Optimal Traitor Tracing from Pairings

Mark Zhandry
NTT Research
mzhandry@gmail.com

Abstract

We use pairings over elliptic curves to give a collusion-resistant traitor tracing scheme where the sizes of public keys, secret keys, and ciphertexts are independent of the number of users. Prior constructions from pairings had size $\Omega(N^{1/3})$. Our construction is non-black box.

1 Introduction

Traitor tracing [CFN94] aims to deter piracy by enabling a distributor of encrypted content to trace the source of leaked decryption keys. *Collusion-resistant* traitor tracing guarantees security, even if arbitrarily many users pool their keys, and even if they attempt to obfuscate their keys within pirate decoders. A central goal has been to develop schemes with small keys and ciphertexts. The first scheme with parameters sub-linear in the number of users N uses cryptographic pairings to achieve parameters of size $\Theta(N^{1/2})$ [BSW06]. Subsequently, other tools such as functional encryption, obfuscation, and lattices have been used to obtain schemes with parameters independent of N [GGH+13, BZ14, GKW18]. However, despite being used for the first sub-linear traitor tracing scheme, pairings have so far been unable to replicate these subsequent successes using other tools. Indeed, the best known pairings-based schemes have parameters of size $\Theta(N^{1/3})$ [Zha20, GLW23]. A major open question for roughly 18 years has therefore been whether optimal traitor tracing is possible from pairings. Our main result resolves this question, showing the following:

Theorem 1. *Assume there exists a pairing over elliptic curves where either (1) K-LIN holds or (2) the pairing is symmetric and Decisional Bilinear Diffie-Hellman (DBDH) holds. Then there exists optimal (embedded identity) traitor tracing.*

Theorem 1 follows from this more general theorem, which is then instantiated via pairings:

Theorem 2. *Assume the existence of (1) selectively secure attribute-based encryption (ABE) for policies represented by log-depth arithmetic formula over an exponentially-large field $\mathbb{F}$ followed by a "not equal to zero" test, and (2) weak pseudorandom functions (PRFs) computable by log-depth arithmetic formula over $\mathbb{F}$* using pre-computation*. Then there exists an (embedded identity) traitor tracing scheme where all parameters sizes are independent of the number of users.*

Above, being computable "using pre-computation" means that we can compute $\mathsf{PRF}(k, r)$ in log-depth with the additional help of a pre-computed "hint" h_r that depends only on the input r but is independent k (see Definition 10). The needed ABE and PRF can be instantiated under assumptions on pairings; see Section 6 which shows how Theorem 1 follows from Theorem 2. Theorem 2 makes non-black box use of the weak PRF, and Theorem 1 makes non-black box use of group operations in the target group of the pairing.

1.1 Techniques

ABE+MFE=TT. To prove Theorem 2, we depart significantly from the usual approaches to build traitor tracing from pairings, and instead start with the abstract approach given in [GKW18] from the lattice-based setting. They construct traitor tracing from two other objects: attribute-based encryption (ABE) and mixed function encryption (MFE). MFE has secret keys for attributes x and ciphertexts with attributes y [1]. "Secret" ciphertexts have attributes and require the master secret key to generate. One security requirement, called *Ciphertext Attribute Security*, is that the adversary learns nothing about y except for the values $R(x, y)$ for secret key attributes x in the adversary's possession. Moreover, MFE has a *public* encryption procedure that produces "public" ciphertexts with no attributes. These ciphertexts decrypt to 1 under all secret keys, and a separate security notion called *Accept Security* requires that public ciphertexts are indistinguishable from secret ciphertexts with attributes y, unless one has a secret key for x such that $R(x, y) = 0$.

In [GKW18], it is shown how to build MFE from specific lattice techniques. Next, [GKW18] shows that attribute-based encryption (ABE) allows for upgrading MFE to include a payload message m, which we call "message-carrying MFE," or mcMFE. Correctness requires message recovery if $R(x, y) = 1$, and *Message Hiding Security* requires that the message is hidden if the adversary only has keys with $R(x, y) = 0$. A special case of mcMFE where the relation is $R(\mathsf{id}, t) = \mathbb{1}(\mathsf{id} \leq t)$ is called Private Linear Broadcast Encryption (PLBE), defined in [BSW06]. [BSW06] show that PLBE implies traitor tracing: the secret key for a user id is simply the secret key for $x = \mathsf{id}$. Traitor tracing ciphertexts are then public attribute-less ciphertexts. To trace a pirate decoder D, compute for each $t = 0, \cdots, N$ the probability p_t that D decrypts ciphertexts with attribute $y = t$. A good decoder implies large decryption probability on public ciphertexts, which then implies via Accept Security a large p_N, since all secret keys decrypt attribute N. Meanwhile, $t = 0$ cannot be decrypted by any secret key, meaning p_0 is small. Thus, there must be a gap between some p_t and p_{t-1}. Then the ciphertext attribute hiding implies this gap can only occur at identities $\mathsf{id} = t$ controlled by the adversary. [GKW18] instantiates the ABE from lattices, as was previously shown in [GVW13].

Attempting to Instantiate with Pairings. In [GKW18], this framework is instantiated with ABE and MFE built from lattices. Here, we will use ABE built from pairings. Unlike lattice-based ABE, ABE from pairings is only known to support shallow computations such as log-depth (arithmetic) formula, and other similar models [GPSW06, IW14, CGW15, LL20]. However, as long as our MFE is computable by such shallow programs, this will not be a limitation.

A more challenging issue is to construct a suitable MFE from pairings. In [GKW18], it is explained that we only need MFE where Ciphertext Attribute and Accept Security hold when the adversary can see only two ciphertexts, but an unbounded number of secret keys. If we ignore the public attribute-less encryption, then MFE coincides with secret key functional encryption (FE), and we can construct such a 2-ciphertext-secure FE scheme from even one-way functions [GVW12]. However, obtaining public attribute-less ciphertexts with Accept Security seems much more challenging, and [GKW18] use specific lattice techniques to achieve this.

In [CVW+18], a potentially more general approach is shown: they build the needed MFE by starting from any 2-ciphertext FE – which can be instantiated from general tools – and compiling it into an MFE scheme using a certain type of obfuscation called lockable obfuscation [GKW17, WZ17].

[1] The definition of MFE given in [GKW18] considers the case $R(x, y) = y(x)$ where y is interpreted as a function, but we consider a more abstract version here.

Unfortunately, we do not have anything remotely close to the functionality of lockable obfuscation from pairings. Therefore, following either [GKW18] or [CVW+18], the available techniques for realizing MFE appear firmly rooted in the capabilities of lattice-based cryptography.

Solution: Pseudorandom Ciphertexts. Our first observation is that we can obtain a form of pseudorandomness for ciphertexts in 2-ciphertext secure FE built from general tools. We start with a 2-ciphertext secure FE scheme such as [SS10, GVW12, KMUW18] built using garbled circuits/randomized encodings. At a high level, ciphertexts for these systems are secret keys for a CPA-secure encryption scheme, and the secret keys are encryptions of garbled circuit labels. By instantiating the CPA-secure scheme and garbled circuits correctly, then we show that a uniformly random ciphertext for the FE scheme – which again is comprised of keys for the CPA-secure scheme – looks exactly like an honest ciphertext for attribute y, as long as the adversary has only secret keys for attributes x such that $R(x, y) = 1$. This means we can publicly encrypt to this FE by simply by choosing a random bit string as our public ciphertext, resulting in an MFE scheme.

Unfortunately, even though ciphertext attribute security holds for two ciphertexts, pseudorandomness only holds for one ciphertext, and fails for if the adversary sees a second ciphertext. Thus, we do not achieve the 2-ciphertext Accept Security required to plug into [GKW18].

Why do we need 2-ciphertext security? We now briefly recall why two ciphertexts are needed. The issue is that security for MFE is described as a game between challenger and adversary, where the adversary has to distinguish the ciphertexts. On the other hand, traitor tracing corresponds to a *decoder*-based notion of security, where the adversary outputs a decoder, and the decoder (rather than the adversary) must distinguish ciphertexts. While it is intuitive that decoder-based security should follow from ordinary security, the naive proof – where the adversary simply runs the decoder it produces on the given ciphertext – actually does not work. This is because some decoders may have positive signed advantage (correct more often than not) while others have negative signed advantage (*in*correct more often than not). It is also infeasible to learn which is the case by testing with just a single ciphertext. The advantage of the adversary in the obvious "proof" is exactly the mean signed advantage over all decoders, which could in fact be zero even if all the decoders have large absolute distinguishing advantage. In this event, one can break decoder-based security whereas the reduction fails to contradict the assumed decoder-free notion of security.

Instead, using two ciphertexts, [GKW18] show that the reduction's advantage can be made the mean *squared* advantage. While this reduces the overall advantage in amplitude, it makes the advantage always positive. Every decoder with non-zero advantage will contribute positively to the reduction's advantage, allowing the proof of decoder-based security to go through. This issue and similar resolutions have also appeared in differential privacy contexts [BZ16, KMUZ16, KMUW18].

Our Solution: allowing some distinguishing advantage. We will attempt to obtain Accept Security for two ciphertexts, and hence decoder-based security, by composing several instances of a scheme with Accept Security for just one ciphertext, such as is guaranteed by the MFE from above. We can do this at either the level of the MFE, or at the level of the mcMFE after combining with ABE. It turns out our proofs are slightly simplified working at the level of mcMFE. Our basic idea is to have ℓ independent (mc)MFE systems. Each user will get an (mc)MFE secret key for each system. To generate a ciphertext, a random index $i \in [\ell]$ is chosen, which indicates which (mc)MFE to use. The ciphertext then consists of i together with the ciphertext c for the corresponding (mc)MFE.

The intuition is that, except with probability $1/\ell$, two ciphertexts will be encrypted under different (mc)MFE instances, allowing us to invoke the underlying 1-ciphertext security for those instances. This unfortunately does not quite work, as secret key sizes grow with ℓ, there is no way to make ℓ any larger than a polynomial, meaning there is a non-negligible chance of both ciphertexts using the same instance. When this happens, there is no security. This prevents us from achieving the existing notions of decoder-based Accept Security.

However, we show that this construction does give *some* guarantee. In particular, if we let Δ be the decoder's advantage, we show that $\mathbb{E}[\Delta^2] \lesssim 1/\ell$. This is not as good as what true 2-ciphertext security can achieve ($\mathbb{E}[\Delta]^2 \approx 0$), and as a result cannot be plugged directly into the framework of [GKW18]. However, by making additional modifications to the framework, we are able to use this weaker guarantee to nevertheless obtain optimal traitor tracing. We now explain.

Let Δ_i be the signed advantage of the decoder in distinguishing secret ciphertexts from public ciphertexts, when the ciphertext uses system i. Δ is then the mean of the Δ_i. We first show that we can bound $\mathbb{E}[\Delta_i \Delta_j] \approx 0$ for $i \neq j$. This adapts the existing proofs of decoder-based security from 2-ciphertext security, but the intuition is that now the proof will involve a ciphertext for i and a ciphertext for j; since $i \neq j$, this means each instance sees at most one ciphertext, so we can rely only on 1-ciphertext security. On the other hand, we do not have any non-trivial bounds for $\mathbb{E}[\Delta_i^2]$, since this would require two ciphertexts for the same instance i [2]. Fortunately, we can always trivially bound $\mathbb{E}[\Delta_i^2] \leq 1$. Taken together, this shows that $\mathbb{E}[\Delta^2] \lesssim 1/\ell$.

What can we do with such a bound? Unfortunately, even with $\mathbb{E}[\Delta^2] \approx 1/\ell$, it is possible for Δ to occasionally be very large, even 1. The only thing we can conclude with this bound is that Δ cannot be *too* high *too* often.

Next, if we take ℓ to be a sufficiently high constant ($\ell = 5$ suffices), we can get the following guarantee: suppose a decoder has advantage, say, 19/20 on public attribute-less ciphertexts. Then in the event that Δ is small (say, $\Delta \leq 9/10$), the decoder must have advantage at least 1/20 on secret ciphertexts with attributes. Moreover, by our bound on $\mathbb{E}[\Delta^2]$, we can conclude that this event happens with probability at least, say, 1/162 [3].

We also observe that while we only get weak decoder-based Accept Security, the construction preserves the 2-ciphertext security of the underlying scheme for the other security properties of the (mc)MFE – ciphertext-attribute and message hiding – and hence we have strong decoder-based security for these properties. Therefore, once we have a decoder that has has non-zero constant decryption probability for secret ciphertexts with attributes, we can employ the existing tracing techniques to accuse a user. Indeed, the existing tracing techniques, when using strong decoder-based security, work even for decoders that have an inverse-polynomially small decryption probability.

Remark 3. *We note that the 2-ciphertext secure FE constructions from the literature similarly starts with a 1-ciphertext secure FE, and compile it into a scheme secure against two (or more) ciphertexts. However, the construction is somewhat more complicated that ours. As a plus, they achieve actual 2-ciphertext security for ciphertext attribute hiding, whereas our simpler compiler is not enough to lift 1-ciphertext to 2-ciphertext security. However, those constructions do* not *work to give 2-ciphertext pseudorandomness.*

Remark 4. *One may be tempted to use our analysis to prove a similar statement for the other mcMFE security properties, removing the need for a 2-ciphertext secure mcMFE entirely. However,*

[2] 2-ciphertext security would show $\mathbb{E}[\Delta_i^2] = 0$.

[3] See the proof of Theorem 44, which gives a more general trade-off of parameters. Different settings of parameters could yield different concrete efficiency in the ultimate traitor tracing scheme.

there are multiple reasons this will not work. One problem is a technical issue that our proof actually requires the reduction to estimate the success probability of the decoder on public attribute-less ciphertexts; this is possible in our proof since such ciphertexts can be generated at will. When translating to the other decoder-based notions – which compare secret ciphertexts with different attributes – we would need to be able to generate many secret ciphertexts with attributes, which is not possible in the reduction. Another problem is that tracing accuses any user corresponding to a jump in decryption probability. In general, we need to accuse users even when the jump size is very small, namely smaller than the reciprocal of the number of collusions. But translating our analysis above to the other mcMFE security properties would only ensure that the jumps at honest users are sometimes bounded by a constant, meaning honest users will be sometimes accused. Instead, we still need strong decoder-based security (following from 2-ciphertext security) to argue that the jumps at honest users are tiny. Fortunately, our compiler preserves the 2-ciphertext security (and hence decoder-based security) for the other mcMFE security properties.

Finishing Touches. We are not quite done. The above scheme only traces decoders that have a high (but constant) success probability (e.g. 19/20). This is called a *threshold* scheme [NP98], and is typically considered to not be a complete solution to traitor tracing. Moreover, even with such a decoder, we are only guaranteed to be able to trace it when Δ is not too large (e.g. $\Delta \leq 9/10$), which is with constant probability (e.g. probability 1/162). Schemes that only guarantee tracing occasionally are called *risky* traitor tracing schemes [GKRW18]. Fortunately, [Zha20] shows a generic compiler which turns any risky threshold scheme into an ordinary (non-risky, non-threshold) scheme. The blow-up is polynomial in the security parameter as well as the inverse of riskiness and the error rate of trace-able decoders. For us, both these quantities are constant, meaning the blow-up is just polynomial in the security parameter, independent of the number of users. This scheme is capable of tracing identities coming from a polynomial-sized set (logarithmic bit-length), but we also show how to extend to a scheme with *embedded identities* where tracing recovers a polynomial-length identity.

We also need to ensure that ABE for low-depth function classes – the best we know how to achieve from pairings – is sufficient. Fortunately, the MFE construction that ABE is applied to only needs a weak PRF plus other simple algebraic operations. By using such a scheme computable in low-depth, we obtain a MFE construction where decryption can be evaluated by log-depth arithmetic formula. We just need the ABE scheme to handle the MFE decryption, meaning ABE for log-depth arithmetic formula suffices. Theorem 2 follows. We then get Theorem 1 by instantiating both the ABE and weak PRF scheme from pairings, which actually requires some work due to apparent gaps in the current understanding of low-depth cryptography; see discussion below in Section 1.2 and solutions in Section 6.

Remark 5. *Our framework above can also be adapted to the lattice setting. While this does not give any new feasibility result, it completely removes the need for lockable obfuscation from [CVW+18], resulting in a much more efficient construction.*

1.2 Discussion, Other Related Work, And Open Problems

Obfuscation from well-founded assumptions. In [JLS21, JLS22], it is shown how to construct obfuscation and functional encryption from assumptions on pairings, plus the assumed hardness of (a slightly nonstandard of) the learning parity with noise (LPN) problem and the assumed existence of pseudorandom generators (PRGs) with polynomial stretch where each output bit depends on a

constant number of input bits (called constant locality). The resulting functional encryption then implies PLBE suitable for traitor tracing. Going this route means making three qualitatively very different computational assumptions, all of which seem crucial to employing their techniques. In contrast, we only need to assume cryptographic pairings.

Parameter-sizes in traitor tracing systems. We only mention a few of the many works on traitor tracing, focusing exclusively on the collusion-bounded setting. The first work achieving sub-linear ciphertexts is [BSW06]. Many works focus on the size of the ciphertext alone, in which case it is possible to achieve constant-sized ciphertexts from general public key encryption [BN08, BP08]. However, these works have massive secret keys growing quadratically in the number of users. Sometimes, other trade-offs in terms of parameter sizes are possible, as shown in [Zha20]. When bounding all terms simultaneously, the best-known traitor tracing schemes prior to our work are: $N^{1/3}$ from pairings [Zha20, GLW23], constant-size from LWE [GKW18], or constant-size from obfuscation and related primitives [GGH+13, BZ14, GVW19].

Private vs public tracing. Our traitor tracing scheme requires the master secret key in order to trace, which we will call *secret tracing*. Many other schemes also require secret tracing, such as the aformentioned works of [BSW06, Zha20, GLW23]. Alternatively, some schemes have *public tracing* where anyone can trace. When restricting to public tracing, the best known schemes are those from obfuscation and related primitives [GGH+13, BZ14, GVW19], as well as [BW06] who achieve $N^{1/2}$-sized parameters from pairings. An interesting open question is then to obtain publicly traceable traitor tracing systems from pairings (or lattices) with constant-sized parameters.

Black-box vs non-black box use of cryptography. Our construction makes non-black box use of the pseudorandom function; in contrast the bulk of traitor tracing schemes from the literature, including all prior pairing-based schemes, only require black-box use of cryptography[4]. An interesting question is therefore whether traitor tracing based on black-box use of pairings can do better than $N^{1/3}$, or whether there is a lower-bound. We note that optimal traitor tracing from LWE [GKW18] also makes non-black-box use of cryptography.

Log-depth Cryptography. Log-depth cryptosystems are known from a number of building blocks [NR97, NRR00, BPR12, ABG+14]. In our work, we use log-depth-computable weak PRFs from certain cryptographic groups. Indeed, in Section 6, we explain how ElGamal encryption [ElG84] and generalizations can be viewed as weak PRFs computable in log-depth (with pre-computation).

While not necessary for this work, our study of low-depth PRFs lead us to the following interesting question. Namely, while we are able to obtain *weak* PRFs computable in log-depth from K-LIN, it does not appear known whether the same is possible for *strong* PRFs. In [NR97], a log-depth strong PRF is given based on DDH in the multiplicative group $\mathbb{Z}_p^*$. While this construction generalizes to have provable security under K-LIN [LW09, EHK+13], it is not clear if the computations are still log-depth. Concretely, whereas [NR97] uses an iterated *scalar* multiplication in the exponent, generalizations [LW09, EHK+13] use an iterated multiplication of $K \times K$ *matrices*. Iterated scalar multiplication can be computed by log-depth boolean formula [BCH84], but the same is unlikely to

[4]Black-box use of cryptography is not to be confused with black box tracing algorithms that only make queries to a decoder rather than inspect its code. The vast majority of schemes in the literature, including ours, employ tracing algorithms making black box use of the decoder. See [Zha21] for an exception.

be true for the iterated multiplication of non-constant-size matrices [5]. Note that for weak PRFs as we need, we can use an ElGamal-like structure, which we show *is* computable in log-depth, with appropriate pre-computation.

Another interesting question is the following. The PRF from [NR97] (and also the weak PRF we use) is only log-depth computable if the underlying group is the multiplicative group over a finite field. However, it is not clear if this generalizes to other groups used in cryptography, such elliptic curve groups. Note that pairings are groups over elliptic curves. However, the target group in pairings fortunately is a finite-field group, and typical assumptions in the pairing imply corresponding assumptions in the target group. Hence, while we cannot use the source group of a pairing for our weak PRF, we can use the target group instead.

Remark 6. *We also observe that [NR97] only claims to be log-depth in the multiplicative group of* prime-order *fields. This is not sufficient for us, since the target group of pairings typically lies in a field of prime-power order. We show how to extend to arbitrary fields in Section 6. This involves performing iterated field multiplications in log depth for arbitrary fields, which follows standard techniques but to the best of our knowledge had not appeared in the literature before.*

1.3 Paper Outline

Here, we explain how Theorem 2 follows from a combination of known and new results, the new results being proved in various sections of this paper. Recall that we assume a weak PRF computable by log-depth arithmetic formula over a field $\mathbb{F}$, as well as ABE for this class of formula.

- NEW, Section 3, Theorem 27: We show that weak PRFs imply a Mixed Functional Encryption (MFE) construction that is both 2-Bounded Selective Ciphertext Attribute Hiding (2-SEL-CTXT) secure and 1-Bounded Selective Accept (1-SEL-ACC) secure (Definitions given in Section 2). If the weak PRF can be computed via log-depth arithmetic formula (using pre-computation), then so can the decryption of our MFE.

- [GKW18] (described formally in Theorem 22): We then use the assumed ABE to lift the MFE into an *message carrying* MFE (mcMFE), which is also 2-SEL-CTXT secure and 1-SEL-ACC secure as well as 2-Bounded Selective Message-Hiding (2-SEL-M) secure (defined in Section 2).

- [GKW18] (described formally in Theorem 23): In the resulting mcMFE, the decoder-based security notions SEL-DEC-CTXT and SEL-DEC-M (defined in Section 2) follow from 2-SEL-CTXT and 2-SEL-M security, respectively, following [GKW18]. However, since we only have 1-SEL-ACC security, the decoder-based notion SEL-DEC-ACC does *not* follow.

- NEW, Section 4, Theorem 44: On the other hand, we show how to compile a mcMFE scheme with 1-SEL-ACC security into one satisfying a *weak* decoder version of Accept security, (called weak SEL-DEC-ACC security, Defined in Section 4).

[5] In the case of *constant* K, we can get very close to log-depth, namely depth $O(\log(n)\log^*(n))$ [All04]. Note also that iterated matrix multiplication for constant-sized matrices can be carried out by log-depth *arithmetic* formula. However, for applications to PRFs, the iterated multiplication needs to be followed by a group exponentiation, which requires the bit representation of the matrix product. With arithmetic formula, this is not possible. Hence, it seems boolean formula are needed for the iterated multiplication, even if we ultimately want an arithmetic formula. It appears open to achieve truly log-depth boolean formula for iterated multiplication of even constant-sized matrices.

- NEW, Section 5, Theorems 48 and 49: Such a mcMFE with SEL-DEC-CTXT, SEL-DEC-M, and *weak* SEL-DEC-ACC security then gives an *risky threshold* tracing scheme with asymptotically optimal parameters (variants of traitor tracing defined in Section 2).
- [Zha20], (described formally in Theorem 21): Such a risky threshold tracing scheme implies an optimal (non-risky, non-threshold) traceable scheme, completing the proof of Theorem 2.

In Section 6 we then discuss how to instantiate the assumed ABE and weak PRF. In particular, both can be instantiated from either the DBDH or K-LIN assumptions on pairings over elliptic curves. This proves Theorem 1.

2 Definitions and Notation

Let PPT denote "probabilistic polynomial time." Let NC^1 be the set of functions with inputs/outputs in $\{0,1\}^*$ computable by polynomial-time-uniform log-depth boolean formula. For a family $\mathbb{F} = (\mathbb{F}_\kappa)_\kappa$ of fields, let $\mathsf{NC}^1(\mathbb{F})$ be the set of functions computable by polynomial-time-uniform *arithmetic* formula over $\mathbb{F}$. We can interpret $\{0,1\} \subseteq \mathbb{F}$ and can "arithmetic-ize" any boolean formula, showing that $\mathsf{NC}^1 \subseteq \mathsf{NC}^1(\mathbb{F})$ for any family of fields $\mathbb{F}$. Finally, let $\mathsf{NC}^1_{\neq 0}(\mathbb{F})$ be the set of functions whose inputs are vectors over $\mathbb{F}$ and outputs are in $\{0,1\}$, computable as $\mathbb{1}(f(x) \neq 0)$, where $f(x) \in \mathsf{NC}^1(\mathbb{F})$. Here $\mathbb{1}(X)$ is the indicator function, which is 1 if X is true and 0 if X is false.

We will sometimes parameterize families of objects by more then one input, in which case we write, e.g., $(R_{\kappa_1,\kappa_2})_{\kappa_1,\kappa_2}$. Note that we can turn such a singe-parameter family using index (κ_1, κ_2), which is mapped to the integers in a standard way.

2.1 Traitor Tracing

We define traitor tracing following the modern conventions established in [NWZ16, GKRW18, GKW18]. Our exact formalization is similar to that from [Zha20]. A traitor tracing scheme for a key space $\mathcal{K} = (\mathcal{K}_\lambda)_\lambda$ is a tuple $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Trace})$ of probabilistic polynomial time (PPT) algorithms with the following syntax:

$$\begin{aligned}
\mathsf{Setup}(1^\lambda, 1^\nu) &\to (\mathsf{mpk}, \mathsf{msk}) \\
\mathsf{KeyGen}(\mathsf{msk}, \mathsf{id}) &\to \mathsf{sk}_{\mathsf{id}} \\
\mathsf{Enc}(\mathsf{mpk}) &\to (c, k) \\
\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, c) &\to k \\
\mathsf{Trace}^{\mathsf{D}}(\mathsf{msk}, 1^N, 1^{1/\epsilon}) &\to A
\end{aligned}$$

where

$$\begin{aligned}
\mathsf{mpk} &: \text{ master public key} \\
\mathsf{msk} &: \text{ master secret key} \\
\mathsf{id} \in \{0,1\}^\nu &: \text{ user's identity} \\
\mathsf{sk}_{\mathsf{id}} &: \text{ secret key for } \mathsf{id} \\
c &: \text{ ciphertext} \\
k \in \mathcal{K}_\lambda &: \text{ encapsulated key} \\
\mathsf{D} &: \text{ pirate decoder} \\
N &: \text{ number of users} \\
\epsilon &: \mathsf{D}\text{'s advantage} \\
A \subseteq \{0,1\}^\nu &: \text{ accused users}
\end{aligned}$$

Above, $\mathsf{Trace}^{\mathsf{D}}$ means that Trace makes queries to D, each query incurring unit cost.

Definition 7 (Correct Traitor Tracing)**.** *A traitor tracing scheme Π is* correct *if, for every polynomial $\nu(\lambda)$ there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$ and $\mathsf{id} \in \{0,1\}^{\nu(\lambda)}$:*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, c) \neq k : \begin{smallmatrix}(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\nu(\lambda)}) \\ (c,k) \leftarrow \mathsf{Enc}(\mathsf{mpk}) \\ \mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{msk},\mathsf{id})\end{smallmatrix}\right] \leq \mathsf{negl}(\lambda) \ .$$

Definition 8 ((ϵ, δ)-Threshold Risky Traceability)**.** *Let $\epsilon(\lambda)$ and $\delta(\lambda)$ be a functions. Π is (ϵ, δ)-threshold risky traceable if, for every polynomial $\nu(\lambda)$ and every PPT stateful adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that for all $\lambda \in \mathbb{N}$, the advantage of $\mathcal{A}$ is at most $\mathsf{negl}(\lambda)$ in the following experiment:*

- *Run $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\nu(\lambda)})$ and send mpk to $\mathcal{A}$.*
- *$\mathcal{A}$ can now make arbitrarily-many queries on identities $\mathsf{id} \in \{0,1\}^{\nu(\lambda)}$, and receives $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{id})$. Let C be the set of queried id.*
- *$\mathcal{A}$ produces a decoder D and an integer 1^N represented in unary such that $|C| \leq N$. Run $A \leftarrow \mathsf{Trace}^{\mathsf{D}}(\mathsf{msk}, 1^N, 1^{1/\epsilon(\lambda)})$.*
- *Let $\mathsf{Good}_{\epsilon(\lambda)}(\mathsf{D}, \mathsf{mpk})$ be the event that $2 \times \Pr\left[\mathsf{D}(c, k_b) = b : \substack{(c,k_0) \leftarrow \mathsf{Enc}(\mathsf{mpk}) \\ b \leftarrow \{0,1\}, k_1 \leftarrow \mathcal{K}_\lambda}\right] - 1 \geq \epsilon(\lambda)$. The advantage of $\mathcal{A}$ is the maximum of $\Pr[A \not\subseteq C]$ and $\delta(\lambda) \times \Pr[\mathsf{Good}_{\epsilon(\lambda)}(\mathsf{D}, \mathsf{mpk})] - \Pr[|A| > 0]$.*

We say Π is δ-risky traceable if it is (ϵ, δ)-risky threshold traceable for all inverse polynomials ϵ. We say Π is ϵ-threshold traceable if it is $(\epsilon, 1)$-risky threshold traceable. Finally, we say that Π is traceable if it is 1-risky traceable.

The requirement that $\Pr[A \not\subseteq C]$ is negligible means that honest users outside of C are never accused. The requirement that $\delta(\lambda) \times \Pr[\mathsf{Good}_{\epsilon(\lambda)}(\mathsf{D}, \mathsf{mpk})] - \Pr[|A| > 0]$ is negligible says that if D is able to distinguish honest keys k from random ones in $\mathcal{K}_\lambda$ (in other words, the decoder is "good"), then there is roughly a $\delta(\lambda)$ chance that the decoder will be traced to some user. When $\delta = 1$, this means that good decoders are essentially always traced to a user. Combined with never accusing an honest user, this means that an adversarial user must accused if the decoder is good.

Notice that the syntax of a traitor tracing scheme as in Definition 8 does not allow the run-times or parameters to depend on the number of keys given out (except for Trace since it depends on N).

Variations. In an *index-only* scheme, we place the additional requirement on the adversary that for each queried id, $\mathsf{id} \in [N]$. Since N must be a polynomial, we can always upper bound $N \leq 2^\lambda$. Hence, we will always take $\nu = \lambda$, and omit ν as an input to Setup.

Schemes defined as in Definition 8 without the index-only modification are typically called *embedded identity* schemes, first explored by [NWZ16]. An intermediate notion called *index-based embedded identity* [GKW19] has identities comprising two parts (i, id), where $\mathsf{id} \in \{0,1\}^\nu$ represents the identity, and $i \in \{0,1\}^\lambda$ is an index. It is guaranteed that the indices of all secret keys given out are distinct, and that the N produced by $\mathcal{A}$ bounds all indices i. Thus, i serves the role of index in an index-only scheme, while id is an identity additionally recovered during tracing.

Note that the key-space $\mathcal{K}$ is mostly irrelevant, since we can expand the key size either by applying pseudorandom generators or by combining several encapsulated keys.

Above, we give a variation of traitor tracing that is a key encapsulation mechanism (KEM). By having the key k one-time pad the message, we can readily turn such a scheme into a traitor tracing scheme that actually encrypts messages. We focus on KEMs for ease of notation.

Optimal Schemes. Our formalization of traitor tracing guarantees that parameters and run-times are independent of the number of colluding users, except for Trace (since it depends on N). We will call such schemes as *optimal*. Most schemes in the literature are not optimal, and have running times and parameter sizes depend polynomially on the upper bound N on the number of users.

2.2 Weak PRFs

Definition 9. *A weak PRF with key space $\mathcal{K} = (\mathcal{K}_\lambda)_\lambda$, input space $\mathcal{R} = (\mathcal{R}_\lambda)_\lambda$ and output space $\mathcal{O} = (\mathcal{O}_\lambda)_\lambda$ is a deterministic polynomial time algorithm $\mathsf{PRF} : \mathcal{K}_\lambda \times \mathcal{R}_\lambda \to \mathcal{O}_\lambda$ such that, for any PPT adversary $\mathcal{A}$ and polynomial $\ell(\lambda)$, there exists a negligible $\mathsf{negl}(\lambda)$ such that for every $\lambda \in \mathbb{N}$,*

$$\left\| \Pr\left[\mathcal{A}(\ (r_i, c_i)_i\ ) = 1 : \begin{smallmatrix} k \leftarrow \mathcal{K}_\lambda, \\ r_1, \cdots, r_\ell \leftarrow \mathcal{R}_\lambda \\ c_i \leftarrow \mathsf{PRF}(k, r_i) \end{smallmatrix}\right] - \Pr\left[\mathcal{A}(\ (r_i, c_i)_i\ ) = 1 : \begin{smallmatrix} k \leftarrow \mathcal{K}_\lambda, \\ r_1, \cdots, r_\ell \leftarrow \mathcal{R}_\lambda \\ c_i \leftarrow \mathcal{O}_\lambda \end{smallmatrix}\right] \right\| \leq \mathsf{negl}(\lambda) \ .$$

Definition 10. *We say that PRF is computable by a circuit class $\mathcal{C} = (\mathcal{C}_\lambda)_\lambda$* with pre-computation *if there exists a uniform family of circuits $C = (C_\lambda)_\lambda, C_\lambda \in \mathcal{C}_\lambda$ as well as a polynomial-time procedure H (not necessarily in $\mathcal{C}$) such that $\mathsf{PRF}(k, r) = C_\lambda(k, H(r))$ for each $k \in \mathcal{K}_\lambda$.*

We will always set $\mathcal{C}$ to be the set $\mathsf{NC}^1(\mathbb{F})$ for a field family $\mathbb{F}$.

2.3 Attribute-Based Encryption (ABE)

Here, we define attribute-based encryption. There are two formulations of ABE in the literature, key-policy ABE where keys are associated with functions and ciphertexts with attributes, and ciphertext-policy ABE where ciphertexts are associated with functions and keys with attributes. In this work, we use a version implied by both.

Let $R = \{R_\kappa\}_\kappa$ where $R_\kappa : \{0,1\}^{m(\kappa)} \times \{0,1\}^{n(\kappa)} \to \{0,1\}$ be a family of binary relations, and $\mathcal{K} = (\mathcal{K}_\lambda)_\lambda$ a family of sets. Then an attribute-based encryption (ABE) scheme for R and $\mathcal{K}$ is a tuple $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:

$$\begin{aligned} \mathsf{Setup}(1^\lambda, 1^\kappa) &\to (\mathsf{mpk}, \mathsf{msk}) \\ \mathsf{KeyGen}(\mathsf{msk}, x) &\to \mathsf{sk}_x \\ \mathsf{Enc}(\mathsf{mpk}, y) &\to (c, k) \\ \mathsf{Dec}(\mathsf{sk}_x, c) &\to k \end{aligned} \qquad \text{where} \qquad \begin{aligned} \mathsf{mpk} &: \text{ master public key} \\ \mathsf{msk} &: \text{ master secret key} \\ x \in \{0,1\}^{m(\kappa)} &: \text{ secret key attribute} \\ y \in \{0,1\}^{n(\kappa)} &: \text{ ciphertext attribute} \\ \mathsf{sk}_x &: \text{ secret key for } x \\ c &: \text{ ciphertext} \\ k \in \mathcal{K}_\lambda &: \text{ key encapsulated in } c \end{aligned}$$

Definition 11 (Correct ABE)**.** *An ABE scheme Π is* correct *if, for every polynomial $\kappa(\lambda)$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that for all $\lambda \in \mathbb{N}$ and all $x, y \in \{0,1\}^{m(\kappa(\lambda))} \times \{0,1\}^{n(\kappa(\lambda))}$ such that $R_{\kappa(\lambda)}(x, y) = 1$,*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_x, c) \neq k : \begin{smallmatrix} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)}) \\ \mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x) \\ (c, k) \leftarrow \mathsf{Enc}(\mathsf{mpk}, y) \end{smallmatrix}\right] \leq \mathsf{negl}(\lambda) \ .$$

Definition 12 (Message Hiding)**.** *An ABE scheme Π is* selectively message hiding secure *(SEL-M secure) if, for every polynomial $\kappa(\lambda)$ and every PPT stateful adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that for all $\lambda \in \mathbb{N}$, the advantage of $\mathcal{A}$ is at most $\mathsf{negl}(\lambda)$ in the following:*

- *Run $\mathcal{A}(1^\lambda)$ to get a ciphertext attribute $y^* \in \{0,1\}^{n(\kappa(\lambda))}$.*
- *Run $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)})$ and send mpk to $\mathcal{A}$.*
- *$\mathcal{A}$ makes arbitrarily-many adaptive queries on secret key attributes $x \in \{0,1\}^{m(\kappa(\lambda))}$ such that $R(x, y^*) = 0$, and receives $\mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x)$.*

- *At some point, $\mathcal{A}$ asks for the ciphertext challenge ciphertext. In response, it receives (c^*, k_b^*) where $(c^*, k_0^*) \leftarrow \mathsf{Enc}(\mathsf{mpk}, y^*)$, $k_1^* \leftarrow \mathcal{K}_\lambda$, and $b \leftarrow \{0,1\}$.*
- *$\mathcal{A}$ can continue making secret key attribute queries on x such that $R(x, y^*) = 0$.*
- *Finally, $\mathcal{A}$ outputs a bit b'. The advantage of $\mathcal{A}$ is $\|2 \times \Pr[b' = b] - 1\|$.*

2.4 Secret Key Functional Encryption (FE) and Mixed FE (MFE)

Let $R = \{R_\kappa\}_\kappa$ where $R_\kappa : \{0,1\}^{m(\kappa)} \times \{0,1\}^{n(\kappa)} \to \mathcal{Z}_\kappa$ be a family of relations. Then a (secret key) functional encryption (FE) scheme for R is a tuple $\Pi_{\mathsf{FE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{EncSK}, \mathsf{Dec})$ and a mixed FE (MFE) is a tuple $\Pi_{\mathsf{MFE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{EncSK}, \mathsf{EncPK}, \mathsf{Dec})$ with the following syntax:

$$\begin{array}{rcl} \mathsf{Setup}(1^\lambda, 1^\kappa) \to & (\mathsf{mpk}, \mathsf{msk}) & \\ \mathsf{KeyGen}(\mathsf{msk}, x) \to & \mathsf{sk}_x & \\ \mathsf{EncSK}(\mathsf{msk}, y) \to & c & \text{where} \\ \mathsf{EncPK}(\mathsf{mpk}) \to & c & \\ \mathsf{Dec}(\mathsf{sk}_x, c) \to & b & \end{array} \qquad \begin{array}{rl} \mathsf{mpk}: & \text{master public key} \\ \mathsf{msk}: & \text{master secret key} \\ x \in \{0,1\}^{m(\kappa)}: & \text{secret key attribute} \\ y \in \{0,1\}^{n(\kappa)}: & \text{ciphertext attribute} \\ \mathsf{sk}_x: & \text{secret key for } x \\ c: & \text{ciphertext} \\ b \in \{0,1\}: & \text{output bit} \end{array}$$

An MFE additionally requires $\mathcal{Z}_\kappa = \{0,1\}$. We will only consider secret key FE and will henceforth drop the modifier "secret key". Note that mpk only serves a role in MFE, but is unused in FE.

Definition 13 (Correct FE)**.** *An FE scheme Π is* correct *if, for every polynomial $\kappa(\lambda)$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that for all $\lambda \in \mathbb{N}$ and all $x, y \in \{0,1\}^{m(\kappa(\lambda))} \times \{0,1\}^{n(\kappa(\lambda))}$,*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_x, c) \neq R_\kappa(x, y) : \begin{smallmatrix}(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)}) \\ \mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x) \\ c \leftarrow \mathsf{EncSK}(\mathsf{msk}, y)\end{smallmatrix}\right] \leq \mathsf{negl}(\lambda) \ .$$

Definition 14 (Correct MFE)**.** *An MFE scheme Π is* correct *if, (1) it is correct as an FE scheme, and (2) for every polynomial $\kappa(\lambda)$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that for all $\lambda \in \mathbb{N}$ and all $x, y \in \{0,1\}^{m(\kappa(\lambda))} \times \{0,1\}^{n(\kappa(\lambda))}$,*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_x, c) \neq 1 : \begin{smallmatrix}(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)}) \\ \mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x) \\ c \leftarrow \mathsf{EncPK}(\mathsf{mpk})\end{smallmatrix}\right] \leq \mathsf{negl}(\lambda) \ .$$

That is, ciphertexts generated from EncPK behave as if they poses attribute y such that $R(x, y) = 1$.

Ciphertext Attribute Security. Security for FE (and one of the needed properties for MFE) roughly requires that nothing about the ciphertext attribute y is revealed, except for the values $R(x, y)$ for x among secret key attributes seen by the adversary.

Definition 15 (q-bounded ciphertext attribute hiding)**.** *Let $q(\lambda)$ be a function. An FE/MFE Π is q-bounded selective ciphertext attribute secure (q-SEL-CTXT secure) if, for every polynomial $\kappa(\lambda)$ and every stateful PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that, for every $\lambda \in \mathbb{N}$, the advantage of $\mathcal{A}$ is at most $\mathsf{negl}(\lambda)$ in the following experiment:*

- *Run $\mathcal{A}(1^\lambda)$ to get two lists of ciphertext attributes $\mathbf{y}_0^*, \mathbf{y}_1^* \in (\{0,1\}^{n(\kappa(\lambda))})^{q(\lambda)}$.*

- *Now run* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)})$ *and send* mpk *to* $\mathcal{A}$.
- $\mathcal{A}$ *makes arbitrary-many queries on key attributes* x *such that* $R(x, y^*_{0,i}) = R(x, y^*_{1,i})$ *for all* $i \in [q(\lambda)]$*, where* $y^*_{b,i} \in \{0,1\}^{n(\kappa(\lambda))}$ *is the* i*th component of* $\mathbf{y}^*_b$*; it receives* $\mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x)$.
- *Choose a random bit* $b \in \{0,1\}$. $\mathcal{A}$ *gets ciphertexts* $\{c^*_{b,i}\}_{i \in [q(\lambda)-1]}$ *for* $c^*_{b,i} \leftarrow \mathsf{EncSK}(\mathsf{msk}, y^*_{b,i})$.
- $\mathcal{A}$ *outputs a bit* b'*. The advantage of* $\mathcal{A}$ *is* $\|2 \times \Pr[b' = b] - 1\|$

Remark 16. *Our use of* q *here is off by 1 from [GKW18]. [GKW18] consider a definition where there is one "challenge" ciphertext that must be distinguished, plus the adversary gets to see* q *additional ciphertexts for attributes of its choice. They use* q *to count the number of* additional *ciphertexts, meaning the total number of ciphertexts seen by the adversary is* $q + 1$*. We instead use* q *as the total number of ciphertexts seen in order to be consistent with usage in the FE literature.*

Accept security. For MFE systems, we also require *accept* security, which captures that public ciphertexts produced by EncPK should be computationally indistinguishable from ciphertexts with any attribute y satisfying $R(x, y) = 1$ for secret keys sk_x that the adversary has seen.

Definition 17 (q-bounded accept security)**.** *Let* $q(\lambda)$ *be a function. An MFE* Π *is* q*-bounded selective accept secure (*q*-SEL-ACC secure) if, for every polynomial* $\kappa(\lambda)$ *and for every stateful PPT adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that, for every* $\lambda \in \mathbb{N}$*, the advantage of* $\mathcal{A}$ *is at most* $\mathsf{negl}(\lambda)$ *in the following experiment:*

- *Run* $\mathcal{A}(1^\lambda)$ *to get a list of ciphertext attributes* $\mathbf{y}^* \in (\{0,1\}^{n(\kappa(\lambda))})^{q(\lambda)-1}$.
- *Now run* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)})$ *and send* mpk *to* $\mathcal{A}$.
- $\mathcal{A}$ *now makes queries on key attributes* x *with the guarantee that* $R(x, y^*_i) = 1$ *for each* $i \in [q(\lambda)]$*, where* y^*_i *is the* i*th component of* $\mathbf{y}^*$*; it receives in response* $\mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x)$.
- *Choose a random bit* $b \in \{0,1\}$. $\mathcal{A}$ *now receives ciphertexts* $\{c^*_{b,i}\}_{i \in [q(\lambda)-1]}$ *where* $c^*_{0,i} \leftarrow \mathsf{EncSK}(\mathsf{msk}, y^*_i)$*, and* $c^*_1 \leftarrow \mathsf{EncPK}(\mathsf{mpk})$.
- $\mathcal{A}$ *outputs a bit* b'*. The advantage of* $\mathcal{A}$ *is* $\|2 \times \Pr[b' = b] - 1\|$.

2.5 Message Carrying MFE (mcMFE)

Here, we define Message Carrying MFE (mcMFE). This includes private linear broadcast encryption (PLBE) as a special case, versions of which imply traitor tracing [BSW06, GKW18]. By considering a more general version, we also include concepts such as *embedded identity* PLBE, which is used to build embedded identity traitor tracing [GKW19].

Let $R = \{R_\kappa\}_\kappa$ where $R_\kappa : \{0,1\}^{m(\kappa)} \times \{0,1\}^{n(\kappa)} \to \mathcal{Z}$ be a family of relations. Then a mcMFE scheme for R and key space $\mathcal{K} = (\mathcal{K}_\lambda)_\lambda$ is a tuple $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{EncSK}, \mathsf{EncPK}, \mathsf{Dec})$ that is a mixed FE scheme with an added KEM functionality, meaning encryption produces additionally a key k. Decryption reveals k if $R(x, y) = 1$; otherwise k is hidden. For publicly generated attribute-less ciphertexts, decryption under *any* secret key reveals k. It has the following syntax:

$$\begin{array}{rl} \mathsf{Setup}(1^\lambda, 1^\kappa) \to & (\mathsf{mpk}, \mathsf{msk}) \\ \mathsf{KeyGen}(\mathsf{msk}, x) \to & \mathsf{sk}_x \\ \mathsf{EncSK}(\mathsf{msk}, y) \to & (c, k) \\ \mathsf{EncPK}(\mathsf{mpk}) \to & (c, k) \\ \mathsf{Dec}(\mathsf{sk}_x, c) \to & k \end{array} \quad \text{where} \quad \begin{array}{rl} \mathsf{mpk}: & \text{master public key} \\ \mathsf{msk}: & \text{master secret key} \\ x \in \{0,1\}^{m(\kappa)}: & \text{secret key attribute} \\ y \in \{0,1\}^{n(\kappa)}: & \text{ciphertext attribute} \\ \mathsf{sk}_x: & \text{secret key for } x \\ c: & \text{ciphertext} \\ k \in \mathcal{K}_\lambda: & \text{key encapsulated in } c \end{array}$$

Definition 18 (Correct mcMFE)**.** *A mcMFE scheme* Π *is* correct *if, for every polynomial* $\kappa(\lambda)$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that for all* $\lambda \in \mathbb{N}$ *and all* $x, y \in \{0,1\}^{m(\kappa(\lambda))} \times \{0,1\}^{n(\kappa(\lambda))}$*,*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_x, c) \neq \begin{cases} k & \text{if } R_\kappa(x,y) = 1 \\ \bot & \text{otherwise} \end{cases} : \begin{array}{c} (\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)}) \\ \mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x) \\ c \leftarrow \mathsf{EncSK}(\mathsf{msk}, y) \end{array}\right] \leq \mathsf{negl}(\lambda) \text{ , and}$$

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_x, c) \neq k : \begin{array}{c} (\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)}) \\ \mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x) \\ (c,k) \leftarrow \mathsf{EncPK}(\mathsf{mpk}) \end{array}\right] \leq \mathsf{negl}(\lambda) \ .$$

Definition 19 (q-bounded Message Hiding)**.** *Let* $q(\lambda)$ *be a function. An mcMFE* Π *is* q*-bounded selective message hiding secure (*q*-SEL-M secure) if, for every polynomial* $\kappa(\lambda)$ *and every stateful PPT adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that, for every* λ*, the advantage of* $\mathcal{A}$ *is at most* $\mathsf{negl}(\lambda)$ *in the following experiment:*

- *Run* $\mathcal{A}(1^\lambda)$ *to get ciphertext attribute* $y^* \in \{0,1\}^{n(\kappa(\lambda))}$ *as well as ciphertext attribute list* $Y \in (\{0,1\}^{n(\kappa(\lambda))})^{q(\lambda)-1}$*.*
- *Now run* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)})$ *and send* mpk *to* $\mathcal{A}$*.*
- $\mathcal{A}$ *can now make arbitrary queries on key attributes* x *with the guarantee that* $R(x, y^*) = 0$*; it receives in response* $\mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x)$*.*
- *Choose a random bit* $b \in \{0,1\}$*.* $\mathcal{A}$ *now receives ciphertexts* $\{(c_i, k_i)\}_{i \in [q(\lambda)-1]}$ *and* (c^*, k_b^*) *where* $(c_i, k_i) \leftarrow \mathsf{EncSK}(\mathsf{msk}, Y_i)$*,* $(c^*, k_0^*) \leftarrow \mathsf{EncSK}(\mathsf{msk}, y^*)$*, and* $k_1^* \leftarrow \mathcal{K}_\lambda$*.*
- $\mathcal{A}$ *outputs a bit* b'*. The advantage of* $\mathcal{A}$ *is* $\|2 \times \Pr[b' = b] - 1\|$*.*

It is also straightforward to define notions of q-SEL-CTXT and q-SEL-ACC for mcMFE, analogous to Definitions 15 and 17.

Decoder-based security. An important tool for realizing traitor tracing *decoder-based security*. This is because traitor tracing adversaries (such as Definition 8) produce decoders, and the natural reductions to the underlying cryptographic building blocks therefore have adversaries produce decoders which then break some security property. It is therefore useful for many of our definitions to consider security against such decoder-producing adversaries.

Definition 20 (Selective *decoder-based* ciphertext attribute security)**.** *An mcMFE* Π *is selective decoder-based ciphertext attribute secure (SEL-DEC-CTXT secure) if, for every polynomial* $\kappa(\lambda)$*, every stateful PPT adversary* $\mathcal{A}$*, and every inverse-polynomial* $\epsilon(\lambda)$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that, for every* $\lambda \in \mathbb{N}$*, the advantage of* $\mathcal{A}$ *is at most* $\mathsf{negl}(\lambda)$ *in the following experiment:*

- *Run* $\mathcal{A}(1^\lambda)$ *to get ciphertext attributes* $y_0^*, y_1^* \in \{0,1\}^{n(\kappa(\lambda))}$*.*

- *Now run* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)})$ *and send* mpk *to* $\mathcal{A}$.
- $\mathcal{A}$ *can now make arbitrary queries on key attributes* x *with the guarantee that* $R(x, y_1^*) = R(x, y_2^*)$*; it receives in response* $\mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x)$.
- $\mathcal{A}$ *outputs a decoder* D.
- *The advantage of* $\mathcal{A}$ *is* $\Pr\left[\mathsf{Good}_{\epsilon(\lambda)}(\mathsf{D}, \mathsf{msk}, y_0^*, y_1^*)\right]$ *where* $\mathsf{Good}_{\epsilon(\lambda)}(\mathsf{D}, \mathsf{msk}, y_0^*, y_1^*)$ *is the event*

$$\| \Pr[\mathsf{D}(\mathsf{EncSK}(\mathsf{msk}, y_0^*)) = 1] - \Pr[\mathsf{D}(\mathsf{EncSK}(\mathsf{msk}, y_1^*)) = 1] \| \geq \epsilon(\lambda) \ .$$

Likewise, it is straightforward to define decoder-based notions SEL-DEC-M and SEL-DEC-ACC.

Private Linear Broadcast Encryption (PLBE). PLBE is mcMFE for the relation $R^{\mathsf{PLBE}} = (R_\kappa^{\mathsf{PLBE}})_\kappa$ where $R_\kappa^{\mathsf{PLBE}} : [2^\kappa] \times [0, 2^\kappa] \to \{0, 1\}$ is defined as $R^{\mathsf{PLBE}}(\mathsf{id}, t) = \mathbb{1}(\mathsf{id} \leq t)$. Note that we normally set $\kappa = \lambda$, meaning all the parameters of the PLBE will be fixed polynomials in λ.

Embedded Identity PLBE (EIPLBE). EIPLBE is mcMFE for the two-parameter relation $R^{\mathsf{PLBE}} = (R_{\kappa,\ell}^{\mathsf{EIPLBE}})_{\kappa,\ell}$ with two indices, where secret key attributes are pairs $(j, \mathsf{id}) \in [2^\kappa] \times \{0, 1\}^\ell$ and ciphertexts attributes are tuples $(t, i) \in [0, 2^\kappa] \times [0, \ell]$. $R_{\kappa,\ell}^{\mathsf{EIPLBE}}$ is defined as

$$R_{\kappa,\ell}^{\mathsf{EIPLBE}}((j, \mathsf{id}), (t, i)) = \begin{cases} \mathbb{1}(\ j \leq t\ ) & \text{if } i = 0 \\ \mathbb{1}(\ j < t \vee (j, \mathsf{id}_i) = (t, 1)\ ) & \text{if } i > 0\ . \end{cases}$$

2.6 Existing Results

Eliminating Riskiness and Thresholds in Tracing. The usual goal has been to construct optimal traceable schemes *without* any riskiness or threshold. However, the following theorem of [Zha20] shows that it suffices to build a (ϵ, δ)-threshold risky scheme for any constant ϵ, δ:

Theorem 21 (Special case of [Zha20])**.** *Fix constants* $\epsilon, \delta \in (0, 1)$*. If there exists an optimal* (ϵ, δ)*-threshold risky traceable scheme, then there exists an optimal traceable scheme.*

From MFE and ABE to mcMFE. The following is easily adapted from [GKW18].

Theorem 22 ([GKW18])**.** *Let* Π_{MFE} *be an MFE scheme for relation* R*. Suppose there exists an ABE scheme* Π_{ABE} *that is SEL-M secure and whose relation is the decryption function for* Π_{MFE}*. Then there exists a mcMFE scheme* Π_{mcMFE} *for* R *such that:*

- Π_{mcMFE} *is* q*-SEL-M secure for any polynomial* q*.*
- *If* Π_{MFE} *is* q*-SEL-CTXT (resp.* q*-SEL-ACC) secure, then so is* Π_{mcMFE}*.*

Obtaining Decoder-based mcMFE security. The following is also from [GKW18].

Theorem 23 ([GKW18])**.** *For* $X \in \{M, CTXT, ACC\}$*, if an mcMFE protocol* Π *is* 2*-SEL-*X *secure, it is also SEL-DEC-*X *secure.*

Traitor Tracing from mcMFE. The following are proved in [GKW18] and [GKW19] respectively.

Theorem 24 ([GKW18])**.** *If there exists a PLBE scheme that is SEL-DEC-M, SEL-DEC-CTXT, and SEL-DEC-ACC secure, then there exists a traceable index-only traitor tracing scheme.*

Theorem 25 ([GKW19])**.** *If there exists an a EIPLBE scheme that is SEL-DEC-M, SEL-DEC-CTXT, and SEL-DEC-ACC secure, then there exists a traceable index-based embedded-identity traitor tracing scheme.*

We also have the following theorem from [GKW19], which shows how to generically turn an index-based embedded-identity scheme into a full embedded-identity scheme:

Theorem 26 ([GKW19])**.** *If there exists a traceable index-based embedded-identity traitor tracing scheme, then there exists a traceable (non-index-based) embedded-identity traitor tracing scheme.*

3 MFE From Weak PRFs

Our main theorem of this section is the following:

Theorem 27. *Let $R = (R_\kappa)_\kappa \in \mathsf{NC}^1(\mathbb{F})$ for a family of fields $\mathbb{F} = (\mathbb{F}_\kappa)_\kappa$ where $\mathbb{F}$ is super-polynomially large in κ. Assume there exists a weak PRF PRF computable in $\mathsf{NC}^1(\mathbb{F})$ using pre-computation. Then there is a MFE scheme for R that is simultaneously 2-SEL-CTXT secure and 1-SEL-ACC secure, where decryption function is in $\mathsf{NC}^1_{\neq 0}(\mathbb{F})$.*

By combining with Theorem 22 due to [GKW18], we immediately obtain:

Corollary 28. *Make the same assumptions as Theorem 27 plus additionally assume the existence of SEL-M secure attribute-based encryption for $\mathsf{NC}^1_{\neq 0}(\mathbb{F})$. Then there exists a mcMFE scheme for R that is simultaneously 2-SEL-CTXT secure, 2-SEL-M secure, and 1-SEL-ACC secure.*

Note that this is not enough to obtain traitor tracing through existing frameworks. This is because applying Theorem 24 from [GKW18] or Theorem 25 from [GKW19] requires decoder-based security for the underlying (message-carrying) MFE. In our case, we can get decoder-based Ciphertext Attribute security by applying Theorem 23 since we have security for two ciphertexts. However, we only obtain Accept security for a single ciphertext, which is insufficient to get decoder-based Accept security. However, we will later show (Section 4) how to turn a 1-SEL-ACC secure scheme into a scheme with a weak form of decoder security, which we show suffices for traitor tracing.

Our construction here will follow the techniques of [SS10, GVW12, KMUW18], but we will take care to instantiate the techniques in a way that enables pseudorandomness of the ciphertexts.

3.1 Building Block: Low-Depth CPA-secure Encryption

Definition 29. *A CPA-secure symmetric encryption scheme is a pair $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ of PPT algorithms and associated message and ciphertext spaces $\mathcal{M} = (\mathcal{M}_\lambda)_\lambda$ and $\mathcal{C} = (\mathcal{C}_\lambda)_\lambda$ with the syntax:*

$$\begin{array}{rl} \mathsf{Enc}(k, m) \to & c \\ \mathsf{Dec}(k, c) \to & m \end{array} \quad \textit{where} \quad \begin{array}{rl} k \in \{0,1\}^\lambda : & \textit{secret key} \\ m \in \mathcal{M}_\lambda : & \textit{message} \\ c : & \textit{ciphertext} \end{array}$$

- **Correctness:** *There exists a negligible function* $\mathsf{negl}(\lambda)$ *such that, for every* $\lambda > 0$ *and every* $m \in \mathcal{M}_\lambda$, $\Pr[\mathsf{Dec}(k, \mathsf{Enc}(k, m) \neq m : k \leftarrow \{0,1\}^\lambda] \leq \mathsf{negl}(\lambda)$.
- **CPA Security:** *For every stateful adversary* $\mathcal{A}$*, there exists a negligible function* negl *such that the advantage of* $\mathcal{A}$ *is at most* $\mathsf{negl}(\lambda)$ *in the following experiment:*
  - *Choose a random bit* b *and random key* $k \leftarrow \{0,1\}^\lambda$.
  - *Run* $\mathcal{A}(1^\lambda)$. $\mathcal{A}$ *can now make arbitrarily many queries on pairs* $m_0, m_1 \in \mathcal{M}_\lambda$. *In response, it receives* $\mathsf{Enc}(k, m_b)$.
  - *Finally,* $\mathcal{A}$ *produces a bit* b'. *The advantage of* $\mathcal{A}$ *is* $2\Pr[b' = b] - 1$.

In this work, we will additionally use the following statistical property in our encryption scheme:

Definition 30. *We say an symmetric key encryption scheme* Π *is* key non-committing under random messages *if there exists a negligible function* negl *such that, for every* $\lambda \in \mathbb{N}$*, the distributions* $(k, k', \mathsf{Enc}(k, m))$ *and* $(k, k', \mathsf{Enc}(k', m))$ *are* $\mathsf{negl}(\lambda)$ *close, where* $k, k' \leftarrow \{0,1\}^\lambda$ *and* $m \leftarrow \mathcal{M}_\lambda$.

Lemma 31. *Assume the existence of a weak PRF* PRF *that is computable in* $\mathsf{NC}^1(\mathbb{F})$ *using pre-computation. Then there exists a CPA-secure symmetric encryption scheme* Π *where decryption is computable in* $\mathsf{NC}^1(\mathbb{F})$. *Moreover,* Π *is key non-committing under random ciphertexts.*

Proof. Let $\mathsf{Enc}(k, m) = (h_r, \mathsf{PRF}(k, r) + m)$ for a random choice of r. Dec then computes $\mathsf{PRF}(k, r)$ from k, h_r in $\mathsf{NC}^1(\mathbb{F})$, and then uses it to un-mask m. CPA-security readily follows from the security of the weak PRF. That decryption is computable in $\mathsf{NC}^1(\mathbb{F})$ follows immediately from the ability to compute $\mathsf{PRF}(k, r)$ in $\mathsf{NC}^1(\mathbb{F})$. For the key non-committing property, on a random message, $\mathsf{Enc}(k, m)$ is just h_r together with a random string. h_r is independent of the key k. Therefore, $\mathsf{Enc}(k, m)$ for a random message m is independent of the key. Note that the message-space can be made arbitrarily long with the same key: divide the message into blocks and encrypt each block separately. This preserves both CPA security and key-non-committing under random messages. □

3.2 Building Block: *Random* Randomized Encodings

Definition 32. *Let* $R = (R_\kappa)_\kappa$ *be a family of two-input functions* $R : \{0,1\}^{m(\kappa)} \times \{0,1\}^{n(\kappa)} \to \mathcal{Z}_\kappa$ *for some sets* Z_κ. *A* random randomized encoding *(RRE) for* R *is a pair of PPT algorithms* $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ *where:*

- $\mathsf{Enc}(x) \to (L_{i,b})_{i \in [n(\kappa)], b \in \{0,1\}}$.
- *For a string* $y \in \{0,1\}^{n(\kappa)}$, $\Pr[\mathsf{Dec}(\ (L_{i,y_i})_{i \in [n(\kappa)]}) = R(x, y)] = 1$.
- *There exists a negligible function* $\mathsf{negl}(\kappa)$ *such that, for any* x, y*, the distribution of the labels* $(L_{i,y_i})_{i \in [n(\kappa)]} \in \{0,1\}^{p(\kappa) \times n(\kappa)}$ *is* $\mathsf{negl}(\kappa)$*-close to a uniform string* $L \in \{0,1\}^{p(\kappa) \times n(\kappa)}$ *conditioned on* $\mathsf{Dec}(L) = R(x, y)$. *Moreover, for a uniform random string* L, $\mathsf{Dec}(L)$ *is* $\mathsf{negl}(\kappa)$*-close to uniform in* $\mathbb{Z}_\kappa$.

Note that plain randomized encodings only require that the distribution of labels for a pair (x, y) is essentially independent of x, y, only depending on $R(x, y)$. The labels could be arbitrarily structured, however. RREs strengthen this to require the labels to be as random as possible, subject to the correctness of the scheme. Nevertheless, we show the following:

Lemma 33. *For any field $\mathbb{F} = (\mathbb{F}_\kappa)_\kappa$ of size super-polynomial in κ and for any $R \in \mathsf{NC}^1(\mathbb{F})$, there is an RRE where $\mathsf{Dec} \in \mathsf{NC}^1(\mathbb{F})$.*

Proof. In [BOC92] it is shown how to turn any depth-d binary-input arithmetic formula $f : \{0,1\}^n \to \mathbb{F}$ into a *matrix* branching program, comprising 2×4^d invertible matrices in $\mathbb{F}^{3\times 3}$. That is, a collection of matrices $(\mathbf{M}_{i,b})_{i\in[\ell],b\in\{0,1\}}$ where $\ell = 4^d$ and an input function $\mathsf{inp} : [\ell] \to [n]$ such that (1) $\mathbf{M}_{j,b}$ is an invertible matrix in $\mathbb{F}^{3\times 3}$ and (2) we have

$$\prod_{j=1}^{\ell} \mathbf{M}_{i,x_{\mathsf{inp}(j)}} = \begin{pmatrix} 1 & f(x) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} .$$

where the product is carried out with $j = 1$ on the left and $j = \ell$ on the right. By padding with identity matrices, we can take $\ell = n \times 2^d$ and $\mathsf{inp}(j) = j \bmod n$, independent of f (except through its depth). Here, it is understood that mod outputs elements in $[n]$. Moreover, by left- and right-multiplying by "booked" matrices $\mathbf{s} = (1\ \ 0\ \ 0)$ and $\mathbf{t} = (0\ \ 1\ \ 0)^T$, we have that

$$\mathbf{s} \cdot \left(\prod_{j=1}^{\ell} \mathbf{M}_{j,x_{j \bmod n}} \right) \cdot \mathbf{t} = f(x) \ .$$

We now apply Kilian re-randomization [Kil88]. We choose random invertible matrices $\mathbf{R}_1, \cdots, \mathbf{R}_{\ell-1} \in \mathbb{F}^{3\times 3}$, and define:

$$\widehat{\mathbf{M}}_{j,b} = \mathbf{R}_{j-1} \cdot \mathbf{M}_{j,b} \cdot \mathbf{R}_j^{-1} \text{ for } j \in [2, t-1] \quad , \quad \widehat{\mathbf{M}}_{1,b} = \mathbf{s} \cdot \mathbf{M}_{1,b} \cdot \mathbf{R}_1^{-1} \quad , \quad \widehat{\mathbf{M}}_{t,b} = \mathbf{R}_{t-1} \cdot \mathbf{M}_{t,b} \cdot \mathbf{t}$$

Such re-randomization maintains that

$$\prod_{j=1}^{\ell} \widehat{\mathbf{M}}_{j,x_{j \bmod n}} = f(x) \ .$$

The algorithm $\mathsf{Enc}(x)$ therefore lets $f_x(y) = R(x,y)$ with x hard-coded, and samples the matrices $\widehat{\mathbf{M}}_{j,b}$ for f_x. Then it sets $L_{i,b} = (\widehat{M}_{j,b})_{j:j \bmod n=i}$. Dec simply extracts all the matrices, orders them appropriately, and multiplies them. Since Dec is just an iterated matrix multiplication with constant-sized matrices, it can be computed in $\mathsf{NC}^1(\mathbb{F})$.

Observe that $(L_{i,y_i})_{i\in[n]} \equiv (\widehat{M}_{j,y_{j \bmod n}})_{j\in[\ell]}$, which, by the re-randomization is seen to be distributed uniformly among lists of matrices conditioned on (1) the matrices being invertible, and (2) their product being $R(x,y)$. Since the field $\mathbb{F}$ has super-polynomial size, random matrices are invertible with all but negligible probability. As such, we can drop condition (1) and only negligibly change the distribution. The result is that $(L_{i,y_i})_{i\in[n]}$ are statistically close to random conditioned on $\mathsf{Dec}((L_{i,y_i})_{i\in[n]}) = R(x,y)$. It is also straightforward that the product of random matrices in $\mathbb{F}$ is statistically close to uniform, as long as $\mathbb{F}$ is super-polynomial. □

3.3 An FE for One Ciphertext

We first focus on the easier case where we only ask for 1-SEL-CTXT security and (a version of) 1-SEL-ACC security. In Section 3.4 we will upgrade the construction to 2-SEL-CTXT security while preserving 1-SEL-ACC security. Our 1-ciphertext secure scheme here will closely follow that

of [SS10], and in Section 3.4 we will upgrade to a 2-ciphertext secure scheme following the techniques in [GVW12, KMUW18]. In both cases, we will point out additional features that allow us to ultimately obtain a MFE scheme with useful Accept security.

Let $R = (R_\kappa)_\kappa \in \mathsf{NC}^1(\mathbb{F})$. Let $\Pi_{\mathsf{sk}} = (\mathsf{Enc}_{\mathsf{sk}}, \mathsf{Dec}_{\mathsf{sk}})$ be a CPA secure symmetric key encryption scheme, and let $\Pi_{\mathsf{RRE}} = (\mathsf{Enc}_{\mathsf{RRE}}, \mathsf{Dec}_{\mathsf{RRE}})$ be a random randomized encoding.

Construction 34. *Let* $\Pi_{\mathsf{FE}} = (\mathsf{Setup}_{\mathsf{FE}}, \mathsf{KeyGen}_{\mathsf{FE}}, \mathsf{EncSK}_{\mathsf{FE}}, \mathsf{Dec}_{\mathsf{FE}})$ *be defined as:*

- $\mathsf{Setup}_{\mathsf{FE}}(1^\lambda, 1^\kappa)$*: sample symmetric encryption keys* $k_{i,b} \leftarrow \{0,1\}^\lambda$ *for* $i \in [n(\kappa)], b \in \{0,1\}$*. Sample* $u \in \{0,1\}^{n(\kappa)}$*. Output* $\mathsf{msk} = (u, (k_{i,b})_{i\in[n(\kappa)],b\in\{0,1\}})$*.* mpk *is empty.*
- $\mathsf{KeyGen}_{\mathsf{FE}}(\mathsf{msk}, x)$*: Run* $\mathsf{Enc}_{\mathsf{RRE}}(x) \to (L_{i,b})_{i\in[n(\kappa)],b\in\{0,1\}}$*. Now set* $s_{i,b} = \mathsf{Enc}_{\mathsf{sk}}(k_{i,b}, L_{i,b\oplus u_i})$*. Output* $\mathsf{sk}_x = (s_{i,b})_{i\in[n(\kappa)],b\in\{0,1\}}$*.*
- $\mathsf{EncSK}_{\mathsf{FE}}(\mathsf{msk}, y)$*: Output* $c = (y', (c_i)_{i\in[n(\kappa)]})$ *where* $y' = y \oplus u$ *and* $c_i = k_{i,y'_i}$*.*
- $\mathsf{Dec}_{\mathsf{FE}}(\mathsf{sk}_x, c)$*: Let* $L'_i = \mathsf{Dec}_{\mathsf{sk}}(c_i, s_{i,y'_i})$*. Output* $\mathsf{Dec}_{\mathsf{RRE}}((L'_i)_{i\in[n(\kappa)]})$*.*

Correctness follows immediately from the correctness of the underlying building blocks. For security, while [SS10] flip the roles of ciphertext and secret key, and also use public key encryption instead of symmetric key encryption, the proof of security nevertheless easy applies to our scheme. In fact, they achieve an even stronger notion of security that hides both the ciphertext attributes *and* key attributes. We define this notion next.

Definition 35 (q-bounded key/ciphertext attribute hiding)**.** *Let* $q(\lambda)$ *be a function. An FE* Π *is* q*-bounded selective key/ciphertext attribute secure (*q*-SEL-KEY-CTXT secure) if, for every polynomial* $\kappa(\lambda)$ *and every stateful PPT adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that, for every* λ*, the advantage of* $\mathcal{A}$ *is at most* $\mathsf{negl}(\lambda)$ *in the following experiment:*

- *Run* $\mathcal{A}(1^\lambda)$ *to get two lists of ciphertext attributes* $\mathbf{y}^*_0, \mathbf{y}^*_1 \in (\{0,1\}^{n(\kappa(\lambda))})^{q(\lambda)}$*.*
- *Choose a random bit* $b \in \{0,1\}$*. Now run* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)})$ *and send* mpk *to* $\mathcal{A}$*.*
- $\mathcal{A}$ *can now make arbitrary queries on pairs of key attributes* x_0, x_1 *with the guarantee that* $R(x_0, y^*_{0,i}) = R(x_1, y^*_{1,i})$ *for all* $i \in [q(\lambda)]$*, where* $y^*_{b,i} \in \{0,1\}^{n(\kappa(\lambda))}$ *is the* i*th component of* $\mathbf{y}^*_b$*; it receives in response* $\mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x_b)$*.*
- $\mathcal{A}$ *now receives ciphertexts* $\{c^*_{b,i}\}_{i\in[q(\lambda)-1]}$ *where* $c^*_{b,i} \leftarrow \mathsf{EncSK}(\mathsf{msk}, y^*_{b,i})$*.*
- $\mathcal{A}$ *outputs a bit* b'*. The advantage of* $\mathcal{A}$ *is* $\|2 \times \Pr[b' = b] - 1\|$*.*

Lemma 36. *If* Π_{sk} *is CPA secure and* Π_{RRE} *is a random randomized encoding, then* Π_{FE} *in Construction 34 is* q*-SEL-KEY-CTXT secure for* $q \in \{0,1\}$*.*

Proof. We give the proof for completeness. We prove the case $q = 1$, the case $q = 0$ follows from a similar but simpler argument. Let $\mathcal{A}$ be a supposed adversary for 1-SEL-KEY-CTXT security. We bound the advantage of $\mathcal{A}$ through a sequence of hybrids, which we show are indistinguishable.

Hybrid 0. In this hybrid, $\mathcal{A}$ plays the 1-SEL-KEY-CTXT game with $b = 0$, meaning the ciphertext c^* it sees is an encryption with ciphertext attribute y^*_0 (since $q = 1$, we drop the index i), and the secret key queries are answered using attribute x_0.

Hybrid 1. This is identical to Hybrid 0, except that in each secret key query, we replace $s_{i,1-y'_i}$ with $s_{i,1-y'_i} \leftarrow \mathsf{Enc}_{\mathsf{sk}}(k_{i,1-y'_i}, 0)$, setting the terms to be encryptions of 0. The indistinguishability between Hybrid 0 and Hybrid 1 follows from the fact that $\mathcal{A}$ never sees $k_{i,1-y'_i}$, meaning we can invoke the CPA security of Π_{sk} to replace any ciphertext encrypted under these keys with encryptions of 0. Notice now that the values $L_{i,(1-y'_i)\oplus u_i} = L_{i,1-y_i}$ are not used to generate the secret key.

Hybrid 2. This is identical to Hybrid 1, except that for secret key query on key attribute x_0, we generate the L_{i,y_i} as uniform random strings, conditioned on $\mathsf{Dec}_{\mathsf{RRE}}((L_{i,y_i})_{i\in[n]}) = R(x_0, y_0^*)$. Indistinguishability from Hybrid 1 follows from the security of Π_{RRE}.

Observe that now the view of the adversary looks like the following. Keys k_i are sampled uniformly (which will ultimately be set to k_{i,y'_i}) as well as keys k'_i (which will ultimately be set to $k_{i,1-y'_i}$). Then for each secret key query on key attributes x_0, x_1, the query is answered as follows: choose random $(L_i)_i$ such that $\mathsf{Dec}_{\mathsf{RRE}}((L_i)_i) = R(x, y_0^*) = R(x_0, y_0^*)$. Let $s_{i,y'_i} = \mathsf{Enc}_{\mathsf{sk}}(k_i, L_i)$ and $s_{i,1-y'_i} = \mathsf{Enc}_{\mathsf{sk}}(k'_i, 0)$. The ciphertext is $c = (y', (k_i)_i)$. Here, $y' = y_0^* \oplus u$.

Hybrid 3. This is identical to Hybrid 2, except that we now generate $(L_i)_i$ as random strings such that $\mathsf{Dec}_{\mathsf{RRE}}((L_i)_i) = R(x_1, y_1^*)$. Moreover, we generate $y' = y_1^* \oplus u$. Observe that $R(x_0, y_0^*) = R(x_1, y_1^*)$, so the distribution on $(L_i)_i$ is unaffected. Moreover, observe that u does not appear anywhere else in the view of $\mathcal{A}$, meaning y' is simply a uniform random bit string in either of Hybrid 3 or Hybrid 2. Thus, Hybrid 2 and Hybrid 3 are perfectly indistinguishable. Now in Hybrid 3, we see that the ciphertext is generated by encrypting the attribute y_1^*, and secret keys are generated according to x_1.

Hybrids 4,5. These are identical to Hybrids 1,0, respectively, except that we change the ciphertext to be generated by encrypting y_1^* and secret keys are generated using attribute x_1. Indistinguishability follows from analogous arguments. The result is that in Hybrid 5, $\mathcal{A}$ plays the 1-SEL-KEY-CTXT game with $b = 1$. By the indistinguishability of each adjacent hybrid, we have that Hybrids 0 and 5 are indistinguishable to $\mathcal{A}$, thus showing that the advantage is negligible. □

Pseudorandom ciphertexts. We prove a useful pseudorandom ciphertext property of Π_{FE}, which will be useful later in Section 3.4.

Definition 37 (Pseudorandom ciphertexts)**.** *An FE Π is has pseudorandom ciphertexts if for every polynomial $\kappa(\lambda)$ and every stateful PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that, for every λ, the advantage of any $\mathcal{A}$ is at most $\mathsf{negl}(\lambda)$ in the following experiment:*

- *Run $\mathcal{A}(1^\lambda)$ to get ciphertext attribute $y^* \in \{0,1\}^{n(\kappa(\lambda))}$.*
- *Now run $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}_{\mathsf{FE}}(1^\lambda, 1^{\kappa(\lambda)})$ and send mpk to $\mathcal{A}$.*
- *$\mathcal{A}$ can now make arbitrary queries on distributions D over key attributes x. The distribution is represented as a sampling circuit, so that the running time of the distribution is no longer than the bit-length of its description. $\mathcal{A}$ guarantees to select D from the set of distributions where $R(x, y^*)$ is uniform. Note that this condition is not efficiently checkable. In response, sample $x \leftarrow D()$, and send $\mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x)$ to $\mathcal{A}$.*

- *Then, choose a random bit $b \in \{0,1\}$. $\mathcal{A}$ receives ciphertexts c_b^* where $c_0^* \leftarrow \mathsf{EncSK}_{\mathsf{FE}}(\mathsf{msk}, y^*)$ and c_1^* is a random bit string in the ciphertext space.*
- *$\mathcal{A}$ outputs a bit b'. The advantage of $\mathcal{A}$ is $\|\Pr[b' = b] - 1/2\|$.*

Lemma 38. *If Π_{sk} is CPA-secure and is key non-committing under random messages (Definition 30), then Π_{FE} in Construction 34 has pseudorandom ciphertexts.*

Proof. Let $\mathcal{A}$ be a supposed adversary for ciphertext pseudorandomness. We prove security through a sequence of hybrids.

Hybrid 0. This is the hybrid where $\mathcal{A}$ receives $c^* \leftarrow \mathsf{EncSK}_{\mathsf{FE}}(\mathsf{msk}, y^*)$. Write $c^* = (y', (c_i)_{i \in [n(\kappa)]})$ where $y' = y \oplus u$ and $c_i = k_{i,y'_i}$.

Hybrid 1. This is identical to Hybrid 0, except that we replace all components $s_{i,1-y'_i}$ in each secret key seen by $\mathcal{A}$ with encryptions of random strings. Since $\mathcal{A}$ never sees $k_{i,1-y'_i}$, indistinguishability of Hybrid 0 and Hybrid 1 follows from the security of Π_{sk}.

Hybrid 2. Here, we additionally replace all components s_{i,y'_i} in each secret key with encryptions of random strings, so all components are random. For indistinguishability from Hybrid 1, consider the secret keys seen in Hybrid 1, which can be simulated only with the labels L_{i,y_i}. By the security of Π_{RRE}, the distribution of the labels $L = (L_{i,y_i})_i$ for each key is statistically close to uniform conditioned on $\mathsf{Dec}_{\mathsf{RRE}}(L) = R(x, y^*)$. But the guarantee of the distribution on x is that $R(x, y^*)$ is uniform. Hence, the distribution of L is simply statistically close to uniform, yielding Hybrid 2.

Hybrid 3. This is identical to Hybrid 2, except we switch to $s_{i,b}$ being encryptions of random strings under keys $k'_{i,b}$ that are chosen independently from $k_{i,b}$. Indistinguishability from Hybrid 2 follows since the $s_{i,b}$ in Hybrid 2 are encryptions of random strings and Π_{sk} is key non-committing, meaning the $s_{i,b}$ in Hybrid 2 are statistically independent of the key used to encrypt them.

At this point, observe that the ciphertext c^* is independent of all secret keys. Moreover, c^* just consists of a list of encryption keys which are all random, together with $y' = u \oplus x$, which is also random since u does not appear anywhere else in Hybrid 3. Hence, in Hybrid 2 it is equivalent to draw c^* uniformly at random.

Hybrids 4,5,6. These are identical to Hybrids 2,1,0, respectively, except that c^* is replaced with random strings. The end result is that in Hybrid 6, $\mathcal{A}$ is playing the ciphertext pseudorandomness game in the case where we generate c^* randomly, thus proving ciphertext pseudorandomness. □

3.4 An FE for Two Ciphertexts

We now achieve ciphertext attribute security for two ciphertexts, following techniques from [GVW12, KMUW18]. This construction will ultimately give the MFE guaranteed by Theorem 27.

Let $R_\kappa \in \mathsf{NC}^1(\mathbb{F})$. Let $R'_\kappa : \mathbb{F}^{m(\kappa)+3} \times \mathbb{F}^{n(\kappa)+1} \to \mathbb{F}$ be the arithmetic formula derived from R_κ as $R'_\kappa((x, r_0, r_1, s), (y, u)) = s \times (1 - R_\kappa(x, y)) + r_0 + ur_1$. The degree of the resulting polynomial is exponential in the depth of the formula for R_κ; since the depth of R_κ is logarithmic, this means the degree of R'_κ is some polynomial $D(\kappa)$. Suppose we instantiate Π_{FE} for the arithmetic formula class $R' = (R'_\kappa)_\kappa$.

Construction 39. *Let* $\Pi_{\mathsf{FE2}} = (\mathsf{Setup}_{\mathsf{FE2}}, \mathsf{KeyGen}_{\mathsf{FE2}}, \mathsf{EncSK}_{\mathsf{FE2}}, \mathsf{Dec}_{\mathsf{FE2}})$ *be defined as:*

- $\mathsf{Setup}_{\mathsf{FE2}}(1^\lambda, 1^\kappa)$*: Let* $U = \lambda D(\kappa) + 1$ *and* $T = U^2$*. Assume* $\mathbb{F}$ *has size at least* U*. For* $t \in [T]$*, run* $\mathsf{msk}_t \leftarrow \mathsf{Setup}_{\mathsf{FE}}(1^\lambda, 1^\kappa)$*. Output* $\mathsf{msk} = (\mathsf{msk}_t)_{t\in[T]}$*.*
- $\mathsf{KeyGen}_{\mathsf{FE2}}(\mathsf{msk}, x)$*: choose two random degree* $U-1$ *polynomials* $r_{x,0}, r_{x,1} : \mathbb{F} \to \mathbb{F}$ *such that* $r_{x_0}(0) = r_{x,1}(0) = 0$*. Choose a random* $s_x \leftarrow \mathbb{F}$*. Then for each* $t \in [T]$*, run* $\mathsf{sk}_{x,t} = \mathsf{KeyGen}_{\mathsf{FE}}(\mathsf{msk}_t, (x, r_{x,0}(t), r_{x,1}(t), s_x))$*. Output* $\mathsf{sk}_x = (\mathsf{sk}_{x,t})_{t\in[T]}$*.*
- $\mathsf{EncSK}_{\mathsf{FE2}}(\mathsf{msk}, y)$*: Choose a random polynomial map* $q : \mathbb{F} \to \mathbb{F}^{m(\lambda)}$ *of degree* λ *such that* $q(0) = y$*. Choose a random set* $S \subseteq [T]$ *of size* U*. Also choose random* $u \in \mathbb{F}$*. For each* $t \in S$*, run* $c_t \leftarrow \mathsf{Enc}_{\mathsf{FE}}(\mathsf{msk}_t, (q(t), u))$*. Write* $S = \{t_1, \cdots, t_U\}$*. Let* $\mathbf{v} \in \mathbb{F}^U$ *be the linear interpolation vector such that* $p(0) = \sum_{i\in[U]} \mathbf{v}_i p(t_i)$ *for all polynomials* p *of degree* $U-1$*. Output* $c = (S, \mathbf{v}, (c_t)_{t\in S})$*.*
- $\mathsf{Dec}_{\mathsf{FE2}}(\mathsf{sk}_x, c)$*: For each* $t \in S$*, let* $p_t = \mathsf{Dec}_{\mathsf{FE}}(\mathsf{sk}_{x,t}, c_t)$*. Then* p_t *is the evaluation of some degree* $U-1$ *polynomial* $p(t)$ *on the* U *points in* S*. Compute* $p(0) = \sum_{i\in S} \mathbf{v}_i p(t_i)$*. Output 1 if* $p(0) = 0$*, and output 0 otherwise.*

Remark 40. *This construction is almost identical to that of [KMUW18], except for the multiplication by* s_x *(whereas they have no* s_x*). Also, the original version of [KMUW18] only used a single polynomial* r_x*. However, their scheme using only a single* r_x *is actually insecure and there is a small bug in their proof (confirmed to us by the authors of [KMUW18]). Using two polynomials as we do is necessary, and we prove it works below in Lemma 41, thus fixing the bug in their proof/construction. In response to our observation, [KMUW18] have also updated their construction analogously.*

We briefly show correctness. By the correctness of Π_{FE}, $p_t = s_x(1 - R(x, q(t))) + r_{x,0}(t) + u r_{x,1}(t)$. Now consider the polynomial $p(t) = s_x(1 - R(x, q(t))) + r_{x,0}(t) + u r_{x,1}(t)$. Since q has degree λ, R has degree $D(\kappa)$ and $r_{x,0}, r_{x,1}$ have degree at most $U - 1 = \lambda D(\kappa)$, $p(t)$ has degree at most $U-1$. Therefore, this $p(t)$ is exactly the polynomial that gets interpolated during decryption. Then since $q(0) = y$ and $r_{x,0}(0) = r_{x,1}(0) = 0$, we must have $p(0) = s_x(1 - R(x, y))$. If $R(x, y) = 0$, then $p(0) = s_x$, which with overwhelming probability is non-zero; likewise if $R(x, y) = 1$, then $p(0) = 0$. Thus, $\mathsf{Dec}_{\mathsf{FE2}}$ outputs (with overwhelming probability) the bit $R(x, y)$. We now explain security:

Lemma 41. *If* $\mathbb{F}$ *is exponentially large and* Π_{FE} *in Construction 34* q*-SEL-KEY-CTXT secure for* $q \in \{0, 1\}$*, then* Π_{FE2} *in Construction 39 is* 2*-SEL-CTXT secure.*

Proof. Consider a supposed 2-SEL-CTXT adversary $\mathcal{A}$ which commits to two pairs of ciphertext attributes $\mathbf{y}_0^*, \mathbf{y}_1^*$, and then only makes key queries on attributes x such that $R_\kappa(x, y_{0,1}^*) = R_\kappa(x, y_{1,1}^*)$ and $R_\kappa(x, y_{0,2}^*) = R_\kappa(x, y_{1,2}^*)$. We prove security though a sequence of hybrids.

Hybrid 0. This is the case where the ciphertexts are encryptions of $y_{0,1}^*$ and $y_{0,2}^*$. Therefore, ciphertext c_β for $\beta \in \{1, 2\}$ is generated by choosing a random map q_β of degree λ such that $q_\beta(0) = y_{0,\beta}^*$, random scalars $u_\beta \in \mathbb{F}$, a random set $S_\beta \subseteq [T]$ of size U, and then setting $c_{\beta,t} \leftarrow \mathsf{Enc}_{\mathsf{FE}}(\mathsf{msk}_t, (q_\beta(t), u_\beta))$ for $t \in S_\beta$. Let c_β is set to $c_\beta = (S_\beta, \mathbf{v}_\beta, (c_{\beta,t})_{t\in S_\beta})$, where $\mathbf{v}_\beta$ is the interpolation vector for S_β. Finally, for each secret key x, let $p_{x,\beta,t} = s_x(1 - R(x, q_\beta(t))) + r_{x,0}(t) + u_\beta r_{x,1}(t)$. Observe that, through decryption, the adversary can learn the values of $p_{x,\beta,t}$ for $t \in S_\beta$.

Let $r_{x,0}, r_{x,1}$ be the polynomials sampled when creating the key sk_x, and write $\mathsf{sk}_x = (\mathsf{sk}_{x,t})_t$ where $\mathsf{sk}_{x,t} \leftarrow \mathsf{KeyGen}_{\mathsf{FE}}(\mathsf{msk}_t, (x, r_{x,0}(t), r_{x,1}(t), s_x))$.

Hybrid 1. This is identical to Hybrid 0, except that now we condition on $|S_1 \cap S_2| \leq \lambda$. Since S_1, S_2 are random subsets of size U in a universe of size U^2, the expected size of $|S_1 \cap S_2|$ is 1. Standard concentration inequalities show that the probability of being larger than λ is $2^{-\Omega(\lambda)}$.

Hybrid 2. Here, we additionally condition on the pairs $(1, u_0)$ and $(1, u_1)$ being linearly independent. This occurs as long as $u_0 \neq u_1$, which is true except with probability $|\mathbb{F}|^{-1}$. By the assumption that $|\mathbb{F}|$ is exponentially large, this probability is negligible.

Hybrid 3. This is identical to Hybrid 2, except that we make the following changes. Let q'_β be a random polynomial of degree λ such that (1) $q'_\beta(t) = q_\beta(t)$ for $t \in S_1 \cap S_2$, and (2) $q'_\beta(0) = y^*_{1,\beta}$. Since $S_1 \cap S_2$ is guaranteed to have size at most λ (by the conditions placed in Hybrid 1), (1) and (2) give at most $\lambda + 1$ constraints on the degree-λ polynomial q'_β, so such a polynomial must exist.

Next, for each secret key attribute x, let $z_{x,\beta}(t) = r_{x,0}(t) + u_\beta r_{x,1}(t)$, and let $z'_{x,\beta}(t) = s_x(1 - R(x, q_\beta(t))) + z_{x,\beta}(t) - s_x(1 - R(x, q'_\beta(t)))$, which have degree $U - 1$. Then let $r'_{x,0}, r'_{x,1}$ be the polynomials defined as:

$$\begin{pmatrix} r'_{x,0}(t) \\ r'_{x,1}(t) \end{pmatrix} = \begin{pmatrix} 1 & u_0 \\ 1 & u_1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} z'_{x,0}(t) \\ z'_{x,1}(t) \end{pmatrix}.$$

Observe that the $r'_{x,0}(t), r'_{x,1}(t)$ have degree $t - 1$.

In Hybrid 3, we now switch to $\mathsf{sk}_{x,t} \leftarrow \mathsf{KeyGen}_{\mathsf{FE}}(\mathsf{msk}_t, (x, r_{x,0}(t)', r_{x,1}(t)', s_x))$ and $c_{\beta,t} \leftarrow \mathsf{Enc}_{\mathsf{FE}}(\mathsf{msk}_t, (q'_\beta(t), u_\beta))$. Observe that, by our choice of $r'_{x,\beta}(t)$ and $q'_\beta(t)$, when the adversary tries to decrypt the ciphertexts using its secret keys, even with this change it still recovers exactly the values $p_{x,\beta,t}$. Moreover, for $t \in S_1 \cap S_2$, we have that $q'_\beta(t) = q_\beta(t)$ and $r_{x,\beta}(t)' = r_{x,\beta}(t)$. Thus, the only t where the ciphertext and secret key attributes changed are $t \notin S_1 \cap S_2$. For these t, the adversary only sees a single ciphertext, and since decryption preserves the values of $p_{x,\beta,t}$, we can invoke the 1-SEL-KEY-CTXT security of Π, showing that this change is un-detectable.

Hybrids 4,5. These are identical to Hybrids 1, 2, respectively, except that we continue using $r'_{x,\beta}(t)$ and $q'_\beta(t)$. Observe that in Hybrid 5, because $q_\beta(t)$ is a random polynomial conditioned on $q_\beta(0) = y^*_{0,\beta}$, we have that $q'_\beta(t)$ is a random polynomial conditioned on $q'_\beta(t) = y^*_{1,\beta}$. While the two polynomials are correlated, $\mathcal{A}$'s view in Hybrid 3 is independent of $q_\beta(t)$ except through $q'_\beta(t)$. Likewise, $r_{x,\beta}(t)'$ are random polynomials with a constant coefficient of 0. Moreover, while $r'_{x,\beta}(t)$ and $r_{x,\beta}(t)$ are correlated, $\mathcal{A}$'s view is independent of $r_{x,\beta}(t)$ except through $r'_{x,\beta}(t)$. Thus, secret keys are still distributed correctly in this case, and ciphertexts are correctly distributed encryptions of $y^*_{1,1}$ and $y^*_{1,2}$. Since Hybrid 5 is indistinguishable from Hybrid 0, this proves Lemma 41. □

3.5 Our MFE

We now explain how to turn Construction 39 into an MFE with 1-SEL-ACC security. To do so, we let $\mathsf{mpk} = (\lambda, \kappa)$, which determines all other parameters in Construction 39. Then we define:

- $\mathsf{EncPK}(\mathsf{mpk})$: choose a random set $S \subseteq [T]$ of size U. Let $\mathbf{v}$ be the interpolation vector for S. For each $t \in S$, let c_t denote $(y'_t, (c_{t,i})_{i \in [n(\kappa)]})$ where $y'_t \geq \{0,1\}^{n(\kappa)}$ and $c_{t,i} \leftarrow \{0,1\}^\lambda$ are chosen uniformly. Output $c = (S, \mathbf{v}, (c_t)_{t \in S})$.

Lemma 42. *Assuming* Π_{FE} *has pseudorandom ciphertexts and q-SEL-KEY-CTXT security for* $q \in \{0,1\}$*, then* $\Pi_{\mathsf{MFE}} = (\mathsf{Setup}_{\mathsf{FE2}}, \mathsf{KeyGen}_{\mathsf{FE2}}, \mathsf{EncSK}_{\mathsf{FE2}}, \mathsf{EncPK}, \mathsf{Dec}_{\mathsf{FE2}})$ *is 1-SEL-ACC secure.*

Proof. Consider a supposed 1-SEL-ACC adversary $\mathcal{A}$ which commits to a ciphertext attribute y^*, and then only makes key queries on attributes x such that $R_\kappa(x, y^*) = 0$. Then we have that $R'_\kappa((x, r_0, r_1, s), (y^*, u)) = s + r_0 + ur$. We prove security via hybrids:

Hybrid 0. Here, the ciphertext seen by $\mathcal{A}$ is generated as $c^* \leftarrow \mathsf{EncSK}_{\mathsf{FE2}}(\mathsf{msk}, y^*)$. Write $c^* = (S, \mathbf{v}, (c_t)_{t \in S})$. We will need the fact that the interpolation vector $\mathbf{v}$ satisfies $\mathbf{v} \cdot \mathbf{1} \neq 0$, where $\mathbf{1}$ is the all-1's vector. Indeed, consider the constant polynomial $p(t) = 1$. The guarantee of $\mathbf{v}$ is that $\mathbf{v} \cdot (p(t))_{t \in S} = \mathbf{v} \cdot \mathbf{1} = p(0) = 1 \neq 0$.

Hybrid 1. This is the same as Hybrid 0, except that for $t \notin S$ we replace the secret key components $\mathsf{sk}_{x,t}$ for each x with $\mathsf{KeyGen}_{\mathsf{FE}}(\mathsf{msk}_t, (x, r'_{x,0}(t), r'_{x,1}, s'_x))$ for arbitrary $r'_{x,0}, r'_{x,1}, s'_x$ independent of $r_{x,0}, r_{x,1}, s_x$. This follows from 0-SEL-KEY-CTXT security since $\mathcal{A}$ receives no ciphertexts relative to msk_t. Note that after this replacement, $\mathcal{A}$ only "sees" $r_{x,0}(t), r_{x,1}(t)$ for $t \in S$. Let $\mathbf{r}_x \in \mathbb{F}^U$ be the vector of the values $r_{x,0}(t) + ur_{x,1}(t)$ for $t \in S$. Since $r_{x,0}(t), r_{x,1}(t)$ are random degree $U-1$ polynomials conditioned having a 0 constant coefficient, so is $r_{x,0}(t) + ur_{x,1}(t)$, and we therefore see that $\mathbf{r}_x$ is a random vector such that $\mathbf{v} \cdot \mathbf{r} = 0$.

Hybrid 2. This is the same as Hybrid 1, except that for $t \in S$, we replace each ciphertext component c_t in c^* with random bits. We claim that this is indistinguishable from Hybrid 1 by the ciphertext pseudorandomness of Π_{FE}. Indeed, observe that the adversary receives a single ciphertext c_t relative to each msk_t. Let $w_x(t) := R'_\kappa((x, r_{x,0}(t), r_{x,1}(t), s_x), (y^*, u)) = s_x + r_{x,0}(t) + ur_{x,1}(t)$. Let $\mathbf{w}_x$ be the vector of the $w_x(t)$, which is equal to $\mathbf{r}_x + s_x\mathbf{1}$. For fixed u, $\mathbf{r}_x$ is uniform in a subspace of dimension $U-1$, and $s_x\mathbf{1}$ is uniform in a subspace of dimension 1. Since $\mathbf{v} \cdot \mathbf{1} \neq 0$ and the subspace containing $\mathbf{r}_x$ is orthogonal to $\mathbf{v}$, the two subspaces containing $\mathbf{r}_x$ and $s_x\mathbf{1}$ only intersect at the origin. Hence, the subspaces span the entirety of $\mathbb{F}^U$, and hence $\mathbf{w}_x = \mathbf{r}_x + s_x\mathbf{1}$ is uniform in $\mathbb{F}^U$. This holds for each secret key attribute x queried by the adversary, and the $\mathbf{w}_x$ are independent. Thus, we have, for each $t \in S$, component t of each secret key is a secret key for Π_{FE} where the attribute (x, r_0, r_1, s) is such that $R'_\kappa((x, r_0, r_1, s), (y^*, u))$ is uniform. We can therefore invoke ciphertext pseudorandomness of Π_{FE} for each component $t \in S$ to conclude that we can replace the corresponding ciphertext components c_t with uniform bits.

Hybrid 3. Here, for $t \notin S$ we go back to $\mathsf{sk}_{x,t} \leftarrow \mathsf{KeyGen}_{\mathsf{FE}}(\mathsf{msk}_t, (x, r_{x,0}(t), r_{x,1}(t), s_x))$, again using 0-SEL-KEY-CTXT. The result is that we have replaced all the c_t in c^* with uniform random bits, which is equivalent to generating $c^* \leftarrow \mathsf{EncPK}(\mathsf{mpk})$. □

3.6 Decryption in $\mathsf{NC}^1_{\neq 0}(\mathbb{F})$

We briefly explain how $\mathsf{Dec}_{\mathsf{FE2}} \in \mathsf{NC}^1_{\neq 0}(\mathbb{F})$, under the assumptions of Theorem 27. $\mathsf{Dec}_{\mathsf{FE2}}$ first runs $\mathsf{Dec}_{\mathsf{FE}}$ several times in parallel to compute $p(t)$. In turn, each run of $\mathsf{Dec}_{\mathsf{FE}}$ performs several parallel symmetric key decryptions $\mathsf{Dec}_{\mathsf{sk}}$ to recover terms in the matrix branching program, followed by an evaluation of $\mathsf{Dec}_{\mathsf{RRE}}$. $\mathsf{Dec}_{\mathsf{sk}}$ just involves a PRF computation and a field subtraction, and we assumed the PRF is computable in log-depth by arithmetic formula. $\mathsf{Dec}_{\mathsf{RRE}}$ is an iterated matrix product of constant-size matrices. This can be computed by a log-depth sequence of matrix

multiplications, and since the matrices are constant-size each matrix multiplication involves a constant number of arithmetic operations. By arranging the multiplications into a binary tree, the result is that $\mathsf{Dec}_{\mathsf{RRE}}$ is computable by log-depth arithmetic computations. Thus, computing $p(t)$ takes only log depth.

Next $\mathsf{Dec}_{\mathsf{FE2}}$ computes an inner product of the vector of $p(t)$ with $\mathbf{v}$, which again can be computed by log-depth arithmetic formula. Finally, $\mathsf{Dec}_{\mathsf{FE2}}$ performs a "not equal to zero" check to get the output. The end result is that $\mathsf{Dec}_{\mathsf{FE2}}$ is computable in $\mathsf{NC}^1_{\neq 0}(\mathbb{F})$.

4 Weak Decoder-Based Accept Security

Here, we assume we have an mcMFE scheme for some relation R, which is simultaneously 2-SEL-CTXT secure, 2-SEL-M secure, and 1-SEL-ACC secure, such as the mcMFE guaranteed by Corollary 28. Following Theorem 23 as proved in [GKW18], the mcMFE also has decoder-based notions SEL-DEC-M and SEL-DEC-CTXT security. But because we do not have 2-SEL-ACC security, we do not have SEL-DEC-ACC security, which means we cannot use existing frameworks (Theorems 24 and 25, [GKW18, GKW19]) to get traitor tracing.

We therefore give a new, significantly weaker, notion of decoder-based accept security for mcMFE. We also show a compiler that takes any 1-SEL-ACC secure mcMFE, and compiles it into a scheme satisfying our weak decoder-based definition, while also preserving 2-SEL-M and 2-SEL-CTXT security. Then in Section 5 we show, despite having a much weaker decoder-based security, that it nevertheless suffices for traitor tracing.

Definition 43 (*Weak* decoder-based accept security)**.** *An mcMFE Π is weak selective decoder-based accept secure (weak SEL-DEC-ACC secure) if there are constants $\alpha, \beta, \gamma \in (0,1)$ such that, for every polynomial $\kappa(\lambda)$ and every stateful PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that and every λ, the following is true:*

- *Run $\mathcal{A}(1^\lambda)$ to get ciphertext attribute $y^* \in \{0,1\}^{n(\kappa(\lambda))}$.*
- *Now run $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)})$ and send mpk to $\mathcal{A}$.*
- *$\mathcal{A}$ can now make arbitrary queries on key attributes x with the guarantee that $R(x, y^*) = 1$; it receives in response $\mathsf{sk}_x \leftarrow \mathsf{KeyGen}(\mathsf{msk}, x)$. Let X be the set of all queries x that are made.*
- *$\mathcal{A}$ outputs a decoder D.*
- *Let $\mathsf{Adv}(\mathsf{D}, \mathsf{mpk}) = 2\Pr[\mathsf{D}(c, k_b) = b : \substack{(c,k_0) \leftarrow \mathsf{EncPK}(\mathsf{mpk}) \\ k_1 \leftarrow \mathcal{K}_\kappa, b \leftarrow \{0,1\}}] - 1$ be the advantage of D in distinguishing real keys from random for publicly generated ciphertexts. Let $\mathsf{Adv}'(\mathsf{D}, \mathsf{msk}, y^*) = 2\Pr[\mathsf{D}(c, k_b) = b : \substack{(c,k_0) \leftarrow \mathsf{EncSK}(\mathsf{msk}, y^*) \\ k_1 \leftarrow \mathcal{K}_\kappa, b \leftarrow \{0,1\}}] - 1$ be the advantage for ciphertexts with attribute y^*.*

  Let $\mathsf{Good}_\alpha(\mathsf{D}, \mathsf{mpk})$ be the event $\mathsf{Adv}(\mathsf{D}, \mathsf{mpk}) \geq \alpha$, and $\mathsf{Good}'_\beta(\mathsf{D}, \mathsf{msk}, y^*)$ be the event $\mathsf{Adv}'(\mathsf{D}, \mathsf{msk}, y^*) \geq \beta$. Then $\Pr[\mathsf{Good}'_\beta(\mathsf{D}, \mathsf{msk}, y^*)] \geq \gamma \Pr[\mathsf{Good}_\alpha(\mathsf{D}, \mathsf{mpk})] - \mathsf{negl}(\lambda)$.

Our main theorem of this section will be:

Theorem 44. *Let Π be a 1-SEL-ACC secure mcMFE for relation R. Then there exists a protocol Π' for R that is weak SEL-DEC-ACC secure, such that the sizes of all parameters and running times in Π' are at most a constant-factor worse than in Π. If Π is q-SEL-CTXT secure, then so is Π'.*

By combining with Corollary 28 and Theorem 23, we immediately obtain:

Corollary 45. *Let $R = (R_\kappa)_\kappa$ be a relation that is computable by log-depth arithmetic formula over an exponentially large field $\mathbb{F} = (\mathbb{F}_\kappa)_\kappa$. Assume the existence of a weak PRF PRF with outputs in $\mathbb{F}$ and which is computable by log-depth arithmetic formula over $\mathbb{F}$ using pre-computation. Moreover assume the existence of SEL-M secure attribute-based encryption for log-depth arithmetic formula over $\mathbb{F}$ followed by a "not equal to zero" test. Then there exists an mcMFE for R that is simultaneously SEL-DEC-CTXT secure, SEL-DEC-M secure, and* weak *SEL-DEC-ACC secure.*

Proof of Theorem 44. Let $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{EncPK}, \mathsf{EncSK}, \mathsf{Dec})$ be the assumed mcMFE. Fix any constants $\ell, \alpha, \beta, \gamma$ such that

$$0 < \beta < \alpha < 1 \quad , \quad 0 < \gamma < 1 \quad , \quad \ell \in \mathbb{N} \quad \text{and} \quad \ell(1-\gamma)(\alpha-\beta)^2 > 4 \ .$$

For example, take $\ell = 5$, $\alpha = 19/20$, $\beta = 1/20$, $\gamma = 1/162$.

Construction 46. *Let $\Pi' = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{EncPK}', \mathsf{EncSK}', \mathsf{Dec}')$ be defined as follows:*

- $\mathsf{Setup}'(1^\lambda, 1^\kappa)$*: for $\zeta \in [\ell]$, run $(\mathsf{mpk}_\zeta, \mathsf{msk}_\zeta) \leftarrow \mathsf{Setup}(1^\lambda, 1^\kappa)$ and output $(\mathsf{mpk}', \mathsf{msk}')$ where $\mathsf{mpk}' = (\mathsf{mpk}_\zeta)_\zeta$ and $\mathsf{msk}' = (\mathsf{msk}_\zeta)_\zeta$.*
- $\mathsf{KeyGen}'(\mathsf{msk}', x)$*: for $\zeta \in [\ell]$ run $\mathsf{sk}_{x,\zeta} \leftarrow \mathsf{KeyGen}(\mathsf{msk}_\zeta, x)$ and output $\mathsf{sk}'_x = (\mathsf{sk}_{x,\zeta})_\zeta$.*
- $\mathsf{EncSK}'(\mathsf{msk}', y)$*: sample $\zeta \leftarrow [\ell]$, run $(c, k) \leftarrow \mathsf{EncSK}(\mathsf{msk}_\zeta, y)$, and output $c' = (\zeta, c)$ and k.*
- $\mathsf{EncPK}'(\mathsf{mpk}')$*: sample $\zeta \leftarrow [\ell]$, run $(c, k) \leftarrow \mathsf{EncPK}(\mathsf{mpk}_\zeta)$, and output $c' = (\zeta, c)$ and k.*
- $\mathsf{Dec}'(\mathsf{sk}'_x, c')$*: run $\mathsf{Dec}(\mathsf{sk}_{x,\zeta}, c)$.*

Correctness of Π' follows immediately from the correctness of Π. Also, that Π' preserves q-SEL-CTXT security is immediate. We now prove weak SEL-DEC-ACC security of Π' assuming the 1-SEL-ACC security of Π. Let $\kappa(\lambda)$ be a polynomial and $\mathcal{A}$ be a hypothetical adversary for the SEL-DEC-ACC security of Π'. Let

$$\delta(\lambda) = \gamma \Pr[\mathsf{Good}_\alpha(\mathsf{D}, \mathsf{mpk}')] - \Pr[\mathsf{Good}'_\beta(\mathsf{D}, \mathsf{msk}', y^*)] \ .$$

We must show that $\delta(\lambda)$ is negligible. Let $p_\zeta = \Pr\left[\mathsf{D}(c, k_b) = b : \substack{(c,k_0) \leftarrow \mathsf{EncSK}(\mathsf{msk}_\zeta, y^*) \\ k_1 \leftarrow \mathcal{K}_\kappa, b \leftarrow \{0,1\}}\right]$ and $q_\zeta = \Pr\left[\mathsf{D}(c, k_b) = b : \substack{(c,k_0) \leftarrow \mathsf{EncPK}(\mathsf{mpk}_\zeta) \\ k_1 \leftarrow \mathcal{K}_\kappa, b \leftarrow \{0,1\}}\right]$. Let $p = \sum_\zeta p_\zeta/\ell$ and $q = \sum_\zeta q_\zeta/\ell$. Let $\Delta_\zeta = p_\zeta - q_\zeta$ and $\Delta = \sum_\zeta \Delta_\zeta/\ell = p - q$. Δ and Δ_ζ are therefore random variables taking values in $[-1, 1]$. Now we will show that the linear correlation between distinct Δ_i is small.

Lemma 47. *Let Ev be an event that is efficiently recognizable given the inputs and outputs of $\mathcal{A}$. If Π is q-SEL-ACC secure, then there is a negligible function negl such that for all $i, j \in [\ell], i \neq j$, $\| \mathbb{E}[\ \Delta_i \Delta_j \mid \mathsf{Ev}\ ] \| \times \Pr[\mathsf{Ev}] \leq \mathsf{negl}(\lambda)$.*

Proof. Let $\mathcal{B}^{(i,j)}$ be the following adversary for the 1-SEL-CTXT security of Π:

- $\mathcal{B}^{(i,j)}(1^\lambda)$ runs $y^* \leftarrow \mathcal{A}(1^\lambda)$, $(\mathsf{mpk}_\zeta, \mathsf{msk}_\zeta) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\kappa(\lambda)})$ for $\zeta \in [\ell] \setminus \{i\}$, and outputs y^*.
- When $\mathcal{B}^{(i,j)}$ receives mpk, it sets $\mathsf{mpk}_i = \mathsf{mpk}$ and $\mathsf{mpk}' = (\mathsf{mpk}_\zeta)_{\zeta \in [\ell]}$, which it sends to $\mathcal{A}$.

- When $\mathcal{A}$ queries key attribute x, $\mathcal{B}$ queries x, receiving sk_x in response. It sets $\mathsf{sk}_{x,i} = \mathsf{sk}_x$ and computes $\mathsf{sk}_{x,\zeta} \leftarrow \mathsf{KeyGen}(\mathsf{msk}_\zeta, x)$ for $\zeta \in [\ell] \setminus \{i\}$. It sends $\mathsf{sk}'_x = (\mathsf{sk}_{x,\zeta})_{\zeta \in [\ell]}$ to $\mathcal{A}$.
- When $\mathcal{A}$ produces a decoder D, $\mathcal{B}^{(i,j)}$ receives ciphertext/key pair (c^*, k^*). It determines in Ev occurs. If not, it outputs a random bit. Otherwise, if Ev does occur, then $\mathcal{B}^{(i,j)}$ sets $k^*_0 = k^*$, and chooses random $z, z^*, \beta \leftarrow \{0,1\}, k^*_1 \leftarrow \mathcal{K}_\lambda$. It lets $(d_0, \ell_{0,0}) \leftarrow \mathsf{EncSK}(\mathsf{msk}_j, y^*)$ and $(d_1, \ell_{1,0}) \leftarrow \mathsf{EncPK}(\mathsf{mpk}_j)$ and $\ell_{0,1}, \ell_{1,1} \leftarrow \mathcal{K}_\lambda$. Finally, it outputs $\beta \oplus (z^* \oplus \mathsf{D}((i, c^*), k^*_{z^*})) \oplus (z \oplus \mathsf{D}((j, d_\beta), \ell_{\beta,z}))$.

Let W^*_0 be 1 if and only if $\mathsf{D}((i, c^*), k^*_{z^*}) = z^*$ when $(c^*, k^*_0) \leftarrow \mathsf{EncSK}(\mathsf{msk}_i, y^*)$. Let W^*_1 be the same but when $(c^*, k^*_0) \leftarrow \mathsf{EncPK}(\mathsf{mpk}_i)$. Likewise define W_β to be 1 if and only if $\mathsf{D}((j, d_\beta), \ell_{\beta,z}) = z$. Remember that $(d_0, \ell_{0,0}) \leftarrow \mathsf{EncSK}(\mathsf{msk}_j, y^*)$ and $(d_1, \ell_{1,0}) \leftarrow \mathsf{EncPK}(\mathsf{mpk}_j)$. Then D is created independently of $b, \beta, z, z^*, c^*, k^*_{z^*}, d_\beta$ and $\ell_{\beta,z}$. Therefore, Ev is independent these terms. Then:

$$\begin{aligned}
\Pr[W^*_0 = W_0 = 1] &= \mathbb{E}[p_i p_j] & \Pr[W^*_0 = W_0 = 0] &= \mathbb{E}[(1-p_i)(1-p_j)] \\
\Pr[W^*_0 = W_1 = 1] &= \mathbb{E}[p_i q_j] & \Pr[W^*_0 = W_1 = 0] &= \mathbb{E}[(1-p_i)(1-q_j)] \\
\Pr[W^*_1 = W_0 = 1] &= \mathbb{E}[q_i p_j] & \Pr[W^*_1 = W_0 = 0] &= \mathbb{E}[(1-q_i)(1-p_j)] \\
\Pr[W^*_1 = W_1 = 1] &= \mathbb{E}[q_i q_j] & \Pr[W^*_1 = W_1 = 0] &= \mathbb{E}[(1-q_i)(1-q_j)]
\end{aligned}$$

The above hold even when the probabilities and expectations are conditioned on Ev. Therefore,

$$\begin{aligned}
\Pr[b' = b] =& \frac{1}{2}\Pr[\neg\mathsf{Ev}] \\
&+ \Pr[b = 0 \wedge \beta = 0]\Pr[W^*_0 = W_0 | \mathsf{Ev}]\Pr[\mathsf{Ev}] \\
&+ \Pr[b = 0 \wedge \beta = 1]\Pr[W^*_0 \neq W_1 | \mathsf{Ev}]\Pr[\mathsf{Ev}] \\
&+ \Pr[b = 1 \wedge \beta = 0]\Pr[W^*_1 \neq W_0 | \mathsf{Ev}]\Pr[\mathsf{Ev}] \\
&+ \Pr[b = 1 \wedge \beta = 1]\Pr[W^*_1 = W_1 \mathsf{Ev}]\Pr[\mathsf{Ev}] \\
=& \frac{1}{2}\Pr[\neg\mathsf{Ev}] + \frac{1}{4}\mathbb{E}[p_i p_j + (1-p_i)(1-p_j) + p_i(1-q_j) + (1-p_i)q_j \\
&+ q_i(1-p_j) + (1-q_i)p_j + q_i q_j + (1-q_i)(1-q_j) | \mathsf{Ev}]\Pr[\mathsf{Ev}] \\
=& \frac{1}{2}\Pr[\neg\mathsf{Ev}] + \frac{1}{2}\mathbb{E}[1 + (p_i - q_i)(p_j - q_j) | \mathsf{Ev}]\Pr[\mathsf{Ev}] \\
=& \frac{1}{2} + \frac{1}{2}\mathbb{E}[\Delta_i \Delta_j | \mathsf{Ev}]\Pr[\mathsf{Ev}] \ .
\end{aligned}$$

By the 1-SEL-ACC security of Π, we therefore have that $\| \mathbb{E}[\Delta_0 \Delta_1 | \mathsf{Ev}] \| \times \Pr[\mathsf{Ev}] \leq \mathsf{negl}(\lambda)$. □

We now finish the proof of Theorem 44. Assume toward contradiction that $\delta(\lambda)$ is non-negligible. Then there is an inverse polynomial $T(\lambda) = n^{-O(1)}$ and infinite set $\Lambda \in \mathbb{N}$ such that $\delta(\lambda) \geq T(\lambda)$ for $\lambda \in \Lambda$. Let $\tau(\lambda) = \Pr[\mathsf{Good}_\alpha(\mathsf{D}, \mathsf{mpk}')]$. Then $\tau(\lambda) \geq \delta(\lambda)/\gamma \geq \delta(\lambda)$ for $\lambda \in \Lambda$.

Let α', γ' be arbitrary constants such that $\beta < \alpha' < \alpha$, $\gamma' > \gamma$, and $\ell(1-\gamma')(\alpha' - \beta)^2 > 4$. Such constants are guaranteed to exist since $\ell(1-\gamma)(\alpha-\beta)^2 > 4$, meaning taking small enough perturbations to α and γ will not violate the inequality.

Now we define Ev. Consider estimating q by generating $O(\lambda)$-many samples $(c^*_i, k^*_{i,0}) \leftarrow \mathsf{EncPK}(\mathsf{mpk}')$, $k^*_{i,1} \leftarrow \mathcal{K}_\lambda$, $z \leftarrow \{0,1\}$ and letting $\tilde{q}$ be the fraction of samples for which $\mathsf{D}(c^*_i, k^*_z) = z$. By standard concentration inequalities, we can guarantee that $\Pr[|q - \tilde{q}| < (\alpha' - \alpha)/4] \geq 1 - 2^{-\lambda}$. Let Ev be the probability that $2\tilde{q} - 1 \geq (\alpha' + \alpha)/2$.

Thus $\Pr[\mathsf{Ev}] \geq \Pr[\mathsf{Ev} \wedge 2q - 1 \geq \alpha] \geq \Pr[2q - 1 \geq \alpha] - \Pr[|\tilde{q} - q| > (\alpha' - \alpha)/4] = \tau - 2^{-\lambda}$. By similar logic, $\Pr[2q - 1 < \alpha' \wedge \mathsf{Ev}] \leq 2^{-\lambda}$. Recall that $\Delta = \sum_\zeta \Delta_\zeta/\ell$, and $\Delta_\zeta \in [-1, 1]$. Thus,

$$\begin{aligned}\mathbb{E}[\Delta^2|\mathsf{Ev}] &= \frac{1}{\ell^2}\left(\sum_\zeta \mathbb{E}[\Delta_\zeta^2|\mathsf{Ev}] + \sum_{\substack{i,j\in[\ell]\\ i\neq j}} \mathbb{E}[\Delta_i\Delta_j|\mathsf{Ev}]\right) \\ &\leq \frac{1}{\ell^2}\left(\ell + \sum_{\substack{i,j\in[\ell]\\ i\neq j}} \mathbb{E}[\Delta_i\Delta_j|\mathsf{Ev}]\right) \leq \frac{1}{\ell} + \mathsf{negl}(\lambda)/\Pr[\mathsf{Ev}] \ .\end{aligned}$$

Then by Markov's inequality, we have:

$$\begin{aligned}\Pr[\Delta \geq (\alpha' - \beta)/2|\mathsf{Ev}] &= \Pr[\Delta^2 \geq (\alpha' - \beta)^2/4|\mathsf{Ev}] \\ &\leq \frac{4\,\mathbb{E}[\Delta^2|\mathsf{Ev}]}{(\alpha' - \beta)^2} = \frac{4}{\ell(\alpha' - \beta)^2} + \mathsf{negl}(\lambda)/\Pr[\mathsf{Ev}] \ .\end{aligned}$$

For large enough $\lambda \in \Lambda$, we can lower-bound $\Pr[\mathsf{Ev}] \geq \tau(\lambda) - 2^{-\lambda} \geq T(\lambda) - 2^{-\lambda} \geq T(\lambda)/2$, which means we can lower bound $\Pr[\Delta \geq (\alpha' - \beta)/2|\mathsf{Ev}] \leq \frac{4}{\ell(\alpha'-\beta)^2} + 2\mathsf{negl}(\lambda)/T(\lambda)$. Since $2\mathsf{negl}(\lambda)/T(\lambda)$ goes to zero for large enough $\lambda \in \Lambda$, we can bound $\Pr[\Delta \geq (\alpha' - \beta)/2|\mathsf{Ev}]$ by any constant larger than $\frac{1}{\ell(\alpha'-\beta)^2}$. In particular, $\Pr[\Delta \geq (\alpha' - \beta)/2|\mathsf{Ev}] \leq (1 - \gamma')$, or equivalently $\Pr[\Delta < (\alpha' - \beta)/2 \wedge \mathsf{Ev}] \geq \gamma' \Pr[\mathsf{Ev}]$. We therefore have that:

$$\begin{aligned}\Pr[\mathsf{Good}'_\beta(\mathsf{D}, \mathsf{msk}, y^*)] &= \Pr[2p - 1 \geq \beta] \\ &\geq \Pr[2q - 1 \geq \alpha' \wedge |p - q| < (\alpha' - \beta)/2] \\ &= \Pr[2q - 1 \geq \alpha' \wedge |\Delta| < (\alpha' - \beta)/2] \\ &\geq \Pr[2q - 1 \geq \alpha' \wedge |\Delta| < (\alpha' - \beta)/2 \wedge \mathsf{Ev}] \\ &\geq \Pr[|\Delta| < (\alpha' - \beta)/2 \wedge \mathsf{Ev}] \\ &\qquad - \Pr[2q - 1 < \alpha' \wedge |\Delta| < (\alpha' - \beta)/2 \wedge \mathsf{Ev}] \\ &\geq \Pr[|\Delta| < (\alpha' - \beta)/2 \wedge \mathsf{Ev}] - \Pr[2q - 1 < \alpha' \wedge \mathsf{Ev}] \\ &\geq \gamma' \Pr[\mathsf{Ev}] - 2^{-\lambda} \geq \gamma'\tau(\lambda) - (1 + \gamma')2^{-\lambda} \\ &\geq \gamma\tau(\lambda) = \gamma \Pr[\mathsf{Good}_\alpha(\mathsf{D}, \mathsf{mpk})] \ ,\end{aligned}$$

for large enough $\lambda \in \Lambda$. In other words, $\delta(\lambda) \leq 0$ for large enough $\lambda \in \Lambda$. But this contradicts that $\delta(\lambda) \geq T(\lambda) > 0$ for $\lambda \in \Lambda$. Thus, $\delta(\lambda)$ must in fact negligible. □

5 From Weak mcMFE to Risky Threshold Traitor Tracing

Following Corollary 45 in Section 4, we have an mcMFE for log-depth arithmetic formula that is SEL-DEC-M and SEL-DEC-CTXT secure, but only *weak* SEL-DEC-ACC secure. Now observe that $R^{\mathsf{PLBE}}, R^{\mathsf{EIPLBE}} \in \mathsf{NC}^1 \subseteq \mathsf{NC}^1(\mathbb{F})$. Thus, under the assumptions of Corollary 45, we have PLBE and EIPLBE schemes that are SEL-DEC-M, SEL-DEC-CTXT, and weak SEL-DEC-ACC secure. We now use these to build traitor tracing:

Theorem 48. *If there exists an a PLBE scheme that is SEL-DEC-M secure, SEL-DEC-CTXT secure, and* weak *SEL-DEC-ACC secure, then there exists constants ϵ, δ and an (ϵ, δ)-threshold risky traceable index-only traitor tracing scheme that is with stateless key generation.*

Theorem 49. *If there exists an a EIPLBE scheme that is SEL-DEC-M secure, SEL-DEC-CTXT secure, and* weak *SEL-DEC-ACC secure, then there exists constants ϵ, δ and an (ϵ, δ)-threshold risky index-based embedded-identity traitor tracing scheme.*

The differences between our Theorems 48 and 49 and the analogous Theorems 24 and 25 as proved in [GKW18, GKW19] is that our theorems only require weak decoder-based accept security, but only achieve threshold risky traitor tracing. Fortunately, we can lift these theorems to the full tracing setting using Theorem 21 from [Zha20]. In the embedded-identity case, we can also employ Theorem 26 to move to a full (non-index-based) embedded-identity scheme. Combining these results together therefore proves Theorem 2. It therefore remains to prove Theorems 48 and 49. Note that Theorem 49 implies Theorem 48, but uses a different more complicated construction. We therefore start with Theorem 48 as a warm-up.

5.1 The Index-only Case (Theorem 48)

Let $\Pi_{\mathsf{mcMFE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{EncSK}, \mathsf{EncPK}, \mathsf{Dec})$ be the assumed PLBE scheme that is SEL-DEC-M secure, SEL-DEC-CTXT secure, and *weak* SEL-DEC-ACC secure. Let (α, β, γ) be the constants for weak SEL-DEC-ACC security. Recall that PLBE schemes are mcMFE schemes with the functionality $R = (R_\kappa)_\kappa$ where $R_\kappa : [2^\kappa] \times [0, 2^\kappa] \to \{0, 1\}$ is defined as $R(\mathsf{id}, t) = \mathbb{1}(\mathsf{id} \leq t)$. We construct the following traitor tracing scheme:

Construction 50. *Let* $\Pi_{\mathsf{TT}} = (\mathsf{Setup}', \mathsf{KeyGen}, \mathsf{Enc} = \mathsf{EncPK}, \mathsf{Dec}, \mathsf{Trace})$*, where* $\mathsf{Setup}'(1^\lambda) = \mathsf{Setup}(1^\lambda, 1^\lambda)$*, setting* $\kappa = \lambda$ [6]*, and where* $\mathsf{Trace}^{\mathsf{D}}(\mathsf{msk}, 1^N, 1^{1/\epsilon})$ *works as follows:*

- *For each* $t \in [0, \kappa]$*, let* $p_t = 2 \times \Pr\left[\mathsf{D}(c, k_b) = b : \substack{(c,k_0) \leftarrow \mathsf{EncSK}(\mathsf{mpk}, t) \\ k_1 \leftarrow \mathcal{K}_\lambda, b \leftarrow \{0,1\}}\right] - 1$*. Compute an estimate* $\tilde{p}_t$ *of* p_t *by taking* $O(\lambda N^2/\beta^2)$ *samples. The number of samples is chosen so that except with probability* $2^{-\lambda}$ *over the choice of samples,* $\|\tilde{p}_t - p_t\| < \beta/6(N+1)$*.*
- *Output* $A = \{t \in [\kappa] : \|\tilde{p}_t - \tilde{p}_{t-1}\| > \beta/2(N+1)\}$*.*

To show Π_{TT} is $(\epsilon = \alpha, \delta = \gamma)$-threshold risky traceable, consider $\mathcal{A}$ outputting a decoder D.

Honest Users are Not Accused. Let $C \subseteq [N]$ be the set of identities id queried by $\mathcal{A}$. Observe that for $t \notin C$, $\mathcal{A}$ does not possess any secret keys for identities $\mathsf{id} = t$. Thus for $\mathsf{id} \in C$, $R_\kappa(\mathsf{id}, t) = R_\kappa(\mathsf{id}, t-1) = 0$ (if $\mathsf{id} > t$) or $R_\kappa(\mathsf{id}, t) = R_\kappa(\mathsf{id}, t-1) = 1$ (if $\mathsf{id} < t$). In other words, $\mathcal{A}$ does not possess any secret keys that can distinguish ciphertext attribute t from $t-1$. Therefore, by SEL-DEC-CTXT security, except with negligible probability, $\|p_t - p_{t-1}\| < \beta/6(N+1)$. Then by the triangle inequality $\|\tilde{p}_t - \tilde{p}_{t-1}\| \leq \|p_t - p_{t-1}\| + \|\tilde{p}_t - p_t\| + \|\tilde{p}_{t-1} - p_{t-1}\| < 3 \times \beta/6(N+1) = \beta/2(N+1)$, except with negligible probability. Therefore, $t \notin A$. Thus, except with negligible probability, $A \subseteq C$.

[6] Recall that in index-only traitor tracing schemes, we can upper-bound the length of identities ν by λ. We therefore set $\kappa = \nu = \lambda$.

Some user is accused with reasonable probability. We now show that if the decoder is sufficiently good, some user will be accused with reasonable probability. For a decoder D and public key mpk, recall that $\mathsf{Good}_\alpha(\mathsf{D}, \mathsf{mpk})$ means $2 \times \Pr\left[\mathsf{D}(c, k_b) = b : \substack{(c,k_0)\leftarrow\mathsf{Enc}(\mathsf{mpk}) \\ b\leftarrow\{0,1\}, k_1\leftarrow\mathcal{K}_\lambda}\right] - 1 \geq \alpha$. Define $\mathsf{Good}'_\beta(\mathsf{D}, \mathsf{msk})$ as $2 \times \Pr\left[\mathsf{D}(c, k_b) = b : \substack{(c,k_0)\leftarrow\mathsf{Enc}(\mathsf{msk},N) \\ b\leftarrow\{0,1\}, k_1\leftarrow\mathcal{K}_\lambda}\right] - 1 \geq \alpha$. Since all id queried by the adversary satisfy $\mathsf{id} \leq N$, we have $R_\lambda(\mathsf{id}, N) = 1$. SEL-DEC-ACC security then means

$$\Pr[\mathsf{Good}'_\beta(\mathsf{D}, \mathsf{msk})] \geq \gamma \Pr[\mathsf{Good}_\alpha(\mathsf{D}, \mathsf{mpk})] - \mathsf{negl}(\lambda) \ .$$

Let $\mathsf{Bad}(\mathsf{D}, \mathsf{msk})$ be the event that $p_0 \geq \beta/(N+1)$. Since $\mathsf{id} \geq 1$, we have that $R_\lambda(\mathsf{id}, 0) = 0$ for all id. By SEL-DEC-M security, we have that $\Pr[\mathsf{Bad}(\mathsf{D}, \mathsf{msk})]$ is negligible.

Now assume that $\mathsf{Good}'_\beta(\mathsf{D}, \mathsf{msk})$ happens and $\mathsf{Bad}(\mathsf{D}, \mathsf{msk})$ does not. We show that we are very likely to trace to a user. Indeed, $\mathsf{Good}'_\beta(\mathsf{D}, \mathsf{msk})$ means that $p_N \geq \beta$ and $\mathsf{Bad}(\mathsf{D}, \mathsf{msk})$ means that $p_0 \leq \beta/(N+1)$. Therefore, there must exist a $t \in [N]$ such that $|p_t - p_{t-1}| \geq \beta/(N+1)$. We then have, via the triangle inequality, that $|\tilde{p}_t - \tilde{p}_{t-1}| \geq \|p_t - p_{t-1}\| - \|\tilde{p}_t - p_t\| - \|\tilde{p}_{t-1} - p_{t-1}\| \geq 2\beta/3(N+1) > \beta/2(N+1)$ except with probability at most $2 \times 2^{-\lambda}$. In this case, $t \in A$, meaning A is non-empty. Thus, $\Pr[|A| > 0] \geq \gamma \times \Pr[\mathsf{Good}_\alpha(\mathsf{D}, \mathsf{mpk})] - \mathsf{negl}(\lambda)$, as desired.

5.2 The Index-Based Embedded-Identity Case (Theorem 49)

While Construction 50 has an identity-space that is exponential and therefore syntactically can be used as an embedded-identity traitor tracing scheme, it lacks a security proof for this use case. This is because tracing an exponentially-large identity-space using PLBE requires a variant of binary search [NWZ16], meaning that the ciphertext attributes that the decoder is tested on depend on the behavior of the decoder on numerous previous ciphertexts. This means the reduction proving of traceability needs many ciphertexts; it turns out that the number of ciphertexts grows with the number of colluding users. Unfortunately, the PLBE we obtain only has security for two ciphertexts, which is insufficient, and any natural generation to handle a large number of ciphertexts will result in ciphertexts that are too large. This issue is also why [GKW18] in the lattice setting cannot be used as an embedded-identity scheme. Instead, as used in the lattice-based setting in [GKW19], we will use EIPLBE, which enables a tracing algorithm where the attributes tested are fixed, and independent of the decoder's behavior on other ciphertexts. This allows a reduction to decoder-based security, and in turn security for two ciphertexts.

Let $\Pi_{\mathsf{mcMFE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{EncSK}, \mathsf{EncPK}, \mathsf{Dec})$ be the given EIPLBE scheme that is SEL-DEC-M, SEL-DEC-CTXT, and *weak* SEL-DEC-ACC secure. Let (α, β, γ) be the constants for weak SEL-DEC-ACC security. Recall that EIPLBE schemes are mcMFE schemes with the functionality $R = (R_{\kappa,\ell})_{\kappa,\ell}$ where $R_{\kappa,\ell}$ takes as input secret key attributes $(j, \mathsf{id}) \in [2^\kappa] \times \{0,1\}^\ell$ and ciphertext attributes $(t, i) \in [0, 2^\kappa] \times [0, \ell]$, and is defined as

$$R_{\kappa,\ell}((j, \mathsf{id}), (t, i, b)) = \begin{cases} \mathbb{1}(\ j \leq t\ ) & \text{if } i = 0 \\ \mathbb{1}(\ j < t \vee (j, \mathsf{id}_i) = (t, 1)\ ) & \text{if } i > 0\ . \end{cases}$$

We construct the following index-based embedded identity traitor tracing scheme:

Construction 51. *Let* $\Pi_{\mathsf{TT}} = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Enc} = \mathsf{EncPK}, \mathsf{Dec}, \mathsf{Trace})$ *where* $\mathsf{Setup}'(1^\lambda, 1^\nu) = \mathsf{Setup}(1^\lambda, 1^{(\lambda,\nu)})$*, setting* $\kappa = \lambda, \ell = \nu$*, and where* $\mathsf{KeyGen}', \mathsf{Trace}$ *are defined as follows:*

- $\mathsf{KeyGen}'(\mathsf{msk}', j, \mathsf{id})$: *run* $\mathsf{KeyGen}(\mathsf{msk}, (t, \mathsf{id}))$*, where* t *is the index, and* id *is the identity.*

- $\mathsf{Trace}^{\mathsf{D}}(\mathsf{msk}, 1^N, 1^{1/\epsilon})$ *work as follows:*
  - *For each* $t \in [0,\kappa], i \in [0,\ell]$, *let* $p_{t,i} = 2 \times \Pr\left[\mathsf{D}(c,k_b) = b : \substack{(c,k_0)\leftarrow\mathsf{EncSK}(\mathsf{mpk},(t,i)) \\ k_1\leftarrow\mathcal{K}_\lambda, b\leftarrow\{0,1\}}\right] - 1$. *Compute an estimate* $\tilde{p}_{t,i}$ *of* $p_{t,i}$ *by taking* $O(\lambda\kappa^2/\beta^2)$ *samples. The number of samples is chosen so that except with probability* $2^{-\lambda}$ *over the choice of samples,* $\|\tilde{p}_{t,i} - p_{t,i}\| < \beta/12(\kappa+1)$.
  - *For each* t *such that* $\|\tilde{p}_{t,0} - \tilde{p}_{t-1,0}\| > \beta/2(\kappa+1)\}$, *let* $\mathsf{id}_t \in \{0,1\}^\ell$ *be the string where* $\mathsf{id}_{t,i}$ *is set to 1 if and only if* $\|\tilde{p}_{t,i} - \tilde{p}_{t,0}\| < \|\tilde{p}_{t,i} - \tilde{p}_{t-1,0}\|$. *Let* A' *be the set of such* t.
  - *Output* $A = \{\mathsf{id}_t : t \in A'\}$.

To show Π_{TT} is $(\epsilon = \alpha, \delta = \gamma)$-threshold risky traceable, consider $\mathcal{A}$ producing decoder D.

Honest Users are Not Accused. Let $\mathsf{id}_t \in \{0,1\}^\ell$ denote the identity associated with index t (remember than in an index-based scheme the indices t are assumed to all be distinct). Let C' be the set of indices t, and C the set of identities id_t. Therefore, $\mathcal{A}$ sees secret keys for Π_{mcMFE} with attributes (t, id_t) for $t \in C'$. By considering ciphertexts with attribute $i = 0$ and by an identical analysis to the proof of Theorem 48 given in Section 5.1, we can conclude that $\Pr[A' \nsubseteq C'] \leq \mathsf{negl}(\lambda)$. We therefore just need to show that for any $t \in A'$, that the corresponding identity produced by Trace is exactly id_t.

Toward that end, fix a t, and assume $t \in A'$. $\mathcal{A}$ receives an mcMFE secret key for attribute (t, id_t), and all other secret keys seen by $\mathcal{A}$ have indices different than t. Consider some position $i \in [\ell]$. If $\mathsf{id}_{t,i} = 1$, then $R_\kappa((j,\mathsf{id}_j),(t,i)) = \mathbb{1}(j \leq t) = R_\kappa((j,\mathsf{id}_j),(t,0))$ for all secret key attributes (j,id_j) seen by $\mathcal{A}$. As such, in this case, by SEL-DEC-CTXT security, we have that except with negligible probability $\|p_{(t,i)} - p_{t,0}\| < \beta/12(\kappa+1)$. By the triangle inequality, except with negligible probability $\|\tilde{p}_{(t,i)} - \tilde{p}_{t,0}\| < 3\beta/12(\kappa+1) = \beta/4(\kappa+1)$. Meanwhile, since $\|\tilde{p}_{t,0} - \tilde{p}_{t-1,0}\| > \beta/2(\kappa+1)$, we have by the triangle inequality that $\|\tilde{p}_{t,i} - \tilde{p}_{t-1,0}\| > \|\tilde{p}_{t,0} - \tilde{p}_{t-1,0}\| - \|\tilde{p}_{t,i} - \tilde{p}_{t,0}\| > \beta/2(\kappa+1) - \beta/4(\kappa+1) = \beta/4(\kappa+1) > \|\tilde{p}_{(t,i)} - \tilde{p}_{t,0}\|$. Thus, Trace will claim $\mathsf{id}_{t,i} = 1$.

On the other hand, if $\mathsf{id}_{t,i} = 0$, then $R_\kappa((j,\mathsf{id}_j),(t,i)) = \mathbb{1}(\mathsf{id} \leq t-1) = R_\kappa((j,\mathsf{id}_j),(t-1,0))$ for all secret key attributes (j,id_j) seen by $\mathcal{A}$. As such, in this case, by SEL-DEC-CTXT security, we have that except with negligible probability $\|p_{(t,i)} - p_{t-1,0}\| < \beta/12(\kappa+1)$. By the triangle inequality, except with negligible probability $\|\tilde{p}_{(t,i)} - \tilde{p}_{t-1,0}\| < 3\beta/12(\kappa+1) = \beta/4(\kappa+1)$. Meanwhile, since $\|\tilde{p}_{t,0} - \tilde{p}_{t-1,0}\| > \beta/2(\kappa+1)$, we have by the triangle inequality that $\|\tilde{p}_{t,i} - \tilde{p}_{t,0}\| > \|\tilde{p}_{t,0} - \tilde{p}_{t-1,0}\| - \|\tilde{p}_{t,i} - \tilde{p}_{t-1,0}\| > \beta/2(\kappa+1) - \beta/4(\kappa+1) = \beta/4(\kappa+1) > \|\tilde{p}_{(t,i)} - \tilde{p}_{t-1,0}\|$. Thus, Trace will claim $\mathsf{id}_{t,i} = 0$. Over all indices $i > 0$, if $t \in A'$, Trace will therefore correctly put $\mathsf{id}_t \in A$.

Therefore, we have that $t \in A'$ if and only if $\mathsf{id}_j \in A$. Since the only j placed in A' are those corresponding to adversarial identities, we therefore have that $A \subseteq C$.

Some user is accused with reasonable probability. By considering ciphertexts with attribute $i = 0$ and by an identical analysis to the proof of Theorem 48 given in Section 5.1, we conclude that $\Pr[|A'| > 0] \geq \gamma \times \Pr[\mathsf{Good}_\alpha(\mathsf{D},\mathsf{mpk})] - \mathsf{negl}(\lambda)$. Since A is empty if and only if A' is empty, we therefore have that $\Pr[|A| > 0] \geq \gamma \times \Pr[\mathsf{Good}_\alpha(\mathsf{D},\mathsf{mpk})] - \mathsf{negl}(\lambda)$.

6 Instantiation Details

We now instantiate the needed ABE and PRF in order to apply Theorem 2 to obtain Theorem 1.

6.1 Log-depth Computation over Finite Fields

In this work, we will need that various operations over finite fields are computable in log-depth. This is well-known in prime-order fields or more generally the ring $\mathbb{Z}_N$ [BCH84]. However, we were not able to find the needed results for finite fields in the literature. So we prove them here:

Lemma 52. *For any finite field, iterated addition and iterated multiplication are computable in* NC^1.

Proof. For iterated addition over a finite field, we view the field as a vector space over the base prime-order field. We can then perform iterated addition component-wise in parallel. Iterated addition over the base field is then performed first over the integers, and then reduced mod the prime order of the field, both operations being log-depth [BCH84].

Iterated multiplication requires a bit more work. Let $\mathbb{F}$ be the field in question, and suppose $\mathbb{F}$ is not of prime order. Then it is an extension field of $\mathbb{Z}_p$ for some prime p. Let the degree of the extension be d. Then $\mathbb{F}$ is the set $\mathbb{Z}_p[X]/q(X)$ for some irreducible polynomial q of degree d. We will assume $p > D$ where $D := n(d-1)$ and where n is the number of field elements to be multiplied.

Consider the goal multiplying field elements/polynomials $r_1(X), \cdots, r_n(X)$. That is we are given the coefficients of these polynomials as $(\alpha_{i,j})_{i\in[n],j\in[0,d-1]}$ representing $r_i(X) = \sum_{j=0}^{d-1} \alpha_{i,j} X^j$. Our goal is to compute the coefficients $(\beta_j)_{j\in[0,d-1]}$ representing the polynomial $s(X) = \sum_{j=0}^{d-1} \beta_j X^j$ such that $s(X) = \prod_{i=1}^{n} r_i(X) \bmod q(X)$. We do this in log-depth as follows:

1. For $i = 1, \cdots, n$ and $u = 0, \cdots, D$, in parallel compute $r_i(u)$. This can be done in log-depth as follows. Pre-compute the matrix $\mathbf{M} \in \mathbb{Z}_p^{(D+1)\times(D+1)}$ defined as $\mathbf{M}_{u,j} = u^j$, where we take $0^0 = 1$. Then the vector $(r_i(u))_{u\in[0,D]}$ is just the matrix-vector product $\mathbf{M} \cdot (\alpha_{i,j})_{j\in[0,D]}$, where $\alpha_{i,j}$ for $j \geq d$ are taken to be 0. Since $\mathbf{M}$ is pre-computed, carrying out the matrix-vector product is simply an inner-product, which is one round of parallel integer multiplications followed by an iterated addition (to compute the product over $\mathbb{Z}$) followed by a modular reduction (to compute the value in $\mathbb{Z}_p$), which are all computable in log-depth [BCH84].

2. Define $s'(X) = \prod_{i=1}^{n} r_i(X)$, which is $s(X)$ without the reduction mod $q(X)$. This is a degree D polynomial. For $u = 0, \cdots, D$, in parallel compute $s'(u) = \prod_{i=1}^{n} r_i(u)$ using the values $r_i(u)$ computed in the previous step. This is an iterated integer multiplication (to compute the product over $\mathbb{Z}$) followed by a modular reduction (to compute the value in $\mathbb{Z}_p$), which are both computable in log-depth [BCH84].

3. Interpolate the values $s'(u)$ into the coefficients β'_k such that $s'(X) = \sum_{k=0}^{D} \beta'_k X^k$. This is computed as $(\beta'_k)_{k\in[0,D]} = \mathbf{M}^{-1} \cdot (s'(u))_{u\in[0,D]}$. By pre-computing $\mathbf{M}^{-1}$ ($\mathbf{M}$ is invertible since it is a square Vandermonde matrix), this is computable in log-depth analogous to Step 1.

4. Reduce $s'(X)$ mod $q(X)$ to obtain $s(X)$. This is done as follows. For $k \in [0, D]$, let $t_k(X) = X^k \bmod q(X)$, which we write as $t_k(X) = \sum_{j=0}^{d-1} \gamma_{k,j} X^j$. Pre-compute the matrix $\mathbf{N} \in \mathbb{Z}_p^{d\times(D+1)}$ be defined as $\mathbf{N}_{j,k} = \gamma_{k,j}$. Now output $(\beta_j)_{j\in[0,d-1]} = \mathbf{N} \cdot (\beta'_k)_{k\in[0,D]}$. Again, this matrix-vector product is computable in log-depth.

Extension to fields of small characteristic. The above required $p > D$. This is the standard setting arising in pairing-based cryptography, where p is typically exponential. Nevertheless, for completeness we here sketch how to remove this restriction. First, we observe that the above

algorithm is simply reducing iterated multiplication over $\mathbb{F}$ to iterated multiplication over a subfield. We then just need to identify a sub-field $\mathbb{F}'$ where (1) $|\mathbb{F}'| > D$ for interpolation, and (2) iterated multiplication in $\mathbb{F}'$ is computable by log-depth boolean formula. The case above used base field $\mathbb{Z}_p$ in the case $p > D$, but any other subfield satisfying (1) and (2) will do.

If $p \leq D$, let k be the smallest integer such that $p^k > D$; then $p^k \leq D^2$. The field $\mathbb{F}_{p^k}$ of size p^k satisfies (1), and we can compute iterated products as follows, satisfying (2):

- If any of the inputs r_i are 0, output 0 and stop. From now on we assume $r_i \neq 0$.
- Let $u \in \mathbb{F}_{p^k}$ be a generator of the multiplicative group of $\mathbb{F}_{p^k}$, which can be pre-computed.
- In parallel for each r_i, compute s_i such that $r_i = u^{s_i}$. This can be done in log depth via pre-computed tables, since the field is polynomial-sized.
- Iteratively add the S_i and reduce mod $p^k - 1$ (the order of the multiplicative group of $\mathbb{F}_{p^k}$), which are both computable in log-depth [BCH84]. Let the result be s.
- Output $u^s \in \mathbb{F}_{p^k}$, again using pre-computed tables.

Now, $\mathbb{F}_{p^k}$ may not be a sub-field of $\mathbb{F}$. But the extension $\mathbb{F}''$ of degree k over $\mathbb{F}$ does have $\mathbb{F}_{p^k}$ as a sub-field. Therefore, to compute an iterated product over $\mathbb{F}$, we compute it over $\mathbb{F}''$, which reduces to computing it over $\mathbb{F}' = \mathbb{F}_{p^k}$, which satisfies the necessary conditions (1) and (2). □

6.2 Log-Depth Weak PRFs

Theorem 53. *Assume K-LIN holds in the multiplicative group of a finite field $\mathbb{F}$. Then there exists a weak PRF with outputs in $\mathbb{F}$ that is computable in NC^1 using pre-computation.*

Proof. Note that for $K = 1$, K-LIN is just DDH, and this case from [NR97], which actually achieves a strong PRF and requires no pre-computation. While [NR97] can be generalized to be secure under K-LIN, the resulting construction [LW09, EHK+13] does not appear to be computable in log-depth. We therefore present a different construction, which is based on ElGamal encryption [ElG84], and its generalizations to K-LIN for $K > 1$. The key is a vector $\mathbf{k} \in \mathbb{Z}_p^K$. Then PRF takes as input vectors $\mathbf{r} \in \mathbb{F}^k$, and computes $\prod_{i=1}^K r_i^{k_i}$, which we will denote as $\mathbf{r}^{\mathbf{k}}$. For computing in log-depth, we let $h_{\mathbf{r}} = (\mathbf{r}, \mathbf{r}^2, \cdots, \mathbf{r}^{2^t})$ where exponentiation is component-wise. Then the bits of $\mathbf{k}$ indicate which of the components of $h_{\mathbf{r}}$ need to be multiplied together.

For security, consider being given many samples of the weak PRF: $(\mathbf{r}_i, \mathbf{r}_i^{\mathbf{k}})$ for $i \in [\ell]$. If we let $\mathbf{r}_i = g^{\mathbf{s}_i}$ where $\mathbf{s}_i \in \mathbb{Z}_p^K$ and exponentiation is component-size, then the samples look like $(g^{\mathbf{s}_i}, g^{\mathbf{s}_i \cdot \mathbf{k}})$. By arranging these samples into a matrix, we have that the input to the adversary is $g^{\mathbf{A}}$ for matrix $\mathbf{A} = (\mathbf{S} \mid \mathbf{S} \cdot \mathbf{k})$. $\mathbf{A}$ is a random matrix in $\mathbb{Z}_p^{\ell \times (K+1)}$ with rank K. As shown by [EHK+13], $g^{\mathbf{A}}$ is indistinguishable from a random matrix with no rank constraints, under K-LIN. But this case corresponds to replacing the PRF samples with uniformly random values. □

6.3 ABE for Arithmetic Formula

Theorem 54 ([IW14, CGW15, LL20])**.** *Assume either (1) there exists a pairing group (symmetric or asymmetric) where the K-LIN assumption holds for some K, or (2) there exists a symmetric pairing group where the DBDH assumption holds. Then there exists a family of prime-order fields $\{\mathbb{Z}_q\}$ and an attribute-based encryption scheme for $\mathsf{NC}^1_{\neq 0}(\{\mathbb{Z}_q\})$.*

Note that ABE systems, such as those referenced in Theorem 54, are usually either key-policy or ciphertext policy, where $R(x, y) = x(y)$ or $R(x, y) = y(x)$, interpreting either x or y as a function of the other attribute. The functions x or y are then called policies instead of attributes. Both key-policy and ciphertext policy ABE schemes for log-depth arithmetic formula yield an ABE as in Definition 12, where R itself is computing the arithmetic formula. For example, by generating keys with policy $R(y, \cdot)$ with y hardcoded, we obtain the needed ABE from key-policy ABE.

6.4 Putting It All Together

First, we observe that K-LIN in a pairing group implies K-LIN in the target group of the pairing, which is a subgroup of the multiplicative group of some finite field $\mathbb{F}$. Likewise, DBDH in the pairing implies DDH (equivalently, 1-LIN) in the target group. Then applying Theorem 53, we obtain the needed PRF with log-depth boolean formula under pre-computation. These boolean formulas can then be "arithmetic-ized" into log-depth arithmetic formula over any field. In particular, we can use the field $\mathbb{Z}_q$ arising from Theorem 54. The result is a weak PRF appropriate for combining with Theorem 54 which can then be plugged into Theorem 2, giving Theorem 1.

References

[ABG+14] Adi Akavia, Andrej Bogdanov, Siyao Guo, Akshay Kamath, and Alon Rosen. Candidate weak pseudorandom functions in $\mathsf{AC}^0 \circ \mathrm{MOD}_2$. In Moni Naor, editor, *ITCS 2014*, pages 251–260. ACM, January 2014.

[All04] Eric Allender. Arithmetic circuits and counting complexity classes. *Complexity of Computations and Proofs, Quaderni di Matematica*, 13:33–72, 2004.

[BCH84] Paul Beame, Stephen A. Cook, and H. James Hoover. Log depth circuits for division and related problems. In *25th FOCS*, pages 1–6. IEEE Computer Society Press, October 1984.

[BN08] Dan Boneh and Moni Naor. Traitor tracing with constant size ciphertext. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 2008*, pages 501–510. ACM Press, October 2008.

[BOC92] Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM Journal on Computing*, 21(1):54–58, 1992.

[BP08] Olivier Billet and Duong Hieu Phan. Efficient traitor tracing from collusion secure codes. In Reihaneh Safavi-Naini, editor, *ICITS 08*, volume 5155 of *LNCS*, pages 171–182. Springer, Heidelberg, August 2008.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737. Springer, Heidelberg, April 2012.

[BSW06] 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, Heidelberg, May / June 2006.

[BW06] 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 2006*, pages 211–220. ACM Press, October / November 2006.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 480–499. Springer, Heidelberg, August 2014.

[BZ16] Mark Bun and Mark Zhandry. Order-revealing encryption and the hardness of private learning. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 176–206. Springer, Heidelberg, January 2016.

[CFN94] Benny Chor, Amos Fiat, and Moni Naor. Tracing traitors. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 257–270. Springer, Heidelberg, August 1994.

[CGW15] Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Heidelberg, April 2015.

[CVW+18] Yilei Chen, Vinod Vaikuntanathan, Brent Waters, Hoeteck Wee, and Daniel Wichs. Traitor-tracing from LWE made simple and attribute-based. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 341–369. Springer, Heidelberg, November 2018.

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013.

[ElG84] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer, Heidelberg, August 1984.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[GKRW18] Rishab Goyal, Venkata Koppula, Andrew Russell, and Brent Waters. Risky traitor tracing and new differential privacy negative results. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 467–497. Springer, Heidelberg, August 2018.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th FOCS*, pages 612–621. IEEE Computer Society Press, October 2017.

[GKW18] Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 660–670. ACM Press, June 2018.

[GKW19] Rishab Goyal, Venkata Koppula, and Brent Waters. New approaches to traitor tracing with embedded identities. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 149–179. Springer, Heidelberg, December 2019.

[GLW23] Junqing Gong, Ji Luo, and Hoeteck Wee. Traitor tracing with $N^{1/3}$-size ciphertexts and $O(1)$-size keys from k-Lin. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 637–668. Springer, Heidelberg, April 2023.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 162–179. Springer, Heidelberg, August 2012.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013.

[GVW19] Rishab Goyal, Satyanarayana Vusirikala, and Brent Waters. Collusion resistant broadcast and trace from positional witness encryption. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 3–33. Springer, Heidelberg, April 2019.

[IW14] Yuval Ishai and Hoeteck Wee. Partial garbling schemes and their applications. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *ICALP 2014, Part I*, volume 8572 of *LNCS*, pages 650–662. Springer, Heidelberg, July 2014.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 60–73. ACM Press, June 2021.

[JLS22] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 670–699. Springer, Heidelberg, May / June 2022.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *20th ACM STOC*, pages 20–31. ACM Press, May 1988.

[KMUW18] Lucas Kowalczyk, Tal Malkin, Jonathan Ullman, and Daniel Wichs. Hardness of non-interactive differential privacy from one-way functions. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 437–466. Springer, Heidelberg, August 2018.

[KMUZ16] Lucas Kowalczyk, Tal Malkin, Jonathan Ullman, and Mark Zhandry. Strong hardness of privacy from weak traitor tracing. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part I*, volume 9985 of *LNCS*, pages 659–689. Springer, Heidelberg, October / November 2016.

[LL20] Huijia Lin and Ji Luo. Compact adaptively secure ABE from k-Lin: Beyond NC^1 and towards NL. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 247–277. Springer, Heidelberg, May 2020.

[LW09] Allison B. Lewko and Brent Waters. Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM CCS 2009*, pages 112–120. ACM Press, November 2009.

[NP98] Moni Naor and Benny Pinkas. Threshold traitor tracing. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 502–517. Springer, Heidelberg, August 1998.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997.

[NRR00] Moni Naor, Omer Reingold, and Alon Rosen. Pseudo-random functions and factoring (extended abstract). In *32nd ACM STOC*, pages 11–20. ACM Press, May 2000.

[NWZ16] Ryo Nishimaki, Daniel Wichs, and Mark Zhandry. Anonymous traitor tracing: How to embed arbitrary information in a key. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 388–419. Springer, Heidelberg, May 2016.

[SS10] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS 2010*, pages 463–472. ACM Press, October 2010.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th FOCS*, pages 600–611. IEEE Computer Society Press, October 2017.

[Zha20] Mark Zhandry. New techniques for traitor tracing: Size $N^{1/3}$ and more from pairings. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 652–682. Springer, Heidelberg, August 2020.

[Zha21] Mark Zhandry. White box traitor tracing. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 303–333, Virtual Event, August 2021. Springer, Heidelberg.