Message-Locked Encryption for Lock-Dependent Messages

Martín Abadi[1,3], Dan Boneh[2,⋆], Ilya Mironov[1],
Ananth Raghunathan[2,⋆,⋆⋆], and Gil Segev[2,⋆,⋆⋆]

[1] Microsoft Research Silicon Valley
[2] Stanford University
[3] University of California, Santa Cruz

Abstract. Motivated by the problem of avoiding duplication in storage systems, Bellare, Keelveedhi, and Ristenpart have recently put forward the notion of Message-Locked Encryption (MLE) schemes which subsumes *convergent encryption* and its variants. Such schemes do not rely on permanent secret keys, but rather encrypt messages using keys derived from the messages themselves.

We strengthen the notions of security proposed by Bellare et al. by considering plaintext distributions that may depend on the public parameters of the schemes. We refer to such inputs as *lock-dependent* messages. We construct two schemes that satisfy our new notions of security for message-locked encryption with lock-dependent messages.

Our main construction deviates from the approach of Bellare et al. by avoiding the use of ciphertext components derived deterministically from the messages. We design a fully randomized scheme that supports an equality-testing algorithm defined on the ciphertexts.

Our second construction has a deterministic ciphertext component that enables more efficient equality testing. Security for lock-dependent messages still holds under computational assumptions on the message distributions produced by the attacker.

In both of our schemes the overhead in the length of the ciphertext is only additive and independent of the message length.

Keywords: Deduplication, convergent encryption, message-locked encryption

1 Introduction

Deduplication, which eliminates redundant copies in user-provided data, is an important space-saving technique in communications and storage (see, for ex-

⋆ This work was supported by NSF, the DARPA PROCEED program, an AFOSR MURI award, a grant from ONR, an IARPA project provided via DoI/NBC, and by Samsung. Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA or IARPA. Distrib. Statement "A:" Approved for Public Release, Distribution Unlimited.

⋆⋆ Part of the work was done at Microsoft Research Silicon Valley.

ample, [28,35,26]). Storage systems that rely on deduplication typically let the server have unfettered access to the clients' data. This set-up creates an obvious confidentiality problem, since the clients must trust the server with not only storing their documents but keeping them secret too.

The first solution for balancing confidentiality and efficiency in deduplication was described by Douceur et al. [18] and called *convergent encryption*. According to this deterministic scheme, a message is encrypted under a message-derived key (a hash of the message) so that identical plaintexts are mapped to identical ciphertexts. After encrypting the message, the client uploads the ciphertext to the server, retaining the hash to allow later decryption. In the meantime, the server can recognize equal ciphertexts, storing only one copy of each: if two clients happen to upload the same file, the resulting ciphertexts will be identical and can be deduplicated. The clients need not coordinate their actions and might not even be aware of each other's existence. Implementations and variants of convergent encryption followed [3,14,25,30,33,2,1] but their precise security guarantees were never fully proven or even stated.

Message-locked encryption. Recently, Bellare, Keelveedhi, and Ristenpart [7] brought much needed rigor into the area, by defining a new encryption primitive, Message-Locked Encryption (MLE), and several definitions that capture various aspects of MLE security. They also constructed and analyzed several schemes in their framework.

We briefly recall the definition of MLE and two security notions of privacy and tag integrity for MLE schemes introduced by Bellare et al. An MLE scheme encapsulates a standard (possibly randomized) symmetric-key encryption scheme where the encryption algorithm accepts a message m and a key k, and outputs a ciphertext c. The decryption algorithm reverses the process, recovering m from c given k. The scheme comes with a *key derivation algorithm* that, unlike a conventional key generation algorithm, is a deterministic function from m to k. It also includes a *tag-generation algorithm* that maps the ciphertext to a tag. Identical plaintexts result in equal tags. The corresponding ciphertexts, which may be randomized, are not necessarily equal. Tag integrity means that no computationally bounded adversary can trick the server into replacing a valid encryption with a ciphertext that does not decrypt to the same plaintext.

It is apparent that MLE, with its *deterministic* tag, cannot satisfy the standard notions of confidentiality (such as semantic security). Indeed, if the plaintexts can be feasibly enumerated, the adversary may always compute their tags and test them against that of the challenge ciphertext. A meaningful security guarantee can be achieved only if the input is sufficiently unpredictable. More concretely, in a CDA game (a chosen-*distribution* attack) the challenge consists either of a uniformly distributed string of bits or an encryption of a message drawn randomly from a distribution provided by the adversary. The security level is characterized by the distinguisher's running time, its advantage over a random guess, and the min-entropy of the distribution that the adversary is allowed to specify. A lower min-entropy requirement corresponds to a stronger security guarantee. This approach—basing security of the scheme on the assumption of unpredictability of the plaintexts—is similar to the theory of deterministic public-key encryption initiated by the work of Bellare, Boldyreva, and O'Neill [4] (see also [6,10,5,12,19,24,32,29]).

Lock-dependent messages. In addition to the min-entropy requirement, there is another constraint on the adversarially chosen distribution of plaintexts implicit in the definition of MLE. If the adversary is allowed to specify a distribution of plaintexts, it may use the fact that the tags are deterministic for leaking unnecessary information on the messages (e.g., select a distribution that is concentrated on messages whose tags share a particular property, such as that they all start with a zero bit, or that the first bit of the tag reveals the first bit of the message). Doing so immediately gives the adversary a constant advantage in answering the challenge (of whether the output was a random string of bits or an encryption of a message drawn from the distribution). Similar attacks can be effective against *any* deterministic encryption scheme, where the adversary tailors the distribution to the scheme's public key. The common way of sidestepping this difficulty is to require that the distribution be chosen independently of the system parameters or, in the case of deterministic encryption, of the system's public key. More formally, the adversary must commit to the distribution of plaintexts before accessing the description of the system.

Since the parameters of the scheme are supposed to be publicly available, they must be included into the view of any realistic adversary. As soon as the adversary learns the parameters of the system and may influence, however indirectly, the distribution of plaintexts, the assumption of independence becomes false, voiding the security guarantees proven under this assumption.

In this paper we ask whether security guarantees can encompass also attacks that may depend on the scheme's parameters. Identifying the public parameters of an encryption scheme with a *lock*, we can paraphrase the problem addressed in this paper as follows:

Can message-locked encryption be secure for lock-dependent messages?

1.1 Our Contributions

In this paper we put forward two approaches for resolving this question in the affirmative, and provide schemes that are secure even for lock-dependent messages in realistic and rigorously defined adversarial models.

Our first approach is to avoid using tags that are derived deterministically from the messages. To this end, we design a fully randomized scheme that supports an equality-testing algorithm defined on the ciphertexts. We show that this enables us to satisfy a strong definition of security for an extension of the MLE notion, allowing the adversary to specify the distribution of the plaintexts adaptively, with no further restrictions on the distribution other than its min-entropy. Our construction is based on standard cryptographic tools in the random oracle model [8] and on a natural variant of Canetti's entropy-based DDH assumption [13]. The ciphertext overhead is only additive and polynomial in the security parameter.

Our second approach, on the other hand, continues using deterministic tags. Security for lock-dependent messages is guaranteed by limiting the computational power of the adversarial message distributions. (This approach is inspired by the recent work of Raghunathan, Segev, and Vadhan [29] who proposed a similar adversarial model for deterministic public-key encryption.) Specifically, in the random oracle model, we consider adversaries that are allowed to choose the distribution of plaintexts adaptively, after seeing the scheme's parameters, subject to the condition that the distribution be efficiently samplable using at most q queries to the random oracle, where q is a pre-determined parameter. Our construction can be based on any semantically secure encryption scheme. Its overhead, defined as the increase in the length of the ciphertext, is additive and depends only on the security parameter.

1.2 Paper Organization

In Section 2, we give a high-level overview of the fully randomized scheme and the deterministic scheme that we construct in this paper. In Section 3, we introduce a few preliminaries required to present our results. In Section 4, we formally define our notion of message-locked encryption for lock-dependent messages. In Section 5, we present the fully randomized scheme. In Section 7, we conclude and mention several interesting directions for further research. Because of space limitations all proofs and some definitions are deferred to the full version.

2 Overview of Our Schemes

In what follows we provide a high-level overview of the main ideas that underlie our schemes. Intuitively, constructing MLE schemes requires solving two technical challenges. We must design an algorithm that encrypts messages under a key that is highly correlated (via the key derivation algorithm) with the message and still remains secure. Secondly, the part of the ciphertext that allows the equality test must not leak any information about messages sampled from an adversarially chosen min-entropy distribution even given the public parameters.

Construction 1: A fully randomized scheme. An encryption of a message m in our first scheme consists of three components: a "payload" which is an encryption of m using some underlying randomized encryption scheme, a tag, and a proof of consistency showing that the payload and the tag correspond to the same message. A tag for a message m is computed as $\tau = \left(g^r, g^{r \cdot h(m)}\right)$, where g is a generator of a bilinear group, h is a sufficiently strong collision-resistant function, and r is chosen uniformly at random. Given two tags $\tau_1 = (g_1, h_1)$ and $\tau_2 = (g_2, h_2)$, the equality-testing algorithm computes the pairings $\hat{e}(g_1, h_2)$ and $\hat{e}(g_2, h_1)$, which match if the tags were derived from the same message (or if a non-trivial collision was found for h). The fact that tags do not reveal any more information than is necessary for the scheme's functionality is based on combining a variant of Canetti's entropy-based DDH assumption [13], and the concept of seed-dependent condensers, recently introduced by Dodis, Ristenpart, and Vadhan [17]. A similar idea for equality testing (without hashing) was explored by Yang et al. [34], who designed public-key encryption schemes that support equality testing but offer a significantly weaker notion of security (only one-wayness).

As for the payload and the consistency proof, a natural approach would be to simply encrypt m using its hash $h(m)$ as a key (as in [7]), and provide a NIZK proof of consistency. This approach, however, seems to fail as we must use an encryption scheme for which it is secure to encrypt a message m under the key $h(m)$. All existing constructions satisfying this property rely on the random oracle paradigm, which rules out using NIZK proofs as the language under consideration is no longer in NP.

We can resolve this issue with a cut-and-choose protocol applied to the encryption of the message. Naïvely, such a protocol would inflate the size of the ciphertext. However, a delicate combination of a secret-sharing scheme and a cut-and-choose technique enables us to realize an encryption scheme with a ciphertext overhead that is only additive and independent of the message length.

Specifically, the payload in our ciphertext consists of a randomized encryption $\mathsf{E}_s(m; r_1)$ of m under a uniformly chosen key s, a commitment $\mathsf{Commit}(s\|t)$ to s and a uniformly chosen key t, an ElGamal encryption $\left(g^{r_2}, g^{r_2 \cdot h(m)} \cdot t\right)$ of t using $h(m)$ as a key, and a circular-secure encryption $(r_3, H(r_3\|t) + s)$ of s using t as a key. The circular-secure encryption scheme is due to Black, Rogaway, and Shrimpton [9] whose proof of security assumes a random oracle H.

The only component that requires a random oracle is the circular-secure encryption of s using t. We can use a NIZK proof for proving that all other components (including the tag) are consistent with the same message m. In addition, we use a cut-and-choose protocol (which we collapse using a random oracle to a non-interactive one) for showing that the commitment $\mathsf{Commit}(s\|t)$ is consistent with the circular-secure encryption $(r_3, H(r_3\|t) + s)$, where s is encoded with a threshold secret-sharing scheme. The commitment $\mathsf{Commit}(s\|t)$ is used in both the NIZK proof and in the cut-and-choose components, and binds the two together to yield a proof of consistency for the entire ciphertext.

To ensure that the overhead of the scheme is additive and independent of the length of the message, first observe that the length of the commitment and the encryption of s under t (and hence the cut-and-choose part of the scheme) depend only on the security parameter. To further minimize the length of ciphertexts, we use a composition of an NIZK proof system with a succinct argument system in the random oracle model, where the length of the arguments depends only on the security parameter.

Construction 2: Deterministic encryption for computationally bounded distributions. As in the previous work [18,7], our second scheme uses any semantically secure randomized encryption $\mathsf{E}_k(m; r)$. It encrypts a message m using a key $k = k_m$ and randomness $r = r_m$ that are derived from m in a deterministic manner (e.g., using a hash function modeled as a random oracle). With lock-dependent message distributions, however, such a scheme does not satisfy a meaningful notion of security since it is completely deterministic (as discussed above).

Following Raghunathan et al. [29] we show that this approach can be made secure even for lock-dependent message distributions, subject to the condition that these distributions are efficiently samplable using at most q queries to the random oracle, where q is a pre-determined parameter. (We do not ask for an a priori bound on the number of oracle calls that are made directly by the adversary.) Concretely, we derive the key k_m and the randomness r_m as $k_m = \oplus_{i=1}^{q+1} H_1(m\|i)$ and $r_m = \oplus_{i=1}^{q+1} H_2(m\|i)$, where H_1 and H_2 are two hash functions modeled as independent random oracles.

Intuitively, the proof of security relies on the fact that k_m and r_m are pseudorandom against *both* the adversary and the sampling circuits of such "q-bounded" message distributions. Pseudorandomness against the adversary relies on the fact that m is sampled with a super-logarithmic min-entropy and that the underlying encryption scheme is secure. Pseudorandomness against the sampling circuits relies on the fact that for learning any information on k_m or r_m it is essential to query the random oracle $q+1$ times.

3 Preliminaries

Notation. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$, by $[a, b]$ the set $\{a, a+1, \ldots, b\}$, and by U_n the uniform distribution over the set $\{0,1\}^n$. For a random variable X we denote by $x \leftarrow X$ the process of sampling a value x according to the distribution of X. Similarly, for a finite set S we denote by $x \leftarrow S$ the process of sampling a value x according to the uniform distribution over S. We denote by $\mathbf{x}$ (and sometimes $\boldsymbol{x}$) a vector $(x_1, \ldots, x_{|\mathbf{x}|})$. We denote by $\mathbf{X} = (X_1, \ldots, X_T)$ a joint distribution of T random variables, and by $\mathbf{x} = (x_1, \ldots, x_T)$ a sample drawn from $\mathbf{X}$. For two bit-strings x and y we denote by $x\|y$ their concatenation. A non-negative function $f\colon \mathbb{N} \to \mathbb{R}$ is negligible if it vanishes faster than any inverse polynomial.

Entropy. The *min-entropy* of a random variable X is defined as $\mathbf{H}_\infty(X) = -\log(\max_x \Pr[X = x])$. A *$k$-source* is a random variable X with $\mathbf{H}_\infty(X) \geq k$. A *$(k_1, \ldots, k_T)$-source* is a random variable $\mathbf{X} = (X_1, \ldots, X_T)$ where each X_i is a k_i-source. A *(T, k)-source* is a random variable $\mathbf{X} = (X_1, \ldots, X_T)$ where for each $i \in [T]$, it holds that X_i is a k-source. A *(T, k)-block-source* is a random variable $\mathbf{X} = (X_1, \ldots, X_T)$ where for every $i \in [T]$ and $x_1, \ldots, x_{i-1}$ it holds that $X_i|_{X_1=x_1,\ldots,X_{i-1}=x_{i-1}}$ is a k-source. The *statistical distance* between two random variables X and Y over a finite domain Ω is $\mathbf{SD}(X, Y) = \frac{1}{2}\sum_{\omega\in\Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$.

The ME-DDH assumption. We state a variant of Canetti's entropy DDH assumption [13]. The β-min-entropy DDH assumption (abbreviated as ME-DDH) states that for a group $\mathbb{G}$ equipped with a non-degenerate bilinear map $\hat{e}\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\mathrm{T}$, of prime order p (where p is a λ-bit prime) for any distribution X over $\mathbb{Z}_p$ with $\mathbf{H}_\infty(X) \geq \beta$, for uniformly sampled $a, c \leftarrow \mathbb{Z}_p$ and $b \leftarrow X$, it holds that the two distributions $\left(g, g^a, g^{ab}\right)$ and (g, g^a, g^c) are computationally indistinguishable. We make two remarks on the ME-DDH assumption:

1. We require $\beta \geq \omega(\log \lambda)$ for the assumption to be plausible. Otherwise, there exists an $x^* \leftarrow X$ such that $\Pr\left[X = x^*\right]$ is non-negligible and a distinguisher that, when given (g, g^a, g^c), checks to see whether $(g^a)^{x^*} = g^c$, succeeds in distinguishing the two distributions with non-negligible probability.
2. If X is the uniform distribution over $\mathbb{Z}_p$, then the assumption is *unconditionally true* as the two distributions ab and c are identical even given a.

4 MLE for Lock-Dependent Messages

Extending the work of Bellare et al. [7], we propose a more general notion of the primitive MLE, which we call MLE2. In MLE2, we allow tags to be randomized and consider a definition of tag correctness that introduces a new polynomial-time algorithm EQ that subsumes the functionality of deterministic tags. In addition, we introduce a new *validity-test* algorithm, denoted Valid, that allows anyone with the public parameters to check if a given ciphertext is a valid ciphertext. In the context of using MLE2 for secure deduplication, EQ allows for deduplication of ciphertexts and Valid allows the server to reject adversarially constructed ciphertexts that subvert deduplication to replace a valid ciphertext with an invalid one that does not decrypt correctly.

The main benefit of the new notion is permitting a stronger security requirement (which we denote $\mathsf{PRV\text{-}CDA2}$) that allows the adversary to see the public parameters *before* issuing oracle queries.

MLE2. A message-locked encryption scheme for lock-dependent messages is a six-tuple $\Pi = (\mathsf{PPGen}, \mathsf{KD}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{EQ}, \mathsf{Valid})$ operating over plaintext space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$, ciphertext space $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$, and keyspace $\mathcal{K} = \{\mathcal{K}_\lambda\}_{\lambda\in\mathbb{N}}$ of polynomial-time randomized algorithms with the following properties:

- The parameter generation algorithm takes as input 1^λ and returns public parameters pp.
- The key derivation function KD takes as input public parameters pp, a message m, and outputs a message-derived key $k_m \leftarrow \mathsf{KD}_{\mathsf{pp}}(m)$.
- The encryption algorithm Enc takes as input public parameters pp, a message m, and a message-derived key k_m. It outputs a ciphertext $c \leftarrow \mathsf{Enc}_{\mathsf{pp}}(k_m, m)$.
- The decryption algorithm Dec takes as input public parameters pp, ciphertext c, and a secret key k and outputs either a message m or $\perp$.
- The (new) equality algorithm EQ takes as input public parameters pp, and two ciphertexts c_1 and c_2 and outputs 1 if both ciphertexts are generated from the same underlying message.
- The (new) validity-test algorithm Valid takes as input public parameters pp and a ciphertext c and outputs 1 if the ciphertext c is a valid ciphertext.

Bellare et al. [7] considered the notion of an equality-checking *tag* analogous to our notion of an equality algorithm EQ. A (publicly computable) tag-generation algorithm, on input a ciphertext c, produces a tag such that if two ciphertexts c_1 and c_2 are generated from the same message, the corresponding tags are equal with high probability. Our equality algorithm is a generalization of such an equality-checking tag. Given any scheme with equality-checking tags, we can describe a simple algorithm EQ that given c_1 and c_2 derives their respective tags and outputs 1 only if the tags are equal.

The notion analogous to *tag correctness* of Bellare et al. [7] requires that for all $\lambda \in \mathbb{N}$, all public parameters $\mathsf{pp} \leftarrow \mathsf{PPGen}(1^\lambda)$, and all messages $m \in \mathcal{M}$, there is a negligible function $\nu(\lambda)$ such that for two encryptions c_1 and c_2 of m with $\mathsf{KD}_{\mathsf{pp}}(m)$ and independent random coins, it holds that $\mathsf{EQ}_{\mathsf{pp}}(c_1, c_2) = 1$ with probability at least $1 - \nu(\lambda)$, where the probability is taken over random coins of all algorithms.

The notion of correctness for the validity-test algorithm Valid requires that for all $\lambda \in \mathbb{N}$, all public parameters $\mathsf{pp} \leftarrow \mathsf{PPGen}(1^\lambda)$, and all messages $m \in \mathcal{M}$, there is a negligible function $\nu(\lambda)$ such that for a ciphertext $c \leftarrow \mathsf{Enc}_{\mathsf{pp}}\left(\mathsf{KD}_{\mathsf{pp}}(m), m\right)$, $\Pr\left[\mathsf{Valid}_{\mathsf{pp}}(c) = 1\right] \geq 1 - \nu(\lambda)$, where the probability is taken over all random coins of all algorithms.

The usual notion of correctness of the decryption algorithm Dec applies. Specifically, for all $\lambda \in \mathbb{N}$, all public parameters $\mathsf{pp} \leftarrow \mathsf{PPGen}(1^\lambda)$, and all messages $m \in \mathcal{M}$, there is a negligible function $\nu(\lambda)$ such that

$$\Pr[\mathsf{Dec}_{\mathsf{pp}}(k_m, \mathsf{Enc}_{\mathsf{pp}}(k_m, m)) = m \mid k_m \leftarrow \mathsf{KD}_{\mathsf{pp}}(m)] \geq 1 - \nu(\lambda),$$

where the probability is taken over all random coins of all algorithms.

MLE2 adversaries. To capture a notion of security against an adversary that attacks the system by choosing messages that may depend on the public parameters, we introduce several adversary models. In what follows, we consider several parameters that are functions of the security parameter; $q = q(\lambda)$ denoting the number of random oracle queries, $k = k(\lambda)$ denoting min-entropy requirements over message sources, $T = T(\lambda)$ denoting the number of blocks in the message source, and $\varGamma = \varGamma(\lambda)$ denoting the size of a circuit that generates message sources.

In particular, inspired by recent work on deterministic encryption [29], for $\mathsf{X} \in \{(T, k)\text{-block}, (T, k)\}$ we define the class of $\varGamma$-sampling complexity X-source adversaries and a generalization to polynomial-size X-source adversaries. We stress that all algorithms are allowed polynomially many calls to the random oracle in the security definitions that follow. Additionally, in schemes that rely random oracles, we define q-query X-source adversaries. Although more restrictive, they are useful in constructing efficient and practical deterministic encryption schemes secure in the random oracle model [29]. We begin by introducing a definition of the real-or-random encryption oracle used in definitions of security.

Definition 4.1 (Real-or-random encryption oracle). *The real-or-random encryption oracle,* RoR*, takes as input triplets of the form* $(\mathsf{mode}, \mathsf{pp}, \mathbf{M})$*, where* $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$*,* pp *denotes public parameters, and* $\mathbf{M}$ *is a polynomial size circuit representing a joint distribution over* T *messages. If* $\mathsf{mode} = \mathsf{real}$ *then the oracle samples* $(m_1, \ldots, m_T) \leftarrow \mathbf{M}$*, and if* $\mathsf{mode} = \mathsf{rand}$ *then the oracle samples* *uniform and independent messages* $m_1, \dots, m_T \leftarrow \mathcal{M}$*. Next, for each* $i \in [T]$*, it samples* $k_i \leftarrow \mathsf{KD}_{\mathsf{pp}}(m_i)$*, computes* $c_i \leftarrow \mathsf{Enc}_{\mathsf{pp}}(k_i, m_i)$ *and outputs the ciphertext vector* $(c_1, \dots, c_T)$*.*

Definition 4.2 (*Γ*-sampling complexity adversary). *Consider an* X*-source where* $\mathsf{X} \in \{(T,k)\text{-}block, (T,k)\}$*. Let* $\mathcal{A}$ *be a probabilistic polynomial-time algorithm that is given as input a pair* $(1^\lambda, \mathsf{pp})$ *and oracle access to* $\mathsf{RoR}(\mathsf{mode}, \mathsf{pp}, \cdot)$ *for some* $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$*. Then,* $\mathcal{A}$ *is a* Γ-sampling complexity X-source adversary *if for each of* $\mathcal{A}$*'s* RoR *queries* $\mathbf{M}$ *it holds that* $\mathbf{M}$ *is an* X*-source that is samplable by a circuit of size at most* Γ*. In addition, for the case of* (T,k)*-source adversaries, we require that for each such query* $\mathbf{M}$ *it holds that* $M_i \neq M_j$ *for all vectors* $(M_1, \dots, M_T)$ *in the support of* $\mathbf{M}$ *and for all* $i \neq j \in [T]$*.*

We consider a stronger adversary that has no *a-priori* bound on the sampling complexity of its queries except that they are efficiently samplable by polynomial size circuits. Such an adversary subsumes Γ-sampling complexity adversaries for all $\Gamma = \text{poly}(\lambda)$.

Definition 4.3 (Polynomial-sampling complexity adversary). *Let* $\mathsf{X} \in \{(T,k)\text{-}block, (T,k)\}$*, and let* $\mathcal{A}$ *be a probabilistic polynomial-time algorithm that is given as input a pair* $(1^\lambda, \mathsf{pp})$ *and oracle access to* $\mathsf{RoR}(\mathsf{mode}, \mathsf{pp}, \cdot)$ *for some* $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$*. Then,* $\mathcal{A}$ *is a* polynomial-size X-source adversary *if for each of* $\mathcal{A}$*'s* RoR*-queries* $\mathbf{M}$ *it holds that* $\mathbf{M}$ *is an* X*-source that is samplable by a circuit of (an arbitrary) polynomial size in the security parameter.*

Definition 4.4 (*q*-query adversary [29]). *Consider an* X*-source where* $\mathsf{X} \in \{(T,k)\text{-}block, (T,k)\}$*. Let* $\mathcal{A}$ *be a probabilistic polynomial-time algorithm that is given as input a pair* $(1^\lambda, \mathsf{pp})$ *and oracle access to* $\mathsf{RoR}(\mathsf{mode}, \mathsf{pp}, \cdot)$ *for some* $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$*. Then,* $\mathcal{A}$ *is a* q-query k-source adversary *if for each of* $\mathcal{A}$*'s* RoR*-queries* $\mathbf{M}$ *it holds that* $\mathbf{M}$ *is an* X*-source that is samplable by a polynomial-size circuit that uses at most* q *queries to the random oracle.*

A stronger notion of message privacy: PRV-CDA2. We define the following security notion with respect to polynomial-size X-source adversaries (see Definition 4.3), which we denote X-source $\mathsf{PRV\text{-}CDA2}$ security. A simple modification to the experiments in the security definition allows us to restrict our class of adversaries to Γ-sampling complexity or q-query X-source adversaries. Such notions are referred to as Γ-sampling complexity or q-query X-source $\mathsf{PRV\text{-}CDA2}$.

Definition 4.5 (PRV-CDA2 security). *An MLE2 scheme* $\Pi = (\mathsf{PPGen}, \mathsf{KD}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{EQ}, \mathsf{Valid})$ *is* X-source $\mathsf{PRV\text{-}CDA2}$ secure*, for* $\mathsf{X} \in \{(T,k)\text{-}block, (T,k)\}$*, if for any probabilistic polynomial-time polynomial-size* X*-source adversary* $\mathcal{A}$*, there exists a negligible function* $\nu(\lambda)$ *such that*

$$\mathbf{Adv}^{\mathsf{PRV\text{-}CDA2}}_{\Pi,\mathcal{A}}(\lambda) \overset{\mathsf{def}}{=} \left| \Pr\left[\mathsf{Expt}^{\mathsf{real}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{rand}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] \right| \leq \nu(\lambda),$$

where for each $\mathsf{mode} \in \{\mathsf{real}, \mathsf{rand}\}$ *and* $\lambda \in \mathbb{N}$ *the experiment* $\mathsf{Expt}^{\mathsf{mode}}_{\Pi,\mathcal{A}}(\lambda)$ *is defined in Figure 1. In addition, such a scheme is* one-time *secure if the above holds for any adversary* $\mathcal{A}$ *that queries the* RoR *oracle at most once.*

$\mathsf{PRV\text{-}CDA2}$ game: $\mathsf{Expt}^{\mathsf{mode}}_{\Pi,\mathcal{A}}(\lambda)$	$\mathsf{TC2/STC2}$ game: $\mathsf{Expt}^{\mathsf{Z}}_{\Pi,\mathcal{A}}(\lambda)$
1. $\mathsf{pp} \leftarrow \mathsf{PPGen}(1^\lambda)$. 2. $b \leftarrow \mathcal{A}^{\mathsf{RoR}(\mathsf{mode},\mathsf{pp},\cdot)}(1^\lambda, \mathsf{pp})$. 3. Output b.	1. $\mathsf{pp} \leftarrow \mathsf{PPGen}(1^\lambda)$. 2. $(m, c') \leftarrow \mathcal{A}(1^\lambda, \mathsf{pp})$. 3. **If** $m = \perp$ **or** $\mathsf{Valid}(c') = 0$ output 0. 4. $k \leftarrow \mathsf{KD}_{\mathsf{pp}}(m)$. 5. $c \leftarrow (\mathsf{Enc}_{\mathsf{pp}}(k, m))$ **and** $m' \leftarrow \mathsf{Dec}_{\mathsf{pp}}(k, c')$. 6. **If** $\mathsf{Z} = \mathsf{TC2}$, $\mathsf{EQ}(c, c') = 1$, $m \neq m'$, **and** $m' \neq \perp$, output 1. 7. **If** $\mathsf{Z} = \mathsf{STC2}$, $\mathsf{EQ}(c, c') = 1$, **and** $m \neq m'$, output 1. 8. **Else**, output 0.

Fig. 1. Security games for Definitions 4.5 and 4.7.

The assumption of the plaintexts' unpredictability and support for equality testing (for use in the context of deduplication) may appear to be at odds with each other. After all, a distribution of plaintexts with sufficiently large min-entropy cannot possibly benefit from deduplication as the number of clones in a moderately sized sample is going to be negligible. However, the definition does not presuppose a particular generative model for the plaintexts. Instead, it bounds from below the amount of uncertainty that the adversary has about a particular plaintext, or in the language of Bayesian probability theory, the adversary's *prior*. In other words, Alice and Bob may share the same document that can be deduplicated on the server and will stay private as long as the server cannot guess its exact content.

Our parameter-dependent security notion enables an immediate reduction of "multi-shot" adversaries to "single-shot" ones, as is standard in public-key encryption schemes. Theorem 4.6 stated below follows via a standard hybrid argument.

Theorem 4.6 (Equivalence of PRV-CDA2 and one-time PRV-CDA2 security). *Let $k = k(\lambda)$, $T = T(\lambda)$, and $\mathsf{X} \in \{(T, k)\text{-block}, (T, k)\}$. Then, an MLE2 scheme is* X*-source* $\mathsf{PRV\text{-}CDA2}$*-secure if and only if it is* one-time X*-source* $\mathsf{PRV\text{-}CDA2}$*-secure.*

Definition 4.7 (Tag consistency). *An MLE2 scheme $\Pi = (\mathsf{PPGen}, \mathsf{KD}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{EQ}, \mathsf{Valid})$ is tag consistent (resp., strongly tag consistent) if for any probabilistic polynomial-time adversary $\mathcal{A}$, there exists a negligible function $\nu(\lambda)$ such that $\mathbf{Adv}^{\mathsf{expt}}_{\Pi,\mathcal{A}}(\lambda) \overset{\mathsf{def}}{=} \Pr\left[\mathsf{Expt}^{\mathsf{expt}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] \leq \nu(\lambda)$, where $\mathsf{expt} = \mathsf{TC2}$ (resp., $\mathsf{expt} = \mathsf{STC2}$) and for each $\mathsf{Z} \in \{\mathsf{TC2}, \mathsf{STC2}\}$, $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}^{\mathsf{Z}}_{\Pi,\mathcal{A}}(\lambda)$, is defined in Figure 1.*

Block-source adversaries vs. single-message adversaries. In the somewhat similar setting of deterministic public-key encryption, Boldyreva et al. [10] showed that for proving security against (T, k)-block-source adversaries it suffices to prove security against $(1, k)$-source adversaries. Their proof, however, does not seem to carry over to our setting, where all message distributions are required to be efficiently samplable by polynomial-sized circuits. Nevertheless, motivated by the works of Bellare et al. [7] and Brakerski and Segev [12], we now present a strengthening of our notion of $\mathsf{PRV\text{-}CDA2}$ security, for which we are able to prove an equivalence between security against (T, k)-block-source adversaries and security against $(1, k)$-source adversaries.

The strengthened notion, to which we refer as $\mathsf{aux\text{-}PRV\text{-}CDA2}$ security (i.e., $\mathsf{PRV\text{-}CDA2}$ security with auxiliary inputs), is obtained by modifying the real-or-random encryption oracle. The modification is that its inputs are now of the form $(\mathsf{mode}, \mathsf{pp}, (\mathbf{M}, \mathsf{Aux}))$, where $(\mathbf{M}, \mathsf{Aux})$ is a joint distribution over messages and auxiliary inputs. If $\mathsf{mode} = \mathsf{real}$ then the oracle samples $(m_1, \ldots, m_T, \mathsf{aux}) \leftarrow (\mathbf{M}, \mathsf{Aux})$, and if $\mathsf{mode} = \mathsf{rand}$ then the oracle samples uniform and independent messages $m_1, \ldots, m_T \leftarrow \mathcal{M}$ and $\mathsf{aux} \leftarrow \mathsf{Aux}$, independent of the messages. Next, for each $i \in [T]$, it samples $k_i \leftarrow \mathsf{KD}_{\mathsf{pp}}(m_i)$, computes $c_i \leftarrow \mathsf{Enc}_{\mathsf{pp}}(k_i, m_i)$ and outputs the vector $(c_1, \ldots, c_T, \mathsf{aux})$. For $\mathsf{X} \in \{(T, k)\text{-block}, (T, k)\}$, we say that a probabilistic polynomial-time algorithm $\mathcal{A}$ is a *polynomial-size* X*-source adversary* if for each of $\mathcal{A}$'s RoR-queries $(\mathbf{M}, \mathsf{Aux})$ it holds that: (1) the joint distribution $(\mathbf{M}, \mathsf{Aux})$ is samplable by a circuit of polynomial size, and (2) for every auxiliary input aux in the support of Aux, it holds that $\mathbf{M}|_{\mathsf{Aux}=\mathsf{aux}}$ is an X-source. Equipped with this modification, we prove the following theorem:

Theorem 4.8 (Equivalence of (T, k)-block-source and $(1, k)$-source adversaries with auxiliary inputs). *Let $k = k(\lambda)$ and $T = T(\lambda)$ be polynomial in λ. Then, an MLE2 scheme is (T, k)-block-source* $\mathsf{aux\text{-}PRV\text{-}CDA2}$*-secure if and only if it is $(1, k)$-source* $\mathsf{aux\text{-}PRV\text{-}CDA2}$*-secure.*

A comparison to the security notion of Bellare et al. [7]. In the security notion of MLE [7], adversaries are not given access to the public parameters pp when interacting with the RoR encryption oracle (unlike in step 2 in our definition). Adversaries receive pp only after all queries to the RoR oracle are completed. (Once pp is published, subsequent oracle queries return $\perp$.) Our security notion of MLE2 considers adversaries that are given access to the public parameters when interacting with the RoR encryption oracle. In particular, this enables adversaries to query the oracle with message distributions that depend on the public parameters pp (in a bounded manner, as described in the various adversary notions defined above).

The security notions of both MLE and MLE2 consider message distributions that are (T, k)-sources. All the MLE constructions of Bellare et al. are secure for (T, k)-sources, and our deterministic MLE2 construction for q-query adversaries is secure for (T, k)-sources as well (for any polynomial $T = T(\lambda)$). However, our fully randomized construction is secure only for k-sources (that is, for $(1, k)$-sources). This limitation seems to be inherent to our approach, which uses seed-dependent condensers. (See the work of Dodis et al. [17] for a discussion on the limitations of seed-dependent condensers in the presence of auxiliary inputs.)

The notions of $\mathsf{TC2}/\mathsf{STC2}$ tag consistency described above follow closely the definitions of Bellare et al., with small modifications to accommodate the more general notion of tags and the new algorithms EQ and Valid. In particular, the experiment outputs 1 only if $\mathsf{EQ}(c, c') = 1$, which corresponds to comparing tags in MLE. Additionally, we discard adversarially constructed ciphertexts c' that can be recognized by algorithm Valid as *invalid*.

5 A Fully Randomized Scheme

In this section, we present the scheme Π_{full}, a fully randomized MLE2 scheme. An overview of the construction is presented in Section 2.

The scheme. Let λ denote the security parameter. Let $\mathsf{GroupGen}$ be a probabilistic polynomial-time algorithm that takes as input a security parameter 1^λ, and outputs $(\mathbb{G}, \mathbb{G}_\mathrm{T}, p, g, \hat{e})$ where $\mathbb{G}$ and $\mathbb{G}_\mathrm{T}$ are groups of prime order p, $\mathbb{G}$ is generated by g, p is a λ-bit prime number, and $\hat{e}\colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_\mathrm{T}$ is a non-degenerate efficiently computable bilinear map. The scheme is parameterized by a parameter n that is polynomial in the security parameter. The MLE2 scheme Π_{full} comprises the following building blocks.

- A one-time secure symmetric-key encryption scheme $\mathcal{SE} = (\mathsf{K}, \mathsf{E}, \mathsf{D})$. As a concrete example, let $G\colon \mathcal{K} \to \mathcal{M}$ be a pseudorandom generator that takes short keys and expands them to the message space. We can use such a PRG as a one-time pad to get a simple, efficient, and one-time secure scheme $\mathsf{E}_k(m) := G(k) \oplus m \in \mathcal{M}$.
- An $(n+1)$-out-of-$(2n+1)$ secret sharing of a key $k \in \mathcal{K}$. The secret is encoded as an element of the field $\mathbb{F}_q$ for a prime q slightly larger than $|\mathcal{K}|$. The additive secret-sharing scheme we use (based on interpolating polynomials) also satisfies the additional property that given $2n+1$ shares of a secret, one can efficiently reconstruct (via Reed-Solomon decoding techniques [23]) the secret as long as at least $\lceil (3n+1)/2 \rceil$ shares are correctly computed.
- Two hash functions $\mathsf{RO}\colon \{0,1\}^* \to \mathbb{F}_q$ and $\mathsf{FS}\colon \{0,1\}^* \to \mathfrak{p}^{n,2n+1}$. The functions will be modeled in the proof of security as random oracles. RO is used to break circularity and FS denotes the random oracle required to implement Fiat-Shamir. Here $\mathfrak{p}^{n,2n+1}$ denotes the set of all subsets of $[2n+1]$ of cardinality n.
- A collection $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{N}}$ of collision-resistant hash functions $h\colon \mathcal{M} \to \mathbb{Z}_p$.
- A commitment scheme $\mathcal{TC} = (\mathsf{CGen}, \mathsf{Commit}, \mathsf{Reveal})$.
- A simulation-sound non-interactive extractable zero-knowledge proof system $\mathcal{ZK} = (\mathsf{ZKGen}, \mathsf{ZKProve}, \mathsf{ZKVer}, \mathsf{ZKFakeGen}, \mathsf{ZKSim}, \mathsf{ZKExt})$ for the NP language $\mathcal{L}$ defined at the end of the description of the scheme.

The scheme $\Pi_{\mathsf{full}} = (\mathsf{PPGen}, \mathsf{KD}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{EQ}, \mathsf{Valid})$ is parameterized by a parameter n that is polynomial in the security parameter and is as follows:

- **Parameter-generation algorithm:** On input 1^λ, algorithm PPGen samples $(\mathbb{G}, \mathbb{G}_\mathrm{T}, p, g, \hat{e}) \leftarrow \mathsf{GroupGen}(1^\lambda)$. It chooses a hash function $h \leftarrow \mathcal{H}$ from the family of collision-resistant hash functions, and specifies two additional hash functions RO and FS. It generates public parameters $\mathsf{pp}_{\mathsf{ZK}} \leftarrow$

$\mathsf{ZKGen}(1^\lambda)$ and $\mathsf{pp_{com}} \leftarrow \mathsf{CGen}(1^\lambda)$ and publishes $\mathsf{pp} = (\mathbb{G}, \mathbb{G}_\mathrm{T}, p, g, \hat{e}, h, \mathsf{pp_{ZK}}, \mathsf{pp_{com}})$.

- **Key-derivation function:** KD takes as input public parameters pp, a message m, and outputs the message-derived key $k_m = h(m)$.
- **Encryption algorithm:** Enc takes as input public parameters pp, a message m, and a message-derived key k_m. It samples $r \leftarrow \mathbb{Z}_p$ and first computes $\tau = \big(g^r, g^{r \cdot h(m)}\big) \in \mathbb{G}^2$.

  Creating shares of s: The algorithm chooses a key $s \leftarrow \mathsf{K}(1^\lambda)$ for scheme $\mathcal{SE}$ and an $(n+1)$-out-of-$(2n+1)$ additive secret sharing of s denoted $\vec{s} = (s_1, \ldots, s_{2n+1}) \in \mathbb{F}_q^{2n+1}$. It computes the encrypted message $d = \mathsf{E}_s(m) \in \mathcal{C}$.

  Committing to deferring elements t_i: The algorithm samples $t_1, \ldots, t_{2n+1} \leftarrow \mathbb{G}$ and lets com_i denote the commitment $\mathsf{Commit}(s_i \| t_i)$. We let $\vec{\mathsf{com}} = (\mathsf{com}_1, \ldots, \mathsf{com}_{2n+1})$ and $\vec{t} = (t_1, \ldots, t_{2n+1})$.

  Encrypting deferring elements t_i: The algorithm samples random elements $u_1, \ldots, u_{2n+1} \leftarrow \mathbb{Z}_p$ and computes ElGamal encryptions of t_i with public key $g^{h(m)}$. Let $\mathsf{et}_i = \big(g^{u_i}, g^{u_i \cdot h(m)} \cdot t_i\big)$. We let $\vec{\mathsf{et}} = (\mathsf{et}_1, \ldots, \mathsf{et}_{2n+1})$ and $\vec{u} = (u_1, \ldots, u_{2n+1})$.

  Encrypting shares of s: The algorithm encrypts s_i under t_i with a construction by Black et al. [9]. The algorithm samples $v_1, \ldots, v_{2n+1} \leftarrow \{0,1\}^\lambda$ and sets es_i to the ciphertext $(v_i, \mathsf{RO}(v_i \| t_i) + s_i \bmod q)$. We let $\vec{\mathsf{es}}$ denote $(\mathsf{es}_1, \ldots, \mathsf{es}_{2n+1})$.

  Zero-knowledge proof: The algorithm computes a proof π using algorithm $\mathsf{ZKProve}$ that the statement $\sigma = (d, \vec{\mathsf{com}}, \vec{\mathsf{et}}, \tau)$ is in the language $\mathcal{L}$ defined below.

  Cut-and-choose: The algorithm computes $X \leftarrow \mathsf{FS}\,(\sigma \| \pi \| \vec{\mathsf{es}})$ where $X \subset [2n+1]$ of cardinality n. The algorithm reveals commitments com_i for $i \in X$, denoted by $\mathsf{r}\vec{\mathsf{com}} = \{\mathsf{Reveal}(\mathsf{com}_i)\}_{i \in X}$.

  The algorithm outputs:

$$c = \big(d, \vec{\mathsf{com}}, \vec{\mathsf{et}}, \vec{\mathsf{es}}, \pi, \mathsf{r}\vec{\mathsf{com}}, \tau\big)\,.$$

- **Validity test:** On input a ciphertext $c = \big(d, \vec{\mathsf{com}}, \vec{\mathsf{et}}, \vec{\mathsf{es}}, \pi, \mathsf{r}\vec{\mathsf{com}}, \tau\big)$, algorithm Valid constructs $\sigma = \big(d, \vec{\mathsf{com}}, \vec{\mathsf{et}}, \tau\big)$. If $\mathsf{ZKVer}(\mathsf{pp_{ZK}}, \sigma, \pi) = 0$, algorithm Valid outputs 0. Next, Valid computes $X = \mathsf{FS}\,(\sigma \| \pi \| \vec{\mathsf{es}}) \in \mathfrak{p}^{n,2n+1}$ and verifies for revealed values $\{\mathsf{rcom}_i\}_{i \in X}$ from $\mathsf{r}\vec{\mathsf{com}}$ that the commitments and encryptions of $\vec{s}$, $\{\mathsf{com}_i, \mathsf{es}_i\}_{i \in X}$, are consistent with opened values $\{s_i, t_i\}_{i \in X}$. It outputs 1 if they are consistent and 0 otherwise.
- **Decryption algorithm:** On input the public parameters of the system pp, the ciphertext $c = \big(d, \vec{\mathsf{com}}, \vec{\mathsf{et}}, \vec{\mathsf{es}}, \pi, \mathsf{r}\vec{\mathsf{com}}, \tau\big)$, and a secret key k_m, if $\mathsf{Valid}(c) = 0$, the decryption algorithm outputs $\perp$. Else, the decryption algorithm first recovers t_i from $\mathsf{et}_i = (\alpha, \beta)$ with secret key k_m by computing $t_i = \beta / \big(\alpha^{k_m}\big)$. Next, using t_i, the algorithm recovers s_i from $\mathsf{es}_i = (\alpha, \beta)$ by computing $s_i = \beta - \mathsf{RO}(\alpha \| t_i) \bmod q$. It reconstructs s, given the $(n+1)$-out-of-$(2n+1)$ additive secret sharing of s, $(s_1, \ldots, s_{2n+1})$. Finally, the decryption algorithm outputs $m \leftarrow \mathsf{D}_s(d)$.

- **Equality-testing algorithm:** On input two ciphertexts, c_1 and c_2, the algorithm recovers τ_1 and τ_2. Let $\tau_1 = (g_1, h_1) \in \mathbb{G}^2$ and $\tau_2 = (g_2, h_2) \in \mathbb{G}^2$. The algorithm outputs 1 if and only if $\hat{e}(g_1, h_2) = \hat{e}(g_2, h_1)$.

The language $\mathcal{L}$ and relation $\mathcal{R}$. Intuitively, the language $\mathcal{L}$ contains only statements $\sigma = (d, \vec{\mathsf{com}}, \vec{\mathsf{et}}, \tau)$ whose components are created with the secret values $(m, r, s, \vec{t}, \vec{u})$ in a consistent manner. More formally, we define the relation $\mathcal{R} = \{(\sigma, w)\}$ of statements σ and corresponding proof strings w below and note that $\mathcal{L} = \{\sigma \ : \ \exists\, w \text{ s.t. } (\sigma, w) \in \mathcal{R}\}$:

$$\mathcal{R} := \left\{ \left((d, \vec{\mathsf{com}}, \vec{\mathsf{et}}, \tau)\ ,\ (\vec{s}, \vec{t}, m) \right) \ \middle| \ \begin{array}{r} d = \mathsf{E}_s(m) \\ \mathsf{com}_i = \mathsf{Commit}(s_i \| t_i)\ \forall\ i \in [2n+1] \\ \mathsf{et}_i = \left(g^{u_i}, g^{u_i \cdot h(m)}\right) \text{ for uniform } u_i \in \mathbb{Z}_p \\ \tau = \left(g^{r}, g^{r \cdot h(m)}\right) \text{ for uniform } r \in \mathbb{Z}_p \end{array} \right\}.$$

Correctness of the scheme $\mathit{\Pi}_{\mathsf{full}}$. Consider a ciphertext $c \leftarrow \mathsf{Enc}_{\mathsf{pp}}(h(m), m)$ with components $(d, \vec{\mathsf{com}}, \vec{\mathsf{et}}, \vec{\mathsf{es}}, \pi, \vec{\mathsf{rcom}}, \tau)$ and the secret key $k_m = h(m)$. If $\mathsf{et}_i = (\alpha, \beta)$, then we have $\beta/(\alpha^{k_m}) = g^{u_i \cdot h(m)} \cdot t_i/(g^{u_i})^{h(m)} = t_i$ as required. Next, if $\mathsf{es}_i = (\alpha, \beta)$, we have $\beta - \mathsf{RO}(\alpha \| t_i) = \mathsf{RO}(v_i \| t_i) + s_i - \mathsf{RO}(v_i \| t_i) = s_i \pmod q$ as required. The secret-sharing scheme correctly reconstructs s given shares $(s_1, \ldots, s_{2n+1})$ (via Reed-Solomon decoding techniques [23]) and therefore correctness of the scheme follows from correctness of the symmetric encryption scheme $\mathcal{SE}$.

Correctness of algorithm EQ follows from properties of groups equipped with bilinear maps. If $\tau_1 = (\alpha_1, \beta_1) \in \mathbb{G}^2$ and $\tau_2 = (\alpha_2, \beta_2) \in \mathbb{G}^2$ are constructed by the encryption scheme with the same underlying message m, then

$$\begin{aligned} \hat{e}\left(\alpha_1, \beta_2\right) &= \hat{e}\left(g^{r_1}, g^{r_2 \cdot h(m)}\right) = \hat{e}\left(g, g\right)^{r_1 r_2 \cdot h(m)}, \text{ and} \\ \hat{e}\left(\alpha_2, \beta_1\right) &= \hat{e}\left(g^{r_2}, g^{r_1 \cdot h(m)}\right) = \hat{e}\left(g, g\right)^{r_1 r_2 \cdot h(m)} \text{ as required.} \end{aligned}$$

Succinct ciphertexts. In order to shrink the ciphertexts in the scheme $\mathit{\Pi}_{\mathsf{full}}$ to be of length $|\mathsf{E}_s(m)| + \mathrm{poly}(\lambda)$, we replace the (long) NIZK proof π in our ciphertext with a non-interactive succinct extractable argument system whose length depends only on the security parameter. (Such argument systems are known to exist in the random oracle model—see the full version for the definition and instantiation.)

Specifically, our parameter generation algorithm outputs additionally the public parameters $\mathsf{pp}_{\mathsf{SA}}$ for the argument system. The encryption scheme first computes an NIZK proof π for the statement $\sigma = (d, \vec{\mathsf{com}}, \vec{\mathsf{et}}, \tau)$, and then uses π as a witness for asserting (with a succinct proof π_{SA}), using the succinct argument system, that there exists a proof π that is accepted by the verifier of the NIZK system for the assertion that $(\sigma, \pi) \in \mathcal{R}$. Finally, we discard the NIZK proof π and only include the succinct argument π_{SA} in the ciphertext. The rest of the components of the ciphertext remain unchanged. Such a technique for shrinking NIZK proofs using succinct arguments was recently used, for example, in the work of Boneh, Segev, and Waters [11]. And finally, we modify the validity test by invoking the verifier SAVer of the succinct argument system on π_{SA} instead of the verifier of the NIZK proof system.

The following theorem states that the scheme Π_{full} is k-source $\mathsf{PRV\text{-}CDA2}$ secure (see Definition 4.5). A proof outline is presented in Section 2.

Theorem 5.1. *Let $\mathcal{SE}$ be a one-time secure symmetric-key encryption scheme, $\mathcal{TC}$ be a statistically-hiding commitment scheme, $\mathcal{ZK}$ be a non-interactive extractable zero-knowledge proof system, and $\mathcal{H}$ be a family of $\left(\mathrm{poly}, 2^{-\omega(\log^2 \lambda)}\right)$-collision-resistant hash functions. Then, under the $\omega(\log^2 \lambda)$-min-entropy DDH assumption and the CDH assumption in group $\mathbb{G}$, for any $k > \omega(\log^2 \lambda)$, Π_{full} is k-source $\mathsf{PRV\text{-}CDA2}$ secure with RO and FS modeled as random oracles.*

Next, we state that the scheme Π_{full} satisfies the notion of *strong* tag consistency as in Definition 4.7.

Theorem 5.2. *Let $\mathcal{TC}$ be a secure commitment scheme, $\mathcal{ZK}$ be a non-interactive extractable zero-knowledge proof system, and $\mathcal{H}$ be a family of $\left(\mathrm{poly}, 2^{-\omega(\log^2 \lambda)}\right)$-collision-resistant hash functions. Then, setting $n \geq \omega(\log \lambda)$, Π_{full} is strongly tag consistent.*

6 A Deterministic Scheme for Bounded Message Distributions

The scheme. Our deterministic MLE2 scheme uses as a building block an IND-CPA secure symmetric-key scheme $\mathcal{SE} = (\mathsf{K}, \mathsf{E}, \mathsf{D})$ with the same message space $\mathcal{M}$ as the MLE2 scheme, key space $\mathcal{K}$, ciphertext space $\mathcal{C}$, and randomness length ρ. It is additionally parameterized by an integer $q = q(\lambda)$. The scheme $\Pi_{\mathsf{det}}^{(q)} = (\mathsf{PPGen}, \mathsf{KD}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{EQ}, \mathsf{Valid})$ is defined as follows:

- **Parameter-generation algorithm:** On input 1^λ, the algorithm PPGen chooses two hash functions $H_1 \colon \{0,1\}^* \to \mathcal{K}$ and $H_2 \colon \{0,1\}^* \to \{0,1\}^\rho$. It outputs the public parameters $\mathsf{pp} = (H_1, H_2, q)$.
- **Key-derivation function:** The algorithm KD takes as input public parameters pp, a message m, and outputs the message-derived key $k_m = H_1(m\|1) \oplus H_1(m\|2) \oplus \cdots \oplus H_1(m\|q+1) \in \mathcal{K}$.
- **Encryption algorithm:** The algorithm Enc takes as input public parameters pp, a message m, and a message-derived key k_m. It computes $r_m = H_2(m\|1) \oplus H_2(m\|2) \oplus \cdots \oplus H_2(m\|q+1)$ and outputs $\mathsf{E}_{k_m}(m; r_m) \in \mathcal{C}$.
- **Validity test:** The algorithm Valid outputs 1 on any input $c \in \mathcal{C}$.
- **Decryption algorithm:** Dec takes as input public parameters pp, a ciphertext c, and a message-derived key k_m and outputs $m \leftarrow \mathsf{D}_{k_m}(c)$.
- **Equality algorithm:** Algorithm EQ on input public parameters pp and ciphertexts c_1 and c_2 outputs 1 if and only if $c_1 = c_2$.

The following theorem, which is analogous to the combination of Theorems 5.1 and 5.2, captures security of $\Pi^{(q)}_{\mathsf{Det}}$. However, security is established in a different, incomparable adversarial model: the source specified by the adversary is allowed to output T, possibly correlated, messages at a time as long as the sampling circuit makes no more than q random oracle queries.

Theorem 6.1. *Let $q \in \mathbb{N}$ be polynomial in the security parameter λ.*

1. *If $\mathcal{SE}$ is an* IND-CPA *secure scheme and H_1 and H_2 are modeled as random oracles, then, for any any $T = \mathrm{poly}(\lambda)$ and any $k = \omega(\log \lambda)$, $\Pi^{(q)}_{\mathsf{det}}$ is q-query (T, k)-source* PRV-CDA2*-secure.*
2. *The scheme $\Pi^{(q)}_{\mathsf{Det}}$ is strongly tag consistent.*

7 Conclusions and Open Problems

Prior definitions and schemes for message-locked encryption (MLE) admit only an adversary who is oblivious to the scheme's public parameters during the initial interaction. We explore two avenues for extending security guarantees of MLE towards a more powerful adversarial model, where the distribution of plaintexts can be correlated with the scheme's parameters (lock-dependent messages). In our first construction we augment the definition of MLE to allow fully random ciphertexts by supporting equality-testing functionality. One challenging aspect of the construction is ensuring ciphertext consistency in the presence of random oracles without inflating the length of the ciphertext. We achieve this goal via a combination of a cut-and-choose technique and NIZKs. The resulting scheme is secure against a fully adaptive adversary. Our second construction assumes a predetermined bound on the complexity of distributions specified by the adversary. It fits the original framework of deterministic MLE while satisfying a stronger security notion.

We formulate the following several directions for further research. First, we ask whether a fully adaptive randomized MLE2 can be constructed and proven secure in the standard model. Second, a randomized scheme for deduplication creates a potential leakage channel that allows one user to test whether her plaintext has already been uploaded to the system (similar to the attack described by Harnik et al. [20] where the deduplication event was observable via traffic analysis). Designing a scheme resistant to this attack, for example, by supporting server-side rerandomization of ciphertexts, constitutes an interesting research question. Note that deterministic MLEs are immune to this problem. Finally, our first scheme requires a pairwise application of the equality-testing algorithm to identify all duplicate ciphertexts, and uses computationally expensive NIZKs as a building block. We leave reducing the overhead of the scheme as an open problem.

Acknowledgments. We thank the anonymous CRYPTO' 13 reviewers for their helpful comments.

References

1. Bitcasa. `http://www.bitcasa.com`
2. GNUNet. `http://www.gnu.org/software/GNUnet/`
3. Adya, A., Bolosky, W.J., Castro, M., Cermak, G., Chaiken, R., Douceur, J.R., Howell, J., Lorch, J.R., Theimer, M., Wattenhofer, R.: FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. In: Culler and Druschel [16], pp. 1–14
4. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. Lecture Notes in Computer Science, vol. 4622, pp. 535–552. Springer (2007)
5. Bellare, M., Brakerski, Z., Naor, M., Ristenpart, T., Segev, G., Shacham, H., Yilek, S.: Hedged public-key encryption: How to protect against bad randomness. In: Matsui, M. (ed.) ASIACRYPT 2009. Lecture Notes in Computer Science, vol. 5912, pp. 232–249. Springer (2009)
6. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic encryption: Definitional equivalences and constructions without random oracles. In: Wagner [31], pp. 360–378
7. Bellare, M., Keelveedhi, S., Ristenpart, T.: Message-locked encryption and secure deduplication. In: Johansson and Nguyen [21], pp. 296–312
8. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM Conference on Computer and Communications Security. pp. 62–73. ACM (1993)
9. Black, J., Rogaway, P., Shrimpton, T.: Encryption-scheme security in the presence of key-dependent messages. In: Nyberg, K., Heys, H.M. (eds.) Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 2595, pp. 62–75. Springer (2002)
10. Boldyreva, A., Fehr, S., O'Neill, A.: On notions of security for deterministic encryption, and efficient constructions without random oracles. In: Wagner [31], pp. 335–359
11. Boneh, D., Segev, G., Waters, B.: Targeted malleability: homomorphic encryption for restricted computations. In: Goldwasser, S. (ed.) ITCS. pp. 350–366. ACM (2012)
12. Brakerski, Z., Segev, G.: Better security for deterministic public-key encryption: The auxiliary-input setting. In: Rogaway, P. (ed.) CRYPTO 2011. Lecture Notes in Computer Science, vol. 6841, pp. 543–560. Springer (2011)
13. Canetti, R.: Towards realizing random oracles: Hash functions that hide all partial information. In: Kaliski, Jr., B.S. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 1294, pp. 455–469. Springer (1997)
14. Cox, L.P., Murray, C.D., Noble, B.D.: Pastiche: Making backup cheap and easy. In: Culler and Druschel [16], pp. 285–298
15. Cramer, R. (ed.): Theory of Cryptography—9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings, Lecture Notes in Computer Science, vol. 7194. Springer (2012)
16. Culler, D.E., Druschel, P. (eds.): 5th Symposium on Operating System Design and Implementation (OSDI 2002), Boston, Massachusetts, USA, December 9–11, 2002. USENIX Association (2002)
17. Dodis, Y., Ristenpart, T., Vadhan, S.P.: Randomness condensers for efficiently samplable, seed-dependent sources. In: Cramer [15], pp. 618–635

18. Douceur, J.R., Adya, A., Bolosky, W.J., Simon, D., Theimer, M.: Reclaiming space from duplicate files in a serverless distributed file system. In: ICDCS. pp. 617–624 (2002)
19. Fuller, B., O'Neill, A., Reyzin, L.: A unified approach to deterministic encryption: New constructions and a connection to computational entropy. In: Cramer [15], pp. 582–599
20. Harnik, D., Pinkas, B., Shulman-Peleg, A.: Side channels in cloud services: Deduplication in cloud storage. IEEE Security & Privacy 8(6), 40–47 (2010)
21. Johansson, T., Nguyen, P.Q. (eds.): Advances in Cryptology—EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings, Lecture Notes in Computer Science, vol. 7881. Springer (2013)
22. Kim, Y., Yurcik, W. (eds.): Proceedings of the 2008 ACM Workshop On Storage Security And Survivability, StorageSS 2008, Alexandria, VA, USA, October 31, 2008. ACM (2008)
23. MacWilliams, F., Sloane, N.: The theory of error-correcting codes. North-Holland (1977)
24. Mironov, I., Pandey, O., Reingold, O., Segev, G.: Incremental deterministic public-key encryption. In: Pointcheval and Johansson [27], pp. 628–644
25. Mislove, A., Post, A., Reis, C., Willmann, P., Druschel, P., Wallach, D.S., Bonnaire, X., Sens, P., Busca, J.M., Arantes, L.B.: POST: A secure, resilient, cooperative messaging system. In: Jones, M.B. (ed.) HotOS. pp. 61–66. USENIX (2003)
26. Muthitacharoen, A., Chen, B., Mazières, D.: A low-bandwidth network file system. In: SOSP 2001. pp. 174–187 (2001)
27. Pointcheval, D., Johansson, T. (eds.): Advances in Cryptology—EUROCRYPT 2012, 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings, Lecture Notes in Computer Science, vol. 7237. Springer (2012)
28. Quinlan, S., Dorward, S.: Venti: A new approach to archival storage. In: Long, D.D.E. (ed.) FAST. pp. 89–101. USENIX (2002)
29. Raghunathan, A., Segev, G., Vadhan, S.P.: Deterministic public-key encryption for adaptively chosen plaintext distributions. In: Johansson and Nguyen [21], pp. 93–110
30. Storer, M.W., Greenan, K.M., Long, D.D.E., Miller, E.L.: Secure data deduplication. In: Kim and Yurcik [22], pp. 1–10
31. Wagner, D. (ed.): Advances in Cryptology—CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings, Lecture Notes in Computer Science, vol. 5157. Springer (2008)
32. Wee, H.: Dual projective hashing and its applications—lossy trapdoor functions and more. In: Pointcheval and Johansson [27], pp. 246–262
33. Wilcox-O'Hearn, Z., Warner, B.: Tahoe: the least-authority filesystem. In: Kim and Yurcik [22], pp. 21–26
34. Yang, G., Tan, C.H., Huang, Q., Wong, D.S.: Probabilistic public key encryption with equality test. In: Pieprzyk, J. (ed.) CT-RSA. Lecture Notes in Computer Science, vol. 5985, pp. 119–131. Springer (2010)
35. Zhu, B., Li, K., Patterson, R.H.: Avoiding the disk bottleneck in the data domain deduplication file system. In: Baker, M., Riedel, E. (eds.) FAST. pp. 269–282. USENIX (2008)