)$.

In Game 4, the choice of the challenge bit b and the distribution of the challenge signature $\Sigma^* = (\widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*)$ are independent. Thus, $\Pr[\mathtt{Suc}_4] = 1/2$ holds. Putting all together, we get

$$\begin{aligned}\mathrm{Adv}^{\textit{anon}}_{\Pi_1,\mathcal{A}}(\lambda, n) \leq& \sum_{i=0}^{3}\left|\Pr[\mathtt{Suc}_i] - \Pr[\mathtt{Suc}_{i+1}]\right| + \left|\Pr[\mathtt{Suc}_4] - 1/2\right| \\ =& \mathrm{Adv}^{zk}_{\mathcal{P}_L,\mathcal{B}_1}(\lambda) + 2 \cdot \mathrm{Adv}^{\textit{ind-cpa}}_{\mathcal{PKE},\mathcal{B}_2}(\lambda) + 2n \cdot \mathrm{Adv}^{\textit{ind-cpa}}_{\mathcal{PKE},\mathcal{B}_3}(\lambda) + 2n \cdot \mathrm{Adv}^{\textit{key-priv}}_{\mathcal{PKE},\mathcal{B}_4}(\lambda).\end{aligned}$$

Since the choice of the parameter n and the adversary $\mathcal{A}$ is arbitrary, our scheme Π_1 satisfies full anonymity if the underlying NIZK proof system $\mathcal{P}_L$ satisfies zero-knowledgeness and the underlying PKE scheme $\mathcal{PKE}$ satisfies IND-CPA security and key privacy. □

Traceability. Intuitively, due to the soundness of $\mathcal{P}_L$, the probability that a valid proof π for a statement $\langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}\rangle \notin L$ can be constructed is negligible where L is the language defined in Section 3.2. Therefore, if $\Sigma = (\mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi, \sigma_{all})$ is a valid signature on m, it holds that $\langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}\rangle \in L$ with high probability. Thus, there exists a witness $\langle \mathsf{ek}^*, \mathsf{vk}^*, \mathsf{cert}^*, \sigma^*, r^*_1, r^*_2\rangle$ satisfying the equations (a) $\widetilde{\mathsf{ct}} = \mathsf{PKE.Enc}(\mathsf{ek}^*, \sigma^*; r^*_1)$, (b) $\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{ek}_{\mathsf{PKE}}, \langle \mathsf{ek}^*, \mathsf{vk}^*, \mathsf{cert}^*\rangle; r^*_2)$, (c) $\mathsf{SIG.Verify}(\mathsf{vk}_{\mathsf{SIG}}, \langle \mathsf{ek}^*, \mathsf{vk}^*\rangle, \mathsf{cert}^*) = 1$, and (d) $\mathsf{SIG.Verify}(\mathsf{vk}^*, \mathsf{vk}^*_{\mathrm{ots}}, \sigma^*) = 1$.

From the EUF-CMA security of the scheme $\mathcal{SIG}$, it is difficult to generate the value cert^* which satisfies Equation (c) for an uncertified key pair $\langle \mathsf{ek}^*, \mathsf{vk}^*\rangle$. Therefore, for some index $i \in [1, n]$, $(\mathsf{ek}^*, \mathsf{vk}^*) = (\mathsf{ek}_i, \mathsf{vk}_i)$ holds. Thus, the only way to generate a forgery is to produce a signature σ^* which satisfies Equation (d). However, it is also difficult to produce such a signature due to the EUF-CMA security of $\mathcal{SIG}$. Therefore, Scheme 1 satisfies traceability. Formally, the following theorem holds.

Theorem 3.3. *Scheme 1 satisfies traceability if the underlying OTS scheme $\mathcal{OTS}$ satisfies strong EUF-CMA security, the underlying NIZK proof system $\mathcal{P}_L$ satisfies soundness and the underlying digital signature scheme $\mathcal{SIG}$ satisfies EUF-CMA security.*

Proof. Let $\mathcal{A}$ be an adversary for traceability of Π_1, and $(m^*, \Sigma^*, \mathsf{RU}^*)$ be the output of $\mathcal{A}$ in the experiment $\mathrm{Exp}^{\textit{trace}}_{\Pi_1,\mathcal{A}}(\lambda, n)$ where $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$. Let i^* be the output of the $\mathsf{GS.Open}$ algorithm with an input (m^*, Σ^*). We consider the following six cases:

$$\textbf{I}.\ \langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*\rangle \notin L, \quad \textbf{II}.\ i^* = \perp, \quad \textbf{III}.\ i^* \notin \mathsf{CU}, \quad \text{and} \quad \textbf{IV}.\ i^* \in \mathsf{RU}^*$$

$$\textbf{V}.\ \text{for some } (i, m, \Sigma) \in \mathsf{QL},\ \mathsf{vk}_{\mathrm{ots}} = \mathsf{vk}^*_{\mathrm{ots}} \text{ and } (m, \Sigma) \neq (m^*, \Sigma^*) \text{ where } \Sigma = (\mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi, \sigma_{all}),$$

$$\textbf{VI}.\ \text{for all } (i, m, \Sigma) \in \mathsf{QL},\ \mathsf{vk}_{\mathrm{ots}} \neq \mathsf{vk}^*_{\mathrm{ots}} \text{ or } (m, \Sigma) = (m^*, \Sigma^*) \text{ where } \Sigma = (\mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi, \sigma_{all}).$$

If the output of the experiment $\mathrm{Exp}^{\textit{trace}}_{\Pi_1,\mathcal{A}}(\lambda, n, T)$ is 1 (i.e., $\mathcal{A}$ succeeds in producing a forged signature), we can classify the type of the forgery as follows:

$$(1)\ \mathrm{I},\ (2)\ \neg\mathrm{I} \wedge \mathrm{II},\ (3)\ \neg\mathrm{I} \wedge \mathrm{III} \wedge \mathrm{V},\ (4)\ \neg\mathrm{I} \wedge \mathrm{III} \wedge \mathrm{VI},\ (5)\ \neg\mathrm{I} \wedge \mathrm{IV},$$

Let $\mathtt{E}_\ell$ be the event that $\mathcal{A}$ outputs a forged signature in Type ℓ. We estimate the each probability that the event $\mathtt{E}_\ell$ happens in the following lemmas.

Lemma 3.5. *There exists a* PPT *algorithm* $\mathcal{B}_1$ *such that* $\Pr[\mathtt{E}_1] \leq \mathrm{Adv}^{sound}_{\mathcal{P}_L,\mathcal{B}_1}(\lambda)$.

Proof of Lemma 3.5. Let $\mathcal{B}_1$ be an adversary for soundness of $\mathcal{P}_L$. First, $\mathcal{B}_1$ receives the common reference string crs from the challenger. Next, $\mathcal{B}_1$ generates the rest of instance for the scheme Π_1, and sends $\mathsf{gpk} = (\mathsf{crs}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{ek}_{\mathsf{PKE}})$ and $\mathbf{grt} = \{\mathsf{grt}[i]\}$ to $\mathcal{A}$ where $\mathsf{grt}[i] = (\mathsf{dk}_i, \mathsf{vk}_i)$. Since $\mathcal{B}_1$ has all signing keys, he can easily simulate the $\mathsf{GS.Sign}$ oracle and the $\mathsf{Corrupt}$ oracle. Let $(m^*, \Sigma^*, \mathsf{RU}^*)$ be the output of $\mathcal{A}$ where $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$. Then, $\mathcal{B}_1$ outputs $(\langle \mathsf{gpk}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle, \pi^*)$ as a forgery for the soundness of $\mathcal{P}_L$. When $\mathcal{A}$'s output $(m^*, \Sigma^*, \mathsf{RU}^*)$ is a forgery in Type 1, $\mathsf{ZK.Verify}(\mathsf{crs}, \langle \mathsf{gpk}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle, \pi^*) = 1$ and $\langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle \notin L$ hold. Therefore, $(\langle \mathsf{gpk}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle, \pi^*)$ is the forgery for the soundness of $\mathcal{P}_L$. Thus, we have $\Pr[\mathtt{E}_1] \leq \mathrm{Adv}^{sound}_{\mathcal{P}_L,\mathcal{B}_1}(\lambda)$.

Lemma 3.6. *There exists a* PPT *algorithm* $\mathcal{B}_2$ *such that* $\Pr[\mathtt{E}_2] \leq \mathrm{Adv}^{unforge}_{\mathcal{SIG},\mathcal{B}_2}(\lambda)$.

Proof of Lemma 3.6. Let $\mathcal{B}_2$ be an adversary for the EUF-CMA security of $\mathcal{SIG}$. First, $\mathcal{B}_2$ receives the verification key vk from the challenger of the EUF-CMA security game, and sets $\mathsf{vk}_{\mathsf{SIG}} \leftarrow \mathsf{vk}$. Next, $\mathcal{B}_2$ generates the rest of instance for the scheme Π_1, except for the certificates cert_i where $i \in [1, n]$. In terms of the certificates, $\mathcal{B}_2$ sends $\langle \mathsf{ek}_i, \mathsf{vk}_i \rangle$ to the Sign oracle of the scheme $\mathcal{SIG}$, and receives cert_i. $\mathcal{B}_2$ sends $\mathsf{gpk} = (\mathsf{crs}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{ek}_{\mathsf{PKE}})$ and $\mathbf{grt} = \{\mathsf{grt}[i]\}$ to $\mathcal{A}$ where $\mathsf{grt}[i] = (\mathsf{dk}_i, \mathsf{vk}_i)$. Since $\mathcal{B}_2$ has all signing keys, he can easily simulate the $\mathsf{GS.Sign}$ oracle and the $\mathsf{Corrupt}$ oracle. Let $(m^*, \Sigma^*, \mathsf{RU}^*)$ be the output of $\mathcal{A}$ where $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$. Then, $\mathcal{B}_2$ outputs $(\langle \mathsf{ek}^*, \mathsf{vk}^* \rangle, \mathsf{cert}^*)$ as a forged signature of $\mathcal{SIG}$. When $\mathcal{A}$'s output $(m^*, \Sigma^*, \mathsf{RU}^*)$ is a forgery in Type 2, $\langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle \in L$ holds. Also, $\mathsf{SIG.Verify}(\mathsf{vk}_{\mathsf{SIG}}, \langle \mathsf{ek}^*, \mathsf{vk}^* \rangle, \mathsf{cert}^*) = 1$ holds where $\langle \mathsf{ek}^*, \mathsf{vk}^*, \mathsf{cert}^* \rangle$ is the decryption result of ct^* by the decryption key $\mathsf{dk}_{\mathsf{PKE}}$. Since for all $i \in [1, n]$, $(\mathsf{ek}^*, \mathsf{vk}^*) \neq (\mathsf{ek}_i, \mathsf{vk}_i)$ holds, $\mathcal{B}_2$ does not send $(\mathsf{ek}^*, \mathsf{vk}^*)$ to the Sign oracle. Thus, $(\langle \mathsf{ek}^*, \mathsf{vk}^* \rangle, \mathsf{cert}^*)$ is a forged signature of the digital signature scheme $\mathcal{SIG}$, and we get $\Pr[\mathtt{E}_2] \leq \mathrm{Adv}^{unforge}_{\mathcal{SIG},\mathcal{B}_2}(\lambda)$.

Lemma 3.7. *There exists a* PPT *algorithm* $\mathcal{B}_3$ *such that* $\Pr[\mathtt{E}_3] \leq q_{\mathsf{GS.Sign}} \cdot \mathrm{Adv}^{strong\text{-}unforge}_{\mathcal{OTS},\mathcal{B}_3}(\lambda)$ *where* $q_{\mathsf{GS.Sign}}$ *is the number of signing queries.*

Proof of Lemma 3.7. Let $\mathcal{B}_3$ be an adversary for the strong EUF-CMA security of $\mathcal{OTS}$. First, $\mathcal{B}_3$ receives the verification key $\mathsf{vk}_{\mathrm{ots}}$ from the challenger of the strong EUF-CMA security game. Next, $\mathcal{B}_3$ chooses random $\widehat{j} \in [1, q_{\mathsf{GS.Sign}}]$ and generates the instance for the scheme Π_1. For the $\widehat{j}$-th signing query $(i, 1, m)$, $\mathcal{B}_3$ generates $(\widetilde{\mathsf{ct}}, \mathsf{ct}, \pi)$ as in the scheme Π_1, sends $\langle m, \mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi \rangle$ to the Sign oracle of the scheme $\mathcal{OTS}$, receives σ_{all}, and sends $\Sigma = (\mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi, \sigma_{all})$ back to $\mathcal{A}$. For any other query, $\mathcal{B}_3$ simulates $\mathsf{GS.Sign}$ oracle and the $\mathsf{Corrupt}$ oracle honestly. Let $(m^*, \Sigma^*, \mathsf{RU}^*)$ be the output of $\mathcal{A}$ where $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$. Then, $\mathcal{B}_3$ outputs $(\langle m^*, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^* \rangle, \sigma^*_{all})$ as a forged signature of $\mathcal{OTS}$. When $\mathcal{A}$'s output $(m^*, \Sigma^*, \mathsf{RU}^*)$ is a forgery in Type 3, $\mathsf{OTS.Verify}(\mathsf{vk}^*_{\mathrm{ots}}, \langle m^*, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^* \rangle, \sigma^*_{all}) = 1$ holds. Furthermore, for at least one query $(i, 1, m)$ to the $\mathsf{GS.Sign}$ oracle and the response Σ to it, it holds that $\mathsf{vk}_{\mathrm{ots}} = \mathsf{vk}^*_{\mathrm{ots}}$ and $(m, \Sigma) \neq (m^*, \Sigma^*)$. The latter implies that $(\langle m^*, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^* \rangle, \sigma^*_{all}) \neq (\langle m, \mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi \rangle, \sigma_{all})$. If the $\widehat{j}$-th query to the $\mathsf{GS.Sign}$ oracle is such a query, $\mathcal{B}_3$'s signing query is $\langle m, \mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi \rangle$ and the response to it is σ_{all}. Then in that case $\mathcal{B}_3$'s output is a forgery for the $\mathcal{OTS}$ scheme. The probability that the $\widehat{j}$-th query is such a query is at least $1/q_{\mathsf{GS.Sign}}$, then we have $(1/q_{\mathsf{GS.Sign}}) \cdot \Pr[\mathtt{E}_3] \leq \mathrm{Adv}^{\boldsymbol{strong\text{-}unforge}}_{\mathcal{OTS},\mathcal{B}_3}(\lambda)$.

Lemma 3.8. *There exists a* PPT *algorithm* $\mathcal{B}_4$ *such that* $\Pr[\mathtt{E}_4] \leq n \cdot \mathrm{Adv}^{unforge}_{\mathcal{SIG},\mathcal{B}_4}(\lambda)$.

Proof of Lemma 3.8. Let $\mathcal{B}_4$ be an adversary for the EUF-CMA security of $\mathcal{SIG}$. First, $\mathcal{B}_4$ receives vk from the challenger of the EUF-CMA security game, and randomly chooses the index $i^* \in [1, n]$. Then, $\mathcal{B}_4$ sets $\mathsf{vk}_{i^*} \leftarrow \mathsf{vk}$. Next, $\mathcal{B}_4$ generates the rest of instance for the scheme Π_1, and sends $\mathsf{gpk} = (\mathsf{crs}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{ek}_{\mathsf{PKE}})$ and $\mathbf{grt} = \{\mathsf{grt}[i]\}$ to $\mathcal{A}$ where $\mathsf{grt}[i] = (\mathsf{dk}_i, \mathsf{vk}_i)$. We remark that $\mathcal{B}_4$ does not know the signing key sk_{i^*} corresponding to vk_{i^*}. Thus, $\mathcal{B}_4$ cannot compute $\mathsf{gsk}[i^*]$. If $\mathcal{A}$ sends (i, m) to the $\mathsf{GS.Sign}$ oracle and it holds $i \neq i^*$, then $\mathcal{B}_4$ easily computes Σ since $\mathcal{B}_4$ knows $\mathsf{gsk}[i]$ for all users $i \neq i^*$. On the other

hand, if $\mathcal{A}$ sends (i^*, m) to the $\mathsf{GS.Sign}$ oracle, $\mathcal{B}_4$ generates $(\mathsf{vk}_{\mathrm{ots}}, \mathsf{sk}_{\mathrm{ots}}) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$ and sends $\mathsf{vk}_{\mathrm{ots}}$ to the Sign oracle of $\mathcal{SIG}$ and receives a signature σ. Then, he computes $(\widetilde{\mathsf{ct}}, \mathsf{ct})$, π and σ_{all} according to the $\mathsf{GS.Sign}$ algorithm, and returns $\Sigma = (\mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi, \sigma_{all})$ to $\mathcal{A}$. If $\mathcal{A}$ sends i^* to the $\mathsf{Corrupt}$ oracle, then $\mathcal{B}_4$ outputs $\perp$. Otherwise, if $\mathcal{A}$ sends i such that $i \neq i^*$ to the $\mathsf{Corrupt}$ oracle, then $\mathcal{B}_4$ returns $\mathsf{gsk}[i]$. Let $(m^*, \Sigma^*, \mathsf{RU}^*)$ be the output of $\mathcal{A}$ where $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$. If $\mathsf{GS.Open}(\mathsf{gpk}, \mathbf{grt}, m^*, \Sigma^*) \neq i^*$, $\mathcal{B}_4$ outputs $\perp$. Otherwise, if $\mathsf{GS.Open}(\mathsf{gpk}, \mathbf{grt}, m^*, \Sigma^*) = i^*$, then $\mathcal{B}_4$ decrypts $\widetilde{\mathsf{ct}}^*$ by using dk_{i^*} and gets the decryption result σ^*. Then, $\mathcal{B}_4$ outputs $(\mathsf{vk}^*_{\mathrm{ots}}, \sigma^*)$ as a forged signature. Since the guess of $i^* \in [1, n]$ and the behavior of $\mathcal{A}$ are independent with each other, the probability that $\mathsf{GS.Open}(\mathsf{gpk}, \mathbf{grt}, m^*, \Sigma^*) = i^*$ holds is $1/n$. If $\mathsf{GS.Open}(\mathsf{gpk}, \mathbf{grt}, m^*, \Sigma^*) = i^*$, $i^* \notin \mathsf{CU}$ holds by the condition of Type 4. Thus, i^* is not queried to the $\mathsf{Corrupt}$ oracle and $\mathcal{B}_4$ can succeed in simulating the $\mathsf{Corrupt}$ oracle. Due to the condition of Type 4, for all $(i, m, \Sigma) \in \mathsf{QL}$, it satisfies either $\mathsf{vk}_{\mathrm{ots}} \neq \mathsf{vk}^*_{\mathrm{ots}}$ or $(m, \Sigma) = (m^*, \Sigma^*)$. Now, we claim that $\mathsf{vk}^*_{\mathrm{ots}}$ is not queried by the above B_4 even in the case $(m, \Sigma) = (m^*, \Sigma^*)$. Let (i, m) be an arbitrary query issued by $\mathcal{A}$ during the execution. Then there is some $(i, m, \Sigma) \in \mathsf{QL}$ where $\Sigma = (\mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi, \sigma_{all})$. For each such a query (i, m) $\mathcal{B}$ might issue a signing query $\mathsf{vk}_{\mathrm{ots}}$ to its Sign oracle. We can assume $(m, \Sigma) = (m^*, \Sigma^*)$. If $i \neq i^*$, then $\mathcal{B}_4$ does not issue a signing query. If $i = i^*$, then $(i^*, m^*, \Sigma^*) = (i, m, \Sigma) \in \mathsf{QL}$, which contradicts the success condition of $\mathcal{A}$. Thus this case never occurs. Moreover, if $\mathsf{GS.Open}(\mathsf{gpk}, \mathbf{grt}, m^*, \Sigma^*) = i^*$, it holds that $\mathsf{GS.Verify}(\mathsf{gpk}, \{\mathsf{grt}[i^*]\}, m^*, \Sigma^*) = 0$. Thus, either $\mathsf{ZK.Verify}(\mathsf{crs}, \langle \mathsf{gpk}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle, \pi^*) = 0$ or $\mathsf{SIG.Verify}(\mathsf{vk}_{i^*}, \mathsf{vk}^*_{\mathrm{ots}}, \mathsf{PKE.Dec}(\mathsf{dk}_{i^*}, \widetilde{\mathsf{ct}}^*)) = 1$ hold. Here, due to the success condition of $\mathcal{A}$, $\mathsf{ZK.Verify}(\mathsf{crs}, \langle \mathsf{gpk}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, m^* \rangle, \pi^*) = 1$ holds. Thus, $\mathsf{SIG.Verify}(\mathsf{vk}_{i^*}, \mathsf{vk}^*_{\mathrm{ots}}, \mathsf{PKE.Dec}(\mathsf{dk}_{i^*}, \widetilde{\mathsf{ct}}^*)) = 1$ holds. Therefore, if $\mathsf{GS.Open}(\mathsf{gpk}, \mathbf{grt}, m^*, \Sigma^*) = i^*$ holds, $(\mathsf{vk}^*_{\mathrm{ots}}, \sigma^*)$ is a forged signature of the $\mathcal{SIG}$ scheme where σ^* is the decryption result of $\widetilde{\mathsf{ct}}^*$ by using the decryption key dk_{i^*}. Since the probability that $\mathsf{GS.Open}(\mathsf{gpk}, \mathbf{grt}, m^*, \Sigma^*) = i^*$ holds is $1/n$, we have $(1/n) \cdot \Pr[\mathtt{E}_4] \leq \mathrm{Adv}^{unforge}_{\mathcal{SIG}, \mathcal{B}_4}(\lambda)$.

Lemma 3.9. $\Pr[\mathtt{E}_5] = 0$ *holds.*

Proof of Lemma 3.9. Let $(m^*, \Sigma^*, \mathsf{RU}^*)$ be the output of $\mathcal{A}$, and i^* be the result of the the $\mathsf{GS.Open}$ algorithm with the input (m^*, Σ^*). If $i^* \in \mathsf{RU}^*$, then $\mathsf{grt}[i^*] \in \mathsf{RL}^*$. Due to the success probability, $\mathsf{GS.Verify}(\mathsf{gpk}, \mathsf{RL}^*, m^*, \Sigma^*) = 1$ holds. Thus, due to the description of the $\mathsf{GS.Verify}$ algorithm, $\mathsf{ZK.Verify}(\mathsf{crs}, \langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle, \pi^*) = 1$ holds. Also, for $\mathsf{grt}[i^*] = (\mathsf{dk}_{i^*}, \mathsf{vk}_{i^*}) \in \mathsf{RL}^*$, it holds that $\mathsf{SIG.Verify}(\mathsf{vk}_{i^*}, \mathsf{vk}^*_{\mathrm{ots}}, \mathsf{PKE.Dec}(\mathsf{dk}_{i^*}, \widetilde{\mathsf{ct}}^*)) = 0$. Moreover, since the opening result is i^*, we have $\mathsf{GS.Verify}(\mathsf{gpk}, \{\mathsf{grt}[i^*]\}, m^*, \Sigma^*) = 0$. Thus, either $\mathsf{ZK.Verify}(\mathsf{crs}, \langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle, \pi^*) = 0$ or $\mathsf{SIG.Verify}(\mathsf{vk}_{i^*}, \mathsf{vk}^*_{\mathrm{ots}}, \mathsf{PKE.Dec}(\mathsf{dk}_{i^*}, \widetilde{\mathsf{ct}}^*)) = 1$ holds. However, this contradicts the condition that $\mathsf{ZK.Verify}(\mathsf{crs}, \langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle, \pi^*) = 1$ and $\mathsf{SIG.Verify}(\mathsf{vk}_{i^*}, \mathsf{vk}^*_{\mathrm{ots}}, \mathsf{PKE.Dec}(\mathsf{dk}_{i^*}, \widetilde{\mathsf{ct}}^*)) = 0$. Thus, we get $\Pr[\mathtt{E}_5] = 0$.

Putting all together, we get

$$\begin{aligned}
\mathrm{Adv}^{trace}_{\Pi_1, \mathcal{A}}(\lambda, n) &= \Pr[\ \mathrm{Exp}^{trace}_{\Pi_1, \mathcal{A}}(\lambda, n) = 1\ ] \\
&= \Pr[\ \mathtt{E}_1 \vee \mathtt{E}_2 \vee \mathtt{E}_3 \vee \mathtt{E}_4 \vee \mathtt{E}_5\ ] \\
&\leq \Pr[\mathtt{E}_1] + \Pr[\mathtt{E}_2] + \Pr[\mathtt{E}_3] + \Pr[\mathtt{E}_4] + \Pr[\mathtt{E}_5] \\
&\leq \mathrm{Adv}^{sound}_{\mathcal{P}_L, \mathcal{B}_1}(\lambda) + \mathrm{Adv}^{unforge}_{\mathcal{SIG}, \mathcal{B}_2}(\lambda) + q_{\mathsf{GS.Sign}} \cdot \mathrm{Adv}^{strong\text{-}unforge}_{\mathcal{OTS}, \mathcal{B}_3}(\lambda) + n \cdot \mathrm{Adv}^{unforge}_{\mathcal{SIG}, \mathcal{B}_4}(\lambda).
\end{aligned}$$

Since the choice of the parameters $q_{\mathsf{GS.Sign}}$ and n, and the adversary $\mathcal{A}$ are arbitrary, our scheme Π_1 satisfies traceability if the underlying NIZK proof system $\mathcal{P}_L$ satisfies soundness and the underlying digital signature scheme $\mathcal{SIG}$ satisfies EUF-CMA security. □

4 A Fully Anonymous VLR-GS Scheme with Backward Unlinkability

In Figure 7, we give a construction of a fully anonymous VLR-GS scheme (Scheme 2) which satisfies backward unlinkability from the same building blocks of Scheme 1. In Scheme 2, encryption/decryption key pairs of a user i are provided for each time period j, and when generating a signature at the time period j, the user uses the corresponding encryption key $\mathsf{ek}_i^{(j)}$ to encrypt a signature σ. Also, the decryption key $\mathsf{dk}_i^{(j)}$ is set to be i's revocation token for the time period j. To force users to use the encryption key related to the appropriate time period, each encryption key $\mathsf{ek}_i^{(j)}$ is certified along with the verification key vk_i by using the manager's signing key $\mathsf{sk}_{\mathsf{SIG}}^{(j)}$ that depends on the time period j.

$\mathsf{GS.Gen}(1^\lambda, n, T)$:
$\mathsf{crs} \leftarrow \mathsf{ZK.Gen}(1^\lambda)$
For $1 \le j \le T$: $(\mathsf{vk}_{\mathsf{SIG}}^{(j)}, \mathsf{sk}_{\mathsf{SIG}}^{(j)}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$
$(\mathsf{ek}_{\mathsf{PKE}}, \mathsf{dk}_{\mathsf{PKE}}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$
For $1 \le i \le n$ and $1 \le j \le T$:
$(\mathsf{ek}_i^{(j)}, \mathsf{dk}_i^{(j)}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$; $(\mathsf{vk}_i, \mathsf{sk}_i) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$
$\mathsf{cert}_i^{(j)} \leftarrow \mathsf{SIG.Sign}(\mathsf{sk}_{\mathsf{SIG}}^{(j)}, \langle \mathsf{ek}_i^{(j)}, \mathsf{vk}_i \rangle)$; $\mathsf{grt}[i][j] \leftarrow (\mathsf{dk}_i^{(j)}, \mathsf{vk}_i)$
$\mathsf{gpk} = (\mathsf{crs}, \{\mathsf{vk}_{\mathsf{SIG}}^{(j)}\}_j, \mathsf{ek}_{\mathsf{PKE}})$; $\mathsf{gsk}[i] = (\{\mathsf{ek}_i^{(j)}\}_j, \mathsf{vk}_i, \mathsf{sk}_i, \{\mathsf{cert}_i^{(j)}\}_j)$
$\mathbf{gsk} = \{\mathsf{gsk}[i]\}_i$; $\mathbf{grt} = \{\mathsf{grt}[i][j]\}_{ij}$
Return $(\mathsf{gpk}, \mathbf{gsk}, \mathbf{grt})$

$\mathsf{GS.Sign}(\mathsf{gpk}, j, \mathsf{gsk}[i], m)$:
$(\mathsf{vk}_{\mathrm{ots}}, \mathsf{sk}_{\mathrm{ots}}) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$
$\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{sk}_i, \mathsf{vk}_{\mathrm{ots}})$
$\widetilde{\mathsf{ct}} \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}_i^{(j)}, \sigma; r_1)$
$\mathsf{ct} \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}_{\mathsf{PKE}}, \langle \mathsf{ek}_i^{(j)}, \mathsf{vk}_i, \mathsf{cert}_i^{(j)} \rangle; r_2)$
$\pi \leftarrow \mathsf{ZK.Prove}(\mathsf{crs}, \langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}^{(j)}, \widetilde{\mathsf{ct}}, \mathsf{ct} \rangle, \langle \mathsf{ek}_i^{(j)}, \mathsf{vk}_i, \mathsf{cert}_i^{(j)}, \sigma, r_1, r_2 \rangle)$
$\sigma_{all} \leftarrow \mathsf{OTS.Sign}(\mathsf{sk}_{\mathrm{ots}}, \langle m, \mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi \rangle)$
Return $\Sigma = (\mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi, \sigma_{all})$

$\mathsf{GS.Verify}(\mathsf{gpk}, j, \mathsf{RL}_j, m, \Sigma)$:
If $\mathsf{OTS.Verify}(\mathsf{vk}_{\mathrm{ots}}, \langle m, \mathsf{vk}_{\mathrm{ots}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi \rangle, \sigma_{all}) = 0$, return 0
If $\mathsf{ZK.Verify}(\mathsf{crs}, \langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}^{(j)}, \widetilde{\mathsf{ct}}, \mathsf{ct} \rangle, \pi) = 0$, return 0
For $(\mathsf{dk}, \mathsf{vk}) \in \mathsf{RL}_j$:
If $\mathsf{SIG.Verify}(\mathsf{vk}, m, \mathsf{PKE.Dec}(\mathsf{dk}, \widetilde{\mathsf{ct}})) = 1$, return 0
Return 1

Figure 7: Scheme 2: A Fully Anonymous VLR-GS Scheme with Backward Unlinkability

For the security requirements, Theorems 4.1 to 4.3 hold. Each theorem can be shown as the case of Scheme 1.

Theorem 4.1. *Scheme 2 is correct if the underlying OTS scheme $\mathcal{OTS}$ satisfies correctness, the underlying NIZK proof system $\mathcal{P}_L$ satisfies completeness, and the underlying digital signature scheme $\mathcal{SIG}$ satisfies EUF-CMA security.*

Theorem 4.2. *Scheme 2 satisfies full anonymity if the underlying NIZK proof system $\mathcal{P}_L$ satisfies zero-knowledgeness and the underlying PKE scheme $\mathcal{PKE}$ satisfies IND-CPA security and key privacy.*

Theorem 4.3. *Scheme 2 satisfies traceability if the underlying OTS scheme $\mathcal{OTS}$ satisfies strong EUF-CMA security, the underlying NIZK proof system $\mathcal{P}_L$ satisfies soundness, and the underlying digital signature scheme $\mathcal{SIG}$ satisfies EUF-CMA security.*

A Drawback of Scheme 2. Although Scheme 2 satisfies backward unlinkability, it has one drawback that the sizes of the group public key and the user signing key depend on the number of time periods. This is because the user needs to change the encryption key to encrypt a signature σ for the time period when generating a signature Σ. Therefore, the user i possesses T encryption keys $\mathsf{ek}_i^{(1)}, \ldots, \mathsf{ek}_i^{(T)}$ as a part of the user signing key where T is the number of time periods. Consequently, the size of certificate is also grown.

Since the number of time periods is fixed in the setup phase and the user signing keys are also fixed at the beginning of using the system, it is not necessary to redistribute the group public key and the user signing keys. However, it is still undesirable that the size of these keys depends on the number of time periods. Therefore, we also propose the VLR-GS scheme with backward unlinkability which overcomes the weakness by using an IBE scheme. The description of the scheme is given in the next section.

5 A Fully Anonymous VLR-GS Scheme with Backward Unlinkability and with Constant-Size Keys

In this section, we reduce the sizes of gpk and gsk by employing an IBE scheme. Intuitively, $\mathsf{grt}[i][j]$ is set as a decryption key of the underlying IBE scheme for the identity $i||j$. Moreover, we employ an IBS scheme that allows us to simplify the certification part. Then, a user i has $\mathsf{gsk}[i] = \mathsf{sk}_i$ where sk_i is the signing key of the IBS scheme. The security of Scheme 3 is discussed as with that of Scheme 1. Specifically, the underlying IBE scheme is required to be key-private as we require the underlying PKE scheme in Scheme 1 to satisfy the security.

5.1 Description

The description of Scheme 3 is given in Figure 8. Concretely, we construct a VLR-GS scheme $\Pi_3 = (\mathsf{GS.Gen}, \mathsf{GS.Sign}, \mathsf{GS.Verify})$ from an IBS scheme $\mathcal{IBS} = (\mathsf{IBS.Gen}, \mathsf{IBS.Ext}, \mathsf{IBS.Sign}, \mathsf{IBS.Verify})$, an OTS scheme $\mathcal{OTS} = (\mathsf{OTS.Gen}, \mathsf{OTS.Sign}, \mathsf{OTS.Verify})$, a PKE scheme $\mathcal{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$, an IBE scheme $\mathcal{IBE} = (\mathsf{IBE.Gen}, \mathsf{IBE.Ext}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$, and an NIZK proof system $\mathcal{P}_{\widehat{L}} = (\mathsf{ZK.Gen}, \mathsf{ZK.Prove}, \mathsf{ZK.Verify})$. We say that a statement $x = \langle \mathsf{params}_{\mathsf{IBS}}, \mathsf{params}_{\mathsf{IBE}}, \mathsf{ek}_{\mathsf{PKE}}, j, \mathsf{vk}_{\mathsf{OTS}}, \widetilde{\mathsf{ct}}, \mathsf{ct} \rangle$ and a witness $w = \langle i, \sigma, r_1, r_2 \rangle$ satisfy the relation $R_{\widehat{L}}$ if the following equations hold:

(a) $\widetilde{\mathsf{ct}} = \mathsf{IBE.Enc}(\mathsf{params}_{\mathsf{IBE}}, i||j, \sigma; r_1)$,

(b) $\mathsf{ct} = \mathsf{PKE.Enc}(\mathsf{ek}_{\mathsf{PKE}}, \langle i, \sigma \rangle; r_2)$, and

(c) $\mathsf{IBS.Verify}(\mathsf{params}_{\mathsf{IBS}}, i, \mathsf{vk}_{\mathsf{OTS}}, \sigma) = 1$

- $\mathsf{GS.Gen}(1^\lambda, N, T)$
  - $\mathsf{crs} \leftarrow \mathsf{ZK.Gen}(1^\lambda)$; $(\mathsf{ek}_{\mathsf{PKE}}, \mathsf{dk}_{\mathsf{PKE}}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$
  - $(\mathsf{params}_{\mathsf{IBS}}, \mathsf{msk}_{\mathsf{IBS}}) \leftarrow \mathsf{IBS.Gen}(1^\lambda)$; $(\mathsf{params}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Gen}(1^\lambda)$
  - For $1 \le i \le N$: $\mathsf{sk}_i \leftarrow \mathsf{IBS.Ext}(\mathsf{msk}_{\mathsf{IBS}}, i)$
  - For $1 \le i \le N$ and $1 \le j \le T$: $\mathsf{d}_{i||j} \leftarrow \mathsf{IBE.Ext}(\mathsf{msk}_{\mathsf{IBE}}, i||j)$
  - $\mathsf{gpk} = (\mathsf{crs}, \mathsf{ek}_{\mathsf{PKE}}, \mathsf{params}_{\mathsf{IBS}}, \mathsf{params}_{\mathsf{IBE}})$; $\mathsf{gsk}[i] = \mathsf{sk}_i$; $\mathsf{grt}[i][j] = \mathsf{d}_{i||j}$
  - Return $(\mathsf{gpk}, \{\mathsf{gsk}[i]\}_i, \{\mathsf{grt}[i][j]\}_{ij})$
- $\mathsf{GS.Sign}(\mathsf{gpk}, j, \mathsf{gsk}[i], m)$
  - $(\mathsf{vk}_{\mathsf{OTS}}, \mathsf{sk}_{\mathsf{OTS}}) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$: $\sigma \leftarrow \mathsf{IBS.Sign}(\mathsf{sk}_i, i, \mathsf{vk}_{\mathsf{OTS}})$
  - $\widetilde{\mathsf{ct}} \leftarrow \mathsf{IBE.Enc}(\mathsf{params}_{\mathsf{IBE}}, i||j, \sigma; r_1)$; $\mathsf{ct} \leftarrow \mathsf{PKE.Enc}(\mathsf{ek}_{\mathsf{PKE}}, \langle i, \sigma \rangle; r_2)$
  - $\pi \leftarrow \mathsf{ZK.Prove}(\mathsf{crs}, \langle \mathsf{params}_{\mathsf{IBS}}, \mathsf{params}_{\mathsf{IBE}}, \mathsf{ek}_{\mathsf{PKE}}, j, \mathsf{vk}_{\mathsf{OTS}}, \widetilde{\mathsf{ct}}, \mathsf{ct} \rangle, \langle i, \sigma, r_1, r_2 \rangle)$
  - $\sigma_{all} \leftarrow \mathsf{OTS.Sign}(\mathsf{sk}_{\mathsf{OTS}}, \langle m, \mathsf{vk}_{\mathsf{OTS}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi \rangle)$
  - Return $\Sigma = (\mathsf{vk}_{\mathsf{OTS}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi, \sigma_{all})$
- $\mathsf{GS.Verify}(\mathsf{gpk}, j, \mathsf{RL}_j, m, \Sigma)$
  - If $\mathsf{OTS.Verify}(\mathsf{vk}_{\mathsf{OTS}}, \langle m, \mathsf{vk}_{\mathsf{OTS}}, \widetilde{\mathsf{ct}}, \mathsf{ct}, \pi \rangle, \sigma_{all}) = 0$, return 0
  - If $\mathsf{ZK.Verify}(\mathsf{crs}, \langle \mathsf{params}_{\mathsf{IBS}}, \mathsf{params}_{\mathsf{IBE}}, \mathsf{ek}_{\mathsf{PKE}}, j, \mathsf{vk}_{\mathsf{OTS}}, \widetilde{\mathsf{ct}}, \mathsf{ct} \rangle, \pi) = 0$, return 0
  - For $1 \le i \le N$ such that $\mathsf{grt}[i][j] \in \mathsf{RL}_j$:
    - If $\mathsf{IBS.Verify}(\mathsf{params}_{\mathsf{IBS}}, i, \mathsf{vk}_{\mathsf{OTS}}, \mathsf{IBE.Dec}(\mathsf{params}_{\mathsf{IBE}}, i||j, \mathsf{grt}[i][j], \widetilde{\mathsf{ct}})) = 1$, return 0
  - Else return 1

Figure 8: Scheme 3: A Fully Anonymous VLR-GS Scheme with Backward Unlinkability and with Constant-Size Keys

For the correctness, the following theorem holds.

Theorem 5.1. *Scheme 3 is correct if the underlying OTS scheme $\mathcal{OTS}$ satisfies correctness, if the underlying NIZK proof system $\mathcal{P}_{\widehat{L}}$ satisfies completeness and the underlying IBS scheme $\mathcal{IBS}$ satisfies EUF-CMA security.*

Basically, it can be shown as the case of Scheme 1 and we give a proof sketch. For $\Sigma^* = (\mathsf{vk}^*_{\mathrm{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*, \sigma^*_{all})$ that $\mathcal{A}$ outputs, σ^*_{all} is always a valid signature due to the correctness of $\mathcal{OTS}$, and π^* is a

valid proof due to the completeness of $\mathcal{P}_{\widehat{L}}$. The remaining part is the case that for some $\tilde{i} \in \mathsf{RU}^*$, $\mathsf{IBS.Verify}(\mathsf{params}_{\mathsf{IBS}}, \tilde{i}, {\mathsf{vk}_{\mathsf{OTS}}}^*, \tilde{\sigma}) = 1$ holds where $\tilde{\sigma} \leftarrow \mathsf{IBE.Dec}(\mathsf{params}_{\mathsf{IBE}}, \tilde{i}||j^*, \mathsf{grt}[\tilde{i}][j^*], \widetilde{\mathsf{ct}}^*)$. Since $i^* \notin \mathsf{RU}_{j^*}$, $\tilde{i} \neq i^*$ holds. Thus, $(\tilde{i}, \mathsf{vk}^*_{\mathrm{ots}}, \tilde{\sigma})$ is a forged signature of the IBS scheme.

5.2 Security

Here, we give the intuition of the security of Scheme 3. Basically, it can be shown as the case of Scheme 1 and we give a proof sketch.

Full Anonymity. As with the case of Scheme 1, the user's information is not revealed from a signature by the IND-ID-CPA security of the IBE scheme, the IND-CPA security of the PKE scheme, and the zero-knowledgeness of the NIZK proof system. Formally, the following theorem holds.

Theorem 5.2. *Scheme 2 satisfies full anonymity if the underlying NIZK proof system $\mathcal{P}_{\widehat{L}}$ satisfies zero-knowledgeness, the underlying PKE scheme $\mathcal{PKE}$ satisfies IND-CPA security, and the underlying IBE scheme $\mathcal{IBE}$ satisfies IND-ID-CPA security and key privacy.*

Let $\mathcal{A}$ be an adversary for full anonymity of Π_3. We consider the following sequence of games. Let $\Pr[\mathtt{Suc}_\ell]$ denote the event that $\mathcal{A}$ succeeds in guessing the challenge bit in Game ℓ. Let b be the challenge bit, i_0 and i_1 be the challenge users, and m^* be the challenge message.

Game 0: This is the experiment $\mathrm{Exp}^{anon}_{\Pi_3,\mathcal{A}}(\lambda, n)$ itself. For simplicity, the challenge bit b is chosen at the beginning of the game. This change does not have an effect on the behavior of the adversary $\mathcal{A}$.

Game 1: This game is the same as Game 0, except that the common reference string crs in the group public key gpk, and a proof π^* in the challenge signature Σ^* are computed by using the simulator $\mathcal{S} = (\mathtt{Sim}_1, \mathtt{Sim}_2)$ of the NIZK proof system.

Game 2: In this game, we change the plaintext of the ciphertext ct^* in the challenge signature Σ^*. Concretely, the plaintext $0^{|\langle i_b, \sigma^* \rangle|}$ is encrypted to the ciphertext ct^* instead of $\langle i_b, \sigma^* \rangle$.

Game 3: In this game, we change the plaintext of the ciphertext $\widetilde{\mathsf{ct}}^*$ in the challenge signature Σ^*. Concretely, the plaintext $0^{|\sigma^*|}$ is encrypted to the ciphertext $\widetilde{\mathsf{ct}}^*$ instead of σ^* where $\sigma^* = \mathsf{IBS.Sign}(\mathsf{sk}_{i_b}, i_b, \mathsf{vk}^*_{\mathrm{ots}})$.

Game 4: In this game, we change the encryption key of the ciphertext $\widetilde{\mathsf{ct}}^*$. Concretely, we use the identity $0^{|i||j|}$ to compute $\widetilde{\mathsf{ct}}^*$ instead of using the key $i||j$.

As in Scheme 1, the following lemmas hold.

Lemma 5.1. *There exists a* PPT *algorithm $\mathcal{B}_1$ such that* $\left|\Pr[\mathtt{Suc}_0] - \Pr[\mathtt{Suc}_1]\right| = \mathrm{Adv}^{zk}_{\mathcal{P}_L,\mathcal{B}_1}(\lambda)$.

Lemma 5.2. *There exists a* PPT *algorithm $\mathcal{B}_2$ such that* $\left|\Pr[\mathtt{Suc}_1] - \Pr[\mathtt{Suc}_2]\right| = 2 \cdot \mathrm{Adv}^{ind\text{-}cpa}_{\mathcal{PKE},\mathcal{B}_2}(\lambda)$.

Lemma 5.3. *There exists a* PPT *algorithm $\mathcal{B}_3$ such that* $\left|\Pr[\mathtt{Suc}_2] - \Pr[\mathtt{Suc}_3]\right| = 2 \cdot \mathrm{Adv}^{ind\text{-}id\text{-}cpa}_{\mathcal{IBE},\mathcal{B}_3}(\lambda)$.

Lemma 5.4. *There exists a* PPT *algorithm $\mathcal{B}_4$ such that* $\left|\Pr[\mathtt{Suc}_3] - \Pr[\mathtt{Suc}_4]\right| = 2 \cdot \mathrm{Adv}^{key\text{-}priv}_{\mathcal{IBE},\mathcal{B}_4}(\lambda)$.

In Game 4, the choice of the challenge bit b and the distribution of the challenge signature $\Sigma^* = (\widetilde{\mathsf{ct}}^*, \mathsf{ct}^*, \pi^*)$ are independent. Thus, $\Pr[\mathtt{Suc}_4] = 1/2$ holds. From the above lemmas, putting all together, we get

$$\begin{aligned}\mathrm{Adv}^{anon}_{\Pi_3,\mathcal{A}}(\lambda, n) \leq & \sum_{i=0}^{3} \left|\Pr[\mathtt{Suc}_i] - \Pr[\mathtt{Suc}_{i+1}]\right| + \left|\Pr[\mathtt{Suc}_4] - 1/2\right| \\ = & \mathrm{Adv}^{zk}_{\mathcal{P}_L,\mathcal{B}_1}(\lambda) + 2 \cdot \mathrm{Adv}^{ind\text{-}cpa}_{\mathcal{PKE},\mathcal{B}_2}(\lambda) + 2n \cdot \mathrm{Adv}^{ind\text{-}id\text{-}cpa}_{\mathcal{IBE},\mathcal{B}_3}(\lambda) + 2n \cdot \mathrm{Adv}^{key\text{-}priv}_{\mathcal{IBE},\mathcal{B}_4}(\lambda).\end{aligned}$$

Since the choice of the parameter n and the adversary $\mathcal{A}$ is arbitrary, our scheme Π_1 satisfies full anonymity if the underlying NIZK proof system $\mathcal{P}_L$ satisfies zero-knowledgeness and the underlying IBE scheme $\mathcal{IBE}$ satisfies IND-ID-CPA security and key privacy.

Remark 2. We remark that the underlying IBE scheme is not required to be adaptive-ID secure, and we can employ a selective-ID secure IBE scheme. Unlike IBE, the number of users (and the number of time periods also) is polynomial of the security parameter, and it is sufficient to guess the challenge user with reduction loss $O(n)$ by the reduction algorithm in the security proof. That is, as in the proof of Lemmas 3.3 and 3.4, the reduction algorithm chooses the index $i^* \in [1, n]$ uniform randomly in the proof of Lemmas 5.3 and 5.4. By employing an adaptive-ID secure IBE scheme, the reduction algorithm does not have to guess i^*. Thus, we have removed the n factor from Lemmas 5.3 and 5.4.

Traceability. In terms of traceability, the security of Scheme 3 can be proved in almost the same way as that of Scheme 1. First, the traceability of Scheme 1 does not rely on the security of the PKE scheme. Therefore, if we use an IBE scheme instead of a PKE scheme, it does not influence the security proof of the traceability. Second, we modify the condition **I**. $\langle \mathsf{ek}_{\mathsf{PKE}}, \mathsf{vk}_{\mathsf{SIG}}, \mathsf{vk}^*_{\mathsf{ots}}, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle \notin L$ in the proof of Theorem 3.3 to **I**. $\langle \mathsf{params}_{\mathsf{IBS}}, \mathsf{params}_{\mathsf{IBE}}, \mathsf{ek}_{\mathsf{PKE}}, j^*, \mathsf{vk}_{\mathsf{OTS}}{}^*, \widetilde{\mathsf{ct}}^*, \mathsf{ct}^* \rangle \notin \widehat{L}$. Then, as in the same discussion of the proof of Theorem 3.3, the following theorem holds.

Theorem 5.3. *Scheme 3 satisfies traceability if the underlying OTS scheme $\mathcal{OTS}$ satisfies strong EUF-CMA security, the underlying NIZK proof system $\mathcal{P}_{\widehat{L}}$ satisfies soundness, and the underlying IBS scheme $\mathcal{IBS}$ satisfies EUF-CMA security.*

6 Cryptanalysis of the Perera-Koshiba Scheme

Here, we review the Perera-Koshiba (PK) scheme [41] and show that it does not satisfy full anonymity.

Firstly, we give the description of the PK scheme. This scheme mainly follows the Langlois-Ling-Nguyen-Wang (LLNW) scheme [26], and specifically, the zero-knowledge protocol in the PK scheme is identical to that in the LLNW scheme.

In the following, we use the notation and the algorithms given in the papers [26, 41]. Let n be a security parameter, $N = 2^\ell$ be the maximum number of group members. We fix other parameters as follows:

- Modulus q : $q = \omega(n^2 \log n)$,
- Dimension m : $m \geq 2n \log q$,
- Gaussian Parameter σ : $\sigma = \omega(\sqrt{n \log q \log n})$,
- Integer norm bound β : $\beta = \lceil \sigma \cdot \log m \rceil$,
- Number of decompositions p : $p = \lfloor \log \beta \rfloor + 1$,
- Sequence of integers $\beta_1, \ldots, \beta_p$: $\beta_1 = \lceil \beta/2 \rceil, \beta_2 = \lceil (\beta - \beta_1)/2 \rceil, \beta_3 = \lceil (\beta - \beta_1 - \beta_2)/2 \rceil, \ldots, \beta_p = 1$,
- Number of protocol repetitions λ : $\lambda = \omega(\log n)$.

Let $\mathcal{H}_1 : \{0,1\}^* \to \mathbb{Z}_q^{n \times \ell}$ and $\mathcal{H}_2 : \{0,1\}^* \to \{1,2,3\}^t$ be hash functions modeled as a random oracle. The description of the PK scheme is as follows:

$\mathsf{GS.Gen}(n, N)$**:** Given security parameter n and the number of members N, the $\mathsf{GS.Gen}$ algorithm works as follows:

1. Run the $\mathsf{TrapGen}(n, m, q)$ algorithm to obtain a pair of a matrix and its trapdoor $(\mathbf{A}_0, \mathbf{R})$, where $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$, and randomly sample matrixes $\mathbf{A}_i^b \leftarrow \mathbb{Z}_q^{n \times m}$ for all $1 \leq i \leq \ell$ and all $b \in \{0,1\}$. Then, define the matrix $\mathbf{A} = [\mathbf{A}_0 | \mathbf{A}_1^0 | \mathbf{A}_1^1 | \ldots | \mathbf{A}_\ell^0 | \mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n \times (2\ell+1)m}$.
2. Sample a random vector $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. For a group member with an index $d \in \{0,1\}^\ell$ where $d = d[1] \ldots d[\ell]$, do the following procedure:
   (a) Sample vectors $\mathbf{x}_1^{d[1]}, \ldots, \mathbf{x}_\ell^{d[\ell]} \leftarrow D_{\mathbb{Z}^m, \sigma}$ and compute $\mathbf{z} = \sum_{i=1}^{\ell} \mathbf{A}_i^{d[i]} \cdot \mathbf{x}_i^{d[i]} \mod q$. Set vectors $\mathbf{x}_1^{1-d[1]}, \ldots, \mathbf{x}_\ell^{1-d[\ell]}$ to be zero-vector 0^m.
   (b) Run the $\mathsf{SampleD}$ algorithm and obtain $\mathbf{x}_0 \leftarrow \mathsf{SampleD}(\mathbf{R}, \mathbf{A}_0, \mathbf{u} - \mathbf{z}, \sigma)$. Then define a vector $\mathbf{x}^{(d)} = (\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1) \in \mathbb{Z}_q^{(2\ell+1)m}$.

If $\|\mathbf{x}^{(d)}\|_\infty > \beta$, go back to the step (a). Let the secret key of the member be $\mathsf{gsk}[d] = \mathbf{x}^{(d)}$, and the revocation token of the member be $\mathsf{grt}[d] = \mathbf{A}_0 \cdot \mathbf{x}_0 \bmod q$.[7]

4. Output $(\mathsf{gpk}, \mathbf{gsk}, \mathbf{grt})$ where $\mathsf{gpk} = (\mathbf{A}, \mathbf{u}, \mathbf{B})$, $\mathbf{gsk} = \{\mathsf{gsk}[1], \ldots, \mathsf{gsk}[N]\}$, and $\mathbf{grt} = \{\mathsf{grt}[1], \ldots, \mathsf{grt}[N]\}$.

$\mathsf{GS.Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M)$**:** Given the pubic key gpk, the member's secret key $\mathsf{gsk}[d]$, and a message M, the $\mathsf{GS.Sign}$ algorithm works as follows:

1. Generate a key pair of a one-time signature scheme $(\mathsf{ovk}, \mathsf{osk})$.
2. Encrypt the index d as follows:
   (a) Compute $\mathbf{G} = \mathcal{H}_1(\mathsf{ovk})$.
   (b) Sample $\mathbf{s} \leftarrow \chi^n$, $\mathbf{e}_1 \leftarrow \chi^m$, and $\mathbf{e}_2 \leftarrow \chi^\ell$.
   (c) Compute $c_1 \leftarrow \mathbf{B}^{\mathrm{T}}\mathbf{s} + \mathbf{e}_1$ and $c_2 \leftarrow \mathbf{G}^{\mathrm{T}}\mathbf{s} + \mathbf{e}_2 + \lfloor 2/q \rfloor d$.
3. Run $t = \omega(\log n)$ times the zero-knowledge protocol [26] with the public parameter $(\mathbf{A}, \mathbf{u})$ and the witness $\mathbf{x}^{(d)}$. Then, compute a triple $\Pi = (\{\mathrm{CMT}^{(k)}\}_{k=1}^t, \{\mathrm{Ch}^{(k)}\}_{k=1}^t, \{\mathrm{RSP}^{(k)}\}_{k=1}^t)$ where $\{\mathrm{Ch}^{(k)}\}_{k=1}^t = \mathcal{H}_2(M, \{\mathrm{CMT}^{(k)}\}_{k=1}^t, c_1, c_2)$.
4. Compute a signature $\mathsf{sig} \leftarrow \mathsf{OSig}(\mathsf{osk}, (c_1, c_2, \Pi))$.
5. Output a signature $\Sigma = (\mathsf{ovk}, c_1, c_2, \Pi, \mathsf{sig})$.

$\mathsf{GS.Verify}(\mathsf{gpk}, \mathsf{RL}, M, \Sigma)$**:** Given a pubic key gpk, a revocation list RL, a message M, and a signature Σ, the $\mathsf{GS.Verify}$ algorithm works as follows:

1. Parse the signature Σ as $(\mathsf{ovk}, c_1, c_2, \Pi, \mathsf{sig})$.
2. If $\mathsf{OVer}(\mathsf{ovk}, (c_1, c_2, \Pi), \mathsf{sig}) = 0$, output 0.
3. For the proof $\Pi = (\{\mathrm{CMT}^{(k)}\}_{k=1}^t, \{\mathrm{Ch}^{(k)}\}_{k=1}^t, \{\mathrm{RSP}^{(k)}\}_{k=1}^t)$, compute the values $\{\overline{\mathrm{Ch}}^{(k)}\}_{k=1}^t \leftarrow \mathcal{H}_2(M, \{\mathrm{CMT}^{(k)}\}_{k=1}^t, c_1, c_2)$. Then, if it holds that $\{\overline{\mathrm{Ch}}^{(k)}\}_{k=1}^t \neq \{\mathrm{Ch}^{(k)}\}_{k=1}^t$, output 0.
4. Check the validity of the response $\{\mathrm{RSP}^{(k)}\}_{k=1}^t$ by running the verification algorithm of the zero-knowledge protocol [26]. That is, for $1 \leq k \leq t$, check whether $\mathrm{RSP}^{(k)}$ is the valid response for the commitment $\mathrm{CMT}^{(k)}$ and the challenge $\mathrm{Ch}^{(k)}$. If some response $\mathrm{RSP}^{(k)}$ is invalid, output 0. Otherwise output 1.

As we mentioned, if a VLR-GS scheme has the structure allowing the revocation token to be constructed from the corresponding user's signing key, the scheme does not satisfy full anonymity. The PK scheme has this structure. More precisely, in the PK scheme, the revocation token of a user with an index d is represented as $\mathsf{grt}[d] = \mathbf{A}_0 \cdot \mathbf{x}_0 \bmod q$. Since the matrix $\mathbf{A} = [\mathbf{A}_0 | \mathbf{A}_1^0 | \mathbf{A}_1^1 | \ldots | \mathbf{A}_\ell^0 | \mathbf{A}_\ell^1]$ is a part of the group public key gpk, and the signing key $\mathsf{gsk}[d]$ is denoted by the vector $\mathbf{x}^{(d)} = (\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1)$, the revocation token can be computed by using the corresponding signing key and the public values. Therefore, in the full anonymity game of the PK scheme, the adversary can compute revocation tokens of all users. Thus, for an any valid signature Σ on a message M, the adversary can identify its signer by computing values $\mathsf{GS.Verify}(\mathsf{gpk}, \{\mathsf{grt}[d]\}, M, \Sigma)$ for all $d \in [1, N]$. This is because if $\mathsf{GS.Verify}(\mathsf{gpk}, \{\mathsf{grt}[d]\}, M, \Sigma) = 0$ holds whereas $\mathsf{GS.Verify}(\mathsf{gpk}, \mathsf{RL}, M, \Sigma) = 1$, this indicates that a user d is the signer of Σ.

Here, we fully describe our attack against the full anonymity of the PK scheme. In the following, we use the notation and the algorithms given in the papers [26, 41]. Firstly, we review how to generate the challenge signature Σ^*. Let d_0 and d_1 be the challenge users, $\Sigma^* = (\mathsf{ovk}^*, c_1^*, c_2^*, \Pi^*, \mathsf{sig}^*)$ be the challenge signature on the message M^*, and b be the challenge bit. Let $\mathsf{gsk}[d_b] = \mathbf{x}^{(d_b)} = (\mathbf{x}_{0,b} \| \mathbf{x}_{1,b}^0 \| \mathbf{x}_{1,b}^1 \| \ldots \| \mathbf{x}_{\ell,b}^0 \| \mathbf{x}_{\ell,b}^1)$. Let COM be the Kawachi-Tanaka-Xagawa commitment scheme [24]. Let $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}(\mathbf{A})$ and $\mathbf{z}_1^*, \ldots, \mathbf{z}_p^* \leftarrow \mathsf{WitnessDE}(\mathbf{x}^{(d_b)})$. Now, we consider the case that it holds $\mathrm{Ch}^{(1)} = 2$ which happens with the probability 1/3. In this case, the challenge signature Σ^* is computed as follows:

[7] According to the paper [26], it is required that the revocation tokens of two different users must be different. Namely, if $d \neq \hat{d}$, it holds that $\mathsf{grt}[d] \neq \mathsf{grt}[\hat{d}]$.

1. Generate a key pair of a one-time signature scheme $(\mathsf{ovk}^*, \mathsf{osk}^*)$.
2. Encrypt the index d as follows:
   (a) Compute $\mathbf{G}^* = \mathcal{H}_1(\mathsf{ovk}^*)$.
   (b) Sample $\mathbf{s}^* \leftarrow \chi^n$, $\mathbf{e}_1^* \leftarrow \chi^m$, and $\mathbf{e}_2^* \leftarrow \chi^\ell$.
   (c) Compute $c_1^* \leftarrow \mathbf{B}^{\mathrm{T}}\mathbf{s}^* + \mathbf{e}_1^*$ and $c_2^* \leftarrow \mathbf{G}^{*\mathrm{T}}\mathbf{s}^* + \mathbf{e}_2^* + \lfloor 2/q \rfloor d_b$.
3. Run $t = \omega(\log n)$ times the zero-knowledge protocol [26] with the public parameter $(\mathbf{A}, \mathbf{u})$ and the witness $\mathbf{x}^{(d_b)}$. Then, compute a proof Π^*. Specifically, the elements $(\mathrm{CMT}^{(1)}, \mathrm{Ch}^{(1)}, \mathrm{RSP}^{(1)})$ where $\mathrm{Ch}^{(1)} = 2$ is computed as follows:
   (a) Sample $e^{(1)} \xleftarrow{\$} \{0,1\}^\ell$, p permutations $\pi_1^{(1)}, \ldots, \pi_p^{(1)} \xleftarrow{\$} \mathcal{S}$, and p vectors $\mathbf{r}_1^{(1)}, \ldots, \mathbf{r}_p^{(1)} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)3m}$ randomly where $\mathcal{S}$ is the set of all permutations that keep the arrangement of the blocks. For each $1 \le j \le p$, let $\mathbf{r}_{j,0}^{(1)}$ be the first m elements of $\mathbf{r}_j^{(1)}$.
   (b) Set
   $$\begin{aligned} c_0^{(1)} &= \mathsf{COM}(e^{(1)},\ \{\pi_j^{(1)}\}_{j=1}^p,\ \mathbf{A}_0 \cdot (\textstyle\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}^{(1)}) \bmod q), \\ c_1^{(1)} &= \mathsf{COM}(e^{(1)},\ \{\pi_j^{(1)}\}_{j=1}^p,\ \mathbf{A}^* \cdot (\textstyle\sum_{j=1}^p \beta_j \cdot \mathbf{r}_j^{(1)}) \bmod q), \\ c_2^{(1)} &= \mathsf{COM}(\{T_{e^{(1)}} \circ \pi_j(\mathbf{r}_j^{(1)})\}_{j=1}^p), \\ c_3^{(1)} &= \mathsf{COM}(\{T_{e^{(1)}} \circ \pi_j(\mathbf{z}_j^* + \mathbf{r}_j^{(1)})\}_{j=1}^p). \end{aligned}$$
   (c) Set $\mathrm{RSP}^{(1)} = (e^{(1)}, \{\pi_j\}_{j=1}^p, \{\mathbf{s}_j^{(1)}\}_{j=1}^p)$ where $\mathbf{s}_j^{(1)} = \mathbf{z}_j^* + \mathbf{r}_j^{(1)}$.
4. Compute a signature $\mathsf{sig}^* \leftarrow \mathsf{OSig}(\mathsf{osk}^*, (c_1^*, c_2^*, \Pi^*))$.
5. Set $\Sigma^* = (\mathsf{ovk}^*, c_1^*, c_2^*, \Pi^*, \mathsf{sig}^*)$.

Secondly, we construct the adversary $\mathcal{A}$ who breaks the full anonymity of the PK scheme. In full anonymity games, the adversary is allowed to possess all user signing keys, that is, the adversary knows $\mathsf{gsk}[d_0] = (\mathbf{x}_{0,0} \| \mathbf{x}_{1,0}^0 \| \mathbf{x}_{1,0}^1 \| \ldots \| \mathbf{x}_{\ell,0}^0 \| \mathbf{x}_{\ell,0}^1)$ and $\mathsf{gsk}[d_1] = (\mathbf{x}_{0,1} \| \mathbf{x}_{1,1}^0 \| \mathbf{x}_{1,1}^1 \| \ldots \| \mathbf{x}_{\ell,1}^0 \| \mathbf{x}_{\ell,1}^1)$. For the challenge signature Σ^*, the adversary $\mathcal{A}$ operates as follows:

1. Compute $\mathbf{w}^* = \mathbf{A}_0 \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}^{(1)}) - \mathbf{A}_0 \cdot \mathbf{x}_{0,0} \bmod q$.
2. If $c_0^{(1)} = \mathsf{COM}(e^{(1)}, \{\pi_j^{(1)}\}_{j=1}^p, \mathbf{w}^*)$, output $\widetilde{b} = 0$. Otherwise output $\widetilde{b} = 1$.

Finally, we show that the adversary $\mathcal{A}$ break the full anonymity of the PK scheme. In terms of the vector $\mathbf{w}^*$, it holds that

$$\begin{aligned} \mathbf{w}^* &= \mathbf{A}_0 \cdot \Big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}^{(1)}\Big) - \mathbf{A}_0 \cdot \mathbf{x}_{0,0} \bmod q \\ &= \mathbf{A}_0 \cdot \Big(\sum_{j=1}^p \beta_j \cdot (\mathbf{z}_{j,0}^* + \mathbf{r}_{j,0}^{(1)})\Big) - \mathbf{A}_0 \cdot \mathbf{x}_{0,0} \bmod q \\ &= \mathbf{A}_0 \cdot \sum_{j=1}^p \beta_j \cdot \mathbf{z}_{j,0}^* + \mathbf{A}_0 \cdot \sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}^{(1)} - \mathbf{A}_0 \cdot \mathbf{x}_{0,0} \bmod q \\ &= \mathbf{A}_0 \cdot \mathbf{x}_{0,b} + \mathbf{A}_0 \cdot \sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}^{(1)} - \mathbf{A}_0 \cdot \mathbf{x}_{0,0} \bmod q. \end{aligned}$$

If $b = 0$, it holds that $\mathbf{w}^* = \mathbf{A}_0 \cdot \sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}^{(1)} \bmod q$. Therefore, we have $c_0^{(1)} = \mathsf{COM}(e^{(1)}, \{\pi_j^{(1)}\}_{j=1}^p, \mathbf{A}_0 \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}^{(1)}) \bmod q) = \mathsf{COM}(e^{(1)}, \{\pi_j^{(1)}\}_{j=1}^p, \mathbf{w}^*)$, and $\mathcal{A}$ outputs $\widetilde{b} = 0$. Thus, $b = \widetilde{b}$ holds. On the other hand, if $b = 1$, it holds that $\mathbf{w}^* = \mathbf{A}_0 \cdot \sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}^{(1)} + \mathbf{A}_0 \cdot \mathbf{x}_{0,1} - \mathbf{A}_0 \cdot \mathbf{x}_{0,0} \bmod q$. Since it holds that $\mathsf{grt}[d_0] = \mathbf{A}_0 \cdot \mathbf{x}_{0,0} \neq \mathbf{A}_0 \cdot \mathbf{x}_{0,1} = \mathsf{grt}[d_1]$, we get $\mathbf{w}^* \neq \mathbf{A}_0 \cdot \sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}^{(1)} \bmod q$. Therefore, from the binding property of the commitment scheme, we have $c_0^{(1)} = \mathsf{COM}(e^{(1)}, \{\pi_j^{(1)}\}_{j=1}^p, \mathbf{A}_0 \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}^{(1)}) \bmod q) \neq \mathsf{COM}(e^{(1)}, \{\pi_j^{(1)}\}_{j=1}^p, \mathbf{w}^*)$. Thus, the adversary outputs $\widetilde{b} = 1$, that is, $b = \widetilde{b}$. Hence, the adversary $\mathcal{A}$ breaks the full anonymity of the PK scheme.

7 Conclusion

In this paper, for the first time we showed that full anonymity can be achieved in the VLR setting. We proposed three schemes that show a minimum requirement for achieving full anonymity (Scheme 1), a minimum requirement for achieving backward unlinkability (Scheme 2), and constant-size keys can be realized in a fully anonymous VLR-GS scheme with backward unlinkability by additionally employing IBE (Scheme 3). Since our schemes employ general NIZK proof systems, proposing an efficient instantiation, by, for example, Groth-Sahai proofs [22], is an open problem of this paper.

References

[1] M. Abdalla, M. Bellare, and G. Neven. Robust encryption. In *TCC*, pages 480–497, 2010.

[2] M. Abdalla and B. Warinschi. On the minimal assumptions of group signature schemes. In *ICICS*, pages 1–13, 2004.

[3] M. Backes, L. Hanzlik, and J. Schneider-Bensch. Membership privacy for fully dynamic group signatures. In *ACM CCS*, 2019.

[4] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT*, pages 566–582, 2001.

[5] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT*, pages 614–629, 2003.

[6] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA*, pages 136–153, 2005.

[7] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO*, pages 41–55, 2004.

[8] D. Boneh, P. A. Papakonstantinou, C. Rackoff, Y. Vahlis, and B. Waters. On the impossibility of basing identity based encryption on trapdoor permutations. In *IEEE FOCS*, pages 283–292, 2008.

[9] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *CCS*, pages 168–177, 2004.

[10] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, and J. Groth. Foundations of fully dynamic group signatures. In *ACNS*, pages 117–136, 2016.

[11] J. Camenisch and J. Groth. Group signatures: Better efficiency and new theoretical aspects. In *SCN*, pages 120–133, 2004.

[12] S. Canard, G. Fuchsbauer, A. Gouget, and F. Laguillaumie. Plaintext-checkable encryption. In *CT-RSA*, pages 332–348, 2012.

[13] D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.

[14] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

[15] C. Delerablée and D. Pointcheval. Dynamic fully anonymous short group signatures. In *VIETCRYPT*, pages 193–210, 2006.

[16] D. Derler and D. Slamanig. Highly-efficient fully-anonymous dynamic group signatures. In *ASIACCS*, pages 551–565, 2018.

[17] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, pages 10–18, 1984.

[18] K. Emura, G. Hanaoka, Y. Sakai, and J. C. N. Schuldt. Group signature implies public-key encryption with non-interactive opening. *Int. J. Inf. Sec.*, 13(1):51–62, 2014.

[19] K. Emura and T. Hayashi. A revocable group signature scheme with scalability from simple assumptions. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 103-A(1):125–140, 2020.

[20] J. Furukawa and H. Imai. An efficient group signature scheme from bilinear maps. In *ACISP*, pages 455–467, 2005.

[21] J. Groth. Fully anonymous group signatures without random oracles. In *ASIACRYPT*, pages 164–180, 2007.

[22] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, pages 415–432, 2008.

[23] A. Ishida, Y. Sakai, K. Emura, G. Hanaoka, and K. Tanaka. Fully anonymous group signature with verifier-local revocation. In *SCN*, pages 23–42, 2018.

[24] A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT*, pages 372–389, 2008.

[25] F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *ASIACRYPT*, pages 41–61, 2013.

[26] A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-based group signature scheme with verifier-local revocation. In *PKC*, pages 345–361, 2014.

[27] B. Libert, F. Mouhartem, T. Peters, and M. Yung. Practical "signatures with efficient protocols" from simple assumptions. In X. Chen, X. Wang, and X. Huang, editors, *ASIACCS*, pages 511–522, 2016.

[28] B. Libert, T. Peters, and M. Yung. Group signatures with almost-for-free revocation. In *CRYPTO*, pages 571–589, 2012.

[29] B. Libert, T. Peters, and M. Yung. Scalable group signatures with revocation. In *EUROCRYPT*, pages 609–627, 2012.

[30] B. Libert, T. Peters, and M. Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In *CRYPTO*, pages 296–316, 2015.

[31] B. Libert and D. Vergnaud. Group signatures with verifier-local revocation and backward unlinkability in the standard model. In *CANS*, pages 498–517, 2009.

[32] S. Ling, K. Nguyen, H. Wang, and Y. Xu. Lattice-based group signatures: Achieving full dynamicity (and deniability) with ease. *Theoretical Computer Science*, 783:71–94, 2019.

[33] S. Ma and Q. Huang. CCA-Almost-Full Anonymous Group Signature with Verifier Local Revocation in the Standard Model. *The Computer Journal*, 2020.

[34] T. Nakanishi, H. Fujii, Y. Hira, and N. Funabiki. Revocable group signature schemes with constant costs for signing and verifying. In *PKC*, pages 463–480, 2009.

[35] T. Nakanishi and N. Funabiki. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In *ASIACRYPT*, pages 533–548, 2005.

[36] T. Nakanishi and N. Funabiki. A short verifier-local revocation group signature scheme with backward unlinkability. In *IWSEC*, pages 17–32, 2006.

[37] L. Nguyen. Accumulators from bilinear pairings and applications. In *CT-RSA*, pages 275–292, 2005.

[38] P. Q. Nguyen, J. Zhang, and Z. Zhang. Simpler efficient group signatures from lattices. In *PKC*, pages 401–426, 2015.

[39] K. Ohara, K. Emura, G. Hanaoka, A. Ishida, K. Ohta, and Y. Sakai. Shortening the Libert-Peters-Yung revocable group signature scheme by using the random oracle methodology. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 102-A(9):1101–1117, 2019.

[40] G. Ohtake, A. Fujii, G. Hanaoka, and K. Ogawa. On the theoretical gap between group signatures with and without unlinkability. In *AFRICACRYPT*, pages 149–166, 2009.

[41] M. N. S. Perera and T. Koshiba. Fully secure lattice-based group signatures with verifier-local revocation. In *AINA*, pages 795–802, 2017.

[42] M. N. S. Perera and T. Koshiba. Achieving almost-full security for lattice-based fully dynamic group signatures with verifier-local revocation. In *ISPEC*, pages 229–247, 2018.

[43] M. N. S. Perera and T. Koshiba. Achieving full security for lattice-based group signatures with verifier-local revocation. In *ICICS*, pages 287–302, 2018.

[44] M. N. S. Perera and T. Koshiba. Achieving strong security and member registration for lattice-based group signature scheme with verifier-local revocation. *J. Internet Serv. Inf. Secur.*, 8(4):1–15, 2018.

[45] M. N. S. Perera and T. Koshiba. Achieving strong security and verifier-local revocation for dynamic group signatures from lattice assumptions. In *STM*, pages 3–19, 2018.

[46] M. N. S. Perera and T. Koshiba. Almost-fully secured fully dynamic group signatures with efficient verifier-local revocation and time-bound keys. In *IDCS*, pages 134–147, 2018.

[47] M. N. S. Perera and T. Koshiba. Zero-knowledge proof for lattice-based group signature schemes with verifier-local revocation. In *NBiS*, pages 772–782, 2018.

[48] M. N. S. Perera, T. Nakamura, M. Hashimoto, and H. Yokoyama. Traceable and fully anonymous attribute based group signature scheme with verifier local revocation from lattices. In *NSS*, pages 675–684, 2019.

[49] M. N. S. Perera, T. Nakamura, M. Hashimoto, and H. Yokoyama. Zero-knowledge proof system for fully anonymous attribute based group signatures from lattices with VLR. In *WISA*, pages 126–140, 2019.

[50] L. Wei and J. Liu. Shorter verifier-local revocation group signature with backward unlinkability. In *Pairing*, pages 136–146, 2010.

[51] Y. Zhang, X. Liu, Y. Hu, Q. Zhang, and H. Jia. Lattice-based group signatures with verifier-local revocation: Achieving shorter key-sizes and explicit traceability with ease. In *CANS*, pages 120–140, 2019.

[52] Y. Zhang, X. Liu, Y. Hu, Q. Zhang, and H. Jia. Cryptanalysis of two fully anonymous attribute-based group signature schemes with verifier-local revocation from lattices. In *WISA*, pages 334–346, 2020.

[53] Y. Zhang, X. Liu, Y. Yin, Q. Zhang, and H. Jia. On new zero-knowledge proofs for fully anonymous lattice-based group signature scheme with verifier-local revocation. In *ACNS Workshop on Secure Cryptographic Implementation*, pages 381–399, 2020.

[54] S. Zhou and D. Lin. Shorter verifier-local revocation group signatures from bilinear maps. In *CANS*, pages 126–143, 2006.