Proper Usage of the Group Signature Scheme in ISO/IEC 20008-2

Ai Ishida [*] Yusuke Sakai [†] Keita Emura [‡] Goichiro Hanaoka [§]
Keisuke Tanaka [¶]

February 18, 2019

Abstract

In ISO/IEC 20008-2, several anonymous digital signature schemes are specified. Among these, the scheme denoted as Mechanism 6, is the only plain group signature scheme that does not aim at providing additional functionalities. The Intel Enhanced Privacy Identification (EPID) scheme, which has many applications in connection with Intel Software Guard Extensions (Intel SGX), is in practice derived from Mechanism 6. In this paper, we firstly show that Mechanism 6 does not satisfy anonymity in the standard security model, i.e., the Bellare-Shi-Zhang model [CT-RSA 2005]. We then provide a detailed analysis of the security properties offered by Mechanism 6 and characterize the conditions under which its anonymity is preserved. Consequently, it is seen that Mechanism 6 is secure under the condition that the issuer, who generates user signing keys, does not join the attack. We also derive a simple patch for Mechanism 6 from the analysis.

Keywords: Group signature, Cryptanalysis, ISO/IEC 20008-2, SGX

1 Introduction

1.1 Background

The ISO/IEC standards are some of the most important reference documents representing a consensus among the experts in the field of information security. In practice, it is generally required to utilize the technologies which are specified in standards to ensure interoperability.

In the case of cryptographic technologies, standardization plays an even more important role of building trust. During the process of cryptographic standardizations, much work and time are required in order to carefully examine the security of a proposed scheme even if it has already been published in a flagship conference. Concretely, it typically takes about 2-3 years to standardize (and revise) a scheme. Due to this strict evaluation process, standardized schemes are some of the most trusted schemes in general.

The ISO/IEC 20008-2 standard [2], which is for privacy-enhanced user authentication technologies, was published in 2013. In this document, seven anonymous digital signature schemes (Mechanism 1 to 7) are specified. Among them, the scheme denoted as Mechanism 6, is the only plain group signature scheme [15] which does not aim at providing additional functionalities.

Due to its simplicity, Mechanism 6 is the most efficient group signature scheme in standards. Therefore, if we need to introduce a (plain) group signature scheme in a practical system, it is considered reasonable to employ Mechanism 6. In fact, the Intel Enhanced Privacy Identification (EPID) scheme [14]

[*]National Institute of Advanced Industrial Science and Technology (AIST), Japan. a.ishida@aist.go.jp
[†]National Institute of Advanced Industrial Science and Technology (AIST), Japan. yusuke.sakai@aist.go.jp
[‡]National Institute of Information and Communications Technology (NICT), Japan. k-emura@nict.go.jp
[§]National Institute of Advanced Industrial Science and Technology (AIST), Japan. hanaoka-goichiro@aist.go.jp
[¶]Tokyo Institute of Technology, Japan. keisuke@is.titech.ac.jp

is based on the Furukawa-Imai scheme [23, 24], from which Mechanism 6 originates.[1] The EPID scheme is an anonymous signature scheme for identification, and there are its many applications (see "Intel EPID Use Cases" in the web page [5] for details) represented by Intel Software Guard Extensions (Intel SGX) [6].

In terms of Mechanism 6's security, the ISO/IEC document says that the associated security proofs are based on the original paper [24]. More precisely, it is considered that Mechanism 6 is secure in the Bellare-Shi-Zhang (BSZ) model [11], which is one of the popular security models for group signatures.[2]

1.2 Our Contribution

In this paper, we firstly prove that Mechanism 6 is not secure in the BSZ model by showing a concrete attack against its anonymity, and then discuss possible countermeasures. Secondly, as the best countermeasure, we provide a detailed analysis of the security properties offered by Mechanism 6 and characterize the conditions under which its anonymity is preserved. Consequently, it is seen that Mechanism 6 is secure under the condition that the issuer does not join the attack. For example, *Mechanism 6 is secure if a unique organization simultaneously plays roles of both issuer and opener*. Finally, we derive a simple patch for Mechanism 6. In the following, we provide more details of our contributions.

Attack against Mechanism 6 in the BSZ model. We show an attack against the anonymity of Mechanism 6 in the BSZ model. More precisely, we show that the issuer, who generates user signing keys by the issuing key, can identify the signer of any signature although only the entity called the opener is allowed to trace the signer in the BSZ model.

In a nutshell, the reason why Mechanism 6 can be attacked is that the underlying proof system does not satisfy simulation soundness. If a proof system is not simulation sound, it might be possible to create a valid proof without a witness after seeing some valid proofs. We note that the proof of original paper [24] is not correct since it is misunderstood that the underlying proof system satisfies simulation soundness but it only satisfies soundness.

In Mechanism 6, this possibility allows an adversary to re-randomize the challenge signature and helps to break its anonymity. Specifically, in our attack, the challenge signature is re-randomized by using the issuing key. Then, the adversary queries the manipulated signature to the opening oracle and obtains the identity of the signer. Since the adversary is allowed to corrupt the issuer and to access the opening oracle in the anonymity game of the BSZ model, our attack is valid in this model.

Countermeasures for Our Attack. We consider the following three countermeasures for our attack: (1) to remove Mechanism 6 from the list and use alternative schemes in the standard, (2) to patch Mechanism 6 and update the document, and (3) to analyze the security properties offered by Mechanism 6 and restrict its use in a way that ensures that its anonymity is preserved.

The countermeasure (1) seems easy but is not desirable. At a first glance, Mechanism 5 and 7 might be considered reasonable substitutes for Mechanism 6. However, this is not always the case since Mechanism 5 and 7 have some drawbacks. More precisely, Mechanism 5 is significantly less efficient than Mechanism 6 due to the fact that Mechanism 5 is based on an RSA-type algebraic structure. Furthermore, Mechanism 7 provides only a weaker security notion of anonymity (the so-called "CPA-anonymity"). Therefore, countermeasure (1) is not very appropriate.

The countermeasure (2) is ideal and should be taken if possible. However, it cannot be carried out immediately since it takes much work and time to standardize a new scheme even though it is just an updated to an existing one. For example, in the case of the ISO/IEC 9796-2 standard [1], one of the standardized schemes was attacked by Coron et al. [16] in 1999, but the final revised version was published in 2002. That is, it took three years to update the document. Thus, although it will most certainly be useful to provide a patched scheme, it is not an immediate countermeasure for the attack.

The countermeasure (3) seems most realistic among the possible countermeasures. Although we see that Mechanism 6 does not satisfy the expected security level by our attack, it is premature to rule out Mechanism 6 as a useful scheme. Specifically, it might be true that Mechanism 6 is still secure to use in

[1] The EPID scheme is listed as Mechanism 3 in the ISO/IEC 20008-2 [2]. We can find the explicit description that the EPID scheme is derived from the Furukawa-Imai scheme in the paper [14] and the conference material [7].

[2] The model in the papers [23, 24] is slightly different from the BSZ model. However, it is easy to see that they are essentially same.

practice since the BSZ model considers a relatively strong level of security. Therefore in this work, we investigate this countermeasure as we consider that this is the most appropriate one.

Rigorous Security Evaluation of Mechanism 6. As mentioned above, the countermeasure (3) is most appropriate among the possible countermeasures. Therefore, we analyze the security properties offered by Mechanism 6 in order to characterize the conditions under which its anonymity is preserved.

As a result of this analysis, we see that no one can extract the signer's information from a signature except for the opener and the issuer. More precisely, this fact indicates that *Mechanism 6 is still secure under the condition that the issuer does not join the attack.* Such a condition is reasonable if a single authority plays roles of both the opener and the issuer.

We stress that finding out the strict security of Mechanism 6 is quite non-trivial, and then it can be considered a theoretically interesting problem. As we mentioned, the flaw of Mechanism 6 is that the underlying proof system does not satisfy simulation soundness, and this property allows to break the anonymity by re-randomizing the challenge signature. In our analysis, we firstly show that such an attack is the only way to break the anonymity. However, it is not very clear how to defend against this attack since it is difficult to find out what essentially allows an adversary to make such an attack. Then, we determine to minutely divide (i.e., 31 cases) this attack and analyze each case one by one. Finally, we give its complete analysis and find out the strict condition to securely use Mechanism 6. Our approach looks simple once it has been described, but we think that it is not so easy to take this approach in practice.

In addition, the formal proof of the Mechanism 6's strict security is non-trivial and non-standard although its intuition can be obtained from the above analysis. Generally, the anonymity of a group signature scheme reduces to the confidentiality of the underlying public key encryption scheme and the zero-knowledgeness of the underlying non-interactive zero-knowredge proof system, and does not reduce to the unforgeability of the underlying signature scheme. However, in the case of Mechanism 6, we also reduce to the unforgeability of the signature scheme since claiming that the issuing key that is essentially a signing key of the signature scheme can be extracted from an adversary breaking the anonymity. Therefore, the reduction algorithm is required to manage to generate users' certificates without the issuing key. For this reason, the proof of the Mechanism 6's security is complicated.

A Patched Scheme. Owning to our analysis of the security of Mechanism 6, we derive a non-trivial patch for the scheme. In fact, it is not so hard to come up with a patched scheme just secure in the BSZ model, but a scheme with a small patch is non-trivial. Our patched scheme could be a candidate for the new standardized scheme when ISO/IEC 20008-2 will be revised in the future.

In the patched scheme, only the signing and verification algorithms are changed, and the signature size increases by only one element in the group $\mathbb{G}_1$ where $\mathbb{G}_1$ is a source group in the used asymmetric bilinear group. More precisely, a signature in the patched scheme consists of two elements from $\mathbb{G}_1$, three elements from $\mathbb{G}$, and six elements from $\mathbb{Z}_p$ (where $\mathbb{G}$ is the group in which the decisional Diffie-Hellman assumption holds). This achieves the comparable efficiency to the existing schemes [18, 19] satisfying the same security level. Also, we need to introduce the external Diffie-Hellman assumption in $\mathbb{G}_1$ to prove the anonymity of the patched scheme, but the other security requirements can be showed under the same assumptions as those of Mechanism 6.

1.3 Paper Organization

In Section 2, we review basic notations, and the definitions of computational assumptions and cryptographic primitives which we use in this paper. Mechanism 6 is reviewed also in Section 2. In Section 3, we describe an attack against the anonymity of Mechanism 6 in the BSZ model and discuss about its countermeasures. In Section 4, we analyze the security properties offered by Mechanism 6. More precisely, we prove that Mechanism 6 satisfies anonymity if the adversary does not make the type of attacks which we give in Section 4.1, and provide further analysis of the attack in Section 4.2. From the result in this section, we can characterize the conditions under which the anonymity of Mechanism 6 is preserved. Then in Section 4.3, we formalize these conditions and prove the strict security of Mechanism 6 under these. In Section 4.4, we discuss the practical implications of our results. Furthermore, we give a patch for Mechanism 6 in Section 5. Lastly, we conclude this paper in Section 6.

2 Preliminaries

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 $\mathcal{O}$. If A and B are statements, $A \Leftrightarrow B$ denotes that A and B are equivalent. If a_i is an indexed element, $\{a_i\}_i$ denotes an ordered set arranged in the index order. 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 Complexity Assumptions

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be multiplicative cyclic groups of order p where p is a λ-bit prime. Let G_1 and G_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Let Ψ be an isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ with $\Psi(G_2) = G_1$. Let e be a computable map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with bilinearity: for all $a, b \in \mathbb{Z}$, $e({G_1}^a, {G_2}^b) = e(G_1, G_2)^{ab}$, and non-degeneracy: $e(G_1, G_2) \neq 1$. We say that groups $(\mathbb{G}_1, \mathbb{G}_2)$ are a bilinear group pair if there exist the map Ψ and the bilinear map e as above, and the group operations in $\mathbb{G}_1$ and $\mathbb{G}_2$, the map Ψ, and the bilinear map e are efficiently computable. In this paper, we consider bilinear maps $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are groups of prime order p.

We define the discrete logarithm (DL) assumption, the external Diffie-Hellman (XDH) assumption, and the q-strong Diffie-Hellman (q-SDH) assumption.

Definition 2.1 (Discrete Logarithm Assumption)**.** *We say that the DL assumption holds in $\mathbb{G}_1$ if for any PPT adversary $\mathcal{A}$, the advantage $\mathrm{Adv}^{DL}_{\mathcal{A}}(\lambda) := \Pr[H = {G_1}^x | x \leftarrow \mathcal{A}(G_1, G_2, H)]$ is negligible, where the probability is taken over the random choices of a generator $G_2 \in \mathbb{G}_2$ with $G_1 = \Psi(G_2)$, of an element $H \in \mathbb{G}_1$, and a random coin of $\mathcal{A}$.*

Definition 2.2 (External Diffie-Hellman Assumption)**.** *We say that the XDH assumption holds in $\mathbb{G}_1$ if for any PPT adversary $\mathcal{A}$, the advantage $\mathrm{Adv}^{XDH}_{\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(G_1, G_2, {G_1}^a, {G_1}^b, {G_1}^{ab})] - \Pr[1 \leftarrow \mathcal{A}(G_1, G_2, {G_1}^a, {G_1}^b, W)]|$ is negligible, where the probability is taken over the random choices of a generator $G_2 \in \mathbb{G}_2$ with $G_1 = \Psi(G_2)$, of elements $a, b \in \mathbb{Z}_p$, and of an element $W \in \mathbb{G}_1$, and a random coin of $\mathcal{A}$.*

Definition 2.3 (q-Strong Diffie-Hellman Assumption)**.** *We say that the q-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if for any PPT adversary $\mathcal{A}$, the advantage $\mathrm{Adv}^{q\text{-}SDH}_{\mathcal{A}}(\lambda) := \Pr[e(C, {G_2}^\gamma \cdot {G_2}^x) = e(G_1, G_2) | (C, x) \leftarrow \mathcal{A}(G_1, G_2, {G_2}^\gamma, {G_2}^{\gamma^2}, \ldots, {G_2}^{\gamma^q})]$ is negligible, where the probability is taken over the random choices of a generator $G_2 \in \mathbb{G}_2$ with $G_1 = \Psi(G_2)$ and of a value $\gamma \in \mathbb{Z}_p^*$, and a random coin of $\mathcal{A}$.*

For simplifying a security proof, we also introduce the simplified q-strong Diffie-Hellman (simplified q-SDH) assumption.

Definition 2.4 (Simplified q-Strong Diffie-Hellman Assumption [13])**.** *We say that the simplified q-SDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if for any PPT adversary $\mathcal{A}$, the advantage $\mathrm{Adv}^{sim\text{-}q\text{-}SDH}_{\mathcal{A}}(\lambda) := \Pr[x \neq x_i \wedge e(C, {G_2}^\gamma \cdot {G_2}^x) = e(G_1, G_2) | (C, x) \leftarrow \mathcal{A}(G_1, G_2, {G_2}^\gamma, \{{G_1}^{\frac{1}{\gamma + x_i}}, x_i\}_{i=1}^q)]$ is negligible, where the probability is taken over the random choices of a generator $G_2 \in \mathbb{G}_2$ with $G_1 = \Psi(G_2)$, of a value $\gamma \in \mathbb{Z}_p^*$, and of values $x_i \in \mathbb{Z}_p$, and a random coin of $\mathcal{A}$.*

The following theorem is known between the q-SDH assumption and the simplified $(q-1)$-SDH assumption. Therefore, we use the simplified $(q-1)$-SDH assumption instead of the q-SDH assumption in our security proof.

Theorem 2.1 ([13])**.** *For any PPT adversary $\mathcal{A}$ and any integer $q > 0$, it holds that $\mathrm{Adv}^{sim\text{-}(q-1)\text{-}SDH}_{\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{q\text{-}SDH}_{\mathcal{A}}(\lambda)$. That is, if the q-SDH assumption holds, the simplified $(q-1)$-SDH assumption also holds.*

In Mechanism 6, another multiplicative cyclic group $\mathbb{G}$ of order p in which the decisional Diffie-Hellman (DDH) assumption holds is introduced. We define the DDH assumption in the following.

Definition 2.5 (Decisional Diffie-Hellman Assumption)**.** *We say that the DDH assumption holds in* $\mathbb{G}$ *if for any PPT adversary* $\mathcal{A}$*, the advantage* $\mathrm{Adv}_{\mathcal{A}}^{DDH}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(G, G^a, G^b, G^{ab})] - \Pr[1 \leftarrow \mathcal{A}(G, G^a, G^b, W)]|$ *is negligible, where the probability is taken over the random choices of a generator* $G \in \mathbb{G}$*, of elements* $a, b \in \mathbb{Z}_p$*, and of an element* $W \in \mathbb{G}$*, and a random coin of* $\mathcal{A}$*.*

2.2 Group Signature

In this section, we review group signature. Here, we follow the Bellare-Shi-Zhang (BSZ) model [11]. A group signature scheme Π_{GS} consists of the following algorithms (GKg, UKg, $\mathsf{Join/Iss}$, GSig, GVf, Open, Judge).

GKg: The group key generation algorithm takes as input a security parameter 1^λ ($\lambda \in \mathbb{N}$), and returns a group public key gpk, an issuing key ik, and an opening key ok.

UKg: The user key generation algorithm, which is run by a user i, takes as input 1^λ and gpk, and returns a public and secret key pair $(\mathsf{upk}_i, \mathsf{usk}_i)$.

$\mathsf{Join/Issue}$: The pair of (interactive) algorithms are run by a user i and the issuer, and takes as input gpk, upk_i, and usk_i from the user i, and gpk, upk_i, and ik from the issuer, respectively. If it is successful, the issuer stores the registration information of the user i in $\mathsf{reg}[i]$ and the user obtains the corresponding signing key gsk_i. We denote $\mathsf{reg} = \{\mathsf{reg}[i]\}_i$.

GSig: The signing algorithm takes as input gpk, gsk_i, and a message m, and returns a group signature Σ.

GVf: The verification algorithm takes as input gpk, Σ, and m, and returns either 1 (indicating that Σ is a valid group signature on m), or 0.

Open: The opening algorithm takes as input gpk, ok, m, Σ, and reg, and returns either (i, τ) or $\perp$ where i is a user identity and τ is a proof that the user i computed Σ. The symbol $\perp$ indicates that the opening procedure fails.

Judge: The judge algorithm takes as input gpk, i, upk_i, m, Σ, and τ, and returns either 1 (indicating that Σ is produced by the user i), or 0.

Bellare et al. [11] formalized correctness, anonymity, non-frameability, and traceability as security requirements. Here, we give only the definition of anonymity since focusing on the anonymity of Mechanism 6.

Firstly, we give the definitions of some oracles. The SndToU oracle is an interactive oracle. Also, HU and CU are the set of honest users and corrupted users, respectively.

$\mathsf{CrptU}(\cdot, \cdot)$: The corrupt-user oracle takes as input a user identity i and upk. This oracle sets $\mathsf{upk}_i \leftarrow \mathsf{upk}$ and adds i to CU.

$\mathsf{SndToU}(\cdot)$: The send-to-user oracle takes as input a user identity i. At first, the oracle produces a user public and secret key pair $(\mathsf{upk}_i, \mathsf{usk}_i) \leftarrow \mathsf{UKg}(1^\lambda, \mathsf{gpk})$ and adds i to HU. Then he interacts with the adversary who corrupts the issuer by running $\mathsf{Join}(\mathsf{gpk}, \mathsf{upk}_i, \mathsf{usk}_i)$. The user i needs to be neither in the set HU nor the set CU. If so, the oracle outputs $\perp$.

$\mathsf{USK}(\cdot)$: The user secret keys oracle takes as input i, and returns the secret keys usk_i and gsk_i if $i \in \mathsf{HU}$. If not, the oracle returns $\perp$.

$\mathsf{WReg}(\cdot, \cdot)$: The write-registration-table oracle takes as input i and a value $\widehat{\mathsf{reg}}$, and writes or modifies the contents of reg by setting $\mathsf{reg}[i] \leftarrow \widehat{\mathsf{reg}}$.

$\mathsf{Ch}(\cdot, \cdot, \cdot, \cdot)$: The challenge oracle takes as input a bit b, two identities i_0, i_1, and a message m^*, and returns $\Sigma^* \leftarrow \mathsf{GSig}(\mathsf{gpk}, \mathsf{gsk}_{i_b}, m^*)$ if both $i_0 \in \mathsf{HU}$ and $i_1 \in \mathsf{HU}$. If not, the oracle returns $\perp$. In this paper, we call b a challenge bit, m^* a challenge message, Σ^* a challenge signature, and i_0, i_1 challenge users.

$\mathsf{Open}(\cdot, \cdot)$: The opening oracle takes as input m and Σ, and returns $(i, \tau) \leftarrow \mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, m, \Sigma, \mathsf{reg})$ if $(m, \Sigma) \neq (m^*, \Sigma^*)$. If not, the oracle returns $\perp$.

Then, we describe the definition of anonymity given in the BSZ model. Intuitively, it ensures that the adversary who can corrupt all users and the issuer cannot extract the information of the signer from a group signature. The formal definition is given as follows.

$\mathsf{GKg}(1^\lambda)$:

$G_2 \xleftarrow{\$} \mathbb{G}_2;\ G \xleftarrow{\$} \mathbb{G};\ G_1 \leftarrow \Psi(G_2);\ \mathsf{H} \xleftarrow{\$} \mathcal{H}$

$H, K \xleftarrow{\$} \mathbb{G}_1;\ w \xleftarrow{\$} \mathbb{Z}_p;\ u, v \xleftarrow{\$} \mathbb{Z}_p^*;\ Y \leftarrow {G_2}^w;\ U \leftarrow G^u;\ V \leftarrow G^v$

Return $(\mathsf{gpk}, \mathsf{ik}, \mathsf{ok}) = ((G_1, G_2, G, \mathsf{H}, H, K, Y, U, V), w, (u, v))$

$\mathsf{UKg}(1^\lambda, \mathsf{gpk})$:

$x_i, z_i' \xleftarrow{\$} \mathbb{Z}_p;\ Q_i \leftarrow G^{x_i};\ H_i \leftarrow H^{x_i} K^{z_i'}$

Return $(\mathsf{upk}_i, \mathsf{usk}_i) = ((Q_i, H_i), (x_i, z_i'))$

$\mathsf{Join}/\mathsf{Issue}$(User i: $\mathsf{gpk}, \mathsf{upk}_i, \mathsf{usk}_i$; Issuer: $\mathsf{gpk}, \mathsf{upk}_i, \mathsf{ik}$):

User: $\rho_{x_i}, \rho_{z_i'} \xleftarrow{\$} \mathbb{Z}_p;\ R_1 \leftarrow G^{\rho_{x_i}};\ R_2 \leftarrow H^{\rho_{x_i}} K^{\rho_{z_i'}}$

Send (R_1, R_2) to the issuer

Issuer: $c_i \xleftarrow{\$} \mathbb{Z}_p$

Send c_i to the user

User: $\sigma_{x_i} \leftarrow x_i \cdot c_i + \rho_{x_i};\ \sigma_{z_i'} \leftarrow z_i' \cdot c_i + \rho_{z_i'}$

Send $(\sigma_{x_i}, \sigma_{z_i'})$ to the issuer

Issuer: $R_1' \leftarrow G^{\sigma_{x_i}} / {Q_i}^{c_i};\ R_2' \leftarrow H^{\sigma_{x_i}} K^{\sigma_{z_i'}} / {H_i}^{c_i}$

Return $\perp$ to the user if $R_1' \neq R_1 \vee R_2' \neq R_2$

$y_i, z_i'' \xleftarrow{\$} \mathbb{Z}_p;\ A_i \leftarrow \left(\frac{G_1}{H_i \cdot K^{z_i''}}\right)^{\frac{1}{w+y_i}};\ \mathsf{reg}[i] \leftarrow Q_i$

Send (A_i, y_i, z_i'') to the user

User: $z_i \leftarrow z_i' + z_i''$

Set $\mathsf{gsk}_i \leftarrow (A_i, y_i, z_i, x_i, Q_i)$ if $e(A_i, Y \cdot {G_2}^{y_i}) e(H^{x_i}, G_2) e(K^{z_i}, G_2) = e(G_1, G_2)$

$\mathsf{GSig}(\mathsf{gpk}, \mathsf{gsk}_i, m)$:

$r, q \xleftarrow{\$} \mathbb{Z}_p;\ T_1 \leftarrow A_i \cdot K^q;\ T_2 \leftarrow G^{x_i + r};\ T_3 \leftarrow U^r;\ T_4 \leftarrow V^r;\ \rho_{x_i}, \rho_{y_i}, \rho_\delta, \rho_q, \rho_r \xleftarrow{\$} \mathbb{Z}_p$

$R_1 \leftarrow e(H, G_2)^{\rho_{x_i}} \cdot e(K, G_2)^{\rho_\delta} \cdot e(K, Y)^{-\rho_q} \cdot e(T_1, G_2)^{\rho_{y_i}};\ R_2 \leftarrow G^{\rho_{x_i} + \rho_r};\ R_3 \leftarrow U^{\rho_r};\ R_4 \leftarrow V^{\rho_r}$

$c \leftarrow \mathsf{H}(\mathsf{gpk}, \{T_i\}_{i \in [1,4]}, \{R_i\}_{i \in [1,4]}, m);\ \delta \leftarrow z_i - q y_i$

$\sigma_{x_i} \leftarrow x_i \cdot c + \rho_{x_i};\ \sigma_{y_i} \leftarrow y_i \cdot c + \rho_{y_i};\ \sigma_\delta \leftarrow \delta \cdot c + \rho_\delta;\ \sigma_q \leftarrow q \cdot c + \rho_q;\ \sigma_r \leftarrow r \cdot c + \rho_r$

Return $\Sigma = (\{T_i\}_{i \in [1,4]}, c, \sigma_{x_i}, \sigma_{y_i}, \sigma_\delta, \sigma_q, \sigma_r)$

$\mathsf{GVf}(\mathsf{gpk}, m, \Sigma)$:

$R_1' \leftarrow e(H, G_2)^{\sigma_{x_i}} \cdot e(K, G_2)^{\sigma_\delta} \cdot e(K, Y)^{-\sigma_q} \cdot e(T_1, G_2)^{\sigma_{y_i}} \cdot \left(\frac{e(G_1, G_2)}{e(T_1, Y)}\right)^{-c}$

$R_2' \leftarrow G^{\sigma_{x_i} + \sigma_r} \cdot {T_2}^{-c};\ R_3' \leftarrow U^{\sigma_r} \cdot {T_3}^{-c};\ R_4' \leftarrow V^{\sigma_r} \cdot {T_4}^{-c}$

Return 1 if $c = \mathsf{H}(\mathsf{gpk}, \{T_i\}_{i \in [1,4]}, \{R_i'\}_{i \in [1,4]}, m)$, else return 0

$\mathsf{Open}(\mathsf{gpk}, \mathsf{ok}, \mathsf{reg}, m, \Sigma)$:

Return $\perp$ if $\mathsf{GVf}(\mathsf{gpk}, m, \Sigma) = 0$

$Q \leftarrow T_2 \cdot ({T_3}^{\frac{1}{u}})^{-1}$

Return $\perp$ if there is no user index i such that $\mathsf{reg}[i] = Q$

$\rho_u \xleftarrow{\$} \mathbb{Z}_p;\ R \leftarrow (Q \cdot {T_2}^{-1})^{\rho_u};\ d \leftarrow \mathsf{H}(\mathsf{gpk}, Q, T_2, T_3, R);\ \sigma_u \leftarrow u \cdot d + \rho_u;\ \tau \leftarrow (d, \sigma_u)$

Return (i, τ)

$\mathsf{Judge}(\mathsf{gpk}, \mathsf{reg}, m, \Sigma, (i, \tau))$:

Return $\perp$ if $\mathsf{GVf}(\mathsf{gpk}, m, \Sigma) = 0$

$Q \leftarrow \mathsf{reg}[i];\ R' \leftarrow (Q \cdot {T_2}^{-1})^{\sigma_u} \cdot {T_3}^{-d}$

Return 1 if $d = \mathsf{H}(\mathsf{gpk}, Q, T_2, T_3, R')$, else return 0

Figure 1: Mechanism 6

Definition 2.6 (Anonymity [11]). *Let $\mathcal{A}$ be an adversary for anonymity. We define the experiment $\mathrm{Exp}^{anon}_{\Pi_{\mathrm{GS}}, \mathcal{A}}(\lambda)$ as follows.*

$$\mathrm{Exp}^{anon}_{\Pi_{\mathrm{GS}}, \mathcal{A}}(\lambda):\ b \leftarrow \{0,1\};\ (\mathsf{gpk}, \mathsf{ik}, \mathsf{ok}) \leftarrow \mathsf{GKg}(1^k);\ \mathsf{CU} \leftarrow \emptyset;\ \mathsf{HU} \leftarrow \emptyset$$
$$\widetilde{b} \leftarrow \mathcal{A}^{\mathsf{CrptU}(\cdot,\cdot), \mathsf{SndToU}(\cdot), \mathsf{USK}(\cdot), \mathsf{WReg}(\cdot,\cdot), \mathsf{Ch}(b,\cdot,\cdot,\cdot), \mathsf{Open}(\cdot,\cdot)}(\mathsf{gpk}, \mathsf{ik})$$
$$\textit{Return } 1 \textit{ if } \widetilde{b} = b, \textit{ otherwise return } 0$$

We say that Π_{GS} *satisfies anonymity if the advantage*

$$\mathrm{Adv}^{anon}_{\Pi_{\mathrm{GS}},\mathcal{A}} := \left|\Pr[\mathrm{Exp}^{anon}_{\Pi_{\mathrm{GS}},\mathcal{A}}(\lambda)=1] - \frac{1}{2}\right|$$

is negligible for any PPT adversary $\mathcal{A}$*.*

2.3 Mechanism 6

In this section, we review Mechanism 6, which is identical to the Furukawa-Imai scheme [23, 24], in the ISO/IEC 20008-2 standard [2]. The formal description is given in Figure 1. Although their model is slightly different from the BSZ model [11], it is easily seen that they are essentially same. Therefore in this paper, we introduce Mechanism 6 by using the algorithms given by Bellare et al. [11]. Originally, the judging algorithm is not defined in Mechanism 6. However, we also describe its judging algorithm since it can be defined implicitly.

Consider a bilinear group pair $(\mathbb{G}_1, \mathbb{G}_2)$ with a computable isomorphism Ψ, and a group $\mathbb{G}$ in which the DDH assumption holds.[3] Here, we denote elements in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, and $\mathbb{G}$ by upper case letters, and elements in $\mathbb{Z}_p$ by lower case letters. $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$ is a family of hash functions treated as random oracles in security proofs.

In Mechanism 6, a user i possesses a SDH pair (A_i, y_i) and a discrete logarithm x_i as a signing key where A_i is the certificate of x_i. When signing on a message m, the user encrypts the certificate A_i and the value $Q_i = G^{x_i}$, and generates a signature of knowledge on m for the statement that the encrypted certificate is valid, and the encryption procedure is honestly done. The signature is accepted when the signature of knowledge is valid. When opening a signature, the opener extracts Q_i by using the decryption key and outputs the ID i with a proof which shows that the decryption is honestly done.

3 Attack against Mechanism 6 in the BSZ Model

In this section, we give an attack against the anonymity of Mechanism 6 and prove that it is not secure in the BSZ model. In a nutshell, the reason why Mechanism 6 can be broken is that the underlying proof system does not satisfy simulation soundness. If a proof system is not simulation sound, it might be possible to create a valid proof without a witness after seeing some valid proofs.

In Mechanism 6, this possibility of creating a valid proof allows for the adversary to re-randomize the challenge signature and helps to break the anonymity. Specifically, in our attack, the challenge signature is re-randomized by using the issuing key. Then, the adversary queries the re-randomized signature to the opening oracle and can obtain the identity of the signer. Since the adversary is allowed to corrupt the issuer and to access the opening oracle in the anonymity game of the BSZ model, our attack is valid in this model. In the following, we provide more details of our attack.

Firstly, we show that the underlying proof system does not satisfy simulation soundness. In the proof system, for the group public key gpk and values $\{T_i\}_{i\in[1,4]}$, four equations are proved with witnesses x, y, δ, q, and r. A valid proof $\sigma_{\mathsf{set}} = \{\sigma_x, \sigma_y, \sigma_\delta, \sigma_q, \sigma_r\}$ satisfies the following equations:

$$R_1 = e(H, G_2)^{\sigma_x} \cdot e(K, G_2)^{\sigma_\delta} \cdot e(K, Y)^{-\sigma_q} \cdot e(T_1, G_2)^{\sigma_y} \cdot \left(\frac{e(G_1, G_2)}{e(T_1, Y)}\right)^{-c},$$
$$R_2 = G^{\sigma_x + \sigma_r} \cdot {T_2}^{-c},\ R_3 = U^{\sigma_r} \cdot {T_3}^{-c},\ R_4 = V^{\sigma_r} \cdot {T_4}^{-c}$$

where R_1, R_2, R_3, and R_4 are the commitments generated in the way of computing a signature, and c is a challenge value computed as $c \leftarrow \mathsf{H}(\mathsf{gpk}, \{T_i\}_{i\in[1,4]}, \{R_i\}_{i\in[1,4]}, m)$ for a message m. When we focus on the first equation, the second and third terms of the right side on the equation have a common base $e(K, G_2)$ since $Y = {G_2}^w$ holds for the issuing key w. Thus, we can denote that $e(K, G_2)^{\sigma_\delta} \cdot e(K, Y)^{-\sigma_q} = e(K, G_2)^{\sigma_\delta - \sigma_q \cdot w}$.

In fact, this property allows to break the simulation soundness by shuffling the discrete logarithms σ_δ and $-\sigma_q$. Now, we set $\widetilde{\sigma}_\delta = \sigma_\delta + w$ and $\widetilde{\sigma}_q = \sigma_q + 1$ where the values can be computed from the issuing key and a given valid proof. Then, the proof $\widetilde{\sigma}_{\mathsf{set}} = \{\sigma_x, \sigma_y, \widetilde{\sigma}_\delta, \widetilde{\sigma}_q, \sigma_r\}$ also satisfies the above

[3] The isomorphism Ψ is used in the security proof of the traceability. Since we focus on the anonymity, the isomorphism Ψ appears only in the setup phase in this paper.

equations. The first equation holds since it holds that $e(K, G_2)^{\widetilde{\sigma}_\delta} \cdot e(K, Y)^{-\widetilde{\sigma}_q} = e(K, G_2)^{\widetilde{\sigma}_\delta - \widetilde{\sigma}_q \cdot w} = e(K, G_2)^{\sigma_\delta + w - (\sigma_q + 1) \cdot w} = e(K, G_2)^{\sigma_\delta - \sigma_q \cdot w} = e(K, G_2)^{\sigma_\delta} \cdot e(K, Y)^{-\sigma_q}$, and the other equations hold trivially. Therefore, the forgery $\widetilde{\sigma}_{\mathsf{set}}$ is valid as an attack against the simulation soundness of the underlying proof system in the sense that it can be generated without a witness after seeing some valid proofs.

Secondly, we show that the above forgery against the simulation soundness derives an attack against the anonymity of Mechanism 6. In the anonymity game of the BSZ model, the adversary is allowed to corrupt the issuer. Thus, the adversary can compute a re-randomized signature $\widetilde{\Sigma}$ for the challenge signature Σ^* as above. Also, since the adversary can access the opening oracle, and the re-randomized signature is not the same as the challenge signature (that is, $\widetilde{\Sigma} \neq \Sigma^*$ holds), the adversary can query the signature $\widetilde{\Sigma}$ to the opening oracle. Here, the signer's information hidden in the re-randomized signature is the same as that of the challenge signature since the difference between them is only the proof part. Thus, the adversary obtains the signer's ID of the challenge signature by this query. In this way, the anonymity of Mechanism 6 can be broken.

Countermeasures for Our Attack. We can consider the following three countermeasures for our attack: (1) to remove Mechanism 6 from the standards and use alternative schemes, (2) to patch Mechanism 6 and update the document, and (3) to analyze the security properties offered by Mechanism 6 and restrict its use in a way that ensures that its anonymity is preserved. In the following, we provide more details of each countermeasure.

The countermeasure (1): This countermeasure seems easy but is not desirable. In fact, Mechanism 5 and 7 in the ISO/IEC 20008-2 standard are also group signature schemes in a broad sense. In addition to the functionality of group signatures, Mechanism 5 (the original paper [26]) introduces a special authority called a user-revocation manager, and Mechanism 7 has a functionality called controllable linkability [25]. Therefore, at a first glance, Mechanism 5 and 7 might be considered reasonable substitutes for Mechanism 6. However, it is not always the case since Mechanism 5 and 7 have some drawbacks. Concretely, Mechanism 5 is significantly less efficient than Mechanism 6 due to the fact that Mechanism 5 is based on an RSA-type algebraic structure. Furthermore, Mechanism 7 provides only weaker security notion of anonymity, CPA-anonymity. This indicates that in Mechanism 7, once the opening result of at least one signature is revealed to the public, the anonymity of signatures is no more ensured. Therefore, the countermeasure (1) is not very appropriate because of these drawbacks.

The countermeasure (2): This countermeasure is ideal and should be taken if possible. However, it cannot be carried out immediately since it takes much work and time to standardize a new scheme even though it is just an updated to an existing one. For example, in the case of the ISO/IEC 9796-2 standard [1] that specifies digital signature schemes for smart cards, one of the standardized schemes (denoted as Scheme 1) was attacked by Coron et al. [16] in 1999,[4] but the final revised version was not published before 2002. Specifically in this case, when it was seen that Scheme 1 is not secure, RSA-PSS [10] was known to be an adequate scheme to replace Scheme 1. That is, it took three long years to finally update the document even though there already existed such a candidate for an alternative scheme. (By the way, due to this delay of the update, Scheme 1 had populated a lot of commercial products (e.g., e-passports [3] and EMV cards [4]).) Therefore, the countermeasure (2) is not immediate countermeasure for the attack.

The countermeasure (3): This countermeasure seems most realistic among the possible countermeasures. Although we see that Mechanism 6 does not satisfy the expected security level by our attack, it is premature to rule out Mechanism 6 as a useful scheme. Specifically, it might be that Mechanism 6 is still secure to use in practice since the BSZ model considers a relatively strong level of security, e.g., dynamic model, double authority, and CCA-anonymity. For example, since the BSZ model considers double authority, all of entities except for the opener can corrupt in the anonymity game of this model. However, this seems not necessarily a real threat. Therefore, the countermeasure (3) seems most reasonable among the possible countermeasures.

From the above discussion, we investigate the countermeasure (3) as we consider that this is the most appropriate one and analyze the security of Mechanism 6 in the next section.

[4] Coron, Naccache and Stern [16] discovered that Scheme 1 is existentially forgeable in theory. After that, Coron, Naccache, Tibouchi, and Weinmann [17] showed a practical forgery for Scheme 1 in 2009.

4 Rigorous Security Evaluation of Mechanism 6

In the previous section, we see that Mechanism 6 does not satisfy anonymity in the BSZ model, that is, it does not satisfy the expected security level in the ISO/IEC document. Here, as the most appropriate countermeasure, we analyze the security properties offered by Mechanism 6 and characterize the conditions under which its anonymity is preserved.

As we mentioned, the flaw of Mechanism 6 is that the underlying proof system does not satisfy simulation soundness, and this property allows to break the anonymity by re-randomizing the challenge signature. In fact, it seems that such an attack is the only way to break the anonymity of Mechanism 6 since the scheme is well structured except for the proof part.

Therefore, we analyze the security of Mechanism 6 in the following way: Firstly, we prove that Mechanism 6 satisfies anonymity under the restricted condition that the adversary does not make such a type of attack (in Section 4.1). Secondly, we provide further analysis of the attack by classifying some cases depending on the types of the adversary's queries (in Section 4.2). From the result of this analysis, we can characterize the conditions under which the anonymity of Mechanism 6 is preserved. Finally, we formalize these conditions and formally prove the strict security of Mechanism 6 under these (in Section 4.3).

4.1 A Proof for the Anonymity of Mechanism 6 under the Restricted Condition

In this section, we formalize the attack to re-randomize the challenge signature by forging its proof part and query it to the open oracle, and then show that Mechanism 6 is secure if the adversary does not make this type of attack. More precisely, we formalize a query of a re-randomized signature generated by forging the proof part (called "related query" in the following), and then prove that Mechanism 6 satisfies anonymity against the adversary who does not generate any such a type of queries.

Firstly, we define a related query. Intuitively, a related query is a query which is obtained by re-randomizing the challenge signature through changing only the proof part. Let m^* and $\Sigma^* = (\{T_i^*\}_{i\in[1,4]}, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*)$ be the challenge message and the challenge signature, respectively. Formally, a related query is defined as follows.

A Related Query: We say that a query $(\widetilde{m}, \widetilde{\Sigma} = (\{\widetilde{T}_i\}_{i\in[1,4]}, \widetilde{c}, \widetilde{\sigma}_x, \widetilde{\sigma}_y, \widetilde{\sigma}_\delta, \widetilde{\sigma}_q, \widetilde{\sigma}_r))$ is a related query if $(\widetilde{m}, \widetilde{\Sigma})$ is accepted by the GVf algorithm, and it holds that

$$(\{\widetilde{T}_i\}_{i\in[1,4]}, \{\widetilde{R}_i\}_{i\in[1,4]}, \widetilde{m}) = (\{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*)$$

where $\{\widetilde{R}_i\}_{i\in[1,4]}$ and $\{R_i^*\}_{i\in[1,4]}$ are the intermediate values computed in the verification of pairs $(\widetilde{m}, \widetilde{\Sigma})$ and (m^*, Σ^*), respectively. However, we do not regard the pair (m^*, Σ^*) itself as a related query since it is not accepted by the opening oracle.

Then, we prove that Mechanism 6 satisfies anonymity if the adversary does not generate a related query. We provide the games Game from 0 to 7, and prove that for $0 \le \ell \le 6$, the advantages of the adversary in Game ℓ and Game $\ell+1$ are almost the same (which we denote Game $\ell \approx$ Game $\ell+1$). Game 0 is the original anonymity game and Game 7 is the game that the adversary wins with the probability 1/2. In fact for $\ell \neq 5$, it holds that Game $\ell \approx$ Game $\ell+1$ for the adversary without restriction on querying. However, when proving Game 5 $\approx$ Game 6, we need the condition that the adversary who does not generate a related query. Formally, we prove the following theorem.

Theorem 4.1. *If the adversary does not generate a related query, Mechanism 6 satisfies anonymity under the DDH assumption in the group $\mathbb{G}$ in the random oracle model.*

Proof. Let $\mathcal{A}$ be an adversary that attacks the anonymity of Mechanism 6 (in the following, the scheme is denoted as Π_{FI}). We consider the following sequence of games. Let S_ℓ denote the event that $\mathcal{A}$ succeeds in guessing the challenge bit b in Game ℓ.

[Game 0]: This is the experiment $\mathrm{Exp}^{anon}_{\Pi_{\mathsf{FI}},\mathcal{A}}(\lambda)$ itself. The challenger manages an inout/output pair of the random oracle in the list L. More precisely, when the adversary queries x to the random oracle, the challenger returns y if there is a pair (x, y) in L. On the other hand if there is no pair $(x, \cdot)$ in L, the challenger samples a value y uniform randomly and returns y to the adversary. Then, the challenger

adds (x, y) to the list L. In the following, we denote $y = \mathsf{H}(x)$ if there exists a pair (x, y) in the list. For the sake of convenience, we assume that the adversary queries $(\mathsf{gpk}, \{T_i\}_{i\in[1,4]}, \{R_i\}_{i\in[1,4]}, m)$ to the random oracle before he queries $(m, \Sigma = \{T_i\}_{i\in[1,4]}, c, \sigma_x, \sigma_y, \sigma_\delta, \sigma_q, \sigma_r))$ to the Open oracle where $R_1 = e(H, G_2)^{\sigma_x} \cdot e(K, G_2)^{\sigma_\delta} \cdot e(K, Y)^{-\sigma_q} \cdot e(T_1, G_2)^{\sigma_y} \cdot \left(\frac{e(G_1,G_2)}{e(T_1,Y)}\right)^{-c}$, $R_2 = G^{\sigma_x+\sigma_r} \cdot {T_2}^{-c}$, $R_3 = U^{\sigma_r} \cdot {T_3}^{-c}$, and $R_4 = V^{\sigma_r} \cdot {T_4}^{-c}$. Since we can construct the adversary who generates the involved random oracle query before querying to the Open oracle by using the adversary who does not generate the involved random oracle query before querying to the Open oracle, the condition can be assumed without loss of generality.

[Game 1]: We modify the way to generate the challenge signature in Game 1. More precisely, if there is already the pair $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), \cdot)$ in the list L when computing the value $\mathsf{H}(\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*)$, the challenger sets $\Sigma^* = \perp$. If there is not such a value, the challenger generates the challenge signature as in Game 0.

[Game 2]: We further modify the way to generate the challenge signature. In this game, the challenge signature is generated as follows:

Step 1. Choose values $r^*, q^* \in \mathbb{Z}_p$ uniformly random and compute $T_1^*, T_2^*, T_3^*, T_4^*$ as in Game 1.

Step 2. Choose $\sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^* \in \mathbb{Z}_p$ and $c^* \in \mathbb{Z}_p$ uniformly random, and compute $R_1^* = e(H, G_2)^{\sigma_x^*} \cdot e(K, G_2)^{\sigma_\delta^*} \cdot e(K, Y)^{-\sigma_q^*} \cdot e(T_1^*, G_2)^{\sigma_y^*} \cdot \left(\frac{e(G_1,G_2)}{e(T_1^*,Y)}\right)^{-c^*}$, $R_2^* = G^{\sigma_x^*+\sigma_r^*} \cdot {T_2^*}^{-c^*}$, $R_3^* = U^{\sigma_r^*} \cdot {T_3^*}^{-c^*}$, and $R_4^* = V^{\sigma_r^*} \cdot {T_4^*}^{-c^*}$.

Step 3. If a value $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), \cdot)$ is not defined in the list L, the value $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), c^*)$ is added to L and the challenge signature Σ^* is set to be $(\{T_i^*\}_{i\in[1,4]}, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*)$. On the other hand, if such a value is already defined, the challenge signature is set to be $\perp$.

[Game 3]: In this game, we modify the way to generate a proof τ in replying queries for the Open oracle. More precisely, if there is already the pair $((\mathsf{gpk}, Q, T_2, T_3, R), \cdot)$ in the list L when computing the value $\mathsf{H}(\mathsf{gpk}, Q, T_2, T_3, R)$ in the generation of τ, the challenger replies $\perp$ as the response of the query.

[Game 4]: We further modify the way to generate a proof τ in replying queries for the Open oracle. The challenger replies for a query $(m, \Sigma = (\{T_i\}_{i\in[1,4]}, c, \sigma_x, \sigma_y, \sigma_\delta, \sigma_q, \sigma_r))$ as follows. We note that steps except for Step 3 are the same as Game 3.

Step 1. If $\mathsf{GVf}(\mathsf{gpk}, m, \Sigma) = 0$, return 0.

Step 2. Compute $Q = T_2 \cdot ({T_3}^{\frac{1}{u}})^{-1}$ and find the index i such that $\mathsf{reg}[i] = Q$ in the list reg. If there is no such i, return $(0, \perp)$.

Step 3. Choose $\sigma_u \in \mathbb{Z}_p$ and $d \in \mathbb{Z}_p$ uniformly random, and set $R = (Q \cdot {T_2}^{-1})^{\sigma_u} \cdot T_3^{-d}$.

Step 4. If the value $((\mathsf{gpk}, Q, T_2, T_3, R), \cdot)$ is not defined in the list L, the value $((\mathsf{gpk}, Q, T_2, T_3, R), d)$ is added to L and reply $(i, \tau = (d, \sigma_u))$ to the adversary. On the other hand, if such a value is already defined, the opening proof τ is set to be $\perp$.

[Game 5]: We modify the way to generate a factor T_4^* in the challenge signature. More precisely, in Game 5, the challenger newly samples a random value $r_2^* \in \mathbb{Z}$ and computes $T_2^* = G^{x_{i_b}+r^*}, T_4^* = G^{r_2^*}$ by comparing Game 4 in which he computes $T_2^* = G^{x_{i_b}+r^*}, T_4^* = V^{r^*}$ where $r^* \in \mathbb{Z}$ is a uniform random value.

[Game 6]: In this game, the key to open signatures is changed from u to v. More precisely, in Game 6, the challenger sets $Q = T_2 \cdot ({T_4}^{\frac{1}{v}})^{-1}$ by comparing Game 5 in which he sets $Q = T_2 \cdot ({T_3}^{\frac{1}{u}})^{-1}$.

[Game 7]: We modify the way to generate a factor T_3^* in the challenge signature. More precisely, in Game 7, the challenger newly samples a random value $r_1^* \in \mathbb{Z}$ and computes $T_2^* = G^{x_{i_b}+r^*}, T_3^* = G^{r_1^*}$ by comparing Game 6 in which he computes $T_2^* = G^{x_{i_b}+r^*}, T_3^* = U^{r^*}$ where $r^* \in \mathbb{Z}$ is a uniform random value.

For the advantage $\mathrm{Adv}^{anon}_{\Pi_{\mathsf{FI}},\mathcal{A}}(\lambda)$, $\mathrm{Adv}^{anon}_{\Pi_{\mathsf{FI}},\mathcal{A}}(\lambda) = \left|\Pr[\mathsf{S}_0] - 1/2\right| \leq \sum_{\ell=0}^{6} \left|\Pr[\mathsf{S}_\ell] - \Pr[\mathsf{S}_{\ell+1}]\right| + \left|\Pr[\mathsf{S}_7] - 1/2\right|$ holds. Moreover, the following lemmas hold.

Lemma 4.1. *Let q_H be the number of $\mathcal{A}$'s random oracle queries. Then, it holds that $\left|\Pr[\mathtt{S}_0]-\Pr[\mathtt{S}_1]\right| \leq q_H/p$ for any PPT $\mathcal{A}$.*

Proof. We define the event $\mathtt{Bad}_\ell^{(1)}$ as follows.

$\mathtt{Bad}_\ell^{(1)}$**:** The event that there is already the pair $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), \cdot)$ in the list L when computing the value $\mathsf{H}(\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*)$ in Game ℓ.

Game 0 and Game 1 are identical unless the events $\mathtt{Bad}_0^{(1)}$ and $\mathtt{Bad}_1^{(1)}$ occur. That is, we get $\Pr[\mathtt{S}_0 \wedge \neg\mathtt{Bad}_0^{(1)}] = \Pr[\mathtt{S}_1 \wedge \neg\mathtt{Bad}_1^{(1)}]$. Therefore, it holds that $\left|\Pr[\mathtt{S}_0]-\Pr[\mathtt{S}_1]\right| = \left|\Pr[\mathtt{S}_0\wedge\mathtt{Bad}_0^{(1)}]+\Pr[\mathtt{S}_0\wedge\neg\mathtt{Bad}_0^{(1)}]-\Pr[\mathtt{S}_1 \wedge \mathtt{Bad}_1^{(1)}] - \Pr[\mathtt{S}_1 \wedge \neg\mathtt{Bad}_1^{(1)}]\right| = \left|\Pr[\mathtt{S}_0 \wedge \mathtt{Bad}_0^{(1)}] - \Pr[\mathtt{S}_1 \wedge \mathtt{Bad}_1^{(1)}]\right| \leq \Pr[\mathtt{Bad}_1^{(1)}]$.

Here, we estimate the probability $\Pr[\mathtt{Bad}_1^{(1)}]$. When the event $\mathtt{Bad}_1^{(1)}$ occurs, $\widetilde{T}_1 = T_1^*$ holds for some defined value $((\cdot, \widetilde{T}_1, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot), \cdot)$ in the list L. Since $q^* \in \mathbb{Z}_p$ is chosen uniform randomly in Game 1, $T_1^* = A_{i_b} \cdot K^{q^*} \in \mathbb{G}_1$ is also uniformly random. Also, the number of values in the list L is at least q_H. Therefore, the probability that $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), \cdot)$ is already stored in L when generating the challenge signature is at most q_H/p. That is, $\Pr[\mathtt{Bad}_1^{(1)}] \leq q_H/p$. Thus, we obtain $\left|\Pr[\mathtt{S}_0] - \Pr[\mathtt{S}_1]\right| \leq q_H/p$. □

Lemma 4.2. *It holds that $\Pr[\mathtt{S}_1] = \Pr[\mathtt{S}_2]$ for any PPT $\mathcal{A}$.*

Proof. For Game 2, we introduce new values $\rho_x^*, \rho_y^*, \rho_\delta^*, \rho_q^*, \rho_r^* \in \mathbb{Z}_p$, and set $\rho_x^* = \sigma_x^* - x_{i_b} \cdot c^*$, $\rho_y^* = \sigma_y^* - y_{i_b} \cdot c^*$, $\rho_\delta^* = \sigma_\delta^* - \delta^* \cdot c^*$, $\rho_q^* = \sigma_q^* - q^* \cdot c^*$, and $\rho_r^* = \sigma_r^* - r^* \cdot c^*$. Then, the following equations hold:

$$\begin{aligned} R_1^* &= e(H, G_2)^{\sigma_x^*} \cdot e(K, G_2)^{\sigma_\delta^*} \cdot e(K, Y)^{-\sigma_q^*} \cdot e(T_1^*, G_2)^{\sigma_y^*} \cdot \Big(\frac{e(G_1, G_2)}{e(T_1^*, Y)}\Big)^{-c^*} \\ &= e(H, G_2)^{\rho_x^*} \cdot e(K, G_2)^{\rho_\delta^*} \cdot e(K, Y)^{-\rho_q^*} \cdot e({T_1}^*, G_2)^{\rho_y^*}, \end{aligned}$$

$$R_2^* = G^{\sigma_x^*+\sigma_r^*} \cdot (T_2^*)^{-c^*} = G^{\rho_x^*+\rho_r^*},\ R_3^* = U^{\sigma_r^*} \cdot (T_3^*)^{-c^*} = U^{\rho_r^*},\ R_4^* = V^{\sigma_r^*} \cdot (T_4^*)^{-c^*} = V^{\rho_r^*}.$$

Moreover, it holds that $\sigma_x^* = x_{i_b} \cdot c^* + \rho_x^*$, $\sigma_y^* = y_{i_b} \cdot c^* + \rho_y^*$, $\sigma_\delta^* = \delta^* \cdot c^* + \rho_\delta^*$, $\sigma_q^* = q^* \cdot c^* + \rho_q^*$, and $\sigma_r^* = r^* \cdot c^* + \rho_r^*$. Furthermore, $\rho_x^*, \rho_y^*, \rho_\delta^*, \rho_q^*, \rho_r^* \in \mathbb{Z}_p$ are uniformly random since $\sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^* \in \mathbb{Z}_p$ are chosen uniform randomly. Therefore, Game 2 is identical to Game 1. That is, $\Pr[\mathtt{S}_1] = \Pr[\mathtt{S}_2]$. □

Lemma 4.3. *Let q_H and q_{open} be the number of $\mathcal{A}$'s random oracle queries and opening queries, respectively. Then, it holds that $\left|\Pr[\mathtt{S}_2] - \Pr[\mathtt{S}_3]\right| \leq q_H \cdot q_{\mathsf{open}}/p$ for any PPT $\mathcal{A}$.*

Proof. We define the event $\mathtt{Bad}_\ell^{(2)}$ as follows.

$\mathtt{Bad}_\ell^{(2)}$**:** The event that there is already the pair $((\mathsf{gpk}, Q, T_2, T_3, R), \cdot)$ in the list L when computing the value $\mathsf{H}(\mathsf{gpk}, Q, T_2, T_3, R)$ during the generation of an opening proof τ in Game ℓ.

Game 2 and Game 3 are identical unless the events $\mathtt{Bad}_2^{(2)}$ and $\mathtt{Bad}_3^{(2)}$ occur. Therefore, we get $\left|\Pr[\mathtt{S}_2] - \Pr[\mathtt{S}_3]\right| \leq \Pr[\mathtt{Bad}_3^{(2)}]$ same as in Lemma 4.1.

Here, we estimate the probability $\Pr[\mathtt{Bad}_3^{(2)}]$. When the event $\mathtt{Bad}_3^{(2)}$ occurs, $\widetilde{R} = R$ holds for the some defined value $((\cdot, \cdot, \cdot, \cdot, \widetilde{R}), \cdot)$ in the list L. Since $\rho_u \in \mathbb{Z}_p$ is chosen uniform randomly in Game 3, $R = (Q \cdot {T_2}^{-1})^{\rho_u} \in \mathbb{G}$ is also uniformly random. Also, the number of values in the list L is at least q_H. Therefore, the probability that $((\mathsf{gpk}, Q, T_2, T_3, R), \cdot)$ is already stored in L when generating an opening proof is at most q_H/p. By the union bound, $\Pr[\mathtt{Bad}_3^{(2)}] \leq q_H \cdot q_{\mathsf{open}}/p$ holds. Thus, we obtain $\left|\Pr[\mathtt{S}_2] - \Pr[\mathtt{S}_3]\right| \leq q_H \cdot q_{\mathsf{open}}/p$. □

Lemma 4.4. *It holds that $\Pr[\mathtt{S}_3] = \Pr[\mathtt{S}_4]$ for any PPT $\mathcal{A}$.*

Proof. For Game 4, we introduce new values ρ_u, and sets $\rho_u = \sigma_u - u \cdot d$. Then, $R_1 = (Q \cdot {T_2}^{-1})^{\sigma_u} \cdot T_3^{-d} = (Q \cdot {T_2}^{-1})^{\rho_u}$ and $\sigma_u = u \cdot d + \rho_u$ hold. Moreover, $\rho_u \in \mathbb{Z}_p$ is uniformly random since $\sigma_u \in \mathbb{Z}_p$ is chosen uniform randomly. Therefore, Game 4 is identical to Game 3. That is, $\Pr[\mathtt{S}_3] = \Pr[\mathtt{S}_4]$. □

Lemma 4.5. *There exists a* PPT *algorithm $\mathcal{B}_1$ such that $\left|\Pr[\mathtt{S}_4]-\Pr[\mathtt{S}_5]\right| = \mathrm{Adv}_{\mathcal{B}_1}^{DDH}(\lambda)$ for any PPT $\mathcal{A}$.*

Proof. Let $\mathcal{B}_1$ be an adversary that tries to solve the DDH problem. First, $\mathcal{B}_1$ receives the DDH tuple $G, V, R, W \in \mathbb{G}$. Let $V = G^v$, and $R = G^r$. The element W is G^{vr} or a random value in $\mathbb{G}$. Next, $\mathcal{B}_1$ generates the instance of the anonymity game. Here for G and V, he uses the ones in the DDH tuple. Other elements are generated by following the GKg algorithm. Let $\mathsf{gpk} = (G_1, G_2, G, \mathsf{H}, H, K, Y, U, V)$, $\mathsf{ik} = w$, and $\mathsf{ok} = (u, v)$, $\mathcal{B}_1$ sends $(\mathsf{gpk}, \mathsf{ik})$ to the adversary $\mathcal{A}$. We note that $\mathcal{B}_1$ does not know the discrete logarithm v of the value V. Although v is the part of the opening key ok, the key that is used for opening in Game 4 and Game 5 is $u = \log_G U$. Therefore, $\mathcal{B}_1$ possesses all keys which are needed to reply oracle queries, and can simulate the replies of all queries. Especially, $\mathcal{B}_1$ generates the challenge signature as follows:

1. Choose $q^* \in \mathbb{Z}_p$ uniform randomly and compute $(T_1^*, T_2^*, T_3^*, T_4^*) = (A_{i_b} \cdot K^{q^*}, Q_{i_b} \cdot R, R^u, W)$ where R and W are the part of the DDH tuple.
2. Choose $\sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^* \in \mathbb{Z}_p$ and $c^* \in \mathbb{Z}_p$ uniform randomly, and computes $R_1^* = e(H, G_2)^{\sigma_x^*} \cdot e(K, G_2)^{\sigma_\delta^*} \cdot e(K, Y)^{-\sigma_q^*} \cdot e(T_1^*, G_2)^{\sigma_y^*} \cdot \left(\frac{e(G_1, G_2)}{e(T_1^*, Y)}\right)^{-c^*}$, $R_2^* = G^{\sigma_x^* + \sigma_r^*} \cdot T_2^{*-c^*}$, $R_3^* = U^{\sigma_r^*} \cdot T_3^{*-c^*}$, and $R_4^* = V^{\sigma_r^*} \cdot T_4^{*-c^*}$.
3. If the value $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), \cdot)$ is not defined in the list L, the value $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), c^*)$ is added to L and the challenge signature Σ^* is set to be $(\{T_i^*\}_{i\in[1,4]}, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*)$. On the other hand, if such a value is already defined, the challenge signature is set to be $\perp$.

Finally, when $\mathcal{A}$ terminates with $\widetilde{b} \in \{0, 1\}$, $\mathcal{B}_1$ outputs 1 if $b = \widetilde{b}$. Otherwise he outputs 0.

If the DDH tuple that $\mathcal{B}_1$ obtains is $(G, V, R, W) = (G, G^v, G^r, G^{vr})$, it holds that $(T_1^*, T_2^*, T_3^*, T_4^*) = (A_{i_b} \cdot K^{q^*}, Q_{i_b} \cdot G^r, G^{ur}, G^{vr}) = (A_{i_b} \cdot K^{q^*}, G^{x_{i_b}+r}, U^r, V^r)$. Then, $\mathcal{B}_1$ perfectly simulates Game 4 for $\mathcal{A}$. On the other hand, if the element W is a random value in $\mathbb{G}$, it holds that $(T_1^*, T_2^*, T_3^*, T_4^*) = (A_{i_b} \cdot K^{q^*}, G^{x_{i_b}+r}, U^r, W)$. Then, $\mathcal{B}_1$ perfectly simulates Game 5 for $\mathcal{A}$. Therefore, it holds that $\mathrm{Adv}_{\mathcal{B}_1}^{DDH}(\lambda) = |\Pr[1 \leftarrow \mathcal{B}_1(G, G^v, G^r, G^{vr})] - \Pr[1 \leftarrow \mathcal{B}_1(G, G^v, G^r, W)]| = |\Pr[b = \widetilde{b} \text{ in Game 4}] - \Pr[b = \widetilde{b} \text{ in Game 5}]| = \left|\Pr[\mathtt{S}_4] - \Pr[\mathtt{S}_5]\right|$. □

Lemma 4.6. *If the adversary does not generate a related query, it holds that* $\left|\Pr[\mathtt{S}_5] - \Pr[\mathtt{S}_6]\right| \leq 1/p$ *for any PPT* $\mathcal{A}$.

Proof. We define the event $\mathtt{Bad}_\ell^{(3)}$ as follows.

$\mathtt{Bad}_\ell^{(3)}$**:** The event that the adversary $\mathcal{A}$ sends the opening query $(m, \Sigma = (\{T_i\}_{i\in[1,4]}, c, \sigma_x, \sigma_y, \sigma_\delta, \sigma_q, \sigma_r))$ such that $\mathsf{GVf}(\mathsf{gpk}, m, \Sigma) = 1$ and $\log_U T_3 \neq \log_V T_4$ in Game ℓ.

Game 5 and Game 6 are identical unless the events $\mathtt{Bad}_5^{(3)}$ and $\mathtt{Bad}_6^{(3)}$ occur. Therefore, we get $\left|\Pr[\mathtt{S}_5] - \Pr[\mathtt{S}_6]\right| \leq \Pr[\mathtt{Bad}_6^{(3)}]$ same as in Lemma 4.1. Moreover, we define the event $\overline{\mathtt{Bad}}_6^{(3)}$ as follows.

$\overline{\mathtt{Bad}}_6^{(3)}$**:** The event that in Game 6, the adversary $\mathcal{A}$ sends the random oracle query $(\mathsf{gpk}, \{T_i\}_{i\in[1,4]}, \{R_i\}_{i\in[1,4]}, m)$ such that $\log_U T_3 \neq \log_V T_4$ and $(\{T_i\}_{i\in[1,4]}, \{R_i\}_{i\in[1,4]}, m) \neq (\{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*)$, and there exists σ_r such that

$$\begin{pmatrix} \log_U R_3 \\ \log_V R_4 \end{pmatrix} = \begin{pmatrix} 1 & -\log_U T_3 \\ 1 & -\log_V T_4 \end{pmatrix} \begin{pmatrix} \sigma_r \\ \widetilde{c} \end{pmatrix} \tag{1}$$

where $\widetilde{c}$ is the reply of the query $(\mathsf{gpk}, \{T_i\}_{i\in[1,4]}, \{R_i\}_{i\in[1,4]}, m)$.

When $\log_U T_3 \neq \log_V T_4$ holds, it holds that

$$\left|\begin{pmatrix} 1 & -\log_U T_3 \\ 1 & -\log_V T_4 \end{pmatrix}\right| = \left|\log_U T_3 - \log_V T_4\right| \neq 0.$$

Therefore, the simultaneous equation (1) has the unique solution $(\sigma_r, \widetilde{c})$. Since it holds that $(\{T_i\}_{i\in[1,4]}, \{R_i\}_{i\in[1,4]}, m) \neq (\{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*)$, $\widetilde{c}$ is chosen uniform randomly for $(\{T_i\}_{i\in[1,4]}, \{R_i\}_{i\in[1,4]},$

m). Thus, the probability that there exists σ_r such that the equation (1) holds for $\widetilde{c}$ is $1/p$. That is, $\Pr[\overline{\mathtt{Bad}}_6^{(3)}] = 1/p$.

In the following, we prove $\left|\Pr[\mathtt{S}_5] - \Pr[\mathtt{S}_6]\right| \leq 1/p$ by showing $\mathtt{Bad}_6^{(3)} \subseteq \overline{\mathtt{Bad}}_6^{(3)}$. We consider that the event $\mathtt{Bad}_6^{(3)}$ happens, that is, the situation that $\mathcal{A}$ sends the opening query $(m, \Sigma = (\{T_i\}_{i\in[1,4]}, c, \sigma_x, \sigma_y, \sigma_\delta, \sigma_q, \sigma_r))$ such that $\mathsf{GVf}(\mathsf{gpk}, m, \Sigma) = 1$ and $\log_U T_3 \neq \log_V T_4$. Since we assume that the adversary $\mathcal{A}$ does not generate related queries, it holds that $(\{T_i\}_{i\in[1,4]}, \{R_i\}_{i\in[1,4]}, m) \neq (\{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*)$. Also from the condition which is made in Game 0, the random oracle query $X = (\mathsf{gpk}, \{T_i\}_{i\in[1,4]}, \{R_i\}_{i\in[1,4]}, m)$ is generated before (m, Σ) is queried to the Open oracle where $R_1 = e(H, G_2)^{\sigma_x} \cdot e(K, G_2)^{\sigma_\delta} \cdot e(K, Y)^{-\sigma_q} \cdot e(T_1, G_2)^{\sigma_y} \cdot \left(\frac{e(G_1,G_2)}{e(T_1,Y)}\right)^{-c}$, $R_2 = G^{\sigma_x+\sigma_r} \cdot {T_2}^{-c}$, $R_3 = U^{\sigma_r} \cdot {T_3}^{-c}$, and $R_4 = V^{\sigma_r} \cdot {T_4}^{-c}$. Let $\widetilde{c}$ be the reply of X. Since $c = \widetilde{c}$ holds when $\mathsf{GVf}(\mathsf{gpk}, m, \Sigma) = 1$, it holds that $R_3 = U^{\sigma_r} \cdot {T_3}^{-\widetilde{c}}$ and $R_4 = V^{\sigma_r} \cdot {T_4}^{-\widetilde{c}}$. For the two equations, we consider the discrete logarithm by considering the base as U and V, and then the simultaneous equation

$$\begin{pmatrix} \log_U R_3 \\ \log_V R_4 \end{pmatrix} = \begin{pmatrix} 1 & -\log_U T_3 \\ 1 & -\log_V T_4 \end{pmatrix} \begin{pmatrix} \sigma_r \\ \widetilde{c} \end{pmatrix}$$

holds. Therefore, the query X satisfies two conditions of the event $\overline{\mathtt{Bad}}_6^{(3)}$, and there exists σ_r which satisfies the equation (1) for the reply $\widetilde{c}$. Thus, $\mathtt{Bad}_6^{(3)} \subseteq \overline{\mathtt{Bad}}_6^{(3)}$ holds and we obtain $\left|\Pr[\mathtt{S}_5] - \Pr[\mathtt{S}_6]\right| \leq \Pr[\mathtt{Bad}_6^{(3)}] \leq \Pr[\overline{\mathtt{Bad}}_6^{(3)}] = 1/p$. □

Lemma 4.7. *There exists a* PPT *algorithm* $\mathcal{B}_2$ *such that* $\left|\Pr[\mathtt{S}_6] - \Pr[\mathtt{S}_7]\right| = \mathrm{Adv}_{\mathcal{B}_2}^{DDH}(\lambda)$ *for any PPT* $\mathcal{A}$.

Proof. Let $\mathcal{B}_2$ be an adversary that tries to solve the DDH problem. First, $\mathcal{B}_2$ receives the DDH tuple $G, U, R, W \in \mathbb{G}$. Let $U = G^u$ and $R = G^r$. The element W is G^{ur} or a random value in $\mathbb{G}$. Next, $\mathcal{B}_2$ generates the instance of the anonymity game. Here for G and U, he uses the ones in the DDH tuple. Other elements are generated by following the GKg algorithm. Let $\mathsf{gpk} = (G_1, G_2, G, \mathsf{H}, H, K, Y, U, V)$, $\mathsf{ik} = w$, and $\mathsf{ok} = (u, v)$, $\mathcal{B}_2$ sends $(\mathsf{gpk}, \mathsf{ik})$ to the adversary $\mathcal{A}$. We note that $\mathcal{B}_2$ does not know the discrete logarithm u of the value U. Although u is the part of the opening key ok, the key that is used for opening in Game 6 and Game 7 is $v = \log_G V$. Therefore, $\mathcal{B}_2$ possesses all keys which are needed to reply oracle queries, and can simulate the replies of all queries. Especially, $\mathcal{B}_2$ generates the challenge signature as follows:

1. Choose $q^*, r_2^* \in \mathbb{Z}_p$ uniform randomly and compute $(T_1^*, T_2^*, T_3^*, T_4^*) = (A_{i_b} \cdot K^{q^*}, Q_{i_b} \cdot R, W, G^{r_2^*})$ where R and W are the part of the DDH tuple.
2. Choose $\sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^* \in \mathbb{Z}_p$ and $c^* \in \mathbb{Z}_p$ uniform randomly, and computes $R_1^* = e(H, G_2)^{\sigma_x^*} \cdot e(K, G_2)^{\sigma_\delta^*} \cdot e(K, Y)^{-\sigma_q^*} \cdot e(T_1^*, G_2)^{\sigma_y^*} \cdot \left(\frac{e(G_1,G_2)}{e(T_1^*,Y)}\right)^{-c^*}$, $R_2^* = G^{\sigma_x^*+\sigma_r^*} \cdot {T_2^*}^{-c^*}$, $R_3^* = U^{\sigma_r^*} \cdot {T_3^*}^{-c^*}$, and $R_4^* = V^{\sigma_r^*} \cdot {T_4^*}^{-c^*}$.
3. If the value $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), \cdot)$ is not defined in the list L, the value $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), c^*)$ is added to L and the challenge signature Σ^* is set to be $(\{T_i^*\}_{i\in[1,4]}, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*)$. On the other hand, if such a value is already defined, the challenge signature is set to be $\perp$.

Finally, when $\mathcal{A}$ terminates with $\widetilde{b} \in \{0, 1\}$, $\mathcal{B}_2$ outputs 1 if $b = \widetilde{b}$. Otherwise he outputs 0.

If the DDH tuple that $\mathcal{B}_2$ obtains is $(G, U, R, W) = (G, G^u, G^r, G^{ur})$, it holds that $(T_1^*, T_2^*, T_3^*, T_4^*) = (A_{i_b} \cdot K^{q^*}, Q_{i_b} \cdot G^r, G^{ur}, G^{r_2^*})$. Then, $\mathcal{B}_2$ perfectly simulates Game 6 for $\mathcal{A}$. On the other hand, if the element W is a random value in $\mathbb{G}$, it holds that $(T_1^*, T_2^*, T_3^*, T_4^*) = (A_{i_b} \cdot K^{q^*}, Q_{i_b} \cdot G^r, W, G^{r_2^*})$. Then, $\mathcal{B}_2$ perfectly simulates Game 7 for $\mathcal{A}$. Therefore, it holds that $\mathrm{Adv}_{\mathcal{B}_2}^{DDH}(\lambda) = |\Pr[1 \leftarrow \mathcal{B}_2(G, G^u, G^r, G^{ur})] - \Pr[1 \leftarrow \mathcal{B}_2(G, G^u, G^r, W)]| = |\Pr[b = \widetilde{b} \text{ in Game 6}] - \Pr[b = \widetilde{b} \text{ in Game 7}]| = \left|\Pr[\mathtt{S}_6] - \Pr[\mathtt{S}_7]\right|$. □

For random values $q^*, r^*, r_1^*, r_2^* \in \mathbb{Z}_p$, the challenge signature in Game 7 is denoted by $\Sigma^* = (\{T_i^*\}_{i\in[1,4]}, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*) = (A_{i_b} \cdot K^{q^*}, Q_{i_b} \cdot G^{r^*}, U^{r_1^*}, V^{r_2^*}, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*)$. Therefore, the

choice of the challenge bit b and the distribution of the challenge signature Σ^* are independent. Thus, we can say that $\Pr[\mathtt{S}_7] = 1/2$. From this fact and Lemma 4.1 to Lemma 4.7, we get

$$\begin{aligned}\mathrm{Adv}^{anon}_{\Pi_{\mathsf{FI}},\mathcal{A}}(\lambda) &\leq \sum_{\ell=0}^{6}\left|\Pr[\mathtt{S}_\ell] - \Pr[\mathtt{S}_{\ell+1}]\right| + \left|\Pr[\mathtt{S}_7] - 1/2\right| \\ &\leq \mathrm{Adv}^{DDH}_{\mathcal{B}_1}(\lambda) + \mathrm{Adv}^{DDH}_{\mathcal{B}_2}(\lambda) + \frac{q_H(1+q_{\mathsf{open}})+1}{p}.\end{aligned}$$

Since q_H and q_{open} are polynomial in λ and p is exponential in λ, we see that $(q_H(1+q_{\mathsf{open}})+1)/p$ is negligible in λ. Therefore, if the adversary does not generate related queries, Mechanism 6 satisfies anonymity under the DDH assumption in the random oracle model. □

4.2 Analysis of Related Queries

From the result of the previous section, we see that the only way to break the anonymity of Mechanism 6 is generating a related query. Therefore in this section, we analyze all cases of a related query and find out the cases in which the adversary might generate it.

Let m^* and $\Sigma^* = (\{T_i^*\}_{i\in[1,4]}, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*)$ be the challenge message and the challenge signature, respectively. Let $(\widetilde{m}, \widetilde{\Sigma} = (\{\widetilde{T}_i\}_{i\in[1,4]}, \widetilde{c}, \widetilde{\sigma}_x, \widetilde{\sigma}_y, \widetilde{\sigma}_\delta, \widetilde{\sigma}_q, \widetilde{\sigma}_r))$ be a related query. From the definition of a related query it holds that $(\{\widetilde{T}_i\}_{i\in[1,4]}, \{\widetilde{R}_i\}_{i\in[1,4]}, \widetilde{m}) = (\{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*)$. Moreover, since $(\widetilde{m}, \widetilde{\Sigma}) \neq (m^*, \Sigma^*)$ holds, it is required that $(\widetilde{\sigma}_x, \widetilde{\sigma}_y, \widetilde{\sigma}_\delta, \widetilde{\sigma}_q, \widetilde{\sigma}_r) \neq (\sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*)$. That is,

$$\widetilde{\sigma}_x \neq \sigma_x^* \ \vee\ \widetilde{\sigma}_y \neq \sigma_y^* \ \vee\ \widetilde{\sigma}_\delta \neq \sigma_\delta^* \ \vee\ \widetilde{\sigma}_q \neq \sigma_q^* \ \vee\ \widetilde{\sigma}_r \neq \sigma_r^*$$

holds. Thus, we have 31 (= {the first part is changed or not} × {the second partis changed or not} × $\cdots$ × {the last part is changed or not} − {any parts are notchanged} $= 2^5 - 1$) cases of a related query.

Although there are many cases, we can narrow down to seven cases. From the equation $\widetilde{R}_3 = R_3^*$, it holds that $\widetilde{R}_3 = R_3^* \Leftrightarrow U^{\widetilde{\sigma}_r} \cdot T_3^{-\widetilde{c}} = U^{\sigma_r^*} \cdot (T_3^*)^{-c^*} \Leftrightarrow U^{\widetilde{\sigma}_r} \cdot (T_3^*)^{-c^*} = U^{\sigma_r^*} \cdot (T_3^*)^{-c^*} \Leftrightarrow U^{\widetilde{\sigma}_r} = U^{\sigma_r^*} \Leftrightarrow u\widetilde{\sigma}_r = u\sigma_r^*$. Since $u \in \mathbb{Z}_p^*$, we get $\widetilde{\sigma}_r = \sigma_r^*$. In a similar way, we get $\widetilde{\sigma}_x = \sigma_x^*$ from the equation $\widetilde{R}_2 = R_2^*$. That is, it ultimately holds that

$$\widetilde{\sigma}_y \neq \sigma_y^* \ \vee\ \widetilde{\sigma}_\delta \neq \sigma_\delta^* \ \vee\ \widetilde{\sigma}_q \neq \sigma_q^*.$$

Thus, we can narrow down to seven ($=2^3 - 1$) cases of a related query described in Table 1. Here, we classify these cases into the following types:

(a) $\widetilde{\sigma}_y \neq \sigma_y^*$ ($\widetilde{\sigma}_\delta$ and $\widetilde{\sigma}_q$ are arbitrary), (b) $\widetilde{\sigma}_y = \sigma_y^* \ \wedge\ \widetilde{\sigma}_\delta \neq \sigma_\delta^* \ \wedge\ \widetilde{\sigma}_q = \sigma_q^*$,
(c) $\widetilde{\sigma}_y = \sigma_y^* \ \wedge\ \widetilde{\sigma}_\delta = \sigma_\delta^* \ \wedge\ \widetilde{\sigma}_q \neq \sigma_q^*$, ($\star$) $\widetilde{\sigma}_y = \sigma_y^* \ \wedge\ \widetilde{\sigma}_\delta \neq \sigma_\delta^* \ \wedge\ \widetilde{\sigma}_q \neq \sigma_q^*$.

Then, we analyze each type. Specifically, the query described in Section 3 as an attack for Mechanism 6 is in Type ($\star$).

$\widetilde{\sigma}_y \stackrel{?}{=} \sigma_y^*$	$\widetilde{\sigma}_\delta \stackrel{?}{=} \sigma_\delta^*$	$\widetilde{\sigma}_q \stackrel{?}{=} \sigma_q^*$	**Type**
No	Yes	Yes	(a)
No	Yes	No	(a)
No	No	Yes	(a)
No	No	No	(a)
Yes	No	Yes	(b)
Yes	Yes	No	(c)
Yes	No	No	($\star$)

Table 1: Possible Cases of Related Queries

Now, we examine the related queries in Type (a), (b), and (c). In fact, the adversary can generate these types of queries with only negligible probability. In the following, we explain the intuition of this fact.

Let $\mathcal{A}$ be the adversary who attacks the anonymity of Mechanism 6. We note that for any related query, it holds that

$$e(K,G_2)^{\widetilde{\sigma}_\delta}\cdot e(K,Y)^{-\widetilde{\sigma}_q}\cdot e(T_1^*,G_2)^{\widetilde{\sigma}_y}=e(K,G_2)^{\sigma_\delta^*}\cdot e(K,Y)^{-\sigma_q^*}\cdot e(T_1^*,G_2)^{\sigma_y^*} \tag{2}$$

since the equation $\widetilde{R}_1 = R_1^*$ holds. From this equation, we can get the following observations on the related queries in Type (a), (b), and (c).

Type (a): We consider the situation that $\mathcal{A}$ generates a related query $(m^*, \Sigma = (\{T_i^*\}_{i\in[1,4]}, c^*, \sigma_x^*, \widetilde{\sigma}_y, \widetilde{\sigma}_\delta, \widetilde{\sigma}_q, \sigma_r^*))$ in Type (a). That is, $\widetilde{\sigma}_y \neq \sigma_y^*$ holds (here, we say nothing whether $\widetilde{\sigma}_\delta \neq \sigma_\delta^*$ and $\widetilde{\sigma}_q \neq \sigma_q^*$). Let $T_1^* = {G_1}^t$, $K = {G_1}^k$, and $H = {G_1}^h$. From Equation (2), it holds that

$$\begin{aligned}
&e(K,G_2)^{\widetilde{\sigma}_\delta}\cdot e(K,Y)^{-\widetilde{\sigma}_q}\cdot e(T_1^*,G_2)^{\widetilde{\sigma}_y}=e(K,G_2)^{\sigma_\delta^*}\cdot e(K,Y)^{-\sigma_q^*}\cdot e(T_1^*,G_2)^{\sigma_y^*}\\
&\quad\Leftrightarrow e({G_1}^k,G_2)^{\widetilde{\sigma}_\delta}\cdot e({G_1}^k,{G_2}^w)^{-\widetilde{\sigma}_q}\cdot e({G_1}^t,G_2)^{\widetilde{\sigma}_y}=e({G_1}^k,G_2)^{\sigma_\delta^*}\cdot e({G_1}^k,{G_2}^w)^{-\sigma_q^*}\cdot e({G_1}^t,G_2)^{\sigma_y^*}\\
&\quad\Leftrightarrow e(G_1,G_2)^{k\widetilde{\sigma}_\delta-kw\widetilde{\sigma}_q+t\widetilde{\sigma}_y}=e(G_1,G_2)^{k\sigma_\delta^*-kw\sigma_q^*+t\sigma_y^*}\\
&\quad\Leftrightarrow k\widetilde{\sigma}_\delta-kw\widetilde{\sigma}_q+t\widetilde{\sigma}_y=k\sigma_\delta^*-kw\sigma_q^*+t\sigma_y^*\\
&\quad\Leftrightarrow t=k\frac{w\Delta\sigma_q-\Delta\sigma_\delta}{\Delta\sigma_y}\qquad\left(\because\widetilde{\sigma}_y\neq\sigma_y^*\right)
\end{aligned}$$

where $\Delta\sigma_\delta = \widetilde{\sigma}_\delta - \sigma_\delta^*$, $\Delta\sigma_q = \widetilde{\sigma}_q - \sigma_q^*$, and $\Delta\sigma_y = \widetilde{\sigma}_y - \sigma_y^*$. Moreover, since $T_1^* = A_{i_b}\cdot K^{q^*} = \left(\frac{G_1}{H^{x_{i_b}}\cdot K^{z_{i_b}}}\right)^{\frac{1}{w+y_{i_b}}}\cdot K^{q^*} = \left(\frac{G_1}{{G_1}^{hx_{i_b}}\cdot {G_1}^{kz_{i_b}}}\right)^{\frac{1}{w+y_{i_b}}}\cdot {G_1}^{kq^*}$ holds, it holds that

$$t=\log_{G_1}T_1^*=\frac{1}{w+y_{i_b}}\left(1-hx_{i_b}-kz_{i_b}\right)+kq^*.$$

From these two equations, we get

$$k\frac{w\Delta\sigma_q-\Delta\sigma_\delta}{\Delta\sigma_y}=\frac{1}{w+y_{i_b}}\left(1-hx_{i_b}-kz_{i_b}\right)+kq^*. \tag{3}$$

From a viewpoint of the challenger who executes the anonymity game with $\mathcal{A}$, the challenger knows the values w and $(y_{i_b}, x_{i_b}, z_{i_b})$ since he generates the issuing key and all signing keys of honest users by himself. Also, q^* is chosen by the challenger. Moreover, the challenger can compute $\Delta\sigma_\delta = \widetilde{\sigma}_\delta - \sigma_\delta^*$, $\Delta\sigma_q = \widetilde{\sigma}_q - \sigma_q^*$, and $\Delta\sigma_y = \widetilde{\sigma}_y - \sigma_y^*$ from the values $\widetilde{\sigma}_\delta$, $\widetilde{\sigma}_q$, and $\widetilde{\sigma}_y$ which are the part of the related query, and the values σ_δ^*, σ_q^*, and σ_y^* which are the part of the challenge signature. The challenger does not know the discrete logarithm of K in usual since the value K is randomly chosen from $\mathbb{G}_1$ in the GKg algorithm. However, if the challenger chooses $k \in \mathbb{Z}_p$ uniform randomly and sets $K = {G_1}^k$, he can know the discrete logarithm k. Now, the challenger knows all values in Equation (3) except for h. This means that the challenger can compute the discrete logarithm h of $H \in \mathbb{G}_1$ from the values he knows. Thus, when $\mathcal{A}$ generates a related query in Type (a), the challenger can solve the DL problem in $\mathbb{G}_1$. That is, if the DL assumption holds in $\mathbb{G}_1$, the probability that $\mathcal{A}$ generates a related query in Type (a) is negligible.

Type (b): Let $K = {G_1}^k$. When the conditions $\widetilde{\sigma}_y = \sigma_y^*$ and $\widetilde{\sigma}_q = \sigma_q^*$ are put in Equation (2), we get $e(K,G_2)^{\widetilde{\sigma}_\delta} = e(K,G_2)^{\sigma_\delta^*} \Leftrightarrow e(G_1,G_2)^{k\widetilde{\sigma}_\delta} = e(G_1,G_2)^{k\sigma_\delta^*} \Leftrightarrow k\widetilde{\sigma}_\delta = k\sigma_\delta^*$. If $k \neq 0$, $\widetilde{\sigma}_\delta = \sigma_\delta^*$ holds. However, since this contradicts $\widetilde{\sigma}_\delta \neq \sigma_\delta^*$ that is the condition of Type (b), a related query in Type (b) does not exist if $k \neq 0$. On the other hand, the probability that $k = 0$ holds is $1/p$ since $K \in \mathbb{G}_1$ is chosen uniform randomly. Therefore, the probability that $\mathcal{A}$ generates a related query in Type (b) is at most $1/p$ which is negligible.

Type (c): Let $K = {G_1}^k$. When the conditions $\widetilde{\sigma}_y = \sigma_y^*$ and $\widetilde{\sigma}_\delta = \sigma_\delta^*$ are put in Equation (2), we get $e(K,Y)^{-\widetilde{\sigma}_q} = e(K,Y)^{-\sigma_q^*} \Leftrightarrow e(G_1,G_2)^{-kw\widetilde{\sigma}_q} = e(G_1,G_2)^{-kw\sigma_q^*} \Leftrightarrow kw\widetilde{\sigma}_q = kw\sigma_q^*$. If $k \neq 0$ and $w \neq 0$, $\widetilde{\sigma}_q = \sigma_q^*$ holds. However, since this contradicts $\widetilde{\sigma}_q \neq \sigma_q^*$ that is the condition of Type (c), a related query

in Type (c) does not exist if $k \neq 0$ and $w \neq 0$. On the other hand, the probability that $k = 0$ or $w = 0$ satisfies $\Pr[k = 0 \vee w = 0] \leq \Pr[k = 0] + \Pr[w = 0] = 2/p$ since $K \in \mathbb{G}_1$ and $w \in \mathbb{Z}_p$ are chosen uniform randomly. Therefore, the probability that $\mathcal{A}$ generates a related query in Type (c) is at most $2/p$ which is negligible.
Therefore, we see that the probability that $\mathcal{A}$ generates the related queries in Type (a), (b), and (c) is negligible if the DL assumption holds in $\mathbb{G}_1$.

On the other hand, we cannot rule out the possibility that the adversary generates a related query in Type ($\star$) since our attack is in this type. Now, we further analyze this type of query. This type of query satisfies $\widetilde{\sigma}_y = \sigma^*_y$. When this equation is put in Equation (2), we get

$$\begin{aligned} & e(K, G_2)^{\widetilde{\sigma}_\delta} \cdot e(K, Y)^{-\widetilde{\sigma}_q} = e(K, G_2)^{\sigma^*_\delta} \cdot e(K, Y)^{-\sigma^*_q} \\ & \Leftrightarrow e(G_1, G_2)^{k\widetilde{\sigma}_\delta} \cdot e(G_1, G_2)^{-kw\widetilde{\sigma}_q} = e(G_1, G_2)^{k\sigma^*_\delta} \cdot e(G_1, G_2)^{-kw\sigma^*_q} \\ & \Leftrightarrow k(\widetilde{\sigma}_\delta - w\widetilde{\sigma}_q) = k(\sigma^*_\delta - w\sigma^*_q). \end{aligned}$$

Since the probability that $k = 0$ holds is $1/p$, it holds that $k \neq 0$ with high probability. If $k \neq 0$, we get $\widetilde{\sigma}_\delta - w\widetilde{\sigma}_q = \sigma^*_\delta - w\sigma^*_q \Leftrightarrow w = (\widetilde{\sigma}_\delta - \sigma^*_\delta)/(\widetilde{\sigma}_q - \sigma^*_q)$. That is, the issuing key w can be computed from the values σ^*_δ and σ^*_q in the challenge signature Σ^* and the values $\widetilde{\sigma}_\delta$ and $\widetilde{\sigma}_q$ in the related query. Therefore, this indicates that the adversary who can generate a related query in Type ($\star$) knows the issuing key.

From the above observations, we see that only a related query in Type ($\star$) might be generated by the adversary. Furthermore, the adversary generating this type of query knows the issuing key. Therefore, the minimum condition of breaking the anonymity of Mechanism 6 seems to be that the adversary knows the issuing key. Thus, we can expect that *if the adversary does not possess the issuing key, Mechanism 6 satisfies anonymity.*

4.3 The Security of Mechanism 6

In this section, we formally prove the expectation given in the previous section. Concretely, we introduce a new security definition of anonymity called "weak anonymity", where the adversary is not allowed to corrupt the issuer. Then, we prove that Mechanism 6 satisfies this security notion.

Now, we define a new security notion called weak anonymity. We firstly define some oracles for the adversary who cannot corrupt the issuer. The definitions of these oracles are followed by Bellare et al. [11]. The SndToI oracle is an interactive oracle. Also, HU and CU are the set of honest users and corrupted users, respectively.

$\mathsf{AddU}(\cdot)$: The add-user oracle takes as input a user identity i, and runs UKg and $\mathsf{Join/Issue}$ protocol to add an honest user i to the group. The oracle returns upk_i and adds i to HU.

$\mathsf{SndToI}(\cdot, \cdot)$: The send-to-issuer oracle takes as input a user identity i and a initial message M_{int}, and interacts with the adversary who corrupts the user i by running $\mathsf{Issue}(\mathsf{gpk}, \mathsf{upk}_i, \mathsf{ik})$. The user i needs to be in the set CU. If $i \notin \mathsf{CU}$, the oracle outputs $\perp$.

$\mathsf{RReg}(\cdot)$: The read-registration-table oracle takes as input i, and returns $\mathsf{reg}[i]$.

Then, we give the definition of weak anonymity by using the above oracles. Intuitively, weak anonymity ensures that the adversary who corrupts all users but not the issuer cannot extract the signer's information from a signature. Formally, it is defined as follows.

Definition 4.1 (Weak Anonymity)**.** *Let $\mathcal{A}$ be an adversary for weak anonymity. We define the experiment $\mathrm{Exp}^{w\text{-}anon}_{\Pi_{\mathrm{GS}},\mathcal{A}}(\lambda)$ as follows.*

$$\begin{aligned} \mathrm{Exp}^{w\text{-}anon}_{\Pi_{\mathrm{GS}},\mathcal{A}}(\lambda): \ & b \leftarrow \{0,1\};\ (\mathsf{gpk}, \mathsf{ik}, \mathsf{ok}) \leftarrow \mathsf{GKg}(1^k);\ \mathsf{CU} \leftarrow \emptyset;\ \mathsf{HU} \leftarrow \emptyset \\ & \widetilde{b} \leftarrow \mathcal{A}^{\mathsf{AddU}(\cdot),\mathsf{CrptU}(\cdot,\cdot),\mathsf{SndToI}(\cdot,\cdot),\mathsf{USK}(\cdot),\mathsf{RReg}(\cdot),\mathsf{Ch}(b,\cdot,\cdot,\cdot),\mathsf{Open}(\cdot,\cdot)}(\mathsf{gpk}) \\ & \textit{Return } 1 \textit{ if } \widetilde{b} = b, \textit{ otherwise return } 0 \end{aligned}$$

We say that Π_{GS} satisfies weak anonymity if the advantage

$$\mathrm{Adv}^{w\text{-}anon}_{\Pi_{\mathrm{GS}},\mathcal{A}} := \left| \Pr[\mathrm{Exp}^{w\text{-}anon}_{\Pi_{\mathrm{GS}},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right|$$

is negligible for any PPT adversary $\mathcal{A}$.

Mechanism 6 satisfies weak anonymity as shown in Theorem 4.2. This theorem implies that *Mechanism 6 is still secure under the condition that the issuer does not join the attack.* Such a condition is reasonable if a single authority plays roles of both the opener and the issuer.

We note that most of the proof is the same as that of the anonymity under the restricted condition (given in Section 4.1) since anonymity in Definition 2.6 implies weak anonymity. However, since it is not assumed that the adversary does not generate a related query in the proof of the weak anonymity, we cannot straightforwardly prove the part corresponding with Game 5 $\approx$ Game 6 in the proof of the anonymity.

In the proof of the weak anonymity, we rule out the possibility that the adversary generates a related query by the computational assumptions. As we observe in Section 4.2, the adversary cannot generate related queries in Type (a), (b), and (c) under the DL assumption. Also in the proof, we prove that the adversary who does not possess the issuing key cannot generate related queries in Type ($\star$) under the q-SDH assumption. This part is the most difficult in this proof since the reduction algorithm needs to deal with generating user signing keys without the issuing key. To overcome this problem, we apply the rewinding technique as in the forking lemma [27] in our security proof.

Theorem 4.2. *Mechanism 6 satisfies weak anonymity under the DL assumption in the group $\mathbb{G}_1$, the DDH assumption in the group $\mathbb{G}$, and the q-SDH assumption in the groups $(\mathbb{G}_1, \mathbb{G}_2)$ in the random oracle model.*

Proof. Let $\mathcal{A}$ be an adversary that attacks the weak anonymity of Mechanism 6. We consider the following sequence of games. Let b be the challenge bit, m^* be the challenge message, i_0, i_1 be the challenge users, and $\Sigma^* = (T_1^*, T_2^*, T_3^*, T_4^*, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*)$ be the challenge signature. Let S_ℓ denote the event that $\mathcal{A}$ succeeds in guessing the challenge bit b in Game ℓ. Also, we specify the random tape of $\mathcal{A}$ in the proof when we use the rewinding technique.

[Game 0]: This is the experiment $\mathrm{Exp}^{w\text{-}anon}_{\Pi_{\mathsf{Fl}},\mathcal{A}}(\lambda)$ itself. As in Game 0 of Theorem 4.1, The challenger manages an inout/output pair of the random oracle in the list L, and we assume that the adversary generates the involved random oracle query before he queries to the Open oracle.

[Game 1 - Game 5]: The modification of each game is the same as that of the game in Theorem 4.1.

[Game 6]: We change the way to generate $H, K \in \mathbb{G}_1$ in the group public key gpk. More precisely, in Game 6, the challenger samples $h, k \in \mathbb{Z}_p$ uniform randomly and sets $H \leftarrow {G_1}^h, K \leftarrow {G_1}^k$ by comparing Game 5 in which H, K are chosen uniform randomly in $\mathbb{G}_1$.

[Game 7]: In Game 7, if the adversary generates the opening query $(m^*, (T_1^*, T_2^*, T_3^*, T_4^*, c^*, \sigma_x^*, \sigma_y, \sigma_\delta, \sigma_q, \sigma_r^*))$ such that $\sigma_y \neq \sigma_y^*$, the challenger returns $\perp$ where σ_y and σ_q are arbitrary.

[Game 8]: In Game 8, if the adversary generates the opening query $(m^*, (T_1^*, T_2^*, T_3^*, T_4^*, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta, \sigma_q^*, \sigma_r^*))$ such that $\sigma_\delta \neq \sigma_\delta^*$, the challenger returns $\perp$.

[Game 9]: In Game 9, if the adversary generates the opening query $(m^*, (T_1^*, T_2^*, T_3^*, T_4^*, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q, \sigma_r^*))$ such that $\sigma_q \neq \sigma_q^*$, the challenger returns $\perp$.

[Game 10]: We modify the way to reply queries for the SndToI oracle. More precisely, the challenger replies the ℓ-th query $(i, (R_1, R_2))$ for the SndToI oracle as follows. Let N be a constant number.

Step 1 (Practice Phase). Execute other N anonymity games with the adversaries $\mathcal{A}_j$ who are the same as the original $\mathcal{A}$ in parallel where $1 \leq j \leq N$. Specifically, for $1 \leq j \leq N$, perform the following operations.

1. Sample $\widehat{c}_i^{(j)} \xleftarrow{r} \mathbb{Z}_p$.
2. Execute $\mathcal{A}_j(\mathsf{gpk}; \rho)$ where ρ is the random tape of the original $\mathcal{A}$. Then, the challenger makes exactly the same replies as those for the original $\mathcal{A}$ until the ℓ-th query $(i, (R_1, R_2))$ is generated. We note that the query is also the same as that of the original $\mathcal{A}$ since the challenger makes the same replies those for the original $\mathcal{A}$ until the ℓ-th query is generated.
3. Send $\widehat{c}_i^{(j)}$ to $\mathcal{A}_j$ as the first reply of the ℓ-th query $(i, (R_1, R_2))$, and obtain $(\widehat{\sigma}_{x_i}^{(j)}, \widehat{\sigma}_{z_i'}^{(j)})$.

Step 2 (First Reply in the Original Game with $\mathcal{A}$). If $i \notin \mathsf{CU}$, return $\perp$. If $i \in \mathsf{CU}$, sample $c_i \xleftarrow{r} \mathbb{Z}_p$ and return c_i as the first reply of the send-to-issuer query $(i, (R_1, R_2))$. Then, obtain $(\sigma_{x_i}, \sigma_{z_i'})$ from $\mathcal{A}$.

Step 3 (Second Reply in the Original Game with $\mathcal{A}$). If $(\sigma_{x_i}, \sigma_{z'_i})$ is invalid, return $\perp$ as the second reply. If $(\sigma_{x_i}, \sigma_{z'_i})$ is valid, find the index $j^* \in [1, N]$ in the replies from $\mathcal{A}_j$ in Step 1 such that $(\widehat{\sigma}_{x_i}^{(j^*)}, \widehat{\sigma}_{z'_i}^{(j^*)})$ is valid and $c_i \neq \widehat{c}_i^{(j^*)}$. If there is no such index j^*, return $\perp$. On the other hand if there exists such index j^*, compute the second reply as follows.

1. Compute $\Delta\sigma_{x_i} \leftarrow \widehat{\sigma}_{x_i}^{(j^*)} - \sigma_{x_i}$, $\Delta\sigma_{z'_i} \leftarrow \widehat{\sigma}_{z'_i}^{(j^*)} - \sigma_{z'_i}$, and $\Delta c_i \leftarrow \widehat{c}_i^{(j^*)} - c_i$, and set $\widetilde{x}_i \leftarrow \Delta\sigma_{x_i}/\Delta c_i$ and $\widetilde{z}'_i \leftarrow \Delta\sigma_{z'_i}/\Delta c_i$.
2. Sample $y_i, z''_i \in \mathbb{Z}_p$ uniform randomly and compute $C_i \leftarrow G_1^{\frac{1}{w+y_i}}$.
3. Compute $A_i \leftarrow C_i^{1-h\widetilde{x}_i - k(\widetilde{z}'_i + z''_i)}$. Then, set $\mathsf{cert}_i \leftarrow (A_i, y_i, z''_i)$ and reply cert_i as the second reply. Register $\mathsf{reg}[i] \leftarrow Q_i$.

[Game 11]: We modify the way to compute the element A_i in the simulation of the AddU oracle. In Game 11, A_i is computed as follows. The challenger chooses a random value y_i and sets $C_i \leftarrow G_1^{\frac{1}{w+y_i}}$ where w is the issuing key. Then, he also samples a random value z''_i and sets $A_i \leftarrow C_i^{1-hx_i-k(z'_i+z''_i)}$ where $\mathsf{usk}_i = (x_i, z'_i)$, $h = \log_{G_1} H$, and $k = \log_{G_1} K$.

[Game 12]: In Game 12, if the adversary generates the opening query $(m^*, (T_1^*, T_2^*, T_3^*, T_4^*, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta, \sigma_q, \sigma_r^*))$ such that $\sigma_y = \sigma_y^*$, $\sigma_\delta \neq \sigma_\delta^*$, and $\sigma_q \neq \sigma_q^*$, the challenger returns $\perp$.

[Game 13, Game 14]: The modification of each game is the same as that of Game 6 and Game 7 in Theorem 4.1, respectively.

For the advantage $\mathrm{Adv}^{w\text{-}anon}_{\Pi_{\mathsf{FI}},\mathcal{A}}(\lambda)$, $\mathrm{Adv}^{w\text{-}anon}_{\Pi_{\mathsf{FI}},\mathcal{A}}(\lambda) = \left|\Pr[\mathtt{S}_0] - 1/2\right| \leq \sum_{\ell=0}^{13} \left|\Pr[\mathtt{S}_\ell] - \Pr[\mathtt{S}_{\ell+1}]\right| + \left|\Pr[\mathtt{S}_{14}] - 1/2\right|$ holds. Moreover, the following lemmas hold.

Lemma 4.8. *Let q_H be the number of $\mathcal{A}$'s random oracle queries. Then, it holds that $\left|\Pr[\mathtt{S}_0] - \Pr[\mathtt{S}_1]\right| \leq q_H/p$ for any PPT $\mathcal{A}$.*

Lemma 4.9. *It holds that $\Pr[\mathtt{S}_1] = \Pr[\mathtt{S}_2]$ for any PPT $\mathcal{A}$.*

Lemma 4.10. *Let q_H and q_{open} be the number of $\mathcal{A}$'s random oracle queries and opening queries, respectively. Then, it holds that $\left|\Pr[\mathtt{S}_2] - \Pr[\mathtt{S}_3]\right| \leq q_H \cdot q_{\mathsf{open}}/p$ for any PPT $\mathcal{A}$.*

Lemma 4.11. *It holds that $\Pr[\mathtt{S}_3] = \Pr[\mathtt{S}_4]$ for any PPT $\mathcal{A}$.*

Lemma 4.12. *There exists a* PPT *algorithm $\mathcal{B}_1$ such that $\left|\Pr[\mathtt{S}_4] - \Pr[\mathtt{S}_5]\right| = \mathrm{Adv}^{DDH}_{\mathcal{B}_1}(\lambda)$*

Lemma 4.8 to 4.12 can be proved as in the case of the anonymity (given in Section 4.1) since the modification of each game is the same as that of the game in Theorem 4.1. Therefore, we omit these proofs.

Lemma 4.13. *It holds that $\Pr[\mathtt{S}_5] = \Pr[\mathtt{S}_6]$ for any PPT $\mathcal{A}$.*

Proof. The difference between Game 5 and Game 6 is the way to generate the values $H, K \in \mathbb{G}_1$. More precisely, H, K are chosen from $\mathbb{G}_1$ uniform randomly in Game 5. On the other hand in Game 6, the challenger chooses $h, k \in \mathbb{Z}_p$ uniform randomly and sets $H \leftarrow {G_1}^h, K \leftarrow {G_1}^k$. However, since the distribution of H, K is uniform in $\mathbb{G}_1$ in each game, Game 5 and Game 6 are identical. Therefore, it holds that $\Pr[\mathtt{S}_5] = \Pr[\mathtt{S}_6]$. □

Lemma 4.14. *There exists a* PPT *algorithm $\mathcal{B}_2$ such that $\left|\Pr[\mathtt{S}_6] - \Pr[\mathtt{S}_7]\right| \leq \mathrm{Adv}^{DL}_{\mathcal{B}_2}(\lambda)$ for any PPT $\mathcal{A}$.*

Proof. We define the event $\mathtt{Bad}_\ell^{(\mathsf{a})}$ as follows.

$\mathtt{Bad}_\ell^{(\mathsf{a})}$**:** The event that the adversary $\mathcal{A}$ generates the related query in Type (a) to the Open oracle in Game ℓ.

Game 6 and Game 7 are identical unless the events $\mathtt{Bad}_6^{(\mathsf{a})}$ and $\mathtt{Bad}_7^{(\mathsf{a})}$ occur. Therefore, we get $\left|\Pr[\mathtt{S}_6] - \Pr[\mathtt{S}_7]\right| \leq \Pr[\mathtt{Bad}_7^{(\mathsf{a})}]$ same as in Lemma 4.1.

In the following, we construct the algorithm $\mathcal{B}_2$ who tries to solve the DL problem in $\mathbb{G}_1$ by using $\mathcal{A}$ and estimate the probability $\Pr[\mathtt{Bad}_7^{(\mathsf{a})}]$. First, $\mathcal{B}_2$ receives the DL tuple $G_1, H \in \mathbb{G}_1$ and $G_2 \in \mathbb{G}_2$. Now, $\mathcal{B}_2$'s goal is to compute the value $\log_{G_1} H$. Next, $\mathcal{B}_2$ generates the instance of the weak anonymity game. Here for $G_1 \in \mathbb{G}_1, G_2 \in \mathbb{G}_2$ and $H \in \mathbb{G}_1$, he uses the ones in the instance of the DL problem. Also, $\mathcal{B}_2$ samples $k \in \mathbb{Z}_p$ uniform randomly and sets $K = {G_1}^k \in \mathbb{G}_1$. Other elements are generated by following the GKg algorithm. Let $\mathsf{gpk} = (G_1, G_2, G, \mathsf{H}, H, K, Y, U, V)$, $\mathsf{ik} = w$, and $\mathsf{ok} = (u, v)$, $\mathcal{B}_2$ sends gpk to the adversary $\mathcal{A}$. Since possessing all keys which are needed to reply oracle queries, $\mathcal{B}_2$ can simulate the replies of all queries. Especially, $\mathcal{B}_2$ generates the challenge signature as follows:

1. Choose $q^*, r^*, r_2^* \in \mathbb{Z}_p$ uniform randomly and compute $(T_1^*, T_2^*, T_3^*, T_4^*) = (A_{i_b} \cdot K^{q^*}, Q_{i_b} \cdot G^{r^*}, U^{r^*}, G^{r_2^*})$.

2. Choose $\sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^* \in \mathbb{Z}_p$ and $c^* \in \mathbb{Z}_p$ uniform randomly, and computes values $R_1^* = e(H, G_2)^{\sigma_x^*} \cdot e(K, G_2)^{\sigma_\delta^*} \cdot e(K, Y)^{-\sigma_q^*} \cdot e(T_1^*, G_2)^{\sigma_y^*} \cdot \left(\frac{e(G_1, G_2)}{e(T_1^*, Y)}\right)^{-c^*}$, $R_2^* = G^{\sigma_x^* + \sigma_r^*} \cdot {T_2^*}^{-c^*}$, $R_3^* = U^{\sigma_r^*} \cdot {T_3^*}^{-c^*}$, and $R_4^* = V^{\sigma_r^*} \cdot {T_4^*}^{-c^*}$.

3. If the value $((\mathsf{gpk}, T_1^*, T_2^*, T_3^*, T_4^*, R_1^*, R_2^*, R_3^*, R_4^*, m^*), \cdot)$ is not defined in the list L, the value $((\mathsf{gpk}, T_1^*, T_2^*, T_3^*, T_4^*, R_1^*, R_2^*, R_3^*, R_4^*, m^*), c^*)$ is added to L and the challenge signature Σ^* is set to be $(T_1^*, T_2^*, T_3^*, T_4^*, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*)$. On the other hand, if such a value is already defined, the challenge signature is set to be $\perp$.

Finally, $\mathcal{A}$ terminates with $\widetilde{b} \in \{0, 1\}$.

When $\mathcal{A}$ generated the related query in Type (a) to the Open oracle, there is the Open query $(m^*, \Sigma = (T_1^*, T_2^*, T_3^*, T_4^*, c^*, \sigma_x^*, \sigma_y, \sigma_\delta, \sigma_q, \sigma_r^*))$ such that $\sigma_y \neq \sigma_y^*$ and $\mathsf{GVf}(\mathsf{gpk}, m^*, \Sigma) = 1$. For this query, $\mathcal{B}_2$ computes

$$h = \frac{1}{x_{i_b}}\Big(1 - kz_{i_b} + (w + y_{i_b})(kq^* + k\frac{\Delta\sigma_\delta - w\Delta\sigma_q}{\Delta\sigma_y})\Big)$$

where $\Delta\sigma_\delta = \sigma_\delta - \sigma_\delta^*$, $\Delta\sigma_q = \sigma_q - \sigma_q^*$, and $\Delta\sigma_y = \sigma_y - \sigma_y^*$. Then, he outputs h as the solution of the DL problem. Since i_b is an honest user, that is, $i_b \in \mathsf{HU}$ holds by the condition of the challenge query, $\mathcal{B}_2$ knows the i_b's signing key $\mathsf{gsk}_{i_b} = (A_{i_b}, y_{i_b}, z_{i_b}, x_{i_b}, Q_{i_b})$ and can compute the above h.

Now, we show the value h is the discrete logarithm of H in the following. Let $T_1^* = {G_1}^t$. Since a related query satisfies Equation (2), it holds that

$$\begin{aligned}
&e(K, G_2)^{\sigma_\delta} \cdot e(K, Y)^{-\sigma_q} \cdot e(T_1^*, G_2)^{\sigma_y} = e(K, G_2)^{\sigma_\delta^*} \cdot e(K, Y)^{-\sigma_q^*} \cdot e(T_1^*, G_2)^{\sigma_y^*} \\
&\Leftrightarrow e({G_1}^k, G_2)^{\sigma_\delta} \cdot e({G_1}^k, {G_2}^w)^{-\sigma_q} \cdot e({G_1}^t, G_2)^{\sigma_y} = e({G_1}^k, G_2)^{\sigma_\delta^*} \cdot e({G_1}^k, {G_2}^w)^{-\sigma_q^*} \cdot e({G_1}^t, G_2)^{\sigma_y^*} \\
&\Leftrightarrow e(G_1, G_2)^{k\sigma_\delta - kw\sigma_q + t\sigma_y} = e(G_1, G_2)^{k\sigma_\delta^* - kw\sigma_q^* + t\sigma_y^*} \\
&\Leftrightarrow k\sigma_\delta - kw\sigma_q + t\sigma_y = k\sigma_\delta^* - kw\sigma_q^* + t\sigma_y^* \\
&\Leftrightarrow t = k\frac{w\Delta\sigma_q - \Delta\sigma_\delta}{\Delta\sigma_y}.
\end{aligned}$$

Therefore, we get

$$\begin{aligned}
{G_1}^h &= {G_1}^{\frac{1}{x_{i_b}}\left(1 - kz_{i_b} + (w + y_{i_b})(kq^* + k\frac{\Delta\sigma_\delta - w\Delta\sigma_q}{\Delta\sigma_y})\right)} \\
&= {G_1}^{\frac{1}{x_{i_b}}\left(1 - kz_{i_b} + (w + y_{i_b})(kq^* - t)\right)} \\
&= \left(G_1 \cdot K^{-z_{i_b}} \cdot (K^{q^*} \cdot (T_1^*)^{-1})^{(w + y_{i_b})}\right)^{\frac{1}{x_{i_b}}} \\
&= \left(\frac{G_1 \cdot K^{q^*(w + y_{i_b})}}{K^{z_{i_b}} \cdot (A_{i_b} \cdot K^{q^*})^{(w + y_{i_b})}}\right)^{\frac{1}{x_{i_b}}} \quad (\because T_1^* = A_{i_b} \cdot K^{q^*}) \\
&= \left(\frac{G_1}{K^{z_{i_b}} \cdot \left(\left(\frac{G_1}{H^{x_{i_b}} \cdot K^{z_{i_b}}}\right)^{\frac{1}{w + y_{i_b}}}\right)^{(w + y_{i_b})}}\right)^{\frac{1}{x_{i_b}}} \quad \left(\because A_{i_b} = \left(\frac{G_1}{H^{x_{i_b}} \cdot K^{z_{i_b}}}\right)^{\frac{1}{w + y_{i_b}}}\right)
\end{aligned}$$

$= H.$

Thus, h is the discrete logarithm of H. That is, if the event $\mathtt{Bad}_7^{(\mathsf{a})}$ occurs, $\mathcal{B}_2$ can solve the DL problem, and $\Pr[\mathtt{Bad}_7^{(\mathsf{a})}] \leq \mathrm{Adv}^{DL}_{\mathcal{B}_2}(\lambda)$ holds. Finally, we get $\left|\Pr[\mathtt{S}_6] - \Pr[\mathtt{S}_7]\right| \leq \Pr[\mathtt{Bad}_7^{(\mathsf{a})}] \leq \mathrm{Adv}^{DL}_{\mathcal{B}_2}(\lambda)$. □

Lemma 4.15. *It holds that* $\left|\Pr[\mathtt{S}_7] - \Pr[\mathtt{S}_8]\right| \leq 1/p$ *for any PPT* $\mathcal{A}$.

Proof. We define the event $\mathtt{Bad}_\ell^{(\mathsf{b})}$ as follows.

$\mathtt{Bad}_\ell^{(\mathsf{b})}$**:** The event that the adversary $\mathcal{A}$ generates the related query in Type (b) to the Open oracle in Game ℓ.

Game 7 and Game 8 are identical unless the events $\mathtt{Bad}_7^{(\mathsf{b})}$ and $\mathtt{Bad}_8^{(\mathsf{b})}$ occur. Therefore, we get $\left|\Pr[\mathtt{S}_7] - \Pr[\mathtt{S}_8]\right| \leq \Pr[\mathtt{Bad}_8^{(\mathsf{b})}]$ same as in Lemma 4.1.

We estimate the probability $\Pr[\mathtt{Bad}_8^{(\mathsf{b})}]$ in the following. Let $K = {G_1}^k$. When the conditions $\sigma_y = \sigma_y^*$ and $\sigma_q = \sigma_q^*$ are put in Equation (2), we get $e(K, G_2)^{\sigma_\delta} = e(K, G_2)^{\sigma_\delta^*} \Leftrightarrow e(G_1, G_2)^{k\sigma_\delta} = e(G_1, G_2)^{k\sigma_\delta^*} \Leftrightarrow k\sigma_\delta = k\sigma_\delta^*$. Therefore, a related query in Type (b) satisfies the equation $k\sigma_\delta = k\sigma_\delta^*$, and then $\sigma_\delta = \sigma_\delta^*$ holds if $k \neq 0$. However, this contradicts the condition of Type (b) (i.e., $\sigma_\delta \neq \sigma_\delta^*$). Thus, a related query in Type (b) does not exist if $k \neq 0$. On the other hand, the probability that $k = 0$ holds is $1/p$ since $K \in \mathbb{G}_1$ is chosen uniform randomly. Thus, the probability that $\mathcal{A}$ generates a related query in Type (b) is at most $1/p$, and $\Pr[\mathtt{Bad}_8^{(\mathsf{b})}] \leq 1/p$ holds. That is, we get $\left|\Pr[\mathtt{S}_7] - \Pr[\mathtt{S}_8]\right| \leq 1/p$. □

Lemma 4.16. *It holds that* $\left|\Pr[\mathtt{S}_8] - \Pr[\mathtt{S}_9]\right| \leq 2/p$ *for any PPT* $\mathcal{A}$.

Proof. We define the event $\mathtt{Bad}_\ell^{(\mathsf{c})}$ as follows.

$\mathtt{Bad}_\ell^{(\mathsf{c})}$**:** The event that the adversary $\mathcal{A}$ generates a related query in Type (c) to the Open oracle in Game ℓ.

Game 8 and Game 9 are identical unless the events $\mathtt{Bad}_8^{(\mathsf{c})}$ and $\mathtt{Bad}_9^{(\mathsf{c})}$ occur. Therefore, we get $\left|\Pr[\mathtt{S}_8] - \Pr[\mathtt{S}_9]\right| \leq \Pr[\mathtt{Bad}_9^{(\mathsf{c})}]$ same as in Lemma 4.1.

We estimate the probability $\Pr[\mathtt{Bad}_9^{(\mathsf{c})}]$ in the following. Let $K = {G_1}^k$. When the conditions $\sigma_y = \sigma_y^*$ and $\sigma_\delta = \sigma_\delta^*$ are put in Equation (2), we get $e(K, Y)^{-\sigma_q} = e(K, Y)^{-\sigma_q^*} \Leftrightarrow e(G_1, G_2)^{-kw\sigma_q} = e(G_1, G_2)^{-kw\sigma_q^*} \Leftrightarrow kw\sigma_q = kw\sigma_q^*$. Therefore, a related query in Type (c) satisfies the equation $kw\sigma_q = kw\sigma_q^*$, and then $\sigma_q = \sigma_q^*$ holds if $k \neq 0$ and $w \neq 0$. However, this contradicts the condition of Type (c) (i.e., $\sigma_q \neq \sigma_q^*$). Thus, a related query in Type (c) does not exist if $k \neq 0$ and $w \neq 0$. On the other hand, the probability that $k = 0$ or $w = 0$ hold is at most $\Pr[k = 0 \vee w = 0] \leq \Pr[k = 0] + \Pr[w = 0] = 2/p$ since $K \in \mathbb{G}_1$ and $w \in \mathbb{Z}_p$ are chosen uniform randomly. Therefore, the probability that $\mathcal{A}$ generates a related query in Type (c) is at most $2/p$, and $\Pr[\mathtt{Bad}_9^{(\mathsf{c})}] \leq 2/p$ holds. That is, we get $\left|\Pr[\mathtt{S}_8] - \Pr[\mathtt{S}_9]\right| \leq 2/p$. □

Lemma 4.17. *Let* q_{iss} *be the number of* $\mathcal{A}$*'s send-to-issuer queries. Then, it holds that* $\left|\Pr[\mathtt{S}_9] - \Pr[\mathtt{S}_{10}]\right| \leq \sum_{\ell=1}^{q_{\mathsf{iss}}} \min\{(1 - \mathsf{prob}_\ell)^N, \mathsf{prob}_\ell\}$ *for any PPT* $\mathcal{A}$.

Proof. We consider the following intermediate games $\overline{\text{Game}}\ 0, \ldots, \overline{\text{Game}}\ q_{\mathsf{iss}}$ to estimate $\left|\Pr[\mathtt{S}_9] - \Pr[\mathtt{S}_{10}]\right|$.

[$\overline{\text{Game}}\ 0$]: This game is identical to Game 9.

[$\overline{\text{Game}}\ 1$]: We modify the way to reply the first send-to-issuer query. More precisely, the challenger replies the first send-to-issuer query by rewinding $\mathcal{A}$ as denoted in Game 10. After the first, the challenger replies send-to-issuer queries by following the Issue algorithm as in Game 9.

$\vdots$

[$\overline{\text{Game}}\ \ell$]: In this game, the challenger replies the first to ℓ-th send-to-issuer queries by rewinding $\mathcal{A}$. On the other hand for the $(\ell + 1)$-th to q_{iss}-th send-to-issuer queries, he replies by following the Issue algorithm.

$\vdots$

[$\overline{\text{Game}}$ q_{iss}]: In this game, the challenger replies all send-to-issuer queries by rewinding $\mathcal{A}$. Thus, this game is identical to Game 10.

Let $\overline{\mathtt{S}}_\ell$ denote the event that $\mathcal{A}$ succeeds in guessing the challenge bit in $\overline{\text{Game}}$ ℓ. Then, the following inequality holds:

$$\left|\Pr[\mathtt{S}_9] - \Pr[\mathtt{S}_{10}]\right| = \left|\Pr[\overline{\mathtt{S}}_0] - \Pr[\overline{\mathtt{S}}_{q_{\mathsf{iss}}}]\right| \leq \sum_{\ell=1}^{q_{\mathsf{iss}}} \left|\Pr[\overline{\mathtt{S}}_{\ell-1}] - \Pr[\overline{\mathtt{S}}_\ell]\right|. \tag{4}$$

Now, we prove that $\left|\Pr[\overline{\mathtt{S}}_{\ell-1}] - \Pr[\overline{\mathtt{S}}_\ell]\right| \leq \min\{(1-\mathsf{prob}_\ell)^N, \mathsf{prob}_\ell\}$ holds for $1 \leq \ell \leq q_{\mathsf{iss}}$ where N is the number of the parallel anonymity games with $\mathcal{A}_j$. The difference between $\overline{\text{Game}}$ $\ell-1$ and $\overline{\text{Game}}$ ℓ is the way to reply the ℓ-th SndToI query. In both games, the first reply of the ℓ-th SndToI query is chosen uniform randomly. Moreover, when $\mathcal{A}$'s output $(\sigma_{x_i}, \sigma_{z'_i})$ for the first reply is invalid, the second reply of the ℓ-th SndToI query will be $\perp$ in both games. Therefore, only when $\mathcal{A}$'s output $(\sigma_{x_i}, \sigma_{z'_i})$ for the first reply is valid, the challengers in $\overline{\text{Game}}$ $\ell-1$ and $\overline{\text{Game}}$ ℓ behave differently. In the following, we consider the case that $(\sigma_{x_i}, \sigma_{z'_i})$ is valid.

In $\overline{\text{Game}}$ ℓ, when $(\sigma_{x_i}, \sigma_{z'_i})$ is valid, the second reply is decided whether there exists the valid output $(\widehat{\sigma}_{x_i}^{(j^*)}, \widehat{\sigma}_{z'_i}^{(j^*)})$ which satisfies $\widehat{c}_i^{(j^*)} \neq c_i$ in the N executions of $\mathcal{A}$. More precisely, if there exists such an output, a certificate $\mathsf{cert}_i = (A_i, y_i, z''_i)$ will be returned. On the other hand, if there is no such an output, the second reply will be $\perp$. Now, we show that in the former case, the second reply cert_i in $\overline{\text{Game}}$ ℓ is the same as that in $\overline{\text{Game}}$ $\ell-1$. First of all, y_i and z''_i are chosen uniform randomly in both games. Thus, the distribution of y_i and z''_i in $\overline{\text{Game}}$ ℓ is the same as that in $\overline{\text{Game}}$ $\ell-1$. Next, we prove the value A_i in $\overline{\text{Game}}$ ℓ is the same as that in $\overline{\text{Game}}$ $\ell-1$. Since $(\sigma_{x_i}, \sigma_{z'_i})$ and $(\widehat{\sigma}_{x_i}^{(j^*)}, \widehat{\sigma}_{z'_i}^{(j^*)})$ are valid, it holds that

$$R_2 = H^{\sigma_{x_i}} K^{\sigma_{z'_i}} / {H_i}^{c_i} \ \wedge \ R_2 = H^{\widehat{\sigma}_{x_i}^{(j^*)}} K^{\widehat{\sigma}_{z'_i}^{(j^*)}} / {H_i}^{\widehat{c}_i^{(j^*)}}.$$

Therefore, we get

$$H_i = H^{\frac{\Delta\sigma_{x_i}}{\Delta c_i}} \cdot K^{\frac{\Delta\sigma_{z'_i}}{\Delta c_i}} \tag{5}$$

where $\Delta\sigma_{x_i} = \widehat{\sigma}_{x_i}^{(j^*)} - \sigma_{x_i}$, $\Delta\sigma_{z'_i} = \widehat{\sigma}_{z'_i}^{(j^*)} - \sigma_{z'_i}$, and $\Delta c_i = \widehat{c}_i^{(j^*)} - c_i$. We note that $\Delta c_i \neq 0$ holds since $\widehat{c}_i^{(j^*)} \neq c_i$. From this equation, the value A_i in $\overline{\text{Game}}$ ℓ satisfies that

$$\begin{aligned}
A_i &= {C_i}^{1-h\widetilde{x}_i - k(\widetilde{z}'_i + z''_i)} \\
&= \left({G_1}^{\frac{1}{w+y_i}}\right)^{1-h\frac{\Delta\sigma_{x_i}}{\Delta c_i} - k\left(\frac{\Delta\sigma_{z'_i}}{\Delta c_i} + z''_i\right)} \\
&= \left({G_1}^{1-h\frac{\Delta\sigma_{x_i}}{\Delta c_i} - k\left(\frac{\Delta\sigma_{z'_i}}{\Delta c_i} + z''_i\right)}\right)^{\frac{1}{w+y_i}} \\
&= \left(\frac{G_1}{{G_i}^{h\frac{\Delta\sigma_{x_i}}{\Delta c_i}} \cdot {G_i}^{k\left(\frac{\Delta\sigma_{z'_i}}{\Delta c_i} + z''_i\right)}}\right)^{\frac{1}{w+y_i}} \\
&= \left(\frac{G_1}{H^{\frac{\Delta\sigma_{x_i}}{\Delta c_i}} \cdot K^{\frac{\Delta\sigma_{z'_i}}{\Delta c_i}} \cdot K^{z''_i}}\right)^{\frac{1}{w+y_i}} \\
&= \left(\frac{G_1}{H_i \cdot K^{z''_i}}\right)^{\frac{1}{w+y_i}}. \qquad (\because \text{Equation(5)})
\end{aligned}$$

This is the same as the value A_i in $\overline{\text{Game}}$ $\ell-1$. Therefore, in the condition that there exists the valid output $(\widehat{\sigma}_{x_i}^{(j^*)}, \widehat{\sigma}_{z'_i}^{(j^*)})$ such that $\widehat{c}_i^{(j^*)} \neq c_i$, the second reply for the ℓ-th SndToI query in $\overline{\text{Game}}$ ℓ is identical to that in $\overline{\text{Game}}$ $\ell-1$.

Now, we define the events $\mathtt{Bad}$ and $\widehat{\mathtt{Bad}}$ as follows.

$\mathtt{Bad}$: The event that there is no valid output $(\widehat{\sigma}_{x_i}^{(j^*)}, \widehat{\sigma}_{z'_i}^{(j^*)})$ which satisfies $\widehat{c}_i^{(j^*)} \neq c_i$ in the N executions of $\mathcal{A}$ in $\overline{\text{Game}}$ ℓ.

$\widehat{\mathsf{Bad}}$**:** The event that $\mathcal{A}$'s output $(\sigma_{x_i}, \sigma_{z'_i})$ for the first reply is valid in $\overline{\mathsf{Game}}$ ℓ.

In the above discussion, when $(\sigma_{x_i}, \sigma_{z'_i})$ is invalid, the replies of the ℓ-th SndToI query in $\overline{\mathsf{Game}}$ $\ell-1$ and $\overline{\mathsf{Game}}$ ℓ are identical. Also, when $(\sigma_{x_i}, \sigma_{z'_i})$ is valid and there exists the valid output $(\widehat{\sigma}_{x_i}^{(j^*)}, \widehat{\sigma}_{z'_i}^{(j^*)})$ which satisfies $\widehat{c}_i^{(j^*)} \neq c_i$ in the N executions of $\mathcal{A}_j$, the replies of the ℓ-th SndToI query in both games are identical. That is, $\overline{\mathsf{Game}}$ ℓ is identical to $\overline{\mathsf{Game}}$ $\ell-1$ unless the events Bad and $\widehat{\mathsf{Bad}}$ occur, and $\Pr[\overline{\mathsf{S}}_{\ell-1}] = \Pr[\overline{\mathsf{S}}_{\ell} \wedge (\neg\mathsf{Bad} \vee \neg\widehat{\mathsf{Bad}})]$ holds. Therefore, we get $\left|\Pr[\overline{\mathsf{S}}_{\ell-1}] - \Pr[\overline{\mathsf{S}}_{\ell}]\right| = \left|\Pr[\overline{\mathsf{S}}_{\ell-1}] - (\Pr[\overline{\mathsf{S}}_{\ell} \wedge (\neg\mathsf{Bad}\vee\neg\widehat{\mathsf{Bad}})] + \Pr[\overline{\mathsf{S}}_{\ell}\wedge(\mathsf{Bad}\wedge\widehat{\mathsf{Bad}})])\right| \leq \Pr[\overline{\mathsf{S}}_{\ell}\wedge(\mathsf{Bad}\wedge\widehat{\mathsf{Bad}})] \leq \Pr[\mathsf{Bad}\wedge\widehat{\mathsf{Bad}}]$. Since $\Pr[\mathsf{Bad}\wedge\widehat{\mathsf{Bad}}] \leq \Pr[\mathsf{Bad}]$ and $\Pr[\mathsf{Bad} \wedge \widehat{\mathsf{Bad}}] \leq \Pr[\widehat{\mathsf{Bad}}]$ hold, we get $\Pr[\mathsf{Bad} \wedge \widehat{\mathsf{Bad}}] \leq \min\{\Pr[\mathsf{Bad}], \Pr[\widehat{\mathsf{Bad}}]\}$. Since $\Pr[\mathsf{Bad}] = (1 - \mathsf{prob}_\ell)^N$ and $\Pr[\widehat{\mathsf{Bad}}] = \mathsf{prob}_\ell$ hold by the definition of the events, it holds that $\Pr[\mathsf{Bad} \wedge \widehat{\mathsf{Bad}}] \leq \min\{(1 - \mathsf{prob}_\ell)^N, \mathsf{prob}_\ell\}$. Therefore, we get $\left|\Pr[\overline{\mathsf{S}}_{\ell-1}] - \Pr[\overline{\mathsf{S}}_{\ell}]\right| \leq \min\{(1 - \mathsf{prob}_\ell)^N, \mathsf{prob}_\ell\}$.

From the above discussion and Equation (4), $\left|\Pr[\mathsf{S}_9]-\Pr[\mathsf{S}_{10}]\right| \leq \sum_{\ell=1}^{q_{\mathsf{iss}}}\left|\Pr[\overline{\mathsf{S}}_{\ell-1}]-\Pr[\overline{\mathsf{S}}_{\ell}]\right| = \sum_{\ell=1}^{q_{\mathsf{iss}}} \min\{(1-\mathsf{prob}_\ell)^N, \mathsf{prob}_\ell\}$ holds. □

Lemma 4.18. *It holds that* $\Pr[\mathsf{S}_{10}] = \Pr[\mathsf{S}_{11}]$ *for any PPT* $\mathcal{A}$*.*

Proof. The difference between Game 10 and Game 11 is the way to compute the element $\mathcal{A}_i$ in the simulation of the AddU oracle. However, the value $\mathcal{A}_i$ generated in Game 11 is identical to that in Game 10 since it holds that $A_i = {C_i}^{1-hx_i-k(z'_i+z''_i)} = ({G_1}^{\frac{1}{w+y_i}})^{1-hx_i-k(z'_i+z''_i)} = ({G_1}^{1-hx_i-k(z'_i+z''_i)})^{\frac{1}{w+y_i}} = (G_1 \cdot H^{-x_i} \cdot K^{-z'_i} \cdot K^{-z''_i})^{\frac{1}{w+y_i}} = (G_1 \cdot {H_i}^{-1} \cdot K^{-z''_i})^{\frac{1}{w+y_i}}$. That is, $\Pr[\mathsf{S}_{10}] = \Pr[\mathsf{S}_{11}]$. □

Lemma 4.19. *Let* q_{add} *and* q_{iss} *be the number of* $\mathcal{A}$*'s add-user queries and send-to-issuer queries, respectively. Then, there exists a* PPT *algorithm* $\mathcal{B}_3$ *such that* $\left|\Pr[\mathsf{S}_{11}] - \Pr[\mathsf{S}_{12}]\right| \leq \mathrm{Adv}_{\mathcal{B}_3}^{(q_{\mathsf{add}} + q_{\mathsf{iss}} + 1)\text{-}SDH}(\lambda) + 1/p$ *for any PPT* $\mathcal{A}$*.*

Proof. We define the event $\mathsf{Bad}_\ell^{(\star)}$ as follows.

$\mathsf{Bad}_\ell^{(\star)}$**:** The event that the adversary $\mathcal{A}$ sends the related query in Type $(\star)$ to the Open oracle in Game ℓ.

Game 11 and Game 12 are identical unless the events $\mathsf{Bad}_{11}^{(\star)}$ and $\mathsf{Bad}_{12}^{(\star)}$ occur. Therefore, we get $\left|\Pr[\mathsf{S}_{11}] - \Pr[\mathsf{S}_{12}]\right| \leq \Pr[\mathsf{Bad}_{12}^{(\star)}]$ same as in Lemma 4.1.

In the following, we construct the algorithm $\mathcal{B}_3$ who tries to solve the simplified $(q_{\mathsf{add}} + q_{\mathsf{iss}})$-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ by using $\mathcal{A}$ and estimate the probability $\Pr[\mathsf{Bad}_{12}^{(\star)}]$. First, $\mathcal{B}_3$ receives a tuple $(G_1, G_2, Y, \{C_i, y_i\}_{i=1}^{q_{\mathsf{add}}+q_{\mathsf{iss}}})$ as the input of the simplified $(q_{\mathsf{add}}+q_{\mathsf{iss}})$-SDH problem. Let $Y = {G_2}^w$, it holds that $C_i = {G_1}^{\frac{1}{w+y_i}}$. Next, $\mathcal{B}_3$ generates the instance of the weak anonymity game. Here for $G_1 \in \mathbb{G}_1, G_2 \in \mathbb{G}_2$ and $Y \in \mathbb{G}_2$, he uses the ones in the tuple of the simplified $(q_{\mathsf{add}} + q_{\mathsf{iss}})$-SDH problem. Also, $\mathcal{B}_3$ samples $h, k \in \mathbb{Z}_p$ uniform randomly and sets $H = {G_1}^h, K = {G_1}^k$. Other elements are generated by following the GKg algorithm. Let $\mathsf{gpk} = (G_1, G_2, G, \mathsf{H}, H, K, Y, U, V)$, $\mathsf{ik} = w$, and $\mathsf{ok} = (u, v)$, $\mathcal{B}_3$ sends gpk to the adversary $\mathcal{A}$. We note that $\mathcal{B}_3$ does not know the discrete logarithm w of the value Y. The CrptU oracle and the RReg oracle are easily simulated since they just set the user public key and retrieve the register, respectively. $\mathcal{B}_3$ simulates other oracles (AddU, USK, and SndToI) as follows.

[The AddU oracle] For an input i, the algorithm $\mathcal{B}_3$ runs the UKg algorithm and obtains $(\mathsf{upk}_i, \mathsf{usk}_i) = ((Q_i, H_i), (x_i, z'_i))$. Also, $\mathcal{B}_3$ samples a random value z''_i and sets $A_i \leftarrow {C_i}^{1-hx_i-k(z'_i+z''_i)}$. Then, he sets $\mathsf{gsk}_i \leftarrow (A_i, y_i, z_i, x_i, Q_i)$. Finally, the user public key upk_i is returned to the adversary, and i is added to HU.

[The USK oracle] For an input i, $\mathcal{B}_3$ returns $\perp$ if $i \notin \mathsf{HU}$. If $i \in \mathsf{HU}$, he returns the secret keys usk_i and gsk_i. Since i is queried to the AddU oracle if $i \in \mathsf{HU}$, $\mathcal{B}_3$ knows the secret keys of the user i.

[The SndToI oracle] For a query $(i, (R_1, R_2))$, $\mathcal{B}_3$ replies as follows:

Step 1 (Practice Phase). Execute other N games with the same adversary $\mathcal{A}$ in parallel with the game of the original $\mathcal{A}$. For $1 \leq j \leq N$, perform the following operations.

1. Sample $\widehat{c}_i^{(j)} \xleftarrow{r} \mathbb{Z}_p$.
2. Execute $\mathcal{A}(\mathsf{gpk}; \rho)$ where the challenger makes the same replies for the original $\mathcal{A}$ until the ℓ-th SndToI query is generated.

3. Send $\widehat{c}_i^{(j)}$ to $\mathcal{A}$ as the first reply of the ℓ-th SndToI query $(i,(R_1,R_2))$, and obtain $(\widehat{\sigma}_{x_i}^{(j)},\widehat{\sigma}_{z'_i}^{(j)})$. We note that the query $(i,(R_1,R_2))$ is the same as that in the game of the original $\mathcal{A}$ since the challenger makes the same replies until the ℓ-th SndToI query is generated.

Step 2 (First Reply in the Original Game with $\mathcal{A}$). If $i \notin \mathsf{CU}$, return $\perp$. If $i \in \mathsf{CU}$, sample $c_i \xleftarrow{r} \mathbb{Z}_p$ and return c_i as the first reply of the ℓ-th SndToI query $(i,(R_1,R_2))$. Then, obtain $(\sigma_{x_i},\sigma_{z'_i})$ from $\mathcal{A}$.

Step 3 (Second Reply in the Original Game with $\mathcal{A}$). If $(\sigma_{x_i},\sigma_{z'_i})$ is invalid, return $\perp$ as the second reply. If $(\sigma_{x_i},\sigma_{z'_i})$ is valid, find the index $j^* \in [1,N]$ for the replies from $\mathcal{A}$ in Step 1 such that $(\widehat{\sigma}_{x_i}^{(j^*)},\widehat{\sigma}_{z'_i}^{(j^*)})$ is valid and $c_i \neq \widehat{c}_i^{(j^*)}$. If there is no such index j^*, return $\perp$ as the second reply. On the other hand if there is such index j^*, compute the second reply as follows.

1. Compute $\Delta\sigma_{x_i} \leftarrow \widehat{\sigma}_{x_i}^{(j^*)} - \sigma_{x_i}$, $\Delta\sigma_{z'_i} \leftarrow \widehat{\sigma}_{z'_i}^{(j^*)} - \sigma_{z'_i}$, and $\Delta c_i \leftarrow \widehat{c}_i^{(j^*)} - c_i$, and set $\widetilde{x}_i \leftarrow \Delta\sigma_{x_i}/\Delta c_i$ and $\widetilde{z}'_i \leftarrow \Delta\sigma_{z'_i}/\Delta c_i$.
2. Sample $z''_i \in \mathbb{Z}_p$ uniform randomly.
3. Compute $A_i \leftarrow C_i^{1-h\widetilde{x}_i-k(\widetilde{z}'_i+z''_i)}$, and set $\mathsf{cert}_i \leftarrow (A_i,y_i,z''_i)$ where (C_i,y_i) is the part of the input of the simplified $(q_{\mathsf{add}}+q_{\mathsf{iss}})$-SDH problem. Then, reply cert_i as the second reply, and register $\mathsf{reg}[i] \leftarrow Q_i$.

Finally, $\mathcal{A}$ terminates with $\widetilde{b} \in \{0,1\}$.

When $\mathcal{A}$ generated a related query in Type $(\star)$ to the Open oracle, there is the Open query $(m^*,\Sigma = (T_1^*,T_2^*,T_3^*,T_4^*,c^*,\sigma_x^*,\sigma_y,\sigma_\delta,\sigma_q,\sigma_r^*))$ such that $\sigma_y = \sigma_y^*$, $\sigma_\delta \neq \sigma_\delta^*$, $\sigma_q \neq \sigma_q^*$, and $\mathsf{GVf}(\mathsf{gpk},m^*,\Sigma) = 1$. For this query, $\mathcal{B}_3$ computes $\widetilde{w} = (\sigma_\delta - \sigma_\delta^*)/(\sigma_q - \sigma_q^*)$. Moreover, he chooses a value $y \notin \{y_1,\ldots.y_q\}$ and computes $C = G_2^{\frac{1}{\widetilde{w}+y}}$. Then, $\mathcal{B}_3$ finally outputs (C,y).

In the following, we show (C,y) is the solution of the simplified $(q_{\mathsf{add}}+q_{\mathsf{iss}})$-SDH problem, that is, $e(C,Y\cdot G_2{}^y) = e(G_1,G_2)$ holds. Since a related query satisfies Equation (2), it holds that

$$\begin{aligned}
&e(K,G_2)^{\sigma_\delta}\cdot e(K,Y)^{-\sigma_q}\cdot e(T_1^*,G_2)^{\sigma_y} = e(K,G_2)^{\sigma_\delta^*}\cdot e(K,Y)^{-\sigma_q^*}\cdot e(T_1^*,G_2)^{\sigma_y^*}\\
&\quad\Leftrightarrow e({G_1}^k,G_2)^{\sigma_\delta}\cdot e({G_1}^k,{G_2}^w)^{-\sigma_q} = e({G_1}^k,G_2)^{\sigma_\delta^*}\cdot e({G_1}^k,{G_2}^w)^{-\sigma_q^*}\\
&\quad\Leftrightarrow e(G_1,G_2)^{k\sigma_\delta - kw\sigma_q + t\sigma_y} = e(G_1,G_2)^{k\sigma_\delta^* - kw\sigma_q^* + t\sigma_y^*}\\
&\quad\Leftrightarrow k(\sigma_\delta - w\sigma_q) = k(\sigma_\delta^* - w\sigma_q^*)\ \ (\because \sigma_y = \sigma_y^*).
\end{aligned}$$

From this, if $k \neq 0$, we get $\sigma_\delta - w\sigma_q = \sigma_\delta^* - w\sigma_q^* \Leftrightarrow w = (\sigma_\delta - \sigma_\delta^*)/(\sigma_q - \sigma_q^*)$. Therefore, $C = G_1^{\frac{1}{\widetilde{w}+y}} = G_1^{\frac{1}{w+y}}$, and then (C,y) is the solution of the simplified $(q_{\mathsf{add}}+q_{\mathsf{iss}})$-SDH problem. On the other hand, the probability that $k=0$ holds is $1/p$ since $k \in \mathbb{Z}_p$ is chosen uniform randomly. Let $\mathtt{BAD}$ be the event that $k=0$ holds in Game 12. Then, it holds that

$$\begin{aligned}
\Pr[\mathtt{Bad}_{12}^{(\star)}] &= \Pr[\mathtt{Bad}_{12}^{(\star)} \wedge \neg\mathtt{BAD}] + \Pr[\mathtt{Bad}_{12}^{(\star)} \wedge \mathtt{BAD}]\\
&\leq \Pr[\mathtt{Bad}_{12}^{(\star)} \wedge \neg\mathtt{BAD}] + \Pr[\mathtt{BAD}]\\
&\leq \mathrm{Adv}_{\mathcal{B}_3}^{sim\text{-}(q_{\mathsf{add}}+q_{\mathsf{iss}})\text{-}SDH}(\lambda) + 1/p.
\end{aligned}$$

In addition to this, by Theorem 2.1, it holds that $\mathrm{Adv}_{\mathcal{B}_3}^{sim\text{-}(q_{\mathsf{add}}+q_{\mathsf{iss}})\text{-}SDH}(\lambda) \leq \mathrm{Adv}_{\mathcal{B}_3}^{(q_{\mathsf{add}}+q_{\mathsf{iss}}+1)\text{-}SDH}(\lambda)$, and then we get $\left|\Pr[\mathtt{S}_{11}] - \Pr[\mathtt{S}_{12}]\right| \leq \Pr[\mathtt{Bad}_{12}^{(\star)}] \leq \mathrm{Adv}_{\mathcal{B}_3}^{(q_{\mathsf{add}}+q_{\mathsf{iss}}+1)\text{-}SDH}(\lambda) + 1/p$. □

Also, the following lemmas hold. Since we can show these lemmas as in the case of Theorem 4.1, we omit the proofs of the lemmas.

Lemma 4.20. *It holds that* $\left|\Pr[\mathtt{S}_{12}] - \Pr[\mathtt{S}_{13}]\right| \leq 1/p$ *for any PPT* $\mathcal{A}$.

Lemma 4.21. *There exists a* PPT *algorithm* $\mathcal{B}_4$ *such that* $\left|\Pr[\mathtt{S}_{13}] - \Pr[\mathtt{S}_{14}]\right| \leq \mathrm{Adv}_{\mathcal{B}_5}^{DDH}(\lambda)$ *for any PPT* $\mathcal{A}$.

For random values $q^*, r^*, r_1^*, r_2^* \in \mathbb{Z}_p$, the challenge signature in Game 14 is denoted by $\Sigma^* = (T_1^*, T_2^*, T_3^*, T_4^*, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*) = (A_{i_b} \cdot K^{q^*}, Q_{i_b} \cdot G^{r^*}, U^{r_1^*}, V^{r_2^*}, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*)$. Therefore, the choice of the challenge bit b and the distribution of the challenge signature Σ^* are independent. Thus, we can say that $\Pr[\mathtt{S}_{14}] = 1/2$. From this fact and Lemma 4.8 to Lemma 4.21, we get

$$\begin{aligned}
\mathrm{Adv}^{w\text{-}anon}_{\Pi_{\mathsf{FI}},\mathcal{A}}(\lambda) &= \left|\Pr[\mathtt{S}_0] - 1/2\right| \\
&\leq \sum_{\ell=0}^{13} \left|\Pr[\mathtt{S}_\ell] - \Pr[\mathtt{S}_{\ell+1}]\right| + \left|\Pr[\mathtt{S}_{14}] - 1/2\right| \\
&\leq \mathrm{Adv}^{DDH}_{\mathcal{B}_1}(\lambda) + \mathrm{Adv}^{DDH}_{\mathcal{B}_4}(\lambda) + \mathrm{Adv}^{DL}_{\mathcal{B}_2}(\lambda) + \mathrm{Adv}^{(q_{\mathsf{add}} + q_{\mathsf{iss}} + 1)\text{-}SDH}_{\mathcal{B}_3}(\lambda) \\
&\qquad + \frac{q_H(1 + q_{\mathsf{open}}) + 5}{p} + \sum_{\ell=1}^{q_{\mathsf{iss}}} \min\{(1 - \mathsf{prob}_\ell)^N, \mathsf{prob}_\ell\}.
\end{aligned}$$

Now, we prove that the last term is negligible when setting N to an appropriate value. Let $N = \lambda^{\mathsf{cnst}+1}$ for an arbitrary constant cnst. If $\mathsf{prob}_\ell \geq 1/\lambda^{\mathsf{cnst}}$, we get $(1 - \mathsf{prob}_\ell)^N \leq (1 - 1/\lambda^{\mathsf{cnst}})^{\lambda^{\mathsf{cnst}+1}} = (1 - 1/\lambda^{\mathsf{cnst}})^{\lambda^{\mathsf{cnst}} \cdot \lambda} < 1/e^\lambda$. Therefore for the sufficiently large λ, it holds that $\min\{(1 - \mathsf{prob}_\ell)^N, \mathsf{prob}_\ell\} \leq (1 - \mathsf{prob}_\ell)^N < 1/\lambda^{\mathsf{cnst}}$. Thus, when $\mathsf{prob}_\ell \geq 1/\lambda^{\mathsf{cnst}}$ holds, $\min\{(1 - \mathsf{prob}_\ell)^N, \mathsf{prob}_\ell\}$ is negligible in λ. On the other hand if $\mathsf{prob}_\ell < 1/\lambda^{\mathsf{cnst}}$, it holds that $\min\{(1 - \mathsf{prob}_\ell)^N, \mathsf{prob}_\ell\} \leq \mathsf{prob}_\ell < 1/\lambda^{\mathsf{cnst}}$. Therefore, also when $\mathsf{prob}_\ell < 1/\lambda^{\mathsf{cnst}}$ holds, $\min\{(1 - \mathsf{prob}_\ell)^N, \mathsf{prob}_\ell\}$ is negligible. Since q_{iss} is polynomial in λ, $\sum_{\ell=1}^{q_{\mathsf{iss}}} \min\{(1 - \mathsf{prob}_\ell)^N, \mathsf{prob}_\ell\}$ is negligible in λ.

Also, since q_H and q_{open} are polynomial in λ and p is exponential in λ, we get $(q_H(1 + q_{\mathsf{open}}) + 5)/p = \mathsf{negl}(\lambda)$. Therefore, Mechanism 6 satisfies weak anonymity under the DDH assumption, the DL assumption, and the q-SDH assumption in the random oracle model. □

4.4 Practical Implications

Now, we discuss the practical implications of our result. Specifically, we highlight the implications for the EPID scheme [14, 5], which is based on Mechanism 6 and is standardized as Mechanism 3 in the ISO/IEC 20008-2 [2]. In fact, the EPID scheme has a lot in common with Mechanism 6, especially, their joining protocols are almost identical.[5]

Firstly, our security analysis helps to understand the security of the EPID scheme. As we showed in Section 3, there exists an attack against the anonymity of Mechanism 6 in the BSZ model. Therefore, it is not sure that the EPID scheme is secure since its security relies on that of Mechanism 6. Specifically, there are concerns that the weakness of Mechanism 6 might be exploited to frame the EPID scheme. However, fortunately our attack does not threaten the EPID scheme for operational reasons. Concretely, since the CPA security is considered in the security model of the EPID scheme [14] (i.e., an adversary is not allowed to access the opening oracle in this model), our attack does not work. In addition, due to our security analysis of Mechanism 6, it seems that the EPID scheme is secure in the proposed security model [14]. More precisely, our result (specifically, Theorem 4.1) implies that Mechanism 6 is secure in the CPA setting since an adversary cannot generate a related query in this setting. Therefore, the EPID scheme also seems to be secure in the CPA setting.

Secondly, our result is a first step to use the EPID scheme in a more demanding situation. Even if the EPID scheme is secure in the CPA setting, there remains a possibility of potential attacks such as Bleichenbacher's attack [12]. Such attacks have been efficiently implemented (e.g., [9, 31]), especially the attacks proposed by Swami [31] is a type of CCA attacks for Intel SGX, which employs the EPID scheme. Since Intel SGX is widely used in many kinds of cryptographic systems [29, 30, 22, 8, 21, 28], it might be possible that the vulnerability of Mechanism 6 is exploited for some deployed system. Therefore, to achieve a higher security level, it is required that the EPID scheme is secure in the CCA setting. Due to our analysis of the rigorous security, we see that Mechanism 6 is CCA secure under the condition that the issuer does not join the attack. (Also, we provide a patched scheme satisfying CCA security in the next section.) Thus, it seems that the EPID scheme could also achieve CCA security if it is used under limited conditions (or it is constructed from the patched scheme instead of Mechanism 6). Although we

[5] Roughly, the values h_1, h_2, A, x, y, and f in the EPID scheme [14] correspond to the values H, K, A, y_i, z_i, and x_i in Mechanism 6 (showed in Figure 1), respectively.

need a more detailed discussion, we hope that we have provided approaches to use the EPID scheme in the CCA setting.

5 A Patched Scheme

In this section, we give a patch of Mechanism 6. As we explained before, the flaw of Mechanism 6 is that the underlying proof system does not satisfy simulation soundness. More precisely, for commitments $\{R_i\}_{i\in[1,4]}$ and a challenge value c, the elements σ_x and σ_r are uniquely determined but the other elements σ_y, σ_δ, and σ_q are redundant. By this redundancy, the adversary can re-randomize the challenge signature, and then Mechanism 6 can be broken.

To achieve that Mechanism 6 satisfies anonymity in the BSZ model, we need to remove this redundancy. A simple way to do this is to make the underlying proof system have unique responses [20, 32] (defined as "strict soundness" in the later paper). That is, for commitments $\{R_i\}_i$ and a challenge value c, there exists only one valid proof. By doing so, the adversary cannot re-randomize a signature since there is no candidate of such a signature. However, when we employ the proof system with unique responses, the resulting group signature scheme becomes inefficient. This is because many equations need to be proved/verified in such a proof system, and then the signature size and the signing/verifying costs in the group signature scheme also increase.

In the proposed patched scheme, we *reduce* the redundancy to prevent from re-randomizing the signature. Concretely, we add an equation to prove about the witness q and also fix the element σ_q. That is, the parts σ_y and σ_δ are still redundant also in the patched scheme. However, from the analysis of related queries in Section 4.2, we see that it is hard to generate related queries in such a situation. When the element σ_q is fixed, possible cases of related queries are " $\widetilde{\sigma}_y \neq \sigma_y^* \ \wedge \ \widetilde{\sigma}_\delta \neq \sigma_\delta^*$ " or " $\widetilde{\sigma}_y \neq \sigma_y^* \ \wedge \ \widetilde{\sigma}_\delta = \sigma_\delta^*$ " or " $\widetilde{\sigma}_y = \sigma_y^* \ \wedge \ \widetilde{\sigma}_\delta \neq \sigma_\delta^*$ ". In Table 1, the former two cases are in Type (a), and the later case is in Type (b). As we proved, the probability that the adversary generates the related queries in Type (a) and (b) is negligible. Therefore, the adversary cannot re-randomize a signature when the element σ_q is fixed.

The description of the patched scheme is given in Figure 2. The changed parts from Mechanism 6 are underlined. In the patched scheme, only the signing and the verification algorithms are changed whereas the other algorithms (GKg, UKg, Join/Issue, Open, and Judge) are the same as those of Mechanism 6. Concretely, to fix the value σ_q, the element $T_0 = {G_1}^q$ is added as a part of a signature, and a signer also proves this equation when generating a signature.

One concern is that the signer's information may leak by adding a new element to a signature. In Mechanism 6, the randomness $q \in \mathbb{Z}_p$ is used to mask the certificate A_i such that $T_1 = A_i \cdot K^q$. Thus, the tuple (T_0, T_1) is an ElGamal encryption of a certificate A_i. Since Type II pairing is considered, the XDH assumption holds. Thus, the ElGamal scheme is secure in $\mathbb{G}_1$, and then the additional element T_0 does not leak the signer's information.

$\mathsf{GSig}(\mathsf{gpk}, \mathsf{gsk}_i, m)$:
$r, q \xleftarrow{\$} \mathbb{Z}_p$; $\underline{T_0 \leftarrow {G_1}^q}$; $T_1 \leftarrow A_i \cdot K^q$; $T_2 \leftarrow G^{x_i + r}$; $T_3 \leftarrow U^r$; $T_4 \leftarrow V^r$; $\rho_{x_i}, \rho_{y_i}, \rho_\delta, \rho_q, \rho_r \xleftarrow{\$} \mathbb{Z}_p$
$R_1 \leftarrow e(H, G_2)^{\rho_{x_i}} \cdot e(K, G_2)^{\rho_\delta} \cdot e(K, Y)^{-\rho_q} \cdot e(T_1, G_2)^{\rho_{y_i}}$
$R_2 \leftarrow G^{\rho_{x_i} + \rho_r}$; $R_3 \leftarrow U^{\rho_r}$; $R_4 \leftarrow V^{\rho_r}$; $\underline{R_5 \leftarrow {G_1}^{\rho_q}}$
$c \leftarrow \mathsf{H}(\mathsf{gpk}, \{T_i\}_{i\in[0,4]}, \{R_i\}_{i\in[1,5]}, m)$; $\delta \leftarrow z_i - q y_i$
$\sigma_{x_i} \leftarrow x_i \cdot c + \rho_{x_i}$; $\sigma_{y_i} \leftarrow y_i \cdot c + \rho_{y_i}$; $\sigma_\delta \leftarrow \delta \cdot c + \rho_\delta$; $\sigma_q \leftarrow q \cdot c + \rho_q$; $\sigma_r \leftarrow r \cdot c + \rho_r$
Return $\Sigma = (\{T_i\}_{i\in[0,4]}, c, \sigma_{x_i}, \sigma_{y_i}, \sigma_\delta, \sigma_q, \sigma_r)$

$\mathsf{GVf}(\mathsf{gpk}, m, \Sigma)$:
$R_1' \leftarrow e(H, G_2)^{\sigma_x} \cdot e(K, G_2)^{\sigma_\delta} \cdot e(K, Y)^{-\sigma_q} \cdot e(T_1, G_2)^{\sigma_y} \cdot \left(\frac{e(G_1, G_2)}{e(T_1, Y)}\right)^{-c}$
$R_2' \leftarrow G^{\sigma_x + \sigma_r} \cdot {T_2}^{-c}$; $R_3' \leftarrow U^{\sigma_r} \cdot {T_3}^{-c}$; $R_4' \leftarrow V^{\sigma_r} \cdot {T_4}^{-c}$; $\underline{R_5' \leftarrow {G_1}^{\sigma_q} \cdot {T_0}^{-c}}$
Return 1 if $c = \mathsf{H}(\mathsf{gpk}, \{T_i\}_{i\in[0,4]}, \{R_i'\}_{i\in[1,5]}, m)$, else return 0

Figure 2: The GSig and the GVf Algorithm of the Patched Scheme

The Security of the Patched Scheme. By the above modification, the patched scheme satisfies anonymity in the BSZ model. We give only its intuition here.

A signature in the patched scheme consists of two ElGamal encryptions (specifically, one is a double encryption) and a zero-knowledge proof. Intuitively, the signer's information is hidden from the adversary by the security of the encryption schemes and the zero-knowledge property of the underlying proof system. Therefore, we can easily see that the patched scheme satisfies anonymity if the adversary does not generate a related query as in Theorem 4.1. Also, there are three cases of a related query in the patched scheme as we mentioned above, but all the cases are eliminated by the analysis in Section 4.2. Thus, the patched scheme satisfies anonymity. Formally, the following theorem holds.

Theorem 5.1. *The patched scheme satisfies anonymity under the DL assumption in the group $\mathbb{G}_1$, the XDH assumption in the group $\mathbb{G}_1$, and the DDH assumption in the group $\mathbb{G}$ in the random oracle model.*

Moreover, the patched scheme satisfies the other security requirements, that is, traceability and non-frameability [11]. Since our modification does not affect these security proofs, we can prove the traceability and non-frameability of the patched scheme in the same way as those of the original scheme. Formally, the following theorems hold.

Theorem 5.2. *The patched scheme satisfies traceability under the q-SDH assumption in the groups $(\mathbb{G}_1, \mathbb{G}_2)$ in the random oracle model.*

Theorem 5.3. *The patched scheme satisfies non-frameability under the DL assumption in the group $\mathbb{G}_1$ in the random oracle model.*

Efficiency. In the patched scheme, the signature size increases by only one element in $\mathbb{G}_1$ from Mechanism 6. More precisely, a signature in the patched scheme consists of two elements from $\mathbb{G}_1$, three elements from $\mathbb{G}$, and six elements from $\mathbb{Z}_p$. This achieves the comparable efficiency to the existing schemes [18, 19] satisfying the same security level. Specifically, a signature in the Delerablée-Pointcheval scheme [18] consists of four elements in $\mathbb{G}_1$ and five elements in $\mathbb{Z}_p$, and in the Derler-Slamanig scheme [19], a signature requires four elements in $\mathbb{G}_1$, two elements in $\mathbb{G}_2$, and three elements in $\mathbb{Z}_p$.

6 Conclusion

Firstly, we have shown an attack against the anonymity of Mechanism 6 in the BSZ model. Specifically, we have proved that the issuer can identify the signer of any signature although only the opener is allowed to trace the signer in the BSZ model.

Secondly, we have analyzed the security properties offered by Mechanism 6 and characterized the conditions under which its anonymity is preserved. Concretely, we have seen that no one can extract the signer's information from a signature except for the opener and the issuer. This fact indicates that Mechanism 6 is still secure under the condition that the issuer does not join the attack. Such a condition is reasonable if *a single authority plays roles of both the opener and the issuer*.

Finally, we have derived a simple patch for Mechanism 6 from our analysis of its security. In the patched scheme, only the signing and verification algorithms are changed, and its signature size increases by only one element in $\mathbb{G}_1$ where $\mathbb{G}_1$ is a source group in the used asymmetric bilinear group. Also, we need to introduce the XDH assumption in $\mathbb{G}_1$ to prove the anonymity of the patched scheme, but the other security requirements can be showed under the same assumptions as those of Mechanism 6.

References

[1] ISO/IEC 9796-2:2010 information technology – security techniques – digital signature schemes giving message recovery – part 2: Integer factorization based mechanisms.

[2] ISO/IEC 20008-2:2013 information technology – security techniques – anonymous digital signatures – part 2: Mechanisms using a group public key.

[3] ICAO NTWG, PKI for Machine Readable Travel Documents Offering ICC Read-Only Access, Technical Report, Version 1.1, 2004.

[4] EMV, Integrated Circuit Card Specifications for Payment Systems, Book 2, Security and Key Management, Version 4.2, 2008.

[5] Intel Enhanced Privacy ID (EPID) Security Technology, https:software.intel.com/en-us/articles/intel-enhanced-privacy-id-epid-security-technology.

[6] Intel Software Guard Extensions (Intel SGX), https://software.intel.com/en-us/sgx.

[7] NISC-PEC, December 2011, http://csrc.nist.gov/groups/ST/PEC2011/presentations2011/brickell.pdf.

[8] R. Bahmani, M. Barbosa, F. Brasser, B. Portela, A. Sadeghi, G. Scerri, and B. Warinschi. Secure multiparty computation from SGX. In *FC*, pages 477–497, 2017.

[9] R. Bardou, R. Focardi, Y. Kawamoto, L. Simionato, G. Steel, and J. Tsay. Efficient padding oracle attacks on cryptographic hardware. In *CRYPTO*, pages 608–625, 2012.

[10] M. Bellare and P. Rogaway. The exact security of digital signatures - how to sign with RSA and rabin. In *EUROCRYPT*, pages 399–416, 1996.

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

[12] D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *CRYPTO*, pages 1–12, 1998.

[13] D. Boneh and X. Boyen. Short signatures without random oracles. In *EUROCRYPT*, pages 56–73, 2004.

[14] E. Brickell and J. Li. Enhanced privacy ID from bilinear pairing for hardware authentication and attestation. In *SocialCom/PASSAT*, pages 768–775, 2010.

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

[16] J. Coron, D. Naccache, and J. P. Stern. On the security of RSA padding. In *CRYPTO*, pages 1–18, 1999.

[17] J. Coron, D. Naccache, M. Tibouchi, and R. Weinmann. Practical cryptanalysis of ISO/IEC 9796-2 and EMV signatures. In *CRYPTO*, pages 428–444, 2009.

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

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

[20] S. Faust, M. Kohlweiss, G. A. Marson, and D. Venturi. On the non-malleability of the fiat-shamir transform. In *INDOCRYPT*, pages 60–79, 2012.

[21] B. Fisch, D. Vinayagamurthy, D. Boneh, and S. Gorbunov. IRON: functional encryption using intel SGX. In *CCS*, pages 765–782, 2017.

[22] B. Fuhry, R. Bahmani, F. Brasser, F. Hahn, F. Kerschbaum, and A. Sadeghi. Hardidx: Practical and secure index with SGX. In *DBSec*, pages 386–408, 2017.

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

[24] J. Furukawa and H. Imai. An efficient group signature scheme from bilinear maps. *IEICE Transactions*, 89-A(5):1328–1338, 2006.

[25] J. Y. Hwang, S. Lee, B. Chung, H. S. Cho, and D. Nyang. Group signatures with controllable linkability for dynamic membership. *Inf. Sci.*, 222:761–778, 2013.

[26] T. Isshiki, K. Mori, K. Sako, I. Teranishi, and S. Yonezawa. Using group signatures for identity management and its implementation. In *Digital Identity Management*, pages 73–78, 2006.

[27] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.

[28] S. Sasy, S. Gorbunov, and C. W. Fletcher. Zerotrace : Oblivious memory primitives from intel SGX. In *NDSS*, 2018.

[29] F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, and M. Russinovich. VC3: trustworthy data analytics in the cloud using SGX. In *Security and Privacy*, pages 38–54, 2015.

[30] J. Seo, B. Lee, S. M. Kim, M. Shih, I. Shin, D. Han, and T. Kim. Sgx-shield: Enabling address space layout randomization for SGX programs. In *NDSS*, 2017.

[31] Y. Swami. SGX remote attestation is not sufficient. *IACR Cryptology ePrint Archive*, 2017:736, 2017.

[32] D. Unruh. Quantum proofs of knowledge. In *EUROCRYPT*, pages 135–152, 2012.