A generic view on trace-and-revoke broadcast encryption schemes

Dennis Hofheinz* and Christoph Striecks**

Karlsruhe Institute of Technology, Karlsruhe, Germany

Abstract. At Eurocrypt 2011, Wee presented a generalization of threshold public key encryption, threshold signatures, and revocation schemes arising from threshold extractable hash proof systems. In particular, he gave instances of his generic revocation scheme from the DDH assumption (which led to the Naor-Pinkas revocation scheme), and from the factoring assumption (which led to a new revocation scheme). We expand on Wee's work in two directions:

(a) We propose threshold extractable hash proof instantiations from the "Extended Decisional Diffie-Hellman" (EDDH) assumption due to Hemenway and Ostrovsky (PKC 2012). This in particular yields EDDH-based variants of threshold public key encryption, threshold signatures, and revocation schemes. In detail, this yields a DCR-based revocation scheme.

(b) We show that our EDDH-based revocation scheme allows for a mild form of traitor tracing (and, thus, yields a new trace-and-revoke scheme). In particular, compared to Wee's factoring-based scheme, our DCR-based scheme has the advantage that it allows to trace traitors.

Keywords: broadcast encryption, revocation scheme, traitor tracing, trace-and-revoke scheme, threshold extractable hash proof system, extended decisional Diffie-Hellman.

1 Introduction

Broadcast encryption, revocation schemes, traitor tracing, and trace-and-revoke schemes. In a broadcast encryption (BE) scheme [18], a sender is able to generate ciphertexts that only members of a privileged set $\mathcal{S} \subseteq \{1, \ldots, N\}$ of users — each given a long-lived user secret key — can decrypt. There exists a large number of BE schemes under various assumptions and with various efficiency characteristics (e.g., [18, 21, 9, 3, 22, 35, 43, 42]). In this work, we focus on revocation schemes, which are a variant of BE schemes, where a set of revoked users (e.g., non-paying subscribers) $\mathcal{R} = \{1, \ldots, N\} \setminus \mathcal{S}$ is given as input to the encryption function. Revocation schemes proposed in the literature are, e.g., [39, 52, 37, 24, 14, 15, 56, 23, 13, 34, 55]. A particularly interesting property a cryptosystem in the broadcast encryption setting can have is traceability [12], i.e., the ability to trace a "pirate" decryption box back to the corrupted user(s), called traitor(s), who constructed it. Thus, traceability allows to identify a traitor (or a coalition of traitors). Such schemes are called traitor tracing schemes and a variety of them was proposed, e.g., [12, 40, 41, 33, 49, 50, 38, 6, 19, 44, 53, 29, 47, 31, 30, 36, 11, 51, 10, 17, 48, 1, 4, 7]. The combination of revocation and traceability is an aspiring goal. We stress that combining these properties is nontrivial (see [8, Section 4.1]). Nevertheless, there are schemes, e.g., [20, 39, 37, 52, 24, 14, 15, 32, 16, 8, 27][1], which provide a solution for this problem. These schemes are called trace-and-revoke schemes.

Threshold extractable hash proof systems. In [55], Wee established threshold extractable hash proof systems (TEHPS) as a generalization of extractable hash proof systems (EHPS) [54]. Applying the concept of TEHPSs, Wee explains threshold public key encryption, threshold signatures, and revocation schemes from the Decisional Diffie-Hellman (DDH), from the Computational Diffie-Hellman (CDH), and from the factoring assumptions which — at least in the case of factoring — led to new cryptosystems. We expand the generic view of [55] by providing a TEHPS from the "Extended Decisional Diffie-Hellman" (EDDH) assumption due to Hemenway and Ostrovsky [25]. The EDDH assumption generalizes the DDH and Decisional Composite Residuosity (DCR) assumptions. By our first result, we obtain threshold public key encryption, threshold signatures, and revocation schemes from the EDDH assumption. In particular, our generic system extends

* Dennis Hofheinz was supported by a DFG grant (GZ HO 4534/2-1).

** Christoph Striecks was supported by a DFG grant (GZ HO 4534/2-1).

[1] Note that the schemes from [37, 24, 14] support a different form of traitor tracing. Particularly, their main goal is to find a setting in which the pirate box is not useful anymore rather than identifying the traitor(s).

the generic view of revocation schemes from [55] (recapped below) and, additionally, via our second result, it yields a new trace-and-revoke scheme from the DCR assumption. (This is not known for the factoring-based instance of [55] and we describe why this seems to be difficult to achieve in Wee's setting.)

A generic revocation scheme. Recently, Wee [55] gave a very simple and elegant generic view of revocation schemes. He explains and generalizes previous constructions (e.g., [39, 52]). The public key in these constructions contains the coefficients of a secret polynomial $f(x) = a_0 + a_1 x + \cdots + a_t x^t$ "in the exponent" as

$$g^{a_0}, g^{a_1}, \ldots, g^{a_t}.$$

Note that this allows to compute values $g^{f(x)}$ for arbitrary x. A ciphertext is of the form

$$C = (\mathcal{R}, u, (u^{f(id)})_{id \in \mathcal{R}}),$$

where $\mathcal{R}$ is a set of t revoked identities. (The $u^{f(id)}$ can be computed from pk, and using knowledge of an exponent r with $u = g^r$.) The corresponding encapsulated key[2] is $s = u^{f(0)}$. Any user with identity id in the system possesses a user secret key $usk_{id} = f(id)$. (Of course, 0 is not an allowed identity for a user.) If $id \notin \mathcal{R}$, then a user can derive a $(t+1)$-st share $u^{usk_{id}} = u^{f(id)}$ and compute $u^{f(0)}$ through Lagrange interpolation of the $t+1$ values $u^{f(id)}$ (for $id \in \mathcal{R} \cup \{id\}$). Depending on the domain over which we are working, and on how a "raw key" $s = u^{f(0)}$ is post-processed, this yields a revocation scheme from the DDH, the CDH, or the factoring assumption. Note that although similar secret sharing techniques are common in broadcast encryption, Wee's scheme is particularly simple and appealing from a conceptual point of view.

Our first result: an EDDH-based TEHPS instance. By giving a slightly different generic view, we extend the work of Wee to obtain threshold extractable hash proof instantiations from the extended decisional Diffie-Hellman assumption. Concretely, the EDDH assumption works in a group $\mathbb{G}$ with subgroups G, H. It states that, given g, g^x, and g^y, elements g^{xy} are computationally indistinguishable from elements $g^{xy} \cdot h$, where $g \in G$ and $h \in H$ are uniformly chosen, and x, y are uniform exponents. For $G = H$, we have the DDH assumption, and if $\mathbb{G} = \mathbb{Z}_N^*$, $G = \{x^N \mid x \in \mathbb{G}\}$, and $H = \langle 1 + N \rangle$, we have the DCR assumption. In particular, our first result yields EDDH-based threshold encryption, signatures, and revocation schemes. We stress that the EDDH-based instances use a potential stronger assumption (i.e., DCR) as opposed to Wee's factoring-based schemes. Nevertheless, to give a foreshadow, this slightly stronger assumption enables us — via our second result — to obtain a new DCR-based trace-and-revoke scheme which, again, is not known to achieve from Wee's factoring-based scheme. Our revocation scheme is similar to the above generic scheme, but has ciphertexts

$$C = (\mathcal{R}, u_1, (u_1^{f(id)})_{id \in \mathcal{R}}, u_2), \tag{1}$$

for $u_1 \in G$ and $u_2 = u_1^{f(0)} \cdot h$ with $h \in H$. The shared key is extracted from h. Hence, instead of directly using $u_1^{f(0)}$ as shared key, we use it to blind the actual key h. This is consistent with the EDDH assumption: EDDH does not state that g^{xy} looks random — it *does* state however that g^{xy} can be used to blind an H-element. The security analysis of this modified scheme is similar to the analysis of previous schemes. The only difficulties arise out of the fact that the group order of G may not be known (e.g., in the case of DCR). Hence, we must avoid inversion operations in the exponent. (Such inversion operations arise during Lagrange interpolation of the polynomial f in the exponent.) More details about the technique we use to avoid inversions in the exponent are given below.

Our second result: traceability of the EDDH-based revocation scheme. We prove that our EDDH-based revocation scheme also supports a mild form of black-box traitor tracing. That is, we prove that any pirate box produced by a coalition of $T \leq (t+1)/2$ corrupted users can be traced back to a user in that coalition. Tracing requires only completely black-box access to the pirate box and works for imperfect decryption boxes (where the box is allowed to decrypt well-formed ciphertexts invalidly down to some threshold). Further, we allow adversarially chosen revoked sets $\mathcal{R}$. Similar black-box tracing strategies in the revocation setting were considered in previous works, e.g., in [52, 16]. But unlike in, e.g., [52], our tracing algorithm works with imperfect pirate boxes that may even only work for an adversarially chosen set $\mathcal{R}$ of revoked users. The tracing model in [16] also considers imperfect decryption boxes and adversarially chosen

[2] Wee's scheme actually is a key encapsulation mechanism, not a full encryption scheme. Hence, a ciphertext does not encrypt a message, but only encapsulates a key that can be used to (symmetrically) encrypt a message.

revoked users, but for a different scheme. (To achieve black-box traceability in the BE setting we note that similar techniques are common, e.g., in [8].) However, we stress that our focus is on the generic view of constructing trace-and-revoke schemes. Nevertheless, our tracing strategy is explained in more detail below.

More on the used techniques. To construct revocation schemes from the EDDH assumption — in which the order of the subgroup G might not be known as opposed to Wee's generic construction above — we use a technique called "clearing the denominator" in the exponent. This tool was used before, but in different scenarios to ours, e.g., in [46, 55, 2]. Hence, we can avoid Lagrangian coefficient inversion in the exponent and are able to construct our EDDH-based revocation scheme. We focus on this construction in Section 3. For traceability, consider *random* ciphertexts of the form

$$C^{\mathcal{R}}_{\mathsf{rnd}} = (\mathcal{R}, u_1, (u_1^{f(id)} h^{z_{id}})_{id \in \mathcal{R}}, u_1^{f(0)} h^{z_0}) \quad \text{for uniform } h \in H \text{ and } z_{id}, z_0.$$

Under the EDDH assumption, such random ciphertexts are indistinguishable from real ones, *even when knowing a single user key* usk_{id}. In particular, a pirate box $\mathcal{B}$ decrypts random ciphertexts just as well as real ones. However, the decryption of random ciphertexts depends highly on which user key usk_{id} is used to decrypt. Hence, to trace a pirate box $\mathcal{B}$ back to its creator, we can simply feed $\mathcal{B}$ with random ciphertexts and compare $\mathcal{B}$'s output with decryption results for various user keys. This strategy only works if the pirate box $\mathcal{B}$ knows only one user key. If $\mathcal{B}$ knows, say, two different user keys, it can distinguish real from random ciphertexts. (For instance, $\mathcal{B}$ could decrypt a given ciphertext under the two keys. If the decryptions do not match, the ciphertext cannot be real. See [28] by Kiayias and Yung for a more general case and a formal analysis.) Thus, we adapt our strategy by considering "semi-random ciphertexts" of the form

$$C^{\mathcal{R},I}_{\mathsf{rnd}} = (\mathcal{R}, u_1, (u_1^{f(id)} h^{f'(id)})_{id \in \mathcal{R}}, u_1^{f(0)} h^{f'(0)}) \quad \begin{array}{l} \text{for } f'(x) \in \mathbb{Z}_q[x] \text{ uniform} \\ \text{of degree} \leq t, \text{ but subject} \\ \text{to } f'(id) = 0 \text{ for } id \in I. \end{array} \tag{2}$$

Such ciphertexts are indistinguishable from real ones, even when knowing the user keys for I. However, when using user keys for identities outside of I, then we will get a different, random result. Our tracing strategy will hence make a guess for the set I of corrupted users, and confirm the guess by checking if $\mathcal{B}$ decrypts ciphertexts $C^{\mathcal{R},I}_{\mathsf{rnd}}$ correctly. (Note that this is very similar to the "black-box confirmation" argument defined by Boneh and Franklin [6].) The main challenge in our proof consists of handling the case when $\mathcal{B}$ knows *some*, but not all user keys for I. In that case, we have to make sure that we output an identity in I that surely corresponds to a traitor. Similar traceability strategies were already considered, e.g., in [6] (but with a restriction on how the pirate box is built), and in [29, 10, 8] (for very different schemes). In the revocation setting the tracing technique of Tzeng and Tzeng [52] also considers semi-random ciphertexts as those from (2). However, the tracing algorithm of [52] assumes a pirate box with perfect decryption, and, more importantly, has to choose the analog of the revoked set $\mathcal{R}$ from (2) by itself. Dodis, Fazio, Kiayias, and Yung [16] consider imperfect pirate boxes and adversarially chosen revoked users in the revocation setting, but for a different scheme. Again, we stress that the novelty of our work lies in the fact that we extend Wee's generic view of revocation schemes by providing an EDDH-based trace-and-revoke variant which, in particular, generalizes (known) DDH-based and (new) DCR-based trace-and-revoke schemes.

2 Preliminaries

Notation. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. Throughout the paper, $k \in \mathbb{N}$ denotes the security parameter. For a finite set $\mathcal{S}$, we denote by $s \leftarrow \mathcal{S}$ the process of sampling s uniformly from $\mathcal{S}$. For a probabilistic algorithm A, we write $y \leftarrow A(x)$ for the process of running A on input x with uniformly chosen random coins, and assigning y the result. If A's running time is polynomial in k, then A is called probabilistic polynomial-time (PPT). A function $f : \mathbb{N} \to \mathbb{R}$ is negligible if it vanishes faster than the inverse of any polynomial (i.e., if $\forall c \exists k_0 \forall k \geq k_0 : |f(k)| \leq 1/k^c$). On the other hand, f is significant if it dominates the inverse of some polynomial (i.e., if $\exists c, k_0 \forall k \geq k_0 : f(k) \geq 1/k^c$).

(Binary) relations for hard search problems [54, 55]. Following the definition of (binary) relations for hard search problems in [55], let R_{pp} be a family of binary relations, where pp is a public parameter. We assume the existence of two PPT algorithms: given the security parameter k in unary, $\mathsf{SampP}(1^k)$ outputs

a public parameter pp together with a secret parameter sp, while $\mathsf{SampR}(1^k, pp)$ outputs a binary relation $(u, s) \in R_{pp}$ such that given only u it is hard to find s. (To make random coins r explicit, we may write $\mathsf{SampR}(1^k, pp; r)$.) Concretely, we define the one-way property of binary relations for hard search problems in the sense that with overwhelming probability over pp, for all u, there exists at most one s such that $(u, s) \in R_{pp}$, and, given an adversary A that gets pp and u with $(u, s) \leftarrow \mathsf{SampR}(1^k, pp)$, there exists an efficiently computable generator G_{pp} such that, for all A,

$$\mathsf{Adv}^{\mathsf{prg}}_A(k) := \Pr\left[A(pp, u, \mathsf{G}_{pp}(s)) = 1\right] - \Pr\left[A(pp, u, R) = 1\right],$$

with uniform R, is negligible in k.

Lagrange interpolation and Vandermonde matrices. Fix a field $\mathbb{F}$ and $d+1$ values $x_0, \ldots, x_d \in \mathbb{F}$. The Vandermonde matrix $V_{x_0,\ldots,x_d} \in \mathbb{F}^{(d+1)\times(d+1)}$ is defined as

$$V_{x_0,\ldots,x_d} := \begin{pmatrix} 1 & x_0 & \ldots & x_0^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_d & \ldots & x_d^d \end{pmatrix}.$$

It is easy to see that $\det(V_{x_0,\ldots,x_d}) = \prod_{i<j}(x_j - x_i)$; in particular, $V_{x_0,\ldots,x_d}$ is invertible iff all x_i are distinct. We can evaluate a polynomial $f(x) = a_0 + a_1 x + \cdots + a_d x^d$ at $x_0, \ldots, x_d$ via

$$(f(x_0), f(x_1), \ldots, f(x_d))^\top = V_{x_0,\ldots,x_d} \cdot (a_0, a_1, \ldots, a_d)^\top.$$

Conversely, given values $y_0, \ldots, y_d \in \mathbb{F}$, we can via

$$(a_0, a_1, \ldots, a_d)^\top = V^{-1}_{x_0,\ldots,x_d} \cdot (y_0, y_1, \ldots, y_d)^\top$$

compute coefficients $a_0, \ldots, a_n \in \mathbb{F}$ of a polynomial $f(x) = a_0 + a_1 x + \cdots + a_d x^d$ such that $f(x_i) = y_i$. It will be useful to perform such matrix-vector multiplications "in the exponent," where generally a matrix $M = (M_{i,j}) \in \mathbb{F}^{n\times n}$ is known, and a vector $x = (x_i) \in \mathbb{F}^n$ is given in the form $X = (X_i) = (g^{x_i})$ for some g. We will write

$$M \circ X := (Y_1, \ldots, Y_n) \quad \text{with} \quad Y_i := \prod_{j=1}^{n} X_j^{M_{i,j}}.$$

If we write $y = (y_i)$ for the "exponent vector" with $Y_i = g^{y_i}$, this achieves $M \cdot x = y$.

The Extended Decisional Diffie-Hellman assumption. In [25], Hemenway and Ostrovsky introduced the Extended Decisional Diffie-Hellman (EDDH) assumption. We say that the EDDH assumption holds for group $\mathbb{G}$ and subgroups $G, H \subseteq \mathbb{G}$ iff

$$\begin{aligned} \mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},H,D}(k) \;:=\; & \Pr\left[D(1^k, \mathsf{ord}(H), g, g^a, g^b, g^{ab}) = 1\right] \\ & - \Pr\left[D(1^k, \mathsf{ord}(H), g, g^a, g^b, g^{ab}h) = 1\right] \end{aligned}$$

is negligible for any PPT distinguisher D, for uniform group elements g and h from G and H, respectively, for uniform exponents a, b, and group order function ord. Additionally, we require that there exists a randomness extractor $\mathsf{G}^{\mathsf{eddh}}_{\mathbb{G},H}$ such that $\mathsf{G}^{\mathsf{eddh}}_{\mathbb{G},H}(h)$ with uniform $h \in H$ is pseudorandom. We note that the EDDH assumption can be instantiated under the DDH and the DCR assumption. (We refer to [25] for further details.)

3 First result: an EDDH-based TEHPS instance

Threshold extractable hash proof systems. We first restate the definition of threshold extractable hash proof systems (TEHPS) from [55], in which Wee explains several cryptosystems, i.e., threshold encryption, threshold signatures, and revocation schemes as arising from TEHPSs for a hard search problem with instances u and solution s (defined as above). For public key hk, we define a family of hash functions H_{hk}, which take as input a tag tag and an instance u, and output a hash value $\mathsf{H}_{hk}(tag, u)$. A TEHPS $\mathsf{TEHPS} = (\mathsf{Gen}, \mathsf{Share}, \mathsf{Pub}, \mathsf{Priv}, \mathsf{Ext})$ with tag space $\mathcal{T}$ consists of the following PPT algorithms:

Setup. Given the security parameter $k \in \mathbb{N}$, the threshold parameter $t \in \mathbb{N}$, and system parameters (pp, sp) (defined as above), $\mathsf{Gen}((pp, sp), 1^k, 1^t)$ generates a public key hk and a master secret key msk.

Key generation. $\mathsf{Share}(msk, tag)$, given the master secret key msk and a tag $tag \in \mathcal{T}$, generates a user secret key usk_{tag} for tag tag.

Public evaluation. $\mathsf{Pub}(hk, tag, r)$, given a public key hk, a tag $tag \in \mathcal{T}$, and random r, outputs a hash value $\mathsf{H}_{hk}(tag, u)$, with $(u, s) = \mathsf{SampR}(1^k, pp; r)$.

Private evaluation. $\mathsf{Priv}(usk_{tag}, u)$, given a user secret key usk_{tag} and an instance u, outputs a hash value $\mathsf{H}_{hk}(tag, u)$.

Extraction. $\mathsf{Ext}(u, (tag_i, \tau_i)_{i\in[t+1]})$, given an instance u, tags $(tag_i)_{i\in[t+1]} \in (\mathcal{T})^{t+1}$, and hash values $(\tau_i)_{i\in[t+1]}$, outputs a value s or $\perp$.

For all $k, t \in \mathbb{N}$ and with overwhelming probability over all values $(pp, sp) \leftarrow \mathsf{SampP}(1^k)$, for all $(hk, msk) \leftarrow \mathsf{Gen}((pp, sp), 1^k, 1^t)$, for all r, for all $(u, s) \leftarrow \mathsf{SampR}(1^k, pp; r)$, we require correctness, $(t+1)$-extraction, and t-simulation:

Correctness. For all $tag \in \mathcal{T}$, all $usk_{tag} \leftarrow \mathsf{Share}(msk, tag)$, we require that $\mathsf{Pub}(hk, tag, r) = \mathsf{H}_{hk}(tag, u) = \mathsf{Priv}(usk_{tag}, u)$.

$(t+1)$-extraction. For all distinct tags $(tag_i)_{i\in[t+1]} \in (\mathcal{T})^{t+1}$, and all hash values $(\tau_i := \mathsf{H}_{hk}(tag_i, u))_{i\in[t+1]}$, for $s = \mathsf{Ext}(u, (tag_i, \tau_i)_{i\in[t+1]})$, we require $(u, s) \in R_{pp}$.

t-simulation. For all distinct $(tag_i)_{i\in[t]} \in (\mathcal{T})^t$, there exists a PPT algorithm $\mathsf{SetupSim}$ such that distributions of

$$\omega = (hk, usk_{tag_1}, \ldots, usk_{tag_t})$$

in the following are statistically close: i.e., we require that

$$\begin{aligned}&\{\omega : (hk, msk) \leftarrow \mathsf{Gen}((pp, sp), 1^k, 1^t), (usk_{tag_i} \leftarrow \mathsf{Share}(msk, tag_i))_{i\in[t]}\}\\ &\overset{s}{\approx} \{\omega : (hk, usk_{tag_1}, \ldots, usk_{tag_t}) \leftarrow \mathsf{SetupSim}(pp, tag_1, \ldots, tag_t)\},\end{aligned}$$

where $\overset{s}{\approx}$ denotes statistically indistinguishable.

A TEHPS for the EDDH relation. We now construct a new EDDH-based threshold extractable hash proof system. As opposed to the DDH-based construction in [55], here, the group order of a subgroup $G \subseteq \mathbb{G}$ may not be known (i.e., in the case of DCR). Hence, we must avoid inversion operations in the exponent. We use a technique called "clearing the denominator" that, in a similar way, was used before but in different scenarios; e.g., in [46, 55, 2]. Further, fix a commutative group $\mathbb{G}$ and a subgroup $H \subseteq \mathbb{G}$ of (known) order n. We assume that a (proper) lower bound d on the smallest prime divisor of n is known. Let $G \subseteq \mathbb{G}$ be a cyclic subgroup of (potentially unknown) order q and let $\mathcal{K} := [B]$ such that for $x \leftarrow \mathcal{K}$, the value $x \bmod q$ is statistically close to uniform. In that case we will sample an exponent x uniformly from $[B]$, where $B = B' \cdot 2^k$ for an upper bound B' on q. (Such an upper bound B' will always be known.) Further, we need to specify a (binary) relation for the EDDH problem. Therefor, consider

$$R_{pp}^{\mathsf{eddh}} = \{(u, s) \in ((G \times \mathbb{G}) \times H) \mid u_2 = u_1^{sp} s\},$$

with $u = (u_1, u_2) \in (G \times \mathbb{G})$, for uniform $s \in H$, uniform $sp \in \mathcal{K}$. We set the public parameter pp to be (n, g, g^{sp}) and assume that we can sample g from G efficiently. Thus, sp and pp are efficiently samplable. (This completes the description of the SampP algorithm for the EDDH relation.) For the second EDDH-relation algorithm, we set $\mathsf{SampR}(1^k, pp; r)$ to output

$$(u, s) := ((g^r, (g^{sp})^r \cdot s), s),$$

for randomness $r \in \mathcal{K}$ and uniform $s \in H$. (This completes the description of SampR.) Further, we set $\mathsf{G}_{pp}(s) := \mathsf{G}_{\mathbb{G},H}^{\mathsf{eddh}}(s)$. Now, we are able to construct:

Construction 3.1 (EDDH-based TEHPS). Let a TEHPS $\mathsf{TEHPS}_{\mathsf{EDDH}} = (\mathsf{Gen}, \mathsf{Share}, \mathsf{Pub}, \mathsf{Ext}, \mathsf{Priv})$ with tag space $\mathcal{T} := [\min\{d, B\}] \subset \mathbb{Z}$, with d and B as above, be as follows:

Setup. $\mathsf{Gen}((pp, sp), 1^k, 1^t)$, with $pp =: (n, g, g^{sp})$, chooses a polynomial

$$f(x) := sp + a_1 x + \cdots + a_t x^t$$

over $\mathcal{K}$, with uniform exponents a_i, for $i \in [t]$. The output is the public key $hk := (n, \widetilde{g}, \widetilde{g}^{sp}, (\widetilde{g}^{a_i})_{i=1}^t)$, with $\widetilde{g} := g^v$, for uniform $v \leftarrow \mathcal{K}$, and master secret key $msk := (sp, (a_i)_{i=1}^t)$. We fix a hash function $\mathsf{H}_{hk}(tag, u) := u_1^{f(tag)}$, with $u = (u_1, u_2)$ and some tag $tag \in \mathcal{T}$. For randomness $r \in \mathcal{K}$, we have $(u, s) = ((\tilde{g}^r, \tilde{g}^{sp \cdot r} \cdot s), s) = \mathsf{SampR}(1^k, (n, \tilde{g}, \tilde{g}^{sp}); r)$. (Note that we re-randomize the g-elements of pp here.)

Sharing. $\mathsf{Share}(msk, tag)$, for $tag \in \mathcal{T}$, returns $usk_{tag} := f(tag)$.

Public Evaluation. Given a public key hk, a tag $tag \in \mathcal{T}$, randomness $r \in \mathcal{K}$, $\mathsf{Pub}(hk, tag, r)$ computes

$$\big(\widetilde{g}^{sp} \cdot \prod_{i=1}^{t} (\widetilde{g}^{a_i})^{tag^i}\big)^r \qquad \Big(= \big(\widetilde{g}^{f(tag)}\big)^r = u_1^{f(tag)} = \mathsf{H}_{hk}(tag, u)\Big),$$

with $(u, s) = \mathsf{SampR}(1^k, (n, \tilde{g}, \tilde{g}^{sp}); r)$ as above.

Private Evaluation. Given usk_{tag} and $u = (u_1, u_2)$, $\mathsf{Priv}(usk_{tag}, u)$ outputs $u_1^{usk_{tag}}$ $(= u_1^{f(tag)})$.

Extraction. $\mathsf{Ext}(u, (tag_i, \tau_{tag_i})_{i \in [t+1]})$, given $u = (u_1, u_2)$, tags $(tag_{t+1})_{i \in [t+1]} \in (\mathcal{T})^{t+1}$, and hash values $(\tau_{tag_i})_{i \in [t+1]}$, efficiently computes fractional Lagrangian coefficients $L_i(0) = \prod_{j=1, i \neq j}^{t+1} \frac{-tag_j}{tag_i - tag_j} \in \mathbb{Q}$ such that $f(0) = \sum_{i=1}^{t+1} L_i(0) \cdot f(tag_i) \bmod q$. (Note that the Lagrangian coefficients can be computed iff all tags $(tag_{t+1})_{i \in [t+1]}$ are distinct. If the tags are not distinct we output $\perp$.) Now, for $\Delta := \mathrm{lcm}\{\prod_{i,j \in [t+1], i \neq j} (tag_i - tag_j) \in \mathbb{Z}\}$ the values $\Delta \cdot L_i(0)$, for all $i \in [t+1]$, are integers. Thus, we are able to extract and output the value

$$\big(\big(\prod_{i=1}^{t+1} \tau_{tag_i}^{\Delta L_i(0)}\big)^{-1} \cdot u_2^{\Delta}\big)^{\Delta^{-1} \bmod n}.$$

(Note that n is always known.)

We now show correctness, $(t+1)$-extraction, and t-simulation of Construction 3.1.

Claim 3.2. *For all $t \in \mathbb{N}$, $\mathsf{TEHPS}_{\mathsf{EDDH}}$ from Construction 3.1 is correct, $(t+1)$-extractable, and t-simulatable.*

Proof sketch. For all $k, t \in \mathbb{N}$, with overwhelming probability over $(pp, sp) \leftarrow \mathsf{SampP}(1^k)$, for all r, for all $(u, s) \leftarrow \mathsf{SampR}(1^k, (n, \tilde{g}, \tilde{g}^{sp}); r)$, with $u = (u_1, u_2)$, for all $(hk, msk) \leftarrow \mathsf{Gen}((pp, sp), 1^k, 1^t)$, for all tags $tag \in \mathcal{T}$, all $usk_{tag} \leftarrow \mathsf{Share}(msk, tag)$, we have:

Correctness. Correctness is easy to verify, i.e., $\mathsf{Pub}(hk, tag, r) = \mathsf{H}_{hk}(tag, u) = \mathsf{Priv}(usk_{tag_i}, u)$.

$(t+1)$-extraction. For all distinct tags $(tag_i)_{i \in [t+1]} \in (\mathcal{T})^{t+1}$, all hash values $(\tau_i := \mathsf{H}_{hk}(tag_i, u))_{i \in [t+1]} (= (u_1^{f(tag_i)})_{i \in [t+1]})$, for Δ and fractional Lagrangian coefficients $L_i(0)$ as above, $\mathsf{Ext}(u, (tag_i, \tau_{tag_i})_{i \in [t+1]})$ yields

$$\begin{aligned}
&\big(\big(\prod_{i=1}^{t+1} \tau_{tag_i}^{\Delta L_i(0)}\big)^{-1} \cdot u_2^{\Delta}\big)^{\Delta^{-1} \bmod n} \overset{(*)}{=} \big(\big(u_1^{\Delta f(0)}\big)^{-1} \cdot (u_1^{sp} \cdot s)^{\Delta}\big)^{\Delta^{-1} \bmod n} \\
&= \big(u_1^{-\Delta sp} \cdot u_1^{\Delta sp} \cdot s^{\Delta}\big)^{\Delta^{-1} \bmod n} = s.
\end{aligned}$$

Recall that all $\Delta \cdot L_i(0)$, for $i \in [t+1]$, are integers and that we used Lagrangian interpolation in the exponent in $(*)$. Thus, we obtain s such that $(u, s) \in R_{pp}^{\mathsf{eddh}}$.

t-simulation. For all distinct tags $(tag_i)_{i \in [t+1]} \in (\mathcal{T})^{t+1}$, there exists a PPT algorithm $\mathsf{SetupSim}$ as follows: Choose uniformly $y_1, \ldots, y_t \leftarrow \mathcal{K}$ and set $f(tag_i) := y_i$, for $i \in [t]$. Further, set $\widehat{g} := g^v$, for uniform $v \leftarrow \mathcal{K}$, and set $\widehat{g}^{f(0)} := (g^{sp})^v = \widehat{g}^{sp}$. Note, that this will uniquely define a polynomial f of degree $\leq t$. Let Δ be as above but with $tag_{t+1} = 0$. That (implicitly) determines a vector

$$(\Delta a_0, \Delta a_1, \ldots, \Delta a_t)^\top := (\Delta \cdot V^{-1}_{tag_{t+1}, tag_1, \ldots, tag_t}) \cdot (sp, y_1, \ldots, y_t)^\top.$$

(That is every Δa_i can be written as linear combination of the y_i, with appropriate integer coefficients. Here, again, we use Δ to "clear the denominator" of V^{-1}'s entries.) Subsequently, output $(n, \widetilde{g}, \widetilde{g}^{a_0}, \widetilde{g}^{a_1}, \ldots, \widetilde{g}^{a_t})$, for $\widetilde{g} := \widehat{g}^{\Delta}$, and $(usk_{tag_1}, \ldots, usk_{tag_t}) := (y_1, \ldots, y_t)$. Thus, the distribution of the output of $\mathsf{SetupSim}$ and and the distribution of $(hk, (\mathsf{Share}(msk, tag_i))_{i \in [t]})$ are statistically indistinguishable. $\square$

Now, by [55, Theorems 1, 2, 3], we derive semantically secure threshold public key encryption, existentially unforgeable threshold signatures in the random oracle model, and semantically secure revocation schemes from the hardness of the EDDH assumption which — at least in the revocation case — yields a new DCR-based revocation scheme. We will now provide details about revocation schemes and recap from [55] how to build them from TEHPSs.

Revocation schemes. Opposed to a broadcast encryption scheme, where a set of privileged users $\mathcal{S} \subseteq \{1, \ldots, N\}$ (for number of users $N \in \mathbb{N}$) is given as input to the encryption function, a revocation scheme receives a set of revoked users $\mathcal{R} := \{1, \ldots, N\} \setminus \mathcal{S}$ as input instead. The system then guarantees that users in $\{1, \ldots, N\} \setminus \mathcal{R}$ are able to decrypt correctly while users in $\mathcal{R}$ cannot decrypt. We will not directly give a construction of a revocation scheme; rather we will define a revocable key encapsulation mechanism which canonically implies an revocation scheme, but allows for a simpler exposition.

Revocable key encapsulation mechanism. For simplicity, and following [55], we define the notion of a revocable key encapsulation mechanism (RKEM). An RKEM with identity space $\mathcal{ID}$ consists of the following PPT algorithms:

Setup. $\mathsf{Gen}(1^k, 1^t)$, given the security parameter $k \in \mathbb{N}$ and a revocation threshold $t \in \mathbb{N}$, generates a public key pk and a master secret key msk.

Key generation. $\mathsf{Share}(msk, id)$, given the master secret key msk and an identity $id \in \mathcal{ID}$, generates a user secret key usk_{id} for identity id.

Encapsulation. $\mathsf{Enc}(pk, \mathcal{R})$, given the public key pk and a subset $\mathcal{R} \subseteq \mathcal{ID}$ that contains the identities of up to t revoked users, outputs a ciphertext C and a corresponding key K.

Decapsulation. $\mathsf{Dec}(id, usk_{id}, C)$, given an identity id, a corresponding user secret key usk_{id}, and a ciphertext C, outputs a key K.

For correctness, we require that for all $k, t \in \mathbb{N}$, all $(pk, msk) \leftarrow \mathsf{Gen}(1^k, 1^t)$, all set $\mathcal{R} \subseteq \mathcal{ID}$ of up to t identities, all $(C, K) \leftarrow \mathsf{Enc}(pk, \mathcal{R})$, all identities $id \in \mathcal{ID} \setminus \mathcal{R}$, and all $usk_{id} \leftarrow \mathsf{Share}(msk, id)$, we have $\mathsf{Dec}(id, usk_{id}, C) = K$. We will not define security for RKEMs. We note that these notions can be defined in a straightforward way, and the RKEMs based on TEHPSs from [55] can be proven secure in this sense. (In fact, [55] only shows selective-identity security; we expect, however, that adaptive-identity security can be achieved along the lines of Dodis and Fazio [15].) As mentioned before, an RKEM implies a revocation scheme. That is, to build a revocation scheme from an RKEM, use the encapsulated key to symmetrically encrypt the message to be broadcasted; analogously, use the decapsulated key for symmetrically decryption.

RKEMs from TEHPSs. Following [55], we recap the construction of an revocable key encapsulation mechanism $\mathsf{RKEM} = (\mathsf{Gen}, \mathsf{Share}, \mathsf{Enc}, \mathsf{Dec})$ with identity space $\mathcal{ID} := \mathcal{T}$ from a threshold extractable hash proof system $\mathsf{TEHPS} = (\mathsf{Gen}', \mathsf{Share}', \mathsf{Pub}, \mathsf{Ext}, \mathsf{Priv})$ with tag space $\mathcal{T}$ as follows:

Setup. $\mathsf{Gen}(1^k, 1^t)$, given security parameter $k \in \mathbb{N}$ and revocation threshold $t \in \mathbb{N}$, samples $(pp, sp) \leftarrow \mathsf{SampP}(1^k)$ and outputs public-key-master-secret-key pair $(pk, msk) := \mathsf{Gen}'((pp, sp), 1^k, 1^t)$.

Key extraction. $\mathsf{Share}(msk, id)$, for $id \in \mathcal{ID}$, returns $usk_{id} \leftarrow \mathsf{Share}'(msk, id)$.

Encapsulation. $\mathsf{Enc}(pk, \mathcal{R})$, for public key pk and $\mathcal{R} \subseteq \mathcal{ID}$ of size exactly t, chooses a random value r, samples $(u, s) \leftarrow \mathsf{SampR}(1^k, pk; r)$, and computes $\tau_{id} := \mathsf{Pub}(hk, id, r)$, for $id \in \mathcal{R}$. The ciphertext is given by $C := (\mathcal{R}, u, (\tau_{id})_{id \in \mathcal{R}})$, the key is $K := \mathsf{G}_{pk}(s)$.

Decapsulation. $\mathsf{Dec}(id, usk_{id}, C)$, with usk_{id} and C as above, retrieves

$$s := \mathsf{Ext}(u, \mathcal{R} \cup \{id\}, (\tau_{id})_{id \in \mathcal{R}}, \mathsf{Priv}(usk_{id}, u))$$

and outputs $K := \mathsf{G}_{pk}(s)$.

Correctness is easy to verify. For semantic security, we point to [55, Theorem 3]. Hence, as a result, we derive an EDDH-based revocation scheme.

4 Second result: $((t+1)/2, \varepsilon)$-traceability of the EDDH-based RKEM instance

Trace-and-revoke schemes. A trace-and-revoke scheme connects the properties of a revocation scheme and the benefits of a traitor tracing scheme. As mentioned before, combining these is nontrivial (see [8, Section 4.1]). Following [6, 16, 10, 8], we define traceability of an RKEM. (Note, this implicitly defines traceability of a revocation scheme due to the results of Section 3 and, thus, we derive a trace-and-revoke

Experiment $\mathsf{Exp}^{\mathsf{trace}}_{\mathsf{RKEM},\mathsf{Trace},A,\varepsilon}(1^k)$	**Experiment** $\mathsf{Exp}^{\mathsf{sid\text{-}trace}}_{\mathsf{RKEM},\mathsf{Trace},A,\varepsilon}(1^k)$
$1^t \leftarrow A(1^k)$	$(1^t, \mathcal{C}) \leftarrow A(1^k)$
$(pk, msk) \leftarrow \mathsf{Gen}(1^k, 1^t)$	$(pk, msk) \leftarrow \mathsf{Gen}(1^k, 1^t)$
$(\mathcal{B}, \mathcal{R}) \leftarrow A^{\mathsf{Share}(msk,\cdot)}(pk)$	$\forall id \in \mathcal{C}$: $usk_{id} \leftarrow \mathsf{Share}(msk, id)$
$id \leftarrow \mathsf{Trace}^{\mathcal{B}(\cdot)}(msk, \mathcal{R})$	$(\mathcal{B}, \mathcal{R}) \leftarrow A(pk, (usk_{id})_{id\in\mathcal{C}})$
if A has queried $\mathsf{Share}(msk, id)$	$id \leftarrow \mathsf{Trace}^{\mathcal{B}(\cdot)}(msk, \mathcal{R})$
or $\mathrm{Q}_{\mathcal{B},\mathcal{R}} < \varepsilon$ return 0	if $id \in \mathcal{C}$ or $\mathrm{Q}_{\mathcal{B},\mathcal{R}} < \varepsilon$ return 0
return 1	return 1

Fig. 1. Security experiments for traceability and sid-traceability of an RKEM.

scheme.) Intuitively, we require an efficient algorithm Trace that can, from oracle access to a stateless pirated box $\mathcal{B}$, deduce the identity of at least one party that has been involved in the construction of $\mathcal{B}$. More concretely, suppose an adversary A corrupts a number of devices (i.e., obtains a number of user keys usk_{id}), and constructs a pirate box $\mathcal{B}$. Suppose that $\mathcal{B}$ successfully decrypts ciphertexts for an adversarially specified set $\mathcal{R}$ of revoked users. Then we want that Trace, given oracle access to $\mathcal{B}$, can deduce at least one of the identities id whose device A has corrupted. We will also define a relaxation of traceability, dubbed sid-traceability, in which the adversary has to commit to corrupted identities in advance, before even seeing the public key.

Definition 1 (Traceable/sid-traceable RKEM). *We say that that an adversary A is T-*valid *if, in experiment $\mathsf{Exp}^{\mathsf{trace}}_{\mathsf{RKEM},\mathsf{Trace},A}$ (defined in Figure 1), it always chooses $t \geq T$, it always outputs a set $\mathcal{R}$ of size at most t, and it always makes at most T Share queries. (Note that this definition does not actually depend on Trace, and that t is specified by A itself.) Furthermore, for given $pk, \mathcal{R}$, we define the* quality *of a pirate box $\mathcal{B}$ output by A as*

$$\mathrm{Q}_{\mathcal{B},\mathcal{R}} := \Pr\left[\mathcal{B}(C) = K \mid (C, K) \leftarrow \mathsf{Enc}(pk, \mathcal{R})\right].$$

An RKEM RKEM is (T, ε)-traceable *if there exists a PPT algorithm Trace (that may depend on T and ε), so that for every PPT T-valid A,*

$$\mathsf{Adv}^{\mathsf{trace}}_{\mathsf{RKEM},A}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{trace}}_{\mathsf{RKEM},\mathsf{Trace},A,\varepsilon}(k) = 1\right]$$

is negligible. RKEM *is* (T, ε)-traceable under selective-identity attacks (short: (T, ε)-sid-traceable) *if the analogous statement holds with respect to*

$$\mathsf{Adv}^{\mathsf{sid\text{-}trace}}_{\mathsf{RKEM},A}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{sid\text{-}trace}}_{\mathsf{RKEM},\mathsf{Trace},A,\varepsilon}(k) = 1\right]$$

and $\mathsf{Exp}^{\mathsf{sid\text{-}trace}}_{\mathsf{RKEM},\mathsf{Trace},A,\varepsilon}$, defined in Figure 1, in which A has to output an identity set $\mathcal{C}$ of corrupted users of size at most t in advance.

From sid-traceability to traceability. There is a trivial (yet expensive) way to convert sid-traceable RKEMs into traceable ones. Namely, we can simply guess the identities for which an adversary (adaptively) requests user keys. Concretely:

Lemma 1 (sid-traceable ⇒ traceable). *Let RKEM by a (T, ε)-sid-traceable RKEM with N identities. If $\binom{N}{T}$ is polynomial in k, then RKEM is also (T, ε)-traceable (with the same Trace algorithm). Concretely, for every adversary A on RKEM's traceability, there is an adversary A' of roughly the same complexity on RKEM's sid-traceability, such that $\mathsf{Adv}^{\mathsf{sid\text{-}trace}}_{\mathsf{RKEM},A'}(k) \geq \mathsf{Adv}^{\mathsf{trace}}_{\mathsf{RKEM},A}(k)/\binom{N}{T}$.*

Proof sketch. First, A' outputs a uniformly chosen subset $\mathcal{C} \subseteq \mathcal{ID}$ of size T, and receives a public key pk along with user keys usk_{id} for $id \in \mathcal{C}$. Then A' internally simulates A, answering A's Share queries using the usk_{id}. If A requests a user secret key for an identity $id \notin \mathcal{C}$, then A' fails. Otherwise, A' relays A's output $(\mathcal{B}, \mathcal{R})$. Since A' chooses $\mathcal{C}$ independently, the event that A' fails is independent of A's output. Besides, the probability that A' does not fail is at least $1/\binom{N}{T}$, which is significant. □

Relation to our second result. Our second result (below) shows the $((t+1)/2, \varepsilon)$-sid-traceability of an EDDH-based RKEM based on threshold extractable hash proofs. Our corresponding tracing algorithm will have a runtime that is linear in $\binom{N}{T}$. Thus, in that case, $\binom{N}{T}$ must be polynomial anyway, and the loss in Lemma 1 seems acceptable.

More about our tracing strategy. We propose a tracing strategy that is similar to the tracing techniques in the revocation setting given by [52, 16]. However, we stress that the tracing algorithm of [52] assumes a pirate box with perfect decryption, i.e., $\varepsilon = 1$, and chooses the revoked set $\mathcal{R}$ by itself. The tracing mode in [16] also considers imperfect decryption boxes, adversarially chosen revoked user sets, and, additionally, allows of querying user secret keys adaptively. (This is possible since their scheme allows to change the public key continuously even after the system setup.) Additionally, both, i.e., [52, 16], only address the DDH setting. Nevertheless, we stress that the novelty of our work lies in the fact that we propose a new generic view of trace-and-revoke schemes.

4.1 Warmup: $(1, 2/3)$-sid-traceability of the EDDH-based RKEM

We can now state our second result; i.e., we show the traceability of $\mathsf{RKEM}_{\mathsf{EDDH}}$ which is an EDDH-based RKEM as defined and constructed in Section 3. (This immediately translates to an EDDH-based trace-and-revoke scheme.) As a warmup, we first showcase the $(1, 2/3)$-sid-traceability of $\mathsf{RKEM}_{\mathsf{EDDH}}$.

Informal proof strategy. To explain the overall idea of our tracing algorithm, observe that the decryption of a ciphertext generated by Enc does not depend on which user key was used to decrypt. (This is necessary for correctness.) Hence, we cannot expect that a pirate box $\mathcal{B}$ can be traced by feeding it valid ciphertexts generated by Enc. Instead, we will feed $\mathcal{B}$ random ciphertexts of the form

$$C^{\mathcal{R}}_{\mathsf{rnd}} = (\mathcal{R}, u_1, (u_1^{f(id)} h^{z_{id}})_{id \in \mathcal{R}}, u_1^{f(0)} h^{z_0}) \quad \text{for uniform } h \in H \text{ and } z_{id}, z_0. \tag{3}$$

We will show that for such random ciphertexts, the result of the (honest) decryption depends on the identity of the used user key usk_{id}. Furthermore, a suitable reduction to the EDDH assumption will show that honestly generated ciphertexts are indistinguishable from random ones. Hence, Trace can go through the set of all possible identities id, and check how often $\mathcal{B}(C^{\mathcal{R}}_{\mathsf{rnd}})$ coincides with $\mathsf{Dec}(id, usk_{id}, C^{\mathcal{R}}_{\mathsf{rnd}})$. In case $\mathcal{B}$ outputs the same as Dec with probability close to $2/3$, chances are that we have found the pirate identity.We can formalize these claims:

Theorem 1 ($(1, 2/3)$-sid-traceability of $\mathsf{RKEM}_{\mathsf{EDDH}}$). *Assuming the EDDH assumption, we have that the RKEM $\mathsf{RKEM}_{\mathsf{EDDH}} = (\mathsf{Gen}, \mathsf{Share}, \mathsf{Enc}, \mathsf{Dec})$, with identity space $\mathcal{ID}$, polynomial number N of identities, and key derivation function $\mathsf{G}(s) = s$, is $(1, 2/3)$-sid-traceable. The corresponding tracing algorithm Trace runs for $\mathbf{O}(kN \log N)$ steps, and makes $\mathbf{O}(k \log N)$ oracle queries. Concretely, for every T-valid adversary A, there is an EDDH adversary D, such that*

$$\left|\mathsf{Adv}^{\mathsf{trace}}_{\mathsf{RKEM},A}(k)\right| \leq \mathbf{O}(2^{-k}),$$

for all k that satisfy

$$\left|\mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},H,D}(k)\right| \leq 1/9 - \varepsilon_{\mathsf{G}},$$

for negligible ε_{G}.

Proof. **The tracing algorithm.** First, $\mathsf{Trace}^{\mathcal{B}(\cdot)}(msk, \mathcal{R})$ approximates for every identity $id \in \mathcal{ID}$ the *random quality*

$$\mathrm{RQ}^{id}_{\mathcal{B},\mathcal{R}} := \Pr\left[\mathcal{B}(C^{\mathcal{R}}_{\mathsf{rnd}}) = \mathsf{Dec}(id, usk_{id}, C^{\mathcal{R}}_{\mathsf{rnd}})\right],$$

where the probability is over $\mathcal{B}$'s random coins and random $C^{\mathcal{R}}_{\mathsf{rnd}}$ as in (3). Concretely, say that for each $id \notin \mathcal{R}$, we check $\mathcal{B}(C^{\mathcal{R}}_{\mathsf{rnd}}) = \mathsf{Dec}(id, usk_{id}, C^{\mathcal{R}}_{\mathsf{rnd}})$ for $\mathbf{O}(k \cdot \log N)$ independent values of $C^{\mathcal{R}}_{\mathsf{rnd}}$. Then a standard argument (i.e., Hoeffding's inequality and a union bound) shows that we obtain approximations $\widetilde{\mathrm{RQ}}^{id}_{\mathcal{B},\mathcal{R}}$ of $\mathrm{RQ}^{id}_{\mathcal{B},\mathcal{R}}$, such that

$$\text{for all } id\text{:} \quad \left|\widetilde{\mathrm{RQ}}^{id}_{\mathcal{B},\mathcal{R}} - \mathrm{RQ}^{id}_{\mathcal{B},\mathcal{R}}\right| < 1/9, \tag{4}$$

except with probability $\mathbf{O}(2^{-k})$. After having obtained all these $\widetilde{\mathrm{RQ}}^{id}_{\mathcal{B},\mathcal{R}}$, Trace outputs an identity with maximal $\widetilde{\mathrm{RQ}}^{id}_{\mathcal{B},\mathcal{R}}$. The whole process takes $\mathbf{O}(Nk \log N)$ steps and (if we re-use $\mathcal{B}$-queries across different identities) $\mathbf{O}(k \log N)$ $\mathcal{B}$-queries.

Why tracing works. To analyze Trace, consider an adversary A in the 1-sid-traceability experiment. We assume without loss of generality that A always requests exactly one user key. Let id^* be the corresponding identity. Furthermore, we assume that the set $\mathcal{R}$ that A finally outputs contains exactly t identities, which we denote by $id_1^*, \ldots, id_t^*$. We finally assume $id^* \notin \mathcal{R}$. (If $id^* \in \mathcal{R}$, then any pirate box $\mathcal{B}$ that is able to decrypt with non-negligible probability would contradict $\mathsf{RKEM}_{\mathsf{EDDH}}$'s semantic security.) We denote by $\mathcal{B}$ the pirate box that A eventually outputs.

Claim 4.2. *There is a EDDH distinguisher D whose runtime is essentially that of the sid-traceability experiment with A, such that*

$$\mathrm{Q}_{\mathcal{B},\mathcal{R}} - \mathrm{RQ}^{id^*}_{\mathcal{B},\mathcal{R}} = \mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},H,D}(k). \tag{5}$$

Proof. On challenge input $n = \mathsf{ord}(H), g, u_1, g^y, Z = u_1^y h^b$, where either $b = 0$ or $b = 1$, D runs the first stage of the sid-traceability experiment to obtain 1^t and $\mathcal{C} = \{id^*\}$ from A. It then constructs an $\mathsf{RKEM}_{\mathsf{EDDH}}$ public key as follows. First, D re-randomizes its input to obtain t tuples

$$(g^{y_1}, Z_1 := u_1^{y_1} h^{bz_1}), \ldots, (g^{y_t}, Z_t := u_1^{y_t} h^{bz_t})$$

with $g^{y_i} := (g^y)^{\alpha_i} g^{\beta_i}$ and $Z_i := Z^{\alpha_i} u_1^{\beta_i} = u_1^{y\alpha_i} h^{b\alpha_i} u_1^{\beta_i} = u_1^{y_i} h^{b\alpha_i}$, for $i \in [t]$ and exponents α_i, β_i that are (statistically close to) uniform modulo n and modulo q. Hence, the y_i and $z_i := \alpha_i \bmod n$, for all i, are independently uniform. Now, choose an arbitrary set $\{id_1, \ldots, id_t\} \subset \mathcal{T}$ of t distinct identities that does not contain id^* and sample $y^* \leftarrow \mathcal{K}$. We (implicitly) define $f(x) := a_0 + a_1 x + \cdots + a_t x^t$ as the unique $\leq t$-degree polynomial over $\mathcal{K}$ that satisfies $f(id_i) = y_i$, for $i \in [t]$, and $f(id^*) = y^*$. Note that D cannot directly compute f. However, D does know id^* and all id_i, as well as all $\widehat{g}^{y_i} = \widehat{g}^{f(id_i)}$ and $\widehat{g}^{y^*} = \widehat{g}^{f(id^*)}$ (with $\widehat{g} := g^v$, for uniform exponent v). Hence, for $\Delta := \mathrm{lcm}\{\prod_{i,j \in [t+1], i \neq j} (id_i - id_j) \in \mathbb{Z}\}$, (with $id_{t+1} := id^*$), D can compute

$$(\widehat{g}^{\Delta a_0}, \ldots, \widehat{g}^{\Delta a_t})^\top := \left(\Delta \cdot V^{-1}_{id_1,\ldots,id_t,id^*}\right) \circ (\widehat{g}^{y_1}, \ldots, \widehat{g}^{y_t}, \widehat{g}^{y^*})^\top$$

without modular inversion in the exponent. Thus, for $\widetilde{g} := \widehat{g}^{\Delta}$, D can set up a public key $pk := (n, \widetilde{g}, (\widetilde{g}^{a_i})_{i=0}^t)$ for A, and run the next stage of the 1-sid-traceability experiment (using y^* as a user key for identity id^*). Now, D obtains a set $\mathcal{R} = \{id_1^*, \ldots, id_t^*\}$ of t revoked identities and a pirate box $\mathcal{B}$ from A. Consider the following $(t+1) \times (t+1)$-matrix $M = (M_{i,j})$ over $\mathcal{K}$ given by

$$M := V_{(id_1^*,\ldots,id_t^*,id^*)} \cdot V^{-1}_{(id_1,\ldots,id_t,id^*)}, \text{ so that } M \cdot \begin{pmatrix} f(id_1) \\ \vdots \\ f(id_t) \\ f(id^*) \end{pmatrix} = \begin{pmatrix} f(id_1^*) \\ \vdots \\ f(id_t^*) \\ f(id^*) \end{pmatrix}. \tag{6}$$

Note that M only depends on (and can be computed efficiently from) the id_i, the id_i^*, and id^*. Furthermore, since all respective identities in $\{id_i\}_i \cup \{id^*\}$ and $\{id_i^*\}_i \cup \{id^*\}$ are distinct, M is invertible. Now, D computes the vector

$$((Z_1')^\Delta, \ldots, (Z_t')^\Delta, (Z_{t+1}')^\Delta)^\top := (\Delta \cdot M) \circ (Z_1, \ldots, Z_t, Z_{t+1})^\top, \tag{7}$$

with $Z_{t+1} := u_1^{y^*}$. With these $t+1$ values $(Z_i')^\Delta$ and $t+1$ identities in $\{id_i^*\}_i \cup \{id^*\}$, we are able to obtain $(Z_0')^\Delta := (u_1^{f(0)} h^{f'(0) \cdot b})^\Delta$ through Lagrangian interpolation (without modular inversion in the exponent), with implicitly defined $\leq t$-degree polynomial f' such that $f'(id_i^*) = z_i$, for all i, and $f'(id^*) = 0$. Intuitively, f' is the "h-exponent" of the Z_i, resp. Z_i'. Finally, D hands a ciphertext $C := (\mathcal{R}, u_1, (Z_1')^\Delta, \ldots, (Z_t')^\Delta, (Z_0')^\Delta \cdot s)$ (with $(Z_0')^\Delta$ as above and $(Z_i')^\Delta$ as in (7), for $i \in [t]$, and uniform $s \in H$) to $\mathcal{B}$ to obtain a potential decryption K. If $K = \mathsf{Dec}(id^*, y^*, C)$, then D outputs 1, else 0. This completes our description of D.

We now turn to D's analysis. First observe that when $b = 0$, $Z_i = u_1^{y_i} = u_1^{f(id_i)}$ for all i, then (6) implies $Z'_i = u_1^{f(id_i^*)}$ for all i. Hence, C is distributed exactly like an honest encryption $\mathsf{Enc}(pk, \mathcal{R})$, and by correctness of $\mathsf{RKEM}_{\mathsf{EDDH}}$, we have

$$\begin{aligned} & \Pr\left[D(1^k, n, g, u_1, g^y, Z) = 1 \mid Z = u_1^y\right] \\ = \ & \Pr\left[\mathcal{B}(C) = K \mid (C, K) \leftarrow \mathsf{Enc}(pk, \mathcal{R})\right] \\ = \ & \mathrm{Q}_{\mathcal{B},\mathcal{R}}, \end{aligned} \tag{8}$$

Conversely, assume $b = 1$, we have $Z_i = u_1^{y_i} h^{z_i}$, for all $i \in [t]$, and $Z_{t+1} := u_1^{y^*}$. Consider the (implicitly) defined degree-$\leq t$ polynomial f' with $f'(id_i) = z_i$ for $i \in [t]$ and $f'(id^*) = 0$. In other words, $Z_i = u_1^{y_i} h^{f'(id_i)}$. By the interpolation properties of M, this sets $Z'_i = u_1^{y_i} h^{f'(id_i^*)}$ and thus $Z'_0 = u_1^{f(0)} h^{f'(0)}$. The ciphertext now includes t values $(Z'_i)^{\Delta}$ and a value $(Z'_0)^{\Delta} \cdot s$, in which the uniform value $s \in H$ blinds $h^{f'(0)}$. That means that, information-theoretically, the adversary sees t evaluations $f'(id_i^*)$ of a polynomial f' that has t degrees of freedom (through the z_i). Hence, D prepares a random ciphertext C distributed exactly as $C^{\mathcal{R}}_{\mathsf{rnd}}$ from (3). Thus,

$$\begin{aligned} & \Pr\left[D(1^k, n, g, u_1, g^y, Z) = 1 \mid Z = u_1^y h\right] \\ = \ & \Pr\left[\mathcal{B}(C^{\mathcal{R}}_{\mathsf{rnd}}) = \mathsf{Dec}(id^*, usk_{id^*}, C^{\mathcal{R}}_{\mathsf{rnd}})\right] \\ = \ & \mathrm{RQ}^{id^*}_{\mathcal{B},\mathcal{R}}. \end{aligned} \tag{9}$$

Taking (8) and (9) together shows (5) as desired. □

Claim 4.2 essentially says that the pirate box $\mathcal{B}$ decrypts even malformed, random ciphertexts just as decryption with the user key usk_{id^*} for the traitor identity id^* would. It remains to prove that this decryption really uniquely identifies the traitor id^*.

Claim 4.3. *For any fixed $pk, id^*, \mathcal{R}$, and any identity $id' \notin \mathcal{R} \cup \{id^*\}$, we have*

$$\mathrm{RQ}^{id'}_{\mathcal{B},\mathcal{R}} \leq 1 - \mathrm{Q}_{\mathcal{B},\mathcal{R}} + \varepsilon_{\mathsf{G}}, \tag{10}$$

for negligible ε_{G}.

Proof. We will prove that for any $pk, id^*, \mathcal{R}, id'$ as above, we have that

$$\Pr\left[\mathsf{Dec}(id^*, usk_{id^*}, C^{\mathcal{R}}_{\mathsf{rnd}}) = \mathsf{Dec}(id', usk_{id'}, C^{\mathcal{R}}_{\mathsf{rnd}})\right] \qquad \text{is negligible,} \tag{11}$$

where the probability is over a random $C^{\mathcal{R}}_{\mathsf{rnd}}$ as in (3). From (11), we can deduce (10) by a union bound on the events that $\mathcal{B}(C^{\mathcal{R}}_{\mathsf{rnd}}) = \mathsf{Dec}(id^*, usk_{id^*}, C^{\mathcal{R}}_{\mathsf{rnd}})$ and $\mathsf{Dec}(id^*, usk_{id^*}, C^{\mathcal{R}}_{\mathsf{rnd}}) \neq \mathsf{Dec}(id', usk_{id'}, C^{\mathcal{R}}_{\mathsf{rnd}})$. To show (11), recall that (honest) decryption under secret key usk_{id^*} computes K through a Lagrange interpolation in the exponent and post-processing. In particular, observe that upon input a random ciphertext $C^{\mathcal{R}}_{\mathsf{rnd}} = (\mathcal{R}, u_1, (u_1^{f(id)} h^{z_{id}})_{id \in \mathcal{R}}, u_1^{f(0)} h^{z_0})$, decryption will output $\mathsf{G}^{\mathsf{eddh}}_{\mathbb{G},H}(h^{z_0 - f^*(0)})$, for the unique degree-$\leq t$ polynomial f^* with $f^*(id) = z_{id}$, for $id \in \mathcal{R}$ and $f^*(id^*) = 0$. (We have $f^*(id^*) = 0$ since decryption uses $u_1^{usk_{id^*}} = u_1^{usk_{id^*}} \cdot h^0$ for interpolation.) Analogously, decryption under secret key $usk_{id'}$ yields $\mathsf{G}^{\mathsf{eddh}}_{\mathbb{G},H}(h^{z_0 - f'(0)})$, for the unique polynomial f' with $f'(id) = z_{id}$, for $id \in \mathcal{R}$ and $f'(id') = 0$. Since $id^* \neq id'$, we have $f^*(0) \neq f'(0)$, except with probability $1/n$. Thus, by the pseudorandomness of $\mathsf{G}^{\mathsf{eddh}}_{\mathbb{G},H}$, it follows that

$$\mathsf{G}^{\mathsf{eddh}}_{\mathbb{G},H}(h^{z_0 - f^*(0)}) \neq \mathsf{G}^{\mathsf{eddh}}_{\mathbb{G},H}(h^{z_0 - f'(0)}),$$

except with negligible probability ε_{G}. This shows the claim. □

Claim 4.3 upper bounds the probability that a decryption under the "wrong" identity yields the "right" result by accident. In particular, if we take $\mathrm{Q}_{\mathcal{B},\mathcal{R}} \geq 2/3$ in (10) and (5), we get

$$\mathrm{RQ}^{id^*}_{\mathcal{B},\mathcal{R}} - \mathrm{RQ}^{id'}_{\mathcal{B},\mathcal{R}} \ \geq \ 1/3 - \mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},H,D}(k) - \varepsilon_{\mathsf{G}} \qquad \text{for all } id' \notin \mathcal{R} \cup \{id^*\}.$$

For the approximations $\widetilde{\mathrm{RQ}}^{id}_{\mathcal{B},\mathcal{R}}$ of $\mathrm{RQ}^{id}_{\mathcal{B},\mathcal{R}}$ computed by Trace, this implies

$$\widetilde{\mathrm{RQ}}^{id^*}_{\mathcal{B},\mathcal{R}} - \widetilde{\mathrm{RQ}}^{id'}_{\mathcal{B},\mathcal{R}} \;\geq\; 1/9 - \mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},H,D}(k) - \varepsilon_{\mathsf{G}} \qquad \text{for all } id' \notin \mathcal{R} \cup \{id^*\}, \tag{12}$$

with overwhelming probability over the approximations. In particular, (12) implies that id^* maximizes $\widetilde{\mathrm{RQ}}^{id}_{\mathcal{B},\mathcal{R}}$ for sufficiently large k. Hence, if $\mathrm{Q}_{\mathcal{B},\mathcal{R}} \geq 2/3$, and $\mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},H,D}(k) \leq 1/9 - \varepsilon_{\mathsf{G}}$, and all the approximations are accurate in the sense of (4), then Trace outputs id^*. □

4.4 General case: $((t+1)/2, \varepsilon)$-sid-traceability of $\mathsf{RKEM}_{\mathsf{EDDH}}$

Why our tracing strategy for $T = 1$ does not work. First, observe that our concrete tracing strategy from the proof of Theorem 1 fails if A requests multiple user keys. For instance, A could use multiple user keys to distinguish valid from random ciphertexts(which would break Claim 4.2). Concretely, A could request two keys usk_{id_1} and usk_{id_2} and let $\mathcal{B}$ first check if a given ciphertext decrypts to the same value under both usk_{id_1} and usk_{id_2}. If the decryptions do not match, then $\mathcal{B}$ immediately fails. (Recall that our proof uses the fact that random ciphertexts decrypt differently under different keys.) Such a box $\mathcal{B}$ would be useless to our tracing algorithm Trace, since Trace feeds $\mathcal{B}$ only random ciphertexts. (See [28] for more details.)

How to adapt our strategy. A natural way to adapt our strategy — this essentially follows the "black-box confirmation" argument from [6] — would seem as follows. Given a set $I \subseteq \mathcal{ID}$ of identities, we can construct "semi-random ciphertexts" of the form

$$C^{\mathcal{R},I}_{\mathsf{rnd}} = (\mathcal{R}, u_1, (u_1^{f(id)} h^{f'(id)})_{id \in \mathcal{R}}, u_1^{f(0)} h^{f'(0)}) \quad \begin{array}{l}\text{for } f'(x) \in \mathbb{Z}_q[x] \text{ uniform} \\ \text{of degree} \leq t, \text{ but subject} \\ \text{to } f'(id) = 0 \text{ for } id \in I.\end{array} \tag{13}$$

We will also define the *random quality* $\mathrm{RQ}^{I}_{\mathcal{B},\mathcal{R}}$ of a box $\mathcal{B}$ relative to a given revoked set $\mathcal{R}$, and an identity set $I \subseteq \mathcal{ID}$:

$$\mathrm{RQ}^{I}_{\mathcal{B},\mathcal{R}} := \Pr\left[\mathcal{B}(C^{\mathcal{R},I}_{\mathsf{rnd}}) = \mathsf{Dec}(id, usk_{id}, C^{\mathcal{R}}_{\mathsf{rnd}}) \text{ for some } id \in I\right]. \tag{14}$$

Intuitively, ciphertexts $C^{\mathcal{R},I}_{\mathsf{rnd}}$ look consistent from the point of a pirate box that only knows user keys for identities in I. Hence, our tracing strategy for a larger number T of traitors will be as follows. We iterate over all $\binom{N}{T}$ identity subsets $I \subseteq \mathcal{ID}$ of size T, and approximate $\mathrm{RQ}^{I}_{\mathcal{B},\mathcal{R}}$. If the approximation indicates that $\mathrm{RQ}^{I}_{\mathcal{B},\mathcal{R}} \geq \varepsilon$, then we have a candidate for the set $\mathcal{C}$ of traitors. Unfortunately, there may be many candidates, and not all of them contain only traitors. To filter out one identity that surely is a traitor, we remove identities from I, one at a time. If the quality $\mathrm{RQ}^{I}_{\mathcal{B},\mathcal{R}}$ drops, we must have removed a traitor. (If the removed identity was no traitor, then $\mathcal{B}$ would not have noticed.) Again, this tracing strategy is similar to that of [6, 29, 52, 16, 10, 8]. More formally:

Theorem 2 ($((t+1)/2, \varepsilon)$-sid-traceability of $\mathsf{RKEM}_{\mathsf{EDDH}}$). *Assuming EDDH, $\mathsf{RKEM}_{\mathsf{EDDH}}$ is (T, ε)-sid-traceable for every $T \leq (t+1)/2$ for which $\binom{N}{T}$ is polynomial, and every significant ε. The corresponding tracing algorithm Trace runs for $\mathbf{O}(k\binom{N}{T}/\varepsilon^2)$ steps, where N denotes the number of identities in the system. Concretely, for every T-valid adversary A, there are adversaries D, E, F, such that*

$$\left|\mathsf{Adv}^{\mathsf{trace}}_{\mathsf{RKEM},A}(k)\right| \;\leq\; \mathbf{O}(2^{-k}),$$

for all k that satisfy

$$\left|\mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},H,D}(k)\right| + \left(\sum_{i=2}^{T}\binom{N}{i}\right) \cdot \left|\mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},H,E}(k)\right| + (N-T) \cdot \left|\mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},F}(k)\right| \;\leq\; \frac{\varepsilon}{3T}.$$

Proof. Fix T and $\varepsilon = \varepsilon(k)$ as above.

The tracing algorithm. First, $\mathsf{Trace}^{\mathcal{B}(\cdot)}(msk, \mathcal{R})$ iterates over all identity sets $I \subseteq \mathcal{ID}$ of size T and approximates the random quality $\mathrm{RQ}^I_{\mathcal{B},\mathcal{R}}$ (as defined in (14)). Again, a standard argument shows that with $\mathbf{O}(k/\varepsilon^2)$ $\mathcal{B}$-queries *for each* I, we obtain approximations $\widetilde{\mathrm{RQ}}^I_{\mathcal{B},\mathcal{R}}$ such that

$$\text{for all } I\text{:} \quad \left|\widetilde{\mathrm{RQ}}^I_{\mathcal{B},\mathcal{R}} - \mathrm{RQ}^I_{\mathcal{B},\mathcal{R}}\right| < \frac{\varepsilon}{3T}, \tag{15}$$

except with probability $\mathbf{O}(2^{-k})$. If no I with $\widetilde{\mathrm{RQ}}^I_{\mathcal{B},\mathcal{R}} > \varepsilon - \varepsilon/(3T)$ is found, Trace halts with output "`fail`". Otherwise, let $I = \{id_1, \ldots, id_T\}$ be such an I, and write $I_i := \{id_i, \ldots, id_T\}$. Now Trace approximates the values $\mathrm{RQ}^{I_i}_{\mathcal{B},\mathcal{R}}$ (for $1 \le i \le T$) as in (15). Finally, Trace outputs id_i for the smallest i that meets

$$\left|\widetilde{\mathrm{RQ}}^{I_i}_{\mathcal{B},\mathcal{R}} - \widetilde{\mathrm{RQ}}^{I_{i+1}}_{\mathcal{B},\mathcal{R}}\right| > \frac{\varepsilon}{T} \tag{16}$$

(or id_T if (16) holds for no $i < T$).

Why tracing works. To analyze Trace, consider an adversary A in the (T, ε)-sid-traceability experiment. We assume without loss of generality that A always requests a set $\mathcal{C}$ of exactly T user keys, and finally outputs a set $\mathcal{R} = \{id^*_1, \ldots, id^*_t\}$, along with a pirate box $\mathcal{B}$.

Our first claim essentially states that tracing does not output "`fail`" (except with small probability):

Claim 4.5. *There is a EDDH distinguisher D whose runtime is essentially that of the sid-traceability experiment with A, such that*

$$\mathrm{Q}_{\mathcal{B},\mathcal{R}} - \mathrm{RQ}^{\mathcal{C}}_{\mathcal{B},\mathcal{R}} = \mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},H,D}(k). \tag{17}$$

Proof sketch. We proceed as in the proof of Claim 4.2. First, D obtains $\mathcal{C}$ from A. Then D prepares a public key pk and user keys $(usk_{id})_{id\in\mathcal{C}}$ for A, and a ciphertext C for B, such that

- if $b = 0$, then C is an honest encryption, and
- if $b \neq 0$, then C is distributed as $C^{\mathcal{R},\mathcal{C}}_{\mathsf{rnd}}$.

(Note that Claim 4.2 can be seen as the special case $\mathcal{C} = \{id^*\}$.) Finally, D outputs 1 if and only if $\mathcal{B}(C) = \mathsf{Dec}(id, usk_{id}, C)$ for some $id \in \mathcal{C}$. The analysis of D is analogous to that from Claim 4.2. □

Next, we show that a pirate box $\mathcal{B}$ does not notice if we remove an identity $id' \notin \mathcal{C}$ from the set I in $C^{\mathcal{R},I}_{\mathsf{rnd}}$:

Claim 4.6. *There is a EDDH distinguisher E whose runtime is essentially that of the sid-traceability experiment with A, such that*

$$\mathrm{RQ}^I_{\mathcal{B},\mathcal{R}} - \mathrm{RQ}^{I\setminus\{id'\}}_{\mathcal{B},\mathcal{R}} = \left(\sum_{i=2}^{T}\binom{N}{i}\right)\cdot \mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},H,E}(k) \quad \begin{array}{l}\text{for all } I \subseteq \mathcal{ID} \text{ with}\\ 2 \le |I| \le T, \text{ and}\\ \text{every } id' \in I \setminus \mathcal{C}.\end{array} \tag{18}$$

Proof sketch. E runs A to obtain $\mathcal{C}$, and then guesses I and id' as above uniformly. Then, E prepares a public key pk and a ciphertext C for A, such that

- D knows the user keys usk_{id} for all $id \in \mathcal{C} \cup I \setminus \{id'\}$,
- if $b = 0$, then C is distributed as $C^{\mathcal{R},I}_{\mathsf{rnd}}$, and
- if $b \neq 0$, then C is distributed as $C^{\mathcal{R},I\setminus\{id\}}_{\mathsf{rnd}}$.

This can be done analogously to the proof of Claim 4.2. We stress, however, that at this point, we use that $T \le (t+1)/2$ to fix the implicitly defined polynomial f at all $id \in \mathcal{C} \cup I$. Finally, D outputs 1 iff $\mathcal{B}(C) = \mathsf{Dec}(id, usk_{id}, C)$ for some $id \in I \setminus \{id'\}$. The analysis of D is again analogous to that from Claim 4.2, and (18) follows through an averaging argument. □

Finally, we show that if tracing ends up with a singleton set $I = \{id\}$ (such that the random quality $\mathrm{RQ}^I_{\mathcal{B},\mathcal{R}}$ still is high), then we must have $id \in \mathcal{C}$.

Claim 4.7. *There is a EDDH adversary D whose runtime is essentially that of the sid-traceability experiment with A, such that*

$$\mathrm{RQ}^{\{id'\}}_{\mathcal{B},\mathcal{R}} = (N - T)\cdot \mathsf{Adv}^{\mathsf{eddh}}_{\mathbb{G},H,D}(k) \qquad \text{for all } id' \in \mathcal{ID} \setminus (\mathcal{C} \cup \mathcal{R}) \tag{19}$$

Proof sketch. D obtains $\mathcal{C}$ from A, and then guesses $id' \in \mathcal{ID} \setminus \mathcal{C}$ uniformly. Then D interprets its EDDH challenge as $g, g^{f(0)}, u_1, u_1^{f(0)} h^b$, and forms a public key pk for A (with otherwise uniform and known f) as in the proof of Claim 4.2. Now observe that the distributions $C_{\mathsf{rnd}}^{\mathcal{R}}$ and $C_{\mathsf{rnd}}^{\mathcal{R},\{id'\}}$ are identical as soon as $id' \notin \mathcal{R}$. (To see this, note that a uniform f' subject to $f'(id') = 0$ still has t degrees of freedom.) Hence, D can generate a ciphertext C with u_1 as above and $u_2 = u_1^{f(0)} h^b s$ for uniform $s \in H$ and uniformly and independently distributed τ_i. Regular decryption would decrypt C to $\mathsf{G}_{\mathbb{G},H}^{\mathsf{eddh}}(h^b s)$ under $usk_{id'}$. So whenever $\mathcal{B}(C)$ outputs $K = \mathsf{G}_{\mathbb{G},H}^{\mathsf{eddh}}(h^b s)$, D can solve its own EDDH challenge (by comparing K to $\mathsf{G}_{\mathbb{G},H}^{\mathsf{eddh}}(s)$), and through an averaging argument, we obtain (19). □

Finishing up. We can now put the pieces together and analyze the tracing algorithm Trace. Let us assume that all approximations are suitably close in the sense of (15). Then, by Claim 4.5, and the assumption about $\mathcal{B}$, Trace will not output "`fail`" (except with negligible probability). Besides, every time Trace finishes because (16) holds for an i, then Claim 4.6 (in contrapositive form) says that $id_i \in \mathcal{C}$ really must be a traitor. Finally, if no $i < T$ meets (16), then $\mathrm{RQ}_{\mathcal{B},\mathcal{R}}^{\{id_T\}}$ must be significant. Claim 4.7 implies that then, $id_T \in \mathcal{C}$ is a traitor. □

Potential generalizations of our tracing result. There are several dimensions in which one might want to improve our tracing result. We will comment on how our result can be generalized (and when a generalization seems problematic).

4.8 Potential generalizations of our tracing result

Full (instead of sid-)traceability. In case of a polynomial number of identities (which is necessary for efficient tracing anyway), Lemma 1 immediately yields:

Corollary 1 ($((t+1)/2,\varepsilon)$-traceability of $\mathsf{RKEM}_{\mathsf{EDDH}}$). *Assuming EDDH, $\mathsf{RKEM}_{\mathsf{EDDH}}$ is (T,ε)-traceable for every $T \le (t+1)/2$ for which $\binom{N}{T}$ is polynomial, and every significant ε.*

Generalization to Wee's factoring-based broadcast encryption scheme. Wee [55] also constructs an RKEM $\mathsf{RKEM}_{\mathsf{Fact}}$ whose semantic security is based on the factoring assumption. (For convenience, we have reproduced $\mathsf{RKEM}_{\mathsf{Fact}}$ in Construction 4.9.) Conceptually, $\mathsf{RKEM}_{\mathsf{EDDH}}$ and $\mathsf{RKEM}_{\mathsf{Fact}}$ are very similar. $\mathsf{RKEM}_{\mathsf{Fact}}$ works over a group $\mathbb{QR}_N^+ \subseteq \mathbb{Z}_N^*$ of size $\varphi(N)/4$ for a Blum integer N. In particular, ciphertexts are of the form $C = (\mathcal{R}, u, (u^{f(id)})_{id\in\mathcal{R}})$ for some degree-$\le t$ polynomial $f(x) = a_0 + a_1 x + \cdots + a_t x^t \in \mathbb{Z}_{\varphi(N)/4}[X]$ implicitly given in the public key. With $\mathsf{RKEM}_{\mathsf{Fact}}$, however, we always have $f(0) = a_0 = 2^{-(t+1)k} \bmod \varphi(N)/4$. Moreover, decryption of an honestly generated ciphertext yields $\mathrm{BBS}_N(s)$ for the BBS pseudorandom generator [5] and $s = u^{-2^k}$. These modifications (compared to $\mathsf{RKEM}_{\mathsf{EDDH}}$) enable a reduction to the factoring assumption; however, they also have a number of other effects.

Specifically, given a potential raw key s, we can always check if s is the correct decryption of a (consistent) ciphertext by checking if $s^{2^k} = u$ holds. This also gives a way to distinguish completely random ciphertexts $C_{\mathsf{rnd}}^{\mathcal{R}}$ from honestly generated ciphertexts. (Random ciphertexts $C_{\mathsf{rnd}}^{\mathcal{R}}$ yield uniform values s upon decryption, which can be recognized.) This leads to problems during the proof of Claim 4.2. Hence, we do not even know if Wee's factoring-based scheme $\mathsf{RKEM}_{\mathsf{Fact}}$ is $(1, 2/3)$-sid-traceable.

Now, we restate Wee's construction based on the hardness of factoring. (Again, this construction is similar to the EDDH-based construction $\mathsf{RKEM}_{\mathsf{EDDH}}$.)

Construction 4.9 (Wee's factoring-based RKEM [55]). Let $\mathsf{RKEM}_{\mathsf{Fact}}$ be as follows:

Setup. $\mathsf{Gen}(1^k, 1^t)$ chooses a Blum integer $N = PQ$, along with a uniform generator g of the group $\mathbb{QR}_N^+$ of signed quadratic residues.[3] Gen then chooses uniform exponents $a_i \in \mathbb{Z}_{\varphi(N)/4}$ (for $i \in [t]$) and sets

$$f(x) := 2^{-(t+1)k} + a_1 x + \ldots a_t x^t \bmod \varphi(N)/4.$$

Output is $pk := (N, g, (g^{a_i 2^{(t+1)k}})_{i=1}^t)$ and $msk := (P, Q, (a_i)_{i=1}^t)$.

[3] If we write $\mathbb{Z}_N = \{-(N-1)/2, \ldots, (N-1)/2\}$, and denote with $\mathbb{J}_N \subseteq \mathbb{Z}_N$ all elements with Jacobi symbol 1, then $\mathbb{QR}_N^+ = \{|x| : x \in \mathbb{J}_N\}$. When letting $x \cdot y := |xy|$ for $x, y \in \mathbb{QR}_N^+$, then $\mathbb{QR}_N^+$ is isomorphic to the group $\mathbb{QR}_N$ of quadratic residues modulo N. In particular, $|\mathbb{QR}_N^+| = \varphi(N)/4$. However, unlike $\mathbb{QR}_N$, $\mathbb{QR}_N^+$ is efficiently recognizable, which can be advantageous in some cases. See [26] for details and further references.

Sharing. $\mathsf{Share}(msk, id)$, for $id \in [\sqrt{N}/4]$, returns $usk_{id} := f(id) \bmod \varphi(N)/4$.
Encapsulation. $\mathsf{Enc}(pk, \mathcal{R})$ chooses an exponent[4] $r \leftarrow \mathbb{Z}_{\lceil N/4 \rceil}$, and computes

$$u := g^{r2^{(t+1)k}}, \quad s := g^{r2^{tk}} \ \left(= u^{2^{-k}} = u^{f(0)\cdot 2^{tk}} \right),$$
$$\tau_{id} := \left(g \cdot \prod_{i=1}^{t} \left(g^{a_i 2^{(t+1)k}} \right)^{id^i} \right)^{r} \ \left(= u^{f(id)} \right).$$

(for $id \in \mathcal{R}$). Ciphertext is $C := (\mathcal{R}, u, (\tau_{id})_{id\in\mathcal{R}})$, and key is $K := \mathrm{BBS}_N(s)$, where $\mathrm{BBS}_N(s)$ is the BBS pseudorandom generator [5] applied to s and modulo N.

Decapsulation. $\mathsf{Dec}(id, usk_{id}, C)$, with $usk_{id} = f(id)$ and C as above, sets $\tau_{id} := u^{f(id)\cdot 2^{(t+1)k}}$, and then retrieves $s := u^{f(0)\cdot 2^{tk}}$ from the τ_{id} through Lagrange interpolation in the exponent. Note that this has to be done via a "gcd in the exponent" argument (see [45]), since decryption cannot compute the fractional Lagrange coefficients directly. (This also explains the slightly tedious additional 2^{tk} factor in the exponent; we refer to [55] for details.)

References

[1] Michel Abdalla, Alexander W. Dent, John Malone-Lee, Gregory Neven, Duong Hieu Phan, and Nigel P. Smart. Identity-based traitor tracing. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 361–376. Springer, April 2007.

[2] Shweta Agrawal, Xavier Boyen, Vinod Vaikuntanathan, Panagiotis Voulgaris, and Hoeteck Wee. Functional encryption for threshold functions (or fuzzy ibe) from lattices. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 280–297. Springer, May 2012.

[3] Adam Barth, Dan Boneh, and Brent Waters. Privacy in encrypted content distribution using private broadcast encryption. In Giovanni Di Crescenzo and Avi Rubin, editors, *FC 2006*, volume 4107 of *LNCS*, pages 52–64. Springer, February / March 2006.

[4] Olivier Billet and Duong Hieu Phan. Efficient traitor tracing from collusion secure codes. In Reihaneh Safavi-Naini, editor, *ICITS*, volume 5155 of *Lecture Notes in Computer Science*, pages 171–182. Springer, 2008. ISBN 978-3-540-85092-2.

[5] Lenore Blum, Manuel Blum, and Mike Shub. Comparison of two pseudo-random number generators. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 61–78. Plenum Press, New York, USA, 1982.

[6] Dan Boneh and Matthew K. Franklin. An efficient public key traitor tracing scheme. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 338–353. Springer, August 1999.

[7] Dan Boneh and Moni Naor. Traitor tracing with constant size ciphertext. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 08*, pages 501–510. ACM Press, October 2008.

[8] Dan Boneh and Brent Waters. A fully collusion resistant broadcast, trace, and revoke system. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06*, pages 211–220. ACM Press, October / November 2006.

[9] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275. Springer, August 2005.

[10] Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 573–592. Springer, May / June 2006.

[11] Hervé Chabanne, Duong Hieu Phan, and David Pointcheval. Public traceability in traitor tracing schemes. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 542–558. Springer, May 2005.

[12] Benny Chor, Amos Fiat, and Moni Naor. Tracing traitors. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 257–270. Springer, August 1994.

[13] Cécile Delerablée, Pascal Paillier, and David Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *PAIRING 2007*, volume 4575 of *LNCS*, pages 39–59. Springer, July 2007.

[14] Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In Joan Feigenbaum, editor, *Digital Rights Management Workshop*, volume 2696 of *Lecture Notes in Computer Science*, pages 61–80. Springer, 2002. ISBN 3-540-40410-4.

[4] While Enc cannot choose a uniform exponent via $r \leftarrow \mathbb{Z}_{\varphi(N)/4}$, choosing $r \leftarrow \mathbb{Z}_{\lceil N/4 \rceil}$ is statistically close.

[15] Yevgeniy Dodis and Nelly Fazio. Public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 100–115. Springer, January 2003.

[16] Yevgeniy Dodis, Nelly Fazio, Aggelos Kiayias, and Moti Yung. Scalable public-key tracing and revoking. *Distributed Computing*, 17(4):323–347, 2005.

[17] Nelly Fazio, Antonio Nicolosi, and Duong Hieu Phan. Traitor tracing with optimal transmission rate. In Juan A. Garay, Arjen K. Lenstra, Masahiro Mambo, and René Peralta, editors, *ISC 2007*, volume 4779 of *LNCS*, pages 71–88. Springer, October 2007.

[18] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 480–491. Springer, August 1993.

[19] Amos Fiat and Tamir Tassa. Dynamic traitor training. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 354–371. Springer, August 1999.

[20] Eli Gafni, Jessica Staddon, and Yiqun Lisa Yin. Efficient methods for integrating traceability and broadcast encryption. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 372–387. Springer, August 1999.

[21] Juan A. Garay, Jessica Staddon, and Avishai Wool. Long-lived broadcast encryption. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 333–352. Springer, August 2000.

[22] Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 171–188. Springer, April 2009.

[23] Michael T. Goodrich, Jonathan Z. Sun, and Roberto Tamassia. Efficient tree-based revocation in groups of low-state devices. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 511–527. Springer, August 2004.

[24] Dani Halevy and Adi Shamir. The LSD broadcast encryption scheme. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 47–60. Springer, August 2002.

[25] Brett Hemenway and Rafail Ostrovsky. Extended-ddh and lossy trapdoor functions. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 627–643. Springer, 2012. ISBN 978-3-642-30056-1.

[26] Dennis Hofheinz and Eike Kiltz. The group of signed quadratic residues and applications. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 637–653. Springer, August 2009.

[27] Hongxia Jin and Jeffery Lotspiech. Renewable traitor tracing: A trace-revoke-trace system for anonymous attack. In Joachim Biskup and Javier López, editors, *ESORICS 2007*, volume 4734 of *LNCS*, pages 563–577. Springer, September 2007.

[28] Aggelos Kiayias and Moti Yung. Self protecting pirates and black-box traitor tracing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 63–79. Springer, August 2001.

[29] Aggelos Kiayias and Moti Yung. On crafty pirates and foxy tracers. In *Digital Rights Management Workshop*, pages 22–39, 2001.

[30] Aggelos Kiayias and Moti Yung. Breaking and repairing asymmetric public-key traitor tracing. In *Digital Rights Management Workshop*, pages 32–50, 2002.

[31] Aggelos Kiayias and Moti Yung. Traitor tracing with constant transmission rate. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 450–465. Springer, April / May 2002.

[32] Chong Hee Kim, Yong Ho Hwang, and Pil Joong Lee. An efficient public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In Chi-Sung Laih, editor, *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 359–373. Springer, November / December 2003.

[33] Kaoru Kurosawa and Yvo Desmedt. Optimum traitor tracing and asymmetric schemes. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 145–157. Springer, May / June 1998.

[34] Allison B. Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *2010 IEEE Symposium on Security and Privacy*, pages 273–285. IEEE Computer Society Press, May 2010.

[35] Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 206–224. Springer, May 2012.

[36] Tatsuyuki Matsushita and Hideki Imai. A public-key black-box traitor tracing scheme with sublinear ciphertext size against self-defensive pirates. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 260–275. Springer, December 2004.

[37] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, August 2001.

[38] Moni Naor and Benny Pinkas. Threshold traitor tracing. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 502–517. Springer, August 1998.

[39] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In Yair Frankel, editor, *FC 2000*, volume 1962 of *LNCS*, pages 1–20. Springer, February 2000.

[40] Birgit Pfitzmann. Trials of traced traitors. In Ross J. Anderson, editor, *Information Hiding*, volume 1174 of *Lecture Notes in Computer Science*, pages 49–64. Springer, 1996. ISBN 3-540-61996-8.

[41] Birgit Pfitzmann and Michael Waidner. Asymmetric fingerprinting for larger collusions. In *ACM CCS 97*, pages 151–160. ACM Press, April 1997.

[42] Duong Hieu Phan, David Pointcheval, Siamak Fayyaz Shahandashti, and Mario Strefler. Adaptive cca broadcast encryption with constant-size secret keys and ciphertexts. *Int. J. Inf. Sec.*, 12(4):251–265, 2013.

[43] Duong Hieu Phan, David Pointcheval, and Viet Cuong Trinh. Multi-channel broadcast encryption. In Kefei Chen, Qi Xie, Weidong Qiu, Ninghui Li, and Wen-Guey Tzeng, editors, *ASIACCS 13*, pages 277–286. ACM Press, May 2013.

[44] Reihaneh Safavi-Naini and Yejing Wang. Sequential traitor tracing. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 316–332. Springer, August 2000.

[45] Adi Shamir. The generation of cryptographically strong pseudo-random sequences. In Allen Gersho, editor, *CRYPTO'81*, volume ECE Report 82-04, page 1. U.C. Santa Barbara, Dept. of Elec. and Computer Eng., 1981.

[46] Victor Shoup. Practical threshold signatures. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 207–220. Springer, May 2000.

[47] Alice Silverberg, Jessica Staddon, and Judy L. Walker. Efficient traitor tracing algorithms using list decoding. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 175–192. Springer, December 2001.

[48] Thomas Sirvent. Traitor tracing scheme with constant ciphertext rate against powerful pirates. In *In Workshop on Coding and Cryptography, 2007*, 2007.

[49] Douglas R. Stinson and Ruizhong Wei. Key preassigned traceability schemes for broadcast encryption. In Stafford E. Tavares and Henk Meijer, editors, *Selected Areas in Cryptography*, volume 1556 of *Lecture Notes in Computer Science*, pages 144–156. Springer, 1998. ISBN 3-540-65894-7.

[50] Douglas R. Stinson and Ruizhong Wei. Combinatorial properties and constructions of traceability schemes and frameproof codes. *SIAM J. Discrete Math.*, 11(1):41–53, 1998.

[51] Dongvu Tonien and Reihaneh Safavi-Naini. An efficient single-key pirates tracing scheme using cover-free families. In Jianying Zhou, Moti Yung, and Feng Bao, editors, *ACNS 06*, volume 3989 of *LNCS*, pages 82–97. Springer, June 2006.

[52] Wen-Guey Tzeng and Zhi-Jia Tzeng. A public-key traitor tracing scheme with revocation using dynamic shares. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 207–224. Springer, February 2001.

[53] Yuji Watanabe, Goichiro Hanaoka, and Hideki Imai. Efficient asymmetric public-key traitor tracing without trusted agents. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 392–407. Springer, April 2001.

[54] Hoeteck Wee. Efficient chosen-ciphertext security via extractable hash proofs. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 314–332. Springer, August 2010.

[55] Hoeteck Wee. Threshold and revocation cryptosystems via extractable hash proofs. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 589–609. Springer, May 2011.

[56] Eun Sun Yoo, Nam-Su Jho, Jung Hee Cheon, and Myung-Hwan Kim. Efficient broadcast encryption using multiple interpolation methods. In Choonsik Park and Seongtaek Chee, editors, *ICISC 04*, volume 3506 of *LNCS*, pages 87–103. Springer, December 2004.