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

Ai Ishida
National Institute of Advanced Industrial Science and Technology
Tokyo, Japan
a.ishida@aist.go.jp

Yusuke Sakai
National Institute of Advanced Industrial Science and Technology
Tokyo, Japan
yusuke.sakai@aist.go.jp

Keita Emura
National Institute of Information and Communications Technology
Tokyo, Japan
k-emura@nict.go.jp

Goichiro Hanaoka
National Institute of Advanced Industrial Science and Technology
Tokyo, Japan
hanaoka-goichiro@aist.go.jp

Keisuke Tanaka
Tokyo Institute of Technology
Tokyo, Japan
keisuke@is.titech.ac.jp

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.

CCS CONCEPTS

• **Security and privacy** → **Cryptography**; Cryptanalysis and other attacks;

KEYWORDS

Group signature; Cryptanalysis; ISO/IEC 20008-2; SGX

ACM Reference Format:
Ai Ishida, Yusuke Sakai, Keita Emura, Goichiro Hanaoka, and Keisuke Tanaka. 2019. Proper Usage of the Group Signature Scheme in ISO/IEC 20008-2. In *ACM Asia Conference on Computer and Communications Security (AsiaCCS '19), July 9–12, 2019, Auckland, New Zealand.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3321705.3329824

*This work was supported by JST CREST Grant Number JPMJCR19F6, Japan.

AsiaCCS '19, July 9–12, 2019, Auckland, New Zealand

ACM ISBN 978-1-4503-6752-3/19/07…$15.00
https://doi.org/10.1145/3321705.3329824

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 in 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 [14] 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 [13] is based on the Furukawa-Imai scheme [22, 23], 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 [23]. 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] 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 [13] and the conference material [7].
[2] The model in the papers [22, 23] is slightly different from the BSZ model. However, it is easy to see that they are essentially the same.

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 remedies. Secondly, as the best remedy, 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 the roles of both the issuer and the opener*. Finally, we derive a simple patch for Mechanism 6. In the following, we provide more details of our contributions.

Attack against Mechanism 6 and Its Remedies. 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 [27], it might be possible to create a valid proof without a witness after seeing some valid proofs. We note that the proof in the original paper [23] is not correct since the authors seem to have misunderstood that the underlying proof system does not satisfy simulation soundness but only satisfies soundness. Moreover, in that paper only a sketch of a proof is given, and therefore it is impossible to identify the wrong part of the proof. However, the existence of our attack implies that at least their statement about the security is not correct.

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.

We can consider the following three remedies 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. We consider that the remedy (3) is most appropriate among the possible remedies (see Section 3 for details). Therefore in this work, we analyze the security properties offered by Mechanism 6 in order to characterize the conditions under which its anonymity is preserved.

Rigorous Security Evaluation of Mechanism 6. As a result of the analysis, we see that no one can extract information about the signer from a signature except for the opener and the issuer. This 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 the 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. 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 it is not so easy to take this approach correctly and completely.

In addition, the formal proof of 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-knowledge proof system, but 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.

Patched Scheme. Owing 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.

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 [17, 18] 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 shown 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 its remedies. 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 some type of attack in Section 4.1, and provide further analysis of this attack in Section 4.2. As a result, 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 conditions. 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 O is a function or an algorithm, A^{O} denotes that A has oracle access to 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 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 cyclic groups, written multiplicatively, 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.

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 randomness 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 randomness 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 randomness of $\mathcal{A}$.*

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

Definition 2.4 (Decisional Diffie-Hellman Assumption). *We say that the DDH assumption holds in $\mathbb{G}$ if for any PPT adversary $\mathcal{A}$, the advantage $\mathrm{Adv}^{DDH}_{\mathcal{A}}(\lambda) := |\Pr[1 \leftarrow \mathcal{A}(G, G^a, G^b, G^{ab})] - \Pr[1 \leftarrow \mathcal{A}(G, G^a, G^b, W)]|$ 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 randomness 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, 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 $(\mathsf{upk}_i, \mathsf{usk}_i)$.
- **Join/Issue:** The pair of (interactive) algorithms, which are run by a user i and the issuer, 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 about the user i in $\mathrm{reg}[i]$ and the user obtains the corresponding signing key gsk_i. We denote $\mathrm{reg} = \{\mathrm{reg}[i]\}_i$.
- **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 we are 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.

- **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.
- **SndToU$(\cdot)$:** The send-to-user oracle takes as input a user identity i. 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 the oracle 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 $\perp$.
- **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 $\perp$.

- WReg$(\cdot,\cdot)$: The write-registration-table oracle takes as input i and a value $\widehat{\text{reg}}$, and writes or modifies the contents of reg by setting $\text{reg}[i] \leftarrow \widehat{\text{reg}}$.
- 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 $\Sigma^* \leftarrow \text{GSig}(\text{gpk}, \text{gsk}_{i_b}, m^*)$ if both $i_0 \in \text{HU}$ and $i_1 \in \text{HU}$. 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.
- Open$(\cdot,\cdot)$: The opening oracle takes as input m and Σ, and returns $(i,\tau) \leftarrow \text{Open}(\text{gpk}, \text{ok}, m, \Sigma, \text{reg})$ if $(m,\Sigma) \neq (m^*, \Sigma^*)$ where m^* and Σ^* are a challenge message and challenge signature, respectively. If not, the oracle returns $\perp$.

Next, 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 about the signer from a group signature.

Definition 2.5 (Anonymity [11]). *Let $\mathcal{A}$ be an adversary for anonymity. We define the experiment $\text{Exp}^{anon}_{\Pi_{\text{GS}},\mathcal{A}}(\lambda)$ as follows.*

$$\text{Exp}^{anon}_{\Pi_{\text{GS}},\mathcal{A}}(\lambda): b \leftarrow \{0,1\};\ (\text{gpk}, \text{ik}, \text{ok}) \leftarrow \text{GKg}(1^k)$$
$$\text{CU} \leftarrow \emptyset;\ \text{HU} \leftarrow \emptyset;\ \widetilde{b} \leftarrow \mathcal{A}^{\text{Oracles}}(\text{gpk}, \text{ik})$$
$$\textit{Return } 1 \textit{ if } \widetilde{b} = b, \textit{ otherwise return } 0$$

In the above experiment, we have Oracles $= \{\text{CrptU}(\cdot,\cdot), \text{SndToU}(\cdot), \text{USK}(\cdot), \text{WReg}(\cdot,\cdot), \text{Ch}(b,\cdot,\cdot,\cdot), \text{Open}(\cdot,\cdot)\}$. *We say that Π_{GS} satisfies anonymity if the advantage*

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

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 [22, 23], 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, 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 derived 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 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 of a public key encryption scheme and outputs ID i with a proof which shows that the decryption is honestly done.

[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.

3 ATTACK AGAINST MECHANISM 6

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 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 $\sigma_{\text{set}} = \{\sigma_x, \sigma_y, \sigma_\delta, \sigma_q, \sigma_r\}$ 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 $c \leftarrow \text{H}(\text{gpk}, \{T_i\}_{i\in[1,4]}, \{R_i\}_{i\in[1,4]}, m)$ 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 derive $e(K, G_2)^{\sigma_\delta} \cdot e(K, Y)^{-\sigma_q} = e(K, G_2)^{\sigma_\delta - \sigma_q \cdot w}$.

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 $\widetilde{\sigma}_{\text{set}} = \{\sigma_x, \sigma_y, \widetilde{\sigma}_\delta, \widetilde{\sigma}_q, \sigma_r\}$ also satisfies the above 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 $\widetilde{\sigma}_{\text{set}}$ 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 leads to 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 issue a query for

$\mathsf{GKg}(1^\lambda)$:
$G_2 \xleftarrow{\$} \mathbb{G}_2;\ G \xleftarrow{\$} \mathbb{G};\ G_1 \leftarrow \Psi(G_2);\ \mathsf{H} \xleftarrow{\$} \mathcal{H};\ H, K \xleftarrow{\$} \mathbb{G}_1$
$w \xleftarrow{\$} \mathbb{Z}_p;\ u, v \xleftarrow{\$} \mathbb{Z}_p^*;\ Y \leftarrow {G_2}^w;\ U \leftarrow G^u;\ V \leftarrow G^v$
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 $(\mathsf{upk}_i, \mathsf{usk}_i) = ((Q_i, H_i), (x_i, z_i'))$

$\mathsf{GVf}(\mathsf{gpk}, m, \Sigma)$:
$R_1' \leftarrow e(H, G_2)^{\sigma_{x_i}} \cdot e(K, G_2)^{\sigma_\delta} \cdot e(K, Y)^{-\sigma_q} \cdot e(T_1, G_2)^{\sigma_{y_i}} \cdot \left(\frac{e(G_1, G_2)}{e(T_1, Y)}\right)^{-c}$
$R_2' \leftarrow G^{\sigma_{x_i} + \sigma_r} \cdot {T_2}^{-c};\ R_3' \leftarrow U^{\sigma_r} \cdot {T_3}^{-c};\ R_4' \leftarrow V^{\sigma_r} \cdot {T_4}^{-c}$
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 $\mathsf{reg}[i] = Q$
$\rho_u \xleftarrow{\$} \mathbb{Z}_p;\ R \leftarrow (T_2 \cdot Q^{-1})^{\rho_u}$
$d \leftarrow \mathsf{H}(\mathsf{gpk}, Q, T_2, T_3, R);\ \sigma_u \leftarrow u \cdot d + \rho_u;\ \tau \leftarrow (d, \sigma_u)$
Return (i, τ)

$\mathsf{Judge}(\mathsf{gpk}, \mathsf{reg}, m, \Sigma, (i, \tau))$:
Return $\perp$ if $\mathsf{GVf}(\mathsf{gpk}, m, \Sigma) = 0$
$Q \leftarrow \mathsf{reg}[i];\ R' \leftarrow (T_2 \cdot Q^{-1})^{\sigma_u} \cdot {T_3}^{-d}$
Return 1 if $d = \mathsf{H}(\mathsf{gpk}, Q, T_2, T_3, R')$, else return 0

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

$\mathsf{Join}/\mathsf{Issue}(\text{User } i\text{: } \mathsf{gpk}, \mathsf{upk}_i, \mathsf{usk}_i;\ \text{Issuer: } \mathsf{gpk}, \mathsf{upk}_i, \mathsf{ik})$:
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 $e(A_i, Y \cdot {G_2}^{y_i}) e(H^{x_i}, G_2) e(K^{z_i}, G_2) = e(G_1, G_2)$

Figure 1: Mechanism 6

the signature $\widetilde{\Sigma}$ to the opening oracle. Here, information about the signer 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.

Remedies for Our Attack. We can consider the following three remedies 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 remedy.

Remedy (1): This remedy 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 [25]) introduces a special authority called a user-revocation manager, and Mechanism 7 has a functionality called controllable linkability [24]. Therefore, at a first glance, Mechanisms 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. 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 a 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, Remedy (1) is not very appropriate because of these drawbacks.

Remedy (2): This remedy 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 when it is just an update 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. [15] in 1999,[4] but the final revised version was not published before 2002. Specifically in that 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 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, Remedy (2) is not an immediate remedy for the attack.

Remedy (3): 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

[4] Coron, Naccache and Stern [15] discovered that Scheme 1 is existentially forgeable in theory. Later, Coron, Naccache, Tibouchi, and Weinmann [16] showed a practical forgery for Scheme 1 in 2009.

Mechanism 6 is still secure to use in practice since the BSZ model considers a relatively strong level of security, e.g., a dynamic model, double authority, and CCA-anonymity. For example, since the BSZ model considers double authority, all entities except for the opener can corrupt in the anonymity game of this model. However, this seems not necessarily a real threat. Therefore, Remedy (3) seems most reasonable among the possible remedies.

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

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.

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 (Section 4.1). Secondly, we provide further analysis of the attack by classifying some cases depending on the types of the adversary's queries (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 (Section 4.3).

4.1 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 querying it to the opening 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.

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 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 adversary wins with the probability 1/2. In fact for $\ell \ne 5$, 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 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 in the random oracle model under the DDH assumption in the group $\mathbb{G}$.*

Proof. Let $\mathcal{A}$ be an adversary that attacks the anonymity of Mechanism 6 (in the following, the scheme is denoted as Π_{Fl}). 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 $\mathrm{Exp}^{anon}_{\Pi_{\mathsf{Fl}},\mathcal{A}}(\lambda)$ itself. The challenger manages an input/output pair of the random oracle in a 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 of generating the challenge signature in Game 1. If the pair $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), \cdot)$ is already 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 no such a value, the challenger generates the challenge signature as in Game 0.

[Game 2]: We further modify the way of generating 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 $R_1^* = e(H, G_2)^{\sigma_x^*} \cdot e(K, G_2)^{\sigma_\delta^*} \cdot e(K, Y)^{-\sigma_q^*}$.

$e(T_1^*, G_2)^{\sigma_y^*} \cdot \left(\frac{e(G_1,G_2)}{e(T_1^*,Y)}\right)^{-c^*}$, $R_2^* = G^{\sigma_x^*+\sigma_r^*} \cdot T_2^{*-c^*}$, $R_3^* = U^{\sigma_r^*} \cdot T_3^{*-c^*}$, 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 of generating a proof τ in replying to queries for the Open oracle. If the pair $((\mathsf{gpk}, Q, T_2, T_3, R), \cdot)$ is already 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 returns $\perp$ as the response of the query.

[Game 4]: We further modify the way to generate a proof τ in replying to queries for the Open oracle. The challenger replies to 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 the steps except for Step 3 are the same as in 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 samples a new random value $r_2^* \in \mathbb{Z}$ and computes $T_2^* = G^{x_{i_b}+r^*}, T_4^* = G^{r_2^*}$ compared to 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, the challenger sets $Q = T_2 \cdot (T_4{}^{\frac{1}{v}})^{-1}$ compared to 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, the challenger samples a new random value $r_1^* \in \mathbb{Z}$ and computes $T_2^* = G^{x_{i_b}+r^*}, T_3^* = G^{r_1^*}$ compared to 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 $\mathsf{Adv}^{anon}_{\Pi_{\mathsf{FI}},\mathcal{A}}(\lambda)$,

$$\mathsf{Adv}^{anon}_{\Pi_{\mathsf{FI}},\mathcal{A}}(\lambda) = \left|\Pr[\mathsf{S}_0] - 1/2\right| \le \sum_{\ell=0}^{6} \left|\Pr[\mathsf{S}_\ell] - \Pr[\mathsf{S}_{\ell+1}]\right| + \left|\Pr[\mathsf{S}_7] - 1/2\right|$$

holds. Moreover, the following lemmas hold. Due to space limitations, we give the proofs in Appendix A.

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

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

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 $\left|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]\right| \le q_H \cdot q_{\mathsf{open}}/p$ for any PPT $\mathcal{A}$.*

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

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

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

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

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 $\Pr[\mathsf{S}_7] = 1/2$. From this fact and Lemma 4.1 to Lemma 4.7, we get

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

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 in the random oracle model under the DDH assumption. □

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 by generating a related query. Therefore, we now analyze all cases of a related query and identify 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. The definition of a related query implies 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 part is changed or not} × ⋯ × {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 these cases down to seven. The equation $\widetilde{R}_3 = R_3^*$ implies 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) $\widetilde{\sigma}_y = \sigma_y^* \wedge \widetilde{\sigma}_\delta = \sigma_\delta^* \wedge \widetilde{\sigma}_q \neq \sigma_q^*$, and ($\star$) $\widetilde{\sigma}_y = \sigma_y^* \wedge \widetilde{\sigma}_\delta \neq \sigma_\delta^* \wedge \widetilde{\sigma}_q \neq \sigma_q^*$. 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: Type 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

$$\begin{aligned} &e(K, G_2)^{\widetilde{\sigma}_\delta} \cdot e(K, Y)^{-\widetilde{\sigma}_q} \cdot e(T_1^*, G_2)^{\widetilde{\sigma}_y} \\ &\quad = e(K, G_2)^{\sigma_\delta^*} \cdot e(K, Y)^{-\sigma_q^*} \cdot e(T_1^*, G_2)^{\sigma_y^*} \end{aligned} \tag{1}$$

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

Type (a): We consider the situation that $\mathcal{A}$ generates a related query $(m^*, \Sigma = (\{T_i^*\}_{i\in[1,4]}, c^*, \sigma_x^*, \widetilde{\sigma}_y, \widetilde{\sigma}_\delta, \widetilde{\sigma}_q, \sigma_r^*))$ of 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 (1), it holds that

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

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 follows 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

$$k\frac{w\Delta\sigma_q - \Delta\sigma_\delta}{\Delta\sigma_y} = \frac{1}{w + y_{i_b}}\left(1 - hx_{i_b} - kz_{i_b}\right) + kq^*. \tag{2}$$

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 as 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 (2) 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 of 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 of Type (a) is negligible.

Type (b): Let $K = {G_1}^k$. When the equalities $\widetilde{\sigma}_y = \sigma_y^*$ and $\widetilde{\sigma}_q = \sigma_q^*$ are put in Equation (1), 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 of 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 of Type (b) is at most $1/p$ which is negligible.

Type (c): Let $K = {G_1}^k$. When the equalities $\widetilde{\sigma}_y = \sigma_y^*$ and $\widetilde{\sigma}_\delta = \sigma_\delta^*$ are put in Equation (1), 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 of 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 of 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 of Type ($\star$) since our attack is of this type. Now, we further analyze this type of query. This type of query satisfies $\widetilde{\sigma}_y = \sigma_y^*$. When this equality is put in Equation (1), 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 of Type ($\star$) knows the issuing key.

From the above observations, we see that only a related query of 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 Security of Mechanism 6

In this section, we formally prove the conjecture 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.

We firstly define some oracles for the adversary who cannot corrupt the issuer. The definitions of these oracles follow Bellare et al. [11]. The SndToI oracle is an interactive oracle. HU and CU are the set of honest users and corrupted users, respectively.

- AddU($\cdot$)**:** The add-user oracle takes as input a user identity i, and runs UKg and Join/Issue protocol to add an honest user i to the group. The oracle returns upk_i and adds i to HU.
- 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 Issue(gpk, upk_i, ik). The user i needs to be in the set CU. If $i \notin$ CU, the oracle outputs $\perp$.
- RReg($\cdot$)**:** The read-registration-table oracle takes as input i, and returns reg[i].

Next, 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 information about the signer 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.*

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

In the above experiment, we have Oracles = {AddU($\cdot$), CrptU($\cdot, \cdot$), SndToI($\cdot, \cdot$), USK($\cdot$), RReg($\cdot$), Ch($b, \cdot, \cdot, \cdot$), Open($\cdot, \cdot$)}. *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 the roles of both the opener and the issuer.

Theorem 4.2. *Mechanism 6 satisfies weak anonymity in the random oracle model 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 $(\mathbb{G}_1, \mathbb{G}_2)$.*

Due to space limitation, we omit the proof of Theorem 4.2. The formal proof is provided in the full version of this paper. We note that most of the proof is the same as that for anonymity under the restricted condition (Section 4.1) since anonymity in Definition 2.5 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 observed in Section 4.2, the adversary cannot generate related queries of Type (a), (b), and (c) under the DL assumption. Also in the proof, we show that the adversary who does not possess the issuing key cannot generate related queries of 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 [26] in our security proof.

4.4 Practical Implications

Now, we discuss the practical implications of our result. Specifically, we highlight the implications for the EPID scheme [5, 13], 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 shown 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 [13] (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 [13]. 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.

[5] Roughly, the values h_1, h_2, A, x, y, and f in the EPID scheme [13] correspond to the values H, K, A, y_i, z_i, and x_i in Mechanism 6 (showed in Figure 1), respectively.

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 [8, 20, 21, 28–30], 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 need a more detailed discussion, we hope that we have provided approaches to use the EPID scheme in the CCA setting.

5 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. Note that 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 by making the underlying proof system have unique responses [19, 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. Thus, the adversary cannot re-randomize a signature since there is no candidate for such a signature. However, when we employ a 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 re-randomizing the signature. Concretely, we add an equation to prove about the witness q (that is, $T_0 = G_1^q$) and 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 of Type (a), and the later case is of Type (b). As we proved, the probability that the adversary generates the related queries of 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. 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. That is, the signature size increases by only one element in $\mathbb{G}_1$ from Mechanism 6.

One concern is that information about the signer 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 information about the signer .

Security of the Patched Scheme. By the above modification, the patched scheme satisfies anonymity in the BSZ model. A signature in the patched scheme consists of two ElGamal encryptions (specifically, one is a double encryption) and a zero-knowledge proof. Intuitively, information about the signer is hidden from the adversary by the security of the encryption schemes and the zero-knowledge property of the underlying proof system. Therefore as in Theorem 4.1, we can easily see that the patched scheme satisfies anonymity if the adversary does not generate a related query.

In the patched scheme, we have four cases of a related query described in Table 2 since the element σ_q is fixed. For each type of related queries, we see that it is eliminated from the analysis in Section 4.2. The probability that an adversary generates a related query in Type (a) is negligible if the DL assumption holds, and the probabilities to generate a related query in Type (b) and (c) are at most $1/p$ and $2/p$, respectively. Thus, we can say that the patched scheme satisfies anonymity.

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

Table 2: Type of Related Queries for the Patched Scheme

Formally, the following theorem holds.

Theorem 5.1. *The patched scheme satisfies anonymity in the random oracle model 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 $\mathbb{G}$.*

Proof. At first, we define a related query for the patched scheme as in the case of Mechanism 6. A query $(\widetilde{m}, \widetilde{\Sigma} = (\{\widetilde{T}_i\}_{i\in[0,4]}, \widetilde{c}, \widetilde{\sigma}_x, \widetilde{\sigma}_y, \widetilde{\sigma}_\delta, \widetilde{\sigma}_q, \widetilde{\sigma}_r))$ is a related query if $(\widetilde{m}, \widetilde{\Sigma})$ is accepted by the verification algorithm, and it holds that

$$(\{\widetilde{T}_i\}_{i\in[0,4]}, \{\widetilde{R}_i\}_{i\in[1,5]}, \widetilde{m}) = (\{T_i^*\}_{i\in[0,4]}, \{R_i^*\}_{i\in[1,5]}, m^*)$$

where $(m^*, \Sigma^* = (\{T_i^*\}_{i\in[1,4]}, c^*, \sigma_x^*, \sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^*))$ is a pair of the challenge message and signature, and $(\widetilde{R}_1, \widetilde{R}_2, \widetilde{R}_3, \widetilde{R}_4, \widetilde{R}_5)$ and $(R_1^*, R_2^*, R_3^*, R_4^*, R_5^*)$ are the intermediate values computed in the verification of pairs $(\widetilde{m}, \widetilde{\Sigma})$ and (m^*, Σ^*), respectively.

GSig(gpk, gsk_i, m):

$r, q \xleftarrow{\$} \mathbb{Z}_p$; $\underline{T_0 \leftarrow {G_1}^q}$; $T_1 \leftarrow A_i \cdot K^q$; $T_2 \leftarrow G^{x_i+r}$; $T_3 \leftarrow U^r$; $T_4 \leftarrow V^r$; $\rho_{x_i}, \rho_{y_i}, \rho_\delta, \rho_q, \rho_r \xleftarrow{\$} \mathbb{Z}_p$

$R_1 \leftarrow e(H, G_2)^{\rho_{x_i}} \cdot e(K, G_2)^{\rho_\delta} \cdot e(K, Y)^{-\rho_q} \cdot e(T_1, G_2)^{\rho_{y_i}}$; $R_2 \leftarrow G^{\rho_{x_i}+\rho_r}$; $R_3 \leftarrow U^{\rho_r}$; $R_4 \leftarrow V^{\rho_r}$; $\underline{R_5 \leftarrow {G_1}^{\rho_q}}$

$c \leftarrow \mathsf{H}(\mathsf{gpk}, \{T_i\}_{i\in[0,4]}, \{R_i\}_{i\in[1,5]}, m)$; $\delta \leftarrow z_i - qy_i$

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

Return $\Sigma = (\{T_i\}_{i\in[0,4]}, c, \sigma_{x_i}, \sigma_{y_i}, \sigma_\delta, \sigma_q, \sigma_r)$

GVf(gpk, m, Σ):

$R'_1 \leftarrow e(H, G_2)^{\sigma_x} \cdot e(K, G_2)^{\sigma_\delta} \cdot e(K, Y)^{-\sigma_q} \cdot e(T_1, G_2)^{\sigma_y} \cdot \left(\frac{e(G_1, G_2)}{e(T_1, Y)}\right)^{-c}$

$R'_2 \leftarrow G^{\sigma_x+\sigma_r} \cdot {T_2}^{-c}$; $R'_3 \leftarrow U^{\sigma_r} \cdot {T_3}^{-c}$; $R'_4 \leftarrow V^{\sigma_r} \cdot {T_4}^{-c}$; $\underline{R'_5 \leftarrow {G_1}^{\sigma_q} \cdot {T_0}^{-c}}$

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: GSig **and** GVf **Algorithms of the Patched Scheme**

Now, we prove the statement by considering a sequence of games. Let $\mathcal{A}$ be an adversary that attacks the anonymity of the patched scheme Π, and S_ℓ denote the event that $\mathcal{A}$ succeeds in guessing the challenge bit b in Game ℓ.

[Game 0]: This is the experiment $\mathsf{Exp}^{anon}_{\Pi,\mathcal{A}}(\lambda)$ itself. The challenger manages an input/output pair of the random oracle in the list L as in the proof of Theorem 4.1.

[Game 1]: We modify the way of generating the challenge signature in Game 1. If the pair $((\mathsf{gpk}, \{T_i^*\}_{i\in[0,4]}, \{R_i^*\}_{i\in[1,5]}, m^*), \cdot)$ is already in the list L when computing the value $\mathsf{H}(\mathsf{gpk}, \{T_i^*\}_{i\in[0,4]}, \{R_i^*\}_{i\in[1,5]}, m^*)$, the challenger sets $\Sigma^* = \perp$.

Since T_1^* is uniformly random value in $\mathbb{G}_1$, the probability that there is already the same pair in L is at most q_H/p where q_H is the number of $\mathcal{A}$'s random oracle queries. Therefore, we have that $\left|\Pr[\mathsf{S}_0] - \Pr[\mathsf{S}_1]\right| \leq q_H/p$. That is, $\left|\Pr[\mathsf{S}_0] - \Pr[\mathsf{S}_1]\right|$ is negligible.

[Game 2]: We further modify the way of generating the challenge signature. Here, the challenge signature is generated as follows: Firstly, the challenger chooses values $r^*, q^* \in \mathbb{Z}_p$ uniformly random and compute $\{T_i^*\}_{i\in[0,4]}$. Secondly, the challenger chooses σ_x^*, $\sigma_y^*, \sigma_\delta^*, \sigma_q^*, \sigma_r^* \in \mathbb{Z}_p$ and $c^* \in \mathbb{Z}_p$ uniformly random, and computes $\{R_i^*\}_{i\in[1,5]}$. Finally, the challenger defines a value $\mathsf{H}(\mathsf{gpk}, \{T_i^*\}_{i\in[0,4]}, \{R_i^*\}_{i\in[1,5]}, m^*)$. If the pair $((\mathsf{gpk}, \{T_i^*\}_{i\in[0,4]}, \{R_i^*\}_{i\in[1,5]}, m^*), \cdot)$ is already in the list L, the challenger sets $\Sigma^* = \perp$.

Since the modification between Game 1 and Game 2 is only conceptual, $\Pr[\mathsf{S}_1] = \Pr[\mathsf{S}_2]$ holds.

[Game 3]: In Game 3, we modify the way of generating a proof τ in replying queries for the Open oracle. If the pair $((\mathsf{gpk}, Q, T_2, T_3, R), \cdot)$ is already in the list L when computing the value $\mathsf{H}(\mathsf{gpk}, Q, T_2, T_3, R)$, the challenger returns $\perp$ as the response of the query.

Here, we can do the same discussion as Game 1. Thus, we can say that $\left|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]\right|$ is negligible.

[Game 4]: We further modify the way of generating a proof τ in replying queries for the Open oracle. When the challenger replies for a query $(m, \Sigma = (\{T_i\}_{i\in[1,4]}, c, \sigma_x, \sigma_y, \sigma_\delta, \sigma_q, \sigma_r))$, the challenger computes $R = (Q \cdot {T_2}^{-1})^{\sigma_u} \cdot T_3^{-d}$ by choosing $\sigma_u \in \mathbb{Z}_p$ and $d \in \mathbb{Z}_p$ uniformly random.

Also, the modification between Game 3 and Game 4 is only conceptual, and therefore $\Pr[\mathsf{S}_3] = \Pr[\mathsf{S}_4]$ holds.

[Game 5]: We modify the way of generating the challenge signature. In this game, the challenger computes $T_0 = {G_1}^{q^*}$, $T_1 = A_i \cdot K^{\widehat{q}^*}$ instead of $T_0 = {G_1}^{q^*}$, $T_1 = A_i \cdot K^{q^*}$ where $\widehat{q}^*$ is a random value in $\mathbb{Z}_p$.

Since the tuples $(G_1, G_2, {G_1}^{q^*}, K, K^{q^*})$ and $(G_1, G_2, {G_1}^{q^*}, K, K^{\widehat{q}^*})$ are indistinguishable by the XDH assumption, this modification does not affect the adversary's behavior. Therefore, it holds that $\left|\Pr[\mathsf{S}_4] - \Pr[\mathsf{S}_5]\right|$ is negligible.

[Game 6]: This game is defined as Game 5 except that the challenger computes $T_2^* = G^{x_{i_b}+r^*}$, $T_4^* = G^{r_2^*}$ where r_2^* is a random value in $\mathbb{Z}$. In the previous games, they were computed as $T_2^* = G^{x_{i_b}+r^*}, T_4^* = V^{r^*}$.

Since the tuples (G, G^{r^*}, V, V^{r^*}) and $(G, G^{r^*}, V, G^{r_2^*})$ are indistinguishable by the DDH assumption, this modification does not affect the adversary's behavior. Thus, it holds that $\left|\Pr[\mathsf{S}_5] - \Pr[\mathsf{S}_6]\right|$ is negligible.

[Game 7]: Here, we modify the way of replying to opening queries. If a query is a related query, return $\perp$ as a opening proof.

For related queries in Table 2, the probability that an adversary generates each type of them is negligible. Type (a) is negligible by the DL assumption, and that Type (b) and (c) are negligible by using information-theoretic arguments. Therefore, we see that $\left|\Pr[\mathsf{S}_6] - \Pr[\mathsf{S}_7]\right|$ is negligible.

[Game 8]: From this game, the challenger use key v to open signatures instead of key u. In Game 8, the challenger sets $Q = T_2 \cdot (T_4^{\frac{1}{v}})^{-1}$ by comparing the previous games in which the challenger sets $Q = T_2 \cdot (T_3^{\frac{1}{u}})^{-1}$.

Since the underlying non-interactive proof system has soundness, it is negligible that $\log_U T_3 \neq \log_V T_4$ holds. Therefore, we see that $\left|\Pr[\mathsf{S}_7] - \Pr[\mathsf{S}_8]\right|$ is negligible.

[Game 9]: In this game, the challenger computes $T_2^* = G^{x_{i_b}+r^*}$, $T_3^* = G^{r_1^*}$ where r_1^* is a random value in $\mathbb{Z}$. In the previous games, they were computed as $T_2^* = G^{x_{i_b}+r^*}$, $T_3^* = U^{r^*}$.

As in Game 6, this modification does not affect the adversary's behavior by the DDH assumption. Thus, it holds that $\left|\Pr[\mathsf{S}_8] - \Pr[\mathsf{S}_9]\right|$ is negligible.

The choice of the challenge bit b and the distribution of the challenge signature Σ^* are independent. Also, the oracles, especially the Open oracle, behave independently of b. Thus, we can say that $\Pr[S_9] = 1/2$. Since it holds that $\mathrm{Adv}^{anon}_{\Pi,\mathcal{A}}(\lambda) \leq \sum_{\ell=0}^{8} \left|\Pr[S_\ell] - \Pr[S_{\ell+1}]\right| + \left|\Pr[S_9] - 1/2\right|$, $\mathrm{Adv}^{anon}_{\Pi,\mathcal{A}}(\lambda)$ is negligible. Therefore, the patched scheme Π satisfies anonymity. □

Moreover, the patched scheme satisfies the other security requirements, that is, traceability and non-frameability [11]. This is because our modification does not affect these security proofs. The underlying proof system is still extractable. Also, an adversary can simulate the Join/Issue protocol as the issuer without the issuing key in the traceability game, and simulate the signing oracle without the honest user's signing key in the non-frameability game. Therefore, we can prove the traceability and non-frameability of the patched scheme in the same way as those of the original scheme. Note that in the sense of traceability and non-frameability, the original scheme is secure as it is. Formally, the following theorems hold.

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

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

Efficiency. In the patched scheme, the signature size increases by only one element in $\mathbb{G}_1$ from Mechanism 6. 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 efficiency comparable to the existing schemes [17, 18] satisfying the same security level. Specifically, a signature in the Delerablée-Pointcheval scheme [17] consists of four elements in $\mathbb{G}_1$ and five elements in $\mathbb{Z}_p$, and in the Derler-Slamanig scheme [18], 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. 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. We have seen that no one can extract information about the signer 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 the 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 shown under the same assumptions as those of Mechanism 6. Our patched scheme could be a candidate for the new standardized scheme when ISO/IEC 20008-2 will be revised in the future.

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/presentations 2011/brickell.pdf.
[8] Bahmani, R., Barbosa, M., Brasser, F., Portela, B., Sadeghi, A., Scerri, G., and Warinschi, B. Secure multiparty computation from SGX. In *FC* (2017), pp. 477–497.
[9] Bardou, R., Focardi, R., Kawamoto, Y., Simionato, L., Steel, G., and Tsay, J. Efficient padding oracle attacks on cryptographic hardware. In *CRYPTO* (2012), pp. 608–625.
[10] Bellare, M., and Rogaway, P. The exact security of digital signatures - how to sign with RSA and rabin. In *EUROCRYPT* (1996), pp. 399–416.
[11] Bellare, M., Shi, H., and Zhang, C. Foundations of group signatures: The case of dynamic groups. In *CT-RSA* (2005), pp. 136–153.
[12] Bleichenbacher, D. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *CRYPTO* (1998), pp. 1–12.
[13] Brickell, E., and Li, J. Enhanced privacy ID from bilinear pairing for hardware authentication and attestation. In *SocialCom/PASSAT* (2010), pp. 768–775.
[14] Chaum, D., and van Heyst, E. Group signatures. In *EUROCRYPT* (1991), pp. 257–265.
[15] Coron, J., Naccache, D., and Stern, J. P. On the security of RSA padding. In *CRYPTO* (1999), pp. 1–18.
[16] Coron, J., Naccache, D., Tibouchi, M., and Weinmann, R. Practical cryptanalysis of ISO/IEC 9796-2 and EMV signatures. In *CRYPTO* (2009), pp. 428–444.
[17] Delerablée, C., and Pointcheval, D. Dynamic fully anonymous short group signatures. In *VIETCRYPT* (2006), pp. 193–210.
[18] Derler, D., and Slamanig, D. Highly-efficient fully-anonymous dynamic group signatures. In *ASIACCS* (2018), pp. 551–565.
[19] Faust, S., Kohlweiss, M., Marson, G. A., and Venturi, D. On the non-malleability of the fiat-shamir transform. In *INDOCRYPT* (2012), pp. 60–79.
[20] Fisch, B., Vinayagamurthy, D., Boneh, D., and Gorbunov, S. IRON: functional encryption using intel SGX. In *CCS* (2017), pp. 765–782.
[21] Fuhry, B., Bahmani, R., Brasser, F., Hahn, F., Kerschbaum, F., and Sadeghi, A. Hardidx: Practical and secure index with SGX. In *DBSec* (2017), pp. 386–408.
[22] Furukawa, J., and Imai, H. An efficient group signature scheme from bilinear maps. In *ACISP* (2005), pp. 455–467.
[23] Furukawa, J., and Imai, H. An efficient group signature scheme from bilinear maps. *IEICE Transactions 89-A*, 5 (2006), 1328–1338.
[24] Hwang, J. Y., Lee, S., Chung, B., Cho, H. S., and Nyang, D. Group signatures with controllable linkability for dynamic membership. *Inf. Sci. 222* (2013), 761–778.
[25] Isshiki, T., Mori, K., Sako, K., Teranishi, I., and Yonezawa, S. Using group signatures for identity management and its implementation. In *Digital Identity Management* (2006), pp. 73–78.
[26] Pointcheval, D., and Stern, J. Security arguments for digital signatures and blind signatures. *J. Cryptology 13*, 3 (2000), 361–396.
[27] Sahai, A. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS* (1999), pp. 543–553.
[28] Sasy, S., Gorbunov, S., and Fletcher, C. W. Zerotrace : Oblivious memory primitives from intel SGX. In *NDSS* (2018).
[29] Schuster, F., Costa, M., Fournet, C., Gkantsidis, C., Peinado, M., Mainar-Ruiz, G., and Russinovich, M. VC3: trustworthy data analytics in the cloud using SGX. In *Security and Privacy* (2015), pp. 38–54.
[30] Seo, J., Lee, B., Kim, S. M., Shih, M., Shin, I., Han, D., and Kim, T. Sgx-shield: Enabling address space layout randomization for SGX programs. In *NDSS* (2017).
[31] Swami, Y. SGX remote attestation is not sufficient. *IACR Cryptology ePrint Archive 2017* (2017), 736.
[32] Unruh, D. Quantum proofs of knowledge. In *EUROCRYPT* (2012), pp. 135–152.

APPENDIX

A Proofs of Lemma 4.1 to 4.7

Here, we show Lemma 4.1 to 4.7 and complete the proof of Theorem 4.1.

A Proof of Lemma 4.1. First of all, we define the event $\mathsf{Bad}_\ell^{(1)}$ *as follows.*

$\mathsf{Bad}_\ell^{(1)}$: *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 $\mathsf{Bad}_0^{(1)}$ *and* $\mathsf{Bad}_1^{(1)}$ *occur. That is, we get* $\Pr[\mathsf{S}_0 \wedge \neg\mathsf{Bad}_0^{(1)}] = \Pr[\mathsf{S}_1 \wedge \neg\mathsf{Bad}_1^{(1)}]$. *Therefore, it holds that* $\left|\Pr[\mathsf{S}_0] - \Pr[\mathsf{S}_1]\right| = \left|\Pr[\mathsf{S}_0 \wedge \mathsf{Bad}_0^{(1)}] + \Pr[\mathsf{S}_0 \wedge \neg\mathsf{Bad}_0^{(1)}] - \Pr[\mathsf{S}_1 \wedge \mathsf{Bad}_1^{(1)}] - \Pr[\mathsf{S}_1 \wedge \neg\mathsf{Bad}_1^{(1)}]\right| = \left|\Pr[\mathsf{S}_0 \wedge \mathsf{Bad}_0^{(1)}] - \Pr[\mathsf{S}_1 \wedge \mathsf{Bad}_1^{(1)}]\right| \leq \Pr[\mathsf{Bad}_1^{(1)}]$.

Here, we estimate the probability $\Pr[\mathsf{Bad}_1^{(1)}]$. *When the event* $\mathsf{Bad}_1^{(1)}$ *occurs,* $\widetilde{T_1} = T_1^*$ *holds for some defined value* $((\cdot, \widetilde{T_1}, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot), \cdot)$ *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,* $\Pr[\mathsf{Bad}_1^{(1)}] \leq q_H/p$. *Thus, we obtain* $\left|\Pr[\mathsf{S}_0] - \Pr[\mathsf{S}_1]\right| \leq q_H/p$. □

A Proof of Lemma 4.2. 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:*

$$\begin{aligned} R_1^* =& e(H, G_2)^{\sigma_x^*} \cdot e(K, G_2)^{\sigma_\delta^*} \cdot e(K, Y)^{-\sigma_q^*} \\ & \cdot e(T_1^*, G_2)^{\sigma_y^*} \cdot \left(\frac{e(G_1, G_2)}{e(T_1^*, Y)}\right)^{-c^*} \\ =& e(H, G_2)^{\rho_x^*} \cdot e(K, G_2)^{\rho_\delta^*} \cdot e(K, Y)^{-\rho_q^*} \cdot e(T_1{}^*, G_2)^{\rho_y^*}, \\ R_2^* =& G^{\sigma_x^* + \sigma_r^*} \cdot (T_2^*)^{-c^*} = G^{\rho_x^* + \rho_r^*}, \\ R_3^* =& U^{\sigma_r^*} \cdot (T_3^*)^{-c^*} = U^{\rho_r^*},\ R_4^* = V^{\sigma_r^*} \cdot (T_4^*)^{-c^*} = V^{\rho_r^*}. \end{aligned}$$

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,* $\Pr[\mathsf{S}_1] = \Pr[\mathsf{S}_2]$. □

A Proof of Lemma 4.3. We define the event $\mathsf{Bad}_\ell^{(2)}$ *as follows.*

$\mathsf{Bad}_\ell^{(2)}$: *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 $\mathsf{Bad}_2^{(2)}$ *and* $\mathsf{Bad}_3^{(2)}$ *occur. Therefore, we get* $\left|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]\right| \leq \Pr[\mathsf{Bad}_3^{(2)}]$ *same as in Lemma 4.1.*

Here, we estimate the probability $\Pr[\mathsf{Bad}_3^{(2)}]$. *When the event* $\mathsf{Bad}_3^{(2)}$ *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,* $\Pr[\mathsf{Bad}_3^{(2)}] \leq q_H \cdot q_{\mathsf{open}}/p$ *holds. Thus, we obtain* $\left|\Pr[\mathsf{S}_2] - \Pr[\mathsf{S}_3]\right| \leq q_H \cdot q_{\mathsf{open}}/p$. □

A Proof of Lemma 4.4. 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,* $\Pr[\mathsf{S}_3] = \Pr[\mathsf{S}_4]$. □

A Proof of Lemma 4.5. 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 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* $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), \cdot)$ *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* $\mathsf{Adv}_{\mathcal{B}_1}^{DDH}(\lambda) = |\Pr[1 \leftarrow \mathcal{B}_1(G, G^v, G^r, G^{vr})] - \Pr[1 \leftarrow \mathcal{B}_1(G, G^v, G^r, W)]| = |\Pr[b = \widetilde{b} \text{ in Game 4}] - \Pr[b = \widetilde{b} \text{ in Game 5}]| = \left|\Pr[\mathsf{S}_4] - \Pr[\mathsf{S}_5]\right|$. □

A Proof of Lemma 4.6. We define the event $\mathsf{Bad}_\ell^{(3)}$ *as follows.*

$\mathsf{Bad}_\ell^{(3)}$: *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 $\mathsf{Bad}_5^{(3)}$ *and* $\mathsf{Bad}_6^{(3)}$ *occur. Therefore, we get* $\left|\Pr[\mathsf{S}_5] - \Pr[\mathsf{S}_6]\right| \leq \Pr[\mathsf{Bad}_6^{(3)}]$ *same as in Lemma 4.1. Moreover, we define the event* $\overline{\mathsf{Bad}}_6^{(3)}$ *as follows.*

$\overline{\mathsf{Bad}}_6^{(3)}$**:** *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*

$$\begin{pmatrix} \log_U R_3 \\ \log_V R_4 \end{pmatrix} = \begin{pmatrix} 1 & -\log_U T_3 \\ 1 & -\log_V T_4 \end{pmatrix} \begin{pmatrix} \sigma_r \\ \widetilde{c} \end{pmatrix} \tag{3}$$

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*

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

Therefore, the simultaneous equation (3) 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 (3) holds for* $\widetilde{c}$ *is* $1/p$*. That is,* $\Pr[\overline{\mathsf{Bad}}_6^{(3)}] = 1/p$.

In the following, we prove $\left|\Pr[\mathsf{S}_5] - \Pr[\mathsf{S}_6]\right| \leq 1/p$ *by showing* $\mathsf{Bad}_6^{(3)} \subseteq \overline{\mathsf{Bad}}_6^{(3)}$*. We consider that the event* $\mathsf{Bad}_6^{(3)}$ *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* $\overline{\mathsf{Bad}}_6^{(3)}$*, and there exists* σ_r *which satisfies the equation (3) for the reply* $\widetilde{c}$*. Thus,* $\mathsf{Bad}_6^{(3)} \subseteq \overline{\mathsf{Bad}}_6^{(3)}$ *holds and we obtain* $\left|\Pr[\mathsf{S}_5] - \Pr[\mathsf{S}_6]\right| \leq \Pr[\mathsf{Bad}_6^{(3)}] \leq \Pr[\overline{\mathsf{Bad}}_6^{(3)}] = 1/p$. □

A Proof of Lemma 4.7. 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* KGg *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 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* $((\mathsf{gpk}, \{T_i^*\}_{i\in[1,4]}, \{R_i^*\}_{i\in[1,4]}, m^*), \cdot)$ *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* $\mathsf{Adv}^{DDH}_{\mathcal{B}_2}(\lambda) = |\Pr[1 \leftarrow \mathcal{B}_2(G, G^u, G^r, G^{ur})] - \Pr[1 \leftarrow \mathcal{B}_2(G, G^u, G^r, W)]| = |\Pr[b = \widetilde{b} \text{ in Game 6}] - \Pr[b = \widetilde{b} \text{ in Game 7}]| = \left|\Pr[\mathsf{S}_6] - \Pr[\mathsf{S}_7]\right|$. □