Improving the Efficiency of Report and Trace Ring Signatures

Xavier Bultel, Ashley Fraser, Elizabeth Quaglia

Improving the Efficiency of Report and Trace Ring Signatures

Xavier Bultel[1], Ashley Fraser[2⋆], and Elizabeth A. Quaglia[3]

[1] INSA Centre Val de Loire, LIFO, France
[2] Department of Computer Science, University of Surrey, UK
[3] Information Security Group, Royal Holloway, University of London, UK

Abstract. Ring signatures allow signers to produce verifiable signatures and remain anonymous within a set of signers (i.e., the ring) while doing so. They are well-suited to protocols that target anonymity as a primary goal, for example, anonymous cryptocurrencies. However, standard ring signatures do not ensure that signers are held accountable if they act maliciously. Fraser and Quaglia (CANS'21) introduced a ring signature variant that they called report and trace ring signatures which balances the anonymity guarantee of standard ring signatures with the need to hold signers accountable. In particular, report and trace ring signatures introduce a reporting system whereby ring members can report malicious message/signature pairs. A designated tracer can then revoke the signer's anonymity if, and only if, a ring member submits a report to the tracer. Fraser and Quaglia present a generic construction of a report and trace ring signature scheme and outline an instantiation for which it is claimed that the complexity of signing is linear in the size of the ring $|R|$.
In this paper, we introduce a new instantiation of Fraser and Quaglia's generic report and trace ring signature construction. Our instantiation uses a pairing-based variant of ElGamal that we define. We demonstrate that our instantiation is more efficient. In fact, we highlight that the efficiency of Fraser and Quaglia's instantiation omits a scaling factor of λ where λ is a security parameter. As such, the complexity of signing for their instantiation grows linearly in $\lambda \cdot |R|$. Our instantiation, on the other hand, achieves signing complexity linear in $|R|$. We also introduce a new pairing-free report and trace ring signature construction reaching a similar signing complexity. Whilst this construction requires some additional group exponentiations, it can be instantiated over any prime order group for which the Decisional Diffie-Hellman assumption holds.

1 Introduction

In the context of distributed systems, it is often necessary to balance the competing goals of anonymity and accountability. On the one hand, there is an expectation of privacy by the system's users; on the other, the system must be able to hold misuse accountable. The need to balance these goals is particularly

⋆ This author was funded by EPSRC under the DECaDE project P/T022485/1.

true for cryptocurrencies, which are typically deployed atop a distributed ledger. Indeed, several cryptocurrencies target user anonymity as a primary security goal [3, 11, 22, 23, 27], ensuring that users can transact without revealing their identity. In particular, Monero [23] uses a ring signature [25], a cryptographic tool that allows users to sign transactions within a group of users known as the ring, thus ensuring that the signer is anonymous within the ring. However, using a standard ring signature means that tracing a fraudulent transactor is difficult. As such, Monero cannot provide a guarantee of accountability.

The notion of a standard ring signature has been extended to incorporate accountability. Specifically, Xu and Yung introduced accountable ring signatures [31], which introduce a designated tracer that can revoke the anonymity of signers. More recently, Fraser and Quaglia presented report and trace ring (RTR) signatures [13]. This new ring signature variant builds upon the functionality of accountable ring signatures, requiring that the designated tracer can revoke anonymity only if a ring member first sends a report of malicious behaviour to the designated tracer[4].

Report and Trace Ring Signatures. Similar to standard ring signatures, RTR signatures allow signers to generate signatures with respect to a group (i.e., ring) of users, and the signer is anonymous within the ring. Additionally, RTR signatures provide a mechanism whereby ring members can produce a report for a signed message. Upon receiving a report, a designated tracer can trace the signer's identity. Fraser and Quaglia defined report and trace ring signatures in [13], and provided a complete security model for the primitive. Accompanying this formalisation, the authors present a provably secure generic construction and concrete instantiation of an RTR signature.

With respect to the instantiation, we note two drawbacks that we aim to address. Firstly, the instantiation is not as efficient as claimed. In fact, during signing, the instantiation uses Stadler's zero-knowledge proof [29] to prove correct encryption of a reporter token. Stadler's proof must be repeated λ times to be secure, where λ is the security parameter. As such, the complexity of the proof is linear in the security parameter. The efficiency analysis of the instantiation presented in [13] omits the security parameter. That is, signing is claimed to be linear in the size of the ring $|R|$ but is, in fact, linear in $\lambda \cdot |R|$. Secondly, the instantiation relies on a number theoretical group. As a consequence, the instantiation does not reap the efficiency benefits of the most efficient groups such as those based on elliptical curves.

Our Contributions. This work addresses the limitations of the existing RTR signature instantiation. Namely, we introduce a new instantiation of the generic construction in [13] (Section 4) that is more efficient than the instantiation in [13] with respect to signing. Then, we introduce a new RTR signature scheme

[4] For simplicity, we also consider a single designated tracer in this work. We note, however, that this role can be distributed using standard secret sharing techniques, making it more suitable for decentralised applications.

construction (Section 5) that can be instantiated with any group for which the Decisional Diffie-Hellman assumption holds. Here, we provide a brief overview of our results.

In the generic construction of [13], during signing, the signer generates a reporter token that is encrypted to each ring member. The signer also generates a proof of correct encryption of the reporter token. In the instantiation of [13], this functionality is realised with standard ElGamal encryption and Stadler's zero-knowledge proof [29]. In this paper, we introduce a new instantiation which relies on a bespoke pairing-based variant of the ElGamal public-key encryption scheme. Furthermore, we demonstrate that the zero-knowledge proof of correct encryption, which requires proof of equality of pairings for our new variant, can be instantiated using the Fiat-Shamir transformation [12] on the variant of the Schnorr protocol from [9]. Accordingly, the complexity of signing for our instantiation grows linearly only in the size of the ring, improving upon the efficiency of the instantiation from [13]. In Section 3, we introduce our new pairing-based public-key encryption scheme and demonstrate how to prove correctness of encryption for our new scheme. We also discuss the security and efficiency of our instantiation.

We then propose a new RTR signature scheme construction in Section 5. This new scheme follows the syntax of an RTR signature, as outlined in Section 2, but differs from existing constructions, namely, the construction of [13]. Our new construction is pairing-free and can be instantiated with any group in which the Decisional Diffie Hellman assumption holds. Thus, our new construction allows for the use of more efficient and standard prime order groups (e.g., elliptic curves) than our instantiation (Section 4) and Fraser and Quaglia's instantiation [13]. We demonstrate that our new construction is secure and can be instantiated using standard cryptographic protocols from the literature. We conclude with a brief discussion of its efficiency, showing that although it requires more group exponentiations for signing and produces signatures that contain more group elements than our instantiation in Section 3, it achieves signing complexity that is linear in the size of the ring.

Other Related Work. Several ring signatures [25] variants aim to balance anonymity with accountability. As mentioned in introduction, accountable ring signatures [31, 4] allow signers to generate ring signatures and remain anonymous within the ring, unless a designated tracer reveals the signer's identity. Moreover, linkable [21] and traceable [14] ring signatures allow tracers to determine whether two signatures are generated by the same, or different, users. Addressing the balance of anonymity and accountability has frequently arisen with respect to other cryptographic protocols. Notably, many group signature [8] variants introduce measures whereby signer anonymity can be revoked [18, 19, 26]. Additionally, anonymity and accountability has been discussed in relation to end-to-end encryption [30] and systems that permit the reporting of malicious, and perhaps criminal, behaviour [1, 17, 20, 24].

2 Preliminaries

In this section, we define the notations and the tools that we use in this paper. More detailed and formal definitions are given in Appendix A.

The Decisional Diffie-Hellman (DDH) assumption. Let $\mathbb{G} = \langle g \rangle$ be a group of prime order p. Picking $b \xleftarrow{\$} \{0,1\}$ and $(x, y, z_1) \xleftarrow{\$} (\mathbb{Z}_p^*)^3$, and setting $z_0 = a \cdot b$ and $(X, Y, Z) = (g^x, g^y, g^{z_b})$, the DDH assumption in $\mathbb{G}$ states that no Probabilistic Polynomial Time (PPT) algorithm is able to return b on input (X, Y, Z) with non-negligible advantage.

Non-Interactive Zero-Knowledge Proof of Knowledge (NIZKP). Let $\mathcal{R}$ be a binary relation and let $\mathcal{L}$ be a language such that $s \in \mathcal{L} \Leftrightarrow (\exists w, (s, w) \in \mathcal{R})$. According to the Camenisch-Stadler notation [5], $\mathsf{NIZK}\{w : (s, w) \in \mathcal{R}\}$ denotes a NIZKP of w for the langage $\mathcal{L}$. A NIZKP is said to be extractable when there exists a PPT knowledge extractor that efficiently extracts a witness w from any PPT algorithm that forges valid proofs of knowledge for a given statement s such that $(s, w) \in \mathcal{R}$. Moreover, a NIZKP is said to be zero-knowledge when there exists a PPT simulator that takes a statement s as input and that produces proofs that are indistinguishable from those outputted by the real NIZKP protocol on s.

Signature of Knowledge (SoK). A SoK [6] on a message m, denoted by $\mathsf{SoK}_m\{w : (s, w) \in \mathcal{R}\}$, is similar to a NIZKP except that the message m is embedded in the proof. w is seen as a secret key and s as the corresponding public key. Since the knowledge of w is required to generate a valid SoK on a message m, a SoK is unforgeable, which is the standard security requirement the digital signatures.

We also recall the ElGamal encryption scheme and the IND-CPA security definition in Appendix A.

3 Syntax and Security Model

We recall the syntax and security model of a report and trace ring (RTR) signature scheme as presented in [13]. In an RTR signature, users sign messages with respect to a ring. The signer cannot be identified (i.e., is anonymous within the ring) unless a ring member generates an anonymous report and transmits the report to the designated tracer, who can then reveal the signer's identity. We adopt the notation conventions of [13], writing T to denote the tracer and U to denote a user from a set of users $\mathcal{U}$.

Definition 1 (RTR signature). *An* RTR signature scheme *is a tuple of algorithms (*Setup, T.KGen, U.KGen, Sign, Verify, Report, Trace, VerTrace*) defined as follows:*

$\mathsf{Setup}(1^\lambda) \to pp$**:** On input security parameter 1^λ, outputs public parameters pp.

$\mathsf{T.KGen}(pp) \to (\mathsf{pk_T}, \mathsf{sk_T})$**:** On input pp, outputs a tracer public key $\mathsf{pk_T}$ and secret key $\mathsf{sk_T}$.

$\mathsf{U.KGen}(pp) \to (\mathsf{pk_U}, sk_\mathsf{U})$**:** On input pp, outputs a user public key $\mathsf{pk_U}$ and secret key $\mathsf{sk_U}$.

$\mathsf{Sign}(pp, sk_\mathsf{U}, \mathsf{pk_T}, m, R) \to \sigma$**:** On input pp, sk_U, $\mathsf{pk_T}$, message m and ring R, outputs a signature σ.

$\mathsf{Verify}(pp, \mathsf{pk_T}, m, R, \sigma) \to \{0,1\}$**:** On input pp, $\mathsf{pk_T}$, m, R and σ, outputs 1 if σ is a valid signature on m with respect to R, and 0 otherwise.

$\mathsf{Report}(pp, \mathsf{pk_T}, sk_\mathsf{U}, m, R, \sigma) \to \mathtt{Rep}$**:** On input pp, $\mathsf{pk_T}$, sk_U, m, R and σ, outputs a reporter token $\mathtt{Rep}$.

$\mathsf{Trace}(pp, sk_\mathsf{T}, m, R, \sigma, \mathtt{Rep}) \to (\mathsf{pk_U}, \mathtt{Tr}, \rho_t)$**:** On input pp, sk_T, m, R, σ and $\mathtt{Rep}$, outputs the signer's identity $\mathsf{pk_U}$, auxiliary information $\mathtt{Tr}$ consisting of the reporter token, and a proof of correct trace ρ_t.

$\mathsf{VerTrace}(pp, \mathsf{pk_T}, m, R, \sigma, \mathsf{pk_U}, \mathtt{Tr}, \rho_t) \to \{0,1\}$**:** On input pp, $\mathsf{pk_T}$, m, R, σ, $\mathsf{pk_U}$, $\mathtt{Tr}$ and ρ_t, outputs 1 if the trace is valid, and 0 otherwise.

An RTR signature must satisfy correctness and trace correctness. Informally, correctness requires that algorithm Verify outputs 1 if the signature is the output of algorithm Sign (and setup/key generation is honestly executed) with overwhelming probability. Trace correctness necessitates that algorithm VerTrace outputs the correct signer's identity for a signature output by algorithm Sign with overwhelming probability. Correctness and trace correctness for RTR signatures are introduced and formally defined in [13], and we recall the formal definitions in Appendix B.

3.1 Security Model

RTR signatures must satisfy anonymity, unforgeability, non-frameability, trace soundness and reporter anonymity. These properties are defined in [13]. Here, we provide an overview of these properties, and present the full formal security model in Appendix B for reference. The security experiments model an attacker that can register and corrupt/control users (i.e., obtain their honestly-generated secret keys/generate keys on their behalf), and generate signatures, reports and traces through access to several oracles. We present these oracles in full in Appendix B alongside the formal definition of the security model.

Anonymity. Anonymity requires that, on the condition that a signature is not reported and the signer traced, a signature does not reveal the signer's identity. Anonymity for RTR signatures, as defined in [13], adjusts the definition of anonymity against adversarially generated keys in [2]. In doing so, it is assumed that the attacker can control users and reporters. However, the tracer is assumed to be honest. In the anonymity experiment, the adversary outputs a message, ring and two potential signers (who are assumed to be honest). The adversary obtains a signature and outputs a bit to indicate which signer produced the signature. An RTR signature is anonymous if the adversary cannot determine which of the two potential signers generated the signature.

Unforgeability. Unforgeability for RTR signatures is adapted from the standard definition of unforgeability for ring signatures presented in [2]. It requires that an attacker cannot produce a valid ring signature on behalf of a member of an honest ring. An attacker is assumed to control the tracer, and can corrupt and control users. In the security experiment, the adversary outputs a message, ring and signature (which is not obtained via a signing oracle). If the signature is valid, we say that the adversary has produced a valid forgery. An RTR signature scheme satisfies unforgeability if the adversary cannot construct a valid forgery.

Non-frameability. Intuitively, non-frameability captures the property that a non-signer cannot be identified as the signer by the designated tracer. The formal non-frameability experiment models an attacker that can control the tracer, and can corrupt and control signers. The adversary outputs a message, ring, signature and trace, where the traced signer is assumed to be honest. An RTR signature scheme satisfies non-frameability if algorithm VerTrace returns 0.

Trace soundness. In [4], trace soundness was introduced as a new security property for accountable ring signatures. Trace soundness states that the signer identified by the tracer must be unique. In other words, two users can be verifiably identified as signers. In [13], the trace soundness property is adapted to the syntax of an RTR signature, and, like the original definition in [4], models an attacker that controls the tracer and can corrupt and control all signers. In the formal security experiment, the adversary outputs two traces (where each trace identifies a different ring member as the signer) alongside a message, ring and signature. The trace soundness property is satisfied if algorithm VerTrace does not output 1 for both traces.

Reporter Anonymity. Reporter anonymity requires that a report does not reveal the ring member that produced it, if it is assumed that the reporter is honest. The attacker can control the tracer and corrupt/control a subset of users. In the reporter anonymity experiment, the adversary outputs a message, ring, signature and the two ring members (i.e., two potential reporters). The adversary obtains a report and outputs a bit to indicate which reporter generated the report. An RTR signature satisfies reporter anonymity if the adversary cannot determine which reporter produced the report.

4 An Efficient Instantiation of Fraser and Quaglia's Protocol

Fraser and Quaglia present a generic construction for an RTR signature in [13]. We provide a brief intuition into their construction here and refer the reader to [13] for full details. During key generation, users (i.e., ring members) and the tracer generate a keypair for a public-key encryption (PKE) scheme. To sign a message, signers generate a fresh key pair for a PKE scheme. The fresh secret key is known as the reporter token. The reporter token is encrypted under the public

key of each ring member. Then, the signer encrypts their identity under the public key of the tracer and then again under the fresh public key. The signer also constructs a zero-knowledge proof (NIZK) that the reporter token is encrypted to all ring members, and a signature of knowledge (SoK) that the signer's identity is encrypted to the tracer. Ring members can report signatures by decrypting the reporter token using their decryption key for the PKE scheme. A tracer can decrypt the signer's identity using their decryption key and the reporter token.

Fraser and Quaglia also present a concrete instantiation of their construction. A central requirement of their construction is that the signer must prove that a ciphertext encrypts a secret key (i.e., the reporter token) that corresponds to a given public key. Fraser and Quaglia propose to instantiate their construction with the original ElGamal cryptosystem and Stadler's zero-knowledge proof [29], which ensures that an ElGamal ciphertext encrypts a discrete logarithm in a zero-knowledge way. However, this approach has two drawbacks. Firstly, Stadler's proof has complexity linear in the security parameter as the proof must be repeated λ times. Secondly, the proof only works for number-theoretic groups of prime order, and cannot be extended to groups based on elliptic curves.

In what follows, we propose an instantiation of Fraser and Quaglia's construction using a variant of ElGamal based on bilinear maps, overcoming the first drawback (we address the second drawback in Section 5). Our new instantiation differs from Fraser and Quaglia's instantiation in the following respect. We use our ElGamal variant to generate the reporter token and encrypt the signer's identity under the reporter token. Then, we modify the zero-knowledge proof for our ElGamal variant. In all other respects, our instantiation is identical. In particular, we use a one-way function to generate the signer's public identity, the SoK of [4], and we use standard ElGamal encryption to encrypt the reporter token to the ring members and the signer's identity to the tracer. Now, we introduce our new ElGamal variant, and then discuss the security and efficiency of our new instantiation.

4.1 A Pairing-Based ElGamal Variant

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_t$ be groups of prime order p, $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ be generators, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$ be a type-3 bilinear pairing. We first recall the standard ElGamal cryptosystem in $\mathbb{G}_1$, which is used by each ring member to generate their key pair and is specified as follows.

- Choose secret key $\mathsf{sk}_{\mathsf{PKE}} \in \mathbb{Z}_p^*$ and let public key $\mathsf{pk}_{\mathsf{PKE}} = g_1^{\mathsf{sk}_{\mathsf{PKE}}}$.
- To encrypt a message m with randomness r, run $\mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{PKE}}, m; r)$, which retruns $(c_1, c_2) = (g_1^r, \mathsf{pk}_{\mathsf{PKE}}^r \cdot m)$.
- To decrypt, run $\mathsf{PKE.Dec}(\mathsf{sk}_{\mathsf{PKE}}, (c_1, c_2))$, which returns $m = \frac{c_2}{c_1^{\mathsf{sk}_{\mathsf{PKE}}}}$.

Our new ElGamal variant, used by the signer during the signature algorithm to generate a fresh PKE key pair, is defined as follows.

- Generate fresh secret key $\mathsf{sk}_{\mathsf{Sign}} \in \mathbb{G}_1$ and define the fresh public key as $\mathsf{pk}_{\mathsf{Sign}} = e(\mathsf{sk}_{\mathsf{Sign}}, g_2)$.
- $\mathsf{PKE.Enc}(\mathsf{pk}_{\mathsf{Sign}}, m; r)$ returns $(c_1, c_2) = (g_2^r, \mathsf{pk}_{\mathsf{Sign}}^r \cdot m)$.

– $\mathsf{PKE.Dec}(\mathsf{sk}_{\mathsf{Sign}}, c)$ returns $m = \frac{c_2}{e(\mathsf{sk}_{\mathsf{Sign}}, c_1)}$.

Note that anyone can transform a ciphertext $(c_1, c_2) = (g_2^r, \mathsf{pk}_{\mathsf{Sign}}^r \cdot m)$ of this ElGamal variant into a standard ElGamal ciphertext in $\mathbb{G}_t$ by computing $(e(g_1, c_1), c_2)$.

We show the following result.

Theorem 1. *The proposed variant of ElGamal satisfies* IND-CPA *security under the Decisional Diffie-Hellman (DDH) assumption in* $\mathbb{G}_2$.

Proof. Assume that there exists a Probabilistic Polynomial Time (PPT) adversary $\mathcal{A}$ that breaks the IND-CPA security of our ElGamal variant with a non-negligible advantage $\epsilon_{\mathcal{A}}(\lambda)$. We show how to build a PPT adversary $\mathcal{B}$ that breaks the DDH assumption in $\mathbb{G}_2$ with a non-negligible advantage $\epsilon_{\mathcal{B}}(\lambda)$ (where λ is the security parameter used to generate $\mathbb{G}_2$).

$\mathcal{B}$ receives the DDH challenge $(X, Y, Z) = (g_2^x, g_2^y, g_2^{z_b})$ and picks $b' \xleftarrow{\$} \{0, 1\}$. It sets $\mathsf{pk}_{\mathsf{Sign}} \leftarrow e(g_1, X)$ and sends it to $\mathcal{A}$, which returns a pair of chosen plaintexts (m_0, m_1). $\mathcal{B}$ computes $c_1 \leftarrow Y$ and $c_2 = e(g_1, Z) \cdot m_{b'}$. It sends (c_1, c_2) to $\mathcal{A}$, which returns b''. If $b' = b''$, then $\mathcal{B}$ returns 0, else it returns 1.

We remark that $c_1 = Y = g_2^y$, and $c_2 = e(g_1, Z) \cdot m_{b'} = e(g_1, g_2)^{z_b} \cdot m_{b'}$. If $b = 0$, then $c_2 = e(g_1, g_2)^{x \cdot y} \cdot m_{b'} = e(g_1, g_2^x)^y \cdot m_{b'} = e(g_1, X)^y \cdot m_{b'} = \mathsf{pk}_{\mathsf{Sign}}^y \cdot m_{b'}$. In this case, the IND-CPA experiment is perfectly simulated for $\mathcal{A}$, so $\mathcal{A}$ returns $b'' = b'$ with the non-negligible advantage $\epsilon_{\mathcal{A}}(\lambda)$. If $b = 1$, then $c_2 = e(g_1, g_2)^{z_1} \cdot m_{b'}$ seems to be random from the point of view of $\mathcal{A}$. In this case, $\mathcal{A}$ has no information about b', so it returns $b'' = b'$ with probability $1/2$ (its advantage is null). Finally, $\epsilon_{\mathcal{B}}(\lambda) = \epsilon_{\mathcal{A}}(\lambda)/2$, so $\epsilon_{\mathcal{B}}(\lambda)$ is non-negligible, which concludes the proof. □

We will now show how to prove that an ElGamal ciphertext in $\mathbb{G}_1$ encrypts a secret key of our ElGamal variant in a zero-knowledge way. We consider the key pair of our ElGamal variant $\mathsf{sk}_{\mathsf{Sign}} \in \mathbb{G}_1$ and $\mathsf{pk}_{\mathsf{Sign}} = e(\mathsf{sk}_{\mathsf{Sign}}, g_2)$, and the ciphertext $(c_1, c_2) = (g_1^r, \mathsf{pk}_{\mathsf{PKE}}^r \cdot \mathsf{sk}_{\mathsf{Sign}}) \in \mathbb{G}_1^2$ which encrypts $\mathsf{sk}_{\mathsf{Sign}}$ with the public key $\mathsf{pk}_{\mathsf{PKE}}$.

We have to prove that $\mathsf{pk}_{\mathsf{Sign}} = e\left(\mathsf{PKE.Dec}(pp_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}, (c_1, c_2)), g_2\right)$. We have the following equivalences:

$$
\begin{aligned}
&\mathsf{pk}_{\mathsf{Sign}} = e\left(\mathsf{PKE.Dec}(pp_{\mathsf{PKE}}, \mathsf{sk}_{\mathsf{PKE}}, (c_1, c_2)), g_2\right) \\
\Leftrightarrow &\mathsf{pk}_{\mathsf{Sign}} = e\left(\frac{c_2}{c_1^{\mathsf{sk}_{\mathsf{PKE}}}}, g_2\right) = \frac{e\left(c_2, g_2\right)}{e\left(c_1^{\mathsf{sk}_{\mathsf{PKE}}}, g_2\right)} = \frac{e\left(c_2, g_2\right)}{e\left(g_1^{r \cdot \mathsf{sk}_{\mathsf{PKE}}}, g_2\right)} = \frac{e\left(c_2, g_2\right)}{e\left(\mathsf{pk}_{\mathsf{PKE}}, g_2\right)^r} \\
\Leftrightarrow &e\left(\mathsf{pk}_{\mathsf{PKE}}, g_2\right)^r = \left(\frac{e\left(c_2, g_2\right)}{\mathsf{pk}_{\mathsf{Sign}}}\right)
\end{aligned}
$$

On the other hand, we have $e(c_1, g_2) = e(g_1^r, g_2) = e(g_1, g_2)^r$.

Finally, in order to prove that the ElGamal ciphertext in $\mathbb{G}_1$ encrypts the secret key of our ElGamal variant in zero-knowledge, we have to prove the following relation, knowing r:

$$
\mathsf{NIZK}\left\{r : e(c_1, g_2) = e(g_1, g_2)^r \wedge e\left(\mathsf{pk}_{\mathsf{PKE}}, g_2\right)^r = \left(\frac{e\left(c_2, g_2\right)}{\mathsf{pk}_{\mathsf{Sign}}}\right)\right\} \tag{1}
$$

This is a proof of discrete logarithm equality in $\mathbb{G}_t$. This zero-knowledge proof can be instantiated with the Fiat-Shamir transform [12] on the variant of the Schnorr protocol given in [9].

4.2 Discussion

We propose to use the above encryptions and NIZK proof to build an *efficient* RTR signature scheme following the generic construction in [13]. We recall that in a type 3 pairing, the DDH assumption holds in $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_t$, which implies that any construction based on the discrete logarithm assumption, the computational Diffie-Hellman assumption, or the decisional Diffie-Hellman assumption remains secure in each of these groups. The construction of Fraser and Quaglia uses only discrete logarithm-based building blocks, so it remains secure in our new pairing setup. Moreover, in order to prove relations among different elements of the signature, this construction uses Schnorr-based proofs of discrete logarithm relation and discrete logarithm knowledge, which work in any group of prime order, even when the relation is proved over different groups of the same order. Since our new encryption instantiation keep the structure of ElGamal, the other zero-knowledge proofs can be instantiated as in [13].

The NIZK proof outlined above is more efficient than the NIZK used in the instantiation in [13]. More specifically, the above NIZK proof requires a constant number of group exponentiations and pairings[5] (2 and 3, respectively) to prove. Similarly, verification of the NIZK proof requires 4 group exponentiations and 3 pairings. The size of the proof is also constant in size: it consists of 2 group elements and 1 field element. Comparatively, the size of the NIZK proof used in Fraser and Quaglia's instantiation, and the computational costs associated with proving and verification, are linear in $|R| \cdot \lambda$ (where λ is the security parameter). With respect to other costs associated with signing and verification, the two instantiations are identical, as shown above. As such, with respect to signature generation and verification, our instantiation has linear space and time complexity in the size of the ring. Therefore, our approach implies that the generic construction can be instantiated more efficiently than originally proposed, *i.e.*, avoiding the linear increase in the security parameter.

To conclude, as a consequence of the security proofs for the generic construction in [13], our pairing instantiation is secure if our new ElGamal variant satisfies IND-CPA security [15] and our zero-knowledge proof of correct encryption (Equation 1) satisfies completeness, knowledge soundness and zero-knowledge, as defined in [16]. As such, our instantiation satisfies the RTR signature security model.

[5] According to [7], type 3 pairings are more efficient than type 1 and 2 pairings, and the computation time of a type 3 pairing is equivalent to 4 exponentiations for the best implementation.

5 A New RTR Signature Construction

In this section, we present a new RTR signature construction. We describe our protocol and present an instantiation. We conclude this section with a security analysis of our protocol and a brief discussion of its efficiency.

5.1 Description of Our Protocol

We outline our protocol following the syntax of an RTR scheme introduced in Definition 1.

Setup and Key Generation. Our construction uses ElGamal-based keys. Each ElGamal encryption key ek is provided together with a proof of knowledge π of the corresponding secret key sk. We will see why these proofs of knowledge are required later in this section. The public key is the pair $\mathsf{pk} = (\mathsf{ek}, \pi)$ and the secret key is sk. We use the part ek of the user public key as their identity.

$\mathsf{Setup}(1^\lambda)$**:** Generates a prime order group setup $pp = (\mathbb{G}, p, g)$ such that the Decisional Diffie-Hellman assumption holds in $\mathbb{G}$.

$\mathsf{T.KGen}(pp)$**:** Picks $\mathsf{sk_T} \xleftarrow{\$} \mathbb{Z}_p^*$, sets $\mathsf{ek_T} \leftarrow g^{\mathsf{sk_T}}$, sets $\pi_\mathsf{T} \leftarrow \mathsf{NIZK}\left\{\mathsf{sk_T} : \mathsf{ek_T} = g^{\mathsf{sk_T}}\right\}$ and outputs $\mathsf{pk_T} \leftarrow (\mathsf{ek_T}, \pi_\mathsf{T})$.

$\mathsf{U.KGen}(pp)$**:** Picks $\mathsf{sk_U} \xleftarrow{\$} \mathbb{Z}_p^*$, sets $\mathsf{ek_U} \leftarrow g^{\mathsf{sk_U}}$, sets $\pi_\mathsf{U} \leftarrow \mathsf{NIZK}\left\{\mathsf{sk_U} : \mathsf{ek_U} = g^{\mathsf{sk_U}}\right\}$ and outputs $\mathsf{pk_U} \leftarrow (\mathsf{ek_U}, \pi_\mathsf{U})$.

Signature Generation and Verification. The idea of the signature is to separate the public key ek of the signer into two shares S_1 and S_2 such that $S_1 \cdot S_2 = \mathsf{ek}$. The signer picks a coin α at random and uses it to encrypt (using ElGamal) S_2 for each public encryption key ek_i in the ring, outputting $|R|$ ciphertexts denoted c_i. The signer then encrypts S_1 for the tracer encryption key $\mathsf{ek_T}$, outputting ciphertext c. The signer then proves that the ring members' ciphertexts encrypt the same message in zero-knowledge. Note that due to the homomorphic properties of ElGamal, each $c_i \cdot c$ encrypts $S_1 \cdot S_2 = \mathsf{ek}$. Finally, the signer signs the message using a signature of knowledge that proves in zero-knowledge that it knows the secret key sk_i for a secret index i (which is its own secret key $\mathsf{sk}_i = \mathsf{sk_U}$) that decrypts $c_i \cdot c$ on the message $\mathsf{ek}_i = \mathsf{ek_U}$.

$\mathsf{Sign}(pp, \mathsf{sk_U}, \mathsf{pk_T}, m, R)$**:** Parses $\mathsf{pk_T}$ as $(\mathsf{ek_T}, \pi_\mathsf{T})$. Sets $n \leftarrow |R|$, parses R as $\{\mathsf{pk}_i\}_{i=1}^n$ and each pk_i as (ek_i, π_i). Verifies each π_i (this step preempts a subtle attack on anonymity that we will detail later). If there are two indices i and j such that $\mathsf{pk}_i \neq \mathsf{pk}_j$ and $\mathsf{ek}_i = \mathsf{ek}_j$, or if there is no index i such that $\mathsf{pk_U} = \mathsf{pk}_i$, then it aborts and returns the failure symbol $\perp$. Picks $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ and sets $h \leftarrow g^\alpha$. Picks $S_1 \xleftarrow{\$} \mathbb{G}$ and sets $S_2 \leftarrow \mathsf{ek_U}/S_1$. Note that S_1 and S_2 are two shares of the secret identity $\mathsf{ek_U} = S_1 \cdot S_2$. Sets $c \leftarrow \mathsf{ek_T}^\alpha \cdot S_1$. For each $i \in [\![n]\!]$:

- Sets $c_i \leftarrow \mathsf{ek}_i^\alpha \cdot S_2$.
- If $i > 1$, sets $\pi'_i \leftarrow \mathsf{NIZK}\left\{\alpha : \left(h = g^\alpha \wedge \left(\frac{c_i}{c_{i-1}}\right) = \left(\frac{\mathsf{ek}_i}{\mathsf{ek}_{i-1}}\right)^\alpha\right)\right\}$, else $\pi'_i \leftarrow \perp$.

The proofs π'_i ensure that each ElGamal ciphertext (h, c_i) encrypts the same message. Note that $(h, c \cdot c_i)$ is the ElGamal encryption of $S_1 \cdot S_2 = \mathsf{ek_U}$ for the public key $(\mathsf{ek_T} \cdot \mathsf{ek}_i)$. Sets $M \leftarrow (pp, \mathsf{pk_T}, m, R, h, c, (c_i, \pi'_i)_{i=1}^n)$, and sets $\sigma_M \leftarrow \mathsf{SoK}_M \left\{(\alpha, \mathsf{sk_U}) : \bigvee_{i=1}^n \left(h = g^\alpha \wedge \left(\frac{c \cdot c_i}{\mathsf{ek}_i}\right) = (\mathsf{ek_T} \cdot \mathsf{ek}_i)^\alpha \wedge \mathsf{ek}_i = g^{\mathsf{sk_U}}\right)\right\}$. The signature of knowledge σ_M ensures that $(h, c \cdot c_i)$ is an ElGamal encryption of one ek_i, and that the signer knows the secret key corresponding to ek_i, which means that ek_i is the identity of the signer. Finally, the algorithm returns $\sigma = (h, c, (c_i, \pi'_i)_{i=1}^n, \sigma_M)$.

$\mathsf{Verify}(pp, \mathsf{pk_T}, m, R, \sigma)$**:** Verify each π_i and σ_M.

Note that, if the keys are honestly generated, the probability that the signature aborts because two encryption keys ek_i and ek_j are equal is negligible.

Report and Trace. To report a signature, a user decrypts the ciphertext c_i that corresponds to their public key in order to learn S_2, and proves the correctness of the decryption using a zero-knowledge proof. To trace the signature, the tracer decrypts c in order to learn S_1, proves the correctness of the decryption using a zero-knowledge proof, and returns the identity that corresponds to the encryption key $\mathsf{ek} = S_1 \cdot S_2$.

$\mathsf{Report}(pp, \mathsf{pk_T}, \mathsf{sk_U}, m, R, \sigma)$**:** Verifies the signature σ. Sets $n \leftarrow |R|$, parses R as $\{\mathsf{pk}_i\}_{i=1}^n$ and each pk_i as (ek_i, π_i). Let j be the index that verifies $\mathsf{pk}_j = (\mathsf{ek_U}, \pi_j)$. Parses σ as $(h, c, (c_i, \pi'_i)_{i=1}^n, \sigma_M)$. Sets $S_2 \leftarrow c_j / h^{\mathsf{sk_U}}$ and $\pi_{\mathtt{Rep}} \leftarrow \mathsf{NIZK}\left\{\mathsf{sk_U} : \bigvee_{i=1}^n \left(\left(\frac{c_i}{S_2}\right) = h^{\mathsf{sk_U}} \wedge \mathsf{ek}_i = g^{\mathsf{sk_U}}\right)\right\}$. The proofs $\pi_{\mathtt{Rep}}$ ensures that one (h, c_i) encrypts S_2. This algorithm returns $\mathtt{Rep} \leftarrow (S_2, \pi_{\mathtt{Rep}})$

$\mathsf{Trace}(pp, sk_\mathsf{T}, m, R, \sigma, \mathtt{Rep})$**:** Verifies the signature σ. Parses $\mathtt{Rep}$ as $(S_2, \pi_{\mathtt{Rep}})$ and σ as $(h, c, (c_i, \pi'_i)_{i=1}^n, \sigma_M)$. Verifies the proof $\pi_{\mathtt{Rep}}$. Sets $S_1 \leftarrow c / h^{\mathsf{sk_T}}$, and $\pi_{\rho_t} \leftarrow \mathsf{NIZK}\left\{\mathsf{sk_T} : \left(\frac{c}{S_1}\right) = h^{\mathsf{sk_T}} \wedge \mathsf{ek_T} = g^{\mathsf{sk_T}}\right\}$.
The proof π_{ρ_t} ensures that one (h, c) encrypts S_1. This algorithm returns $\rho_t \leftarrow (S_1, \pi_{\rho_t})$

$\mathsf{VerTrace}(pp, \mathsf{pk_T}, m, R, \sigma, \mathsf{pk_U}, \mathtt{Tr}, \rho_t)$**:** Sets $n \leftarrow |R|$, parses R as $\{\mathsf{pk}_i\}_{i=1}^n$ and each pk_i as (ek_i, π_i). Verifies $\pi_{\mathtt{Rep}}$ and π_{ρ_t}. If one of these proofs is not valid, then it returns the failure symbol $\perp$, else it returns the key pk_i that verifies $\mathsf{ek}_i = S_1 \cdot S_2$.

5.2 Instantiation

In the following, we propose an instantiation for each of the proofs and signatures of knowledge used in our protocol. The proof $\mathsf{NIZK}\{x : h = g^x\}$ used in π_U and π_T can be instantiated with the Fiat-Shamir transform on the Schnorr protocol [28]. The proof $\mathsf{NIZK}\{x : h_1 = g_1^x \wedge h_2 = g_2^x\}$ used in π_{ρ_t} and each π'_i can be instantiated with the Fiat-Shamir transform on the variant of the Schnorr protocol given in [9]. The proof $\mathsf{NIZK}\left\{x : \bigvee_{i=1}^n \left(h_{i,1} = g_{i,1}^x \wedge h_{i,2} = g_{i,2}^x\right)\right\}$ used in $\pi_{\mathtt{Rep}}$ can be instantiated with the Cramer-Damgård-Schoenmakers transform [10] (which transforms a zero-knowledge proof of a statement into a

zero-knowledge proof of 1-out-of-n statements) and the Fiat-Shamir transform applied on the variant of the Schnorr protocol given in [9]. Finally, the proof $\mathsf{NIZK}\left\{(x,y) : \bigvee_{i=1}^{n} \left(h_{i,1} = g_{i,1}^{x} \wedge h_{i,2} = g_{i,2}^{x} \wedge h_{i,3} = g_{i,3}^{y}\right)\right\}$ used in σ_M can be instantiated with the Fiat-Shamir transform and the Cramer-Damgård-Schoenmakers transform [10] applied on the successive executions of the Schnorr protocol [28] and the variant of the Schnorr protocol given in [9]. To transform this proof into a signature of knowledge, it suffices to add the message to the hashed elements during the creation of the challenge (this method works with any protocol resulting from the Fiat-Shamir transform [6]).

All these proofs of knowledge use only group operations and do not require any specific tool to be instantiated. The non-interactive version of the proofs and the signature of knowledge require a hash function modeled by a random oracle.

5.3 Security Analysis

Our new construction satisfies the security properties for an RTR signature scheme and, as such, we obtain Theorem 2. The formal proof of this theorem is given in Appendix C and we informally explain why these properties hold here.

Theorem 2. *Our protocol instantiated with extractable and zero-knowledge proofs and signatures of knowledge is unforgeable, anonymous, non-frameable, trace sound, and reporter anonymous under the Decisional Diffie-Hellman assumption in the standard model.*

Unforgeability: To forge a signature, an adversary must forge a signature of knowledge σ_M, which requires the knowledge of one of the secret keys of the ring, which is the discrete logarithm of one of the public encryption keys. If an adversary produces such a signature, then the extractor of the signature of knowledge can be used to break the discrete logarithm assumption (which is hard under the Decisional Diffie-Hellman assumption).

Anonymity: To deduce the identity of the signer, the share S_1 of the signer identity is required by the adversary. This share is encrypted using the ElGamal encryption on the honest tracer public key. Thus, breaking the anonymity is at least as difficult as breaking the $\mathsf{IND\text{-}CPA}$ security of ElGamal, which depends on the Decisional Diffie-Hellman assumption.

Trace soundness: The proofs and signatures of knowledge ensure that the identity of the signer ek is actually $S_1 \cdot S_2$ (from σ_M), each c_i encrypts the same S_2 (from π'_i), the reporter returns S_2 (from $\pi_{\mathtt{Rep}}$), and the tracer returns S_1 (from π_{ρ_t}). If an adversary is able to report the same signature for two different identities, then it forges a proof on a false statement that cannot be correctly extracted, which contradicts the extractability.

Non-frameability: As it is shown for the trace soundness, the proofs ensure that the report and trace mechanism are sound. Thus, to attack non-frameability the adversary must produce a fresh valid and traceable signature for an honest user. As for unforgeability, such an adversary can be used to extract the discrete logarithm of the public encryption key of an honest user, which is hard under the Decisional Diffie-Hellman assumption.

Reporter anonymity: Each reporter returns the same S_2 (according to the proofs of knowledge that we use in the protocol), and a zero-knowledge proof that gives no information about their identity. Therefore, an adversary cannot deduce the identity of the reporter.

The role of the zero-knowledge proofs on the public keys. We recall that each public key is associated with a proof of correctness, and that these proofs are verified before each signature. In what follows, we will show that this mechanism avoids a subtle attack on anonymity. Assume that the users do not prove the knowledge of their secret keys (*i.e.* $\mathsf{pk} = \mathsf{ek}$). In this case, an attacker $\mathcal{A}$ can break the anonymity of our construction using the following attack. $\mathcal{A}$ chooses the public keys $(\mathsf{pk}_0, \mathsf{pk}_1)$ of two honest users, picks $\mathsf{sk}_2 \xleftarrow{\$} \mathbb{Z}_p^*$, sets $\mathsf{ek}_2 \leftarrow g^{\mathsf{sk}_2}$, and sets $\mathsf{pk}_2 \leftarrow \mathsf{ek}_2$. $\mathcal{A}$ then picks $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$, sets $\mathsf{ek}_3 \leftarrow \mathsf{ek}_\mathsf{T}^\gamma$, and sets $\mathsf{pk}_3 \leftarrow \mathsf{ek}_3$. $\mathcal{A}$ chooses a message m, sets $R \leftarrow \{\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{pk}_3\}$ sends $(m, R, \mathsf{pk}_0, \mathsf{pk}_1, st)$ to the challenger, and receives a signature $\sigma = (h, c, (c_i, \pi_i')_{i=1}^n, \sigma_M)$. Since σ has been generated correctly, we have that $c = \mathsf{ek}_\mathsf{T}^\alpha \cdot S_1$ and $\forall\ i, c_i = \mathsf{ek}_i^\alpha \cdot S_2$ (where α denotes the discrete logarithm of h). $\mathcal{A}$ computes $S_2 \leftarrow c_2/h^{\mathsf{sk}_2}$ and $S_1' \leftarrow c/\left(\frac{c_3}{S2}\right)^{\frac{1}{\gamma}}$. If $S_1' \cdot S_2 = \mathsf{ek}_0$, then $\mathcal{A}$ returns 0, else if $S_1' \cdot S_2 = \mathsf{ek}_1$, then $\mathcal{A}$ returns 1. We observe that:

$$S_1' = \frac{c}{\left(\frac{c_3}{S2}\right)^{\frac{1}{\gamma}}} = \frac{\mathsf{ek}_\mathsf{T}^\alpha \cdot S_1}{\left(\frac{\mathsf{ek}_3^\alpha \cdot S_2}{S2}\right)^{\frac{1}{\gamma}}} = \frac{\mathsf{ek}_\mathsf{T}^\alpha \cdot S_1}{(\mathsf{ek}_3^\alpha)^{\frac{1}{\gamma}}} = \frac{\mathsf{ek}_\mathsf{T}^\alpha \cdot S_1}{((\mathsf{ek}_\mathsf{T}^\gamma)^\alpha)^{\frac{1}{\gamma}}} = \frac{\mathsf{ek}_\mathsf{T}^\alpha \cdot S_1}{\mathsf{ek}_\mathsf{T}^{\alpha \cdot \frac{\gamma}{\gamma}}} = \frac{\mathsf{ek}_\mathsf{T}^\alpha \cdot S_1}{\mathsf{ek}_\mathsf{T}^\alpha} = S_1.$$

Thus, $S_1' \cdot S_2$ gives the identity of the signer with probability 1.

Efficiency of our protocol and comparison. Similarly to our instantiation in Section 4, the protocol presented in Section 5.1, when instantiated as in Section 5.2, has space and time complexity that is linear in the size of the ring. More explicitly, a signature can be computed with $11|R|-3$ group exponentiations and verified with $10|R| - 4$ group exponentiations. A signature consists of $6|R|$ group elements and $4|R|-2$ field elements. On the other hand, in our instantiation in Section 4, a signature can be computed with $5|R| + 21$ group exponentiations and 4 pairings, and verified with $3|R| + 23$ group exponentiations and 3 pairings. A signature consists of $2|R| + 20$ group elements and $|R| + 7$ field elements. Thus, our instantiation from Section 4 requires less group exponentiations, moreover, it generates reporter tokens of constant size, while the size of the tokens grows linearly with the number of users in the new construction. In return, our new construction can be instantiated with any prime order group, including pairing-free groups based on elliptic curves, which are known to optimize the size of the group elements and the computation cost of the operations for an equivalent level of security.

6 Concluding Remarks

We introduced a new instantiation of an RTR signature scheme that follows the generic construction in [13]. Our instantiation has space and time complexity

linear in the size of the ring. Consequently, our instantiation significantly increases the efficiency of the construction in [13], but requires pairings. We also introduce a new RTR signature construction with similar complexity that does not require pairings and can be instantiated with any prime order group. In return, our construction requires more group exponentiations than our instantiation of [13]. An interesting open question is whether it is possible to design an RTR signature that simultaneously reaps the benefits of our instantiation and new construction. That is, we ask, is it possible to design an RTR signature that is (at least) as efficient as our instantiation of the construction from [13] *and* can be instantiated with any group?

References

1. Venkat Arun, Aniket Kate, Deepak Garg, Peter Druschel, and Bobby Bhattacharjee. Finding safety in numbers with secure allegation escrows. In *The Network and Distributed System Security Symposium*. Internet Society, 2020.
2. Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In *Theory of Cryptography Conference*, pages 60–79. Springer, 2006.
3. Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A Kroll, and Edward W Felten. Mixcoin: Anonymity for bitcoin with accountable mixes. In *International Conference on Financial Cryptography and Data Security*, pages 486–504. Springer, 2014.
4. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on ddh. In *European Symposium on Research in Computer Security*, pages 243–265. Springer, 2015.
5. Jan Camenisch and Markus Stadler. Proof systems for general statements about discrete logarithms. In *Technical Report No. 260*. Dept. of Computer Science, ETH Zurich, 1997.
6. Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, pages 78–96, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
7. Sanjit Chatterjee, Alfred Menezes, and Francisco Rodrguez-Henrquez. On instantiating pairing-based protocols with elliptic curves of embedding degree one. *IEEE Transactions on Computers*, 66(6):1061–1070, 2017.
8. David Chaum and Eugène van Heyst. Group signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 257–265. Springer, 1991.
9. David Chaum and Torben P. Pedersen. Wallet databases with observers. In *Annual international cryptology conference*, pages 89–105. Springer, 1992.
10. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology — CRYPTO*. Springer, 1994.
11. Prastudy Fauzi, Sarah Meiklejohn, Rebekah Mercer, and Claudio Orlandi. Quisquis: A new design for anonymous cryptocurrencies. In *International conference on the theory and application of cryptology and information security*, pages 649–678. Springer, 2019.

12. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — CRYPTO*. Springer, 1987.
13. Ashley Fraser and Elizabeth A Quaglia. Report and trace ring signatures. In *International Conference on Cryptology and Network Security*, pages 179–199. Springer, 2021.
14. Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In *International Workshop on Public Key Cryptography*, pages 181–200. Springer, 2007.
15. Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC. ACM, 1982.
16. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for np. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 339–358. Springer, 2006.
17. Alejandro Hevia and Ilana Mergudich-Thal. Implementing secure reporting of sexual misconduct-revisiting whotoo. In *International Conference on Cryptology and Information Security in Latin America*, pages 341–362. Springer, 2021.
18. Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Traceable signatures. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 571–589. Springer, 2004.
19. Markulf Kohlweiss and Ian Miers. Accountable metadata-hiding escrow: A group signature case study. *Proc. Priv. Enhancing Technol.*, 2015(2):206–221, 2015.
20. Benjamin Kuykendall, Hugo Krawczyk, and Tal Rabin. Cryptography for# metoo. *Proc. Priv. Enhancing Technol.*, 2019(3):409–429, 2019.
21. Joseph K Liu, Victor K Wei, and Duncan S Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In *Australasian Conference on Information Security and Privacy*, pages 325–335. Springer, 2004.
22. Gregory Maxwell. Coinjoin: Bitcoin privacy for the real world. In *Post on Bitcoin forum*, volume 3, page 110, 2013.
23. Shen Noether, Adam Mackenzie, et al. Ring confidential transactions. *Ledger*, 1:1–18, 2016.
24. Anjana Rajan, Lucy Qin, David W Archer, Dan Boneh, Tancrede Lepoint, and Mayank Varia. Callisto: A cryptographic approach to detecting serial perpetrators of sexual misconduct. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, pages 1–4, 2018.
25. Ronald L Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *International conference on the theory and application of cryptology and information security*, pages 552–565. Springer, 2001.
26. Yusuke Sakai, Keita Emura, Goichiro Hanaoka, Yutaka Kawai, Takahiro Matsuda, and Kazumasa Omote. Group signatures with message-dependent opening. In *International Conference on Pairing-Based Cryptography*, pages 270–294. Springer, 2012.
27. Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE symposium on security and privacy*, pages 459–474. IEEE, 2014.
28. Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 1991.
29. Markus Stadler. Publicly verifiable secret sharing. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 190–199. Springer, 1996.

30. Nirvan Tyagi, Ian Miers, and Thomas Ristenpart. Traceback for end-to-end encrypted messaging. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 413–430, 2019.
31. Shouhuai Xu and Moti Yung. Accountable ring signatures: A smart card approach. In *Smart Card Research and Advanced Applications VI*, pages 271–286. Springer, 2004.

A Building Blocks

We recall the formal definition of the building blocks we need in our construction, as well as some well-known properties.

Definition 2 (Discrete Logarithm assumption (DL)). *Let $\mathbb{G}$ be a group of prime order p and $g \in \mathbb{G}$ be a generator. The discrete logarithm assumption states that there is no PPT algorithm that takes a random element $X \xleftarrow{\$} \mathbb{G}$ as input and that returns $x \in \mathbb{Z}_p^*$ such that $g^x = X$.*

Definition 3 (Decisional Diffie-Hellman assumption (DDH)). *Let $\mathbb{G}$ be a group of prime order p and $g \in \mathbb{G}$ be a generator. Let x, y, and z_0 be three random elements picked in the uniform distribution on $\mathbb{Z}_p^*$. We set $z_1 = x \cdot y$. The Decisional Diffie-Hellman assumption states that there is no PPT algorithm that distinguishes (g^x, g^y, g^{z_0}) from (g^x, g^y, g^{z_1}) with non-negligible advantage (*i.e. *with a probability significantly different from $1/2$).*

Lemma 1. *The DL assumption holds under the DDH assumption.*

Definition 4 (Non-Interactive Zero-Knoledge Proof (NIZKP)). *Let $\mathcal{R}$ be a binary relation and let $\mathcal{L}$ be a language such that $s \in \mathcal{L} \Leftrightarrow (\exists w, (s, w) \in \mathcal{R})$. A* non-interactive ZKP *(NIZKP) for the language $\mathcal{L}$ is a couple of algorithms $(\mathsf{NIZK}, \mathsf{NIZK.Verify})$ such that:*
$\mathsf{NIZK}\{w : (s, w) \in \mathcal{R}\}$**.** *This algorithm outputs a proof π.*
$\mathsf{NIZK.Verify}(s, \pi)$**.** *This algorithm outputs a bit b.*
A NIZKP proof has the following properties:

Corectness. *For any s, w, $\mathcal{R}$ such that $(s, w) \in \mathcal{R}$, and $\pi \leftarrow \mathsf{NIZK}\{w : (s, w) \in \mathcal{R}$, $\mathsf{NIZK.Verify}(s, \pi)$ returns 1.*

Soundness. *There is no polynomial time adversary $\mathcal{A}$ such that $\mathcal{A}(\mathcal{L})$ outputs (s, π) such that $\mathsf{NIZK.Verify}(s, \pi) = 1$ and $s \notin \mathcal{L}$ with non-negligible probability.*

Extractability. *There exists a PPT knowledge extractor Ext and a negligible function ϵ_{NIZK} such that for any algorithm $\mathcal{A}^{\mathsf{Sim}(\cdot)}(\lambda)$ having access to a simulator that forges signatures for chosen statement and that outputs a fresh pair (s, π) with $\mathsf{NIZK.Verify}(s, \pi) = 1$, the extractor $\mathsf{Ext}^{\mathcal{A}}(\lambda)$ outputs w such that $(s, w) \in \mathcal{R}$ having access to $\mathcal{A}(\lambda)$ with probability at least $1 - \epsilon_{\mathsf{NIZK}}(\lambda)$.*

Zero-knowledge. *A proof π* leaks no information*, i.e., there exists a polynomial time algorithm Sim (called the* simulator*) such that $\mathsf{NIZK}\{w : (s, w) \in \mathcal{R}\}$ and $\mathsf{Sim}(s)$ follow the same probability distribution.*

Signatures of Knowledge (SoK) are defined in a similar way, except that each proof is associated with a message m:
$\mathsf{SoK}_m\{w : (s, w) \in \mathcal{R}\}$**.** *This algorithm outputs a proof π.*
$\mathsf{SoK.Verify}(m, s, \pi)$**.** *This algorithm outputs a bit b.*

Definition 5 (ElGamal Encryption). *The ElGamal Encryption on* $\mathsf{pp} = (G, g, p)$ *where* $\mathbb{G} = \langle g \rangle$ *is a group of prime order p is a public key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *defined as follows:*

$\mathsf{Gen}(\mathsf{pp})$**:** *picks* $\mathsf{sk} \xleftarrow{\$} \mathbb{Z}_p^*$*, sets* $\mathsf{pk} \leftarrow g^{\mathsf{sk}}$ *and returns* $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{End}(\mathsf{pp}, \mathsf{pk}, m)$**:** *picks* $r \xleftarrow{\$} \mathbb{Z}_p^*$*, sets* $c_1 \leftarrow g^r$*,* $c_2 \leftarrow \mathsf{pk}^r \cdot m$*, and returns* $c \leftarrow (c_1, c_2)$.

$\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}, c)$**:** *returns* $m \leftarrow c_2 / c_1^{\mathsf{sk}}$.

Definition 6 (IND-CPA security). *An encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *on a public parameter* pp *is said to be IND-CPA when there exists a negligible function $\epsilon(\lambda)$ such that for any pair of PPT algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$:*

$$\left| \Pr\left[\begin{array}{l} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp});\ (m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(\mathsf{pp},\mathsf{pk}); \\ b \xleftarrow{\$} \{0,1\};\ c \leftarrow \mathsf{Enc}(\mathsf{pp},\mathsf{pk},m_b);\ b' \leftarrow \mathcal{A}_2(\mathsf{pp},\mathsf{pk},\mathsf{st},c); \end{array} : \ b' = b \right] - \frac{1}{2} \right| \leq \epsilon(\lambda).$$

Lemma 2. *ElGamal is IND-CPA secure under the DDH assumption.*

B Security Model for Report and Trace Ring Signatures

Throughout this section, λ denotes the security parameter.

Definition 7 (Correctness). *An* RTR *signature is correct if, for any* $n = \mathsf{poly}(\lambda)$*, $j \in [n]$ and message m, there exists a negligible function ϵ such that,*

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda); \\ (\mathsf{pk}_\mathsf{T}, sk_\mathsf{T}) \leftarrow \mathsf{T.KGen}(pp); \\ \textbf{for } i = 1,\ldots,n : (\mathsf{pk}_{\mathsf{U}_i}, sk_{\mathsf{U}_i}) \leftarrow \mathsf{U.KGen}(pp); \\ R = \{pk_{\mathsf{U}_1},\ldots,\mathsf{pk}_{\mathsf{U}_n}\}; \\ \sigma \leftarrow \mathsf{Sign}(pp, sk_{\mathsf{U}_j}, \mathsf{pk}_\mathsf{T}, m, R); \\ b \leftarrow \mathsf{Verify}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma) \end{array} : \ b = 1 \right] \geq 1 - \epsilon(\lambda).$$

Definition 8 (Trace correctness). *An* RTR *signature satisfies trace correctness if for any* $n = \mathsf{poly}(\lambda)$*, $j, k \in [n]$ where $j \neq k$, and message m, there exists a negligible function ϵ such that,*

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda); \\ (\mathsf{pk}_\mathsf{T}, sk_\mathsf{T}) \leftarrow \mathsf{T.KGen}(pp); \\ \textbf{for } i = 1,\ldots,n : (\mathsf{pk}_{\mathsf{U}_i}, sk_{\mathsf{U}_i}) \leftarrow \mathsf{U.KGen}(pp); \\ R = \{pk_{\mathsf{U}_1},\ldots,\mathsf{pk}_{\mathsf{U}_n}\}; \\ \sigma \leftarrow \mathsf{Sign}(pp, sk_{\mathsf{U}_j}, \mathsf{pk}_\mathsf{T}, m, R); \\ \mathtt{Rep} \leftarrow \mathsf{Report}(pp, \mathsf{pk}_\mathsf{T}, sk_{\mathsf{U}_k}, m, R, \sigma); \\ (\mathsf{pk}_\mathsf{U}, \mathtt{Tr}, \rho_t) \leftarrow \mathsf{Trace}(pp, sk_\mathsf{T}, m, R, \sigma, \mathtt{Rep}); \\ b \leftarrow \mathsf{VerTrace}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_\mathsf{U}, \mathtt{Tr}, \rho_t) \end{array} : \ b = 1 \right] \geq 1 - \epsilon(\lambda).$$

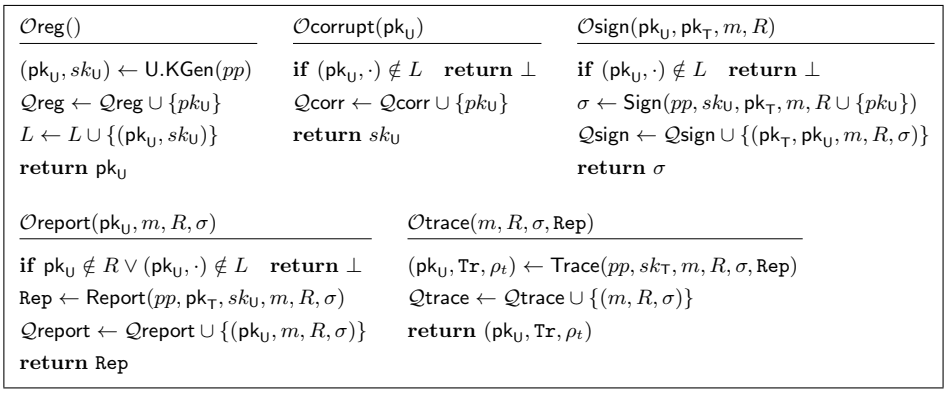

Fig. 1: Oracles for the report and trace ring signature security model from [13].

Definition 9 (Anonymity). *An* RTR *signature is anonymous with respect to adversarially generated keys if, for any probabilistic, polynomial-time (PPT) adversary $\mathcal{A}$, there exists a negligible function ϵ such that,*

$$\left| \Pr \left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda); \\ L,\ \mathcal{Q}\mathsf{reg},\ \mathcal{Q}\mathsf{corr},\ \mathcal{Q}\mathsf{sign},\ \mathcal{Q}\mathsf{report},\ \mathcal{Q}\mathsf{trace} \leftarrow \emptyset; \\ (\mathsf{pk}_\mathsf{T}, sk_\mathsf{T}) \leftarrow \mathsf{T.KGen}(pp); \\ (m, R, \mathsf{pk}_{\mathsf{U}_0}, \mathsf{pk}_{\mathsf{U}_1}, st) \\ \qquad \leftarrow \mathcal{A}^{\mathcal{O}\mathsf{reg}, \mathcal{O}\mathsf{corrupt}, \mathcal{O}\mathsf{sign}, \mathcal{O}\mathsf{report}, \mathcal{O}\mathsf{trace}}(pp, \mathsf{pk}_\mathsf{T}); \\ b \leftarrow \{0,1\}; \\ \sigma \leftarrow \mathsf{Sign}(pp, sk_{\mathsf{U}_b}, \mathsf{pk}_\mathsf{T}, m, R \cup \{pk_{\mathsf{U}_0}, \mathsf{pk}_{\mathsf{U}_1}\}); \\ b' \leftarrow \mathcal{A}^{\mathcal{O}\mathsf{reg}, \mathcal{O}\mathsf{corrupt}, \mathcal{O}\mathsf{sign}, \mathcal{O}\mathsf{report}, \mathcal{O}\mathsf{trace}}(\sigma, st) \end{array} : \begin{array}{l} b' = b \\ \wedge\ (m, R, \sigma) \notin \mathcal{Q}\mathsf{trace} \\ \wedge\ pk_{\mathsf{U}_0} \in \mathcal{Q}\mathsf{reg} \backslash \mathcal{Q}\mathsf{corr} \\ \wedge\ pk_{\mathsf{U}_1} \in \mathcal{Q}\mathsf{reg} \backslash \mathcal{Q}\mathsf{corr} \end{array} \right] - \frac{1}{2} \right| \leq \epsilon(\lambda).$$

Definition 10 (Unforgeability). *An* RTR *signature scheme is unforgeable if, for any PPT adversary $\mathcal{A}$, there exists a negligible function ϵ such that,*

$$\Pr \left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda); \\ L,\ \mathcal{Q}\mathsf{reg},\ \mathcal{Q}\mathsf{corr},\ \mathcal{Q}\mathsf{sign},\ \mathcal{Q}\mathsf{report} \leftarrow \emptyset; \\ (\mathsf{pk}_\mathsf{T}, m, R, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}\mathsf{reg}, \mathcal{O}\mathsf{corrupt}, \mathcal{O}\mathsf{sign}, \mathcal{O}\mathsf{report}}(pp) \end{array} : \begin{array}{l} \mathsf{Verify}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma) = 1 \\ \wedge\ R \subseteq \mathcal{Q}\mathsf{reg} \backslash \mathcal{Q}\mathsf{corr} \\ \wedge\ (\mathsf{pk}_\mathsf{T}, \cdot, m, R, \sigma) \notin \mathcal{Q}\mathsf{sign} \end{array} \right] \leq \epsilon(\lambda).$$

Definition 11 (Non-frameability). *An* RTR *signature is non-frameable if for any PPT adversary $\mathcal{A}$, there exists a negligible function ϵ such that,*

$$\Pr \left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda); \\ L,\ \mathcal{Q}\mathsf{reg},\ \mathcal{Q}\mathsf{corr},\ \mathcal{Q}\mathsf{sign},\ \mathcal{Q}\mathsf{report} \leftarrow \emptyset; \\ (\mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_\mathsf{U}, \mathtt{Tr}, \rho_t) \\ \qquad \leftarrow \mathcal{A}^{\mathcal{O}\mathsf{reg}, \mathcal{O}\mathsf{corrupt}, \mathcal{O}\mathsf{sign}, \mathcal{O}\mathsf{report}}(pp); \\ b \leftarrow \mathsf{VerTrace}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_\mathsf{U}, \mathtt{Tr}, \rho_t) \end{array} : \begin{array}{l} b = 1 \wedge \mathsf{pk}_\mathsf{U} \in \mathcal{Q}\mathsf{reg} \backslash \mathcal{Q}\mathsf{corr} \\ \wedge\ \mathsf{Verify}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma) = 1 \\ \wedge\ (\mathsf{pk}_\mathsf{T}, \mathsf{pk}_\mathsf{U}, m, R, \sigma) \notin \mathcal{Q}\mathsf{sign} \end{array} \right] \leq \epsilon(\lambda).$$

Definition 12 (Trace soundness). *An* RTR *signature satisfies trace soundness if for any PPT adversary $\mathcal{A}$, there exists a negligible function ϵ such that,*

$$\Pr \left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda); \\ L,\ \mathcal{Q}\mathsf{reg},\ \mathcal{Q}\mathsf{corr},\ \mathcal{Q}\mathsf{sign},\ \mathcal{Q}\mathsf{report} \leftarrow \emptyset; \\ (\mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_{\mathsf{U}_i}, \mathtt{Tr}_i, \rho_{t_i}, \mathsf{pk}_{\mathsf{U}_j}, \mathtt{Tr}_j, \rho_{t_j}) \\ \qquad \leftarrow \mathcal{A}^{\mathcal{O}\mathsf{reg}, \mathcal{O}\mathsf{corrupt}, \mathcal{O}\mathsf{sign}, \mathcal{O}\mathsf{report}}(pp) \\ b_1 \leftarrow \mathsf{VerTrace}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_{\mathsf{U}_i}, \mathtt{Tr}_i, \rho_{t_i}); \\ b_2 \leftarrow \mathsf{VerTrace}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_{\mathsf{U}_j}, \mathtt{Tr}_j, \rho_{t_j}) \end{array} : \begin{array}{l} b_1 = 1 \wedge b_2 = 1 \\ \wedge\ \mathsf{pk}_{\mathsf{U}_i} \neq \mathsf{pk}_{\mathsf{U}_j} \end{array} \right] \leq \epsilon(\lambda).$$

Definition 13 (Reporter anonymity). *An* RTR *signature is reporter anonymous if, for any PPT adversary $\mathcal{A}$, there exists a negligible function ϵ such that,*

$$\left|\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda);\\ L,\ \mathcal{Q}\mathsf{reg},\ \mathcal{Q}\mathsf{corr},\ \mathcal{Q}\mathsf{sign},\ \mathcal{Q}\mathsf{report} \leftarrow \emptyset;\\ (\mathsf{pk_T}, m, R, \sigma, \mathsf{pk_{U_0}}, \mathsf{pk_{U_1}}, st)\\ \qquad \leftarrow \mathcal{A}^{\mathcal{O}\mathsf{reg},\mathcal{O}\mathsf{corrupt},\mathcal{O}\mathsf{sign},\mathcal{O}\mathsf{report}}(pp);\\ b \leftarrow \{0,1\};\\ \mathsf{Rep} \leftarrow \mathsf{Report}(pp, \mathsf{pk_T}, sk_{\mathsf{U}_b}, m, R, \sigma);\\ b' \leftarrow \mathcal{A}^{\mathcal{O}\mathsf{reg},\mathcal{O}\mathsf{corrupt},\mathcal{O}\mathsf{sign},\mathcal{O}\mathsf{report}}(\sigma, st) \end{array} : \begin{array}{l} b' = b\\ \wedge\ pk_{\mathsf{U}_0} \in (R \cap \mathcal{Q}\mathsf{reg}) \backslash \mathcal{Q}\mathsf{corr}\\ \wedge\ pk_{\mathsf{U}_1} \in (R \cap \mathcal{Q}\mathsf{reg}) \backslash \mathcal{Q}\mathsf{corr}\\ \wedge\ (m, R, \sigma, \mathsf{pk_{U_0}}) \notin \mathcal{Q}\mathsf{report}\\ \wedge\ (m, R, \sigma, \mathsf{pk_{U_1}}) \notin \mathcal{Q}\mathsf{report} \end{array}\right] - \frac{1}{2}\right| \leq \epsilon(\lambda).$$

C Proofs

In what follows, we prove Theorem 2 by proving each of the claimed properties. Each of the following lemmas corresponds to one of these properties (namely correctness and trace correctness, unforgeability, anonymity, non-frameability, trace soundness, and reporter anonymity).

Lemma 3. *If the NIZKP and the SoK are correct, then the protocol is correct and trace correct.*

Proof. First, assume that the signature algorithm does not return a failure. In this case, the correctness follows trivialy from the correctness of the proofs and signatures of knowledge, and from the correctness of the ElGamal encryption, and the probability of success is 1.

We deduce that the probability of failure of correctness is the same as the probability of failure of the signature algorithm (when used honestly with honestly generated keys). The signature algorithm returns a failure if and only if there is two encryption keys ek_i and ek_j that are equals ($\mathsf{ek}_i = \mathsf{ek}_j$). The encryption keys come from the uniform distribution on $\mathbb{G}$, thus according to the birthday paradox, this probability is $\epsilon(\lambda) = 1 - \prod_{i=0}^{n-1}\left(1 - \frac{i}{|\mathbb{G}|}\right)$ and is negligible in λ (we recall that n is polynomial in λ ans $|\mathbb{G}|$ is exponential in λ). □

Lemma 4. *If the NIZKP and the SoK are extractable and zero-knowledge, then the protocol is unforgeable under the discrete logarithm assumption.*

Proof. Let $\mathcal{A}$ be a PPT adversary that breaks the unforgeability of our scheme with probability $\epsilon_{\mathcal{A}}(\lambda)$. We will show how to use it to break the discrete logarithm assumption. Let $\epsilon_{\mathsf{NIZK}}(\lambda)$ be the maximum knowledge error among all the non-interactive proofs and signatures of knowledge used in the protocol. We use the following sequence of games:

Game G_0: This game is the unforgeability experiment as described in Definition 10 on our protocol. We have:

$$\epsilon_{\mathcal{A}}(\lambda) = \Pr[\mathcal{A} \text{ wins } G_0]$$

Game G_1: This game is similar to G_0 except that the challenger sets an empty dictionary SIG at the beginning of the experiment, and each time that the challenger generates a signature σ for a tracer key $\mathsf{pk_T}$ message m and a ring R with the secret key $\mathsf{sk_U}$, it stores the element $\alpha \in \mathbb{Z}_p^*$ (according to the notation we use in the definition of the signature algorithm) and $\mathsf{sk_U}$ at the key $(\mathsf{pk_T}, m, R, \sigma)$ of the dictionary SIG:

$$\mathsf{SIG}[(\mathsf{pk_T}, m, R, \sigma)] \leftarrow (\alpha, \mathsf{sk_U}).$$

This modification does not alter the advantage of $\mathcal{A}$:

$$\Pr[\mathcal{A} \text{ wins } G_0] = \Pr[\mathcal{A} \text{ wins } G_1]$$

Game G_2: This game is similar to G_1 except that each time that the challenger verifies a valid signature σ on a message m and the ring R forged by $\mathcal{A}$, it uses the knowledge extractor on each proof of knowledge and each signature of knowledge in σ (or the challenger retrieves the corresponding witness if it has previously produced the same proof/signature during the game) in order to recover α and $\mathsf{sk_U}$, and stores the witness extracted from the signature of knowledge in the dictionary at the key $(\mathsf{pk_T}, m, R, \sigma)$:

$$\mathsf{SIG}[(\mathsf{pk_T}, m, R, \sigma)] \leftarrow (\alpha, \mathsf{sk_U}).$$

If one extraction returns a false witness, the challenger aborts the game and returns 0. Let n be the number of registered users, and q_r be the number of calls to the oracle $\mathcal{O}\mathsf{report}$. Each signature contains n proofs/signatures of knowledge. The challenger verifies $(q_r + 1)$ signatures (one for each call to the oracle $\mathcal{O}\mathsf{report}$ and one during the experiment), so the probability that the challenger aborts the game is the probability that the extractor fails on a valid proof, which is at most $(q_r + 1) \cdot \epsilon_{\mathsf{NIZK}}(\lambda)$. We deduce that:

$$|\Pr[\mathcal{A} \text{ wins } G_1] - \Pr[\mathcal{A} \text{ wins } G_2]| \leq (q_r + 1) \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda).$$

Game G_3: Let $(\mathsf{pk_T}, m, R, \sigma)$ be the values returned by the adversary. At this step, if the challenger does not abort the game, then it knows the key $\mathsf{sk_U}$ and α (from the extraction of the signature of knowledge in σ) such that, parsing σ as $(h, c, (c_i, \pi_i')_{i=1}^n, \sigma_M)$, R as $\{\mathsf{pk}_i\}_{i=1}^n$, and each pk_i as (ek_i, π_i):

$$\exists j, h = g^\alpha \text{ and } \left(\frac{c \cdot c_i}{\mathsf{ek}_j}\right) = (\mathsf{ek_T} \cdot \mathsf{ek}_j)^\alpha \text{ and } \mathsf{ek}_j = g^{\mathsf{sk_U}}$$

Moreover, $R \subseteq \mathcal{Q}\mathsf{reg} \backslash \mathcal{Q}\mathsf{corr}$, so $\mathsf{pk}_j \in \mathcal{Q}\mathsf{reg} \backslash \mathcal{Q}\mathsf{corr}$.

Let n be the number of registered users (*i.e.*, the number of queries to the oracle $\mathcal{O}\mathsf{reg}$), and let $\mathsf{pk}_i' = (\mathsf{ek}_i', \pi_i')$ be the public key generated at the i^{th} query. At the beginning of the experiment the challenger picks $\theta \xleftarrow{\$} [\![n]\!]$. If $\mathsf{ek}_\theta' \neq \mathsf{ek}_j$, then the challenger aborts the game and returns 0. We have that:

$$\frac{1}{n} \cdot \Pr[\mathcal{A} \text{ wins } G_2] = \Pr[\mathcal{A} \text{ wins } G_3].$$

Game G_4: In this game, the challenger simulates each proof and signature of knowledge using the corresponding simulators. Since the proofs and signatures of knowledge are perfectly zero-knowledge, we have:

$$\Pr[\mathcal{A} \text{ wins } G_3] = \Pr[\mathcal{A} \text{ wins } G_4].$$

At this step, the challenger does not need to know the secret key corresponding to pk'_θ for simulating the signature oracle $\mathcal{O}\mathsf{sign}$.

Game G_5: This game is the same as G_4 except that each time the adversary sends a query $(\mathsf{pk_U}, \mathsf{pk_T}, m, R, \sigma)$ to the oracle $\mathcal{O}\mathsf{report}$ such that σ is valid, the challenger retrieves $(\alpha, \mathsf{sk_U}) \leftarrow \mathsf{SIG}[(\mathsf{pk_T}, m, R, \sigma)]$ and computes S_2 as follows, using the same notation as in the description of the protocol:

$$S_2 \leftarrow \frac{c_j}{\mathsf{pk}_\mathsf{U}^\alpha}.$$

Since:

$$\frac{c_j}{\mathsf{pk}_\mathsf{U}^\alpha} = \frac{c_j}{g^{\alpha \cdot \mathsf{sk_U}}} = \frac{c_j}{h^{\mathsf{sk_U}}},$$

we deduce that:

$$\Pr[\mathcal{A} \text{ wins } G_4] = \Pr[\mathcal{A} \text{ wins } G_5].$$

At this step, the challenger does not need to know the secret key corresponding to pk'_θ (which is its discrete logarithm) for simulating the signature oracle $\mathcal{O}\mathsf{sign}$ and the report oracle $\mathcal{O}\mathsf{report}$, so it never uses this secret key in the game G_5.

Let $\epsilon_{\mathsf{dl}}(\lambda)$ be the probability of the best algorithm that solves the discrete logarithm problem. We claim that:

$$\Pr[\mathcal{A} \text{ wins } G_5] \leq \epsilon_{\mathsf{dl}}(\lambda).$$

We prove this claim by reduction. Assume that $\mathcal{A}$ has a non-negligible probability of winning the game G_5. We will show how to build a PPT algorithm $\mathcal{B}$ that breaks the discrete logarithm assumption with a similar probability. $\mathcal{B}$ receives the group element X. It simulates perfectly the game G_5 to $\mathcal{A}$ except that it replaces ek'_θ by X and sk'_θ by $\perp$.

Let $(\mathsf{pk_T}, m, R, \sigma)$ be the values returned by the adversary. At the end of the game, $\mathcal{B}$ sets $N \leftarrow |R|$, parses R as $\{\mathsf{pk}_i\}_{i=1}^N$, pk_i as (ek_i, π_i) for all indexes i, and σ as $(h, c, (c_i, \pi'_i)_{i=1}^N, \sigma_M)$. According to the unforgeability experiment, for all pk we have $(\mathsf{pk_T}, \mathsf{pk}, m, R, \sigma) \notin \mathcal{Q}\mathsf{sign}$, so the challenger has never produced the signature of knowledge σ_M on the message $M = (pp, \mathsf{pk_T}, m, R, h, c, (c_i, \pi'_i)_{i=1}^n)$ during the experiment, which implies that if the game has not been aborted by $\mathcal{B}$, then there exists a key sk that has been extracted from σ_M during the experiment such that there exists j such that $g^{\mathsf{sk}} = \mathsf{ek}_j = \mathsf{ek}'_\theta = X$. Finally, If $\mathcal{B}$ does not abort the game, then it returns sk, which is the discrete logarithm of X. This concludes the proof of the claim.

Conclusion: Finally, we have:

$$\epsilon_{\mathcal{A}}(\lambda) \leq n \cdot \epsilon_{\mathsf{dl}}(\lambda) + (q_r + 1) \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda).$$

We deduce that $\epsilon_{\mathcal{A}}(\lambda)$ is negligible, which concludes the proof. □

Lemma 5. *If the NIZKP and the SoK are extractable and zero-knowledge, then the protocol is anonymous under the IND-CPA security of ElGamal in a prime order group.*

Proof. Let $\mathcal{A}$ be a PPT adversary that breaks the anonymity of our scheme with advantage $\epsilon_{\mathcal{A}}(\lambda)$. We will show how to use it to break the IND-CPA security of the ElGamal encryption. Let $\epsilon_{\mathsf{NIZK}}(\lambda)$ be the maximum knowledge error among all the non-interactive proofs and signatures of knowledge used in the protocol. We use the following sequence of games:

Game G_0: This game is the anonymity experiment as described in Definition 9 on our protocol. We have:

$$\epsilon_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A} \text{ wins } G_0] - 1/2|$$

Game G_1: This game is similar to G_0 except that the challenger sets an empty dictionary SIG at the beginning of the experiment, and each time that the challenger generates a signature σ for a tracer key $\mathsf{pk_T}$, a message m and a ring R with the secret key $\mathsf{sk_U}$, it stores the element $\alpha \in \mathbb{Z}_p^*$ (according to the notation we use in the definition of the signature algorithm) and $\mathsf{sk_U}$ at the key $(\mathsf{pk_T}, m, R, \sigma)$ of the dictionary SIG:

$$\mathsf{SIG}[(\mathsf{pk_T}, m, R, \sigma)] \leftarrow (\alpha, \mathsf{sk_U}).$$

This modification does not alter the advantage of $\mathcal{A}$:

$$\Pr[\mathcal{A} \text{ wins } G_0] = \Pr[\mathcal{A} \text{ wins } G_1]$$

Game G_2: This game is similar to G_1 except that each time that the challenger verifies a valid signature σ on a message m and the ring R forged by $\mathcal{A}$, it uses the knowledge extractor on each proof of knowledge and each signature of knowledge in σ in order to recover α and $\mathsf{sk_U}$, and stores the witness extracted from the signature of knowledge in the dictionary at the key $(\mathsf{pk_T}, m, R, \sigma)$:

$$\mathsf{SIG}[(\mathsf{pk_T}, m, R, \sigma)] \leftarrow (\alpha, \mathsf{sk_U}).$$

If one extraction returns a false witness, the challenger aborts the game and returns 0 with probability 1/2. Let n be the number of users, q_r be the number of call to the oracle $\mathcal{O}\mathsf{report}$, and q_t be the number of call to the oracle $\mathcal{O}\mathsf{trace}$. Each signature contains n proofs/signatures of knowledge. The challenger verifies $(q_r + q_t)$ signatures, so the probability that the challenger aborts the game

is the probability that the extractor fails on a valid proof, which is at most $(q_r + q_t) \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda)$. We deduce that:

$$|\Pr[\mathcal{A} \text{ wins } G_1] - \Pr[\mathcal{A} \text{ wins } G_2]| \leq (q_r + q_t) \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda).$$

Game G_3: In this game, the challenger simulates each proof and signature of knowledge using the corresponding simulators. Since the proofs and signatures of knowledge are perfectly zero-knowledge, we have:

$$\Pr[\mathcal{A} \text{ wins } G_2] = \Pr[\mathcal{A} \text{ wins } G_3].$$

Game G_4: This game is the same as G_3 except that each time the adversary sends a query $(\mathsf{pk_U}, \mathsf{pk_T}, m, R, \sigma)$ to the oracle $\mathcal{O}\mathsf{trace}$ such that σ is valid, the challenger retrieves $(\alpha, \mathsf{sk_U}) \leftarrow \mathsf{SIG}[(\mathsf{pk_T}, m, R, \sigma)]$ and computes S_1 as follows, using the same notation as in the description of the protocol:

$$S_1 \leftarrow \frac{c}{\mathsf{pk}_{\mathsf{Trace}}^{\alpha}}.$$

Since:

$$\frac{c}{\mathsf{pk}_{\mathsf{Trace}}^{\alpha}} = \frac{c}{g^{\alpha \cdot \mathsf{sk}_{\mathsf{Trace}}}} = \frac{c}{h^{\mathsf{sk}_{\mathsf{Trace}}}},$$

we deduce that:

$$\Pr[\mathcal{A} \text{ wins } G_4] = \Pr[\mathcal{A} \text{ wins } G_5].$$

At this step, the challenger does not need to know the secret key corresponding to $\mathsf{pk}_{\mathsf{Trace}}$ for simulating the trace oracle $\mathcal{O}\mathsf{trace}$.
Game G_5: This game is the same as G_4 except that each time the challenger build a signature on a ring :

- for each public key $\mathsf{pk}_i = (\mathsf{ek}_i, \pi_i)$ in $R = \{\mathsf{pk}_i\}_{i=1}^n$, if $\mathsf{pk}_i \notin \mathcal{Q}\mathsf{reg}$, then it uses the knowledge extractor on π_i in order to retrieve sk_i such that $\mathsf{pk}_i = g_i^{\mathsf{sk}}$. If the extraction fails, the challenger aborts the game and returns 0 with probability 1/2.
- The challenger computes $c_i \leftarrow h^{\mathsf{sk}_i} \cdot S_2$ instead of $c_i \leftarrow {\mathsf{ek}_i}^{\alpha} \cdot S_2$ (using the same notation as in the description of the protocol). Note that $h^{\mathsf{sk}_i} = g^{\alpha \cdot \mathsf{sk}_i} = {\mathsf{ek}_i}^{\alpha}$, so this does not impact the simulation of the game.

Besides that, the challenger simulates the signatures as in the previous game.

Let n be the number of users, and q_s be the number of call to the oracle $\mathcal{O}\mathsf{sign}$. The challenger builds $(q_s + 1)$ signatures (one for each call to the oracle $\mathcal{O}\mathsf{sign}$ and one during the experiment). The probability that the challenger aborts the game is the probability that the extractor fails on a valid proof, which is at most $(q_s + 1) \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda)$. We deduce that:

$$|\Pr[\mathcal{A} \text{ wins } G_4] - \Pr[\mathcal{A} \text{ wins } G_5]| \leq (q_s + 1) \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda).$$

At this step, the challenger no longer needs to know the discrete logarithm of h to build a valid signature except for computing c.

Let $\epsilon_{\mathsf{EG}}(\lambda)$ be the advantage of the best algorithm that breaks the IND-CPA security of the ElGamal encryption $\mathsf{EG} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. We claim that:

$$|\Pr[\mathcal{A} \text{ wins } G_5] - 1/2| \leq \epsilon_{\mathsf{EG}}(\lambda).$$

We prove this claim by reduction. Assume that $\mathcal{A}$ has a non-negligible advantage on the game G_5. We will show how to build a PPT algorithm $\mathcal{B}$ that breaks the IND-CPA security of ElGamal with a similar advantage. $\mathcal{B}$ receives the public key pk. It simulates perfectly the game G_5 to $\mathcal{A}$ except that it replaces ek_T by pk and sk_T by $\perp$, and builds the signature $\sigma \leftarrow \mathsf{Sign}(pp, sk_{\mathsf{U}_b}, \mathsf{pk}_\mathsf{T}, m, R \cup \{pk_{\mathsf{U}_0}, \mathsf{pk}_{\mathsf{U}_1}\})$ as follows:

- $\mathcal{B}$ picks $S_2 \xleftarrow{\$} \mathbb{G}$, sets $S_{1,0} \leftarrow \frac{\mathsf{pk}_0}{S_2}$ and $S_{1,1} \leftarrow \frac{\mathsf{pk}_1}{S_2}$.
- $\mathcal{B}$ sends $(S_{1,0}, S_{1,1})$ to its challenger and receives the ciphertext $e = (e_1, e_2)$, which is the ElGamal encryption of $S_{1,b}$ for a random bit b.
- $\mathcal{B}$ sets $h \leftarrow e_1$ and $c \leftarrow e_2$.
- It sets $\mathsf{SIG}[(\mathsf{pk}_\mathsf{T}, m, R, \sigma)] \leftarrow \perp$

Besides that, the challenger simulates the signatures as in the previous game. We observe that:

$$c = h^{\mathsf{sk}_\mathsf{T}} \cdot S_{1,b}.$$

We also recall that the challenger aborts the game on the signature (m, R, σ), so $\mathcal{B}$ will never use $\mathsf{SIG}[(\mathsf{pk}_\mathsf{T}, m, R, \sigma)]$. Finally, $\mathcal{B}$ perfectly simulates the game G_5 and wins its IND-CPA attack with the same probability that $\mathcal{A}$ wins the game G_5, which concludes the proof of the claim.

Conclusion: Finally, we have:

$$\epsilon_{\mathcal{A}}(\lambda) \leq (q_r + q_t + q_s + 1) \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda) + \epsilon_{\mathsf{EG}}(\lambda).$$

We deduce that $\epsilon_{\mathcal{A}}(\lambda)$ is negligible, which concludes the proof. □

Lemma 6. *If the NIZKP and the SoK are extractable and zero-knowledge, then the protocol is non-frameable under the discrete logarithm assumption.*

Proof. Let $\mathcal{A}$ be a PPT adversary that breaks the non-frameability of our scheme with probability $\epsilon_{\mathcal{A}}(\lambda)$. We will show how to use it to break the discrete logarithm assumption. Let $\epsilon_{\mathsf{NIZK}}(\lambda)$ be the maximum knowledge error among all the non-interactive proofs and signatures of knowledge used in the protocol. We use the following sequence of games:

Game G_0: This game is the non-frameability experiment as described in Definition 11 on our protocol. We have:

$$\epsilon_{\mathcal{A}}(\lambda) = \Pr[\mathcal{A} \text{ wins } G_0]$$

Game G_1: This game is similar to G_0 except that the challenger sets an empty dictionary SIG at the beginning of the experiment, and each time that the challenger generates a signature σ for a tracer key pk_T, a message m and a

ring R with the secret key sk_U, it stores the element $\alpha \in \mathbb{Z}_p^*$ (according to the notation we use in the definition of the signature algorithm) and sk_U at the key $(\mathsf{pk}_\mathsf{T}, m, R, \sigma)$ of the dictionary SIG:

$$\mathsf{SIG}[(\mathsf{pk}_\mathsf{T}, m, R, \sigma)] \leftarrow (\alpha, \mathsf{sk}_\mathsf{U}).$$

This modification does not alter the advantage of $\mathcal{A}$:

$$\Pr[\mathcal{A} \text{ wins } G_0] = \Pr[\mathcal{A} \text{ wins } G_1]$$

Game G_2: This game is similar to G_1 except that each time that the challenger verifies a valid signature σ on a message m and the ring R forged by $\mathcal{A}$, it uses the knowledge extractor on each proof of knowledge and each signature of knowledge in σ (or the challenger retrieves the corresponding witness if it has previously produced the same proof/signature during the game) in order to recover α and sk_U, and stores the witness extracted from the signature of knowledge in the dictionary at the key $(\mathsf{pk}_\mathsf{T}, m, R, \sigma)$:

$$\mathsf{SIG}[(\mathsf{pk}_\mathsf{T}, m, R, \sigma)] \leftarrow (\alpha, \mathsf{sk}_\mathsf{U}).$$

If one extraction returns a false witness, the challenger aborts the game and returns 0 with probability 1/2. Let n be the number of users, and q_r be the number of call to the oracle $\mathcal{O}\mathsf{report}$. Each signature contains n proofs/signatures of knowledge. The challenger verifies $(q_r + 1)$ signatures (one for each call to the oracle $\mathcal{O}\mathsf{report}$ and one during the experiment), so the probability that the challenger aborts the game is the probability that the extractor fails on a valid proof, which is at most $(q_r + 1) \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda)$. We deduce that:

$$|\Pr[\mathcal{A} \text{ wins } G_1] - \Pr[\mathcal{A} \text{ wins } G_2]| \leq (q_r + 1) \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda).$$

Game G_3: Let $(\mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_\mathsf{U}, \mathtt{Tr}, \rho_t)$ be the values returned by the adversary. Let n be the number of registered users (*i.e.*, the number of queries to the oracle $\mathcal{O}\mathsf{reg}$), and let $\mathsf{pk}_i = (\mathsf{ek}_i, \pi_i)$ be the public key generated at the i^{th} query. At the beginning of the experiment the challenger picks $\theta \xleftarrow{\$} [\![n]\!]$. If $\mathsf{pk}_\theta \neq \mathsf{pk}_\mathsf{U}$, then the challenger aborts the game and returns 0. We have that:

$$\frac{1}{n} \cdot \Pr[\mathcal{A} \text{ wins } G_2] = \Pr[\mathcal{A} \text{ wins } G_3]\,.$$

Game G_4: Let $(\mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_\mathsf{U}, \mathtt{Tr}, \rho_t)$ be the values returned by the adversary. This game is the same as G_3 except that if $\exists \mathsf{pk}$ such that $(\mathsf{pk}_\mathsf{T}, \mathsf{pk}, m, R, \sigma) \in \mathcal{Q}\mathsf{sign}$, then the challenger aborts the game and returns 0.

We recall that if the challenger does not abort the game, then all the relations proved by the zero-knowledge proofs and the signatures of knowledge are valid. We set $N \leftarrow |R|$, and we parse R as $\{\mathsf{pk}'_i\}_{i=1}^N$, pk'_i as (ek'_i, π'_i) for all indexes i, $\mathtt{Tr}$ as $(S_2, \pi_{\mathtt{Rep}})$, ρ_t as (S_1, π_{ρ_t}), and σ as $(h, c, (c_i, \pi''_i)_{i=1}^N, \sigma_M)$. Let i be the index of the key ek'_i such that $\mathsf{ek}'_i = \mathsf{ek}_\theta$ (we recall that $\mathsf{pk}_\theta = (\mathsf{ek}_\theta, \pi_\theta) = (\mathsf{ek}_\mathsf{U}, \pi_\mathsf{U}) = \mathsf{pk}_\mathsf{U}$, otherwise the game would have been aborted). The proofs of knowledge π_{ρ_t}, $\pi_{\mathtt{Rep}}$, and $\{\pi''_i\}_{i=1}^N$ ensure that there exists α such that $h = g^\alpha$ and:

– $c_i = \mathsf{ek}_i'^{\alpha} \cdot S_2$, so $\frac{c_i}{\mathsf{ek}_i'^{\alpha}} = S_2$.
– $c = \mathsf{ek}_{\mathsf{T}}^{\alpha} \cdot S_1$.
– For all $j \in [\![2, N]\!]$, $\frac{c_j}{c_{j-1}} = \left(\frac{\mathsf{ek}_j'}{\mathsf{ek}_{j-1}'}\right)^{\alpha}$.

We deduce that for all $j \in [\![2, N]\!]$, $\frac{c_j}{\mathsf{ek}_j'^{\alpha}} = \frac{c_{j-1}}{\mathsf{ek}_{j-1}'^{\alpha}}$, so for all j and j' in $[\![1, N]\!]$, $\frac{c_j}{\mathsf{ek}_j'^{\alpha}} = \frac{c_{j'}}{\mathsf{ek}_{j'}'^{\alpha}}$. Since $\frac{c_i}{\mathsf{ek}_i'^{\alpha}} = S_2$, we deduce that for all $j \in [\![1, N]\!]$, we have $\frac{c_j}{\mathsf{ek}_j'^{\alpha}} = S_2$, so $c_j = \mathsf{ek}_j'^{\alpha} \cdot S_2$. The signature of knowledge σ_M ensures that there exists sk such that:

$$\exists j, \left(\frac{c \cdot c_j}{\mathsf{ek}_j'}\right) = \left(\mathsf{ek}_{\mathsf{T}} \cdot \mathsf{ek}_j'\right)^{\alpha} \text{ and } \mathsf{ek}_j' = g^{\mathsf{sk}}$$

We observe that:

$$\left(\frac{c \cdot c_j}{\mathsf{ek}_j'}\right) = \left(\mathsf{ek}_{\mathsf{T}} \cdot \mathsf{ek}_j'\right)^{\alpha} \Leftrightarrow \left(\frac{\mathsf{ek}_{\mathsf{T}}^{\alpha} \cdot S_1 \cdot \mathsf{ek}_j'^{\alpha} \cdot S_2}{\mathsf{ek}_j'}\right) = \mathsf{ek}_{\mathsf{T}}^{\alpha} \cdot \mathsf{ek}_j'^{\alpha} \Leftrightarrow S_1 \cdot S_2 = \mathsf{ek}_j'.$$

Since $\mathsf{VerTrace}(pp, \mathsf{pk}_{\mathsf{T}}, m, R, \sigma, \mathsf{pk}_{\mathsf{U}}, \mathtt{Tr}, \rho_t) = 1$, we deduce that $S_1 \cdot S_2 = \mathsf{ek}_{\mathsf{U}} = \mathsf{ek}_\theta$. Since $S_1 \cdot S_2 = \mathsf{ek}_j'$, we deduce that $\mathsf{ek}_j' = \mathsf{ek}_{\mathsf{U}}$. According to the correctness of our protocol, since the challenger correctly simulates the oracle $\mathcal{O}\mathsf{sign}$, if $\exists \mathsf{pk}$ such that $(\mathsf{pk}_{\mathsf{T}}, \mathsf{pk}, m, R, \sigma) \in \mathcal{Q}\mathsf{sign}$, then $\mathsf{pk} = \mathsf{pk}_{\mathsf{U}}$. In this case the challenger aborts the game according to the non-frameability experiment. We deduce that:

$$\Pr[\mathcal{A} \text{ wins } G_3] = \Pr[\mathcal{A} \text{ wins } G_4].$$

Game G_5: In this game, the challenger simulates each proof and signature of knowledge using the corresponding simulators. Since the proofs and signatures of knowledge are perfectly zero-knowledge, we have:

$$\Pr[\mathcal{A} \text{ wins } G_4] = \Pr[\mathcal{A} \text{ wins } G_5].$$

At this step, the challenger does not need to know the secret key corresponding to pk_θ for simulating the signature oracle $\mathcal{O}\mathsf{sign}$.

Game G_6: This game is the same as G_5 except that each time the adversary sends a query $(\mathsf{pk}_{\mathsf{U}}, \mathsf{pk}_{\mathsf{T}}, m, R, \sigma)$ to the oracle $\mathcal{O}\mathsf{report}$ such that σ is valid and $\mathsf{pk}_\theta = \mathsf{pk}_{\mathsf{U}}$, the challenger retrieves $(\alpha, \mathsf{sk}_{\mathsf{U}}) \leftarrow \mathsf{SIG}[(\mathsf{pk}_{\mathsf{T}}, m, R, \sigma)]$ and computes S_2 as follows, using the same notation as in the description of the protocol:

$$S_2 \leftarrow \frac{c_j}{\mathsf{pk}_{\mathsf{U}}^{\alpha}}.$$

Since:

$$\frac{c_j}{\mathsf{pk}_{\mathsf{U}}^{\alpha}} = \frac{c_j}{g^{\alpha \cdot \mathsf{sk}_{\mathsf{U}}}} = \frac{c_j}{h^{\mathsf{sk}_{\mathsf{U}}}},$$

we deduce that:

$$\Pr[\mathcal{A} \text{ wins } G_5] = \Pr[\mathcal{A} \text{ wins } G_6].$$

At this step, the challenger does not need to know the secret key corresponding to pk_θ (which is its discrete logarithm) for simulating the signature oracle $\mathcal{O}\mathsf{sign}$ and the report oracle $\mathcal{O}\mathsf{report}$, so it never uses this secret key in the game G_6.

Let $\epsilon_{\mathsf{dl}}(\lambda)$ be the probability of the best algorithm that solves the discrete logarithm problem. We claim that:

$$\Pr[\mathcal{A} \text{ wins } G_6] \leq \epsilon_{\mathsf{dl}}(\lambda).$$

We prove this claim by reduction. Assume that $\mathcal{A}$ has a non-negligible probability of winning the game G_6. We will show how to build a PPT algorithm $\mathcal{B}$ that breaks the discrete logarithm assumption with a similar probability. $\mathcal{B}$ receives the group element X. It simulates perfectly the game G_6 to $\mathcal{A}$ except that it replaces ek_θ by X and sk_θ by $\perp$.

We recall that if the challenger does not abort the game, then all the relations proved by the zero-knowledge proofs and the signatures of knowledge are valid. Let $(\mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_\mathsf{U}, \mathtt{Tr}, \rho_t)$ be the values returned by the adversary. At the end of the game, $\mathcal{B}$ sets $N \leftarrow |R|$, parses R as $\left\{\mathsf{pk}'_i\right\}_{i=1}^N$, pk'_i as (ek'_i, π'_i) for all indexes i, $\mathtt{Tr}$ as $(S_2, \pi_{\mathtt{Rep}})$, ρ_t as (S_1, π_{ρ_t}), and σ as $(h, c, (c_i, \pi''_i)_{i=1}^N, \sigma_M)$. Let i be the index of the key ek'_i such that $\mathsf{ek}'_i = \mathsf{ek}_\theta$ (we recall that $\mathsf{ek}_\theta = \mathsf{ek}_\mathsf{U}$, otherwise the game would have been aborted). The proofs/signatures of knowledge π_{ρ_t}, $\pi_{\mathtt{Rep}}$, $\{\pi''_i\}_{i=1}^N$, and σ_M ensure that there exists α and sk such that $h = g^\alpha$ and:

- $c_i = \mathsf{ek}_i'^\alpha \cdot S_2$.
- $c = \mathsf{ek}_\mathsf{T}^\alpha \cdot S_1$.
- For all $j \in [\![2, N]\!]$, $\frac{c_j}{c_{j-1}} = \left(\frac{\mathsf{ek}'_j}{\mathsf{ek}'_{j-1}}\right)^\alpha$.
- $\exists j, \left(\frac{c \cdot c_j}{\mathsf{ek}'_j}\right) = \left(\mathsf{ek}_\mathsf{T} \cdot \mathsf{ek}'_j\right)^\alpha$ and $\mathsf{ek}'_j = g^{\mathsf{sk}}$.

As it is shown on the game G_4, since $\mathsf{VerTrace}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_\mathsf{U}, \mathtt{Tr}, \rho_t) = 1$, we can deduce that $S_1 \cdot S_2 = \mathsf{ek}_\mathsf{U} = \mathsf{ek}_\theta = X$. Since $S_1 \cdot S_2 = \mathsf{ek}'_j$ and $\mathsf{ek}'_j = g^{\mathsf{sk}}$, we deduce that $X = g^{\mathsf{sk}}$, so sk is the discrete logarithm of X.

$\mathcal{B}$ retrieves $(\alpha, \mathsf{sk}) \leftarrow \mathsf{SIG}[(\mathsf{pk}_\mathsf{T}, m, R, \sigma)]$. If $\mathsf{sk} = \perp$, then σ has been produced by the oracle $\mathcal{O}\mathsf{sign}$ on input $(\mathsf{pk}_\mathsf{U}, \mathsf{pk}_\mathsf{T}, m, R)$, so $(\mathsf{pk}_\mathsf{T}, \mathsf{pk}_\mathsf{U}, m, R, \sigma) \in \mathcal{Q}\mathsf{sign}$, and $\mathcal{B}$ aborts as in the real game G_5. Else, according to game G_4, for all pk we have $(\mathsf{pk}_\mathsf{T}, \mathsf{pk}, m, R, \sigma) \notin \mathcal{Q}\mathsf{sign}$, so the challenger has never produced the signature of knowledge σ_M on the message $M = (pp, \mathsf{pk}_\mathsf{T}, m, R, h, c, (c_i, \pi''_i)_{i=1}^n)$ during the experiment, which implies that sk has been extracted from σ_M.

Finally, $\mathcal{B}$ returns sk. It perfectly simulates the game G_6 to $\mathcal{A}$, and if $\mathcal{A}$ wins the simulated game G_6, then $\mathcal{B}$ returns the discrete logarithm of X, which concludes the proof of the claim.

Conclusion: Finally, we have:

$$\epsilon_\mathcal{A}(\lambda) \leq n \cdot \epsilon_{\mathsf{dl}}(\lambda) + (q_r + 1) \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda).$$

We deduce that $\epsilon_\mathcal{A}(\lambda)$ is negligible, which concludes the proof. □

Lemma 7. *If the NIZKP and the SoK are extractable, then the protocol is sound.*

Proof. We claim that if the adversary returns two $(\mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_{\mathsf{U}_i}, \mathtt{Tr}_i, \rho_{t_i}, \mathsf{pk}_{\mathsf{U}_j}, \mathtt{Tr}_j, \rho_{t_j})$ such that $\mathsf{VerTrace}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_{\mathsf{U}_i}, \mathtt{Tr}_i, \rho_{t_i}) = \mathsf{VerTrace}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_{\mathsf{U}_j}, \mathtt{Tr}_j, \rho_{t_j}) = 1$ and $\mathsf{pk}_{\mathsf{U}_i} \neq \mathsf{pk}_{\mathsf{U}_j}$, then it produces a valid proof/signature of knowledge on a false statement (which appends with negligible probability since the proofs/signatures are extractable). To prove this claim, we prove its contraposition: if all the valid proofs/signatures of knowledge are on true statements and $\mathsf{VerTrace}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_{\mathsf{U}_i}, \mathtt{Tr}_i, \rho_{t_i}) = \mathsf{VerTrace}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_{\mathsf{U}_j}, \mathtt{Tr}_j, \rho_{t_j}) = 1$, then $\mathsf{pk}_{\mathsf{U}_i} = \mathsf{pk}_{\mathsf{U}_j}$.

We set $N \leftarrow |R|$, and we parse R as $\{\mathsf{pk}_l\}_{l=1}^N$, pk_i as (ek_l, π_l) for all indexes l, $\mathtt{Tr}_i$ as $(S_{2,i}, \pi_{\mathtt{Rep},i})$, ρ_{t_i} as $(S_{1,i}, \pi_{\rho_{t_i}})$, $\mathtt{Tr}_j$ as $(S_{2,j}, \pi_{\mathtt{Rep},j})$, ρ_{t_j} as $(S_{1,j}, \pi_{\rho_{t_j}})$, and σ as $(h, c, (c_l, \pi'_l)_{l=1}^N, \sigma_M)$. Since $\mathsf{VerTrace}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_{\mathsf{U}_i}, \mathtt{Tr}_i, \rho_{t_i}) = \mathsf{VerTrace}(pp, \mathsf{pk}_\mathsf{T}, m, R, \sigma, \mathsf{pk}_{\mathsf{U}_j}, \mathtt{Tr}_j, \rho_{t_j}) = 1$, the proofs/signatures of knowledge $(\pi'_l)_{l=1}^N$, σ_M, $\pi_{\mathtt{Rep},i}$, $\pi_{\mathtt{Rep},j}$, $\pi_{\rho_{t_i}}$, and $\pi_{\rho_{t_j}}$ ensure that:

- There exists α such that $h = g^\alpha$ and for all $l \in [\![2, N]\!]$:

$$\frac{c_l}{c_{l-1}} = \left(\frac{\mathsf{ek}_l}{\mathsf{ek}_{l-1}}\right)^\alpha,$$

  and there exists $\mathsf{pk}_d = (\mathsf{ek}_d, \pi_d)$ such that:

$$\frac{c \cdot c_d}{\mathsf{ek}_d} = (\mathsf{ek}_\mathsf{T} \cdot \mathsf{ek}_d)^\alpha.$$

- There exists i' such that $\mathsf{pk}_{i'} = \mathsf{pk}_{\mathsf{U}_i}$, $\frac{c_{i'}}{S_{2,i}} = h^{\mathsf{sk}_{i'}}$ and $\frac{c}{S_{1,i}} = h^{\mathsf{sk}_\mathsf{T}}$.
- There exists j' such that $\mathsf{pk}_{j'} = \mathsf{pk}_{\mathsf{U}_j}$, $\frac{c_{j'}}{S_{2,j}} = h^{\mathsf{sk}_{j'}}$ and $\frac{c}{S_{1,j}} = h^{\mathsf{sk}_\mathsf{T}}$.

Assume, without loss of generality, that $i' < j'$. We deduce:

- $\frac{c}{S_{1,i}} = h^{\mathsf{sk}_\mathsf{T}}$ and $\frac{c}{S_{1,j}} = h^{\mathsf{sk}_\mathsf{T}} \Rightarrow S_{1,i} = S_{1,j}$.
- $\frac{c_{i'}}{S_{2,i}} = h^{\mathsf{sk}_{i'}}$ and $\frac{c_{j'}}{S_{2,j}} = h^{\mathsf{sk}_{j'}} \Rightarrow S_{2,i} = \frac{c_{i'}}{h^{\mathsf{sk}_{i'}}}$ and $S_{2,j} = \frac{c_{j'}}{h^{\mathsf{sk}_{j'}}}$, so:

$$S_{2,i} = \frac{c_{i'}}{h^{\mathsf{sk}_{i'}}} = \frac{c_{i'}}{g^{\alpha \cdot \mathsf{sk}_{i'}}} = \frac{c_{i'}}{\mathsf{ek}_{i'}^\alpha},$$

$$S_{2,j} = \frac{c_{j'}}{h^{\mathsf{sk}_{j'}}} = \frac{c_{j'}}{g^{\alpha \cdot \mathsf{sk}_{j'}}} = \frac{c_{j'}}{\mathsf{ek}_{j'}^\alpha}.$$

- For all $l \in [\![2, N]\!]$:

$$\frac{c_l}{c_{l-1}} = \left(\frac{\mathsf{ek}_l}{\mathsf{ek}_{l-1}}\right)^\alpha \Rightarrow \prod_{l=i'}^{j'} \frac{c_l}{c_{l-1}} = \prod_{l=i'}^{j'} \left(\frac{\mathsf{ek}_l}{\mathsf{ek}_{l-1}}\right)^\alpha \Rightarrow \frac{c_{j'}}{c_{i'}} = \left(\frac{\mathsf{ek}_{j'}}{\mathsf{ek}_{i'}}\right)^\alpha \Rightarrow S_{2,i} = S_{2,j}$$

Finally, $\mathsf{ek}_{\mathsf{U}_i} = S_{1,i} \cdot S_{2,i} = S_{1,j} \cdot S_{2,j} = \mathsf{ek}_{\mathsf{U}_j}$, so $\mathsf{pk}_{\mathsf{U}_i} = \mathsf{pk}_{\mathsf{U}_j}$, which concludes the proof of the claim.

Conclusion: Let $\mathcal{A}$ be a PPT adversary against the reporter anonymity of our scheme with advantage $\epsilon_\mathcal{A}(\lambda)$. Let $\epsilon_{\mathsf{NIZK}}(\lambda)$ be the maximum knowledge error among all the non-interactive proofs and signatures of knowledge used in the protocol. We have:

$$\epsilon_\mathcal{A}(\lambda) \leq \epsilon_{\mathsf{NIZK}}(\lambda).$$

We deduce that $\epsilon_\mathcal{A}(\lambda)$ is negligible, which concludes the proof. □

Lemma 8. *If the NIZKP and the SoK are extractable and zero-knowledge, then the protocol is reporter anonymous.*

Proof. Let $\mathcal{A}$ be a PPT adversary against the reporter anonymity of our scheme with advantage $\epsilon_{\mathcal{A}}(\lambda)$. Let $\epsilon_{\mathsf{NIZK}}(\lambda)$ be the maximum knowledge error among all the non-interactive proofs and signatures of knowledge used in the protocol. We use the following sequence of games:

Game G_0: This game is the reporter anonymity experiment as described in Definition 13 on our protocol. We have:

$$\epsilon_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A} \text{ wins } G_0] - 1/2|$$

Game G_1: This game is similar to G_0 except that the challenger sets an empty dictionary SIG at the beginning of the experiment, and each time that the challenger generates a signature σ for a tracer key $\mathsf{pk_T}$, a message m and a ring R with the secret key $\mathsf{sk_U}$, it stores the element $\alpha \in \mathbb{Z}_p^*$ (according to the notation we use in the definition of the signature algorithm) and $\mathsf{sk_U}$ at the key $(\mathsf{pk_T}, m, R, \sigma)$ of the dictionary SIG:

$$\mathsf{SIG}[(\mathsf{pk_T}, m, R, \sigma)] \leftarrow (\alpha, \mathsf{sk_U}).$$

This modification does not alter the advantage of $\mathcal{A}$:

$$\Pr[\mathcal{A} \text{ wins } G_0] = \Pr[\mathcal{A} \text{ wins } G_1]$$

Game G_2: This game is similar to G_1 except that each time that the challenger verifies a valid signature σ on a message m and the ring R forged by $\mathcal{A}$, it uses the knowledge extractor on each proof of knowledge and each signature of knowledge in σ in order to recover α and $\mathsf{sk_U}$, and stores the witness extracted from the signature of knowledge in the dictionary at the key $(\mathsf{pk_T}, m, R, \sigma)$:

$$\mathsf{SIG}[(\mathsf{pk_T}, m, R, \sigma)] \leftarrow (\alpha, \mathsf{sk_U}).$$

If one extraction returns a false witness, the challenger aborts the game and returns 0 with probability $1/2$. Let n be the number of users, and q_r be the number of call to the oracle $\mathcal{O}\mathsf{report}$. Each signature contains n proofs/signatures of knowledge. The challenger verifies q_r signatures, so the probability that the challenger aborts the game is the probability that the extractor fails on a valid proof, which is at most $q_r \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda)$. We deduce that:

$$|\Pr[\mathcal{A} \text{ wins } G_1] - \Pr[\mathcal{A} \text{ wins } G_2]| \leq q_r \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda).$$

At this step, if the challenger does not abort the game, then all the proofs and signatures prove valid statements. Let $\sigma = (h, c, (c_i, \pi'_i)_{i=1}^N, \sigma_M)$ be a valid signature on a message m and the ring $R = \{\mathsf{pk}_i\}_{i=1}^N$, where $\mathsf{pk}_i = (\mathsf{ek}_i, \pi_i)$ for all indexes i. The proofs of knowledge $\{\pi'_i\}_{i=1}^N$ ensure that there exists α such that $h = g^\alpha$ and for all There exists α such that $h = g^\alpha$ and for all $i \in [\![2, N]\!]$:

$$\frac{c_i}{c_{i-1}} = \left(\frac{\mathsf{ek}_i}{\mathsf{ek}_{i-1}}\right)^\alpha.$$

We deduce that for all $i \in [\![2, N]\!]$:

$$\prod_{j=2}^{i} \frac{c_j}{c_{j-1}} = \prod_{j=2}^{i} \left(\frac{\mathsf{ek}_j}{\mathsf{ek}_{j-1}}\right)^{\alpha} \Rightarrow \frac{c_i}{c_1} = \left(\frac{\mathsf{ek}_j}{\mathsf{ek}_1}\right)^{\alpha} \Rightarrow \frac{c_i}{\mathsf{ek}_i^{\alpha}} = \frac{c_1}{\mathsf{ek}_1^{\alpha}}$$

Since for all $i \in [\![N]\!]$, $\mathsf{ek}_i^{\alpha} = g^{\mathsf{sk}_i \cdot \alpha} = h^{\mathsf{sk}_i}$, we deduce that:

$$\frac{c_i}{h^{\mathsf{sk}_i}} = \frac{c_1}{h^{\mathsf{sk}_1}}.$$

Game G_3: In this game, the challenger simulates each proof and signature of knowledge using the corresponding simulators. Since the proofs and signatures of knowledge are perfectly zero-knowledge, we have:

$$\Pr[\mathcal{A} \text{ wins } G_2] = \Pr[\mathcal{A} \text{ wins } G_3].$$

Game G_4: This game is the same as G_3 except that instead of computing $\mathtt{Rep} \leftarrow \mathsf{Report}(pp, \mathsf{pk}_\mathsf{T}, \mathsf{sk}_{\mathsf{U}_b}, m, R, \sigma)$, the challenger retrieves $(\alpha, \mathsf{sk}_\mathsf{U}) \leftarrow \mathsf{SIG}[(\mathsf{pk}_\mathsf{T}, m, R, \sigma)]$, parses R as $\{\mathsf{pk}_i\}_{i=1}^n$ and each pk_i as (ek_i, π_i), finds the index j such that $\mathsf{pk}_j = \mathsf{pk}_{\mathsf{U}_b}$ and computes S_2 as follows, using the same notation as in the description of the protocol:

$$S_2 \leftarrow \frac{c_1}{\mathsf{ek}_1^{\alpha}}.$$

Since:

$$\frac{c_j}{h^{\mathsf{sk}_j}} = \frac{c_1}{h^{\mathsf{sk}_1}} = \frac{c_1}{g^{\alpha \cdot \mathsf{sk}_1}} = \frac{c_1}{\mathsf{ek}_1^{\alpha}},$$

we deduce that:

$$\Pr[\mathcal{A} \text{ wins } G_3] = \Pr[\mathcal{A} \text{ wins } G_4].$$

In G_4, the bit b is ever used by the challenger, which implies that:

$$\Pr[\mathcal{A} \text{ wins } G_4] = 1/2.$$

Conclusion: Finally, we have:

$$\epsilon_{\mathcal{A}}(\lambda) \leq q_r \cdot n \cdot \epsilon_{\mathsf{NIZK}}(\lambda).$$

We deduce that $\epsilon_{\mathcal{A}}(\lambda)$ is negligible, which concludes the proof. □