Report and Trace Ring Signatures

Ashley Fraser[1,2(✉)⋆] and Elizabeth A. Quaglia[3]

[1] Information Security Group, Royal Holloway, University of London, UK
[2] Department of Computer Science, University of Surrey, UK
a.fraser@surrey.ac.uk
[3] Information Security Group, Royal Holloway, University of London, UK
Elizabeth.Quaglia@rhul.ac.uk

Abstract. We introduce *report and trace* ring signature schemes, balancing the desire for signer anonymity with the ability to report malicious behaviour and subsequently revoke anonymity. We contribute a formal security model for report and trace ring signatures that incorporates established properties of anonymity, unforgeability and traceability, and captures a new notion of reporter anonymity. We present a construction of a report and trace ring signature scheme, proving its security and analysing its efficiency, comparing with the state of the art in the accountable ring signatures literature. Our analysis demonstrates that our report and trace scheme is efficient, particularly for the choice of cryptographic primitives that we use to instantiate our construction. We contextualise our new primitive with respect to related work, and highlight, in particular, that report and trace ring signature schemes protect the identity of the reporter even after tracing is complete.

Keywords: ring signatures, accountability, security model, construction

1 Introduction

Group signatures [8] and ring signatures [20] provide signers with anonymity within a set of users. Anonymity is a sought-after property, yet, under certain circumstances, it is also desirable to provide a guarantee of *traceability*, which means that anonymity can be revoked. This presents an interesting problem: how does a group or ring signature guarantee anonymity *and* traceability?

Group signatures rely on a trusted group manager to achieve these conflicting aims. The group manager determines the members of the group and issues key pairs to group members. Signers are anonymous within the group, but the group manager can learn the identity of signers and revoke anonymity. On the other

⋆ The author conducted the majority of this work at Royal Holloway, University of London and was supported by the EPSRC and the UK government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway under grant number EP/P009301/1. This work was also partly supported by the EPSRC Next Stage Digital Economy Centre in the Decentralised Digital Economy (DECaDE) under grant number EP/T022485/1.

hand, ring signatures do not rely on a trusted manager. In fact, signers generate their key pairs and select the group of users, known as the ring, within which the signer is anonymous. The solution to achieving anonymous and traceable ring signatures is *accountable* ring signatures [25, 6], which define a designated tracer who can identify signers. Accountable ring signatures retain the versatility of ring signatures, allowing signers to generate their keys and select the anonymity ring, and additionally allow signer anonymity to be revoked.

In practice, to begin the tracing process, the designated tracer in an accountable ring signature will often receive a report of malicious behaviour from a reporter. However, the reporter is outside the scope of the syntax and security model of accountable ring signatures. Consequently, it is implicit that the tracer must be trusted not to revoke anonymity without first receiving a report. Moreover, by omitting the role of the reporter from the security model, it is not possible to make any formal statements about the privacy of the reporter.

To address this, we introduce a new type of ring signature, which we call a *report and trace* ring signature. The underlying idea of report and trace is that a designated tracer can revoke anonymity of a signer if and only if a report of malicious behaviour is made by a user. In other words, a user reports a malicious message to the tracer, and the tracer must receive a report to revoke anonymity of the signer. Accordingly, report and trace achieves the balance between anonymity and traceability of accountable ring signatures, ensuring that the anonymity of a signer is preserved until tracing is complete. Additionally, report and trace incorporates a reporting system that preserves the anonymity of the signer *and* the reporter.

1.1 Related Work

Group signatures were introduced in [8], and the first security models were presented in [2, 3]. Generally, the group manager can revoke the anonymity of a signer and must be trusted to preserve signer anonymity in the absence of malicious behaviour. Several variants of group signatures have been proposed to limit trust in the group manager and protect the anonymity of non-malicious signers. For example, accountable tracing signatures [15] require that the group manager produce a proof of correct tracing and, if tracing occurred, a proof denying tracing cannot be produced. Traceable signatures [14] define a designated authority that can trace all signatures produced by a particular signer if the group manager first provides the authority with a tracing token related to that signer. In this way, the anonymity of non-malicious signers is protected; the group manager need not revoke anonymity of *all* signers to determine which signatures were produced by the signer in question. Furthermore, group signatures with message dependent opening (MDO signatures) [21] allow the group manager to revoke the anonymity of all signers that produced a signature for a particular message if and only if a reporter first produces a report related to that message. Our report and trace ring signature provides a similar distributed tracing function, but, in our setting, the report is attached to a signature rather than a message. Additionally, MDO signatures define the reporter to be a fixed entity with a

secret key generated during setup. Report and trace ring signatures, on the other hand, model reporters as system users, and our security model ensures anonymity of the reporter. Finally, we note that report and trace is a variation of a *ring* signature and, as such, does not rely upon a trusted group manager to issue key pairs to users and allows users to select their anonymity ring.

Ring signatures were first formally defined in [20] and a security model for ring signatures was presented in [4]. Following this, numerous variations of ring signatures have appeared (see, for example, [24] for a survey of some of these variations). Specifically, a number of ring signature variants offer some notion of traceability. For instance, linkable [17] and traceable [11] ring signatures provide limited tracing functionality, allowing two signatures generated by the same signer to be linked. In particular, linkable ring signatures determine whether two signatures are created by the same signer, without revealing the identity of the signer, and traceable ring signatures link ring signatures created by the same signer with respect to the same 'tag', and reveal the identity of the signer. Moreover, accountable ring signatures, introduced in [25] and formalised in [6], allow revocation of signer anonymity by a designated tracer and are, as a result, most closely related to our work. In fact, report and trace ring signatures can be viewed as an extension of accountable ring signatures, where the role of the tracer is distributed and the reporter is modelled as an anonymous system user.

A closely related line of work is purpose-built reporting systems [1, 16, 19]. Analogously to our work, these systems allow a user to report another user and subsequently allow revocation of anonymity by a designated tracer. However, unlike our report and trace scheme, these systems are stand-alone reporting systems. Specifically, their design allows a user to identify an individual that has, for example, harassed or assaulted the user, hiding the identity of the accused and reporter until a threshold of reports related to the accused are submitted, at which point a tracer reveals the identity of the accused and the reporter(s). We note that, critically, these systems require a *threshold* of reports to revoke anonymity of the accused. This design decision empowers reporters, allowing them to submit accusations with the confidence that they will remain anonymous unless (or until) a number of other reporters have come forward. Finally, in [23], a report and trace scheme was introduced in the context of end-to-end encrypted messaging. In such systems, a message receiver can report a malicious message to a designated tracer, and the tracer can revoke anonymity of the sender. The tracer learns nothing about the sender unless a report is provided by the recipient of that message, and the identity of the reporter is revealed only to the tracer, albeit the reporter's identity is known to the tracer before tracing is complete.

1.2 Our Contributions

We define syntax and a security model for report and trace (R&T) ring signatures (§2). Our syntax defines a reporting user who provides the tracer with a reporter token, recovered from a signature, and a designated tracer who uses the reporter token to revoke anonymity of the signer. Our security model extends the generic definitions of correctness, anonymity and unforgeability for ring signatures defined

in [4] to capture ring signatures with a report and trace functionality. Furthermore, we define *traceability*, adapting the security properties of an accountable ring signature to our setting. We complete our security model with a new definition of *reporter anonymity* for report and trace ring signatures, which ensures that the reporter is anonymous even *after* tracing is complete.

We demonstrate feasibility of report and trace by providing a construction of an R&T ring signature scheme that relies on standard cryptographic primitives (§3). Briefly, the signer provably encrypts their identity under the tracer's public key for a public-key encryption scheme and then encrypts the resulting ciphertext using a one-time key-pair for a public-key encryption scheme. Additionally, the signer provable encrypts the one-time decryption key (which we call the reporter token) to all potential reporters. Then, the reporter decrypts their token, and the tracer requires the reporter token to recover the signer's identity. Our construction is based on the accountable ring signature of [6] in which the signer provable encrypts their identity under the tracer's public key and the tracer can revoke the signer's anonymity by decrypting the resulting ciphertext. We choose this construction due to its efficiency and because its security relies upon standard, well-understood cryptographic hardness assumptions (namely, the decisional Diffie-Hellman assumption). Furthermore, this approach allows us to clearly demonstrate the additional cost of reporting. We prove that our construction is correct, anonymous, unforgeable, traceable and reporter anonymous, and our proof of security relies on standard notions of security for the cryptographic primitives used in our construction.

We analyse the efficiency of our construction (§3.3), summarising the computational and communication costs associated with signing, reporting and tracing for our scheme. We provide an instantiation of our construction (§3.2), which demonstrates that it can be implemented efficiently. In fact, for the cryptographic primitives we select, our construction performs favourably to the accountable ring signature of [6], and the additional cost of reporting is small.

Finally, we extend our construction to support multiple reporters (§4) using threshold publicly verifiable secret sharing [22]. We provide each potential reporter with a share of the reporter token, and a threshold of shares are required to recover the reporter token. We conclude with an efficiency analysis for our multiple reporter construction.

1.3 Contextualising R&T Ring Signatures

In this paper, we introduce a new primitive, an R&T ring signature, and provide a way to achieve it. We are also interested in placing this primitive in the context of related schemes and in highlighting the advantages it brings. We explore this next and summarise our findings in Table 1.

Revoking Anonymity of the Accused. All primitives with tracing functionality discussed so far [1, 16, 19, 23, 14, 15, 21, 6, 25] hide the identity of the accused (i.e., the signer in an R&T ring signature schemes) until tracing is complete, at

which point, anonymity of the accused is revoked. We note that [1, 16, 19, 23] reveal the identity of the accused only to the tracer. However, for accountable ring signatures schemes [25, 6], group signature variants [14, 15] and our R&T ring signature, anonymity of the accused is *publicly* revoked to allow for public verification of the tracing process. Accordingly, the tracer is *accountable* for their actions and can only (provably) revoke the anonymity of a real accused user.

Entities Revoking Anonymity. To complete tracing, every primitive we consider [1, 16, 19, 23, 14, 15, 21, 6, 25] requires a designated tracer. In some systems, e.g., [1, 16, 25], the tracer is distributed. Whilst our R&T ring signature construction (§3), and our multiple reporter construction (§4), model the tracer as a single entity, we remark that we can also distribute the tracer, thus distributing trust amongst a set of tracers. Trust in the tracer can be further reduced by requiring a reporter. Our R&T ring signature and [23, 1, 16, 19, 21] define a reporter such that the reporter and tracer must cooperate to revoke anonymity. Additionally, purpose-built tracing systems [1, 16, 19] require a *threshold* of reports to trigger the tracing process. We provide both options: our R&T ring signature construction (§3) requires a single report; our multiple reporter construction (§4) requires a threshold of reports.

Anonymity of the Reporter. Our (single reporter) R&T scheme ensures that the reporter is anonymous even *after* tracing. This is not true in the context of MDO signatures [21], where the reporter is a fixed, publicly-known, entity. Also, for end-to-end encrypted messaging [23], the tracer learns the identity of the reporter *before* starting the tracing process. Moreover, purpose-build reporting systems [1, 16, 19] intentionally reveal the identity of reporters after tracing. Recall that reporting systems allow reporters to communicate the identity of an accused person (e.g., a person accused of assault or illegal activity). Therefore, to follow up on allegations, revealing the reporter's identity is necessary. As the tracer in our R&T scheme does not require the identity of the reporter to follow up on an allegation (in fact, the allegation is that the message signed by the accused is malicious, and the message is public), we can protect the reporter's anonymity even after tracing. This empowers reporters to report malicious signers without fear of identity exposure.

Integrated Functionality. Finally, we highlight that, in comparison to purpose-built reporting systems [1, 16, 19], our R&T scheme has integrated functionality. That is, our R&T scheme is a ring signature scheme with a report and trace function. Similarly, several primitives build a tracing function atop a group or ring signature [14, 15, 21, 6, 25], and traceable end-to-end encrypted messaging [23] incorporates a tracing function into an end-to-end encrypted messaging scheme.

Application. We illustrate the usefulness of report and trace by describing a potential application. Consider a forum platform and a set of registered users that

	Publicly verifiable tracing	Entities revoking anonymity	Reporter Anonymity	Integrated functionality
Group signature variants [14, 15]	✓	Tracer	N/A	Signature
Group signature with message dependent opening [21]	✗	Reporter Tracer	✗	Signature
Accountable ring signatures [6, 25]	✓	Tracer	N/A	Signature
Traceable E2E encrypted messaging [23]	✗	Reporter Tracer	✗	Encryption
Reporting systems [1, 16, 19]	✗	Reporter (threshold) Tracer	✓*	None
R&T **ring signatures** (This work)	✓	Reporter Tracer	✓	Signature
R&T **ring signatures (multiple reporters)** (This work)	✓	Reporter (threshold) Tracer	✗	Signature

Table 1. Contextualising R&T ring signatures. $*$ denotes anonymity only holds until tracing is complete.

can post messages to the forum. Users may wish to post messages anonymously, while also providing a signature proving that they are a registered user. Moreover, if a user posts a malicious message, the platform may wish to hold the signer accountable. Certainly, standard group and ring signature facilitate the ability of a user to sign a message anonymously. Furthermore, group signatures and accountable ring signatures balance anonymity and traceability. However, we believe that R&T ring signatures provide a unique solution to this scenario. Firstly, R&T ring signatures (and group signatures with message dependent opening) do not rely solely on a designated tracer to revoke anonymity and, as such, provide additional protection for the signer's identity above that provided by accountable ring signatures and group signatures. Moreover, distributing the tracing function in a busy forum scenario reduces the burden on the tracer to check for malicious messages. Indeed, the tracer need only check messages for which the tracer receives a report. Additionally, our R&T signature allows the tracer to revoke anonymity only for the reported signature. That is, the signer preserves their anonymity with respect to all other signatures and no other signer who posts the same message will be de-anonymised. In our forum scenario, it may not be desirable to revoke anonymity for all signatures produced by the signer of a single malicious message. Moreover, it may be the case that a signed message is malicious in the context of which it is reported, but may be entirely innocuous in a different context. Consequently, R&T ring signatures are more appropriate than traceable signatures or MDO signatures for this setting. Finally, R&T ring signatures retain the versatility of ring signatures and define the reporter to be a system user, which can foster a sense of community responsibility for content posted on the forum, and provide a unique guarantee of anonymity for the reporter which can empower users to report malicious behaviour without fear of repercussions.

2 Syntax and Security

We introduce the syntax of a *report and trace* (R&T) ring signature scheme and accompanying security model. In a standard ring signature, users digitally sign messages with respect to a set of users, known as a ring. Ring signatures ensure that the signer cannot be identified; any ring member is equally likely to have produced the signature. R&T ring signatures extend this notion, allowing a signer to be identified if an anonymous report is made to a designated tracer, who then traces the signer.

Alongside a set of users $\mathcal{U}$, an R&T ring signature scheme involves the following entities. A *reporter* produces a report. Within our syntax and security model, reporters are ring members, though this need not be the case. A *designated tracer*, denoted T, revokes the signer's anonymity if the tracer received a report for the signature in question. Anybody can verify the correctness of the report and trace by running a public verification algorithm. Formally, we define an R&T ring signature in Definition 1.

Definition 1 (R&T ring signature). *An* R&T ring signature scheme *is a tuple of algorithms (*Setup*,* $\mathsf{T.KGen}$*,* $\mathsf{U.KGen}$*,* Sign*,* Verify*,* Report*,* Trace*,* $\mathsf{VerTrace}$*) such that*

- $\mathsf{Setup}(1^\lambda)$: On input security parameter 1^λ, Setup outputs public parameters pp.
- $\mathsf{T.KGen}(pp)$: On input pp, $\mathsf{T.KGen}$ outputs a tracer key pair $(pk_\mathsf{T}, sk_\mathsf{T})$. We write that $pk_\mathsf{T} \leftarrow \mathsf{T.KGen}(pp; sk_\mathsf{T})$.
- $\mathsf{U.KGen}(pp)$: On input pp, $\mathsf{U.KGen}$ outputs a user key pair $(pk_\mathsf{U}, sk_\mathsf{U})$. We write that $pk_\mathsf{U} \leftarrow \mathsf{U.KGen}(pp; sk_\mathsf{U})$.
- $\mathsf{Sign}(pp, sk_\mathsf{U}, pk_\mathsf{T}, m, R)$: On input pp, sk_U, pk_T, message m and ring R, Sign outputs a signature σ.
- $\mathsf{Verify}(pp, pk_\mathsf{T}, m, R, \sigma)$: On input pp, pk_T, m, R and σ, Verify outputs 1 if σ is a valid signature on m with respect to R, and 0 otherwise.
- $\mathsf{Report}(pp, pk_\mathsf{T}, sk_\mathsf{U}, m, R, \sigma)$: On input pp, pk_T, sk_U, m, R and σ, Report outputs a reporter token $\mathtt{Rep}$.
- $\mathsf{Trace}(pp, sk_\mathsf{T}, m, R, \sigma, \mathtt{Rep})$: On input pp, sk_T, m, R, σ and $\mathtt{Rep}$, Trace outputs the signer's identity pk_U, auxiliary information $\mathtt{Tr}$ consisting of the reporter token, and a proof of correct trace ρ_t.
- $\mathsf{VerTrace}(pp, pk_\mathsf{T}, m, R, \sigma, pk_\mathsf{U}, \mathtt{Tr}, \rho_t)$: On input pp, pk_T, m, R, σ, pk_U, $\mathtt{Tr}$ and ρ_t, $\mathsf{VerTrace}$ outputs 1 if the trace is valid, and 0 otherwise.

We define correctness for our syntax as the property that honestly generated signatures are verifiable.

Definition 2 (Correctness). *An* R&T *ring signature is* correct *if, for any* $n = \mathsf{poly}(\lambda)$*,* $j \in [n]$ *and message* m*, there exists a negligible function* negl *such that,*

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

2.1 Security Model

We present a security model for our syntax that incorporates accepted security properties from the ring signature literature. Firstly, we extend well-established definitions of anonymity and unforgeability for standard ring signature schemes, presented in [4], to our setting. Then, we cast the security requirements of an accountable ring signature into our syntax. Namely, we define traceability, which captures notions of trace correctness, non-frameability and tracing soundness defined in [6]. Finally, we present a definition of reporter anonymity, a new security property for our report and trace setting.

In Figure 1, we define a number of oracles for our security experiments. We write $\mathcal{O}\mathsf{X}_{(y_1,\ldots,y_n)}(z_1,\ldots,z_n)$ to denote oracle X that has access to parameters and sets $y_1,\ldots,y_n$ and takes as input $z_1,\ldots,z_n$. Oracles $\mathcal{O}\mathsf{reg}$, $\mathcal{O}\mathsf{corrupt}$ and $\mathcal{O}\mathsf{sign}$ operate as expected: they model registration of users, corruption of users, and signature generation respectively. Moreover, $\mathcal{O}\mathsf{report}$ is called to obtain a reporter token for a message and $\mathcal{O}\mathsf{trace}$ is called to trace the signer of a message.

Our security model considers entities (i.e., users, reporters and tracers) that are either honest, corrupt, or under the attacker's control. In detail, honest entities do not provide an attacker with secret keys; corrupt entities generate their keys honestly, but may later provide the attacker with their secret keys; the attacker can generate keys on behalf of controlled entities. An attacker that has credentials of users, reporters or tracers can generate signatures, reports or traces respectively.

$\underline{\mathcal{O}\mathsf{reg}_{(pp,\mathcal{Q}\mathsf{reg},L)}()}$

$(pk_\mathsf{U}, sk_\mathsf{U}) \leftarrow \mathsf{U.KGen}(pp)$
$\mathcal{Q}\mathsf{reg} \leftarrow \mathcal{Q}\mathsf{reg} \cup \{pk_\mathsf{U}\}$
$L \leftarrow L \cup \{(pk_\mathsf{U}, sk_\mathsf{U})\}$
return pk_U

$\underline{\mathcal{O}\mathsf{corrupt}_{(L,\mathcal{Q}\mathsf{corr})}(pk_\mathsf{U})}$

if $(pk_\mathsf{U}, \cdot) \notin L$ **return** $\perp$
$\mathcal{Q}\mathsf{corr} \leftarrow \mathcal{Q}\mathsf{corr} \cup \{pk_\mathsf{U}\}$
return sk_U

$\underline{\mathcal{O}\mathsf{sign}_{(pp,L,\mathcal{Q}\mathsf{sign})}(pk_\mathsf{U}, pk_\mathsf{T}, m, R)}$

if $(pk_\mathsf{U}, \cdot) \notin L$ **return** $\perp$
$\sigma \leftarrow \mathsf{Sign}(pp, sk_\mathsf{U}, pk_\mathsf{T}, m, R \cup \{pk_\mathsf{U}\})$
$\mathcal{Q}\mathsf{sign} \leftarrow \mathcal{Q}\mathsf{sign} \cup \{(pk_\mathsf{T}, pk_\mathsf{U}, m, R, \sigma)\}$
return σ

$\underline{\mathcal{O}\mathsf{report}_{(pp,pk_\mathsf{T},L,\mathcal{Q}\mathsf{report})}(pk_\mathsf{U}, m, R, \sigma)}$

if $pk_\mathsf{U} \notin R \vee (pk_\mathsf{U}, \cdot) \notin L$ **return** $\perp$
$\mathsf{Rep} \leftarrow \mathsf{Report}(pp, pk_\mathsf{T}, sk_\mathsf{U}, m, R, \sigma)$
$\mathcal{Q}\mathsf{report} \leftarrow \mathcal{Q}\mathsf{report} \cup \{(pk_\mathsf{U}, m, R, \sigma)\}$
return $\mathtt{Rep}$

$\underline{\mathcal{O}\mathsf{trace}_{(pp,sk_\mathsf{T},\mathcal{Q}\mathsf{trace})}(m, R, \sigma, \mathsf{Rep})}$

$(pk_\mathsf{U}, \mathtt{Tr}, \rho_t) \leftarrow \mathsf{Trace}(pp, sk_\mathsf{T}, m, R, \sigma, \mathsf{Rep})$
$\mathcal{Q}\mathsf{trace} \leftarrow \mathcal{Q}\mathsf{trace} \cup \{(m, R, \sigma)\}$
return $(pk_\mathsf{U}, \mathtt{Tr}, \rho_t)$

Fig. 1: Oracles used in the experiments for anonymity, unforgeability, traceability and reporter anonymity of an R&T ring signature scheme.

Anonymity. At an intuitive level, anonymity for an R&T ring signature is the property that a signature does not reveal the identity of the signer *unless* the signature is reported and the signer traced. Our formal definition of anonymity captures anonymity against adversarially generated keys as defined in [4]. As such, we assume that the adversary can corrupt and control users and reporters, but that the tracer is honest. We require that the adversary, when provided with a challenge signature, cannot determine which of two potential honest signers generated the signature, on the condition that the adversary does not obtain a trace for the challenge signature.

Definition 3 (Anonymity). *An* R&T *ring signature is* anonymous with respect to adversarially generated keys *if, for any probabilistic, polynomial-time (PPT) adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, there exists a negligible function* negl *such that,*

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda); \\ L,\ \mathcal{Q}\mathsf{reg},\ \mathcal{Q}\mathsf{corr},\ \mathcal{Q}\mathsf{sign},\ \mathcal{Q}\mathsf{report},\ \mathcal{Q}\mathsf{trace} \leftarrow \emptyset; \\ (pk_\mathsf{T}, sk_\mathsf{T}) \leftarrow \mathsf{T.KGen}(pp); \\ (m, R, pk_{\mathsf{U}_0}, pk_{\mathsf{U}_1}, st) \leftarrow \mathcal{A}_1^{\mathcal{O}}(pp, pk_\mathsf{T}); \\ b \leftarrow \{0,1\}; \\ \sigma \leftarrow \mathsf{Sign}(pp, sk_{\mathsf{U}_b}, pk_\mathsf{T}, m, R \cup \{pk_{\mathsf{U}_0}, pk_{\mathsf{U}_1}\}); \\ b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(\sigma, st) \end{array} : \begin{array}{l} b' = b \wedge (m,R,\sigma) \notin \mathcal{Q}\mathsf{trace} \\ \wedge\ \{pk_{\mathsf{U}_0}, pk_{\mathsf{U}_1}\} \subseteq \mathcal{Q}\mathsf{reg} \backslash \mathcal{Q}\mathsf{corr} \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

where $\mathcal{O} = \{\mathcal{O}\mathsf{reg}, \mathcal{O}\mathsf{corrupt}, \mathcal{O}\mathsf{sign}, \mathcal{O}\mathsf{report}, \mathcal{O}\mathsf{trace}\}$ *are the oracles defined in Figure 1.*

Unforgeability. We require that signatures are unforgeable. That is, an attacker cannot output a valid signature on behalf of an honest user, even if an attacker can trace the identity of honest signers through the report and trace functionality. Formally, we consider an unforgeability definition similar to that presented in [4]. Thus, in our unforgeability experiment, we assume that the adversary controls the tracer and can corrupt and control users and reporters. We require that the adversary cannot output a valid signature for a ring of honest users, where the signature is not the output of the signing oracle.

Definition 4 (Unforgeability). *An* R&T *ring signature scheme is* unforgeable *if, for any PPT adversary* $\mathcal{A}$*, there exists a negligible function* negl *such that,*

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

where $\mathcal{O} = \{\mathcal{O}\mathsf{reg}, \mathcal{O}\mathsf{corrupt}, \mathcal{O}\mathsf{sign}, \mathcal{O}\mathsf{report}\}$ *are the oracles defined in Figure 1.*

Traceability. R&T signatures must satisfy *traceability*. In other words, it must be possible to identify the signer of a message. Traceability comprises three conditions: trace correctness, non-frameability and trace soundness. *Trace correctness* requires that an honestly generated signature must be traceable to the correct signer. Accordingly, any trace output by the tracer must be valid. We capture trace

correctness in an experiment that requires an honestly generated report and trace for an honestly generated signature to verify. *Non-frameability* states that a report and trace mechanism cannot identify a non-signer as the signer. To this end, our non-frameability definition requires that the adversary, with control of the tracer and a subset of users, cannot output a valid trace such that the trace identifies a non-signer. Finally, *trace soundness*, defined in [6], stipulates that the signer identified by the report and trace mechanism is unique. That is, it is not possible to verifiably identify two users as the signer of a single message. The trace soundness definition in [6], which we cast into our syntax, considers an adversary that controls the tracer and can corrupt and control users and reporters. Trace soundness requires that the adversary cannot output two valid traces that identify two different signers for the same message.

Definition 5 (Traceability). *An* R&T *ring signature satisfies* traceability *if the following conditions are satisfied:*

1. Trace correctness: *for any* $n = \mathsf{poly}(\lambda)$, $j, k \in [n]$ *where* $j \neq k$, *and message* m, *there exists a negligible function* negl *such that,*

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

2. Non-frameability; *for any PPT adversary* $\mathcal{A}$, *there exists a negligible function* negl *such that,*

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

3. Trace soundness: *for any PPT adversary* $\mathcal{A}$, *there exists a negligible function* negl *such that,*

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

where $\mathcal{O} = \{\mathcal{O}\mathsf{reg}, \mathcal{O}\mathsf{corrupt}, \mathcal{O}\mathsf{sign}, \mathcal{O}\mathsf{report}\}$ *are the oracles defined in Figure 1.*

Reporter Anonymity. We define *reporter anonymity*, a new property that requires that a report does not reveal the ring member that produced it. We formally define reporter anonymity as the property that an adversary, when provided with a report for a signature, cannot determine which of two potential reporters produced the report. Our definition captures an adversary that can corrupt and control users and reporters, and controls the tracer. However, we require that the adversary does not corrupt either of the potential reporters and does not obtain a report through access to oracles.

Definition 6 (Reporter anonymity). *An* R&T *ring signature is* reporter anonymous *if, for any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, there exists a negligible function* negl *such that,*

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda); \\ L,\ \mathcal{Q}\mathsf{reg},\ \mathcal{Q}\mathsf{corr},\ \mathcal{Q}\mathsf{sign},\ \mathcal{Q}\mathsf{report} \leftarrow \emptyset; \\ (pk_\mathsf{T}, m, R, \sigma, pk_{\mathsf{U}_0}, pk_{\mathsf{U}_1}, st) \leftarrow \mathcal{A}_1^{\mathcal{O}}(pp); \\ b \leftarrow \{0,1\}; \\ \mathsf{Rep} \leftarrow \mathsf{Report}(pp, pk_\mathsf{T}, sk_{\mathsf{U}_b}, m, R, \sigma); \\ b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(\sigma, st) \end{array} : \begin{array}{l} b' = b \\ \wedge \{pk_{\mathsf{U}_0}, pk_{\mathsf{U}_1}\} \subseteq (R \cap \mathcal{Q}\mathsf{reg}) \backslash \mathcal{Q}\mathsf{corr} \\ \wedge\ (m, R, \sigma, pk_{\mathsf{U}_0}) \notin \mathcal{Q}\mathsf{report} \\ \wedge\ (m, R, \sigma, pk_{\mathsf{U}_1}) \notin \mathcal{Q}\mathsf{report} \end{array}\right]$$

$$\leq \tfrac{1}{2} + \mathsf{negl}(\lambda)$$

where $\mathcal{O} = \{\mathcal{O}\mathsf{reg}, \mathcal{O}\mathsf{corrupt}, \mathcal{O}\mathsf{sign}, \mathcal{O}\mathsf{report}\}$ *are the oracles defined in Figure 1.*

3 A Report and Trace Ring Signature Construction

We present an R&T ring signature construction, formally defined in Figure 2. Our construction requires a one-way function $f : X \to Y$ such that, given $y = f(x)$, it is hard to compute x, and a standard public key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.KGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ that is secure against chosen plaintext attacks (IND-CPA) [12]. We require a zero-knowledge proof system $\mathsf{NIZK} = (\mathsf{NIZK.Setup}, \mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ that satisfies completeness, knowledge soundness and zero-knowledge where such security definitions are drawn from [13]. Finally, we utilise a signature of knowledge $\mathsf{SOK} = (\mathsf{SoK.Setup}, \mathsf{SoK.Sign}, \mathsf{SoK.Verify})$ [7], that satisfies correctness, simulatability and extractability, all of which are defined in [6].

The idea behind our construction is as follows. The tracer and users obtain a key pair for a PKE scheme. The signer generates a fresh key pair for a PKE scheme, and the freshly generated decryption key (known as the reporter token in our construction) is encrypted to all members of the ring using a PKE scheme. The signer then uses a double layer of encryption to encrypt their public identity, which is generated via one-way function f. That is, the signer encrypts their public identity under the public key of the tracer, and encrypts the resulting ciphertext under the freshly generated encryption key. In this way, a reporter *and* the tracer are required to recover the identity of the signer. Indeed, any ring member can decrypt the reporter token, and the tracer requires the reporter token, along with their own decryption key, to remove the double-layer of encryption and revoke anonymity of the signer. Our construction additionally employs NIZK proofs and an SOK to ensure that operations are performed correctly, i.e., that the signer encrypts the correct public identity and reporter token, and that the reporter and tracer identify the correct signer.

Our construction is similar to the construction in [6], which provides an efficient accountable ring signature scheme that allows a designated tracer to revoke signer anonymity. In [6], the signer uses a PKE scheme to encrypt their public identity under the tracer's public key, and the tracer recovers the signer's identity by decrypting the ciphertext. This construction also relies on an SOK that allows the signer to prove that they have encrypted a public identity for

which they know a corresponding secret, and a NIZK proof such that the tracer can prove correct decryption, i.e., that they traced the correct signer. Our R&T construction differs from [6] in the following way. We require the encryption of a token to a set of reporters and provide a NIZK proof of correct encryption. Additionally, we use a double-layer of encryption, which is crucial to ensuring that the tracer cannot decrypt the signer's identity without a reporter token.

3.1 Description of Our Construction

We now describe the details of our construction. A trusted third party runs Setup, performing setup for the PKE, NIZK and SOK schemes. We assume that the public parameters generated for each scheme defines the public/secret key, randomness and message spaces (which we denote respectively as PK, SK, Rand and M) as appropriate. T.KGen generates a tracer key pair for a PKE scheme, and U.KGen is run to generate user key pairs. In particular, users generate a signing/verification key pair $(pk_{\mathsf{RS}}, sk_{\mathsf{RS}})$ using one-way function f, and a key pair $(pk_{\mathsf{PKE}}, sk_{\mathsf{PKE}})$ for a PKE scheme.

To sign a message m with respect to a ring R, the signer runs algorithm Sign. The signer generates a key pair $(pk_{\mathsf{Sign}}, sk_{\mathsf{Sign}})$ for a PKE scheme and encrypts the reporter token sk_{Sign} to each member of the ring (i.e., encrypts sk_{Sign} under the public encryption key of each ring member). The signer proves that each PKE ciphertexts encrypt the reporter token sk_{Sign} associated with pk_{Sign}, which is included in the signature. That is, the signer provides a NIZK proof for the following relation:

$$\mathcal{R}_{\mathsf{Enc}} = \left\{ \begin{array}{c} (pp,(pk_{\mathsf{Sign}},pk_{\mathsf{PKE}_i},\mathbf{c_1}),(r_{1,1},\ldots,r_{1,|R|},sk_{\mathsf{Sign}})) \ : \ pk_{\mathsf{Sign}} \coloneqq \mathsf{PKE.KGen}(pp_{\mathsf{PKE}};sk_{\mathsf{Sign}}) \\ \wedge \ \{\forall \ i \in 1,\ldots,|R| \ : \ c_{1,i} \coloneqq \mathsf{PKE.Enc}(pp_{\mathsf{PKE}},pk_{\mathsf{PKE}_i},sk_{\mathsf{Sign}};r_{1,i})\} \end{array} \right\} \quad (1)$$

Then, the signer's verification key pk_{RS} is encrypted under the tracer's public key, resulting in ciphertext c_2, which is then encrypted under the freshly generated public key pk_{Sign}, giving ciphertext c_3. Finally, the signer produces a signature of knowledge, which proves that c_3 encrypts a verification key in the ring such that the signer knows the associated signing key. The signature of knowledge is associated with the following relation:

$$\mathcal{R}_{\mathsf{SOK}} = \left\{ \begin{array}{c} (pp,(pk_{\mathsf{T}},pk_{\mathsf{Sign}},R,c_3),(r_2,r_3,sk_{\mathsf{RS}}),m) \ : \ c_3 \coloneqq \mathsf{PKE.Enc}(pp_{\mathsf{PKE}},pk_{\mathsf{Sign}},c_2;r_3) \\ \wedge \ c_2 \coloneqq \mathsf{PKE.Enc}(pp_{\mathsf{PKE}},pk_{\mathsf{T}},pk;r_2) \ \wedge \ pk \coloneqq f(sk_{\mathsf{RS}}) \in R \end{array} \right\} \quad (2)$$

To report a message, a member of the ring runs Report to decrypt the reporter token. The reporter additionally provides a proof of correct decryption, without revealing which member of the ring decrypted the token. This is given by the following relation:

$$\mathcal{R}_{\mathsf{Dec}_r} = \left\{ \begin{array}{c} (pp,(R,\mathbf{c_1},sk_{\mathsf{Sign}}),sk_{\mathsf{PKE}}) \ : \ sk_{\mathsf{Sign}} \coloneqq \mathsf{PKE.Dec}(pp_{\mathsf{PKE}},sk_{\mathsf{PKE}},c_{1,i}) \\ \wedge \ c_{1,i} \in \mathbf{c_1} \ \wedge \ pk_{\mathsf{PKE}} \coloneqq \mathsf{PKE.KGen}(pp_{\mathsf{PKE}};sk_{\mathsf{PKE}}) \in R \end{array} \right\} \quad (3)$$

On receipt of a report, the tracer runs Trace to decrypt ciphertexts c_3 and c_2, thus revealing the signer's verification key. As sk_{Sign} is included in the report,

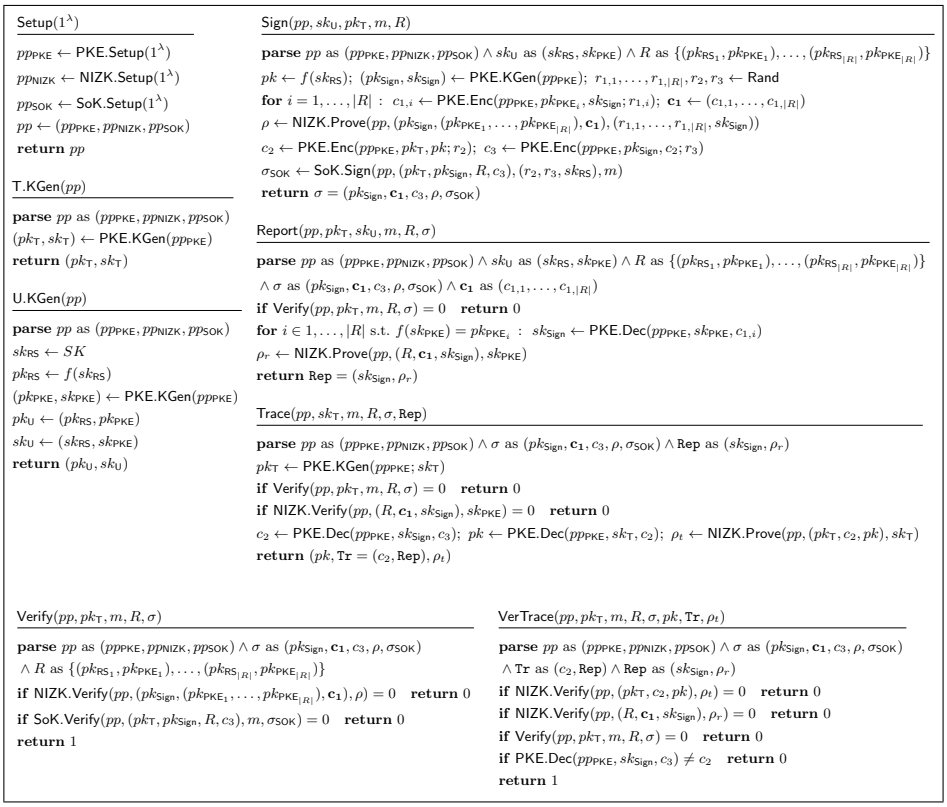

Fig. 2: Our R&T ring signature construction.

anyone can decrypt c_3, hence checking correct decryption directly. As such, the tracer need only prove correct decryption of c_2, which is given by the following relation:

$$\mathcal{R}_{\mathsf{Dec}_t} = \left\{ \begin{matrix} (pp, (pk_{\mathsf{T}}, c_2, pk_{\mathsf{RS}}), sk_{\mathsf{T}}) \ : \ pk \coloneqq \mathsf{PKE.Dec}(pp_{\mathsf{PKE}}, sk_{\mathsf{T}}, c_2) \\ \wedge \ pk_{\mathsf{T}} \coloneqq \mathsf{PKE.KGen}(pp_{\mathsf{PKE}}; sk_{\mathsf{T}}) \end{matrix} \right\} \tag{4}$$

Our construction additionally provides a public signing verification algorithm Verify, which ensures that the signer provides an encryption of their own public key, enabling tracing if the message is malicious. Moreover, a public trace verification algorithm VerTrace ensures that the correct signer is traced.

We prove that our construction satisfies correctness, anonymity, unforgeability, traceability and reporter anonymity as defined in Section 2. We obtain Theorem 1, which we formally prove in Appendix A.

Theorem 1. *The construction in Figure 2 satisfies correctness, anonymity, unforgeability, traceability and reporter anonymity as defined in Definitions 2– 6.*

3.2 Instantiating Our Construction

Our construction can be instantiated with ElGamal encryption and the signature of knowledge of [6], modified to account for the double layer of encryption of the signer's identity. Additionally, we instantiate our NIZK protocols for signing, reporting and tracing with various Σ-protocols, and outline these below. For our choice of cryptographic primitives, our instantiation is secure. Indeed, our chosen schemes satisfy the security requirements for the construction to be secure, and we provide sketch proofs of these results in Appendix B.

Setup and Key Generation. We let $pp_{\mathsf{PKE}} = pp_{\mathsf{NIZK}} = (\mathbb{G}, g, q)$ where $\mathbb{G}$ is a cyclic group of order q with generator g. Moreover, $pp_{\mathsf{SOK}} = (ek, ck)$ where ek is an ElGamal encryption key and ck is a key for a commitment scheme. We define one-way function f to perform group exponentiation such that $f(x) = g^x$ for some $x \in \mathbb{Z}_q$. We write $pp = (pp_{\mathsf{PKE}}, pp_{\mathsf{SOK}})$ as the output of algorithm Setup. We write the key pair for the tracer as $(pk_{\mathsf{T}} = g^{sk_{\mathsf{T}}}, sk_{\mathsf{T}})$ and the key pair for the user as $(pk_{\mathsf{U}}, sk_{\mathsf{U}}) = ((g^{sk_{\mathsf{RS}}}, g^{sk_{\mathsf{PKE}}}), (sk_{\mathsf{RS}}, sk_{\mathsf{PKE}}))$ where $sk_{\mathsf{T}}, sk_{\mathsf{RS}}, sk_{\mathsf{PKE}} \in \mathbb{Z}_q$.

Sign. We use a standard ElGamal encryption scheme to generate ciphertext c_1 and to double-encrypt the signer's identity in algorithm Sign. That is, we define ciphertexts $c_2 = (A_2, B_2) = (g^{r_2}, pk_{\mathsf{T}}^{r_2} \cdot pk)$ and $c_3 = (A_2, A_3, B_3) = (g^{r_2}, g^{r_3}, pk_{\mathsf{Sign}}^{r_3} \cdot pk_{\mathsf{T}}^{r_2} \cdot pk)$.To generate proof ρ for the relation in Equation 1, we use the Σ-protocol of [22], defined in Figure 3, which shows that c_1 encrypts the discrete log of a public element pk_{Sign} and can be transformed into a NIZK proof using the Fiat-Shamir transform [10].

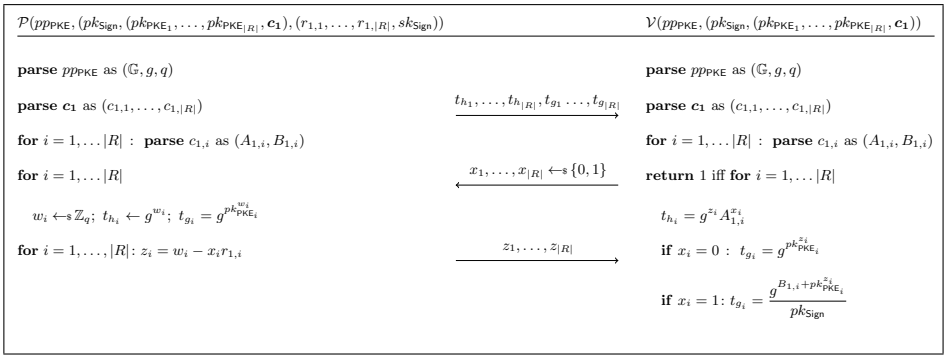

Fig. 3: A Σ-protocol for proving that ciphertext $\boldsymbol{c_1}$ encrypts a discrete log of pk_{Sign}.

Finally, we modify the SOK of [6] to generate σ_{SOK}, which proves that the signer knows sk_{RS} such that $pk_{\mathsf{RS}} = g^{sk_{\mathsf{RS}}}$ is an element of the ring. Our modification accounts for the double-layer of encryption used in our construction, rather than the single ElGamal encryption required in the accountable ring

signature construction of [6] and we obtain the Σ-protocol for Equation 2 in Figure 4. We note that this protocol, and the Σ-protocol of [6], relies on two additional Σ-protocols, one to prove that a commitment opens to a sequence of bits which contains a single 1, and a second to prove that a list of ElGamal ciphertexts contains an encryption of 1. These Σ protocols are required to prove that the signer knows the signing key corresponding to a verification key in the ring, without revealing their identity in the ring. We can use the additional Σ-protocols from [6] without any adjustments, and refer the reader to [6, Figs. 4&5] for full details.

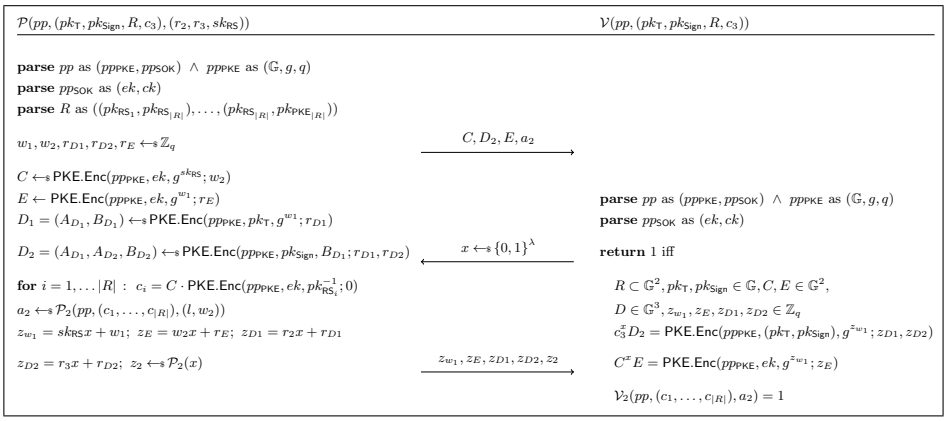

Fig. 4: A modified Σ-protocol for $\mathcal{R}_{\mathsf{SOK}}$ from [6].

Report and Trace. The reporter runs $\mathsf{PKE.Dec}$ to generate sk_{Sign}. For our instantiation, the reporter need not generate a proof of knowledge of correct decryption for the relation in Equation 3. In fact, as $\mathbb{G}$ is a cyclic group, $g^i = g^j$ if $i = j \mod q$. Therefore, pk_{Sign} is uniquely defined by $g^{sk_{\mathsf{Sign}}}$. The tracer (and any public verifier) can trivially check correct decryption of the reporter token by computing a single group exponentiation, i.e., check $g^{sk_{\mathsf{Sign}}} = pk_{\mathsf{Sign}}$ where sk_{Sign} is returned by the reporter and pk_{Sign} is included in the signature. Then, the tracer decrypts the signer's identity by computing $B_3/(A_2^{sk_{\mathsf{T}}} A_3^{sk_{\mathsf{Sign}}})$ for $c_3 = (A_2, A_3, B_3)$. The tracer can prove correct decryption of ciphertext c_2 using a standard Σ-protocol as outlined in Figure 5.

3.3 Efficiency of Our Construction

Here, we discuss the efficiency of our construction, showing that our construction incurs reasonable costs with respect to the functionality provided and that our proposed instantiation is practical. We also highlight the additional costs associated with our report and trace functionality by comparing with the accountable ring signature in [6]. We show that our construction compares favourably to

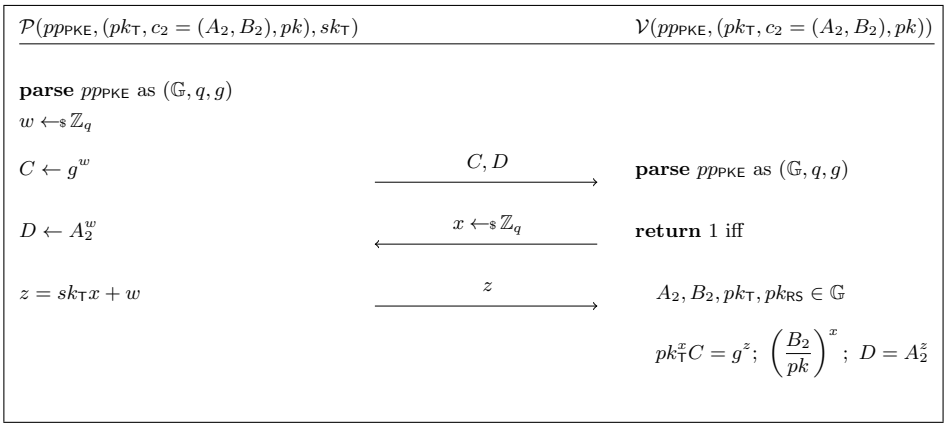

Fig. 5: A Σ-protocol for proving correct decryption of an ElGamal ciphertext.

the accountable ring signature construction of [6]: indeed, we require additional computation, but these computations are minimal considering the extra functionality provided by our R&T construction. We summarise the computation and communication costs of our generic construction and instantiation in Tables 2 and 3, respectively. We now briefly describe the costs incurred by the signer, reporter, tracer and verifier.

Signer. The signer's computation costs are dominated by the SOK and PKE computations, both of which grow linearly in the size of the ring. In comparison to the accountable ring signature of [6], we see that encrypting a reporter token doubles the computation costs for the signer. Moreover, the size of the signature also increases, requiring the communication of a number of group and field elements that grow linearly in the size of the ring (whereas the accountable ring signature communicates a constant number of group elements).

Reporter. The computation and communication costs incurred by the reporter, which is unique to our construction, are minimal. In fact, for our instantiation, the reporter need only perform a single decryption (i.e., 1 group exponentiation). As such, a report consists of a single field element. This demonstrates that, though our generic construction allows for the case where a reporter proves correct decryption, by basing our instantiation on cyclic group operations, it is possible to provide an efficient instantiation in which the computation costs of the reporter are minimised.

Tracer. The tracer's costs are small and compare favourably to accountable ring signatures. Indeed, computation of the trace requires a constant number of group exponentiations, calling for only 2 additional group exponentiation when compared to accountable ring signatures. In particular, to verify correct decryption of the reporter token, the tracer need only perform a single group exponentiation, rather than verifying a NIZK proof (cf. §3.2). Furthermore, the

		Accountable ring signature [6]	**Our R&T construction** (Fig.3)	**R&T with multiple reporters** (§5)
Sign	Comp.	1 PKE.Enc 1 SoK.Sign	$\|R\|+2$ PKE.Enc 1 NIZK.Prove ($\|R\|$ enc) 1 SoK.Sign	$\|R\|+2$ PKE.Enc $\|R\|$ NIZK.Prove (enc) 1 SoK.Sign $\|R\|$ public share gen
	Comm.	1 PKE ciphertext 1 SOK	$\|R\|+1$ PKE ciphertext 1 NIZK proof ($\|R\|$ enc) 1 SOK 1 element pk_{Sign}	$\|R\|+1$ PKE ciphertext $\|R\|$ NIZK proof (enc) 1 SOK $\|R\|$ PVSS public shares 1 element S
Verify	Comp.	1 SoK.Verify	1 SoK.Verify 1 NIZK.Verify ($\|R\|$ enc)	1 SoK.Verify $\|R\|$ NIZK.Verify (enc) 1 SS.Verify
	Comm.	N/A	N/A	N/A
Report	Comp.	N/A	1 PKE.Dec 1 NIZK.Prove(dec)	1 PKE.Dec 1 NIZK.Prove(dec)
	Comm.	N/A	1 token sk_{Sign} 1 NIZK proof (dec)	1 sub-token s_i 1 NIZK proof (dec)
Trace	Comp.	1 PKE.Dec 1 NIZK.Prove (dec)	2 PKE.Dec 1 NIZK.Prove (dec) 1 NIZK.Verify(dec)	2 PKE.Dec 1 NIZK.Prove (dec) t NIZK.Verify(dec) 1 SS.Combine
	Comm.	1 pk of signer 1 NIZK proof (dec)	1 pk of signer 1 token sk_{Sign} 1 PKE ciphertext 2 NIZK proof (dec)	1 pk of signer t sub-tokens $s_1, \ldots, s_t$ 1 PKE ciphertext 2 NIZK proof (dec)
VerTrace	Comp.	1 NIZK.Verify (dec)	2 NIZK.Verify (PKE dec) 1 PKE.Dec	$t+1$ NIZK.Verify (dec) 1 PKE.Dec 1 SS.Combine
	Comm.	N/A	N/A	N/A

Table 2. Computation (comp.) and communication (comm.) costs of generic constructions. We write (enc) and (dec) to indicate a proof of correct encryption or decryption respectively. For our multiple reporters construction, we provide costs relative to a ring of size $|R|$ and a threshold of t.

size of a trace is constant and requires only 1 extra field element and 2 extra group elements, when compared to accountable ring signatures.

Verifier. As for signature generation, signature verification costs are dominated by the SOK and PKE scheme. Specifically, our PKE computations require computation costs similar to those required to verify the SOK, which means verification of an R&T ring signature incurs double the computational costs of an accountable ring signature. On the other hand, verification of the trace requires only 6 group exponentiations, only 2 more than accountable ring signatures.

		Accountable ring signature [6]	**Our R&T instantiation** (§3.2)
Sign	Comp.	$4\|R\| + 14$	$8\|R\| + 19$
	Comm.	$14\mathbb{G} + (\|R\| + 7)\mathbb{Z}_q$	$(3\|R\| + 19)\mathbb{G} + (2\|R\| + 7)\mathbb{Z}_q$
Verify	Comp.	$3\|R\| + 19$	$6\|R\| + 23$
	Comm.	N/A	N/A
Report	Comp.	N/A	1
	Comm.	N/A	$1\mathbb{Z}_q$
Trace	Comp.	3	5
	Comm.	$3\mathbb{G} + 1\mathbb{Z}_q$	$5\mathbb{G} + 2\mathbb{Z}_q$
VerTrace	Comp.	4	6
	Comm.	N/A	N/A

Table 3. Computation (comp.) and communication (comm.) costs of instantiations. We present costs relative to a ring of size $|R|$. Our computation costs are given in terms of the number of group exponentiations required, and our communication costs are presented in terms of the number of group elements from $\mathbb{G}$ and field elements from $\mathbb{Z}_q$.

Potential Efficiency Improvements. Our instantiation builds upon the accountable ring signature of [6]. It is possible that techniques from group signature literature (e.g., [14, 15, 21]) could be used to provide a more efficient construction. However, we opt to build upon the construction of [6] to clearly demonstrate the additional costs associated with reporting when reporting is incorporated into an accountable ring signature. Indeed, our instantiation presents worst-case efficiency results and can perform reasonably for a small ring. Here, we highlight potential modifications to our instantiation that can lead to efficiency improvements.

Our instantiation could use a broadcast encryption (BE) scheme [9] to encrypt the reporter token to all members of the ring. Potentially, by using a BE scheme that is based on bilinear maps (as, for example, [5] and subsequent works), our costs would be similar to those of the accountable ring signature construction [6]. Our reasons for not following this approach are twofold. First, in contrast with BE schemes based on bilinear maps, our construction does not require an interactive key generation protocol. In fact, in our construction, users generate their public/secret key pair without the need for a trusted key distributor that holds a master secret key. Accordingly, our instantiation retains the benefit of ring signatures with respect to non-interactive key generation. Second, NIZK proofs of correct encryption for schemes based on bilinear maps are currently unknown [18]. Our construction, on the other hand, requires Σ-protocols to prove correct encryption and decryption of the reporter token. Moreover, by relying on cyclic groups, our instantiation does not require a proof of correct decryption for the reporter token, and the tracer can efficiently verify correct decryption.

Furthermore, we note two modifications that lead to a more efficient, yet more limited, protocol. Firstly, during setup, the signer could choose a *static* set of possible reporters that share a secret key for a PKE scheme. Then, more simply, the reporter share is encrypted under the corresponding public key.

Though this approach is more efficient, we opt to allow the signer to *dynamically* choose the reporter set, fostering a sense of user empowerment and capturing the functionality of a user posting entries in different fora. Secondly, if reporter anonymity is not a concern and a proof of correct decryption is required by the reporter, the reporter can produce a proof that indicates which reporter decrypted the token, which would decrease the computation costs incurred by the reporter. However, we opt to provide a construction that meets a strong security model, ensuring that reporters can produce reports without the concern of their identity being leaked.

4 Extending R&T to Multiple Reporters

In our construction, we assume that the reporter and tracer do not collude and, hence, if a reported message is not malicious, the tracer does not reveal the identity of the signer. That being said, we can further mitigate against a malicious reporter by requiring that the tracer receive *multiple* reports to trigger the tracing process. We describe an extension of our construction to multiple reporters that requires a (t, n)-publicly verifiable secret sharing scheme $\mathsf{PVSS}(\mathsf{SS.Gen}, \mathsf{SS.Verify}, \mathsf{SS.Combine})$, with syntax drawn from [22], where $n = |R|$ is the size of the ring and t is the number of shares required to reconstruct the secret.

The PVSS scheme is used to generate $|R|$ shares of the reporter token, i.e., reporter token $s = (s_1, \ldots, s_{|R|})$. Each reporter share is encrypted under a ring member's public key for a PKE scheme, rather than encrypting a single reporter token to all members of the ring. This extension requires minimal changes to our construction, which we outline here.

$\mathsf{Sign}(pp, sk_{\mathsf{U}}, pk_{\mathsf{T}}, m, R)$

parse pp as $(pp_{\mathsf{PKE}}, pp_{\mathsf{NIZK}}, pp_{\mathsf{SOK}}) \wedge sk_{\mathsf{U}}$ as $(sk_{\mathsf{RS}}, sk_{\mathsf{PKE}})$
$\wedge\ R$ as $\{(pk_{\mathsf{RS}_1}, pk_{\mathsf{PKE}_1}), \ldots, (pk_{\mathsf{RS}_{|R|}}, pk_{\mathsf{PKE}_{|R|}})\}$
$pk \leftarrow f(sk_{\mathsf{RS}});\ s \leftarrow SK;\ r_{1,1}, \ldots, r_{1,|R|}, r_2, r_3 \leftarrow \mathsf{Rand}$
$(\{s_1, \ldots, s_{|R|}\}, \{S_1, \ldots, S_{|R|}\}, S) \leftarrow \mathsf{SS.Gen}(s, t, |R|)$
for $i = 1, \ldots, |R|$
  $c_{1,i} \leftarrow \mathsf{PKE.Enc}(pp_{\mathsf{PKE}}, pk_{\mathsf{PKE}_i}, s_i; r_{1,i})$
  $\rho_i \leftarrow \mathsf{NIZK.Prove}(pp, (S_i, pk_{\mathsf{PKE}_i}, c_{1,i}), (r_{1,i}, s_i));\ \mathsf{Share}_i \leftarrow (S_i, c_{1,i}, \rho_{1,i})$
$c_2 \leftarrow \mathsf{PKE.Enc}(pp_{\mathsf{PKE}}, pk_{\mathsf{T}}, pk; r_2);\ c_3 \leftarrow \mathsf{PKE.Enc}(pp_{\mathsf{PKE}}, S, c_2; r_3)$
$\sigma_{\mathsf{SOK}} \leftarrow \mathsf{SoK.Sign}(pp, (pk_{\mathsf{T}}, S, R, c_3), (r_2, r_3, sk_{\mathsf{RS}}), m)$
return $\sigma = (S, (\mathsf{Share}_1, \ldots, \mathsf{Share}_{|R|}), c_3, \sigma_{\mathsf{SOK}})$

Fig. 6: Algorithm Sign for our construction with multiple reporters.

To sign a message, the signer uses the PVSS scheme to produce $|R|$ reporter sub-tokens. Each sub-token s_i is encrypted under the public key of a ring member

and accompanied with a NIZK proof of correct encryption and a public version of the sub-token, S_i. For clarity, we outline the changes to the signing algorithm in Figure 6. As before, our construction provides public verification algorithm Verify, which additionally requires that the verifier run algorithm SS.Verify to verify the secret sharing operations.

A ring member reports a message by decrypting their sub-token and sending the sub-token to the tracer, accompanied with a proof of correct decryption. Once the tracer has received a threshold number of sub-tokens, the tracer runs algorithm SS.Combine to recover the reporter token, then decrypts the signer's identity as per the original construction.

Anonymity of Reporters. Recall that a feature of our single reporter construction is that the reporter is anonymous, even after tracing. For our multiple reporter construction, this is no longer the case. Each reporter is provided with a unique sub-token, and public versions of each sub-token are published in order to verify correctness of the PVSS scheme. In this way, when a reporter produces a report, i.e., their sub-token, anyone can check which share (denoted Share) this belongs to. Thus, it is possible to determine the identity of the reporters in our multiple reporter construction. For this reason, reporter sub-tokens should be treated carefully by the tracer *before* tracing. That is, the sub-tokens should not be revealed by the tracer until tracing is completed. In this way, the identities of reporters are known to the tracer before tracing but are not made public until after tracing.

Efficiency of Multiple Reporters. Our construction with multiple reporters is less efficient than our generic construction with a single reporter, where the communication and computation costs are outlined in Table 2. Specifically, producing a signature requires the additional computation and communication of $|R|$ reporter sub-tokens, and the computation and communication costs of the tracer grow linearly in the size of the threshold. Moreover, the computation costs associated with signature and trace verification grow linearly in the size of the ring and threshold, respectively. However, these costs can be minimised. In particular, secret share generation and combination, for efficient PVSS schemes (e.g., [22]), simply requires the computation of group exponentiations and the addition of field elements respectively [22]. Additionally, the NIZK proofs associated with these extra costs can be instantiated with efficient primitives, as in our single reporter construction. That being said, we note that the size and computation costs of a report are identical to our single reporter construction, consisting of a single reporter sub-token, and requiring a single decryption of a PKE ciphertext.

5 Conclusion

We introduced and defined report and trace ring signatures, and presented an accompanying security model. We showed that our new primitive not only protects

the identity of the signer until tracing is complete, it also protects the identity of the reporter even after tracing. We presented a construction of an R&T ring signature scheme that satisfies our security model, and extended our construction to the multiple reporter setting. Additionally, we provided an instantiation of our single reporter construction and compared its efficiency with accountable ring signatures [6], demonstrating the additional costs associated with our report and trace functionality, and showing that our proposed instantiation is practical.

Though our construction can be efficiently instantiated, the costs incurred by the signer and the verifier grow linearly in the size of the ring. An interesting area of future research is to define an efficient, yet (efficiently) verifiable, broadcast encryption scheme that can be used to instantiate our construction. We show that our report and trace construction can be extended to the multiple reporter setting. This setting could be formalised by extending our existing security model to capture multiple reporters, and, additionally, a construction that satisfies reporter anonymity could be sought.

We believe that the cryptographic solution that report and trace provides can be enhanced through the use of policy and enforcement. In fact, a policy can define malicious messages, and the possibility of enforcement encourages reporters and tracers to follow policy. Therefore, we believe that our report and trace notion forms part of a solution that, when supplemented with policy and enforcement, leads to a well-functioning system that protects the anonymity of its participants, with limited room for abuse.

References

1. Venkat Arun, Aniket Kate, Deepak Garg, Peter Druschel, and Bobby Bhattacharjee. Finding safety in numbers with secure allegation escrows. In *NDSS'20*. Internet Society, 2020.
2. Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT'03*, pages 614–629. Springer, Berlin, Heidelberg, 2003.
3. Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA'05*, pages 136–153. Springer, Berlin, Heidelberg, 2005.
4. Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In *Theory of Cryptography*, pages 60–79. Springer, Berlin, Heidelberg, 2006.
5. Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO'05*, pages 258–275. Springer, Berlin, Heidelberg, 2005.
6. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on ddh. In *ESORICS'15*, pages 243–265. Springer, Berlin, Heidelberg, 2015.
7. Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups. In *CRYPTO '97*, pages 410–424. Springer, Berlin, Heidelberg, 1997.
8. David Chaum and Eugène Van Heyst. Group signatures. In *EUROCRYPT'91*, pages 257–265. Springer, Berlin, Heidelberg, 1991.

9. Amos Fiat and Moni Naor. Broadcast encryption. In *CRYPTO'93*, pages 480–491. Springer, Berlin, Heidelberg, 1994.
10. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO' 86*, pages 186–194. Springer, Berlin, Heidelberg, 1986.
11. Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In *PKC'07*, pages 181–200. Springer, Berlin, Heidelberg, 2007.
12. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
13. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for np. In *EUROCRYPT'06*, pages 339–358. Springer, Berlin, Heidelberg, 2006.
14. Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Traceable signatures. In *EUROCRYPT'04*, pages 571–589. Springer, Berlin, Heidelberg, 2004.
15. Markulf Kohlweiss and Ian Miers. Accountable metadata-hiding escrow: A group signature case study. *PoPETs'15*, 2015(2):206–221, 2015.
16. Benjamin Kuykendall, Hugo Krawczyk, and Tal Rabin. Cryptography for #metoo. *PoPETs'19*, 2019(3):409–429, 2019.
17. Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In *Information Security and Privacy*, pages 325–335. Springer, Berlin, Heidelberg, 2004.
18. Duong-Hieu Phan, David Pointcheval, Siamak F Shahandashti, and Mario Strefler. Adaptive cca broadcast encryption with constant-size secret keys and ciphertexts. *International Journal of Information Security*, 12(4):251–265, 2013.
19. 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 *COMPASS'18*, pages 1–4, 2018.
20. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *ASIACRYPT'01*, pages 552–565. Springer, Berlin, Heidelberg, 2001.
21. Yusuke Sakai, Keita Emura, Goichiro Hanaoka, Yutaka Kawai, Takahiro Matsuda, and Kazumasa Omote. Group signatures with message-dependent opening. In *Pairing-Based Cryptography'12*, pages 270–294. Springer, Berlin, Heidelberg, 2013.
22. Markus Stadler. Publicly verifiable secret sharing. In *EUROCRYPT '96*, pages 190–199. Springer, Berlin, Heidelberg, 1996.
23. Nirvan Tyagi, Ian Miers, and Thomas Ristenpart. Traceback for end-to-end encrypted messaging. In *CCS'19*, pages 413–430. Association for Computing Machinery, 2019.
24. Lingling Wang, Guoyin Zhang, and Chunguang Ma. A survey of ring signature. *Frontiers of Electrical and Electronic Engineering in China*, 3(1):10–19, 2008.
25. Shouhuai Xu and Moti Yung. Accountable ring signatures: A smart card approach. In *Smart Card Research and Advanced Applications VI*, pages 271–286. Springer, Berlin, Heidelberg, 2004.

A Security of Our Construction

Here, we prove Theorem 1 via a series of Lemmata.

Lemma 1. *Our construction satisfies* correctness *if public-key encryption scheme* PKE *is correct, non-interactive zero-knowledge proof system* NIZK *is complete, and signature of knowledge* SOK *is correct.*

Proof. Consider a signature $\sigma = (pk_{\mathsf{Sign}}, \boldsymbol{c_1}, c_3, \rho, \sigma_{\mathsf{SOK}})$ output by algorithm Sign with respect to a message $m \in \mathsf{M}$ and ring R that consists of public keys generated by algorithm U.KGen. Let pp be the output of algorithm Setup and let $(pk_{\mathsf{T}}, sk_{\mathsf{T}})$ be the output of algorithm T.KGen. Moreover, let $pk = f(sk_{\mathsf{RS}}) \in R$ where sk_{RS} is an element of SK and is used to sign message m.

By definition of correctness, our construction is correct if $\mathsf{Verify}(pp, pk_{\mathsf{T}}, m, R, \sigma)$ outputs 1 with overwhelming probability. Assume that algorithm Verify does not return 1. Then, it must be the case that $\mathsf{NIZK.Verify}(pp, (pk_{\mathsf{Sign}}, (pk_{\mathsf{PKE}_1}, \ldots, pk_{\mathsf{PKE}_{|R|}}), \boldsymbol{c_1}), \rho) = 0$ or $\mathsf{SoK.Verify}(pp, (pk_{\mathsf{T}}, pk_{\mathsf{Sign}}, R, c_3), m, \sigma_{\mathsf{SOK}}) = 0$. We consider these two possibilities in turn. We conclude that, if all building blocks are correct, our construction satisfies correctness.

First, assume that $\mathsf{NIZK.Verify}(pp, (pk_{\mathsf{Sign}}, (pk_{\mathsf{PKE}_1}, \ldots, pk_{\mathsf{PKE}_{|R|}}), \boldsymbol{c_1}), \rho) = 0$. By assumption, the PKE scheme satisfies correctness and so $\boldsymbol{c_1}$ encrypts sk_{Sign}, which is cryptographically linked to pk_{Sign}. Then, the inputs to algorithm NIZK.Verify are correctly generated and, by completeness of the NIZK scheme, algorithm NIZK.Verify returns 1 with overwhelming probability.

Now, assume that $\mathsf{SoK.Verify}(pp, (pk_{\mathsf{T}}, pk_{\mathsf{Sign}}, R, c_3), m, \sigma_{\mathsf{SOK}}) = 0$. As before, by assumption of correctness of the PKE scheme, ciphertexts c_2 and c_3 encrypt pk and c_2 respectively. Then, the inputs to algorithm SoK.Sign are generated correctly and, by correctness of the SOK scheme, algorithm SoK.Verify returns 1 with overwhelming probability.

Lemma 2. *Our construction satisfies* anonymity *if public-key encryption scheme* PKE *satisfies* IND-CPA*, non-interactive zero-knowledge proof system* NIZK *satisfies zero-knowledge and knowledge soundness, and signature of knowledge* SOK *satisfies simulatability and extractability.*

Proof. Let $\mathcal{A}$ be an adversary in the anonymity experiment. We proceed through a series of game hops that we show are indistinguishable to the adversary. We define Game 0 as the anonymity experiment with b chosen randomly and let S_i denote the event that $\mathcal{A}$ correctly guesses b after Game i.

Game 1 is identical to Game 0 except that we replace algorithm SoK.Setup with algorithm SimSOK.Setup and, for queries to $\mathcal{O}$sign and the signature output to $\mathcal{A}$, we replace algorithm SoK.Sign with algorithm SimSOK.Sign. Then, generating the signature of knowledge does not require a witness. By the simulatability of the SOK scheme, $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{negl}(\lambda)$.

Game 2 is identical to Game 1 except that we replace algorithm NIZK.Setup with algorithm SimNIZK.Setup and, for queries to $\mathcal{O}$report and $\mathcal{O}$trace, we replace algorithm NIZK.Prove with algorithm SimNIZK.Prove. As such, generating a proof

of correct decryption does not require a witness. By the zero-knowledge property of the NIZK scheme, $|\Pr[S_1] - \Pr[S_2]| \leq \mathsf{negl}(\lambda)$.

Game 3 requires a further change to $\mathcal{O}\mathsf{report}$. Rather than decrypting a ciphertext from $\boldsymbol{c_1}$, we run algorithm NIZK.Extract to output the witness $(r_{1,1}, \ldots, r_{1,|R|}, sk_{\mathsf{Sign}})$ from proof of correct decryption ρ. By the knowledge soundness property of the NIZK scheme, $|\Pr[S_2] - \Pr[S_3]| \leq \mathsf{negl}(\lambda)$.

Game 4 changes $\mathcal{O}\mathsf{trace}$ to run algorithm SoK.Extract to obtain the identity of the signer rather than decrypting ciphertexts c_3 and c_2. By the simulation extractability property of the SOK scheme, $|\Pr[S_3] - \Pr[S_4]| \leq \mathsf{negl}(\lambda)$.

Game 5 requires a change to the challenge signature returned to $\mathcal{A}$ when $b = 1$. We replace pk_1 with pk_0. By the IND-CPA property of the PKE scheme, $|\Pr[S_4] - \Pr[S_5]| \leq \mathsf{negl}(\lambda)$.

The view of $\mathcal{A}$ is now identical for $b = 0$ and $b = 1$. Therefore, $\Pr[S_5] = 1/2$ and the result holds.

Lemma 3. *Our construction satisfies* unforgeability *if f is a one-way function and signature of knowledge* SOK *satisfies extractability.*

Proof. Let $\mathcal{A}$ be an adversary in the unforgeability experiment and assume that the construction does not satisfy unforgeability. Then, $\mathcal{A}$ can output a tuple $(pk_{\mathsf{T}}, m, R, \sigma)$ such that $\mathsf{Verify}(pp, pk_{\mathsf{T}}, m, R, \sigma) = 1$ where the ring is honest and the tuple output is not queried to $\mathcal{O}\mathsf{sign}$. On behalf of an honest ring member, $\mathcal{A}$ can perform all steps of Sign *except* produce signature of knowledge σ_{SOK}. For this, $\mathcal{A}$ must obtain the secret credential of the signer. If $\mathcal{A}$ can obtain the secret credential without corrupting the signer, $\mathcal{A}$ can break the one-wayness of function f. Else, $\mathcal{A}$ can output a valid SOK without a valid witness sk_{RS}. If $\mathcal{A}$ can output a valid SOK without a witness sk_{RS} then $\mathcal{A}$ can be used to construct an adversary against the extractability property of the SOK. Therefore, by contradiction, $\mathcal{A}$ can succeed in the unforgeability experiment with negligible probability.

Lemma 4. *Our construction satisfies* traceability*. That is:*

1. *Our construction satisfies* tracing correctness *if public-key encryption scheme* PKE *is correct, non-interactive zero-knowledge proof system* NIZK *is complete, and signature of knowledge* SOK *is correct.*
2. *Our construction satisfies* non-frameability *if f is a one-way function, public-key encryption scheme* PKE *is correct, non-interactive zero-knowledge proof system* NIZK *satisfies knowledge soundness, and signature of knowledge* SOK *satisfies extractability.*
3. *Our construction satisfies* soundness *if public-key encryption scheme* PKE *is correct and non-interactive zero-knowledge proof system* NIZK *satisfies knowledge soundness.*

Proof. We show that all three properties of traceability are satisfied.

Tracing correctness: Consider a signature $\sigma = (pk_{\mathsf{Sign}}, \boldsymbol{c_1}, c_3, \rho, \sigma_{\mathsf{SOK}})$ output by algorithm Sign with respect to a message $m \in \mathsf{M}$ and ring R that consists of public keys generated by algorithm U.KGen. Let pp be the output of algorithm

Setup and let $(pk_{\mathsf{T}}, sk_{\mathsf{T}})$ be the output of algorithm $\mathsf{T.KGen}$. Let $pk = f(sk_{\mathsf{RS}}) \in R$ where sk_{RS} is an element of SK and is used to sign message m. Moreover, let $\mathtt{Rep} = (sk_{\mathsf{Sign}}, \rho_r)$ and $(pk, \mathtt{Tr} = (c_2, \mathtt{Rep}), \rho_t)$ be the outputs of algorithms Report and Trace respectively.

By definition of tracing correctness, our construction is correct if $\mathsf{VerTrace}(pp, pk_{\mathsf{T}}, m, R, \sigma, pk, \mathtt{Tr}, \rho_t)$ outputs 1 with overwhelming probability. Assume that algorithm $\mathsf{VerTrace}$ does not return 1. Then, it must be the case that $\mathsf{NIZK.Verify}(pp, (pk_{\mathsf{T}}, c_2, pk), \rho_t) = 0$, $\mathsf{NIZK.Verify}(pp, (R, \boldsymbol{c_1}, sk_{\mathsf{Sign}}), \rho_r) = 0$, $\mathsf{Verify}(pp, pk_{\mathsf{T}}, m, R, \sigma) = 0$ or $\mathsf{PKE.Dec}(pp_{\mathsf{PKE}}, sk_{\mathsf{Sign}}, c_3) \neq c_2$. We consider these possibilities in turn. We conclude that, if all building blocks are correct, our construction satisfies tracing correctness.

First, recall that, by correctness of the construction (Lemma 1), algorithm Verify returns 1 with overwhelming probability unless one of the building blocks do not satisfy correctness. Second, as in our proof of correctness, algorithm $\mathsf{NIZK.Verify}$ returns 1 with overwhelming probability if the NIZK scheme satisfies completeness and the PKE scheme is correct. Finally, by correctness of the PKE scheme, $c_2 = \mathsf{PKE.Enc}(pp_{\mathsf{PKE}}, pk_{\mathsf{T}}, c_3)$ with overwhelming probability, and the result holds.

Non-frameability: Let $\mathcal{A}$ be an adversary in the non-frameability experiment and assume that the construction does not satisfy non-frameability. Then $\mathcal{A}$ can output a tuple $(pk_{\mathsf{T}}, m, R, \sigma, pk, \mathtt{Tr} = (c_2, \mathtt{Rep}), \rho_t)$ such that $\mathsf{VerTrace}(pp, pk_{\mathsf{T}}, m, R, \sigma, pk, \mathtt{Tr}, \rho_t) = 1$ where pk is honest and the tuple output by $\mathcal{A}$ is not queried to $\mathcal{O}\mathsf{sign}$.

First, we assume that $\mathcal{A}$ produces a signature $\sigma = (pk_{\mathsf{Sign}}, \boldsymbol{c_1}, c_3, \rho, \sigma_{\mathsf{SOK}})$ for a corrupt ring member pk for message m and ring R such that $\mathsf{Verify}(pp, pk_{\mathsf{T}}, m, R, \sigma) = 1$. Then $\mathcal{A}$ must construct a report and trace that identifies a different, honest, ring member pk' as the signer. $\mathcal{A}$ must construct proofs ρ_r and ρ_t such that algorithm $\mathsf{NIZK.Verify}$ returns 1. However, if the public-key encryption scheme is correct, by knowledge soundness of the NIZK scheme, an adversary can create valid proofs without a valid witness with negligible probability. Therefore, if $\mathcal{A}$ can output a valid report and trace, $\mathcal{A}$ can be used to construct an adversary against the knowledge soundness property of the NIZK scheme.

Otherwise, $\mathcal{A}$ must output a valid signature on behalf of an honest ring member. However, by unforgeability of our construction (Theorem 3), e.g., one-wayness of function f and extractability of the SOK, $\mathcal{A}$ can do this with negligible probability. Then, by contraction, $\mathcal{A}$ succeeds in the non-frameability experiment with negligible probability.

Soundness: Let $\mathcal{A}$ be an adversary in the soundness experiment and assume that the construction does not satisfy soundness. Then, $\mathcal{A}$ can output a tuple $(pk_{\mathsf{T}}, m, R, \sigma, pk_i, \mathtt{Tr}_i, \rho_{t_i}, pk_j, \mathtt{Tr}_j, \rho_{t_j})$ such that algorithm $\mathsf{VerTrace}(pp, pk_{\mathsf{T}}, m, R, \sigma, pk_k, \mathtt{Tr}_k, \rho_{t,k}) = 1$ for $k \in \{0, 1\}$ and $pk_i \neq pk_j$. Without loss of generality, we assume that tuple $(pk_i, \mathtt{Tr}_i, \rho_{t_i})$ identifies the signer (which $\mathcal{A}$ can generate as all potential signers are corruptible).

$\mathcal{A}$ must output proofs ρ_{r_j} and ρ_{t_j} such that algorithm $\mathsf{NIZK.Verify}$ returns 1. However, if the public-key encryption scheme is correct, by knowledge soundness

of NIZK, an adversary can create valid proofs without a valid witness with negligible probability. Therefore, if $\mathcal{A}$ can output two valid proofs, $\mathcal{A}$ can be used to construct an adversary against NIZK knowledge soundness. Therefore, by contradiction, $\mathcal{A}$ succeeds in the soundness experiment with negligible probability.

Lemma 5. *Our construction satisfies* reporter anonymity *if non-interactive zero-knowledge proof system* NIZK *satisfies zero-knowledge and knowledge soundness.*

Proof. Let $\mathcal{A}$ be an adversary in the reporter anonymity experiment. We proceed through a series of game hops that we show are indistinguishable to the adversary. In the final game, the view of $\mathcal{A}$ is identical for $b = 0$ and $b = 1$. We define Game 0 as the reporter anonymity experiment with b chosen randomly and let S_i denote the event that $\mathcal{A}$ correctly guesses b after Game i.

Game 1 is identical to Game 0 except that, we replace algorithm NIZK.Setup with algorithm SimNIZK.Setup and, for queries to $\mathcal{O}$report and the report output to $\mathcal{A}$, we replace algorithm NIZK.Prove with algorithm SimNIZK.Prove. In this way, generating a reporter proof of correct decryption does not require a witness sk_{PKE}. By the zero-knowledge property of the NIZK scheme, $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{negl}(\lambda)$.

Game 2 is identical to Game 1 except that, for queries to $\mathcal{O}$report and the report output to $\mathcal{A}$, we run algorithm NIZK.Extract to output the witness $(r_{1,1}, \ldots, r_{1,|R|}, sk_{\mathsf{Sign}})$ from proof of correct encryption ρ, rather than decrypting a ciphertext from $\boldsymbol{c_1}$. By the knowledge soundness property of the NIZK scheme, $|\Pr[S_1] - \Pr[S_2]| \leq \mathsf{negl}(\lambda)$.

Now, the view of $\mathcal{A}$ is identical for $b = 0$ and $b = 1$. In fact, the reporter token is generated without the secret key of the reporter. Therefore, $\Pr[S_2] = 1/2$ and the result holds.

B Security of Our Instantiation

Here we provide sketch proofs of the security of our instantiation, demonstrating that our choices of cryptographic primitives meet the requirements for security, that is, the proofs in Appendix A hold. We restrict ourselves to sketch proofs noting that details of the proofs are very similar to the proofs presented in [6], and we refer the reader to [6] for more details.

Lemma 6. *Function f, where $f(x) = g^x$ for some $x \in \mathbb{Z}_q$ and group $(\mathbb{G}, g, q)$, is a one-way function if the discrete logarithm assumption holds with respect to group $(\mathbb{G}, g, q)$.*

Proof (Sketch). Recall that the discrete log assumption holds relative to group $(\mathbb{G}, g, q)$ if an adversary, on input of $(\mathbb{G}, g, q)$ and a group element y, cannot output a discrete log x such that $y = g^x$, except with negligible probability. Let $\mathcal{A}$ be an adversary in the hard to invert property of one-way function f that outputs an element x' such that $f(x') = y$. Then we can use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ against the discrete logarithm assumption. Trivially, $\mathcal{B}$ can output x' and the result holds.

Lemma 7. *The double-layer of encryption that defines ciphertext c_3 is correct and satisfies* IND-CPA *security.*

Proof (Sketch). Correctness holds by direct verification. Moreover, if there exists an adversary $\mathcal{A}$ against the IND-CPA security of ciphertext c_3, then we can use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ against the IND-CPA property of the underlying standard ElGamal encryption scheme. By assumption, ElGamal satisfies IND-CPA security so, by contradiction, ciphertext c_3 is IND-CPA secure.

Lemma 8. *The Σ-protocol in Figure 4 satisfies completeness, special honest verifier zero-knowledge, 2-special soundness and has quasi-unique responses. Moreover, applying the Fiat-Shamir transform to the Σ-protocol leads to an* SOK *in the ROM that is correct, simulatable and extractable.*

Proof (Sketch). This result follows directly from the security of the Σ-protocol of [6], including the security of the two additional Σ-protocols upon which it is based. We refer the reader to [6] for full details of the proofs. Briefly, completeness follows from direct verification, and correctness of the SOK follows from completeness of the Σ-protocol. For special honest verifier zero-knowledge, we define algorithm Sim to choose responses $z_{w1}, z_E, z_{D1}, z_{D2} \leftarrow_\$ \mathbb{Z}_q$ and choose $C \leftarrow_\$ \mathbb{G}^2$. Ciphertexts D_2 and E can be obtained from the verification equations. Then, by the DDH assumption, C output by Sim is indistinguishable from C when generated according to the Σ protocol if the DDH assumption holds with respect to group $(\mathbb{G}, g, q)$. Moreover, $z_{w1}, z_E, z_{D1}, z_{D2}$ are uniform and uniquely define D_2 and E when generated by $\mathcal{P}$ or Sim and the result holds. Simulatability of the SOK scheme follows. For 2-special soundness, we define two challenges x, x' and responses $z_{w1}, z_E, z_{D1}, z_{D2}$ and $z'_{w1}, z'_E, z'_{D1}, z'_{D2}$, such that the verifier returns 1 for each challenge/response pair. Then by computing r_2, r_3 and sk_{RS} from the first verification equation $c_3^{(x-x')} = \mathsf{PKE.Enc}(pp_{\mathsf{PKE}}, (pk_{\mathsf{T}}, pk_{\mathsf{Sign}}), g^{z_{w1}-z'_{w1}}; z_{D1} - z'_{D1}, z_{D2} - z'_{D2})$ and replacing in the second verification equation, it is clear that c_3 encrypts pk as required. Extractability of the SOK scheme follows.

Lemma 9. *The Σ-protocol in Figure 3 satisfies completeness, special honest verifier zero-knowledge and 2-special soundness. Moreover, applying the Fiat-Shamir transform to the Σ-protocol leads to an* NIZK *proof in the ROM that satisfies completeness, zero-knowledge, soundness and knowledge extractability.*

Proof (Sketch). This result is similar to the proof of Lemma 8. In fact, completeness follows from direct verification and completeness of the NIZK proof follows. Special honest verifier zero knowledge holds because Sim can choose responses $z_1, \ldots, z_n \leftarrow_\$ \mathbb{Z}_q$ and selects challenges t_h and $t_{g_1}, \ldots, t_{g_n}$ based on the random responses and x. As such, the result holds and simulatability of the NIZK proof follows. Finally, special soundness holds as two challenges x, x' and responses $z_1, \ldots, z_n$ and $z'_1, \ldots, z'_n$ ensure that c_1 encrypts a discrete log as required. As before, soundness and knowledge extractability of the NIZK proof follows.

Lemma 10. *The Σ-protocol in Figure 5 satisfies completeness, special honest verifier zero-knowledge, 2-special soundness and has unique responses. Moreover,*

applying the Fiat-Shamir transform to the Σ-protocol leads to an NIZK *proof in the ROM that satisfies completeness, zero-knowledge, soundness and knowledge extractability.*

Proof (Sketch). The proof of this result follows is identical to the proof presented in [6], which presents a Σ-protocol for the same relation. Moreover, completeness of the NIZK proof follows from completeness of the Σ-protocol, zero-knowledge follows from special honest verifier zero-knowledge and soundness and knowledge extractability follows from special soundness.