On the Traceability of Group Signatures: Uncorrupted User Must Exist

Keita Emura*[1,2]

[1.]Kanazawa University, Japan.
[2.]AIST, Japan.

December 13, 2024

Abstract

Group signature (GS) is a well-known cryptographic primitive providing anonymity and traceability. Several implication results have been given by mainly focusing on the several security levels of anonymity, e.g., fully anonymous GS implies public key encryption (PKE) and selfless anonymous GS can be constructed from one-way functions and non-interactive zero knowledge poofs, and so on. In this paper, we explore an winning condition of full traceability: an adversary is required to produce a valid group signature whose opening result is an uncorrupted user. We demonstrate a generic construction of GS secure in the Bellare-Micciancio-Warinschi (BMW) model except the above condition from PKE only. We emphasize that the proposed construction is quite artificial and meaningless in practice because the verification algorithm always outputs 1 regardless of the input. This result suggests us the winning condition is essential in full traceability, i.e., an uncorrupted user must exist. We also explore a public verifiability of GS-based PKE scheme and introduce a new formal security definition of public verifiability by following BUFF (Beyond UnForgeability Features) security. Our definition guarantees that the decryption result of a valid cyphertext is in the message space specified by the public key. We show that the GS-based PKE scheme is publicly verifiable if the underlying GS scheme is fully traceable.

1 Introduction

1.1 Group Signatures and Security Models

Group signatures (GS) [13] is an extension of digital signatures. Briefly, a user generates a signature on a message and the verifier checks the validity of the signature without identifying the user. A special authority called the opener can identify the signer. Two security notions are mainly considered, full anonymity and full traceability [5]. Informally, the anonymity guarantees that nobody, except the opener, can distinguish whether two signatures are generated by the same user or not (i.e., it implies unlinkability of signers). If the anonymity holds even when an adversary obtains all signing keys including the challenge users' ones, then it is called full anonymity. On the contrary, if an adversary is not allowed to obtain the signing keys of challenge users, then it is called selfless anonymity. The traceability guarantees that if a signature is valid, then the corresponding signer is always identified by the opener. After the seminal work by Bellare-Micciancio-Warinschi

*k-emura@se.kanazawa-u.ac.jp

(BMW) [5], that gave formal definitions of full anonymity and full traceability in a static group setting (i.e., all users' signing keys are generated in the setup phase, and no new user can join the group after the setup phase), many extensions in terms of security and functionalities have been provided. To name a few, GS in a dynamic group setting [7, 25] (including authorities separation between the opener and the issuer), considering the opening soundness [37], revocable GSs [9,22,26, 27,27,29,30], and fully dynamic GS [10] where users can join or leave the group after the setup phase. Currently, the security model of fully dynamic GS given by Bootle et al. [10] is the most enhanced one in terms of security and dynamicity. The fully dynamic model is recognized as a standard security definition in the GS area, and is instantiated by several complexity assumptions [8, 11, 28]. The basic difference between the static group model and the dynamic group model is how to generate a signing key. In the static group model (the BMW model), the group manager generates a signing key and issues it to a user. In the dynamic group model, the issuer and a user run an interactive key issuing protocol. One important fact is that the group manager knows all signing keys in the static group model whereas the issuer knows a part of signing key only in the dynamic group setting. Thus, in addition to a signing key that is known by the issuer, a user uses own secret key for generating a group signature. This setting allows us to define a security notion from an another perspective, i.e., non-frameability. Non-frameability guarantees that the rest of the group as well as the issuer (and the opener) are fully corrupt, they cannot falsely attribute a signature to an honest user who did not produce it.

1.2 Implication Results among GS and Other Cryptographic Primitives

In the BMW paper [5], a generic construction of GS secure in the BMW model was given from public key encryption (PKE), signatures, and non-interactive zero knowledge (NIZK) proofs. A generic construction of GS secure in the Bellare-Shi-Zhang (BSZ) model from the same building blocks was proposed [7] and it was shown that the generic construction provides the opening soundness [37]. As the opposite direction, Abdalla and Warinschi [1] showed that fully anonymous GS implies PKE. Intuitively, a public key consists of $(\mathsf{gpk}, \mathsf{gsigk}_1, \mathsf{gsigk}_2)$ where gpk is a group public key and $(\mathsf{gsigk}_1, \mathsf{gsigk}_2)$ are two signing keys. A ciphertext of a plaintext $m \in \{0, 1\}$ is a group signature generated by gsigk_{m+1}. The opening key ok is set as a decryption key since it determines which signing key was used for generating a group signature. The anonymity hides information of which signing key was used for generating the signature. Moreover, due to full anonymity, the anonymity holds even if signing keys $(\mathsf{gsigk}_1, \mathsf{gsigk}_2)$ are publicly opened. This construction methodology, revealing signing keys as a public key, was employed to show other implications. Emura et al. [19] showed that GS secure in the BSZ model (with the opening soundness) implies public key encryption with non-interactive opening (PKENO) [15] and Sakai et al. [35] showed that GS with message-dependent opening (GS-MDO) implies identity-based encryption (IBE). Sakai et al. [36] utilized Nakanishi et al.'s revocable GS scheme [29] to construct a PKE scheme allowing a message space restriction. Here, revocation in GS allows us to revoke signing keys. Intuitively revoking a signing key in GS means that restricting a message space in the GS-based PKE context since $(\mathsf{gsigk}_1, \mathsf{gsigk}_2)$ specifies the message space $\{0, 1\}$.

The implication result given by Abdalla and Warinschi [1] demonstrates that PKE is an essential building block for constructing fully anonymous GS.[1] In other words, GS can be constructed from weaker primitives if full anonymity is not required. Camenisch and Groth [12] showed that selfless anonymous GS can be constructed from one-way functions (OWF) and NIZK. Katsumata and Yamada [24] showed that NIZK is not necessary to construct selfless anonymous GS.[2] Ohtake et

[1] Bellare and Fuchsbauer [4] showed that GS can be constructed from policy-based signatures and PKE.

[2] They gave a generic construction of selfless anonymous GS from indexed attribute-based signatures (ABS).

al. [31] showed that if unlinkability (which is implied by full anonymity) is not required, then GS can be constructed from identity-based signatures (which can be constructed from the standard signatures [6], and thus from OWF [34]). To sum up, whether the anonymity holds or not when signing keys are exposed is the essential condition to separate the anonymity.

From now on, we focus on the BMW model in this paper with the following three reasons. First, it contains all essential security and functionalities of GS, full anonymity and full traceability, and is one of the most widely accepted definition as mentioned in [24]. Second, the dynamic aspects of the group signature scheme are actually not required to construct public key encryption primitives from GS [1,19,35]. For example, the GS-based PKENO construction [19] employed the BSZ model which considers a dynamic group setting, the key generation algorithm of PKENO internally runs the interactive join protocol by himself. Third, non-freamability is also not required to construct public key encryption primitives from GS [1,19,35] since all signing keys including user secret keys need to be set as a public key and thus there is no way to employ non-freamability. Of course, we do not deny any possibility to construct a cryptographic primitive including an interactive procedure from GS and then it may require the dynamic group setting as an essential condition.

1.3 Full Traceability, Revisited

So far, the above mentioned implication results focused on the (several security levels of) anonymity, and did not focus on the traceability. One exception is the GS-based PKENO construction [19]: If an untraceable but a valid signature is contained in a prove query,[3] then the prove oracle cannot respond the query since it is not publicly detectable. The traceability is employed to exclude the case. However, the result does not answer what the essential condition of the traceability is.

Informally, three winning conditions are considered in the definition of full traceability in the BMW model [5]. Let n be the number of users. gpk and ok are given to an adversary $\mathcal{A}$. Moreover, $\mathcal{A}$ is allowed to issue corrupt queries $i \in [n]$ that obtains gsigk_i where $[n] = \{1, 2, \ldots, n\}$. $\mathcal{A}$ is also allowed to issue signing queries (i, M) where $i \in [n]$. Finally $\mathcal{A}$ outputs a pair of a group signature and a message (Σ^*, M^*). Let i be the opening result of (Σ^*, M^*). It is required that (Σ^*, M^*) is accepted by the verification algorithm. $\mathcal{A}$ wins if one of the following three conditions holds.

1. $i = \perp$.
2. $i \neq \perp \wedge i \notin [n]$.
3. $i \in [n]$, i is not corrupted by $\mathcal{A}$, and (i, M^*) is not sent as a signing query.

Due to the definition, full traceability ensures that a valid group signature is always opened correctly. More clearly, since an adversary colludes with some users, it ensures that no colluding set of users can produce group signatures that cannot be opened, or group signatures that cannot be traced back to some user of the coalition. We remark that, as a well-known folklore, a GS scheme without the opening functionality (i.e., without traceability) can be constructed by sharing a signing key of a signature scheme among all users. That is, for a verification and signing key pair of a signature scheme $(\mathsf{verk}, \mathsf{sigk}) \leftarrow \mathsf{Sig.KeyGen}(1^\lambda)$, set $\mathsf{gpk} = \mathsf{verk}$, and $\mathsf{gsigk}_i = (\mathsf{sigk}, i)$ for $i \in [n]$. Then,

Though it can be instantiated from the standard LWE and SIS assumptions, precisely, it is not clear whether indexed ABS is weaker than PKE or not, to the best of our knowledge.

[3] In the syntax of PKENO, the Prove algorithm, that takes a secret key and a ciphertext, outputs a proof π. The verification algorithm, that takes a ciphertext, a plaintext (which may be $\perp$), and a proof, outputs 1 if the decryption result of the ciphertext is the plaintext. In their GS-based PKENO construction, ok is set as π when a ciphertext is valid but the decryption result is $\perp$. Because the reduction algorithm does not know ok, the traceability is employed to exclude the case, and is not required to respond a decryption query.

full anonymity holds since $\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sigk}, M)$ does not contain information of i (here, M is an arbitrary signed message). However, this folklore does not answer what the essential condition of full traceability is. We explore this research question by analyzing the winning conditions, separately.

1.4 Our Contribution

In this paper, we focus on the last winning condition of full traceability: an adversary is required to produce a valid group signature whose opening result is an uncorrupted user. We demonstrate a generic construction of GS secure in the BMW model *except the last winning condition of full traceability* from PKE only. Especially, the construction provides full anonymity without NIZK whereas previous NIZK-free constructions (except the above folklore) considered a weaker notion of anonymity, selfless anonymity [24] or linkability/pseudonymity [31]. We emphasize that the proposed construction is quite artificial and meaningless in practice because the verification algorithm always outputs 1 regardless of the input and the opening algorithm outputs a user index $i' \in [n]$ if the opening result of a group signature i is either $i = \perp$ or $i \notin [n]$. In other words, GS is meaningless in practice if we do not consider the last winning condition of full traceability. The main observation here is the first and second winning conditions can be considered in the opening algorithm regardless of the behavior of the adversary whereas the last condition depends on the behavior of the adversary, i.e., who will be corrupted by the adversary. This result suggests us the last winning condition is essential in full traceability.

As the second result of this paper, we revisited public verifiability of PKE. As a folklore, a GS-based PKE scheme provides public verifiability since a ciphertext is a group signature. Though Ohtake et al. [31] mentioned that "*It remarks that the PKE scheme, which is constructed with the above method, has public verifiability. Namely, anyone can verify validity of a ciphertext by using* gpk.", they did not give a formal treatment of the public verifiability. Intuitively, due to the traceability of the underlying GS, the public verifiability here seems to guarantee that a valid ciphertext is always decryptable. That is, when the GS scheme is instantiated by $n = 2$ and $(\mathsf{gpk}, \mathsf{gsigk}_1, \mathsf{gsigk}_2)$ is set as a public key of the PKE scheme, then it is expected that the decryption result of a valid ciphertext is not $\perp$ and is in a message space $\{0, 1\}$. Here, the last winning condition does not appear because all signing keys are revealed (i.e., under full corruption). Due to our first result, the public verifiability can be realized via the artificial construction: the verification algorithm outputs always 1 regardless of the input, and if the decryption result is either $\perp$ or is not in $\{0, 1\}$, then the decryption algorithm outputs 0 or 1. To explore what public verifiability can be realized when the underlying GS scheme provides full traceability, we introduce a formal security definition of public verifiability by following BUFF (Beyond UnForgeability Features) security [3, 14, 17]. Informally, BUFF security guarantees that no adversary can output a forged verification key that indicates a signature is valid under two distinct verification keys. Briefly, our definition of public verifiability guarantees that no adversary can output a forged public key that defines a distinct message space from the original one. We show that publicly verifiable PKE can be constructed from fully traceable GS.

2 Preliminaries: PKE

Let $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ be a PKE scheme.

Definition 1 (Syntax of PKE)**.**

$\mathsf{PKE.KeyGen}$**:** *The key generation algorithm takes as a security parameter λ as input, and outputs a public key and decryption key pair $(\mathsf{pk}, \mathsf{dk})$. pk contains a plaintext space which we denote $\mathsf{MessageSpace}(\mathsf{pk})$.*

$\mathsf{PKE.Enc}$**:** *The encryption algorithm takes pk and a plaintext $m \in \mathsf{MessageSpace}(\mathsf{pk})$, and outputs a ciphertext C.*

$\mathsf{PKE.Dec}$**:** *The decryption algorithm takes dk and C as input, and outputs m or $\perp$.*

Correctness: We say that a PKE scheme is correct if for all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}, \mathsf{dk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$, all $m \in \mathsf{MessageSpace}(\mathsf{pk})$, $\mathsf{PKE.Dec}(\mathsf{dk}, \mathsf{PKE.Enc}(\mathsf{pk}, m)) = m$ holds with overwhelming probability.

The IND-CCA security is defined as follows.

Definition 2 (IND-CCA)**.** *Let $\mathcal{C}$ be the challenger and $\mathcal{A}$ be an adversary. We define the security model via the security game between $\mathcal{C}$ and $\mathcal{A}$. At the beginning of the game, $\mathcal{C}$ runs $(\mathsf{pk}, \mathsf{dk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$, and gives pk to $\mathcal{A}$. $\mathcal{A}$ is allowed to issue decryption queries.*

Decryption Query: *$\mathcal{A}$ sends a ciphertext C. $\mathcal{C}$ returns the result of $\mathsf{PKE.Dec}(\mathsf{dk}, C)$.*

$\mathcal{A}$ declares two equal-length plaintexts $m_0^*, m_1^* \in \mathsf{MessageSpace}(\mathsf{pk})$. $\mathcal{C}$ flips $b \xleftarrow{\$} \{0,1\}$, computes the challenge ciphertext $C^* \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, m_b^*)$, and sends C^* to $\mathcal{A}$.

$\mathcal{A}$ is further allowed to issue decryption queries.

Decryption Query: *$\mathcal{A}$ sends a ciphertext $C \neq C^*$. $\mathcal{C}$ returns the result of $\mathsf{PKE.Dec}(\mathsf{dk}, C)$.*

Finally, $\mathcal{A}$ outputs $b' \in \{0,1\}$. We say that a PKE scheme is IND-CCA secure if $|\Pr[b = b'] - 1/2|$ is negligible in λ for any PPT adversaries $\mathcal{A}$.

3 GS and the BMW Model

In this section, we introduce the BMW model [5]. Let $\mathsf{GS} = (\mathsf{Setup}, \mathsf{GSign}, \mathsf{GVerify}, \mathsf{Open})$ be a GS scheme defined as follows.

Definition 3 (Syntax of GS)**.**

Setup**:** *The setup algorithm takes a security parameter λ and the number of group users n, and outputs a group public key gpk, an opening key ok, and signing keys $\{\mathsf{gsigk}_i\}_{i\in[n]}$.*

GSign**:** *The signing algorithm takes gsigk_i and a message M, and outputs a group signature Σ.*

$\mathsf{GVerify}$**:** *The verification algorithm takes gpk, Σ, and M as input, and outputs 0 or 1.*

Open**:** *The opening algorithm takes ok, Σ, and M, and outputs an index i or a failure symbol $\perp$.*

Correctness: For all $\lambda, n \in \mathbb{N}$, all $(\mathsf{gpk}, \mathsf{ok}, \{\mathsf{gsigk}_i\}_{i\in[n]}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$, all $i \in [n]$, and all $M \in \{0,1\}^*$, $\mathsf{GVerify}(\mathsf{gpk}, \mathsf{GSign}(\mathsf{gsigk}_i, M), M) = 1$ and $\mathsf{Open}(\mathsf{ok}, \mathsf{GSign}(\mathsf{gsigk}_i, M), M) = i$ hold with overwhelming probability.

Next, we define full anonymity. It guarantees that an adversary $\mathcal{A}$, who does not possess ok, cannot distinguish which signing key is used for generating a group signature, even $\mathcal{A}$ has all signing keys.

Definition 4 (Full Anonymity)**.** *Let $\mathcal{C}$ be the challenger and $\mathcal{A}$ be an adversary. We define the security model via the security game between $\mathcal{C}$ and $\mathcal{A}$. At the beginning of the game, $\mathcal{C}$ runs* $(\mathsf{gpk}, \mathsf{ok}, \{\mathsf{gsigk}_i\}_{i\in[n]}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$ *and gives* $(\mathsf{gpk}, \{\mathsf{gsigk}_i\}_{i\in[n]})$ *to $\mathcal{A}$. $\mathcal{A}$ is allowed to issue opening queries.*

Opening Query: *$\mathcal{A}$ sends (Σ, M). $\mathcal{C}$ returns the result of* $\mathsf{Open}(\mathsf{ok}, \Sigma, M)$.

$\mathcal{A}$ declares (i_0^*, i_1^*, M^*) where $i_0^*, i_1^* \in [n]$. $\mathcal{C}$ flips $b \xleftarrow{\$} \{0,1\}$, computes $\Sigma^* \leftarrow \mathsf{GSign}(\mathsf{gsigk}_{i_b^*}, M^*)$, *and gives Σ^* to $\mathcal{A}$. $\mathcal{A}$ is further allowed to issue opening queries.*

Opening Query: *$\mathcal{A}$ sends $(\Sigma, M) \neq (\Sigma^*, M^*)$. $\mathcal{C}$ returns the result of* $\mathsf{Open}(\mathsf{ok}, \Sigma, M)$.

Finally, $\mathcal{A}$ outputs $b' \in \{0,1\}$. We say that the GS scheme is fully anonymous if the advantage

$$\mathsf{Adv}^{\mathsf{full\text{-}anon}}_{\mathsf{GS},\mathcal{A}}(\lambda, n) = |\Pr[b = b'] - 1/2|$$

is negligible in λ for any PPT adversaries $\mathcal{A}$.

The above definition is also called full CCA anonymity since $\mathcal{A}$ is allowed to issue opening queries. We can consider a weaker version called full CPA anonymity where an adversary is not allowed to issue an opening query.

Next, we define full traceability. It guarantees that an adversary $\mathcal{A}$ cannot produce a valid group signature whose opening result is either $\perp$ or an uncorrupted user. In the original definition, the second winning condition below (i.e., the opening result is $i \notin [n]$) is implicitly contained in other conditions. However, it seems ambiguous which condition contains the second condition. For example, we can define that the Open algorithm outputs $\perp$ if the opening result is $i \notin [n]$, or we can also regard that a user $i \notin [n]$ is not corrupted since $\mathcal{A}$ is allowed to issue a corrupt query for $i \in [n]$. In order to avoid any confusion, we separately define the condition below. We also remark that we can define full traceability by following strong unforgeability. Nevertheless, we follow the original BMW model where (i, M^*) is not sent as a signing query.

Definition 5 (Full Traceability)**.** *Let $\mathcal{C}$ be the challenger and $\mathcal{A}$ be an adversary. We define the security model via the security game between $\mathcal{C}$ and $\mathcal{A}$. At the beginning of the game, $\mathcal{C}$ runs* $(\mathsf{gpk}, \mathsf{ok}, \{\mathsf{gsigk}_i\}_{i\in[n]}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$ *and gives* $(\mathsf{gpk}, \mathsf{ok})$ *to $\mathcal{A}$. $\mathcal{C}$ initialize* $\mathsf{CorrU} = \emptyset$. *$\mathcal{A}$ is allowed to issue corruption and signing queries.*

Corrupt User Query: *$\mathcal{A}$ sends $i \in [n]$. Then $\mathcal{C}$ updates* $\mathsf{CorrU} = \mathsf{CorrU} \cup \{i\}$ *and returns* gsigk_i *to $\mathcal{A}$.*

Signing Query: *$\mathcal{A}$ sends $i \in [n]$ and M. Then, $\mathcal{C}$ runs* $\Sigma \leftarrow \mathsf{GSign}(\mathsf{gsigk}_i, M)$ *and returns Σ to $\mathcal{A}$.*

Finally, $\mathcal{A}$ outputs (Σ^*, M^*). $\mathcal{C}$ outputs 0 if $\mathsf{GVerify}(\mathsf{gpk}, \Sigma^*, M^*) = 0$. *Otherwise, $\mathcal{C}$ runs* $i \leftarrow \mathsf{Open}(\mathsf{ok}, \Sigma^*, M^*)$. *$\mathcal{C}$ outputs 1 if one of the followings hold.*

1. $i = \perp$.
2. $i \neq \perp \wedge i \notin [n]$.
3. $i \in [n]$, $i \notin \mathsf{CorrU}$, *and (i, M^*) is not sent as a signing query.*

We say that the GS scheme is fully traceable if the advantage

$$\mathsf{Adv}^{\mathsf{full\text{-}trace}}_{\mathsf{GS},\mathcal{A}}(\lambda, n) = \Pr[\mathcal{C} \to 1]|$$

is negligible in λ for any PPT adversaries $\mathcal{A}$.

4 Essential Condition of Full Traceability

In this section, we show that the last winning condition of full traceability is essential. To show this, we construct a GS scheme secure in the BMW model except the last winning condition of full traceability from PKE only.

We construct a GS scheme from $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ as follows. As mentioned in the introduction, the construction is quite artificial and meaningless in practice because the $\mathsf{GVerify}$ algorithm always outputs 1 regardless of the input. In other words, GS is meaningless in practice if we do not consider the last winning condition of full traceability. This result suggests us the last winning condition is essential in full traceability.

$\mathsf{Setup}(1^\lambda, 1^n)$**:** Run $(\mathsf{pk}, \mathsf{dk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and choose $i' \xleftarrow{\$} [n]$. Output $\mathsf{gpk} = \mathsf{pk}$, $\mathsf{ok} = (\mathsf{dk}, i')$, and $\mathsf{gsigk}_i = i$ for $i \in [n]$.

$\mathsf{GSign}(\mathsf{gsigk}_i, M)$**:** Parse $\mathsf{gsigk}_i = i$. Compute $C \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, i)$ and output $\Sigma = C$.

$\mathsf{GVerify}(\mathsf{gpk}, \Sigma, M)$**:** Output 1.

$\mathsf{Open}(\mathsf{ok}, \Sigma, M)$**:** Parse $\mathsf{ok} = (\mathsf{dk}, i')$ and $\Sigma = C$. Run $i \leftarrow \mathsf{PKE.Dec}(\mathsf{dk}, C)$. If $i \in [n]$, output i. Otherwise, if $i = \perp$ or $i \notin [n]$, then output i'.

Theorem 1. *The proposed construction is correct if the underlying PKE scheme is correct.*

Proof. Since the $\mathsf{GVerify}$ algorithm always outputs 1, $\mathsf{GVerify}(\mathsf{gpk}, \mathsf{GSign}(\mathsf{gsigk}_i, M), M) = 1$ holds for all $i \in [n]$ and $M \in \{0,1\}^*$. We need to show that $\mathsf{Open}(\mathsf{ok}, \mathsf{GSign}(\mathsf{gsigk}_i, M), M) = i$ holds for all $i \in [n]$ and $M \in \{0,1\}^*$. Now, an honestly generated group signature is represented as $\Sigma = C$ where $C \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, i)$. Since the underlying PKE scheme is assumed to be correct, for $i' \leftarrow \mathsf{PKE.Dec}(\mathsf{dk}, C)$, $i' = i$ holds with overwhelming probability. Thus, the Open algorithm correctly outputs i with overwhelming probability. This concludes the proof. □

Theorem 2. *The proposed construction is fully CPA anonymous if the underlying PKE scheme is IND-CPA secure.*[4]

Proof. Let $\mathcal{C}$ be the challenger of the IND-CPA security and $\mathcal{A}$ be an adversary of full anonymity. We construct an algorithm $\mathcal{B}$ that breaks the IND-CPA security as follows. At the beginning of the game, $\mathcal{C}$ runs $(\mathsf{pk}, \mathsf{dk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and gives pk to $\mathcal{B}$. $\mathcal{B}$ sets $\mathsf{gpk} = \mathsf{pk}$, and sends $(\mathsf{gpk}, \{i\}_{i\in[n]})$ to $\mathcal{A}$. $\mathcal{A}$ declares (i_0^*, i_1^*, M^*) where $i_0^*, i_1^* \in [n]$. $\mathcal{B}$ sets $(m_0^*, m_1^*) = (i_0^*, i_1^*)$ and sends (m_0^*, m_1^*) to $\mathcal{C}$. $\mathcal{C}$ returns the challenge ciphertext $C^* \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, m_b^*)$ where $b \in \{0,1\}$. $\mathcal{B}$ sets $\Sigma^* = C^*$ and gives Σ^* to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs $b' \in \{0,1\}$. $\mathcal{B}$ outputs the same b' and breaks full anonymity at least the advantage of $\mathcal{A}$. □

Theorem 3. *The proposed construction is fully traceable without the last condition.*

[4]Remark that the construction is not fully CCA anonymous even if the underlying PKE scheme is IND-CCA secure. Briefly, an adversary is allowed to issue an opening query (Σ^*, M) where $\Sigma^* = C^*$ is the challenge ciphertext of the PKE scheme and $M \neq M^*$. Since the $\mathsf{GVerify}$ algorithm always outputs 1, the reduction algorithm needs to respond the query but the reduction algorithm does not send C^* as a decryption query.

Proof. Since the Open algorithm never outputs $\perp$ or $i \notin [n]$, the advantage $\mathsf{Adv}^{\mathsf{full\text{-}trace}}_{\mathsf{GS},\mathcal{A}}(\lambda, n) = 0$. □

Remark 1. The proof of Theorem 3 relies on the fact that no group signature exists in the construction for which the opening result is $\perp$. To exclude the case, we can add the condition "there exists a group signature for which the opening result is $\perp$". However, this is not sufficient to bypass the artificial construction. For example, $\mathsf{GVerify}(\mathsf{gpk}, \Sigma, M)$ outputs 0 only when $(\Sigma, M) = (\perp, \perp)$, and $\mathsf{Open}(\mathsf{ok}, \Sigma, M)$ outputs $\perp$ only when $(\Sigma, M) = (\perp, \perp)$. Since $\mathcal{A}$ is required to produce a valid group signature in the definition of full traceability, $\mathcal{A}$ cannot output $(\Sigma^*, M^*) = (\perp, \perp)$ and thus $\mathsf{Adv}^{\mathsf{full\text{-}trace}}_{\mathsf{GS},\mathcal{A}}(\lambda, n) = 0$ holds.

5 Publicly Verifiable PKE, Revisited

In this section, we explore the public verifiability of PKE constructed by GS via the Abdalla-Warinschi methodology [1] where signing keys of GS are set as a public key of PKE.

5.1 GS-based PKE

Frist, we construct a (1-bit) PKE scheme[5] from $\mathsf{GS} = (\mathsf{Setup}, \mathsf{GSign}, \mathsf{GVerify}, \mathsf{Open})$ as follows.

$\mathsf{PKE.KeyGen}(1^\lambda)$**:** Run $(\mathsf{gpk}, \mathsf{ok}, \mathsf{gsigk}_1, \mathsf{gsigk}_2) \leftarrow \mathsf{Setup}(1^\lambda, 1^2)$ and output $\mathsf{pk} = (\mathsf{gpk}, \mathsf{gsigk}_1, \mathsf{gsigk}_2)$ and $\mathsf{dk} = \mathsf{ok}$.

$\mathsf{PKE.Enc}(\mathsf{pk}, m)$**:** Parse $\mathsf{pk} = (\mathsf{gpk}, \mathsf{gsigk}_1, \mathsf{gsigk}_2)$. For $m \in \mathsf{MessageSpace}(\mathsf{pk}) = \{0,1\}$, choose $M \in \{0,1\}^*$ and run $\Sigma \leftarrow \mathsf{GSign}(\mathsf{gsigk}_{m+1}, M)$. Output $C = (\Sigma, M)$.

$\mathsf{PKE.Dec}(\mathsf{dk}, C)$**:** Parse $\mathsf{dk} = \mathsf{ok}$ and $C = (\Sigma, M)$. Run $i \leftarrow \mathsf{Open}(\mathsf{ok}, \Sigma, M)$. If $i = \perp$, then output $\perp$. Otherwise, output $m = i - 1$.

The following theorems are straightforward, but we explicitly give them to clarify that full traceability is not necessary to provide the correctness and the IND-CCA security.

Theorem 4. *The PKE scheme is correct if the underlying GS scheme is correct.*

Proof. For an honestly generated group signature $\Sigma \leftarrow \mathsf{GSign}(\mathsf{gsigk}_{m+1}, M)$ and $i \leftarrow \mathsf{Open}(\mathsf{ok}, \Sigma, M)$, $i = m + 1$ holds due to the correctness of GS. □

Theorem 5. *The PKE scheme is IND-CCA secure if the underlying GS scheme is fully CCA anonymous.*

Proof. Let $\mathcal{C}$ be the challenger of full CCA anonymity and $\mathcal{A}$ be an adversary of the IND-CCA security. We construct an algorithm $\mathcal{B}$ that breaks full CCA anonymity as follows. At the beginning of the game, $\mathcal{C}$ runs $(\mathsf{gpk}, \mathsf{ok}, \mathsf{gsigk}_1, \mathsf{gsigk}_2) \leftarrow \mathsf{Setup}(1^\lambda, 1^2)$ and sends $(\mathsf{gpk}, \mathsf{gsigk}_1, \mathsf{gsigk}_2)$ to $\mathcal{B}$. $\mathcal{B}$ sets $\mathsf{pk} = (\mathsf{gpk}, \mathsf{gsigk}_1, \mathsf{gsigk}_2)$ and sends pk to $\mathcal{A}$. For a decryption query $C = (\Sigma, M)$ issued by $\mathcal{A}$, $\mathcal{B}$ forwards (Σ, M) to $\mathcal{C}$ as an opening query. If $\mathcal{C}$ returns $\perp$, then $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. Otherwise, when $\mathcal{C}$ returns $i \in \{1, 2\}$, then $\mathcal{B}$ returns $i - 1$ to $\mathcal{A}$. $\mathcal{A}$ declares (m_0^*, m_1^*). Without loss of generality, we

[5]To amplify the message space, we can use M (a signed message) as a tag/label where for a key pair of a one-time signature scheme $(\mathsf{verk}, \mathsf{sigk})$, compute $\Sigma_i \leftarrow \mathsf{GSign}(\mathsf{gsigk}_{m_i+1}, \mathsf{verk})$ for encrypting the i-th bit of the plaintext m_i, and a ciphertext is $(\sigma_{\mathsf{OTS}}, \{\Sigma_i\})$. This message space amplification technique was introduced in [19].

set $(m_0^*, m_1^*) = (0, 1)$. $\mathcal{B}$ randomly chooses $M^* \in \{0,1\}^*$ and sends $(1, 2, M^*)$ to $\mathcal{C}$ as the challenge query. $\mathcal{C}$ returns $\Sigma^* \leftarrow \mathsf{GSign}(\mathsf{gpk}, \mathsf{gsigk}_{b+1}, M^*)$ to $\mathcal{B}$ where $b \in \{0,1\}$. $\mathcal{B}$ sets $C^* = (\Sigma^*, M^*)$ and sends C^* to $\mathcal{A}$. For a decryption query $C = (\Sigma, M)$ issued by $\mathcal{A}$, $\mathcal{B}$ responds the query as in the previous phase. We note that $\mathcal{B}$ can forward any query since $C \neq C^*$, $(\Sigma, M) \neq (\Sigma^*, M^*)$ holds. Finally, $\mathcal{A}$ outputs $b' \in \{0,1\}$. $\mathcal{B}$ outputs the same b' and breaks full anonymity at least the same advantage of $\mathcal{A}$. □

The following theorem also holds (we omit the proof because it can be obtained by the proof of Theorem 5).

Theorem 6. *The PKE scheme is IND-CPA secure if the underlying GS scheme is fully CPA anonymous.*

5.2 Definition of Public Verifiability

Since $C = (\Sigma, M)$ in the GS-based PKE scheme given in Section 5.1, the validity of C can be publicly checked via $\mathsf{GVerify}(\mathsf{gpk}, \Sigma, M)$. Here, the Open algorithm is internally run in the $\mathsf{PKE.Dec}$ algorithm. Thus, intuitively the verifiability guarantees that if $\mathsf{GVerify}(\mathsf{gpk}, \Sigma, M) = 1$ for $C = (\Sigma, M)$, then $m \in \mathsf{MessageSpace}(\mathsf{pk})$ holds for $m \leftarrow \mathsf{PKE.Dec}(\mathsf{dk}, C)$. However, to reduce full traceability, the reduction algorithm issues corruption queries 1 and 2, to obtain $(\mathsf{gsigk}_1, \mathsf{gsigk}_2)$. Under the full corruption, the last condition of full traeability does not happen. That is, to construct an IND-CPA secure GS-based PKE scheme, it is sufficient that the underlying GS scheme provides the correctness, full CPA anonymity, and full traceability without the last condition that is exactly artificially constructed in Section 4. Since such a GS scheme can be constructed from PKE only and is meaningless in practice, the above intuition does not capture what we can achieve when a publicly verifiable PKE scheme is constructed from GS secure in the BMW model. Thus, in this section, we explore what the verifiability guarantees in the GS-based PKE scheme.

First, we define the verification algorithm of PKE, $\mathsf{PKE.Verify}$, as follows. We introduce a verification key vk here.

Definition 6 (Syntax of Verifiable PKE)**.**

$\mathsf{PKE.KeyGen}$**:** *The key generation algorithm takes as a security parameter λ as input, and outputs a public key, a verification key, and a decryption key* $(\mathsf{pk}, \mathsf{vk}, \mathsf{dk})$*.* pk *contains a plaintext space which we denote* $\mathsf{MessageSpace}(\mathsf{pk})$*.*

$\mathsf{PKE.Enc}$**:** *The encryption algorithm takes* pk *and a plaintext $m \in \mathsf{MessageSpace}(\mathsf{pk})$, and outputs a ciphertext C.*

$\mathsf{PKE.Dec}$**:** *The decryption algorithm takes* dk *and C as input, and outputs m or $\perp$.*

$\mathsf{PKE.Verify}$**:** *The verification algorithm takes* vk *and C as input, and output 0 or 1.*

We implicitly assume that dk contains vk, and the $\mathsf{PKE.Dec}$ algorithm internally runs the $\mathsf{PKE.Verify}$ algorithm, and outputs $\perp$ if $\mathsf{PKE.Verify}(\mathsf{vk}, C) = 0$.

Correctness: For all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}, \mathsf{vk}, \mathsf{dk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$, and all $m \in \mathsf{MessageSpace}(\mathsf{pk})$, $\mathsf{PKE.Dec}(\mathsf{dk}, \mathsf{PKE.Enc}(\mathsf{pk}, m)) = m$ and $\mathsf{PKE.Verify}(\mathsf{vk}, C) = 1$ hold with overwhelming probability.

Definition 7 (IND-CCA Security)**.** *Let $\mathcal{C}$ be the challenger and $\mathcal{A}$ be an adversary. We define the security model via the security game between $\mathcal{C}$ and $\mathcal{A}$. At the beginning of the game, $\mathcal{C}$ runs* $(\mathsf{pk}, \mathsf{vk}, \mathsf{dk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ *and gives* $(\mathsf{pk}, \mathsf{vk})$ *to $\mathcal{A}$. $\mathcal{A}$ is allowed to issue decryption queries.*

Decryption Query: *$\mathcal{A}$ sends C. $\mathcal{C}$ returns the result of* $\mathsf{PKE.Dec}(\mathsf{dk}, C)$.

$\mathcal{A}$ declares two equal-length plaintexts $m_0^*, m_1^* \in \mathsf{MessageSpace}(\mathsf{pk})$. $\mathcal{C}$ chooses $b \xleftarrow{\$} \{0,1\}$, computes $C^* \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, m_b^*)$*, gives C^* to $\mathcal{A}$. $\mathcal{A}$ is further allowed to issue decryption queries.*

Decryption Query: *$\mathcal{A}$ sends $C \neq C^*$. $\mathcal{C}$ returns the result of* $\mathsf{PKE.Dec}(\mathsf{dk}, C)$.

Finally, $\mathcal{A}$ outputs $b' \in \{0,1\}$. We say that the PKE scheme is IND-CCA secure if the advantage

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = |\Pr[b = b'] - 1/2|$$

is negligible in λ for any PPT adversaries $\mathcal{A}$.

Next, we define public verifiability. Basically, it guarantees that the decryption result of a valid ciphertext is in $\mathsf{MessageSpace}(\mathsf{pk})$. However, as mentioned above, a straightforward definition does not capture what we can achieve when a publicly verifiable PKE scheme is constructed from GS secure in the BMW model. To capture this, we introduce a BUFF-like security model. We require that an adversary of the public verifiability, who is given $(\mathsf{pk}, \mathsf{vk}, \mathsf{dk})$, outputs a ciphertext C^* and a public key pk^* such that C^* is valid under vk, the decryption result is in $\mathsf{MessageSpace}(\mathsf{pk}^*)$, and $\mathsf{MessageSpace}(\mathsf{pk}) \cap \mathsf{MessageSpace}(\mathsf{pk}^*) = \emptyset$. Our definition guarantees that the decryption result of a valid ciphertext under vk is in $\mathsf{MessageSpace}(\mathsf{pk})$. Moreover, producing such a pk^* violates the last winning condition of full traceability since pk^* contains a signing key of an uncorrupted user in the GS context. This, we can reduce the public verifiability to full traceability of the underlying GS scheme, and it seems that our definition adequately captures what security can be achieved when a publicly verifiable PKE scheme is constructed from GS secure in the BMW model.

Definition 8 (Public Verifiability)**.** *Let $\mathcal{C}$ be the challenger and $\mathcal{A}$ be an adversary. We define the security model via the security game between $\mathcal{C}$ and $\mathcal{A}$. At the beginning of the game, $\mathcal{C}$ runs* $(\mathsf{pk}, \mathsf{vk}, \mathsf{dk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ *and gives* $(\mathsf{pk}, \mathsf{vk}, \mathsf{dk})$ *to $\mathcal{A}$. $\mathcal{A}$ outputs (C^*, pk^*). Let* $m^* \leftarrow \mathsf{PKE.Dec}(\mathsf{dk}, C^*)$*. We say that $\mathcal{A}$ wins if* $\mathsf{PKE.Verify}(\mathsf{vk}, C^*) = 1$*, $m^* \in \mathsf{MessageSpace}(\mathsf{pk}^*)$, and* $\mathsf{MessageSpace}(\mathsf{pk}) \cap \mathsf{MessageSpace}(\mathsf{pk}^*) = \emptyset$ *hold. We say that the PKE scheme is publicly verifiable if the advantage*

$$\mathsf{Adv}^{\mathsf{pub\text{-}ver}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = \Pr[\mathcal{A} \textit{ wins}]$$

is negligible in λ for any PPT adversaries $\mathcal{A}$.

Remark 2. Nieto et al. [21] defined strictly non-trivial public verification (we give their definition in Section 6 for the sake of clarity). They introduced a verification algorithm and it is required that the verification algorithm outputs 0 for all ciphertexts if and only if the decryption result of the ciphertexts are $\perp$ (Condition 1). Moreover, it is also required that there exists a ciphertext for which the decryption result is $\perp$ (Condition 2). Especially, it seems that Condition 2 is introduced to exclude a trivial construction where the verification algorithm never outputs $\perp$. However, still we can consider an artificial construction. See Section 6 for details.

5.3 Publicly Verifiable GS-based PKE

We explicitly construct a publicly verifiable PKE scheme from GS secure in the BMW model as follows. Here, we set $n > 2$ that allows an adversary to produce a valid group signature whose opening result is an uncorrupted user.

$\mathsf{PKE.KeyGen}(1^\lambda)$**:** Run $(\mathsf{gpk}, \mathsf{ok}, \{\mathsf{gsigk}_i\}_{i\in[n]}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$ for $n > 2$. Pick $\mathsf{gsigk}, \mathsf{gsigk}' \xleftarrow{\$} \{\mathsf{gsigk}_i\}_{i\in[n]}$ and set $(\mathsf{gsigk}_1, \mathsf{gsigk}_2) = (\mathsf{gsigk}, \mathsf{gsigk}')$. Output $\mathsf{pk} = (\mathsf{gsigk}_1, \mathsf{gsigk}_2)$, $\mathsf{vk} = \mathsf{gpk}$, and $\mathsf{dk} = (\mathsf{vk}, \mathsf{ok})$.

$\mathsf{PKE.Enc}(\mathsf{pk}, m)$**:** Parse $\mathsf{pk} = (\mathsf{gsigk}_1, \mathsf{gsigk}_2)$. For $m \in \mathsf{MessageSpace}(\mathsf{pk}) = \{0,1\}$, choose $M \in \{0,1\}^*$ and run $\Sigma \leftarrow \mathsf{GSign}(\mathsf{gsigk}_{m+1}, M)$. Output $C = (\Sigma, M)$.

$\mathsf{PKE.Dec}(\mathsf{dk}, C)$**:** Parse $\mathsf{dk} = (\mathsf{vk}, \mathsf{ok})$ and $C = (\Sigma, M)$. Output $\perp$ if $\mathsf{PKE.Verify}(\mathsf{vk}, C) = 0$. Otherwise, run $i \leftarrow \mathsf{Open}(\mathsf{ok}, \Sigma, M)$. If $i = \perp$, then output $\perp$. Otherwise, output $m = i - 1$.

$\mathsf{PKE.Verify}(\mathsf{vk}, C)$**:** Parse $\mathsf{vk} = \mathsf{gpk}$ and $C = (\Sigma, M)$. Output the result of $\mathsf{GVerify}(\mathsf{gpk}, \Sigma, M)$.

Theorem 7. *The PKE scheme is correct if the underlying GS scheme is correct.*

Proof. The same as the proof of Theorem 4.

Theorem 8. *The PKE scheme is IND-CCA secure if the underlying GS scheme is fully CCA anonymous.*

Basically, the proof is the same as the proof of Theorem 5. However, we give the proof to confirm the impact for introducing vk and the $\mathsf{PKE.Verify}$ algorithm.

Proof. Let $\mathcal{C}$ be the challenger of full CCA anonymity and $\mathcal{A}$ be an adversary of the IND-CCA security. We construct an algorithm $\mathcal{B}$ that breaks full CCA anonymity as follows. At the beginning of the game, $\mathcal{C}$ runs $(\mathsf{gpk}, \mathsf{ok}, \{\mathsf{gsigk}_i\}_{i\in[n]}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$ and sends $(\mathsf{gpk}, \{\mathsf{gsigk}_i\}_{i\in[n]})$ to $\mathcal{B}$. $\mathcal{B}$ picks $\mathsf{gsigk}, \mathsf{gsigk}' \xleftarrow{\$} \{\mathsf{gsigk}_i\}_{i\in[n]}$, sets $(\mathsf{gsigk}_1, \mathsf{gsigk}_2) = (\mathsf{gsigk}, \mathsf{gsigk}')$, and sends $(\mathsf{pk}, \mathsf{vk})$ to $\mathcal{A}$. For a decryption query $C = (\Sigma, M)$ issued by $\mathcal{A}$, $\mathcal{B}$ returns $\perp$ if $\mathsf{PKE.Verify}(\mathsf{vk}, C) = 0$. Otherwise, $\mathcal{B}$ forwards (Σ, M) to $\mathcal{C}$ as an opening query. If $\mathcal{C}$ returns $\perp$, then $\mathcal{B}$ returns $\perp$ to $\mathcal{A}$. Otherwise, when $\mathcal{C}$ returns $i \in \{1,2\}$, then $\mathcal{B}$ returns $i-1$ to $\mathcal{A}$. $\mathcal{A}$ declares (m_0^*, m_1^*). Without loss of generality, we set $(m_0^*, m_1^*) = (0,1)$. $\mathcal{B}$ randomly chooses $M^* \in \{0,1\}^*$ and sends $(1, 2, M^*)$ to $\mathcal{C}$ as the challenge query. $\mathcal{C}$ returns $\Sigma^* \leftarrow \mathsf{GSign}(\mathsf{gpk}, \mathsf{gsigk}_{b+1}, M^*)$ to $\mathcal{B}$ where $b \in \{0,1\}$. $\mathcal{B}$ sets $C^* = (\Sigma^*, M^*)$ and sends C^* to $\mathcal{A}$. For a decryption query $C = (\Sigma, M)$ issued by $\mathcal{A}$, $\mathcal{B}$ responds the query as in the previous phase. We remark that $\mathcal{B}$ can forward any query since $C \neq C^*$, $(\Sigma, M) \neq (\Sigma^*, M^*)$ holds. Finally, $\mathcal{A}$ outputs $b' \in \{0,1\}$. $\mathcal{B}$ outputs the same b' and breaks full anonymity at least the same advantage of $\mathcal{A}$. □

Theorem 9. *The PKE scheme is publicly verifiable if the underlying GS scheme is fully traceable.*

Proof. Let $\mathcal{C}$ be the challenger of full traceability and $\mathcal{A}$ be an adversary of the public verifiability. We construct an algorithm $\mathcal{B}$ that breaks full traceability as follows. At the beginning of the game, $\mathcal{C}$ runs $(\mathsf{gpk}, \mathsf{ok}, \{\mathsf{gsigk}_i\}_{i\in[n]}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$ and sends $(\mathsf{gpk}, \mathsf{ok})$ to $\mathcal{B}$. $\mathcal{B}$ randomly chooses distinct $i, j \xleftarrow{\$} [n]$ and sends i and j as corrupt queries. $\mathcal{C}$ returns two signing keys and $\mathcal{B}$ sets them as

gsigk_1 and gsigk_2. $\mathcal{B}$ sets $\mathsf{pk} = (\mathsf{gsigk}_1, \mathsf{gsigk}_2)$ and $\mathsf{vk} = \mathsf{gpk}$, and sends $(\mathsf{pk}, \mathsf{vk}, \mathsf{ok})$ to $\mathcal{A}$. Here, $\mathsf{MessageSpace}(\mathsf{pk}) = \{0,1\}$.

$\mathcal{A}$ outputs (C^*, pk^*) where $\mathsf{PKE.Verify}(\mathsf{vk}, C^*) = 1$ (i.e., $\mathsf{GVerify}(\mathsf{gpk}, \Sigma^*, M^*) = 1$ where $C^* = (\Sigma^*, M^*)$). Due to the winning condition of $\mathcal{A}$, for $m^* \leftarrow \mathsf{PKE.Dec}(\mathsf{dk}, \mathcal{C}^*)$, $m^* \in \mathsf{MessageSpace}(\mathsf{pk}^*)$ and $\mathsf{MessageSpace}(\mathsf{pk}) \cap \mathsf{MessageSpace}(\mathsf{pk}^*) = \emptyset$ hold. Let $\mathsf{pk}^* = \mathsf{gsigk}$ (though pk^* may contain two or more signing keys, one key is sufficient to reduce full traceability). Since gsigk is not a response of a corrupt query, the user whose signing key gsigk can be regarded as an uncorrupted user. $\mathcal{B}$ outputs (Σ^*, M^*) and breaks full traceability. □

Implication Results. The generic construction of GS secure in the BMW model [5] provides full anonymity if the underlying PKE scheme is IND-CCA secure and the non-interactive proof system is simulation sound, and provides full traceability if the underlying signature scheme is EUF-CMA secure and the non-interactive proof system is sound. By combining the generic construction and our result, we can construct a publicly verifiable PKE scheme from these cryptographic primitives. In terms of complexity assumptions, we can instantiate a public verifiable PKE scheme from GSs constructed from DL-based assumptions (pairings [16, 23] or DDH (w/o pairing) [11]), lattices (LWE/SIS [28]), isogenies (CSIDH [8]), and codes (McEliece [20]). Though a hash-based group signature scheme has been proposed in [18], it does not provide full anonymity (it provides selfless anonymity) and thus we cannot employ it as a building block.

6 Discussion

Nieto et al. [21] defined strictly non-trivial public verification for general encryption that contains PKE, identity-based encryption, and tag-based encryption. Because we focus on PKE in this paper, we remove IDSp and TagSp (in their notions) below. They introduced a parameter generation algorithm PG that takes a security parameter λ and outputs public parameters par, and the key generation algorithm $\mathsf{PKE.KeyGen}$ takes par as input. We follow the notion below. We also remark that the syntax of their verification algorithm is differ from ours where the Ver algorithm outputs either $\perp$ if the ciphertext fails the validation or a (transformed) ciphertext. The decryption algorithm Dec' takes the transformed ciphertext (and the decryption key), and outputs a plaintext or $\perp$. The actual decryption algorithm (Dec in the following definition) is represented as the composition of the verification algorithm and the decryption algorithm, i.e., $\mathsf{Dec} = \mathsf{Dec}' \circ \mathsf{Ver}$.

Definition 9 (Strictly Non-Trivial Public Verification [21]). *Let* $\mathsf{par} \leftarrow \mathsf{PG}(1^\lambda)$. Ver *is said to be strictly non-trivial if, for all* $(\mathsf{pk}, \mathsf{dk}) \leftarrow \mathsf{PKE.KeyGen}(\mathsf{par})$,

Condition 1. $\mathsf{Ver}(\mathsf{par}, \mathsf{pk}, C) = 0 \iff \mathsf{Dec}(\mathsf{par}, \mathsf{pk}, \mathsf{dk}, C) = \perp$ *for all* C.

Condition 2. *There exists a ciphertext* C *for which* $\mathsf{Dec}(\mathsf{par}, \mathsf{pk}, \mathsf{dk}, C) = \perp$.

They mentioned that "*Condition 1 requires that successful public verification is both necessary and sufficient for the decryption algorithm not to fail.*". Moreover, they mentioned that "*Condition 2 formally excludes IND-CCA-secure schemes where Dec never outputs* $\perp$ *(e.g. [2, 32, 33] where modified (challenge) ciphertexts decrypt to random messages) to capture the intuition that in order to determine whether* C *carries some meaningful message one must have at least partial knowledge of the private key (which contradicts the goals of strictly non-trivial public verification).*". However, still we can consider an artificial construction. For any IND-CCA secure PKE scheme where there is a ciphertext for which the decryption result is $\perp$, the key generation algorithm chooses an invalid ciphertext, say C_{invalid}, and contains it to the public key. For a ciphertext C, the verification

algorithm Ver outputs 0 only when C_{invalid} is input, and outputs C otherwise. If the decryption algorithm Dec' takes 0, then it outputs $\perp$. Otherwise, now Dec' takes C (and the decryption key) as input. If the decryption algorithm of the underlying PKE scheme returns $\perp$, then Dec' outputs a random non-$\perp$ plaintext (it can be pre-defined when Dec' is a deterministic algorithm). Otherwise, if the decryption algorithm of the underlying PKE scheme does not return $\perp$, then Dec' outputs the decryption result. This artificial construction does not affect the correctness since honestly generated ciphertext is correctly decrypted due to the correctness of the underlying PKE scheme. Moreover, it provides strictly non-trivial public verification. Here, what we argue in this paper is that the definition of strictly non-trivial public verification *by itself* can be bypassed by an artificial construction, and we do not claim that the above artificial construction does not affect the IND-CCA security. As a final remark, Nieto et al. mentioned that "*all IND-CCA-secure PKE schemes trivially achieve public verifiability with respect to* $\mathsf{Ver}(\mathsf{par}, \mathsf{pk}, C) := C$ *and* $\mathsf{Dec}' := \mathsf{Dec}$*. We are often interested in the case where something non-trivial is occurring in* Ver*, i.e. where the consistency check is essential for successful decryption.*".[6] We totally agree with the concept and follow it to give our definition for public verifiability in this paper.

7 Conclusion

In this paper, we explore the winning conditions of full traceability. We show that the condition, where an adversary is required to produce a valid group signature whose opening result is an uncorrupted user, is essential. We also explore the public verifiability of a GS-based PKE scheme. Although we have shown implication results of publicly verifiable PKE from GS, the construction is not efficient. Basically, the construction involves NIZK as its building block. It would be interesting to construct an efficient publicly verifiable PKE scheme without NIZK, or to construct a generic transformation adding the public verifiability to non-verifiable PKE. We leave them as future works of this paper. Considering other implication results among GSs and other primitives besides public verifiability is also an interesting future work.

Acknowledgment: The content in Section 6 is based on the discussion with Prof. Toshihiro Ohigashi, Prof. Kazumasa Omote, and Mr. Tatsuya Suzuki. The authors would like to thank them for their invaluable comments and suggestions. This work was supported by JSPS KAKENHI Grant Number JP21K11897.

References

[1] Michel Abdalla and Bogdan Warinschi. On the minimal assumptions of group signature schemes. In Javier López, Sihan Qing, and Eiji Okamoto, editors, *ICICS 04*, volume 3269 of *LNCS*, pages 1–13. Springer, Berlin, Heidelberg, October 2004.

[2] Masayuki Abe, Eike Kiltz, and Tatsuaki Okamoto. Chosen ciphertext security with optimal ciphertext overhead. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 355–371. Springer, Berlin, Heidelberg, December 2008.

[3] Thomas Aulbach, Samed Düzlü, Michael Meyer, Patrick Struck, and Maximiliane Weishäupl. Hash your keys before signing - BUFF security of the additional NIST PQC signatures. In Markku-Juhani Saarinen and Daniel Smith-Tone, editors, *Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Part II*, pages 301–335. Springer, Cham, June 2024.

[6] Here, we replace "generic encryption" to "PKE" and remove a tag t from the input of Ver from their sentences.

[4] Mihir Bellare and Georg Fuchsbauer. Policy-based signatures. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 520–537. Springer, Berlin, Heidelberg, March 2014.

[5] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer, Berlin, Heidelberg, May 2003.

[6] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security proofs for identity-based identification and signature schemes. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 268–286. Springer, Berlin, Heidelberg, May 2004.

[7] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–153. Springer, Berlin, Heidelberg, February 2005.

[8] Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 95–126. Springer, Cham, May / June 2022.

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

[10] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, and Jens Groth. Foundations of fully dynamic group signatures. In Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider, editors, *ACNS 16International Conference on Applied Cryptography and Network Security*, volume 9696 of *LNCS*, pages 117–136. Springer, Cham, June 2016.

[11] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on DDH. In Günther Pernul, Peter Y. A. Ryan, and Edgar R. Weippl, editors, *ESORICS 2015, Part I*, volume 9326 of *LNCS*, pages 243–265. Springer, Cham, September 2015.

[12] Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In Carlo Blundo and Stelvio Cimato, editors, *SCN 04*, volume 3352 of *LNCS*, pages 120–133. Springer, Berlin, Heidelberg, September 2005.

[13] David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *EUROCRYPT'91*, volume 547 of *LNCS*, pages 257–265. Springer, Berlin, Heidelberg, April 1991.

[14] Cas Cremers, Samed Düzlü, Rune Fiedler, Marc Fischlin, and Christian Janson. BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures. In *2021 IEEE Symposium on Security and Privacy*, pages 1696–1714. IEEE Computer Society Press, May 2021.

[15] Ivan Damgård, Dennis Hofheinz, Eike Kiltz, and Rune Thorbek. Public-key encryption with non-interactive opening. In Tal Malkin, editor, *CT-RSA 2008*, volume 4964 of *LNCS*, pages 239–255. Springer, Berlin, Heidelberg, April 2008.

[16] David Derler and Daniel Slamanig. Highly-efficient fully-anonymous dynamic group signatures. In Jong Kim, Gail-Joon Ahn, Seungjoo Kim, Yongdae Kim, Javier López, and Taesoo Kim, editors, *ASIACCS 18*, pages 551–565. ACM Press, April 2018.

[17] Jelle Don, Serge Fehr, Yu-Hsuan Huang, Jyun-Jie Liao, and Patrick Struck. Hide-and-seek and the non-resignability of the BUFF transform. In Elette Boyle and Mohammad Mahmoody, editors, *Theory of Cryptography - 22nd International Conference, TCC 2024, Part III*, volume 15366 of *LNCS*, pages 347–370. Springer, 2024.

[18] Rachid El Bansarkhani and Rafael Misoczki. G-merkle: A hash-based group signature scheme from standard assumptions. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018*, pages 441–463. Springer, Cham, 2018.

[19] Keita Emura, Goichiro Hanaoka, Yusuke Sakai, and Jacob C. N. Schuldt. Group signature implies public-key encryption with non-interactive opening. *International Journal of Information Security*, 13(1):51–62, 2014.

[20] Martianus Frederic Ezerman, Hyung Tae Lee, San Ling, Khoa Nguyen, and Huaxiong Wang. Provably secure group signature schemes from code-based assumptions. *IEEE Transactions on Information Theory*, 66(9):5754–5773, 2020.

[21] Juan Manuel González Nieto, Mark Manulis, Bertram Poettering, Jothi Rangasamy, and Douglas Stebila. Publicly verifiable ciphertexts. In Ivan Visconti and Roberto De Prisco, editors, *SCN 12*, volume 7485 of *LNCS*, pages 393–410. Springer, Berlin, Heidelberg, September 2012.

[22] Ai Ishida, Yusuke Sakai, Keita Emura, Goichiro Hanaoka, and Keisuke Tanaka. Fully anonymous group signature with verifier-local revocation. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 23–42. Springer, Cham, September 2018.

[23] Ai Ishida, Yusuke Sakai, Keita Emura, Goichiro Hanaoka, and Keisuke Tanaka. Proper usage of the group signature scheme in ISO/IEC 20008-2. In Steven D. Galbraith, Giovanni Russello, Willy Susilo, Dieter Gollmann, Engin Kirda, and Zhenkai Liang, editors, *ASIACCS 19*, pages 515–528. ACM Press, July 2019.

[24] Shuichi Katsumata and Shota Yamada. Group signatures without NIZK: From lattices in the standard model. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 312–344. Springer, Cham, May 2019.

[25] Aggelos Kiayias and Moti Yung. Secure scalable group signature with dynamic joins and separable authorities. *International Journal of Security and Networks*, 1(1/2):24–45, 2006.

[26] Benoît Libert, Thomas Peters, and Moti Yung. Group signatures with almost-for-free revocation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 571–589. Springer, Berlin, Heidelberg, August 2012.

[27] Benoît Libert, Thomas Peters, and Moti Yung. Scalable group signatures with revocation. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 609–627. Springer, Berlin, Heidelberg, April 2012.

[28] San Ling, Khoa Nguyen, Huaxiong Wang, and Yanhong Xu. Lattice-based group signatures: Achieving full dynamicity with ease. In Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi,

editors, *ACNS 17International Conference on Applied Cryptography and Network Security*, volume 10355 of *LNCS*, pages 293–312. Springer, Cham, July 2017.

[29] Toru Nakanishi, Hiroki Fujii, Yuta Hira, and Nobuo Funabiki. Revocable group signature schemes with constant costs for signing and verifying. In Stanislaw Jarecki and Gene Tsudik, editors, *PKC 2009*, volume 5443 of *LNCS*, pages 463–480. Springer, Berlin, Heidelberg, March 2009.

[30] Toru Nakanishi and Nobuo Funabiki. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 533–548. Springer, Berlin, Heidelberg, December 2005.

[31] Go Ohtake, Arisa Fujii, Goichiro Hanaoka, and Kazuto Ogawa. On the theoretical gap between group signatures with and without unlinkability. In Bart Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 149–166. Springer, Berlin, Heidelberg, June 2009.

[32] Duong Hieu Phan and David Pointcheval. Chosen-ciphertext security without redundancy. In Chi-Sung Laih, editor, *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 1–18. Springer, Berlin, Heidelberg, November / December 2003.

[33] Duong Hieu Phan and David Pointcheval. OAEP 3-round: A generic and secure asymmetric encryption padding. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 63–77. Springer, Berlin, Heidelberg, December 2004.

[34] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990.

[35] Yusuke Sakai, Keita Emura, Goichiro Hanaoka, Yutaka Kawai, Takahiro Matsuda, and Kazumasa Omote. Group signatures with message-dependent opening. In Michel Abdalla and Tanja Lange, editors, *PAIRING 2012*, volume 7708 of *LNCS*, pages 270–294. Springer, Berlin, Heidelberg, May 2013.

[36] Yusuke Sakai, Keita Emura, Goichiro Hanaoka, Yutaka Kawai, and Kazumasa Omote. Methods for restricting message space in public-key encryption. *IEICE Fundamentals of Electronics, Communications and Computer Sciences*, 96-A(6):1156–1168, 2013.

[37] Yusuke Sakai, Jacob C. N. Schuldt, Keita Emura, Goichiro Hanaoka, and Kazuo Ohta. On the security of dynamic group signatures: Preventing signature hijacking. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 715–732. Springer, Berlin, Heidelberg, May 2012.