

Simple Extremely Lossy Functions from Small-Exponent Hashing

DAMIANO ABRAM¹, AGNI DATTA¹, ARCHISMAN DUTTA², AND LAWRENCE ROY³

¹The University of Edinburgh, UK

abram.damiano@protonmail.com

agnidatta.org@gmail.com

²Aarhus University, Denmark

archismandutta@proton.me

³IBM Zurich Research, Switzerland

ldr709@gmail.com

October 5, 2026

Abstract. Extremely Lossy Functions (ELFs) are a standard model primitive that captures many useful properties of random oracles (Zhandry, Crypto 2016). While there are many variations of ELFs with additional properties, every construction (excluding obfuscation) has followed essentially the same template of bootstrapping from a sequence of ELFs secure only against fixed-size adversaries, and every construction was based on only the exponential hardness of DDH (or k -Lin), an assumption that is only reasonable over elliptic curves.

We introduce and construct Extremely Lossy Trapdoor Hashing (ELTDH), a stronger notion that implies all known variations of ELFs. Our construction achieves ELTDH in one go, without bootstrapping from schemes secure for only fixed-size adversaries, which makes it simpler and more efficient than existing ELFs. We assume exponential security of the small-exponent discrete logarithm, together with polynomial security of decisional composite residuosity (DCR). Exponential security is only required in the size of the secret exponent, not the size of the group, so the assumption plausibly holds for multiplication modulo N^2 (and for many other cryptographic groups), despite the subexponential-time discrete logarithm attacks from index calculus. Our results diversify the assumptions underlying ELFs, while also giving a simpler construction.

CONTENTS

Abstract	1
1 Introduction	3
1.1 Our Results: Extremely Lossy Trapdoor Hashing	4
1.2 Our Results: Applications	5
1.3 Our Construction	6
1.4 Comparison with Prior Work	8
1.5 Related Work	9
2 Preliminaries	10
2.1 Notation and Conventions	10
2.2 Trapdoor Hashing	11
2.3 Extremely Lossy Functions	13
2.4 Cyclic Group Samplers	14
2.5 Damgård–Jurik Groups	15
3 Image-Size Bounds and Small-Exponent Assumptions	18
3.1 A Lower Bound on the Image Size	19
3.2 Small-Exponent Assumptions with a Fixed Bound	22
3.3 Tunable Small-Exponent Assumptions	24
3.4 Generic-Group Bounds for Bounded Powers	27
3.5 Tightness and Preprocessing	30
4 Extremely Lossy Trapdoor Hashing	35
4.1 The Damgård–Jurik Construction	38
4.2 Correctness, Input Privacy, and Lossiness	39
4.3 Function Privacy and Mode Indistinguishability	43
4.4 Rate-One Trapdoor Hashing	49
5 Extremely Lossy Functions	50
5.1 The Trapdoor Construction	52
5.2 Mode Indistinguishability	55
5.3 Enumerating the Lossy Image	56
5.4 Linear-Size Plain ELF	60
6 All-But-Some and All-But-One Trapdoor ELFs	65
6.1 All-But-Some and All-But-One Syntax	65
6.2 Branch Polynomials	67
6.3 All-But-Some Construction	68
6.4 Enumerating Selected-Branch Images	73
6.5 The All-But-One Case	74
AI Disclosure and Provenance	74
Acknowledgements	74
References	76

1 Introduction

Cryptographic hash functions are used for a wide variety of different purposes, many of which do not follow from standard properties of collision resistance or pseudorandomness. Security is often analyzed in the random oracle model, a very powerful model in which the hash function behaves as a statistically uniform function, which the adversary can only ever see a negligible piece of through its queries. However, real hash functions are described by small circuits, and so are not random oracles, making any result proven in the random oracle model potentially vacuous when instantiated with real cryptographic hash functions. To bridge this gap, many strong security properties for standard model hash functions have been defined, including correlation intractability [CGH98], correlation robustness [IKNP03], correlated input security [GOR11], universal computational extraction [BHK13], and extreme lossiness [Zha16].

This last property captures how a random oracle can only be queried on a polynomial number of positions, allowing a reduction to efficiently iterate over the effective range of the oracle. An extremely lossy function (ELF) is a hash function that can be set up in two indistinguishable modes: an injective mode, and a lossy mode where the range of the ELF becomes polynomial in size. Applications of ELFs include auxiliary input point obfuscation (a model of password hashing), leakage resilience, and output intractable hash functions [Zha16]. Several variants of the ELF definition have also been proposed, with applications to deterministic encryption [Zha19], correlated-input password hashing [Zha19], and anamorphic-resistant encryption [ABG⁺26].

We address four limitations in the ELF literature.

- (1) There is only one construction (excluding obfuscation) of ELFs and its variants, based on exponential hardness of DDH (or k -Lin).
- (2) Due to subexponential-time index calculus attacks on most cryptographic groups, this construction can only be plausibly instantiated using elliptic curve groups.
- (3) This construction builds a sequence of ELFs targeting different fixed adversary runtimes, and then uses a combiner to build an ELF for all polynomial adversaries. This structure is complicated, inefficient, and makes it harder to expose additional structure or homomorphism in the ELF.
- (4) Each variant of ELFs requires redoing this entire subexponential DDH and combiner pipeline, creating a mess of many similar constructions.

In this work, we define and construct Extremely Lossy Trapdoor Hashing (ELTDH), a primitive that combines ELFs with the Trapdoor Hashing (TDH) of [DGI⁺19]. We show how this notion implies ELFs as well as all known variants of ELFs; given the wide application of TDHs, we expect that new applications of ELTDHs can be found as well.

Our construction follows a significantly different structure from [Zha16]. We assume exponential security of small-exponent discrete logarithm, together with polynomial security of decisional composite residuosity (DCR), diversifying the set of assumptions from which ELFs can be built. For small-exponent discrete logarithm, we only need security to be exponential in the size of the secret exponent, not the size of the group, so the assumption plausibly holds for any cryptographic group, even ones where index calculus gives a subexponential-time attack on discrete logarithm. Making the group large enough foils index calculus attacks, and in this case there are no known algorithms for small-exponent discrete logarithm that do better than the generic algorithm of baby-step–giant-step, which takes exponential time. The security in this assumption also can easily be tuned without

changing out the group, simply by changing the size of the small exponent; this tunability lets us avoid using a combiner.

1.1 Our Results: Extremely Lossy Trapdoor Hashing

Extremely Lossy Trapdoor Hashing can be viewed as standard trapdoor hashing with optimal compression. We recall the interaction pattern of trapdoor hashing in [FIGURE 1.1](#). At the beginning of a trapdoor hashing execution, the evaluator holds no information about the output $f(x)$; however, if we reveal the hash h and the hasher output share e , the value of $f(x)$ can be derived with (almost) perfect accuracy. This means that, even given ek , the pair (h, e) carries at least as much information as the output $f(x)$.

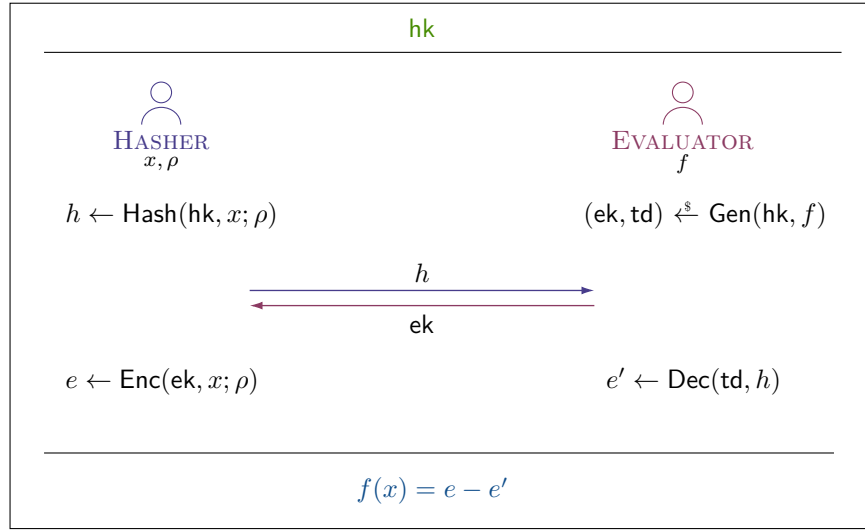


FIGURE 1.1. Interaction in trapdoor hashing. **Green** denotes the public hash key hk , **purple** the hasher with input x and random tape ρ , **pink** the evaluator with predicate f , and **blue** the predicate value recovered at the end of the interaction. The hasher computes $h \leftarrow \text{Hash}(hk, x; \rho)$ and sends h . The evaluator runs $(ek, td) \xleftarrow{\$} \text{Gen}(hk, f)$, sends the public encoding key ek , and keeps the trapdoor td . Reusing the same tape ρ , the hasher computes $e \leftarrow \text{Enc}(ek, x; \rho)$. The evaluator then derives $e' \leftarrow \text{Dec}(td, h)$, so that $f(x) = e - e'$.

For security reasons, however, the entropy of (h, e) conditioned on ek must be strictly greater than that of $f(x)$. If that were not the case, the triple (h, e, ek) would be uniquely determined by $f(x)$. The hasher could exploit this to distinguish between the encoding key for a function f_0 and that for a function f_1 . Specifically, the hasher could pick two inputs x and $\hat{x}$ such that $f_0(x) = f_0(\hat{x})$ but $f_1(x) \neq f_1(\hat{x})$. Then, it can be derived

$$\begin{aligned} h &\leftarrow \text{Hash}(hk, x) & e &\leftarrow \text{Enc}(ek, x) \\ \hat{h} &\leftarrow \text{Hash}(hk, \hat{x}) & \hat{e} &\leftarrow \text{Enc}(ek, \hat{x}) \end{aligned}$$

and check whether $(h, e) = (\hat{h}, \hat{e})$. In such a case, the hasher can be confident that the evaluator encoded the function f_0 .

This attack can be generalized to the setting in which, for every ek , there are at most r pairs (h, e) that encode $f(x)$: a malicious hasher can repeat the attack on independently sampled inputs and obtain a collision with probability at the birthday scale, namely $\Omega(k^2/r)$ after k evaluations, up

to the probability of repeated inputs. Thus, when r is polynomially bounded, polynomially many evaluations suffice to distinguish the two cases. To overcome this issue, in standard trapdoor hashing constructions, the value of r grows superpolynomially in the security parameter λ . This ensures security against all polynomial-time attackers at the same time.

In this work, we take a more fine-grained approach, following the blueprint of [Zha16]. We allow the setup procedure of trapdoor hashing to take as input the targeted parameter r . If we pick a superpolynomial r , the construction will be secure against any polynomial-time attacker; however, when we pick a polynomial r , the construction will still provide weaker but nevertheless useful security guarantees. Specifically, for all PPT attackers of bounded complexity, the hash key hk corresponding to a polynomial choice of r will be almost indistinguishable from a hk for a superpolynomial r . More formally, for every fixed PPT adversary $\mathcal{A}$ and every targeted inverse polynomial function $\delta(\lambda)$, there exists a sufficiently large polynomial $r(\lambda)$ such that $\mathcal{A}$ cannot distinguish between $hk \xleftarrow{\$} \text{Setup}(1^\lambda, r)$ and $hk \xleftarrow{\$} \text{Setup}(1^\lambda, R)$ with advantage greater than $\delta(\lambda)$, where R is the superpolynomial parameter of the standard scheme.

To understand why this property is useful, consider a cryptographic construction that relies on trapdoor hashing and that has a security loss proportional to the number r of pairs (h, e) that encode the same output $f(x)$. In standard trapdoor hashing, the parameter r is superpolynomial, and therefore the security proof will incur a superpolynomial security loss. Using extremely lossy trapdoor hashing, we can avoid this. In the real world, we instantiate the construction using a superpolynomial r , e.g. 2^λ , as for standard trapdoor hashing. Now, suppose that there exists a PPT attacker $\mathcal{A}$ that breaks the security of this unspecified cryptographic construction with some inverse polynomial advantage $\varepsilon(\lambda)$. By the properties of the extremely lossy trapdoor hashing, there exists a polynomial choice for r that makes the hash key hk at most $\varepsilon/2$ -indistinguishable from the original one. In other words, even if we instantiate the construction with this weaker key hk , the adversary $\mathcal{A}$ would still break security with advantage at least $\varepsilon - \varepsilon/2 = \varepsilon/2$. At this point, we can continue the security proof for the construction as before; however, given that r is polynomial, we will incur only a polynomial security loss.

1.2 Our Results: Applications

Extremely Lossy Functions. It is easy to see that extremely lossy trapdoor hashing implies extremely lossy functions (ELFs) [Zha16] following the connection between trapdoor hashing and lossy cryptography [PW08] highlighted by Döttling et al. [DGI⁺19]. The ELF description corresponds to the pair (hk, ek) , whereas the evaluation on some value x corresponds to the pair (h, e) where $h = \text{Hash}(hk, x)$ and $e = \text{Enc}(ek, x)$. When the ELF is in injective mode, we generate hk at the full image parameter, and we let ek encode the identity function id . Notice that the trapdoor td allows us to invert (h, e) to recover $x = e - \text{Dec}(td, h)$.¹ For the lossy mode, instead, we generate hk using a polynomial value r and we let ek encode the zero function $f \equiv 0$.

However, extremely lossy trapdoor hashing is stronger than extremely lossy functions. First of all, due to their potential injectivity, ELFs must have a large output. Moreover, they typically allow us to extract information about the input from the output when they are set in injective mode [Zha16]. On the other hand, by introducing small modifications to the above blueprint of ELFs from extremely lossy trapdoor hashing, we can easily build a stronger variant of the primitive

¹This trapdoor lets us turn it into a trapdoor ELF as well, since it allows efficient inversion.

that achieves compression and allows the extraction of partial information about the input even when the ELF is in lossy mode. This more closely resembles the properties of a random oracle, which ELFs attempt to emulate [Zha16]. We additionally show that the image of the public map is at most the image parameter times the range of the predicate, so a predicate of small range leaves the map lossy while we can still recover its value with the trapdoor.

A Plain ELF and Exact Images. In fact, we can drop the trapdoor altogether and get something simpler: an ordinary ELF over any cyclic group sampler that satisfies tunable DSEDL. It is published as n bases and evaluated to a single group element. Indeed, composite residuosity does not play any part here. Both ELFs here have exactly enumerable images, as do the selected branches of the branch variants. This means that, from a lossy key and r alone, we can use a prefix dynamic program to list the image in time $\text{poly}(r, \log m)$, which lets an application sample from it uniformly.

Branch Variants. Applications involving multiple correlated ELF evaluations (such as correlated input secure one-way functions and CCA-secure deterministic public key encryption [Zha19]) require more than a single function from a trapdoor ELF. We therefore add branch functionality, where a designated set of branches is lossy, and every other branch stays invertible under an inversion oracle [Zha19]. The same hash gives this directly. We publish two independent bundles of encoding rows, so that one bundle still answers the inversion oracle while a function-privacy hybrid rewrites the other, as selective security requires. The all-but-one primitive of [Zha19] is the case of a single selected branch.

Anamorphic-Resistant Encryption. Without its trapdoor, our trapdoor ELF is always injective in the injective mode, as the ELF definition of [ABG⁺26] requires, so the anamorphic-resistant encryption of [ABG⁺26] from ELFs and fully unique non-interactive zero-knowledge arguments holds under our assumptions. Its second construction, which avoids indistinguishability obfuscation, uses a partial trapdoor of a variant of the trapdoor ELF of [Zha19], and whether our construction admits such a partial trapdoor is open.

1.3 Our Construction

In this work, we present an extremely lossy trapdoor hashing scheme that supports the evaluation of any linear function.² Our construction rests on tunable DSEDL over a cyclic group $\mathbb{G} = \langle g \rangle$, a decisional form of the small-exponent discrete logarithm problem. At a fixed interval $[\gamma]$, it asks whether g^s , for s sampled from $[\gamma]$, can be distinguished from a uniform group element. A fixed polynomial interval is searchable by baby-step-giant-step in $\mathcal{O}(\sqrt{\gamma})$ group operations, so the assumption rests on the order of the quantifiers, with the tunable form fixing the observer first and only then choosing an interval large enough to defeat that one observer. For every polynomial size bound t and every inverse polynomial δ , there is a polynomial Q such that every adversary $\mathcal{A}$ of size t separates a uniform element of $\mathbb{G}$ from g^s with advantage at most δ , for every interval bound $\gamma \geq Q(\lambda)$. Zhandry argues that ELFs call for an exponentially hard assumption too [Zha16].

How Small the Image Can Be. The tuning is forced on us. Evaluation is public, so anyone can run the function on k random inputs and watch for a repeat, and an image of r points produces one

²We expect the same technique to reach functions described by multivariate polynomials whose monomials come from a fixed set Ω of polynomial size, although we prove nothing about that class here. The recipe would ask one party to hash the evaluation of all the monomials in Ω on their input x , and the other party to provide the encoding key corresponding to the coefficients of the multivariate polynomial describing f .

at the birthday scale k^2/r . Any ELF that hides its lossy mode from every observer of size k up to advantage ε therefore needs $r = \Omega(k^2/\varepsilon)$, which is why the image parameter has to be chosen after the observer and never before. We obtain the matching upper bound in the generic group model, where a distinguisher making T queries gains $\mathcal{O}(T^2/\gamma)$ in a group whose order has large prime factors, and baby-step-giant-step attains it.

Recalling DDLog-based Trapdoor Hashing. To better understand our construction, it is worth recalling the group-based trapdoor hashing schemes of [DGI⁺19, RS21]. These build on the distributed discrete logarithm framework of [ADOS22], which can be instantiated over Damgård–Jurik groups or class groups (which have a transparent setup). The framework is defined over a group $\mathbb{H}$ that can be decomposed as the direct product of two cyclic groups $\mathbb{D} = \langle \hat{g} \rangle$ and $\mathbb{G} = \langle g \rangle$, where the uniform distribution over $\mathbb{H}$ is computationally indistinguishable from the uniform distribution over $\mathbb{G}$. The groups also satisfy the property that discrete logarithms are easy in $\mathbb{D}$ and hard in $\mathbb{G}$. Moreover, an efficient procedure called distributed discrete logarithm converts group elements g_0, g_1 with $g_1/g_0 = \hat{g}^\sigma$ into integers $s_0 = \text{DDLog}(g_0)$ and $s_1 = \text{DDLog}(g_1)$ satisfying $s_1 - s_0 = \sigma \bmod N$, where $N := \text{ord}(\hat{g})$.

In the construction, the hash key hk consists of d random group elements $g_1, \dots, g_d$, where $g_i \xleftarrow{\$} \mathbb{G}$. To hash an integer string $\mathbf{x} := (x_1, \dots, x_d) \in (\mathbb{Z}/N\mathbb{Z})^d$ we simply compute $h = g_1^{x_1} \cdots g_d^{x_d}$.

To encode the linear function $f_{\mathbf{y}}(\mathbf{x}) := \langle \mathbf{x}, \mathbf{y} \rangle$, where $\mathbf{y} = (y_1, \dots, y_d) \in (\mathbb{Z}/N\mathbb{Z})^d$, we sample a random σ over a sufficiently large interval and we compute

$$K_1 = \hat{g}^{y_1} \cdot g_1^\sigma, \quad \dots \quad K_d = \hat{g}^{y_d} \cdot g_d^\sigma.$$

The encoding key ek consists of the tuple $(K_1, \dots, K_d)$, whereas the trapdoor is $\text{td} = \sigma$. Observe that ek hides $\mathbf{y}$, since a uniform element of $\mathbb{G}$ is indistinguishable from a uniform element of $\mathbb{H}$.

Finally, we recall how the parties can derive a secret-sharing of $f_{\mathbf{y}}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{y} \rangle$ from ek and h . The hasher computes

$$\begin{aligned} h_1 &= K_1^{x_1} \cdots K_d^{x_d} \\ &= \hat{g}^{x_1 \cdot y_1} \cdot g_1^{x_1 \cdot \sigma} \cdots \hat{g}^{x_d \cdot y_d} \cdot g_d^{x_d \cdot \sigma} \\ &= \hat{g}^{\langle \mathbf{x}, \mathbf{y} \rangle} \cdot h^\sigma. \end{aligned}$$

On the other hand, the evaluator can compute $h_0 = h^\sigma$ using the fact that $\text{td} = \sigma$. At this point, the parties hold group elements h_1 and h_0 such that $h_1/h_0 = \hat{g}^{\langle \mathbf{x}, \mathbf{y} \rangle}$. By locally applying DDLog to their group elements, the parties derive a secret sharing of the result $\langle \mathbf{x}, \mathbf{y} \rangle$.

Our Extremely Lossy Trapdoor Hashing Scheme. Our construction is essentially the same as the construction of [DGI⁺19, RS21] with two main modifications. First of all, instead of directly hashing $\mathbf{x}$, we hash the bit decomposition $\mathbf{G}^{-1}(\mathbf{x})$, the bits of the coordinates of $\mathbf{x}$ laid out in one vector (the hk needs to become $\lceil \log_2 N \rceil$ times bigger). On the other hand, instead of encoding $f_{\mathbf{y}}$, we ask the evaluator to encode $f_{\mathbf{y} \otimes \mathbf{g}}$ where $\mathbf{g} = (1, 2, \dots, 2^{\lceil \log_2 N \rceil - 1})$. Notice that correctness is preserved as

$$f_{\mathbf{y} \otimes \mathbf{g}}(\mathbf{G}^{-1}(\mathbf{x})) = \langle \mathbf{G}^{-1}(\mathbf{x}), \mathbf{y} \otimes \mathbf{g} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle.$$

The second adjustment is that, instead of sampling the elements $g_1, \dots, g_d$ in the hash key hk uniformly at random, we sample s_i over a small set $[\gamma]$ of polynomial size and set $g_i \leftarrow g^{s_i}$. Tunable DSEDL immediately makes the new hash key indistinguishable from the old one for every adversary

running in time well below $\sqrt{\gamma}$, up to a small inverse polynomial advantage.

Now, fix hk and ek and let us count how many pairs (h, e) encode a given output $\langle \mathbf{x}, \mathbf{y} \rangle$. First of all, observe that for every h , there is at most one e that is consistent with $\langle \mathbf{x}, \mathbf{y} \rangle$ (assuming perfect correctness). Indeed, this must be $\langle \mathbf{x}, \mathbf{y} \rangle + \text{Dec}(\text{td}, h)$. Therefore, the question boils down to computing the size of the image of $\text{Hash}(hk, \cdot)$. Notice that every hash h is of the form g^z where $z = \langle \mathbf{G}^{-1}(\mathbf{x}), \mathbf{s} \rangle$. Since $\mathbf{G}^{-1}(\mathbf{x})$ is a bit vector of length $d \lceil \log_2 N \rceil$ and every s_i lies in $[\gamma]$, we conclude that $z \in [0, d\gamma \lceil \log_2 N \rceil]$. Notice that this is an interval of polynomial size. In other words, we have proven that there are at most polynomially many pairs (h, e) that encode a given output $\langle \mathbf{x}, \mathbf{y} \rangle$. The rest of the scheme survives the change. The two shares still differ by exactly the predicate value, so the construction is perfectly correct and rate one, and the digest still hides the input statistically.

From the Hash to a Trapdoor ELF. A single residue is already enough for a whole ELF input. For the domain $[m]$, we send u to the residue $x = u - 1$ and publish the pair (h, e) , the digest beside the encoder's share. Injective generation puts the identity predicate in the encoding key at a superpolynomial r , which makes $e - \text{Dec}(\text{td}, h) = x$, so the trapdoor inverts every output. Turning to the lossy mode, we instead take the zero predicate at a polynomial r ; the share e is then already fixed by h , and the public map inherits the image of the hash, of size at most r . The bit decomposition never leaves the hash.

Branch Arithmetic. We encode a branch $b \in \{0, 1\}^a$ by the integer $\beta(b)$ and associate a selected set $\mathcal{S}^*$ with the polynomial

$$P_{\mathcal{S}^*}(T) = \prod_{b^* \in \mathcal{S}^*} (\beta(b^*) - T) = (-1)^s T^s + \sum_{i=0}^{s-1} c_i T^i.$$

Each coefficient c_i becomes one scalar ELTDH predicate $x \mapsto c_i x$. On branch b , weighting the decoded rows by $1, \beta(b), \dots, \beta(b)^{s-1}$ reconstructs $P_{\mathcal{S}^*}(\beta(b))x$. For selective security, we duplicate these coefficient rows into two bundles: while one bundle changes row by row under function privacy, the other retains the trapdoors needed to answer inversion queries.

1.4 Comparison with Prior Work

TABLE 1.1 sets the constructions side by side at domain size m , writing $n = \lceil \log_2 m \rceil$ for the input length, a for the branch length, and $s^* = |\mathcal{S}^*|$ for the number of selected branches, with ABO and ABS the all-but-one and all-but-some variants. Group, field, exponent, residue, hash, and group-operation units appear as G, F, E, R, H, and Op. Our rows count elements beside the common group parameters and integer metadata, with $w = 2\ell(\lambda_{\text{grp}}(\cdot))$ the padded bit width of one residue, for the group-security function λ_{grp} and the prime bit length ℓ , evaluated at n for the trapdoor ELF, at $n + a$ for ABO, and at $n + as^*$ for ABS. One group element of the plain construction takes $\ell_{\text{grp}}(\lambda)$ bits at $\lambda = \lceil cn \rceil$, for the representation-length bound ℓ_{grp} of its group sampler, so its element count is linear while its bit length grows faster.

The trapdoor rows follow from the construction of [Zha19] rather than from a bound stated there. That construction composes $t = \lceil \sqrt{\log \log m} \rceil$ layers of the matrix-in-the-exponent function of [FGK⁺10] at $r_i = 2^{2^i}$, and the bounded-adversary output bound $\log N \leq C \log M \log r$ leaves the last layer with $n^{3/2+o(1)}$ input bits, so its square matrix dominates the key at $n^{3+o(1)}$ elements. A branch length a multiplies each layer by the $2a + 1$ matrices of $A_b = B + \sum_i A_{i,b_i}$.

CONSTRUCTION	PUBLIC	TRAPDOOR	OUTPUT	EVALUATION
PRIOR WORK				
ELF [Zha16]	$\Theta(n^2)G + \Theta(n)H$	–	$3n$ bits	$\Theta(n^2)Op + \Theta(n)$ hashes
TELF [Zha19]	$n^{3+o(1)}G$	$n^{3+o(1)}F$	$n^{3/2+o(1)}G$	$n^{3+o(1)}Op$
ABO [Zha19]	$an^{3+o(1)}G$	$an^{3+o(1)}F$	$n^{3/2+o(1)}G$	$an^{3+o(1)}Op$
THIS WORK				
Plain ELF	nG	–	$1G$	$\mathcal{O}(n)Op$
Trapdoor ELF	$(2w + 2)G$	$1E$	$1G + 1R$	$\mathcal{O}(n)Op$
Selective ABO	$3(w + 1)G$	$2E$	$1G + 2R$	$\mathcal{O}(n)Op$
Selective ABS	$(2s^* + 1)(w + 1)G$	$2s^*E$	$1G + 2R$	$\mathcal{O}(s^*n)Op$

TABLE 1.1. Asymptotic efficiency comparison of extremely lossy functions and their trapdoor variants.

Outline. We state our results in §1 and give the technical overview in §1.3. In §2 we fix the syntax and algebraic facts that our constructions use, and in §3 we take up the image-size barrier, the tunable DSEDL assumption, and the evidence from the generic group model. We construct ELTDH in §4, compile it into trapdoor and plain ELFs in §5, and build the branch variants in §6, with polynomially many selected branches and ABO as the singleton case. We compare with prior work at length in §1.5.

1.5 Related Work

Lossy and Extremely Lossy Functions. For lossy trapdoor functions [PW08] and their many refinements, namely compact and correlation-secure variants [BW10, FGK⁺10], slight and identity-based lossiness [MY10, BKPW12, Wee12], lossy codes [DM13], lossy encryption [HLOV11, HO13], all-but-many branches [Hof12, BL17, LSSS17], adversary-dependent and targeted lossiness [YYHK16, QWW21], and code-based constructions [DJ24], the size of the lossy image is fixed before the adversary is known, and lossiness does not amplify generically [PRS12]. For extremely lossy functions [Zha16], in contrast, a polynomial image is chosen after the observer, which is exactly what random-oracle-style reductions need, and Zhandry [Zha19] uses their trapdoor and branch variants for deterministic encryption and correlated-input security. Outside obfuscation [IL25], every construction is obtained by bootstrapping from a sequence of exponentially DDH-hard instances, while direct algebraic constructions are consistent with the black-box separations of [FR23]. We give such a direct construction, in which we replace the sequence of security levels by one bounded subset-sum space and obtain the trapdoor, all-but-one, and all-but-some variants from one hash.

Trapdoor Hashing and Succinct Computation. Trapdoor hashing [DGI⁺19] followed work on short openings at hidden positions [OPWW15, HW15], and from its DCR instance, Roy and Singh [RS21] obtain rate-one trapdoor hashing and large-message HSS. It shares the goal of succinct computation with laconic oblivious transfer, function evaluation, and conditional disclosure [CDG⁺17, QWW18, DGGM19, BDHL25, AMR25b], with somewhere-extractable hashing [DGKV22, BDSZ24], and with enhanced, vector, and sublinear trapdoor hashes [BCD⁺25, BG25, CHP25, AMR25a]. Its group-based instances are based on distributed discrete logarithms from homomorphic and function secret sharing [GI14, BGI15, BGI16b, BGI16a, BCG⁺17, BGI⁺18, LMS18, DKK18, BKS19, OSY21, ILM21, DIJL23, ACK23]. In all of these instances, the hash bases are uniform, and we change only

their distribution, so that the digest depends on one short subset sum and trapdoor hashing acquires a tunable, extremely small image. With a counting argument, we show that independently bounded rows cannot do the same.

Short Exponents and Generic Evidence. Short exponents were studied for efficiency and pseudorandomness [vOW96, PS98, KK04] and as the c -decisional short discrete logarithm problem [AFLT12, AFLT16], always over superpolynomial intervals, while constrained discrete logarithms depend on the shape of the allowed set [MMN06] and preprocessing can help decisional problems [CK18]. Our tunable assumption instead needs polynomial intervals chosen after the observer, so we analyze it directly. Against the classical generic lower bounds [Nec94, Sho97, Mau05] and square-root searches [Sha71, Pol78, vOW96, Pol00], we prove an $\mathcal{O}(T^2/\gamma + T^2/\pi_q)$ bound for T generic operations, tight even for one challenge, and extend it to auxiliary input by presampling [CDG18], while the results of [CHW26] in structured generic models do not yet include the decisional claim. The interval must grow quadratically in the observer’s work, the same scale that public collision testing forces on every ELF.

2 Preliminaries

We fix our notation and recall the definitions of trapdoor hashing [DGI⁺19] and of extremely lossy functions [Zha16], which we construct in this paper. We also define the cyclic group samplers over which we state our assumptions, and we recall the Damgård–Jurik group, which is such a sampler under the safe-prime setup convention, and the composite-residuosity assumption.

2.1 Notation and Conventions

We fix the encodings of integers and residues, our probability notation, and the quantitative indistinguishability relation in which we state every mode-indistinguishability bound.

SETS AND ENCODINGS. We denote the security parameter by λ and write $\{0, 1\}$ for the binary alphabet. For integers $a \leq b$, we write $[a..b] := \{a, a+1, \dots, b\}$, and for every positive integer n , we write $[n] := [1..n]$. For $u \in [2^n]$, we write $x(u) = (x_1(u), \dots, x_n(u)) \in \{0, 1\}^n$ for the unique bit vector such that

$$u - 1 = \sum_{j=1}^n 2^{j-1} x_j(u).$$

For an integer $M \geq 1$ and $v \in \mathbb{Z}/M\mathbb{Z}$, we write $\text{rep}(v) \in [0..M-1]$ for the least nonnegative representative of v . For every integer $w \geq \lceil \log_2 M \rceil$, we write $\text{bin}_w(v) = (v_0, \dots, v_{w-1}) \in \{0, 1\}^w$ for the unique padded binary expansion of v , that is, the unique bit vector such that

$$\text{rep}(v) = \sum_{k=0}^{w-1} 2^k v_k.$$

For a residue vector $\mathbf{x} = (x_1, \dots, x_d)$, we apply bin_w coordinatewise and write $x_{j,k}$ for the k th bit of x_j . We write vectors and matrices in boldface, $\mathbf{v}_j$ or $\mathbf{v}[j]$ for a coordinate of a vector, $\mathbf{A}_{i,j}$ for an entry of a matrix, and $\mathbf{A}^\top$ for the transpose.

ALGORITHMS AND PROBABILITY. For a randomized algorithm $\mathcal{A}$, $y \stackrel{\$}{\leftarrow} \mathcal{A}(x)$ denotes the output of $\mathcal{A}$ on input x and independent uniform coins, and $y \leftarrow \mathcal{A}(x; \rho)$ denotes its output on random tape

ρ . For a deterministic algorithm $\mathcal{A}$, we write $y \leftarrow \mathcal{A}(x)$, and for a finite set S , $y \stackrel{\$}{\leftarrow} S$ denotes a uniform sample from S . The notation $\rho \stackrel{\$}{\leftarrow} \{0, 1\}^*$ denotes an infinite tape of independent uniform bits, and a polynomial-time algorithm depends only on a polynomially bounded prefix of it. For $k \in \mathbb{N}$, we write $\rho|_k \in [0..2^k - 1]$ for the integer whose binary representation is the first k bits of ρ . For a proposition P , we write $1[P] \in \{0, 1\}$ for the indicator that equals one if and only if P holds. Unless we say otherwise, we allow every PPT adversary to be nonuniform.

ASYMPTOTICS. We write $\text{poly}(n)$ for an unspecified function bounded above by a polynomial in n . Unless we say otherwise, a polynomial in an asymptotic quantifier is positive on all sufficiently large inputs. When we use such a polynomial as an integer parameter or as an endpoint of an integer interval, we replace it by an integer-valued polynomial upper bound, and this replacement does not change any asymptotic statement.

DISTRIBUTIONS. For ensembles $X = (X_k)_{k \in \mathbb{N}}$ and $Y = (Y_k)_{k \in \mathbb{N}}$ over a common domain, we write $X \equiv Y$, $X \equiv_s Y$, and $X \equiv_c Y$ when X and Y are identically distributed, statistically indistinguishable, and computationally indistinguishable against nonuniform PPT distinguishers, respectively. For random variables X_k and Y_k at a fixed parameter k , we write $X_k \equiv_{s, \alpha} Y_k$ when every probabilistic circuit of size at most $s(k)$ has advantage at most $\alpha(k)$ in distinguishing X_k from Y_k . The first subscript is a size bound and the second an advantage bound, so a target advantage $1/\delta(k)$ is written as the subscript $1/\delta$, and the relation is required at every parameter k that the surrounding quantifiers fix.

In every proof of mode indistinguishability, we fix one threshold for all circuits of a given polynomial size, and such a threshold exists by the following consequence of nonuniform indistinguishability.

LEMMA 2.1 (Threshold for Polynomial-Size Circuits). Let $X = (X_k)_{k \in \mathbb{N}}$ and $Y = (Y_k)_{k \in \mathbb{N}}$ be ensembles with $X \equiv_c Y$. For every polynomial p and every positive polynomial δ , there exists $k_0 \in \mathbb{N}$ such that $X_k \equiv_{p, 1/\delta} Y_k$ for every $k \geq k_0$. $\dashv$

Proof. We combine one violating circuit for each bad parameter into one nonuniform family, whose advantage then exceeds $1/\delta(k)$ infinitely often. We suppose, toward a contradiction, that the conclusion fails for some p and δ . Then for infinitely many k , there is a probabilistic circuit A_k of size at most $p(k)$ such that

$$|\Pr[A_k(X_k) = 1] - \Pr[A_k(Y_k) = 1]| > 1/\delta(k).$$

For each such k , we choose one violating circuit and complement its output if necessary, so that

$$\Pr[A_k(X_k) = 1] - \Pr[A_k(Y_k) = 1] > 1/\delta(k),$$

and for every other k , we let A_k output 0. The family $A = (A_k)_{k \in \mathbb{N}}$ is nonuniform and has polynomial size, and its distinguishing advantage exceeds $1/\delta(k)$ for infinitely many k . Since this advantage is not negligible, $X \equiv_c Y$ is false, a contradiction. $\blacksquare$

2.2 Trapdoor Hashing

In trapdoor hashing [DGI⁺19], a short digest of a long input x and a trapdoor for a predicate f determine a short encoding of $f(x)$. We require that the encoding computed from x equals the candidate that the trapdoor and the digest select for $f(x)$, that the digests of any two inputs of equal length are indistinguishable, and that the encoding keys of any two predicates are indistinguishable.

When a scheme has auxiliary public parameters shared by all algorithms, we give them to every algorithm and adversary and suppress them from the syntax.

We use the syntax of [DGI⁺19] with three changes: the predicate class can depend on the hash key, we leave the lengths of digests and encodings open, and we require the decoding algorithm to be deterministic.

DEFINITION 2.2 (Trapdoor Hashing). Let $\mathcal{C}$ be a predicate class indexed by $\lambda, n \in \mathbb{N}$ and by a hash key, such that every $f \in \mathcal{C}_{\lambda, n, \text{hk}}$ maps $\{0, 1\}^n$ to $\{0, 1\}$. A trapdoor-hashing scheme for $\mathcal{C}$ is a tuple $\text{THF} = (\text{TDH.Setup}, \text{TDH.Hash}, \text{TDH.Gen}, \text{TDH.Enc}, \text{TDH.Dec})$ of the following five probabilistic polynomial-time algorithms.

- $\text{hk} \xleftarrow{\$} \text{TDH.Setup}(1^\lambda, 1^n)$. On input the security parameter and an input length n , the setup algorithm samples a hash key hk for inputs in $\{0, 1\}^n$.
- $h \leftarrow \text{TDH.Hash}(\text{hk}, x; \rho)$. On input a hash key hk , an input $x \in \{0, 1\}^n$, and a random tape ρ , the hashing algorithm deterministically outputs a digest h .
- $(\text{ek}, \text{td}) \xleftarrow{\$} \text{TDH.Gen}(\text{hk}, f)$. On input a hash key hk and a predicate $f \in \mathcal{C}_{\lambda, n, \text{hk}}$, the generation algorithm samples a public encoding key ek and a trapdoor td .
- $t \leftarrow \text{TDH.Enc}(\text{ek}, x; \rho)$. On input an encoding key ek , an input $x \in \{0, 1\}^n$, and a random tape ρ , the encoding algorithm deterministically outputs an encoding t .
- $(t_0, t_1) \leftarrow \text{TDH.Dec}(\text{td}, h)$. On input a trapdoor td and a digest h , the deterministic decoding algorithm outputs the two candidate encodings t_0 and t_1 , indexed by the two possible values of $f(x)$. J

We state the correctness of [DGI⁺19] as one failure bound δ , for every hash key in the support of setup, on the event that the encoding equals the candidate indexed by $f(x)$ and the two candidates differ. For a negligible δ , this implies the full correctness of [DGI⁺19], and in the lossy constructions, we add δ to the other failure events by a union bound.

DEFINITION 2.3 (Correctness). Let $\delta: \mathbb{N} \rightarrow [0, 1]$. A trapdoor-hashing scheme THF for $\mathcal{C}$ is δ -correct if for every $\lambda, n \in \mathbb{N}$, every $x \in \{0, 1\}^n$, every $\text{hk} \in \text{supp}(\text{TDH.Setup}(1^\lambda, 1^n))$, and every $f \in \mathcal{C}_{\lambda, n, \text{hk}}$,

$$\Pr \left[t = t_{f(x)} \wedge t_0 \neq t_1 \mid \begin{array}{l} \rho \xleftarrow{\$} \{0, 1\}^*, \\ h \leftarrow \text{TDH.Hash}(\text{hk}, x; \rho), \\ (\text{ek}, \text{td}) \xleftarrow{\$} \text{TDH.Gen}(\text{hk}, f), \\ t \leftarrow \text{TDH.Enc}(\text{ek}, x; \rho), \\ (t_0, t_1) \leftarrow \text{TDH.Dec}(\text{td}, h) \end{array} \right] \geq 1 - \delta(\lambda).$$

We call THF fully correct if it is ε -correct for some negligible function ε , and perfectly correct if it is 0-correct. J

Input privacy is the notion of [DGI⁺19], with nonuniform sequences of input pairs, to which we add a perfect variant.

DEFINITION 2.4 (Input Privacy). Let $x^0 = (x_\lambda^0)_{\lambda \in \mathbb{N}}$ and $x^1 = (x_\lambda^1)_{\lambda \in \mathbb{N}}$ be sequences of polynomial length with $n_\lambda := |x_\lambda^0| = |x_\lambda^1|$ for every $\lambda \in \mathbb{N}$. For $b \in \{0, 1\}$, define the ensemble $\mathfrak{D}_b^{\text{input}} = (\mathfrak{D}_{b, \lambda}^{\text{input}})_{\lambda \in \mathbb{N}}$ by

$$\mathfrak{D}_{b, \lambda}^{\text{input}} := \left\{ (\text{hk}_\lambda, h_\lambda^b) \mid \begin{array}{l} \text{hk}_\lambda \xleftarrow{\$} \text{TDH.Setup}(1^\lambda, 1^{n_\lambda}), \\ \rho_\lambda^b \xleftarrow{\$} \{0, 1\}^*, \\ h_\lambda^b \leftarrow \text{TDH.Hash}(\text{hk}_\lambda, x_\lambda^b; \rho_\lambda^b) \end{array} \right\}.$$

A trapdoor-hashing scheme THF is input private if $\mathfrak{D}_0^{\text{input}} \equiv_c \mathfrak{D}_1^{\text{input}}$ for every such pair of sequences. It has statistical input privacy if $\mathfrak{D}_0^{\text{input}} \equiv_s \mathfrak{D}_1^{\text{input}}$, and perfect input privacy if $\mathfrak{D}_0^{\text{input}} \equiv \mathfrak{D}_1^{\text{input}}$, for every such pair. $\lrcorner$

Function privacy is the analogue for trapdoor hashing of mode indistinguishability for lossy trapdoor functions [PW08]. In the function privacy of [DGI⁺19], the two predicate sequences are fixed in advance. Since our predicate class depends on the hash key, we let a nonuniform selector choose the two predicates after it receives the hash key, and we give its state st to the distinguisher.

DEFINITION 2.5 (Function Privacy). Let $n: \mathbb{N} \rightarrow \mathbb{N}$ be polynomially bounded, and write $n_\lambda := n(\lambda)$. A selector is a nonuniform PPT algorithm S that on every input $(1^\lambda, \text{hk})$ outputs a triple (f^0, f^1, st) with $f^0, f^1 \in \mathcal{C}_{\lambda, n_\lambda, \text{hk}}$. A trapdoor-hashing scheme THF for $\mathcal{C}$ is function private if for every selector S ,

$$\begin{aligned} & \left\{ (\text{hk}, \text{st}, \text{ek}) \left| \begin{array}{l} \text{hk} \xleftarrow{\$} \text{TDH.Setup}(1^\lambda, 1^{n_\lambda}), \\ (f^0, f^1, \text{st}) \xleftarrow{\$} S(1^\lambda, \text{hk}), \\ (\text{ek}, \text{td}) \xleftarrow{\$} \text{TDH.Gen}(\text{hk}, f^0) \end{array} \right. \right\}_{\lambda \in \mathbb{N}} \\ & \equiv_c \left\{ (\text{hk}, \text{st}, \text{ek}) \left| \begin{array}{l} \text{hk} \xleftarrow{\$} \text{TDH.Setup}(1^\lambda, 1^{n_\lambda}), \\ (f^0, f^1, \text{st}) \xleftarrow{\$} S(1^\lambda, \text{hk}), \\ (\text{ek}, \text{td}) \xleftarrow{\$} \text{TDH.Gen}(\text{hk}, f^1) \end{array} \right. \right\}_{\lambda \in \mathbb{N}}. \end{aligned}$$

 $\lrcorner$

2.3 Extremely Lossy Functions

[PW08] defines lossy trapdoor functions, whose injective and lossy modes are computationally indistinguishable, and constructs chosen-ciphertext-secure encryption from them, and [BFO08] constructs deterministic encryption from them. [Zha16] defines extremely lossy functions, in which the bound r on the lossy image can be any sufficiently large polynomial chosen after the running time and target advantage of the adversary.

We use the definition of [Zha16] with three changes. We write m and $N_{\text{out}}(m)$ for the domain size and the bound on the range, which [Zha16] calls M and N , because N denotes a modulus in this paper, and we require $\log N_{\text{out}}(m) = \text{poly}(\log m)$ as [Zha19] does when it recalls this definition. We give evaluation as a separate deterministic algorithm. We state mode indistinguishability for probabilistic circuits of size t , where [Zha16] quantifies over adversaries whose running time is at most a polynomial p .

DEFINITION 2.6 (Extremely Lossy Function). For every $m \geq 2$, write $n_m := \lceil \log_2 m \rceil$. An extremely lossy function $\text{ELF} = (\text{ELF.Gen}, \text{ELF.Eval})$ consists of a function N_{out} with $\log N_{\text{out}}(m) = \text{poly}(n_m)$ and the following two algorithms.

- $f \xleftarrow{\$} \text{ELF.Gen}(m, r)$. On input integers $m \geq 2$ and $r \in [m]$, the probabilistic polynomial-time generation algorithm samples a description f of a function $f: [m] \rightarrow [N_{\text{out}}(m)]$.
- $y \leftarrow \text{ELF.Eval}(f, u)$. On input a description f and $u \in [m]$, the deterministic evaluation algorithm outputs $y = f(u)$ in time $\text{poly}(n_m)$.

We identify a description f with the function $u \mapsto \text{ELF.Eval}(f, u)$. The pair ELF satisfies the following three properties.

- **Injective Mode.** If $f \xleftarrow{\$} \text{ELF.Gen}(m, m)$, then f is injective except with probability negligible in n_m .

- **Lossy Mode.** For every $r \in [m]$, if $f \xleftarrow{s} \text{ELF.Gen}(m, r)$, then $|\text{Im}(f)| \leq r$ except with probability negligible in n_m .
- **Mode Indistinguishability.** For every polynomial t and every positive polynomial δ , there exists a polynomial Q such that for every sufficiently large m and every $r \in [Q(n_m)..m]$,

$$\left\{ f \mid f \xleftarrow{s} \text{ELF.Gen}(m, m) \right\} \equiv_{t, 1/\delta} \left\{ f \mid f \xleftarrow{s} \text{ELF.Gen}(m, r) \right\},$$

where t and δ are evaluated at n_m . ┘

2.4 Cyclic Group Samplers

We state our bounded-exponent assumptions and our trapdoor hash over a finite cyclic group with a public generator, efficient group operations, and an efficient uniform sampler, whose order need not be computable from the public parameters. Since the order can be hidden, we define the sampler without an exponent space.

DEFINITION 2.7 (Cyclic Group Sampler). A cyclic group sampler is a randomized polynomial-time algorithm $\text{CGen}(\cdot)$ that on input 1^λ outputs either a distinguished failure symbol $\perp$ or parameters pp , such that the probability that $\text{CGen}(1^\lambda)$ outputs $\perp$ is negligible in λ , and such that there exist a positive integer-valued polynomial ℓ_{grp} and a negligible function $\nu_{\mathbb{G}}$ for which every $\lambda \in \mathbb{N}$ and every $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda)) \setminus \{\perp\}$ satisfy the following four conditions.

- **Group.** The parameters pp determine a finite cyclic group $\mathbb{G} = \mathbb{G}(\text{pp})$ of order $\text{ord}(\text{pp}) := |\mathbb{G}| \geq 2^\lambda$ and a generator $g = g(\text{pp})$ of $\mathbb{G}$.
- **Representation.** The parameters, the generator, and every group element have canonical representations of length at most $\ell_{\text{grp}}(\lambda)$.
- **Operations.** The group operation, inversion, and exponentiation by an integer of polynomially bounded bit length take time $\text{poly}(\lambda)$.
- **Sampling.** A randomized algorithm $\text{Samp}_{\mathbb{G}}(\text{pp})$ runs in time $\text{poly}(\lambda)$ and outputs an element of $\mathbb{G}$ or $\perp$, with

$$\Pr[\text{Samp}_{\mathbb{G}}(\text{pp}) = \perp] \leq \nu_{\mathbb{G}}(\lambda),$$

and its output conditioned on $\text{Samp}_{\mathbb{G}}(\text{pp}) \neq \perp$ is uniform on $\mathbb{G}$. ┘

Membership in $\mathbb{G}$ and the order $\text{ord}(\text{pp})$ need not be efficiently computable from pp . In our experiments, we only sample from $\mathbb{G}$ and compute in it, and in our inversion algorithms, we decide membership only in an ambient group that contains $\mathbb{G}$.

Every quantifier over $\text{supp}(\text{CGen}(1^\lambda))$, or over the support of an algorithm that calls CGen , ranges over outputs other than $\perp$. An algorithm that receives $\perp$ from its sampler or as input outputs $\perp$, unless it explicitly outputs a declared fallback object, and a generator with such an explicit fallback branch has the fallback object in its support. We write $G \xleftarrow{s} \mathbb{G}(\text{pp})$ for a call to $\text{Samp}_{\mathbb{G}}(\text{pp})$ and identify a call that does not output $\perp$ with the uniform distribution on $\mathbb{G}$, and an experiment in which such a call outputs $\perp$ outputs $\perp$. By a union bound, the probability that one of polynomially many such calls outputs $\perp$ is negligible, and in each quantitative statement with a target advantage $1/\delta$, we enlarge the threshold by this probability.

2.5 Damgård–Jurik Groups

Modulo the square of an RSA modulus N , the unit group is the direct product of a subgroup with easy discrete logarithms and a subgroup in which computing them is assumed hard. [Pai99] bases Paillier encryption on this decomposition, and [DJ01] generalizes it to higher powers of N . In our construction, the encoded predicate value lies in the easy subgroup, and the shared exponent, which the trapdoor removes, lies in the hard subgroup. We follow [RS21], which computes distributed discrete logarithms without error in this setting, and we use only the modulus N^2 , whose message space is $\mathbb{Z}/N\mathbb{Z}$. For a function $\ell: \mathbb{N} \rightarrow \mathbb{N}$, we write

$$\mathcal{P}_\ell(\lambda) := \left\{ p \in \text{Primes} \mid (p-1)/2 \in \text{Primes}, 2^{\ell(\lambda)-1} < p < 2^{\ell(\lambda)} \right\},$$

and for $p, q \in \mathcal{P}_\ell(\lambda)$, we write $p' := (p-1)/2$, $q' := (q-1)/2$, and $\varphi := (p-1)(q-1)$ for Euler's totient of $N := pq$.

We use the standard setup convention that distinct safe primes of a prescribed bit length can be sampled in probabilistic polynomial time with negligible failure probability. Its unconditional asymptotic validity is open, because the density, and even the infinitude, of safe primes is not known [vzGS13]. We use this convention only to generate the public parameters, and each of our computational assumptions is an assumption about the resulting distribution of moduli. [RS21] also uses safe-prime moduli in its trapdoor-hashing construction, and we realize the convention by a bounded search with $W(\lambda)$ trials for a polynomial W .

For our sampler, we take the safe-prime key generation of [RS21] and the Damgård–Jurik setting of [DJ01] at the modulus N^2 , with three changes: we publish a generator g of the subgroup $\mathbb{G}_N$ of $2N$ -th powers, which we obtain by rejection sampling with the factorization of N , we output the ordered pair (N, g) , and we output $\perp$ when a search exhausts its trials. Since we reject every element that does not generate $\mathbb{G}_N$, the published element always generates $\mathbb{G}_N$.

DEFINITION 2.8 (Damgård–Jurik Sampler). Let ℓ be an efficiently computable positive integer-valued polynomial with $\ell(\lambda) \geq \lambda + 2$ for every $\lambda \in \mathbb{N}$, and let W be a positive integer-valued polynomial such that each search in the first step exhausts its $W(\lambda)$ trials with probability negligible in λ . The Damgård–Jurik sampler $\text{DJGen}(1^\lambda)$ is the following randomized algorithm.

- **Safe primes.** It runs $(p, q) \xleftarrow{\$} \text{SafeGen}_{\ell, W}(1^\lambda)$, which performs two independent searches, each drawing at most $W(\lambda)$ independent uniform odd integers from $(2^{\ell(\lambda)-1}, 2^{\ell(\lambda)})$ and keeping the first one that lies in $\mathcal{P}_\ell(\lambda)$. If either search exhausts its trials or $p = q$, then $\text{SafeGen}_{\ell, W}$ outputs $\perp$, and so does DJGen .
- **Group.** It sets $N := pq$ and $\mathbb{G}_N := \{z^{2N} \mid z \in (\mathbb{Z}/N^2\mathbb{Z})^\times\}$.
- **Generator.** With the factorization of N , it repeats the steps $z \xleftarrow{\$} (\mathbb{Z}/N^2\mathbb{Z})^\times$ and $g \leftarrow z^{2N}$, for at most λ independent trials, until $g^{p'} \neq 1$ and $g^{q'} \neq 1$, and it outputs $\perp$ if all λ trials are rejected.
- **Output.** Otherwise, it outputs $\text{pp} := (N, g)$ and discards the factorization. $\lrcorner$

We use safe primes as in [RS21], so that $\mathbb{G}_N$ is cyclic and $\varphi/4 = p'q'$ is a product of two primes. [DGI⁺19] assumes composite residuosity for a uniform Blum integer and uses only the subgroup decomposition. Under the setup convention, $\text{SafeGen}_{\ell, W}$ outputs $\perp$ with negligible probability, and conditioned on success, its output is uniform among ordered pairs of distinct elements of $\mathcal{P}_\ell(\lambda)$. An inverse-polynomial lower bound on the density of $\mathcal{P}_\ell(\lambda)$ among the odd integers of the interval

suffices to realize the setup convention.

We need the easy subgroup at every modulus $N \geq 2$, including moduli that are not products of two safe primes, so we state it without the subgroup decomposition. [RS21] uses the same isomorphism modulo N^{s+1} for every $s \geq 1$, and we need only $s = 1$.

LEMMA 2.9 (Easy Part at an Arbitrary Modulus). Let $N \geq 2$ be an integer. The map $\exp(x) := 1 + Nx$ is a group isomorphism from $(\mathbb{Z}/N\mathbb{Z})^+$ onto the subgroup $1 + N\mathbb{Z}/N^2\mathbb{Z}$ of $(\mathbb{Z}/N^2\mathbb{Z})^\times$, which has order N , and its inverse $\log$ satisfies $\log(1 + Nx) = x$. Both maps are computable in time $\text{poly}(\log N)$. $\lrcorner$

Proof. In $\mathbb{Z}/N^2\mathbb{Z}$, we have

$$(1 + Nx)(1 + Ny) = 1 + N(x + y) + N^2xy = 1 + N(x + y),$$

so $\exp$ is a homomorphism from $(\mathbb{Z}/N\mathbb{Z})^+$. It is injective, since $1 + Nx = 1 + Ny$ implies $N(x - y) \equiv 0 \pmod{N^2}$ and hence $x = y$ in $\mathbb{Z}/N\mathbb{Z}$. Its image is the set of the N residues congruent to 1 modulo N , and each of them is coprime to N , hence a unit modulo N^2 . We compute $\exp$ with one multiplication and $\log$ with one division by N , in time $\text{poly}(\log N)$. $\blacksquare$

For a successfully sampled modulus, we need the order of $(\mathbb{Z}/N^2\mathbb{Z})^\times$, the cyclicity and order of $\mathbb{G}_N$, uniform sampling from $\mathbb{G}_N$ with the public modulus, and the trivial intersection of $\mathbb{G}_N$ with the easy subgroup. [BG10] shows that $(\mathbb{Z}/N^2\mathbb{Z})^\times$ is the direct product of $1 + N\mathbb{Z}/N^2\mathbb{Z}$ and the subgroup of N -th residues, of coprime orders N and φ , for primes p and q of equal length. We prove the corresponding facts for the subgroup $\mathbb{G}_N$ of $2N$ -th powers and safe primes, and we prove that DJGen is a cyclic group sampler.

LEMMA 2.10 (Structure of the Damgård–Jurik Group). Let ℓ be the polynomial of [DEFINITION 2.8](#), let $p, q \in \mathcal{P}_\ell(\lambda)$ with $p \neq q$, and let $N := pq$ and $M := p'q'$. Then the following hold.

- (a) $|(\mathbb{Z}/N^2\mathbb{Z})^\times| = N\varphi$.
- (b) $\mathbb{G}_N$ is cyclic of order $M = \varphi/4 > 2^{2\ell(\lambda)-4} \geq 2^{2\lambda}$, and the map $z \mapsto z^{2N}$ sends the uniform distribution on $(\mathbb{Z}/N^2\mathbb{Z})^\times$ to the uniform distribution on $\mathbb{G}_N$.
- (c) $\mathbb{G}_N \cap 1 + N\mathbb{Z}/N^2\mathbb{Z} = \{1\}$, and the map $\mathbb{G}_N \times \mathbb{Z}/N\mathbb{Z} \rightarrow (\mathbb{Z}/N^2\mathbb{Z})^\times$, $(c, x) \mapsto c\exp(x)$, is injective.
- (d) Under the setup convention of [DEFINITION 2.8](#), DJGen is a cyclic group sampler in the sense of [DEFINITION 2.7](#). $\lrcorner$

Proof. We compute the image of $z \mapsto z^{2N}$ in each factor of the Chinese-remainder decomposition of $(\mathbb{Z}/N^2\mathbb{Z})^\times$. For [ITEM \(a\)](#) of [LEMMA 2.10](#), by the Chinese remainder theorem, $(\mathbb{Z}/N^2\mathbb{Z})^\times \cong (\mathbb{Z}/p^2\mathbb{Z})^\times \times (\mathbb{Z}/q^2\mathbb{Z})^\times$, and the two factors are cyclic of orders $p(p-1)$ and $q(q-1)$. Hence, $|(\mathbb{Z}/N^2\mathbb{Z})^\times| = pq(p-1)(q-1) = N\varphi$.

For [ITEM \(b\)](#) of [LEMMA 2.10](#), we fix a cyclic group C of order $p(p-1) = 2pp'$. The image of $c \mapsto c^{2N}$ on C is the unique subgroup of order $2pp'/\gcd(2N, 2pp')$, and $\gcd(2N, 2pp') = 2p\gcd(q, p')$. Every element of $\mathcal{P}_\ell(\lambda)$ has bit length $\ell(\lambda)$, so $p' = (p-1)/2 < 2^{\ell(\lambda)-1} < q$. Hence, q and p' are distinct primes, $\gcd(2N, 2pp') = 2p$, and the image has order p' . In the same way, $q' < 2^{\ell(\lambda)-1} < p$, and the image in the second factor has order q' . Since p' and q' are distinct primes, the product of the two images is cyclic of order $p'q' = M$, and $M = \varphi/4$ because $\varphi = 4p'q'$. Since p and q are odd, distinct, and larger than $2^{\ell(\lambda)-1}$, we have $p', q' \geq 2^{\ell(\lambda)-2}$ with at least one strict inequality,

so $M > 2^{2\ell(\lambda)-4} \geq 2^{2\lambda}$ because $\ell(\lambda) \geq \lambda + 2$. By the definition of $\mathbb{G}_N$, the map $z \mapsto z^{2N}$ is a homomorphism from $(\mathbb{Z}/N^2\mathbb{Z})^\times$ onto $\mathbb{G}_N$. All its preimages are cosets of its kernel and have the same size, so it sends the uniform distribution on $(\mathbb{Z}/N^2\mathbb{Z})^\times$ to the uniform distribution on $\mathbb{G}_N$.

For **ITEM (c)** of **LEMMA 2.10**, the orders $M = p'q'$ and $N = pq$ are coprime, because $p', q' < 2^{\ell(\lambda)-1} < p, q$ and all four numbers are prime. By **LEMMA 2.9**, $1 + N\mathbb{Z}/N^2\mathbb{Z}$ is a subgroup of order N . An element of $\mathbb{G}_N \cap 1 + N\mathbb{Z}/N^2\mathbb{Z}$ generates a subgroup whose order divides both M and N , hence equals 1. If $c\exp(x) = c'\exp(x')$ with $c, c' \in \mathbb{G}_N$ and $x, x' \in \mathbb{Z}/N\mathbb{Z}$, then $c'^{-1}c = \exp(x' - x) \in \mathbb{G}_N \cap 1 + N\mathbb{Z}/N^2\mathbb{Z} = \{1\}$, so $c = c'$, and $x = x'$ because $\exp$ is injective.

For **ITEM (d)** of **LEMMA 2.10**, we first analyze the generator step of **DEFINITION 2.8**. Since $\mathbb{G}_N$ is cyclic of order $p'q'$, its elements g with $g^{p'} = 1$ or $g^{q'} = 1$ form its subgroups of orders p' and q' , so $g \in \mathbb{G}_N$ generates $\mathbb{G}_N$ if and only if $g^{p'} \neq 1$ and $g^{q'} \neq 1$. Hence, every accepted g generates $\mathbb{G}_N$. By **ITEM (b)** of **LEMMA 2.10**, each trial samples g uniformly from $\mathbb{G}_N$, and $\mathbb{G}_N$ has $(p' - 1)(q' - 1)$ generators, so each trial is accepted with probability

$$\left(1 - \frac{1}{p'}\right) \left(1 - \frac{1}{q'}\right) \geq \left(1 - \frac{1}{2}\right) \left(1 - \frac{1}{3}\right) = \frac{1}{3},$$

where we use that p' and q' are distinct primes. The λ trials use independent samples z , so they are independent, and all of them are rejected with probability at most $(2/3)^\lambda$.

We verify the conditions of **DEFINITION 2.7** for an output $(N, g) \neq \perp$ of $\text{DJGen}(1^\lambda)$. The group $\mathbb{G}_N$ is cyclic, g generates it, and since $p, q > 2^{\ell(\lambda)-1}$ and $\ell(\lambda) \geq \lambda + 2$, its order satisfies $M = p'q' \geq 2^{2\ell(\lambda)-4} \geq 2^{2\lambda} \geq 2^\lambda$. The sampler $\text{DJGen}(1^\lambda)$ outputs $\perp$ only if $\text{SafeGen}_{\ell, W}$ does, which happens with negligible probability under the setup convention, or if all generator trials are rejected, which happens with probability at most $(2/3)^\lambda$. Every element of $\mathbb{Z}/N^2\mathbb{Z}$ has a canonical representative in $[0..N^2 - 1]$ of length at most $4\ell(\lambda)$ bits, and the group operation, inversion, and exponentiation by an integer of polynomially bounded bit length take time $\text{poly}(\log N) = \text{poly}(\lambda)$.

For the sampling condition, we define $\text{Samp}_{\mathbb{G}}(N, g)$ as follows: it draws $z \xleftarrow{\$} \mathbb{Z}/N^2\mathbb{Z}$, outputs $\perp$ if $\gcd(z, N) \neq 1$, and otherwise outputs z^{2N} . Conditioned on $\gcd(z, N) = 1$, the element z is uniform on $(\mathbb{Z}/N^2\mathbb{Z})^\times$, so the output is uniform on $\mathbb{G}_N$ by **ITEM (b)** of **LEMMA 2.10**. Moreover, since $p, q > 2^{\ell(\lambda)-1}$ and $\ell(\lambda) \geq \lambda + 2$,

$$\Pr[\gcd(z, N) \neq 1 \mid z \xleftarrow{\$} \mathbb{Z}/N^2\mathbb{Z}] = \frac{1}{p} + \frac{1}{q} - \frac{1}{N} < 2^{2-\ell(\lambda)} \leq 2^{-\lambda},$$

so $\text{Samp}_{\mathbb{G}}$ runs in strict polynomial time and satisfies the sampling condition with $\nu_{\mathbb{G}}(\lambda) := 2^{-\lambda}$. ■

To decode, we must recover the easy part of a product $c\exp(x)$ for an unknown $c \in \mathbb{G}_N$. **[BGI16a]** computes distributed discrete logarithms in prime-order groups with inverse-polynomial error. **[OSY21]** and **[RS21]** independently compute them without error, over Paillier and Damgård–Jurik groups, respectively. Following **[RS21]**, we recover the easy part of a unit by dividing it by its reduction modulo N . The resulting map equals the distance function of **[RS21]** at $s = 1$ and the output of the algorithm of **[OSY21]**, and we define it at every modulus $N \geq 2$.

DEFINITION 2.11 (Distributed Discrete Logarithm). Let $N \geq 2$ be an integer. For $c \in (\mathbb{Z}/N^2\mathbb{Z})^\times$, let $w \in [0..N - 1]$ be the least nonnegative representative of c modulo N , viewed as an element of $\mathbb{Z}/N^2\mathbb{Z}$, and define $\text{DDLog}_N(c) := \log(c \cdot w^{-1}) \in \mathbb{Z}/N\mathbb{Z}$. ■

Since $\gcd(w, N) = \gcd(c, N) = 1$, we have $w \in [1..N - 1]$, and w is a unit modulo N^2 because N

and N^2 have the same prime divisors. The element cw^{-1} is congruent to 1 modulo N , so it lies in $1 + N\mathbb{Z}/N^2\mathbb{Z}$, the domain of $\log$ in [LEMMA 2.9](#), and DDLog_N is well defined and computable in time $\text{poly}(\log N)$ at every modulus $N \geq 2$.

Since $\text{ord}(\mathbb{G}_N)$ is not public, we sample exponents from an interval whose length is a power of two and exceeds the modulus by a factor of at least 2^λ , and we bound once the statistical distance of the reduced exponent from uniform. [\[BG10\]](#) bounds this distance by $M'/2^{L_{\text{exp}}}$, and we prove the sharper bound $M'/(4 \cdot 2^{L_{\text{exp}}})$, which is at most $2^{-\lambda-2}$ for $M' \leq N$.

LEMMA 2.12 (Exponent Smoothing). Let $N \geq 2$ and $\lambda \in \mathbb{N}$, let $L_{\text{exp}}(\lambda, N) := \lceil \log_2 N \rceil + \lambda$, and let ρ be uniform on $[0..2^{L_{\text{exp}}(\lambda, N)} - 1]$. For every integer M' with $1 \leq M' \leq N$, the statistical distance between $\rho \bmod M'$ and the uniform distribution on $\mathbb{Z}/M'\mathbb{Z}$ is at most $2^{-\lambda-2}$. $\lrcorner$

Proof. We count the preimages of each residue modulo M' . We write $L_{\text{exp}} := L_{\text{exp}}(\lambda, N)$ and $2^{L_{\text{exp}}} = aM' + b$ with $a \in \mathbb{N}$ and $b \in [0..M' - 1]$. Exactly b residues have $a + 1$ preimages and the remaining $M' - b$ residues have a preimages, so

$$\begin{aligned} (a+1)2^{-L_{\text{exp}}} - \frac{1}{M'} &= \frac{M'(a+1) - 2^{L_{\text{exp}}}}{2^{L_{\text{exp}}} M'} = \frac{M' - b}{2^{L_{\text{exp}}} M'}, \\ \frac{1}{M'} - a2^{-L_{\text{exp}}} &= \frac{2^{L_{\text{exp}}} - aM'}{2^{L_{\text{exp}}} M'} = \frac{b}{2^{L_{\text{exp}}} M'}. \end{aligned}$$

Both differences are nonnegative, and the b residues of the first kind and the $M' - b$ residues of the second kind contribute equally to the statistical distance. Therefore, the distance is

$$\frac{b(M' - b)}{2^{L_{\text{exp}}} M'} \leq \frac{M'}{4 \cdot 2^{L_{\text{exp}}}} \leq \frac{N}{4 \cdot 2^{\lceil \log_2 N \rceil + \lambda}} \leq 2^{-\lambda-2},$$

where we use $b(M' - b) \leq (M'/2)^2$ in the first inequality, $M' \leq N$ in the second, and $N \leq 2^{\lceil \log_2 N \rceil}$ in the last. $\blacksquare$

In our proof of function privacy, we use the decisional composite residuosity assumption of [\[Pai99\]](#), the indistinguishability of a uniform unit modulo N^2 from a uniform N -th power, which we state for the moduli of DJGen .

DEFINITION 2.13 (Decisional Composite Residuosity). The decisional composite residuosity (DCR) assumption holds for $\text{DJGen}(\cdot)$ if

$$\left\{ (N, z) \mid \begin{array}{l} (N, g) \xleftarrow{\$} \text{DJGen}(1^\lambda), \\ z \xleftarrow{\$} (\mathbb{Z}/N^2\mathbb{Z})^\times \end{array} \right\}_{\lambda \in \mathbb{N}} \equiv_c \left\{ (N, z^N) \mid \begin{array}{l} (N, g) \xleftarrow{\$} \text{DJGen}(1^\lambda), \\ z \xleftarrow{\$} (\mathbb{Z}/N^2\mathbb{Z})^\times \end{array} \right\}_{\lambda \in \mathbb{N}}.$$

3 Image-Size Bounds and Small-Exponent Assumptions

Every publicly evaluable ELF has a lossy image of size $\Omega(\delta k^2)$ against observers that make k evaluations and have advantage at most $1/\delta$, and in the generic group model, an algorithm that makes T group operations distinguishes d powers with exponents from an interval of length γ from uniform group elements with advantage $\mathcal{O}((T + d)^2/\gamma)$ when the smallest prime factor of the group order is large. Since we can search every polynomially bounded interval in polynomial time, the

interval threshold in the bounded-exponent assumption that we use depends on the size bound and the target advantage of the adversary.

3.1 A Lower Bound on the Image Size

Since evaluation of an ELF is public, an observer can evaluate the function on independent uniform inputs and test whether two distinct inputs collide, so the image parameter cannot be arbitrarily small. We make quantitative the observation of [Zha16] that $\mathcal{O}(\sqrt{r})$ evaluations detect an image of size r . [Fel68] treats uniform outputs, [Mun77, Ber80] prove that a nonuniform distribution on r points has collision probability at least that of the uniform distribution on r points, and [BK04] gives the corresponding analysis for hash functions. With k evaluations, the collision probability in an image of size r is of order k^2/r , whereas distinct uniform inputs from $[m]$ collide with probability $\mathcal{O}(k^2/m)$, and from these two bounds we determine the smallest image size that an ELF can hide, independently of the algebraic assumption used to construct it.

Our starting point is the birthday bound of [Fel68] for uniform outputs and its extension to nonuniform outputs [Mun77, Ber80]. We adapt it to a comparison with an injective oracle on a domain of size m , which adds the term $k(k-1)/(2m)$ for repeated inputs, and we use a distinguisher that depends only on m and k , so that the same algorithm applies to every oracle, including a sampled function.

THEOREM 3.1 (Birthday Collision Bound). Let $m, k \in \mathbb{N}$ with $m \geq 1$ and $k \geq 2$, let $\mathcal{Y}$ be a set, and let $\mathcal{A}_{m,k}$ be the oracle algorithm that samples $(X_1, \dots, X_k) \xleftarrow{\$} [m]^k$, queries its oracle at $X_1, \dots, X_k$, receives the answers $Y_1, \dots, Y_k$, and returns 1 if and only if $X_1, \dots, X_k$ are pairwise distinct and $Y_1, \dots, Y_k$ are not.

(a) Every injective function $f_{\text{inj}}: [m] \rightarrow \mathcal{Y}$ satisfies

$$\Pr[\mathcal{A}_{m,k}^{f_{\text{inj}}} = 1] = 0.$$

(b) Every $r \in \mathbb{N}_{>0}$ and every function $f: [m] \rightarrow \mathcal{Y}$ with $|\text{Im}(f)| \leq r$ satisfy

$$\Pr[\mathcal{A}_{m,k}^f = 1] \geq 1 - \exp\left(-\frac{k(k-1)}{2r}\right) - \frac{k(k-1)}{2m}.$$

□

Proof. We bound the probability that the outputs collide from below by a comparison with the uniform distribution on the image, and we bound the probability of repeated inputs by a union bound.

We write $\mathbf{E}_{\text{in}} := \{\exists i < j : X_i = X_j\}$ and $\mathbf{E}_{\text{out}} := \{\exists i < j : Y_i = Y_j\}$, so that $\mathcal{A}_{m,k}$ returns 1 exactly on $\mathbf{E}_{\text{coll}} := \mathbf{E}_{\text{out}} \setminus \mathbf{E}_{\text{in}}$. If the oracle is injective, then $\mathbf{E}_{\text{out}} \subseteq \mathbf{E}_{\text{in}}$, so $\mathbf{E}_{\text{coll}} = \emptyset$, and **ITEM (a)** of **THEOREM 3.1** holds.

For **ITEM (b)** of **THEOREM 3.1**, we write $S := \text{Im}(f)$, $s := |S| \leq r$, $\mathbf{E}_{\text{distinct}} := \neg \mathbf{E}_{\text{out}}$, and

$$p_y := \Pr[f(X_1) = y]$$

for every $y \in S$. If $s < k$, then two of the k outputs coincide, so $\mathbf{E}_{\text{distinct}} = \emptyset$. We suppose that $s \geq k$ and write e_k for the k th elementary symmetric polynomial. Since $X_1, \dots, X_k$ are independent

and the p_y sum to one, we obtain from Maclaurin's inequality [Mac29, Ber80]

$$\Pr[E_{\text{distinct}}] = k! e_k((p_y)_{y \in S}) \leq k! \binom{s}{k} \left(\frac{1}{s} \sum_{y \in S} p_y \right)^k = k! \binom{s}{k} s^{-k}.$$

Since $s \geq k$ and $r \geq s$, every factor $1 - i/s$ and $1 - i/r$ with $0 \leq i < k$ is nonnegative. We use $s \leq r$ and $1 - z \leq e^{-z}$ for every real z , and obtain

$$\begin{aligned} k! \binom{s}{k} s^{-k} &= \prod_{i=0}^{k-1} \left(1 - \frac{i}{s} \right) \leq \prod_{i=0}^{k-1} \left(1 - \frac{i}{r} \right) \\ &\leq \exp \left(-\frac{1}{r} \sum_{i=0}^{k-1} i \right) = \exp \left(-\frac{k(k-1)}{2r} \right). \end{aligned}$$

The resulting bound also holds when $s < k$, since then $E_{\text{distinct}} = \emptyset$. By a union bound over the $\binom{k}{2}$ unordered pairs of inputs,

$$\Pr[E_{\text{in}}] \leq \sum_{1 \leq i < j \leq k} \Pr[X_i = X_j] = \binom{k}{2} \frac{1}{m} = \frac{k(k-1)}{2m}.$$

Finally, $E_{\text{out}} = E_{\text{coll}} \sqcup (E_{\text{out}} \cap E_{\text{in}})$ and $E_{\text{out}} \cap E_{\text{in}} \subseteq E_{\text{in}}$, so

$$\begin{aligned} \Pr[\mathcal{A}_{m,k}^f = 1] &= \Pr[E_{\text{coll}}] \\ &= \Pr[E_{\text{out}}] - \Pr[E_{\text{out}} \cap E_{\text{in}}] \\ &\geq 1 - \Pr[E_{\text{distinct}}] - \Pr[E_{\text{in}}] \\ &\geq 1 - \exp \left(-\frac{k(k-1)}{2r} \right) - \frac{k(k-1)}{2m}. \end{aligned}$$

Hence, **ITEM (b)** of **THEOREM 3.1** holds. ■

We invert **THEOREM 3.1** and obtain a lower bound on the image size of every function that k evaluations do not distinguish from an injective function.

COROLLARY 3.2 (Quadratic Lower Bound on the Image Size). Let $m, r, k \in \mathbb{N}$ and $\delta \in \mathbb{R}$ with $m, r \geq 1$, $k \geq 2$, $\delta \geq 3$, and $m \geq \delta k(k-1)$, let $f: [m] \rightarrow \mathcal{Y}$ satisfy $|\text{Im}(f)| \leq r$, and let $f_{\text{inj}}: [m] \rightarrow \mathcal{Y}$ be injective. Suppose that every oracle algorithm $\mathcal{A}$ that makes at most k oracle queries satisfies

$$\left| \Pr[\mathcal{A}^f = 1] - \Pr[\mathcal{A}^{f_{\text{inj}}} = 1] \right| \leq \frac{1}{\delta}.$$

Then

$$r \geq \frac{k(k-1)}{2 \ln(1/(1-3/(2\delta)))} \geq \frac{(2\delta-3)k(k-1)}{6} \geq \frac{\delta k^2}{12}.$$

Proof. We apply the hypothesis to $\mathcal{A}_{m,k}$ of **THEOREM 3.1**, solve the resulting inequality for r , and bound the logarithm. ┐

The algorithm $\mathcal{A}_{m,k}$ makes k oracle queries, so by **ITEMS (a) and (b)** of **THEOREM 3.1** and

the hypothesis, $1 - \exp(-k(k-1)/(2r)) - k(k-1)/(2m) \leq 1/\delta$. Since $m \geq \delta k(k-1)$, we have $k(k-1)/(2m) \leq 1/(2\delta)$, so

$$1 - \exp\left(-\frac{k(k-1)}{2r}\right) \leq \frac{1}{\delta} + \frac{k(k-1)}{2m} \leq \frac{3}{2\delta}.$$

We write $z := 3/(2\delta)$. Since $\delta \geq 3$, we have $0 < z \leq 1/2$, and we take logarithms in $\exp(-k(k-1)/(2r)) \geq 1 - z$ and obtain

$$r \geq \frac{k(k-1)}{2 \ln(1/(1-z))}.$$

Since $\ln(1+x) \leq x$ for every $x > -1$,

$$\ln\left(\frac{1}{1-z}\right) = \ln\left(1 + \frac{z}{1-z}\right) \leq \frac{z}{1-z} = \frac{3}{2\delta-3},$$

and we obtain $r \geq (2\delta-3)k(k-1)/6$. Since $2\delta-3 \geq \delta$ and $k(k-1) \geq k^2/2$, we also obtain $r \geq \delta k^2/12$. $\blacksquare$

COROLLARY 3.2 concerns one pair of functions, whereas the mode indistinguishability of an ELF compares two generation algorithms. Since the algorithm $\mathcal{A}_{m,k}$ of **THEOREM 3.1** depends only on m and k , we can run it on a sampled public description and transfer the bound to the threshold Q of **DEFINITION 2.6**.

PROPOSITION 3.3 (Lower Bound on the ELF Threshold). Let $\text{ELF} = (\text{ELF.Gen}, \text{ELF.Eval})$ satisfy **DEFINITION 2.6**, and for every $m \geq 2$, write $n_m := \lceil \log_2 m \rceil$. Let t be a polynomial and c_{eval} a positive polynomial such that, for every $m \geq 2$ and every $k \in \mathbb{N}$ with $2 \leq k \leq t(n_m)$, the algorithm $\mathcal{A}_{m,k}$ of **THEOREM 3.1**, with its oracle queries replaced by evaluations of a public description of ELF, is a probabilistic circuit of size at most $k c_{\text{eval}}(n_m)$. Let δ be a positive polynomial with $\delta(n) \geq 3$ for every n , let Q be a polynomial witnessing mode indistinguishability for t and δ with $Q(n) \in \mathbb{N}_{>0}$ for every sufficiently large n , and define

$$k(n) := \min \left\{ \lfloor t(n) \rfloor, \left\lfloor \frac{t(n)}{c_{\text{eval}}(n)} \right\rfloor \right\}.$$

Then $Q(n_m) \geq \delta(n_m)k(n_m)^2/24$ for every sufficiently large m with $k(n_m) \geq 2$. $\dashv$

Proof. Since $\mathcal{A}_{m,k}$ depends only on m and k , we can run it on a sampled public description and average its advantage over the generation coins. Since the two failure events of the modes are negligible, we absorb them into the target advantage, and we then repeat the algebra of the proof of **COROLLARY 3.2** for $r = Q(n_m)$.

For every $m \geq 2$ with $Q(n_m) \in [m]$, we write $\varepsilon_{\text{inj}}(m)$ for the probability that $f \xleftarrow{\$} \text{ELF.Gen}(m, m)$ is not injective and $\varepsilon_{\text{los}}(m)$ for the probability that $f \xleftarrow{\$} \text{ELF.Gen}(m, Q(n_m))$ satisfies $|\text{Im}(f)| > Q(n_m)$. We choose m_0 such that every $m \geq m_0$ satisfies the following conditions.

- (i) The mode-indistinguishability clause of **DEFINITION 2.6** holds at m , which is the case from some m on by the asymptotic quantifier of that clause.
- (ii) $Q(n_m) \in \mathbb{N}_{>0}$, which is the case from some m on because n_m grows without bound.
- (iii) $Q(n_m) \leq m$, which is the case from some m on because $m > 2^{n_m-1}$ and $Q(n_m)$ is polynomial in n_m .

- (iv) $\varepsilon_{\text{los}}(m) + \varepsilon_{\text{inj}}(m) \leq 1/(2\delta(n_m))$, which is the case from some m on because $\varepsilon_{\text{los}}(m)$ and $\varepsilon_{\text{inj}}(m)$ are negligible in n_m by the injective-mode and lossy-mode clauses of [DEFINITION 2.6](#), while δ is polynomial.
- (v) $k(n_m)(k(n_m) - 1)/(2m) \leq 1/(2\delta(n_m))$, which is the case from some m on because k and δ are polynomially bounded and $m > 2^{n_m-1}$.

We fix $m \geq m_0$ with $k(n_m) \geq 2$, abbreviate $n := n_m$ and $k := k(n)$, and set $r := Q(n)$. By [ITEMS \(ii\)](#) and [\(iii\)](#), we have $r \in [Q(n)..m] \subseteq [m]$, so $\text{ELF.Gen}(m, r)$ is defined, and by [ITEM \(i\)](#), mode indistinguishability holds at the image parameter r . We take $\mathcal{A}$ to be $\mathcal{A}_{m,k}$ with its oracle queries replaced by evaluations of the public description. Since k is the smaller of two floors, k is an integer with $k \leq t(n)$ and $k c_{\text{eval}}(n) \leq t(n)$. By the first inequality, k lies in the range where the cost hypothesis applies, and by the second, $\mathcal{A}$ is a probabilistic circuit of size at most $t(n)$, so mode indistinguishability holds for $\mathcal{A}$.

We write $B := 1 - \exp(-k(k-1)/(2r))$. By [ITEMS \(a\)](#) and [\(b\)](#) of [THEOREM 3.1](#), every f with $|\text{Im}(f)| \leq r$ and every injective f' satisfy

$$\Pr[\mathcal{A}(f) = 1] \geq B - \frac{k(k-1)}{2m}, \quad \Pr[\mathcal{A}(f') = 1] = 0.$$

After we average over the generation coins, we obtain

$$\begin{aligned} \Pr[\mathcal{A}(f) = 1 \mid f \xleftarrow{\$} \text{ELF.Gen}(m, r)] &\geq B - \frac{k(k-1)}{2m} - \varepsilon_{\text{los}}(m), \\ \Pr[\mathcal{A}(f) = 1 \mid f \xleftarrow{\$} \text{ELF.Gen}(m, m)] &\leq \varepsilon_{\text{inj}}(m). \end{aligned}$$

By mode indistinguishability, the difference of these two probabilities is at most $1/\delta(n)$. Therefore, by [ITEMS \(iv\)](#) and [\(v\)](#),

$$B \leq \frac{1}{\delta(n)} + \frac{k(k-1)}{2m} + \varepsilon_{\text{los}}(m) + \varepsilon_{\text{inj}}(m) \leq \frac{2}{\delta(n)}.$$

We write $z := 2/\delta(n)$, so that $0 < z \leq 2/3$ because $\delta(n) \geq 3$. As in the proof of [COROLLARY 3.2](#), the inequality $\exp(-k(k-1)/(2r)) \geq 1 - z$ implies $r \geq k(k-1)/(2\ln(1/(1-z)))$, and $\ln(1/(1-z)) \leq z/(1-z) = 2/(\delta(n)-2)$, so $r \geq (\delta(n)-2)k(k-1)/4$. Finally, $\delta(n)-2 \geq \delta(n)/3$, and since $k \geq 2$, we have $k(k-1) \geq k^2/2$, so $r \geq \delta(n)k^2/24$. Since $r = Q(n)$, the claim holds. $\blacksquare$

3.2 Small-Exponent Assumptions with a Fixed Bound

By [COROLLARY 3.2](#), an image that k evaluations distinguish from an injective one with advantage at most $1/\delta$ has size at least $\delta k^2/12$, so the lossy mode needs bounded exponents from an interval whose length is polynomial and grows with the resources of the observer. We first state the bounded-exponent assumptions for a fixed interval and prove that every polynomially bounded fixed interval violates them.

[\[vOW96\]](#) studies short exponents in Diffie–Hellman key agreement for faster exponentiation, and [\[PS98\]](#) uses them for pseudorandom generation. With baby-step–giant-step [\[Sha71\]](#) or the kangaroo method [\[Pol78, Pol00\]](#), we recover an exponent from an interval of length γ with $\mathcal{O}(\sqrt{\gamma})$ group operations, and [\[Pol78\]](#) gives the rho method, a generic collision algorithm over the whole exponent space. [\[vOW96, PS98\]](#) take the interval large enough for generic security and also study

bounded exponents modulo a composite, where baby-step-giant-step [Sha71] and the kangaroo method [Pol78, Pol00] recover bounded exponents in the same number of group operations. We are not aware of an attack on the subgroup of DEFINITION 2.8 that is faster than these square-root algorithms.

We sample bounded exponents from $[\gamma] = \{1, \dots, \gamma\}$. The interval $[0.. \gamma - 1]$ yields equivalent search and decision problems, since multiplication by the generator g maps g^u to g^{u+1} and multiplication by g^{-1} inverts this map. The order of the group of DEFINITION 2.8 is hidden, so we state every assumption of this subsection for an arbitrary cyclic group sampler $\text{CGen}(\cdot)$ in the sense of DEFINITION 2.7, with a uniform group element as the alternative to a bounded power. Since g generates $\mathbb{G}(\text{pp})$, a uniform group element is distributed as g^u for $u \xleftarrow{\$} \mathbb{Z}_{\text{ord}(\text{pp})}$, so this alternative agrees with a uniform exponent whenever $\text{ord}(\text{pp})$ is public.

The following search assumption adapts the short-exponent discrete-logarithm problem of [vOW96, PS98] to a cyclic group sampler with a possibly hidden order and to exponents from $[\gamma(\lambda)]$.

DEFINITION 3.4 (Bounded SEDL). Let $\gamma: \mathbb{N} \rightarrow \mathbb{N}_{>0}$ satisfy $\gamma(\lambda) < \text{ord}(\text{pp})$ for every sufficiently large λ and every $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$. For every adversary $\mathcal{A}$, define

$$\text{Adv}_{\text{CGen}(\cdot), \mathcal{A}}^{\text{SEDL}_\gamma}(\lambda) := \Pr \left[\hat{s} = s \mid \begin{array}{l} \text{pp} \xleftarrow{\$} \text{CGen}(1^\lambda), \\ s \xleftarrow{\$} [\gamma(\lambda)], \\ \hat{s} \xleftarrow{\$} \mathcal{A}(1^\lambda, \text{pp}, \gamma(\lambda), g^s) \end{array} \right].$$

The γ -bounded small-exponent discrete-logarithm assumption SEDL_γ holds for $\text{CGen}(\cdot)$ if, for every PPT adversary $\mathcal{A}$, the function $\text{Adv}_{\text{CGen}(\cdot), \mathcal{A}}^{\text{SEDL}_\gamma}$ is negligible. $\lrcorner$

Since $\gamma(\lambda) < \text{ord}(\text{pp})$, the map $s \mapsto g^s$ is injective on $[\gamma(\lambda)]$, so g^s determines s . In the decisional version, the adversary must distinguish a bounded power from a uniform group element, and a distinguisher need not recover the exponent. Decisional short-exponent problems appear as the c -decisional short discrete logarithm problem of [AFLT12], in which the exponent has c bits, and we state ours for the interval $[\gamma]$ and a uniform group element as the alternative.

DEFINITION 3.5 (Bounded DSEDL). For every $\lambda \in \mathbb{N}$ and every positive integer γ smaller than the order of every group in $\text{supp}(\text{CGen}(1^\lambda))$, define

$$\begin{aligned} \mathfrak{D}_{0, \lambda, \gamma}^{\text{DSEDL}} &:= \left\{ (1^\lambda, \text{pp}, \gamma, g^s) \mid \begin{array}{l} \text{pp} \xleftarrow{\$} \text{CGen}(1^\lambda), \\ s \xleftarrow{\$} [\gamma] \end{array} \right\}, \\ \mathfrak{D}_{1, \lambda, \gamma}^{\text{DSEDL}} &:= \left\{ (1^\lambda, \text{pp}, \gamma, G) \mid \begin{array}{l} \text{pp} \xleftarrow{\$} \text{CGen}(1^\lambda), \\ G \xleftarrow{\$} \mathbb{G}(\text{pp}) \end{array} \right\}. \end{aligned}$$

Let $\gamma: \mathbb{N} \rightarrow \mathbb{N}_{>0}$ satisfy $\gamma(\lambda) < \text{ord}(\text{pp})$ for every sufficiently large λ and every $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$. The γ -bounded decisional small-exponent discrete-logarithm assumption DSEDL_γ holds for $\text{CGen}(\cdot)$ if

$$\left(\mathfrak{D}_{0, \lambda, \gamma(\lambda)}^{\text{DSEDL}} \right)_{\lambda \in \mathbb{N}} \equiv_c \left(\mathfrak{D}_{1, \lambda, \gamma(\lambda)}^{\text{DSEDL}} \right)_{\lambda \in \mathbb{N}}.$$

A recovered exponent is a witness that we can verify by exponentiation, so DSEDL_γ implies SEDL_γ whenever $[\gamma(\lambda)]$ has negligible density in the group. The implication from SEDL_γ to DSEDL_γ would require a search-to-decision reduction.

THEOREM 3.6 (DSED L_γ Implies SED L_γ). Let $\gamma: \mathbb{N} \rightarrow \mathbb{N}_{>0}$, and suppose that there is a negligible function ε_0 such that $\gamma(\lambda)/\text{ord}(\text{pp}) \leq \varepsilon_0(\lambda)$ for every sufficiently large λ and every $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$. If DSED L_γ holds for $\text{CGen}(\cdot)$, then SED L_γ holds for $\text{CGen}(\cdot)$. $\square$

Proof. We verify a recovered exponent by exponentiation, and on a uniform challenge, the check passes only if the challenge is a power with exponent in $[\gamma(\lambda)]$, which has probability at most $\varepsilon_0(\lambda)$.

We prove the contrapositive and suppose that SED L_γ does not hold, so that there is a PPT adversary $\mathcal{A}$ whose success function $\alpha(\lambda) := \text{Adv}_{\text{CGen}(\cdot), \mathcal{A}}^{\text{SED}_\gamma}(\lambda)$ is not negligible. We construct a PPT distinguisher $\mathcal{B}$ that returns 0 on the failure symbol. On input $(1^\lambda, \text{pp}, \gamma(\lambda), h) \neq \perp$, it samples $\hat{s} \xleftarrow{\$} \mathcal{A}(1^\lambda, \text{pp}, \gamma(\lambda), h)$ and returns $1[\hat{s} \in [\gamma(\lambda)] \wedge g^{\hat{s}} = h]$.

If $h = g^s$ for $s \xleftarrow{\$} [\gamma(\lambda)]$, then $\mathcal{B}$ accepts whenever $\mathcal{A}$ returns s , hence

$$\Pr \left[\mathcal{B}(1^\lambda, \text{pp}, \gamma(\lambda), g^s) = 1 \mid \begin{array}{l} \text{pp} \xleftarrow{\$} \text{CGen}(1^\lambda), \\ s \xleftarrow{\$} [\gamma(\lambda)] \end{array} \right] \geq \alpha(\lambda).$$

For a uniform group element $G \xleftarrow{\$} \mathbb{G}(\text{pp})$, acceptance implies that $G = g^s$ for one of the $\gamma(\lambda)$ integers $s \in [\gamma(\lambda)]$. After we condition on parameters $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$, the acceptance probability is at most $\gamma(\lambda)/\text{ord}(\text{pp})$, and we average over the sampler and obtain, for every sufficiently large λ ,

$$\Pr \left[\mathcal{B}(1^\lambda, \text{pp}, \gamma(\lambda), G) = 1 \mid \begin{array}{l} \text{pp} \xleftarrow{\$} \text{CGen}(1^\lambda), \\ G \xleftarrow{\$} \mathbb{G}(\text{pp}) \end{array} \right] \leq \varepsilon_0(\lambda).$$

Hence, the distinguishing advantage of $\mathcal{B}$ is at least $\alpha(\lambda) - \varepsilon_0(\lambda)$. Since α is not negligible, there are a positive polynomial p and an infinite set $I \subseteq \mathbb{N}$ with $\alpha(\lambda) \geq 1/p(\lambda)$ for every $\lambda \in I$, and since ε_0 is negligible, $\varepsilon_0(\lambda) \leq 1/(2p(\lambda))$ for every sufficiently large λ . Hence, $\mathcal{B}$ has advantage at least $1/(2p(\lambda))$ for infinitely many λ , which is impossible under DSED L_γ , so SED L_γ holds. $\blacksquare$

For a polynomially bounded γ , the decisional assumption is false. We write $v := \lceil \sqrt{\gamma(\lambda)} \rceil$, and with baby-step-giant-step [Sha71] restricted to $[\gamma(\lambda)]$, we compute g^j for $j \in [0..v]$ and hg^{-iv} for $i \in [v]$ from a challenge $h \neq \perp$, and we accept if and only if $hg^{-iv} = g^j$ for some $i \in [0..v]$ and $j \in [0..v-1]$ with $iv + j \in [\gamma(\lambda)]$. Every $s \in [\gamma(\lambda)]$ satisfies $s \leq v^2$, so by Euclidean division, $s = iv + j$ with $i \in [0..v]$ and $j \in [0..v-1]$, and we accept every bounded power. Conversely, after we condition on $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$, we accept a uniform $G \xleftarrow{\$} \mathbb{G}(\text{pp})$ only if $G \in \{g^s \mid s \in [\gamma(\lambda)]\}$, which has probability $\gamma(\lambda)/\text{ord}(\text{pp}) \leq \gamma(\lambda)2^{-\lambda}$ by DEFINITION 2.7. Hence, we distinguish the two ensembles of DEFINITION 3.5 with $\mathcal{O}(\sqrt{\gamma(\lambda)})$ group operations and advantage at least $1 - \gamma(\lambda)2^{-\lambda}$ minus the negligible failure probability of $\text{CGen}(1^\lambda)$, so DSED L_γ is false for every polynomially bounded γ against PPT adversaries.

3.3 Tunable Small-Exponent Assumptions

Since DSED L_γ is false for every polynomially bounded γ , we use the quantifier order of [Zha16] for the extremely lossy construction: we first fix the circuit-size bound and the target advantage of the adversary and then choose a sufficiently large polynomial interval. Hence, in tunable DSED L , the interval threshold Q depends on the size bound p and the target advantage $1/\delta$. The definition follows the short-exponent assumptions of [vOW96, PS98], except that the alternative is a uniform

group element and the interval threshold is chosen after the size bound and the target advantage, as in [Zha16]. The exponent bound γ is part of both distributions, so the adversary knows it.

DEFINITION 3.7 (Tunable DSEDL). Let $\mathfrak{D}_{0,\lambda,\gamma}^{\text{DSEDL}}$ and $\mathfrak{D}_{1,\lambda,\gamma}^{\text{DSEDL}}$ be the distributions of [DEFINITION 3.5](#). We say that $\text{CGen}(\cdot)$ satisfies tunable decisional small-exponent discrete logarithm if, for every polynomial p and every positive polynomial δ , there exist a polynomial Q and $\lambda_0 \in \mathbb{N}$ such that $\mathfrak{D}_{0,\lambda,\gamma}^{\text{DSEDL}} \equiv_{p,1/\delta} \mathfrak{D}_{1,\lambda,\gamma}^{\text{DSEDL}}$ for every $\lambda \geq \lambda_0$ and every $\gamma \in \mathbb{N}$ satisfying $Q(\lambda) \leq \gamma < \text{ord}(\text{pp})$ for all $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$. $\dashv$

We can sample both distributions of [DEFINITION 3.7](#) from public parameters, so we can check a claimed violation of tunable DSEDL at given λ and γ by sampling. In the sense of [Nao03, GW11], an assumption is falsifiable when an efficient algorithm can check a claimed violation of it, and in the following lemma, we give such an algorithm for tunable DSEDL.

LEMMA 3.8 (Falsifiability of Tunable DSEDL). Let $\text{CGen}(\cdot)$ be a cyclic group sampler, let $\lambda \in \mathbb{N}$, and let γ be a positive integer smaller than the order of every group in $\text{supp}(\text{CGen}(1^\lambda))$. For a probabilistic circuit C and $b \in \{0, 1\}$, write

$$q_b(C) := \Pr \left[C(X) = 1 \mid X \xleftarrow{\$} \mathfrak{D}_{b,\lambda,\gamma}^{\text{DSEDL}} \right].$$

There is a probabilistic algorithm $\mathcal{V}$ that, on input $(1^\lambda, \gamma, 1^d, C)$ with $d \in \mathbb{N}_{>0}$, runs in time polynomial in λ , d , and the size of C , returns a bit, and satisfies the following.

- (a) If $|q_0(C) - q_1(C)| \geq 2/d$, then $\mathcal{V}$ returns 1 with probability at least $1 - 2^{-\lambda}$.
- (b) If $|q_0(C) - q_1(C)| \leq 1/d$, then $\mathcal{V}$ returns 1 with probability at most $2^{-\lambda}$. $\dashv$

Proof. We estimate both acceptance probabilities empirically and compare the difference of the estimates with $3/(2d)$, the midpoint between the two cases.

We write $k := 8d^2(\lambda + 2)$. For each $b \in \{0, 1\}$, the algorithm $\mathcal{V}$ draws k independent samples $X_{b,1}, \dots, X_{b,k}$ of $\mathfrak{D}_{b,\lambda,\gamma}^{\text{DSEDL}}$: it runs $\text{CGen}(1^\lambda)$ and then either samples $s \xleftarrow{\$} [\gamma]$ and computes g^s when $b = 0$, or calls $\text{Samp}_{\mathbb{G}}(\text{pp})$ when $b = 1$, with the propagation convention of [§2.4](#) on failures. It runs C on every sample with fresh coins, computes the empirical frequencies

$$\hat{q}_b := \frac{1}{k} \sum_{i=1}^k C(X_{b,i}),$$

and returns 1 if and only if $|\hat{q}_0 - \hat{q}_1| > 3/(2d)$. By [DEFINITION 2.7](#), every group in $\text{supp}(\text{CGen}(1^\lambda))$ has at most $2^{\ell_{\text{grp}}(\lambda)+1}$ elements, so γ has at most $\ell_{\text{grp}}(\lambda) + 1$ bits, and each of the $2k = 16d^2(\lambda + 2)$ samples costs time $\text{poly}(\lambda)$. With the $2k$ evaluations of C , the running time is polynomial in λ , d , and the size of C .

For each b , the samples and the coins of C are independent, so $C(X_{b,1}), \dots, C(X_{b,k})$ are independent bits with mean $q_b(C)$. By Hoeffding's inequality [[Hoe63](#)],

$$\Pr \left[|\hat{q}_b - q_b(C)| \geq \frac{1}{4d} \right] \leq 2 \exp \left(-\frac{2k}{16d^2} \right) = 2e^{-(\lambda+2)} \leq 2^{-\lambda-1}.$$

By a union bound over $b \in \{0, 1\}$, the inequality $|\hat{q}_b - q_b(C)| < 1/(4d)$ holds for both b with

probability at least $1 - 2^{-\lambda}$. On this event, we obtain from the triangle inequality

$$|\hat{q}_0 - \hat{q}_1| - |q_0(C) - q_1(C)| < \frac{1}{2d}.$$

If $|q_0(C) - q_1(C)| \geq 2/d$, then $|\hat{q}_0 - \hat{q}_1| > 2/d - 1/(2d) = 3/(2d)$ on this event, and $\mathcal{V}$ returns 1. If $|q_0(C) - q_1(C)| \leq 1/d$, then $|\hat{q}_0 - \hat{q}_1| < 1/d + 1/(2d) = 3/(2d)$ on this event, and $\mathcal{V}$ returns 0. ■

A violation of [DEFINITION 3.7](#) consists of polynomials p and δ such that, for every polynomial Q , infinitely many λ admit an interval bound $\gamma \geq Q(\lambda)$ below the group order and a circuit of size at most $p(\lambda)$ whose advantage exceeds $1/\delta(\lambda)$. With $d := \lceil 2\delta(\lambda) \rceil$, this advantage is at least $2/d$, so by [ITEM \(a\)](#) of [LEMMA 3.8](#), we accept each such instance in time polynomial in λ with probability at least $1 - 2^{-\lambda}$, and by [ITEM \(b\)](#) of [LEMMA 3.8](#), we accept a circuit of advantage at most $1/d$ with probability at most $2^{-\lambda}$.

Since our constructions publish d_λ bounded powers at once, we need tunable DSEDL for vectors, and the coordinate hybrid costs a factor d_λ in the target advantage.

LEMMA 3.9 (Tunable Vector DSEDL). Suppose that $\text{CGen}(\cdot)$ satisfies tunable DSEDL. Let $d: \mathbb{N} \rightarrow \mathbb{N}_{>0}$ be polynomially bounded, and write $d_\lambda := d(\lambda)$. For every $\lambda \in \mathbb{N}$ and every positive integer γ smaller than the order of every group in $\text{supp}(\text{CGen}(1^\lambda))$, define

$$\begin{aligned} \mathfrak{D}_{0,\lambda,\gamma}^{\text{vDSEDL}} &:= \left\{ (1^\lambda, \text{pp}, \gamma, (g^{s_j})_{j \in [d_\lambda]}) \mid \begin{array}{l} \text{pp} \xleftarrow{s} \text{CGen}(1^\lambda), \\ (s_j)_{j \in [d_\lambda]} \xleftarrow{s} [\gamma]^{d_\lambda} \end{array} \right\}, \\ \mathfrak{D}_{1,\lambda,\gamma}^{\text{vDSEDL}} &:= \left\{ (1^\lambda, \text{pp}, \gamma, (G_j)_{j \in [d_\lambda]}) \mid \begin{array}{l} \text{pp} \xleftarrow{s} \text{CGen}(1^\lambda), \\ (G_j)_{j \in [d_\lambda]} \xleftarrow{s} \mathbb{G}(\text{pp})^{d_\lambda} \end{array} \right\}. \end{aligned}$$

Then, for every polynomial p and every positive polynomial δ , there exist a polynomial Q and $\lambda_0 \in \mathbb{N}$ such that $\mathfrak{D}_{0,\lambda,\gamma}^{\text{vDSEDL}} \equiv_{p,1/\delta} \mathfrak{D}_{1,\lambda,\gamma}^{\text{vDSEDL}}$ for every $\lambda \geq \lambda_0$ and every $\gamma \in \mathbb{N}$ satisfying $Q(\lambda) \leq \gamma < \text{ord}(\text{pp})$ for all $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$. ■

Proof. We apply the scalar assumption with the target advantage multiplied by a polynomial upper bound on the vector dimension, and we obtain the vector bound from the telescoping identity over the d_λ transitions of a coordinate hybrid.

We fix p and δ , and we let p^* be a polynomial that bounds the size of a circuit that samples $d_\lambda - 1$ exponents or uniform group elements, performs the corresponding exponentiations, runs a circuit of size at most $p(\lambda)$, and performs one equality test against the failure symbol and the corresponding branch. Since d is polynomially bounded, there is a positive polynomial d^* with $d^*(\lambda) \geq d(\lambda)$ for every λ , and we define the positive polynomial $\delta^*(\lambda) := d^*(\lambda)\delta(\lambda)$. We apply [DEFINITION 3.7](#) with p^* and δ^* and obtain Q and λ_0 .

HYBRIDS. We fix $\lambda \geq \lambda_0$ and $\gamma \in \mathbb{N}$ with $Q(\lambda) \leq \gamma < \text{ord}(\text{pp})$ for every $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$, and we let $\mathcal{A}$ be a circuit of size at most $p(\lambda)$. For every $k \in [0..d_\lambda]$, we let Hyb_k be the distribution with uniform group elements in its first k coordinates and bounded powers in its remaining coordinates, as in [FIGURE 3.1](#), and we define

$$p_k := \Pr[\mathcal{A}(X) = 1 \mid X \xleftarrow{s} \text{Hyb}_k].$$

Under the propagation convention of [§2.4](#), if a parameter or group-element sample fails, then

<u>Hyb_{k-1}</u>	<i>* coordinate k bounded *</i>	<u>Hyb_k</u>	<i>* coordinate k uniform *</i>
00 $\text{pp} \xleftarrow{\$} \text{CGen}(1^\lambda)$		08 $\text{pp} \xleftarrow{\$} \text{CGen}(1^\lambda)$	
01 if $\text{pp} = \perp$ return $\perp$		09 if $\text{pp} = \perp$ return $\perp$	
02 $g \leftarrow g(\text{pp})$		10 $g \leftarrow g(\text{pp})$	
03 $(G_j)_{j < k} \xleftarrow{\$} \mathbb{G}(\text{pp})^{k-1}$		11 $(G_j)_{j < k} \xleftarrow{\$} \mathbb{G}(\text{pp})^{k-1}$	
04 $s_k \xleftarrow{\$} [\gamma]; G_k \leftarrow g^{s_k}$		12 $G_k \xleftarrow{\$} \mathbb{G}(\text{pp})$	
05 $(s_j)_{j > k} \xleftarrow{\$} [\gamma]^{d_\lambda - k}$		13 $(s_j)_{j > k} \xleftarrow{\$} [\gamma]^{d_\lambda - k}$	
06 $(G_j)_{j > k} \leftarrow (g^{s_j})_{j > k}; \mathbf{G} \leftarrow (G_j)_{j \in [d_\lambda]}$		14 $(G_j)_{j > k} \leftarrow (g^{s_j})_{j > k}; \mathbf{G} \leftarrow (G_j)_{j \in [d_\lambda]}$	
07 return $(1^\lambda, \text{pp}, \gamma, \mathbf{G})$		15 return $(1^\lambda, \text{pp}, \gamma, \mathbf{G})$	

FIGURE 3.1. One coordinate transition in the tunable vector-DSEDL hybrid.

the hybrid outputs $\perp$, and this outcome contributes zero to p_k . In particular, p_0 and p_{d_λ} are the acceptance probabilities of $\mathcal{A}$ on $\mathfrak{D}_{0,\lambda,\gamma}^{\text{vDSEDL}}$ and $\mathfrak{D}_{1,\lambda,\gamma}^{\text{vDSEDL}}$, respectively.

THE REDUCTION. We construct a scalar circuit $\mathcal{B}$ of size at most $p^*(\lambda)$. On input $\perp$, the circuit $\mathcal{B}$ returns 0. On a challenge other than $\perp$, it samples $L \xleftarrow{\$} [d_\lambda]$, places its scalar challenge in coordinate L , samples the coordinates $j < L$ as uniform group elements of $\mathbb{G}(\text{pp})$ and the coordinates $j > L$ as bounded powers, runs $\mathcal{A}$ on the resulting vector, and returns the output of $\mathcal{A}$, and it returns 0 if a further group-element sample fails. We test for $\perp$ and branch on failures because the circuits in [DEFINITION 3.7](#) output a bit and because $\mathcal{B}$ cannot sample the remaining coordinates without pp . For $b \in \{0, 1\}$, we write

$$q_b := \Pr \left[\mathcal{B}(X) = 1 \mid X \xleftarrow{\$} \mathfrak{D}_{b,\lambda,\gamma}^{\text{vDSEDL}} \right].$$

For each value $L = k$, the circuit $\mathcal{B}$ and the hybrids Hyb_{k-1} and Hyb_k make the same parameter and group-element sampling calls and assign acceptance probability zero if a call fails, so on a bounded scalar challenge the input of $\mathcal{A}$ is distributed as Hyb_{k-1} , and on a uniform one as Hyb_k . Hence,

$$q_0 - q_1 = \frac{1}{d_\lambda} \sum_{k=1}^{d_\lambda} (p_{k-1} - p_k) = \frac{p_0 - p_{d_\lambda}}{d_\lambda}.$$

By the choice of Q and λ_0 in [DEFINITION 3.7](#), we have $|q_0 - q_1| \leq 1/\delta^*(\lambda)$. We multiply the identity for $q_0 - q_1$ by d_λ and obtain

$$|p_0 - p_{d_\lambda}| = d_\lambda |q_0 - q_1| \leq \frac{d(\lambda)}{\delta^*(\lambda)} = \frac{d(\lambda)}{d^*(\lambda)\delta(\lambda)} \leq \frac{1}{\delta(\lambda)},$$

as required. ■

3.4 Generic-Group Bounds for Bounded Powers

Tunable DSEDL is the only nonstandard decisional assumption in our constructions, and we bound its dependence on the resources of the adversary in the generic group model [[Nec94](#), [Sho97](#)]. We must allow a composite order, since the group of [DEFINITION 2.8](#) has composite order and its exponent ring is not a field. Therefore, we carry out the analysis for an arbitrary order q and write π_q for its smallest prime factor, so that $\pi_q = q$ for prime q . A nonzero coefficient in $\mathbb{Z}_q$ need not be invertible, so a nonzero affine form can vanish at many points, but after we fix all variables

except one with a nonzero coefficient, the solutions of the resulting univariate congruence form an arithmetic progression with common difference at least π_q .

In the generic group model, the group elements are random labels, and the adversary computes with them only through an oracle for the group operation, so its view is determined by which of its affine combinations of the challenge exponents coincide.

DEFINITION 3.10 (Generic Bounded-Power Problem). For finite sets X and Y with $|X| = |Y|$, we write $\psi(X, Y)$ for the set of bijections from X to Y . Let $q, \gamma, d \in \mathbb{N}$ with $q \geq 2$ and $1 \leq \gamma < q$, and let $\mathcal{L}$ be a set with $|\mathcal{L}| = q$. For $\xi \in \psi(\mathbb{Z}_q, \mathcal{L})$, the generic group oracle $\mathcal{O}_\xi$ is defined by $\mathcal{O}_\xi(\xi(a), \xi(b), \varepsilon) := \xi(a + (-1)^\varepsilon b)$ for $a, b \in \mathbb{Z}_q$ and $\varepsilon \in \{0, 1\}$. A generic algorithm is an oracle algorithm $\mathcal{A}$ that queries $\mathcal{O}_\xi$ only on labels that it received as input or as answers of $\mathcal{O}_\xi$, and whose nonuniform advice depends only on q, γ , and d . For $b \in \{0, 1\}$, let $\mathcal{X}_0 := [\gamma]^d$ and $\mathcal{X}_1 := \mathbb{Z}_q^d$, and define

$$p_b(\mathcal{A}) := \Pr \left[\mathcal{A}^{\mathcal{O}_\xi} \left(q, \gamma, \xi(1), (\xi(x_j))_{j \in [d]} \right) = 1 \mid \begin{array}{l} (x_j)_{j \in [d]} \xleftarrow{\$} \mathcal{X}_b, \\ \xi \xleftarrow{\$} \psi(\mathbb{Z}_q, \mathcal{L}) \end{array} \right].$$

The generic advantage of $\mathcal{A}$ is $\text{Adv}_{q, \gamma, d}^{\text{ggm-DSEDL}}(\mathcal{A}) := |p_0(\mathcal{A}) - p_1(\mathcal{A})|$. $\lrcorner$

For bounded exponents and composite order, we adapt the symbolic-transcript argument of [Nec94, Sho97]. We have not found this decisional bounded-interval statement in the literature.

THEOREM 3.11 (Generic-Group Bound). Let $q, \gamma, d, T \in \mathbb{N}$ with $q \geq 2$, $1 \leq \gamma < q$, and $d \geq 1$, let $\pi_q := \min\{p \in \text{Primes} \mid q \in p\mathbb{Z}\}$, and let $\mathcal{A}$ be a generic algorithm that makes at most T oracle queries. Then

$$\text{Adv}_{q, \gamma, d}^{\text{ggm-DSEDL}}(\mathcal{A}) \leq \min \left\{ 1, \binom{T + d + 1}{2} \left(\frac{1}{\gamma} + \frac{2}{\pi_q} \right) \right\}.$$

Proof. We compare both experiments with one symbolic execution in which each label stands for an affine form in the challenge exponents. The transcript of each experiment has the same distribution as the symbolic transcript unless two distinct forms of the execution take the same value, and we bound the probability of this collision event separately for bounded and for uniform exponents.

We write $\mathbf{n} := T + d + 1$ within this proof. If

$$\binom{\mathbf{n}}{2} \left(\frac{1}{\gamma} + \frac{2}{\pi_q} \right) \geq 1,$$

then the bound holds because $\text{Adv}_{q, \gamma, d}^{\text{ggm-DSEDL}}(\mathcal{A}) \leq 1$. We suppose henceforth that the left-hand side is smaller than one, so that $\binom{\mathbf{n}}{2} < \pi_q/2 \leq q/2$. Then $\mathbf{n} \leq q$, since $\binom{v}{2} \geq v$ for every integer $v \geq 3$, and $\mathbf{n} \leq 2 \leq q$ otherwise. Hence, the at most $\mathbf{n}$ forms of the symbolic execution can receive distinct labels from $\mathcal{L}$, and the symbolic oracle always has an unused label.

We fix the coins and the nonuniform advice of $\mathcal{A}$. Each label of the symbolic execution stands for an affine form

$$F(\mathbf{X}) = \alpha_0 + \sum_{j=1}^d \alpha_j X_j \in \mathbb{Z}_q[X_1, \dots, X_d],$$

and the initial labels stand for $1, X_1, \dots, X_d$. The symbolic oracle keeps an injective table from coefficient vectors in $\mathbb{Z}_q^{d+1}$ to labels and assigns uniformly random distinct labels to these $d + 1$ forms. On a query with the labels of forms F_1 and F_2 and with $\varepsilon \in \{0, 1\}$, the oracle computes

$F_1 + (-1)^\varepsilon F_2$, returns the existing label if its coefficient vector is in the table, and otherwise assigns a uniformly random unused label. The execution begins with $d + 1$ forms, and each query adds at most one form, so at most n forms occur. The symbolic transcript depends on the coins of $\mathcal{A}$ and on the assigned labels and is independent of the values of $X_1, \dots, X_d$.

We fix a complete symbolic transcript and, for a distribution $\mathfrak{D}$ on $\mathbb{Z}_q^d$, write $E_{\mathfrak{D}}$ for the event that two distinct forms of the transcript take the same value at $\mathbf{X} \leftarrow^s \mathfrak{D}$. We fix distinct forms F_1 and F_2 of the transcript and write

$$R(\mathbf{X}) := F_1(\mathbf{X}) - F_2(\mathbf{X}) = \beta_0 + \sum_{j=1}^d \beta_j X_j,$$

which is nonzero because $F_1 \neq F_2$. If $\beta_1 = \dots = \beta_d = 0$, then $\beta_0 \neq 0$, and R does not vanish. Otherwise, we fix j with $\beta_j \neq 0$. After we fix every variable other than X_j , the condition $R(\mathbf{X}) = 0$ becomes the congruence $\beta_j X_j \equiv c \pmod{q}$ for some $c \in \mathbb{Z}_q$, whose solution set is either empty or an arithmetic progression of $\gcd(\beta_j, q)$ residues with common difference $q/\gcd(\beta_j, q)$. Since $\beta_j \neq 0$, the integer $q/\gcd(\beta_j, q)$ is a divisor of q larger than one, hence at least π_q . Since $\gamma < q$, the elements of $[\gamma]$ are distinct modulo q , and an interval of γ consecutive integers meets an arithmetic progression with common difference at least π_q in at most $\lceil \gamma/\pi_q \rceil \leq \gamma/\pi_q + 1$ points. Since X_j is uniform on $[\gamma]$ and independent of the other coordinates, we obtain

$$\Pr[R(\mathbf{X}) = 0 \mid \mathbf{X} \leftarrow^s [\gamma]^d] \leq \frac{1}{\gamma} \left(\frac{\gamma}{\pi_q} + 1 \right) = \frac{1}{\pi_q} + \frac{1}{\gamma}.$$

For uniform exponents, after the same conditioning, $\gcd(\beta_j, q)$ of the q residues of X_j satisfy the congruence, so

$$\Pr[R(\mathbf{X}) = 0 \mid \mathbf{X} \leftarrow^s \mathbb{Z}_q^d] \leq \frac{\gcd(\beta_j, q)}{q} = \frac{1}{q/\gcd(\beta_j, q)} \leq \frac{1}{\pi_q}.$$

By a union bound over the at most $\binom{n}{2}$ unordered pairs of forms,

$$\Pr[E_{[\gamma]^d}] \leq \binom{n}{2} \left(\frac{1}{\gamma} + \frac{1}{\pi_q} \right), \quad \Pr[E_{\mathbb{Z}_q^d}] \leq \binom{n}{2} \frac{1}{\pi_q}.$$

Outside the corresponding collision event, two evaluated forms take the same value exactly when their coefficient vectors agree, so the equality pattern among the group elements of the experiment agrees with the symbolic table. The symbolic oracle assigns uniformly random distinct labels to the forms that occur, which is the distribution of the labels of their values under a uniformly random bijection $\xi: \mathbb{Z}_q \rightarrow \mathcal{L}$. Hence, outside the corresponding collision event, the transcript of each of the two experiments has the same distribution as the symbolic transcript, and we obtain

$$\text{Adv}_{q, \gamma, d}^{\text{ggm-DSEDL}}(\mathcal{A}) \leq \Pr[E_{[\gamma]^d}] + \Pr[E_{\mathbb{Z}_q^d}] \leq \binom{n}{2} \left(\frac{1}{\gamma} + \frac{2}{\pi_q} \right).$$

This bound holds for every fixed choice of coins, hence also after we average over the coins of a randomized $\mathcal{A}$, and with the case treated at the start of the proof, the bound in the statement holds. ■

THEOREM 3.11 holds for every $\gamma \in [q - 1]$, including $\gamma \geq \pi_q$, for which a nonzero affine form can vanish at several points of an interval of length γ . For the sampler of **DEFINITION 2.8**, the group order is $q = p'q'$ with $p', q' \geq 2^{\ell(\lambda)-2}$, and since $\ell(\lambda) \geq \lambda + 2$, we have $\pi_q \geq 2^{\ell(\lambda)-2} \geq 2^\lambda$. Hence, every interval with $\gamma < 2^\lambda$ satisfies $\gamma < \pi_q$, and the term $2/\pi_q$ is at most $2^{1-\lambda}$.

The factor d_λ in the target advantage of **LEMMA 3.9** comes from the d_λ transitions of the coordinate hybrid. For $T \geq d$, we have

$$\binom{T+d+1}{2} \leq \binom{2T+1}{2} = T(2T+1) \leq 6 \binom{T+2}{2},$$

so the bound of **THEOREM 3.11** for d challenge powers is at most six times the bound for one challenge power with the same T . Hence, a vector version of tunable DSEDL, assumed directly, would remove the factor d_λ from every interval threshold that we derive from **LEMMA 3.9**, and we take the scalar assumption as our starting point and keep this factor.

With square-and-multiply, we can compute $\xi(\sigma x)$ from $\xi(x)$ for $\sigma \in [0..q-1]$ with at most $2 \lceil \log_2 q \rceil$ oracle queries. Hence, a generic algorithm that makes T queries and, in addition, computes the σ th powers of D challenge elements makes at most $T + 2D \lceil \log_2 q \rceil$ queries, and **THEOREM 3.11** applies with T replaced by $T + 2D \lceil \log_2 q \rceil$.

To show that the dependence on γ in **THEOREM 3.11** is tight at constant advantage, we restrict baby-step-giant-step [**Sha71**] to the interval $[\gamma]$, as in [**vOW96**].

THEOREM 3.12 (Matching Generic Attack). For all $q, \gamma \in \mathbb{N}$ with $q \geq 2$ and $1 \leq \gamma < q$, there is a deterministic generic algorithm $\mathcal{A}$ that makes at most $2 \lceil \sqrt{\gamma} \rceil + 1$ oracle queries and satisfies $\text{Adv}_{q,\gamma,1}^{\text{ggm-DSEDL}}(\mathcal{A}) = 1 - \gamma/q$. $\square$

Proof. We run baby-step-giant-step on the interval $[\gamma]$ and add a range check. Because of the range check, acceptance is equivalent to membership of the challenge exponent in $[\gamma]$.

We set $v := \lceil \sqrt{\gamma} \rceil$. On input $\xi(1)$ and the challenge label $\xi(x)$, the algorithm $\mathcal{A}$ computes and stores the labels $\xi(j)$ for $j \in [0..v]$ with at most v oracle queries, then computes $\xi(x - iv)$ for every $i \in [v]$ with v further queries, and accepts if and only if there are $i \in [0..v]$ and $j \in [0..v-1]$ with $\xi(x - iv) = \xi(j)$ and $iv + j \in [\gamma]$.

If $x \in [\gamma]$, then $x \leq \gamma \leq v^2$, and by Euclidean division, $x = iv + j$ with $i := \lfloor x/v \rfloor \in [0..v]$ and $j \in [0..v-1]$, so $\mathcal{A}$ accepts every bounded challenge. Conversely, if $\mathcal{A}$ accepts, then $x \equiv iv + j \pmod{q}$ for an integer $iv + j \in [\gamma]$. Since $\gamma < q$, both x and $iv + j$ are their own representatives in $[0..q-1]$, so $x = iv + j \in [\gamma]$. Hence, $\mathcal{A}$ accepts a uniform $x \xleftarrow{\$} \mathbb{Z}_q$ with probability γ/q , so $\text{Adv}_{q,\gamma,1}^{\text{ggm-DSEDL}}(\mathcal{A}) = 1 - \gamma/q$, and $\mathcal{A}$ makes at most $2v + 1$ queries. $\blacksquare$

3.5 Tightness and Preprocessing

By **THEOREM 3.12**, the quadratic dependence on the number of queries in **THEOREM 3.11** is tight at constant advantage, and we show that it is tight at every advantage and that the upper bound extends to encoding-dependent preprocessing. When we truncate the grid of **THEOREM 3.12**, we accept exactly the challenge exponents in an initial segment of $[\gamma]$ whose size is quadratic in the grid parameter.

PROPOSITION 3.13 (Truncated Generic Attack). Let $q, \gamma, v \in \mathbb{N}$ with $q \geq 2$, $1 \leq \gamma < q$, and $1 \leq v \leq \lfloor \sqrt{\gamma} \rfloor$. There is a deterministic generic algorithm $\mathcal{A}$ that makes at most $2v - 1$ oracle queries

and satisfies

$$\text{Adv}_{q,\gamma,1}^{\text{ggm-DSEDL}}(\mathcal{A}) = (v^2 - 1) \left(\frac{1}{\gamma} - \frac{1}{q} \right).$$

In particular, if $\gamma \leq q/2$, then for every odd $T \in \mathbb{N}$ with $3 \leq T \leq 2 \lfloor \sqrt{\gamma} \rfloor - 1$, the algorithm for $v = (T + 1)/2$ makes at most T queries and satisfies $\text{Adv}_{q,\gamma,1}^{\text{ggm-DSEDL}}(\mathcal{A}) \geq T^2/(8\gamma)$. $\lrcorner$

Proof. The two acceptance probabilities are the densities of the segment $[v^2 - 1]$ in $[\gamma]$ and in $\mathbb{Z}_q$.

On input $\xi(1)$ and the challenge label $\xi(x)$, the algorithm $\mathcal{A}$ computes and stores the labels $\xi(j)$ for $j \in [0..v]$ with at most v oracle queries, then computes $\xi(x - iv)$ for every $i \in [v - 1]$ with at most $v - 1$ further queries, and accepts if and only if there are $i, j \in [0..v - 1]$ with $\xi(x - iv) = \xi(j)$ and $iv + j \in [\gamma]$.

Since $v \leq \lfloor \sqrt{\gamma} \rfloor$, we have $v^2 - 1 < \gamma < q$, so every element of $[v^2 - 1]$ is its own representative modulo q and lies in $[\gamma]$. Since $\{iv + j \mid i, j \in [0..v - 1]\} = [0..v^2 - 1]$, the algorithm $\mathcal{A}$ accepts if and only if $x \bmod q \in [v^2 - 1]$. Hence, $\mathcal{A}$ accepts a bounded challenge $x \xleftarrow{\$} [\gamma]$ with probability $(v^2 - 1)/\gamma$ and a uniform challenge $x \xleftarrow{\$} \mathbb{Z}_q$ with probability $(v^2 - 1)/q$, and $\text{Adv}_{q,\gamma,1}^{\text{ggm-DSEDL}}(\mathcal{A})$ is their difference.

For the last claim, $v = (T + 1)/2$ satisfies $1 \leq v \leq \lfloor \sqrt{\gamma} \rfloor$, and since $T \geq 3$, we have $v^2 - 1 = T^2/4 + (2T - 3)/4 \geq T^2/4$. Since $\gamma \leq q/2$, we have $1/\gamma - 1/q \geq 1/(2\gamma)$, so $\text{Adv}_{q,\gamma,1}^{\text{ggm-DSEDL}}(\mathcal{A}) \geq T^2/(8\gamma)$. $\blacksquare$

Preprocessing. In [THEOREM 3.11](#), the nonuniform advice is independent of the encoding ξ , whereas a concrete nonuniform adversary can preprocess one fixed group representation. We extend [THEOREM 3.11](#) to the auxiliary-input generic-group model of [\[CDG18\]](#), in which an unbounded preprocessing algorithm receives the complete encoding ξ and passes an S -bit advice string to an online algorithm that makes at most T group-oracle queries.

We first bound the advantage in the bit-fixing model of [\[CDG18\]](#), in which at most P points of the encoding are fixed in advance. For a list $\mathcal{F} = ((r_i, \ell_i))_{i \in [P']}$ with $P' \leq P$, pairwise distinct $r_i \in \mathbb{Z}_q$, and pairwise distinct labels $\ell_i \in \mathcal{L}$, we write $\psi_{\mathcal{F}}$ for the set of bijections $\xi \in \psi(\mathbb{Z}_q, \mathcal{L})$ with $\xi(r_i) = \ell_i$ for every $i \in [P']$. The online algorithm receives the complete list $\mathcal{F}$ and arbitrary advice derived from it, and since it can ignore $\mathcal{F}$, this model contains the bit-fixing model of [\[CDG18\]](#). As in the bit-fixing bounds of [\[CDG18\]](#), we adapt the symbolic argument of [THEOREM 3.11](#) to this model.

THEOREM 3.14 (Bit-Fixing Generic Bound). Let $q, \gamma, d, P, T \in \mathbb{N}$ with $q \geq 2$ and $1 \leq \gamma < q$, let π_q be the smallest prime factor of q , let $\mathcal{F} = ((r_i, \ell_i))_{i \in [P']}$ with $P' \leq P$, pairwise distinct $r_i \in \mathbb{Z}_q$, and pairwise distinct $\ell_i \in \mathcal{L}$, and let z be a string. Let $\mathcal{A}$ be a generic algorithm that makes at most T queries to $\mathcal{O}_{\xi}$, and for $b \in \{0, 1\}$, let $\mathcal{X}_0 := [\gamma]^d$, $\mathcal{X}_1 := \mathbb{Z}_q^d$, and

$$p_b := \Pr \left[\mathcal{A}^{\mathcal{O}_{\xi}} \left(z, \mathcal{F}, q, \gamma, \xi(1), (\xi(x_j))_{j \in [d]} \right) = 1 \mid \begin{array}{l} (x_j)_{j \in [d]} \xleftarrow{\$} \mathcal{X}_b, \\ \xi \xleftarrow{\$} \psi_{\mathcal{F}} \end{array} \right].$$

Then

$$|p_0 - p_1| \leq \min \left\{ 1, \binom{T + d + P + 1}{2} \left(\frac{1}{\gamma} + \frac{2}{\pi_q} \right) \right\}.$$

Proof. We repeat the symbolic execution of the proof of [THEOREM 3.11](#), which now begins with the constant forms of the exponents r_i of $\mathcal{F}$ in addition to $1, X_1, \dots, X_d$. The labels of the constant

forms are prescribed by $\mathcal{F}$, every new coefficient vector receives a uniformly random unused label, and the collision calculation does not depend on z and $\mathcal{F}$, which are fixed.

We fix the coins of $\mathcal{A}$ and write $\mathbf{n} := T + d + P + 1$ within this proof. At most $P + d + 1$ forms occur initially, and every query adds at most one form, so the transcript contains at most $\mathbf{n}$ forms. For two distinct forms F_1 and F_2 , the difference $F_1 - F_2$ is a nonzero affine polynomial in $X_1, \dots, X_d$. If one of its variable coefficients is nonzero, we repeat the progression argument of the proof of [THEOREM 3.11](#), conditioned on the other variables, and obtain that $F_1 - F_2$ vanishes with probability at most $1/\gamma + 1/\pi_q$ for bounded exponents and at most $1/\pi_q$ for uniform exponents. If all its variable coefficients are zero, then $F_1 - F_2$ is a nonzero constant and does not vanish.

As in the proof of [THEOREM 3.11](#), if

$$\binom{\mathbf{n}}{2} \left(\frac{1}{\gamma} + \frac{2}{\pi_q} \right) \geq 1,$$

then the bound holds because the advantage is at most one. Otherwise, $\binom{\mathbf{n}}{2} < \pi_q/2 \leq q/2$, so $\mathbf{n} \leq q$ by the argument of that proof, and the symbolic execution always has an unused label. By a union bound over the unordered pairs of forms, the collision event has probability at most

$$\binom{\mathbf{n}}{2} \left(\frac{1}{\gamma} + \frac{1}{\pi_q} \right), \quad \binom{\mathbf{n}}{2} \frac{1}{\pi_q}$$

for bounded and for uniform exponents, respectively. Outside the collision event, the equality pattern among the evaluated exponents agrees with the symbolic table, and the labels of the exponents outside $\mathcal{F}$ are uniformly distributed without replacement among the unused labels, which is their distribution under a uniformly random bijection $\xi \xleftarrow{\$} \psi_{\mathcal{F}}$. We compare each of the two experiments with the symbolic execution, average over the coins of $\mathcal{A}$, and obtain the bound. $\blacksquare$

We state the auxiliary-input version of the problem of [DEFINITION 3.10](#), adapted from the auxiliary-input generic group of [\[CDG18\]](#).

DEFINITION 3.15 (Auxiliary-Input Generic Bounded-Power Problem). Let $q, \gamma, d, S, T \in \mathbb{N}$ with $q \geq 2$ and $1 \leq \gamma < q$, and let $\mathcal{X}_0 := [\gamma]^d$ and $\mathcal{X}_1 := \mathbb{Z}_q^d$. A two-stage auxiliary-input generic algorithm is a pair $\mathcal{A} = (\mathcal{A}_{\text{pre}}, \mathcal{A}_{\text{on}})$ of a computationally unbounded algorithm $\mathcal{A}_{\text{pre}}$ that, on input $\xi \in \psi(\mathbb{Z}_q, \mathcal{L})$, outputs a string $z \in \{0, 1\}^{\leq S}$, and a generic algorithm $\mathcal{A}_{\text{on}}$ that makes at most T queries to $\mathcal{O}_{\xi}$. For $b \in \{0, 1\}$, define

$$p_b^{\text{ai-ggm-DSEDL}}(\mathcal{A}) := \Pr \left[\mathcal{A}_{\text{on}}^{\mathcal{O}_{\xi}} \left(z, q, \gamma, \xi(1), (\xi(x_j))_{j \in [d]} \right) = 1 \mid \begin{array}{l} \xi \xleftarrow{\$} \psi(\mathbb{Z}_q, \mathcal{L}), \\ z \xleftarrow{\$} \mathcal{A}_{\text{pre}}(\xi), \\ (x_j)_{j \in [d]} \xleftarrow{\$} \mathcal{X}_b \end{array} \right],$$

and $\text{Adv}_{q, \gamma, d, S, T}^{\text{ai-ggm-DSEDL}}(\mathcal{A}) := \left| p_0^{\text{ai-ggm-DSEDL}}(\mathcal{A}) - p_1^{\text{ai-ggm-DSEDL}}(\mathcal{A}) \right|$. $\dashv$

We combine [THEOREM 3.14](#) with the additive part of the auxiliary-input-to-bit-fixing theorem of [\[CDG18\]](#) and obtain the following bound.

THEOREM 3.16 (Auxiliary-Input Generic Bound). Let $q, \gamma, d, S, T \in \mathbb{N}$ with $q \geq 16$ and $1 \leq \gamma < q$, let π_q be the smallest prime factor of q , let $\vartheta \in (0, 1)$, and let $P \in [q]$. Every two-stage auxiliary-input

generic algorithm $\mathcal{A}$ in the sense of [DEFINITION 3.15](#) satisfies

$$\begin{aligned} \text{Adv}_{q,\gamma,d,S,T}^{\text{ai-ggm-DSEDL}}(\mathcal{A}) \leq & \min \left\{ 1, \binom{T+d+P+1}{2} \left(\frac{1}{\gamma} + \frac{2}{\pi_q} \right) \right\} \\ & + \frac{12(S + \log_2(1/\vartheta))(T+d+1)}{P} + 2\vartheta. \end{aligned}$$

┘

Proof. We apply the additive part of the auxiliary-input-to-bit-fixing theorem of [\[CDG18\]](#) to the problem of [DEFINITION 3.15](#). We take the auxiliary-input and bit-fixing generic groups of [\[CDG18\]](#) to be $\text{AI-GG}(q, q)$ and $\text{BF-GG}(P, q, q)$, so the random injection of [\[CDG18\]](#) is the bijection ξ , and the range set published by the bit-fixing oracle of [\[CDG18\]](#) is $\mathcal{L}$, which carries no information about ξ . The theorem of [\[CDG18\]](#) concerns random injections, so it holds for every q , including a composite order. The slack parameter of [\[CDG18\]](#), which [\[CDG18\]](#) calls γ , is our ϑ , and the advantage of [\[CDG18\]](#) for an indistinguishability application is twice the deviation of the success probability from one half, which equals $\text{Adv}_{q,\gamma,d,S,T}^{\text{ai-ggm-DSEDL}}(\mathcal{A})$. The combined query complexity of the online algorithm and the challenger is at most $T + d + 1$, since the challenger reveals $\xi(1)$ and the d challenge labels in addition to the T queries of $\mathcal{A}_{\text{on}}$. By the theorem of [\[CDG18\]](#), we can replace the auxiliary-input experiment by a convex combination of bit-fixing experiments with at most P fixed points, at additive loss

$$\frac{12(S + \log_2(1/\vartheta))(T + d + 1)}{P} + 2\vartheta.$$

Since the bound of [THEOREM 3.14](#) holds for every list of at most P pairs and every auxiliary string, we apply it to every member of the convex combination and substitute the result into the presampling bound. ■

We choose the free parameters of [THEOREM 3.16](#) and obtain a polynomial interval threshold against encoding-dependent preprocessing.

COROLLARY 3.17 (Generic Bound with Preprocessing). Let $\delta, S, T, d: \mathbb{N} \rightarrow \mathbb{N}$ be polynomially bounded with $\delta(\lambda) \geq 1$ for every λ , and for every $\lambda \in \mathbb{N}$, define

$$\begin{aligned} \vartheta(\lambda) &:= \frac{1}{8\delta(\lambda)}, \\ P(\lambda) &:= \lceil 48\delta(\lambda)(S(\lambda) + \lceil \log_2(8\delta(\lambda)) \rceil)(T(\lambda) + d(\lambda) + 1) \rceil, \\ T_{\text{ai}}(\lambda) &:= T(\lambda) + d(\lambda) + P(\lambda) + 1. \end{aligned}$$

Let $\lambda, q, \gamma \in \mathbb{N}$ satisfy

$$\pi_q \geq 8\delta(\lambda) \binom{T_{\text{ai}}(\lambda)}{2}, \quad 4\delta(\lambda) \binom{T_{\text{ai}}(\lambda)}{2} \leq \gamma < q,$$

where π_q is the smallest prime factor of q . Then every two-stage auxiliary-input generic algorithm $\mathcal{A}$ whose preprocessing stage outputs at most $S(\lambda)$ bits and whose online stage makes at most $T(\lambda)$ queries to $\mathcal{O}_\xi$ satisfies $\text{Adv}_{q,\gamma,d(\lambda),S(\lambda),T(\lambda)}^{\text{ai-ggm-DSEDL}}(\mathcal{A}) \leq 1/\delta(\lambda)$. Moreover, the function $\lambda \mapsto 4\delta(\lambda) \binom{T_{\text{ai}}(\lambda)}{2}$ is polynomially bounded. ┘

Proof. We bound the three terms of the presampling bound separately by $1/(2\delta)$, $1/(4\delta)$, and $1/(4\delta)$, where we write δ , S , T , d , ϑ , P , and T_{ai} for their values at λ .

Since $T_{\text{ai}} \geq P + 1 \geq 49$, we have $\binom{T_{\text{ai}}}{2} \geq T_{\text{ai}} > P$, so $q \geq \pi_q \geq 8\delta \binom{T_{\text{ai}}}{2} \geq P$ and $q \geq 16$. Hence, [THEOREM 3.16](#) applies with these ϑ and P , and its bound is at most

$$\binom{T_{\text{ai}}}{2} \left(\frac{1}{\gamma} + \frac{2}{\pi_q} \right) + \frac{12(S + \log_2(1/\vartheta))(T + d + 1)}{P} + 2\vartheta.$$

By the conditions on γ and π_q , the first term is at most $1/(4\delta) + 1/(4\delta) = 1/(2\delta)$. Since $\log_2(1/\vartheta) = \log_2(8\delta) \leq \lceil \log_2(8\delta) \rceil$, the choice of P bounds the second term by $12/(48\delta) = 1/(4\delta)$, and $2\vartheta = 1/(4\delta)$. Hence, the advantage is at most $1/\delta$. Since S , T , d , and δ are polynomially bounded, so are P , T_{ai} , and $4\delta \binom{T_{\text{ai}}}{2}$. ■

Parameter Scale. By [THEOREM 3.11](#), every generic algorithm that makes at most T queries satisfies $\text{Adv}_{q,\gamma,d}^{\text{ggm-DSEDL}}(\mathcal{A}) \leq 1/\delta$ whenever

$$\gamma \geq 2\delta \binom{T + d + 1}{2}, \quad \pi_q \geq 4\delta \binom{T + d + 1}{2},$$

since each of the two terms of the bound is then at most $1/(2\delta)$. By [COROLLARY 3.17](#), a polynomial interval threshold also suffices against polynomial encoding-dependent advice. By [THEOREM 3.12](#) and [PROPOSITION 3.13](#), the quadratic dependence on T is tight for $d = 1$, at advantage $1 - \gamma/q$ and at every inverse-polynomial advantage, respectively.

Scope of the Generic Bounds. [THEOREMS 3.11](#) and [3.16](#) hold in the generic group model, so concrete parameters for tunable DSEDL remain open, and DCR, which [DEFINITION 2.13](#) states, lies outside this analysis. In [THEOREM 3.16](#), the advice depends on the complete random encoding, and the interval threshold grows with the advice length S , which is necessary, since advice that contains the labels of a set $\mathcal{I} \subseteq [\gamma]$ yields a distinguisher with zero queries and advantage $|\mathcal{I}|/\gamma - |\mathcal{I}|/q$. In both theorems, the algorithm receives the group order q , so the bounds hold without the secrecy of the order in [DEFINITION 2.8](#). In the structured generic group model of [\[CHW26\]](#), labels have a controlled amount of structure, but the bounds of [\[CHW26\]](#) concern search discrete logarithms and do not imply the bounded-versus-uniform statement of [THEOREM 3.11](#), so generic bounds in a model that exposes the group representation remain open.

The generic model does not cover the following attacks that use the representation of the Damgård–Jurik group: arithmetic modulo N and modulo N^2 , both of which apply to a Damgård–Jurik element; factoring N , which splits the group by the Chinese remainder theorem; index-calculus and number-field-sieve methods, which use the arithmetic of the ambient ring [\[Web95, CS06\]](#); preprocessing against a fixed family of moduli, which is a number-theoretic object shared by every instance, whereas the advice in [THEOREM 3.16](#) depends on a random encoding; and statistics of the representation that are independent of the exponent. Whether one of these attacks distinguishes the two distributions of [DEFINITION 3.7](#) faster than baby-step–giant-step [\[Sha71\]](#) is open.

4 Extremely Lossy Trapdoor Hashing

Trapdoor hashing [DGI⁺19] and extremely lossy functions [Zha16] have been developed independently, and we construct one algebraic object with the properties of both. From its short digest, we can recover a predicate value with a trapdoor, as from a trapdoor hash, and under an indistinguishable choice of public parameters, its hash maps an exponential domain onto a polynomial image, as an extremely lossy function does. We define this primitive and build it over a Damgård–Jurik group by replacing the uniform hash bases with bounded powers of a generator. From [RS21], we adapt the distributed discrete logarithm DDLog_N , which maps the digest and the encoding to exact additive shares of a linear predicate.

In our formulation, the inputs are residue vectors modulo the modulus N of the hash key. Inside the construction, we decompose each residue into bits and multiply the offset of bit k by 2^k , so that the predicate value is a linear function of the residues and the binary expansion appears only in hashing and encoding.

We extend trapdoor hashing [DGI⁺19], recalled in DEFINITION 2.2, by an image parameter r of the setup algorithm and change its clauses as follows. Since in the Damgård–Jurik construction the inputs are residue vectors modulo the modulus of the hash key, the input set $\mathcal{X}$ and the predicate range $\mathbb{K}$ depend on the hash key. The encoding takes values in an abelian group $\mathbb{K}$, decoding returns one element of $\mathbb{K}$, and for correctness, we require the encoding and the decoded value to differ by the predicate value. At the full parameter $R_{\text{full}}(\lambda, d)$, the hash key has the full-range distribution, and at every smaller parameter r , the hash with the all-zero tape must take at most r values, so that the image bound is independent of the encoding key. For mode indistinguishability, we require the image parameters it covers to form a nonempty interval for all sufficiently large λ , since otherwise the clause would hold vacuously. For input privacy and function privacy, we require only the full parameter, since at nonfull parameters with polynomially bounded exponents, we can recover the predicate offsets from the encoding key.

DEFINITION 4.1 (Extremely Lossy Trapdoor Hashing). Let $\mathcal{X}$ and $\mathbb{K}$ be families of finite sets and finite abelian groups indexed by $\lambda \in \mathbb{N}$, $d \in \mathbb{N}_{>0}$, and a hash key. Let $\mathcal{C}$ be a function class indexed in the same way, with every $f \in \mathcal{C}_{\lambda,d,\text{hk}}$ mapping $\mathcal{X}_{\lambda,d,\text{hk}}$ to $\mathbb{K}_{\lambda,d,\text{hk}}$. When no ambiguity arises, we write $\mathcal{X} = \mathcal{X}_{\lambda,d,\text{hk}}$, $\mathbb{K} = \mathbb{K}_{\lambda,d,\text{hk}}$, and $\mathcal{C} = \mathcal{C}_{\lambda,d,\text{hk}}$. We require efficient membership testing and canonical representations for $\mathcal{X}$ and $\mathbb{K}$, efficient equality and group operations in $\mathbb{K}$, and polynomial-size descriptions and polynomial-time evaluation for every predicate in $\mathcal{C}$. Let $R_{\text{full}}: \mathbb{N} \times \mathbb{N}_{>0} \rightarrow \mathbb{N}$ be efficiently computable with $\log R_{\text{full}}(\lambda, d) = \text{poly}(\lambda + d)$.

An extremely lossy trapdoor-hashing scheme for $\mathcal{C}$ is a tuple of probabilistic polynomial-time algorithms $\text{ELTDH} := (\text{ELTDH.Setup}, \text{ELTDH.Hash}, \text{ELTDH.Gen}, \text{ELTDH.Enc}, \text{ELTDH.Dec})$ with the following syntax.

- $\text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^d, r)$. On input the security parameter, the arity d , and an image parameter $r \in [R_{\text{full}}(\lambda, d)]$, the setup algorithm samples a hash key hk , which fixes $\mathcal{X}$ and $\mathbb{K}$.
- $h \leftarrow \text{ELTDH.Hash}(\text{hk}, x; \rho)$. The hashing algorithm maps the hash key, an input $x \in \mathcal{X}$, and a random tape ρ to a digest h .
- $(\text{ek}, \text{td}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, f)$. The generation algorithm samples a public encoding key ek and a trapdoor td for a predicate $f \in \mathcal{C}$.
- $e \leftarrow \text{ELTDH.Enc}(\text{ek}, x; \rho)$. The encoding algorithm maps the encoding key, an input $x \in \mathcal{X}$, and the random tape ρ of hashing to an encoding $e \in \mathbb{K}$.

- $e' \leftarrow \text{ELTDH.Dec}(\text{td}, h)$. The deterministic decoding algorithm maps the trapdoor td and a digest h to a decoded value $e' \in \mathbb{K}$.

For fixed keys, we write

$$\begin{aligned}\Phi_{\text{hk}}(x) &:= \text{ELTDH.Hash}(\text{hk}, x; 0), \\ \Delta_{\text{hk}, \text{ek}, \text{td}}(x) &:= \text{ELTDH.Enc}(\text{ek}, x; 0) - \text{ELTDH.Dec}(\text{td}, \Phi_{\text{hk}}(x))\end{aligned}$$

for the hash and for the difference between the encoding and the decoded value at the all-zero tape 0. The scheme satisfies the following properties.

- **Compact Descriptions.** Hash keys, encoding keys, and trapdoors have length $\text{poly}(\lambda + d)$, digests and encodings have length $\text{poly}(\lambda)$ independent of d , and every algorithm runs in time $\text{poly}(\lambda + d)$.
- **Correctness.** For a function $\delta: \mathbb{N} \rightarrow [0, 1]$, the scheme is δ -correct if, for every $\lambda \in \mathbb{N}$, every $d \in \mathbb{N}_{>0}$, every $r \in [R_{\text{full}}(\lambda, d)]$, every $\text{hk} \in \text{supp}(\text{ELTDH.Setup}(1^\lambda, 1^d, r))$, every $f \in \mathcal{C}$, and every $x \in \mathcal{X}$,

$$\Pr \left[e - e' = f(x) \mid \begin{array}{l} \rho \xleftarrow{\$} \{0, 1\}^*, \\ h \leftarrow \text{ELTDH.Hash}(\text{hk}, x; \rho), \\ (\text{ek}, \text{td}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, f), \\ e \leftarrow \text{ELTDH.Enc}(\text{ek}, x; \rho), \\ e' \leftarrow \text{ELTDH.Dec}(\text{td}, h) \end{array} \right] \geq 1 - \delta(\lambda).$$

The scheme is fully correct if it is ε -correct for some negligible function ε , and perfectly correct if it is 0-correct.

- **Lossiness.** There exists a negligible function ν_{los} such that, for every $\lambda \in \mathbb{N}$, every $d \in \mathbb{N}_{>0}$, and every $r \in [R_{\text{full}}(\lambda, d)]$,

$$\Pr \left[|\text{Im}(\Phi_{\text{hk}})| \leq r \mid \text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^d, r) \right] \geq 1 - \nu_{\text{los}}(\lambda).$$

- **Mode Indistinguishability.** Let $d: \mathbb{N} \rightarrow \mathbb{N}_{>0}$ be polynomially bounded, write $d_\lambda := d(\lambda)$, and define

$$\mathfrak{D}_{\lambda, r}^{\text{mode}} := \left\{ \text{hk} \mid \text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^{d_\lambda}, r) \right\}.$$

For every polynomial t and every positive polynomial δ , there exist a polynomial Q and $\lambda_0 \in \mathbb{N}$ such that, for every $\lambda \geq \lambda_0$, the interval $[Q(\lambda) .. R_{\text{full}}(\lambda, d_\lambda)]$ is nonempty and $\mathfrak{D}_{\lambda, r_0}^{\text{mode}} \equiv_{t, 1/\delta} \mathfrak{D}_{\lambda, r_1}^{\text{mode}}$ for all $r_0, r_1 \in [Q(\lambda) .. R_{\text{full}}(\lambda, d_\lambda)]$.

- **Input Privacy.** Let $d: \mathbb{N} \rightarrow \mathbb{N}_{>0}$ be polynomially bounded, and let S be a nonuniform PPT selector that on input $(1^\lambda, \text{hk})$ returns (x^0, x^1, st) with $x^0, x^1 \in \mathcal{X}$. For $b \in \{0, 1\}$, define the ensemble $\mathfrak{D}_b^{\text{in}} = (\mathfrak{D}_{b, \lambda}^{\text{in}})_{\lambda \in \mathbb{N}}$ by

$$\mathfrak{D}_{b, \lambda}^{\text{in}} := \left\{ (\text{hk}, \text{st}, h^b) \mid \begin{array}{l} \text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^{d(\lambda)}, R_{\text{full}}(\lambda, d(\lambda))), \\ (x^0, x^1, \text{st}) \xleftarrow{\$} S(1^\lambda, \text{hk}), \\ \rho \xleftarrow{\$} \{0, 1\}^*, \quad h^b \leftarrow \text{ELTDH.Hash}(\text{hk}, x^b; \rho) \end{array} \right\}.$$

The scheme has input privacy if $\mathfrak{D}_0^{\text{in}} \equiv_c \mathfrak{D}_1^{\text{in}}$ for every such S , statistical input privacy if $\mathfrak{D}_0^{\text{in}} \equiv_s \mathfrak{D}_1^{\text{in}}$ for every such S , and perfect input privacy if $\mathfrak{D}_0^{\text{in}} \equiv \mathfrak{D}_1^{\text{in}}$ for every such S .

- **Function Privacy.** Let $d: \mathbb{N} \rightarrow \mathbb{N}_{>0}$ be polynomially bounded, and let S be a nonuniform PPT selector that on input $(1^\lambda, \text{hk})$ returns (f^0, f^1, st) with $f^0, f^1 \in \mathcal{C}$. For $b \in \{0, 1\}$, define the ensemble $\mathcal{D}_b^{\text{fn}} = (\mathcal{D}_{b,\lambda}^{\text{fn}})_{\lambda \in \mathbb{N}}$ by

$$\mathcal{D}_{b,\lambda}^{\text{fn}} := \left\{ (\text{hk}, \text{st}, \text{ek}^b) \left| \begin{array}{l} \text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^{d(\lambda)}, R_{\text{full}}(\lambda, d(\lambda))), \\ (f^0, f^1, \text{st}) \xleftarrow{\$} S(1^\lambda, \text{hk}), \\ (\text{ek}^b, \text{td}^b) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, f^b) \end{array} \right. \right\}.$$

The scheme is function private if $\mathcal{D}_0^{\text{fn}} \equiv_c \mathcal{D}_1^{\text{fn}}$ for every such S .

- **Rate.** For $\mu: \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$, the scheme has rate at least μ if, for all $\lambda, d, r, \text{hk}, f$, and x quantified as in the correctness clause, every $(\text{ek}, \text{td}) \in \text{supp}(\text{ELTDH.Gen}(\text{hk}, f))$ and every random tape ρ satisfy

$$\frac{\text{len}(f(x))}{\text{len}(\text{ELTDH.Enc}(\text{ek}, x; \rho))} \geq \mu(\lambda),$$

where $\text{len}(u)$ denotes the length of the canonical representation of u . The scheme has rate one if the two lengths are equal for all these objects. $\dashv$

For $\mathcal{X} = \{0, 1\}^d$, $\mathbb{K} = \mathbb{Z}/2\mathbb{Z}$, and the full parameter, a scheme as in [DEFINITION 4.1](#) is a trapdoor-hashing scheme in the sense of [DEFINITION 2.2](#), with setup $\text{ELTDH.Setup}(1^\lambda, 1^d, R_{\text{full}}(\lambda, d))$, encoder ELTDH.Enc , and decoder $\text{TDH.Dec}(\text{td}, h) := (e', e' + 1)$ for $e' \leftarrow \text{ELTDH.Dec}(\text{td}, h)$. Since $e - e' = f(x)$ holds if and only if $e = e' + f(x)$, the candidate indexed by $f(x)$ equals the encoding, and $e' \neq e' + 1$, so we obtain the correctness condition of [DEFINITION 2.3](#). The input-privacy and function-privacy clauses of [DEFINITION 4.1](#) at the full parameter imply those of [DEFINITIONS 2.4](#) and [2.5](#), because the selector of [DEFINITION 4.1](#) may choose the inputs after it sees the hash key.

The lossiness clause of [DEFINITION 4.1](#) concerns only the hash, and its correctness clause holds for each input separately. To obtain a trapdoor extremely lossy function from the scheme, we also need correctness for every input under one key, for an injective predicate and for a constant predicate.

REMARK 4.2 (From Lossy Hashing to Trapdoor Extremely Lossy Functions). Suppose that a predicate f of a scheme is injective with an efficient inverse, that a predicate $f_\star$ of the scheme is constant with value c , and that there is a negligible function ν_{sim} such that both of the following conditions hold.

- For every supported hash key hk at the full parameter, with probability at least $1 - \nu_{\text{sim}}(\lambda)$ over $(\text{ek}, \text{td}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, f)$, we have $\Delta_{\text{hk}, \text{ek}, \text{td}}(x) = f(x)$ for every input x .
- For every image parameter r of the lossy mode and every supported hash key hk at r , with probability at least $1 - \nu_{\text{sim}}(\lambda)$ over $(\text{ek}, \text{td}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, f_\star)$, we have $\Delta_{\text{hk}, \text{ek}, \text{td}}(x) = f_\star(x)$ for every input x .

We define the deterministic public map $F_{\text{hk}, \text{ek}}(x) := (\Phi_{\text{hk}}(x), \text{ELTDH.Enc}(\text{ek}, x; 0))$. At the full parameter and under the first condition, we recover $f(x) = e - \text{ELTDH.Dec}(\text{td}, h)$ from $(h, e) = F_{\text{hk}, \text{ek}}(x)$ and td , and then x with the inverse of f . Under the second condition, $e = c + \text{ELTDH.Dec}(\text{td}, h)$, so

$$F_{\text{hk}, \text{ek}}(x) = J_{\text{td}, c}(\Phi_{\text{hk}}(x)), \quad J_{\text{td}, c}(h) := (h, c + \text{ELTDH.Dec}(\text{td}, h)).$$

Since the first coordinate of $J_{\text{td}, c}(h)$ is h , the map $J_{\text{td}, c}$ is injective and $|\text{Im}(F_{\text{hk}, \text{ek}})| = |\text{Im}(\Phi_{\text{hk}})|$. We first change f to $f_\star$ at the full parameter under function privacy and then change the image parameter of the hash key to r under mode indistinguishability. By hash lossiness and the two

conditions, $F_{\text{hk},\text{ek}}$ is a trapdoor extremely lossy function. $\lrcorner$

4.1 The Damgård–Jurik Construction

We realize [DEFINITION 4.1](#) over a Damgård–Jurik group, with uniform hash bases at the full parameter and bounded powers of a generator at every other parameter. We use the sampler DJGen of [DEFINITION 2.8](#) in every setup, and support quantifiers exclude $\perp$ as in [§2.4](#). We make every construction total by a fallback output on the event that DJGen outputs $\perp$, and on this event, we use one fixed modulus that satisfies the same two-sided size bound as the honest moduli. Since we use the same fallback hash key at every image parameter, the fallback branch adds no term to the bound for mode indistinguishability, and its probability appears only in the bounds that we derive from function privacy.

DEFINITION 4.3 (Fallback Modulus). Let ℓ be the integer-valued polynomial of [DEFINITION 2.8](#), and define $N_{\perp}(\lambda) := 2^{2\ell(\lambda)-1} + 1$. $\lrcorner$

The fallback modulus $N_{\perp}$ is odd and satisfies

$$2^{2\ell(\lambda)-2} < N_{\perp}(\lambda) < 2^{2\ell(\lambda)}, \quad \lceil \log_2 N_{\perp}(\lambda) \rceil = 2\ell(\lambda),$$

and since $2^{\ell(\lambda)-1} < p, q < 2^{\ell(\lambda)}$, every honest modulus $N = pq$ of [DEFINITION 2.8](#) satisfies $2^{2\ell(\lambda)-2} < N < 2^{2\ell(\lambda)}$ as well.

We fix $\lambda \in \mathbb{N}$ and $d \in \mathbb{N}_{>0}$, and we set $w := 2\ell(\lambda)$ and $R_{\text{full}}(\lambda, d) := 2^{dw}$, where $\ell(\lambda)$ is the bit length of each prime factor of a modulus sampled by DJGen in [DEFINITION 2.8](#), so that w is the width of one residue. Every modulus of a hash key is smaller than 2^w , including the fallback modulus $N_{\perp}(\lambda) = 2^{2\ell(\lambda)-1} + 1$ of [DEFINITION 4.3](#), which is the modulus of the hash key when parameter generation fails. For a hash key with modulus N , we take $\mathcal{X} := (\mathbb{Z}/N\mathbb{Z})^d$ and $\mathbb{K} := \mathbb{Z}/N\mathbb{Z}$, and the predicate class $\mathcal{C}^{\text{lin}} = \mathcal{C}_{\lambda,d,\text{hk}}^{\text{lin}}$ consists of the residue-linear predicates

$$L_{\mathbf{a}}(\mathbf{x}) := \sum_{j=1}^d a_j x_j \in \mathbb{Z}/N\mathbb{Z}, \quad \mathbf{a} = (a_1, \dots, a_d) \in (\mathbb{Z}/N\mathbb{Z})^d.$$

We give the algorithms in [FIGURE 4.1](#).

For a nonfull image parameter r , we write $\bar{r} := \min\{r, 2^\lambda\}$, and for $r > 2^\lambda$, we sample the hash key from the distribution for 2^λ , whose image bound 2^λ is smaller than r . With this cap, every exponent bound γ in [FIGURE 4.1](#) satisfies $\gamma < 2^\lambda < |\mathbb{G}_N|$. The hash key contains one base for every bit of the padded binary expansions of the d residues, and for $\text{bin}_w(x_j) = (x_{j,0}, \dots, x_{j,w-1})$, the encoding-key entry indexed by (j, k) is $g_{j,k}^\sigma \exp(2^k a_j)$. Since $\text{bin}_w(x_j)$ is the binary expansion of $\text{rep}(x_j)$, the product of the offsets selected by an input $\mathbf{x}$ is

$$\exp \left(\sum_{j=1}^d \sum_{k=0}^{w-1} 2^k a_j x_{j,k} \right) = \exp \left(\sum_{j=1}^d a_j x_j \right).$$

The same exponent $\rho' := \rho|_{L_{\text{exp}}}$, with $L_{\text{exp}} := \lceil \log_2 N \rceil + \lambda$, appears in hashing and in encoding. We totalize the setup with one fixed fallback key, independent of the image parameter,

$$\text{hk}_{\perp} := ((N_{\perp}(\lambda), 1), d, w, L_{\text{exp}_{\perp}}, 1, (1)_{j,k}), \quad L_{\text{exp}_{\perp}} := \lceil \log_2 N_{\perp}(\lambda) \rceil + \lambda,$$

which the setup returns when DJGen fails and, at the full parameter, also when one of the $dw + 1$ samples from $\mathbb{G}_N$ fails. Every other hash key differs from $\text{hk}_\perp$, because its generator g generates $\mathbb{G}_N \neq \{1\}$. We sample group elements only at the full parameter, so at every nonfull parameter, the setup returns $\text{hk}_\perp$ exactly when DJGen fails. We write $f_{\text{DJ}}(\lambda)$ for the failure probability of $\text{DJGen}(1^\lambda)$, E_{full} for the event that the setup at the full parameter returns $\text{hk}_\perp$, and

$$f_{\text{full}}(\lambda, d) := \Pr \left[\text{hk} = \text{hk}_\perp \mid \text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^d, R_{\text{full}}(\lambda, d)) \right]$$

for its probability. By the union bound, $f_{\text{full}}(\lambda, d) \leq f_{\text{DJ}}(\lambda) + (dw + 1)\nu_{\mathbb{G}}(\lambda)$, where $\nu_{\mathbb{G}}(\lambda)$ bounds the failure probability of one sample from $\mathbb{G}_N$ as in [DEFINITION 2.7](#), so f_{full} is negligible whenever d is polynomially bounded.

ELTDH.Setup($1^\lambda, 1^d, r$) <pre> 00 $w \leftarrow 2\ell(\lambda); R \leftarrow 2^{dw}$ 01 assert $d \geq 1 \wedge r \in [R]$ 02 $(N, g) \xleftarrow{\\$} \text{DJGen}(1^\lambda)$ 03 if $(N, g) = \perp$ return $\text{hk} \leftarrow \text{hk}_\perp$ 04 $L_{\text{exp}} \leftarrow \lceil \log_2 N \rceil + \lambda$ 05 if $r = R$ 06 $g_0 \xleftarrow{\\$} \mathbb{G}_N; (g_{j,k})_{j,k} \xleftarrow{\\$} \mathbb{G}_N^{dw}$ 07 if $g_0 = \perp \vee (\exists j, k: g_{j,k} = \perp)$ return $\text{hk} \leftarrow \text{hk}_\perp$ 08 else: 09 $\bar{r} \leftarrow \min\{r, 2^\lambda\}; \gamma \leftarrow \lfloor (\bar{r} - 1)/(dw) \rfloor$ 10 if $\gamma = 0$ 11 $g_0 \leftarrow 1; (g_{j,k})_{j,k} \leftarrow (1)_{j,k}$ 12 else: 13 $s_0 \xleftarrow{\\$} [\gamma]; g_0 \leftarrow g^{s_0}$ 14 for $(j, k) \in [d] \times [0..w - 1]$ 15 $s_{j,k} \xleftarrow{\\$} [\gamma]; g_{j,k} \leftarrow g^{s_{j,k}}$ 16 return $\text{hk} \leftarrow ((N, g), d, w, L_{\text{exp}}, g_0, (g_{j,k})_{j,k})$ </pre> ELTDH.Hash($\text{hk}, \mathbf{x}; \rho$) <pre> 17 parse $\text{hk} = ((N, g), d, w, L_{\text{exp}}, g_0, (g_{j,k})_{j,k})$ 18 assert $\mathbf{x} \in (\mathbb{Z}/N\mathbb{Z})^d$ 19 for $j \in [d]$ 20 $(x_{j,0}, \dots, x_{j,w-1}) \leftarrow \text{bin}_w(x_j)$ 21 $\rho' \leftarrow \rho _{L_{\text{exp}}}$ 22 return $h \leftarrow g_0^{\rho'} \prod_{j=1}^d \prod_{k=0}^{w-1} g_{j,k}^{x_{j,k}}$ </pre>	ELTDH.Gen($\text{hk}, \mathbf{a}$) <pre> 23 parse $\text{hk} = ((N, g), d, w, L_{\text{exp}}, g_0, (g_{j,k})_{j,k})$ 24 assert $\mathbf{a} = (a_1, \dots, a_d) \in (\mathbb{Z}/N\mathbb{Z})^d$ 25 $\sigma \xleftarrow{\\$} [0..2^{L_{\text{exp}}} - 1]$ 26 $K_0 \leftarrow g_0^\sigma$ 27 for $(j, k) \in [d] \times [0..w - 1]$ 28 $K_{j,k} \leftarrow g_{j,k}^\sigma \exp(2^k a_j)$ 29 $\text{ek} \leftarrow (N, d, w, L_{\text{exp}}, K_0, (K_{j,k})_{j,k}); \text{td} \leftarrow (N, \sigma)$ 30 return (ek, td) </pre> ELTDH.Enc($\text{ek}, \mathbf{x}; \rho$) <pre> 31 parse $\text{ek} = (N, d, w, L_{\text{exp}}, K_0, (K_{j,k})_{j,k})$ 32 assert $\mathbf{x} \in (\mathbb{Z}/N\mathbb{Z})^d$ 33 for $j \in [d]$ 34 $(x_{j,0}, \dots, x_{j,w-1}) \leftarrow \text{bin}_w(x_j)$ 35 $\rho' \leftarrow \rho _{L_{\text{exp}}}$ 36 $c \leftarrow K_0^{\rho'} \prod_{j=1}^d \prod_{k=0}^{w-1} K_{j,k}^{x_{j,k}}$ 37 return $e \leftarrow \text{DDLog}_N(c)$ </pre> ELTDH.Dec(td, h) <pre> 38 parse $\text{td} = (N, \sigma)$ 39 assert $h \in (\mathbb{Z}/N^2\mathbb{Z})^\times$ 40 return $e' \leftarrow \text{DDLog}_N(h^\sigma)$ </pre>
---	--

FIGURE 4.1. Extremely lossy trapdoor hash for residue-linear predicates. Here bin_w is the padded w -bit binary expansion, $\rho|_{L_{\text{exp}}}$ is the integer formed by the first L_{exp} bits of ρ , $[\gamma] = \{1, \dots, \gamma\}$, $\text{hk}_\perp$ is the fallback key, and an algorithm whose **assert** condition fails returns $\perp$.

4.2 Correctness, Input Privacy, and Lossiness

Correctness, input privacy, and lossiness of the construction hold unconditionally, and we need computational assumptions only for function privacy and mode indistinguishability. Correctness follows from the common exponent σ of the encoding key and the trapdoor, input privacy from the uniform randomizer base g_0 at the full parameter, and lossiness from the bounded exponents of the

hash bases at nonfull parameters.

Every element that we multiply in the construction is a unit modulo N^2 . At a modulus that DJGen samples successfully, the published bases g_0 and $g_{j,k}$ lie in the subgroup $\mathbb{G}_N$ of order $p'q'$, and so do every digest h and the randomizer entry $K_0 = g_0^\sigma$, since each of them is a product of powers of those bases. The other encoding-key entries lie in cosets of $\mathbb{G}_N$. The entry $K_{j,k} = g_{j,k}^\sigma \exp(2^k a_j)$ lies in $\mathbb{G}_N \exp(2^k a_j)$, and the encoder product

$$K_0^{\rho'} \prod_{j,k} K_{j,k}^{x_{j,k}} = h^\sigma \exp(L_{\mathbf{a}}(\mathbf{x}))$$

lies in $\mathbb{G}_N \exp(L_{\mathbf{a}}(\mathbf{x}))$. By [ITEM \(c\)](#) of [LEMMA 2.10](#), each of these products has a unique factorization $c \exp(y)$ with $c \in \mathbb{G}_N$ and $y \in \mathbb{Z}/N\mathbb{Z}$, so we can recover the predicate value $L_{\mathbf{a}}(\mathbf{x})$ with DDLog_N without knowing σ . The message space $\mathbb{Z}/N\mathbb{Z}$, which $\exp$ maps isomorphically onto the easy subgroup, contains the ELTDH input residues, the predicate coefficients, the predicate value, and the two shares of the encoder and the decoder.

Two parties who hold $c \exp(x)$ and c for an unknown $c \in (\mathbb{Z}/N^2\mathbb{Z})^\times$ obtain additive shares of x without communication, since DDLog_N maps $c \exp(x)$ and c to two residues whose difference is x . [\[RS21\]](#) and [\[OSY21\]](#) prove this identity, and we state it at the modulus N^2 for every $N \geq 2$, which includes the fallback modulus.

LEMMA 4.4 (Shift Invariance). Let $N \geq 2$ be an integer. For every $c \in (\mathbb{Z}/N^2\mathbb{Z})^\times$ and every $x \in \mathbb{Z}/N\mathbb{Z}$, $\text{DDLog}_N(c \exp(x)) - \text{DDLog}_N(c) = x$. ┐

Proof. Since $\exp(x) = 1 + Nx \equiv 1 \pmod{N}$, we have $c \exp(x) \equiv c \pmod{N}$, so the two arguments have the same representative $w \in [1..N-1]$ in [DEFINITION 2.11](#). Since $\log$ is an isomorphism from $1 + N\mathbb{Z}/N^2\mathbb{Z}$ onto $(\mathbb{Z}/N\mathbb{Z})^+$ by [LEMMA 2.9](#), we obtain

$$\begin{aligned} \text{DDLog}_N(c \exp(x)) - \text{DDLog}_N(c) &= \log(c \exp(x) w^{-1}) - \log(c w^{-1}) \\ &= \log\left(\left(c \exp(x) w^{-1}\right) \left(c w^{-1}\right)^{-1}\right) \\ &= \log(\exp(x)) \\ &= x, \end{aligned}$$

where we use $\log \circ \exp = \text{id}$ in the last equality. ■

Since the encoding key and the trapdoor share the exponent σ , we obtain an exact identity between the two shares from shift invariance of DDLog_N [\[OSY21, RS21\]](#). We adapt this identity of [\[RS21\]](#) to hash bases with bounded exponents and to the fallback key $\text{hk}_\perp$.

THEOREM 4.5 (Exact Share Identity). Let $\lambda \in \mathbb{N}$, $d \in \mathbb{N}_{>0}$, $r \in [R_{\text{full}}(\lambda, d)]$, and $\text{hk} \in \text{supp}(\text{ELTDH.Setup}(1^\lambda, 1^d, r))$ with modulus N . For every $\mathbf{a} \in (\mathbb{Z}/N\mathbb{Z})^d$, every $(\text{ek}, \text{td}) \in \text{supp}(\text{ELTDH.Gen}(\text{hk}, \mathbf{a}))$, every $\mathbf{x} \in (\mathbb{Z}/N\mathbb{Z})^d$, and every tape ρ , the digest $h := \text{ELTDH.Hash}(\text{hk}, \mathbf{x}; \rho)$, the encoding $e := \text{ELTDH.Enc}(\text{ek}, \mathbf{x}; \rho)$, and the decoded value $e' := \text{ELTDH.Dec}(\text{td}, h)$ satisfy $e - e' = L_{\mathbf{a}}(\mathbf{x})$. ┐

Proof. We factor the encoder product as $h^\sigma \exp(L_{\mathbf{a}}(\mathbf{x}))$ and apply shift invariance of DDLog_N . We write $\text{bin}_w(x_j) = (x_{j,0}, \dots, x_{j,w-1})$. By construction, $K_0 = g_0^\sigma$ and $K_{j,k} = g_{j,k}^\sigma \exp(2^k a_j)$, and the

same ρ' appears in hashing and in encoding. Since $\exp$ is a homomorphism, the product c computed by ELTDH.Enc satisfies

$$\begin{aligned} c &= K_0^{\rho'} \prod_{j=1}^d \prod_{k=0}^{w-1} K_{j,k}^{x_{j,k}} \\ &= h^\sigma \exp \left(\sum_{j=1}^d \sum_{k=0}^{w-1} 2^k a_j x_{j,k} \right) \\ &= h^\sigma \exp \left(\sum_{j=1}^d a_j x_j \right), \end{aligned}$$

where the last equality holds because $\text{bin}_w(x_j)$ is the binary expansion of $\text{rep}(x_j)$. Hence, by [LEMMA 4.4](#) at $h^\sigma \in (\mathbb{Z}/N^2\mathbb{Z})^\times$ and $L_{\mathbf{a}}(\mathbf{x}) \in \mathbb{Z}/N\mathbb{Z}$,

$$e - e' = \text{DDLog}_N(h^\sigma \exp(L_{\mathbf{a}}(\mathbf{x}))) - \text{DDLog}_N(h^\sigma) = L_{\mathbf{a}}(\mathbf{x}).$$

The calculation holds for every choice of bases, and [LEMMA 2.9](#) and [4.4](#) hold at every modulus $N \geq 2$, so the identity also holds at $\text{hk}_\perp$. $\blacksquare$

Since the identity holds for every tape, it implies perfect correctness and, at the all-zero tape, one identity for all inputs under one key.

COROLLARY 4.6 (Perfect Correctness). The scheme ELTDH is 0-correct. Moreover, for every $\lambda \in \mathbb{N}$, $d \in \mathbb{N}_{>0}$, $r \in [R_{\text{full}}(\lambda, d)]$, $\text{hk} \in \text{supp}(\text{ELTDH.Setup}(1^\lambda, 1^d, r))$ with modulus N , $\mathbf{a} \in (\mathbb{Z}/N\mathbb{Z})^d$, and $(\text{ek}, \text{td}) \in \text{supp}(\text{ELTDH.Gen}(\text{hk}, \mathbf{a}))$, we have $\Delta_{\text{hk}, \text{ek}, \text{td}}(\mathbf{x}) = L_{\mathbf{a}}(\mathbf{x})$ for every $\mathbf{x} \in (\mathbb{Z}/N\mathbb{Z})^d$. $\lrcorner$

Proof. The encoder returns e and the decoder returns e' , so 0-correctness follows from [THEOREM 4.5](#). At the all-zero tape, we obtain $\Delta_{\text{hk}, \text{ek}, \text{td}}(\mathbf{x}) = L_{\mathbf{a}}(\mathbf{x})$ for every $\mathbf{x} \in (\mathbb{Z}/N\mathbb{Z})^d$ under the same keys from [THEOREM 4.5](#). $\blacksquare$

The digest must hide the input at the full parameter. There, the randomizer base g_0 is uniform in $\mathbb{G}_N$, so it generates $\mathbb{G}_N$ except with probability at most $2^{-\ell(\lambda)+3}$, and for a generator g_0 , the digest is close to uniform on $\mathbb{G}_N$ for every input. We adapt the statistical input-privacy argument of [\[RS21\]](#) to the exponent range $[0..2^{L_{\text{exp}}} - 1]$ and to the fallback key.

THEOREM 4.7 (Statistical Input Privacy). Let $d: \mathbb{N} \rightarrow \mathbb{N}_{>0}$ be polynomially bounded, and let S be a selector as in the input-privacy clause of [DEFINITION 4.1](#). For every $\lambda \in \mathbb{N}$, the statistical distance between $\mathfrak{D}_{0,\lambda}^{\text{in}}$ and $\mathfrak{D}_{1,\lambda}^{\text{in}}$ is at most $2^{-\lambda-1} + 2^{-\ell(\lambda)+4}$. In particular, ELTDH has statistical input privacy. $\lrcorner$

Proof. We compare both ensembles with one reference distribution and apply the triangle inequality. We write $\mathfrak{D}_\lambda^{\text{ref}}$ for the distribution of $(\text{hk}, \text{st}, u)$, where hk and st are sampled as in $\mathfrak{D}_{b,\lambda}^{\text{in}}$, u is uniform in $\mathbb{G}_N$ if $\text{hk} \neq \text{hk}_\perp$, and $u = 1$ if $\text{hk} = \text{hk}_\perp$.

We fix $\text{hk} \neq \text{hk}_\perp$ in the support of the setup at the full parameter and an output (x^0, x^1, st) of $S(1^\lambda, \text{hk})$, and we write $c_b \in \mathbb{G}_N$ for the product of the bases selected by the bits of x^b , so that $h^b = g_0^{\rho'} c_b$. Conditioned on these values, ρ' is uniform on $[0..2^{L_{\text{exp}}} - 1]$. If g_0 generates $\mathbb{G}_N$, then $s \mapsto g_0^s$ is a bijection from $\mathbb{Z}/|\mathbb{G}_N|\mathbb{Z}$ onto $\mathbb{G}_N$, so by [LEMMA 2.12](#) with $M' = |\mathbb{G}_N| \leq N$, the element $g_0^{\rho'}$ is within statistical distance $2^{-\lambda-2}$ of uniform on $\mathbb{G}_N$. Multiplication by c_b is a bijection of $\mathbb{G}_N$,

so h^b is within the same distance of uniform on $\mathbb{G}_N$. If g_0 does not generate $\mathbb{G}_N$, we bound the distance by 1.

Conditioned on $\text{hk} \neq \text{hk}_\perp$ and on the modulus $N = pq$, the base g_0 is uniform in the cyclic group $\mathbb{G}_N$ of order $p'q'$ by [LEMMA 2.10](#). Since $p' \geq 2^{\ell(\lambda)-2}$ and $q' \geq 2^{\ell(\lambda)-2}$, we obtain

$$\Pr[\langle g_0 \rangle \neq \mathbb{G}_N \mid g_0 \xleftarrow{s} \mathbb{G}_N] \leq \frac{1}{p'} + \frac{1}{q'} \leq 2^{-\ell(\lambda)+3}.$$

At $\text{hk}_\perp$, the digest is 1 for both inputs, as in $\mathfrak{D}_\lambda^{\text{ref}}$. Hence, for each $b \in \{0, 1\}$, the statistical distance between $\mathfrak{D}_{b,\lambda}^{\text{in}}$ and $\mathfrak{D}_\lambda^{\text{ref}}$ is at most $2^{-\lambda-2} + 2^{-\ell(\lambda)+3}$, and by the triangle inequality, we obtain the bound of the theorem. $\blacksquare$

At a nonfull parameter, the hash with the all-zero tape depends on the input only through one subset sum of the base exponents, which takes at most r values.

THEOREM 4.8 (Image Bound). Let $\lambda \in \mathbb{N}$, $d \in \mathbb{N}_{>0}$, and $r \in [R_{\text{full}}(\lambda, d)]$. For every $\text{hk} \in \text{supp}(\text{ELTDH.Setup}(1^\lambda, 1^d, r))$, we have $|\text{Im}(\Phi_{\text{hk}})| \leq r$. In particular, ELTDH satisfies the lossiness clause of [DEFINITION 4.1](#) with $\nu_{\text{los}} = 0$. $\dashv$

Proof. We count the values of one bounded subset sum at nonfull parameters and treat the full parameter, the fallback key, and the branch $\gamma = 0$ separately. We write $w = 2\ell(\lambda)$ and $R = 2^{dw}$. If $r = R$, then $|\text{Im}(\Phi_{\text{hk}})| \leq |\mathcal{X}| = N^d < 2^{dw} = r$, since $N < 2^w$. If $r < R$, we write $\bar{r} = \min\{r, 2^\lambda\}$ and $\gamma := \lfloor (\bar{r} - 1)/(dw) \rfloor$.

At $\text{hk}_\perp$ and in the branch $\gamma = 0$, all hash bases are 1, so Φ_{hk} is constant and $|\text{Im}(\Phi_{\text{hk}})| = 1 \leq r$. We suppose that $\gamma \geq 1$, and we write $s_{j,k} \in [\gamma]$ for the exponents of the bases $g_{j,k} = g^{s_{j,k}}$. For $\text{bin}_w(x_j) = (x_{j,0}, \dots, x_{j,w-1})$, we define

$$A_{\mathbf{x}} := \sum_{j=1}^d \sum_{k=0}^{w-1} s_{j,k} x_{j,k}.$$

Then $0 \leq A_{\mathbf{x}} \leq dw\gamma$, and at the all-zero tape, $\Phi_{\text{hk}}(\mathbf{x}) = g^{A_{\mathbf{x}}}$. Hence, Φ_{hk} takes at most $dw\gamma + 1$ values, and $|\text{Im}(\Phi_{\text{hk}})| \leq dw\gamma + 1 \leq \bar{r} \leq r$. $\blacksquare$

We count group elements and bits to verify the compactness and rate clauses of [DEFINITION 4.1](#).

LEMMA 4.9 (Efficiency, Rate, and Compactness). Let $\lambda \in \mathbb{N}$, $d \in \mathbb{N}_{>0}$, and $r \in [R_{\text{full}}(\lambda, d)]$, let $\text{hk} \in \text{supp}(\text{ELTDH.Setup}(1^\lambda, 1^d, r))$ have modulus N and exponent length L_{exp} , and let $w = 2\ell(\lambda)$. Excluding the modulus, the generator, and integer metadata, the hash key contains $dw + 1$ group elements, every encoding key generated from it contains $dw + 1$ group elements, and every corresponding trapdoor contains one exponent of L_{exp} bits. A digest occupies at most $2 \lceil \log_2 N \rceil$ bits and an encoding exactly $\lceil \log_2 N \rceil$ bits. Every algorithm runs in time $\text{poly}(\lambda + d)$, and the scheme has rate one. $\dashv$

Proof. We count the entries of the two public key vectors and their representation lengths. Each vector contains one randomizer entry and one entry for each pair $(j, k) \in [d] \times [0..w-1]$. Since $w = 2\ell(\lambda) = \text{poly}(\lambda)$ and every exponent has at most $L_{\text{exp}} = \text{poly}(\lambda)$ bits, the lengths of both vectors and the work needed to form or evaluate them are polynomial in $\lambda + d$. The digest is one unit modulo N^2 , of at most $2 \lceil \log_2 N \rceil$ bits, and the encoding is one residue modulo N , of $\lceil \log_2 N \rceil$ bits.

Apart from N , the trapdoor consists of $\sigma \in [0..2^{L_{\text{exp}}} - 1]$. The predicate value $L_{\mathbf{a}}(\mathbf{x})$ and the encoder output both lie in $\mathbb{Z}/N\mathbb{Z}$, so they have the same representation length and the rate is one. By [DEFINITION 4.3](#), the same counts hold at $\text{hk}_{\perp}$. ■

4.3 Function Privacy and Mode Indistinguishability

For the two remaining properties, we change the predicate offsets of the encoding key and the distribution of the hash bases, one at a time. With the interactive vector game of [\[BG10\]](#) under decisional composite residuosity [\[Pai99\]](#), as in [\[DGI⁺19\]](#), we remove polynomially many offsets in $1 + N\mathbb{Z}/N^2\mathbb{Z}$ from elements of $\mathbb{G}_N$ with one common exponent, and under tunable DSEDL as in [DEFINITION 3.7](#), we replace bounded bases by uniform ones.

We remove the predicate-dependent offsets from the encoding key with the interactive vector game of [\[BG10\]](#), in which the adversary receives a vector of subgroup elements and then chooses offsets adaptively. [\[BG10\]](#) bounds the advantage in this game under quadratic residuosity, and [\[BG10\]](#) proves the same bound for every subgroup-indistinguishability problem, which we recall in our notation.

LEMMA 4.10 (Interactive Vector Game). Let $\mathbb{G}_U = \mathbb{G}_M \times \mathbb{G}_L$ be a finite abelian group with $\gcd(|\mathbb{G}_M|, |\mathbb{G}_L|) = 1$, where $\mathbb{G}_M$ is cyclic and has a generator that is efficiently computable from the description of $\mathbb{G}_U$. Suppose that the group operations in $\mathbb{G}_U$ and uniform sampling from $\mathbb{G}_M$ and from $\mathbb{G}_L$ are efficient, and let $T \geq |\mathbb{G}_U|$ be a known integer. For an algorithm $\mathcal{B}$, let

$$\varepsilon_{\text{SG}}(\mathcal{B}) := |\Pr[\mathcal{B}(u) = 1 \mid u \xleftarrow{\$} \mathbb{G}_U] - \Pr[\mathcal{B}(u) = 1 \mid u \xleftarrow{\$} \mathbb{G}_L]|.$$

Let $D \in \mathbb{N}_{>0}$. In the interactive D -vector game, the challenger samples $b \xleftarrow{\$} \{0, 1\}$ and $\mathbf{g} \xleftarrow{\$} \mathbb{G}_L^D$ and gives $\mathbf{g}$ to the adversary. On each query $\mathbf{a} \in \mathbb{G}_M^D$, which the adversary chooses adaptively after it receives $\mathbf{g}$, the challenger samples $r \xleftarrow{\$} [T^2]$ and returns the coordinatewise product $\mathbf{a}^{1-b} \cdot \mathbf{g}^r$. Finally, the adversary outputs a bit b' . For an adversary $\mathcal{A}$, let

$$\varepsilon_{\text{IV}}(\mathcal{A}) := |\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]|.$$

Then for every adversary $\mathcal{A}$ that makes at most t queries, there exists an adversary $\mathcal{B}$ such that

$$\varepsilon_{\text{IV}}(\mathcal{A}) \leq 4tD \cdot \varepsilon_{\text{SG}}(\mathcal{B}) + \frac{2tD}{T}.$$

┘

[\[BG10\]](#) gives no bound on the running time of $\mathcal{B}$ in this lemma. In the proof of [\[BG10\]](#), $\mathcal{B}$ runs $\mathcal{A}$ once and otherwise performs $\text{poly}(t, D, \log T)$ group operations in $\mathbb{G}_U$ and samplings, so $\mathcal{B}$ is PPT whenever $\mathcal{A}$ is PPT and t , D , and $\log T$ are polynomially bounded.

Since the adversary chooses the offsets after it receives the bases, the predicates can depend on the modulus, and every coordinate of an answer contains a power of the corresponding base with the same exponent r . By [\[BG10\]](#), the Damgård–Jurik group is an instance, with $\mathbb{G}_U = (\mathbb{Z}/N^2\mathbb{Z})^\times$, with $\mathbb{G}_M = 1 + N\mathbb{Z}/N^2\mathbb{Z}$, which is cyclic of order N and generated by $1 + N$, and with $\mathbb{G}_L$ the subgroup of N -th residues, of order φ . These two orders are coprime, because N is odd and $p', q' \notin \{p, q\}$, and $T = N^2$ is a valid bound, since $|\mathbb{G}_U| = N\varphi < N^2$ by [ITEM \(a\)](#) of [LEMMA 2.10](#).

In our encoding keys, the bases are uniform in $\mathbb{G}_N$, the exponent is uniform in $[0..2^{L_{\text{exp}}} - 1]$, and

the offsets are chosen after the modulus and the bases. The function-privacy arguments of [DGI⁺19] and [RS21] remove such offsets for bases r^N with r uniform in $(\mathbb{Z}/N^2\mathbb{Z})^\times$ and for exponents uniform in $[0..N-1]$. We adapt the offset removal of [BG10, DGI⁺19] to our distributions and reduce it to LEMMA 4.10 at one query.

LEMMA 4.11 (Offset Removal). Suppose that $\text{DJGen}(\cdot)$ satisfies DCR against nonuniform PPT adversaries, and let $d: \mathbb{N} \rightarrow \mathbb{N}_{>0}$ be polynomially bounded. For a nonuniform PPT algorithm $\mathcal{A}$ that on input $(N, \mathbf{g})$ returns a state st and offsets $\beta_1, \dots, \beta_{d(\lambda)} \in \mathbb{Z}/N\mathbb{Z}$, and for $b \in \{0, 1\}$, define the ensemble $\mathfrak{D}_b^{\text{off}} = (\mathfrak{D}_{b,\lambda}^{\text{off}})_{\lambda \in \mathbb{N}}$ by

$$\mathfrak{D}_{b,\lambda}^{\text{off}} := \left\{ \left(N, \mathbf{g}, \text{st}, \left(g_j^\sigma \exp(b\beta_j) \right)_{j \in [d(\lambda)]} \right) \mid \begin{array}{l} (N, g) \xleftarrow{\$} \text{DJGen}(1^\lambda), \\ \mathbf{g} := (g_j)_{j \in [d(\lambda)]} \xleftarrow{\$} \mathbb{G}_N^{d(\lambda)}, \\ (\text{st}, (\beta_j)_j) \xleftarrow{\$} \mathcal{A}(N, \mathbf{g}), \\ \sigma \xleftarrow{\$} [0..2^{L_{\text{exp}}} - 1] \end{array} \right\},$$

where $L_{\text{exp}} = \lceil \log_2 N \rceil + \lambda$, and where the sample is $\perp$ if $\text{DJGen}(1^\lambda)$ or one of the $d(\lambda)$ samples from $\mathbb{G}_N$ returns $\perp$. Then $\mathfrak{D}_1^{\text{off}} \equiv_c \mathfrak{D}_0^{\text{off}}$ for every such $\mathcal{A}$. $\square$

Proof. We reduce to LEMMA 4.10 at one query. Since squaring maps the N -th powers of the game onto $\mathbb{G}_N$ and doubles the offsets, we halve the offsets first, and we then account for the exponent range of the game and for the failures of the sampler of $\mathbb{G}_N$.

THE INSTANCE. For a successful output (N, g) of $\text{DJGen}(1^\lambda)$, we write $N = pq$, $\varphi = (p-1)(q-1)$, and $\mathbb{G}_L := \{z^N \mid z \in (\mathbb{Z}/N^2\mathbb{Z})^\times\}$. We repeat the computation of LEMMA 2.10 with exponent N in place of $2N$ and find $|\mathbb{G}_L| = \varphi$. As we noted after LEMMA 4.10, the hypotheses of that lemma hold for $D := d(\lambda)$, $\mathbb{G}_U := (\mathbb{Z}/N^2\mathbb{Z})^\times$, $\mathbb{G}_M := 1 + N\mathbb{Z}/N^2\mathbb{Z}$, this $\mathbb{G}_L$, and $T := N^2$. Since $z \mapsto z^N$ maps the uniform distribution on $(\mathbb{Z}/N^2\mathbb{Z})^\times$ to the uniform distribution on $\mathbb{G}_L$, distinguishing a uniform element of $\mathbb{G}_U$ from a uniform element of $\mathbb{G}_L$ is distinguishing the two ensembles of DEFINITION 2.13 at the modulus N . The group in LEMMA 4.10 is fixed, and [BG10] proves the lemma for a group output by a generator, with both advantages taken over the generator. We apply it in that form, with the group determined by the modulus of $\text{DJGen}(1^\lambda)$ conditioned on success.

THE REDUCTION. We fix $\mathcal{A}$ and a nonuniform PPT distinguisher $\mathcal{D}$, which outputs a bit on every input, including $\perp$. We define an adversary $\mathcal{C}$ for the interactive D -vector game as follows. On input N and the bases $\mathbf{u} = (u_j)_{j \in [D]} \in \mathbb{G}_L^D$, the adversary $\mathcal{C}$ sets $g_j \leftarrow u_j^2$ for $j \in [D]$ and $\mathbf{g} := (g_j)_{j \in [D]}$, runs $(\text{st}, (\beta_j)_j) \xleftarrow{\$} \mathcal{A}(N, \mathbf{g})$, and makes the single query $\mathbf{a} := (\exp(2^{-1}\beta_j))_{j \in [D]} \in \mathbb{G}_M^D$, where 2^{-1} is the inverse of 2 in $\mathbb{Z}/N\mathbb{Z}$, which exists because N is odd. On the answer $\mathbf{y} = (y_j)_{j \in [D]}$, it outputs $\mathcal{D}(N, \mathbf{g}, \text{st}, (\mathbf{y}_j^2)_{j \in [D]})$. The adversary $\mathcal{C}$ is nonuniform PPT because $\mathcal{A}$ and $\mathcal{D}$ are.

THE SIMULATED DISTRIBUTION. Since $\mathbb{G}_N = \{(z^N)^2 \mid z \in (\mathbb{Z}/N^2\mathbb{Z})^\times\}$, squaring is a surjective homomorphism from $\mathbb{G}_L$ onto $\mathbb{G}_N$, so $\mathbf{g}$ is uniform on $\mathbb{G}_N^D$. For the challenge bit b' of the game, the answer has the coordinates $y_j = \exp(2^{-1}\beta_j)^{1-b'} u_j^r$ with $r \xleftarrow{\$} [T^2]$. Since $\exp$ is a homomorphism and $2 \cdot 2^{-1}\beta_j = \beta_j$ in $\mathbb{Z}/N\mathbb{Z}$,

$$y_j^2 = \exp(\beta_j)^{1-b'} g_j^r \quad (j \in [D]).$$

Hence, for $b = 1 - b'$ and conditioned on the modulus N , the input of $\mathcal{D}$ is distributed as a sample of $\mathfrak{D}_{b,\lambda}^{\text{off}}$ conditioned on N and on the success of the D samples from $\mathbb{G}_N$, except that the exponent is $r \xleftarrow{\$} [T^2]$.

THE EXPONENT RANGE. Every g_j lies in $\mathbb{G}_N$, so the input of $\mathcal{D}$ depends on the exponent only

through its residue modulo $M := |\mathbb{G}_N| \leq N$. For L consecutive integers, the computation in the proof of [LEMMA 2.12](#) bounds the statistical distance of their residue modulo M from uniform on $\mathbb{Z}/M\mathbb{Z}$ by $M/(4L)$. For $r \xleftarrow{\$} [T^2]$, where $L = N^4$, this distance is at most $1/(4N^3) \leq 2^{-\lambda-2}$, and for $\sigma \xleftarrow{\$} [0..2^{L_{\text{exp}}} - 1]$, it is at most $2^{-\lambda-2}$ by [LEMMA 2.12](#). Hence, replacing r by σ changes each acceptance probability of $\mathcal{D}$ by at most $2^{-\lambda-1}$.

THE BOUND. For a successful output of $\text{DJGen}(1^\lambda)$ with modulus N , we write π_N for the probability that $\text{DJGen}(1^\lambda)$ returns a pair with modulus N and $f(N)$ for the probability that one of the D samples from $\mathbb{G}_N$ fails. We write $\Delta(N)$ for the probability that $\mathcal{D}$ outputs 1 on $\mathfrak{D}_{1,\lambda}^{\text{off}}$ minus the probability that it outputs 1 on $\mathfrak{D}_{0,\lambda}^{\text{off}}$, both conditioned on N and on the success of these samples, and $\Delta_{\text{IV}}(N)$ for the probability that $\mathcal{C}$ outputs 1 at $b' = 0$ minus the probability that it outputs 1 at $b' = 1$ in the game at the modulus N . After a failed sample, the output is $\perp$ for both values of b , $f(N) \leq D\nu_{\mathbb{G}}(\lambda)$ by [DEFINITION 2.7](#), and $|\Delta(N) - \Delta_{\text{IV}}(N)| \leq 2^{-\lambda}$ by the exponent range. For the adversary $\mathcal{B}$ that [LEMMA 4.10](#) constructs from $\mathcal{C}$, with output 0 on input $\perp$, we write

$$\varepsilon_{\text{DCR}}(\mathcal{B}) := \left| \Pr \left[\mathcal{B}(X) = 1 \mid X \xleftarrow{\$} \mathfrak{D}_{0,\lambda}^{\text{DCR}} \right] - \Pr \left[\mathcal{B}(X) = 1 \mid X \xleftarrow{\$} \mathfrak{D}_{1,\lambda}^{\text{DCR}} \right] \right|.$$

We sum over the successful outputs of $\text{DJGen}(1^\lambda)$ and apply [LEMMA 4.10](#) with one query and $N > 2^{2\ell(\lambda)-2}$, and we obtain

$$\begin{aligned} & \left| \Pr \left[\mathcal{D}(X) = 1 \mid X \xleftarrow{\$} \mathfrak{D}_{1,\lambda}^{\text{off}} \right] - \Pr \left[\mathcal{D}(X) = 1 \mid X \xleftarrow{\$} \mathfrak{D}_{0,\lambda}^{\text{off}} \right] \right| \\ &= \left| \sum_N \pi_N (1 - f(N)) \Delta(N) \right| \\ &\leq \left| \sum_N \pi_N \Delta_{\text{IV}}(N) \right| + 2^{-\lambda} + D\nu_{\mathbb{G}}(\lambda) \\ &\leq 4D \cdot \varepsilon_{\text{DCR}}(\mathcal{B}) + 2D \cdot 2^{-4\ell(\lambda)+4} + 2^{-\lambda} + D\nu_{\mathbb{G}}(\lambda). \end{aligned}$$

The adversary $\mathcal{B}$ is nonuniform PPT, so $\varepsilon_{\text{DCR}}(\mathcal{B})$ is negligible under DCR, and since $D = d(\lambda)$ is polynomially bounded and $\nu_{\mathbb{G}}$ is negligible, the right-hand side is negligible. $\blacksquare$

We apply [LEMMA 4.11](#) once for each of the two predicates and obtain function privacy at the full parameter, adapting the function-privacy argument of [\[RS21\]](#) to encoding keys with one entry for every bit of every residue and to the fallback key.

THEOREM 4.12 (Function Privacy). Let $d: \mathbb{N} \rightarrow \mathbb{N}_{>0}$ be polynomially bounded. If $\text{DJGen}(\cdot)$ satisfies DCR against nonuniform PPT adversaries, then ELTDH is function private for $\mathcal{C}^{\text{lin}}$ in the sense of [DEFINITION 4.1](#). $\lrcorner$

Proof. We compare both encoding keys with an encoding key without offsets and apply [LEMMA 4.11](#) at each of the two transitions, and we account separately for the fallback event $\mathbf{E}_{\text{full}}$ and for replacing the generator of DJGen with a uniform element of $\mathbb{G}_N$. In the hybrids of [FIGURE 4.2](#), the hash key at the full parameter is sampled in the same way, and we write $\mathbf{g} = (g_{j,k})_{j,k}$ and $\mathbf{K} = (K_{j,k})_{j,k}$.

We fix a selector $\mathbf{S}$ and a distinguisher $\mathcal{A}$, and we write $d = d(\lambda)$, $w = 2\ell(\lambda)$, and $f_{\text{full}}(\lambda) := f_{\text{full}}(\lambda, d)$ for the probability of the fallback event $\mathbf{E}_{\text{full}}$ of [§4.1](#). For $b \in \{0, 1\}$, we write p_b for the probability that $\mathcal{A}$ outputs 1 on a sample of $\mathfrak{D}_{b,\lambda}^{\text{fn}}$. The hybrids make the same calls to DJGen and to the sampler of $\mathbb{G}_N$ as the setup at $R_{\text{full}}(\lambda, d)$, so all these calls succeed with probability $1 - f_{\text{full}}(\lambda)$.

Hyb_0^{fp}	<i>* first predicate *</i>	Hyb_1^{fp}	<i>* remove offsets *</i>	Hyb_2^{fp}	<i>* second predicate *</i>
00 $w \leftarrow 2\ell(\lambda); (N, g) \xleftarrow{\$} \text{DJGen}(1^\lambda)$		09 $w \leftarrow 2\ell(\lambda); (N, g) \xleftarrow{\$} \text{DJGen}(1^\lambda)$		18 $w \leftarrow 2\ell(\lambda); (N, g) \xleftarrow{\$} \text{DJGen}(1^\lambda)$	
01 $L_{\text{exp}} \leftarrow \lceil \log_2 N \rceil + \lambda;$		10 $L_{\text{exp}} \leftarrow \lceil \log_2 N \rceil + \lambda;$		19 $L_{\text{exp}} \leftarrow \lceil \log_2 N \rceil + \lambda;$	
$(g_0, \mathbf{g}) \xleftarrow{\$} \mathbb{G}_N^{dw+1}$		$(g_0, \mathbf{g}) \xleftarrow{\$} \mathbb{G}_N^{dw+1}$		$(g_0, \mathbf{g}) \xleftarrow{\$} \mathbb{G}_N^{dw+1}$	
02 $\text{hk} \leftarrow ((N, g), d, w, L_{\text{exp}}, g_0, \mathbf{g})$		11 $\text{hk} \leftarrow ((N, g), d, w, L_{\text{exp}}, g_0, \mathbf{g})$		20 $\text{hk} \leftarrow ((N, g), d, w, L_{\text{exp}}, g_0, \mathbf{g})$	
03 $(\mathbf{a}^0, \mathbf{a}^1, \text{st}) \xleftarrow{\$} \text{S}(1^\lambda, \text{hk})$		12 $(\mathbf{a}^0, \mathbf{a}^1, \text{st}) \xleftarrow{\$} \text{S}(1^\lambda, \text{hk})$		21 $(\mathbf{a}^0, \mathbf{a}^1, \text{st}) \xleftarrow{\$} \text{S}(1^\lambda, \text{hk})$	
04 $\sigma \xleftarrow{\$} [0..2^{L_{\text{exp}}} - 1]; K_0 \leftarrow g_0^\sigma$		13 $\sigma \xleftarrow{\$} [0..2^{L_{\text{exp}}} - 1]; K_0 \leftarrow g_0^\sigma$		22 $\sigma \xleftarrow{\$} [0..2^{L_{\text{exp}}} - 1]; K_0 \leftarrow g_0^\sigma$	
05 for $(j, k) \in [d] \times [0..w-1]$		14 for $(j, k) \in [d] \times [0..w-1]$		23 for $(j, k) \in [d] \times [0..w-1]$	
06 $K_{j,k} \leftarrow g_{j,k}^\sigma \exp(2^k a_j^0)$		15 $K_{j,k} \leftarrow g_{j,k}^\sigma$		24 $K_{j,k} \leftarrow g_{j,k}^\sigma \exp(2^k a_j^1)$	
07 $\text{ek} \leftarrow (N, d, w, L_{\text{exp}}, K_0, \mathbf{K})$		16 $\text{ek} \leftarrow (N, d, w, L_{\text{exp}}, K_0, \mathbf{K})$		25 $\text{ek} \leftarrow (N, d, w, L_{\text{exp}}, K_0, \mathbf{K})$	
08 return $(\text{hk}, \text{st}, \text{ek})$		17 return $(\text{hk}, \text{st}, \text{ek})$		26 return $(\text{hk}, \text{st}, \text{ek})$	

FIGURE 4.2. Function-privacy hybrids for ELTDH. Each hybrid outputs $(\text{hk}, \text{st}, \text{ek})$, and the highlighted line is the line changed from the previous hybrid.

For $i \in \{0, 1, 2\}$, we write $\tilde{p}_i$ for the probability that $\mathcal{A}$ outputs 1 on the output of Hyb_i^{fp} , conditioned on the success of all these calls, and $q_i := (1 - f_{\text{full}}(\lambda))\tilde{p}_i$.

THE FALLBACK EVENT. Conditioned on the complement of E_{full} , the setup at $R_{\text{full}}(\lambda, d)$ samples (N, g) and the $dw + 1$ bases as the hybrids do, and ELTDH.Gen samples σ from the same range $[0..2^{L_{\text{exp}}} - 1]$. Hence, conditioned on the complement of E_{full} , a sample of $\mathfrak{D}_{0,\lambda}^{\text{fn}}$ is distributed as the output of Hyb_0^{fp} and a sample of $\mathfrak{D}_{1,\lambda}^{\text{fn}}$ as the output of Hyb_2^{fp} , each conditioned on the success of all sampling calls, so

$$p_b = (1 - f_{\text{full}}(\lambda))\tilde{p}_{2b} + f_{\text{full}}(\lambda)p_b^\perp \quad (b \in \{0, 1\}),$$

where $p_b^\perp$ is the probability that $\mathcal{A}$ outputs 1 conditioned on E_{full} , that is, at $\text{hk} = \text{hk}_\perp$. At $\text{hk}_\perp$, the encoding key has the entries $K_{j,k} = \exp(2^k a_j^b)$, so we use only the bound $|p_0^\perp - p_1^\perp| \leq 1$. Hence,

$$|p_0 - p_1| \leq f_{\text{full}}(\lambda) + (1 - f_{\text{full}}(\lambda))|\tilde{p}_0 - \tilde{p}_2| = f_{\text{full}}(\lambda) + |q_0 - q_2|.$$

THE TWO TRANSITIONS. We bound $|q_0 - q_1|$ and $|q_1 - q_2|$ with LEMMA 4.11 at $dw + 1$ coordinates, a number that is polynomially bounded because d and w are. For $i \in \{0, 1\}$, we define an offset-choosing algorithm $\mathcal{B}_i$ and a distinguisher $\mathcal{D}_i$ for that lemma. On input $(N, (g_0, \mathbf{g}))$, the algorithm $\mathcal{B}_i$ samples $z \xleftarrow{\$} \mathbb{Z}/N^2\mathbb{Z}$, sets $g \leftarrow z^{2N}$, $L_{\text{exp}} \leftarrow \lceil \log_2 N \rceil + \lambda$, and $\text{hk} \leftarrow ((N, g), d, w, L_{\text{exp}}, g_0, \mathbf{g})$, runs $(\mathbf{a}^0, \mathbf{a}^1, \text{st}) \xleftarrow{\$} \text{S}(1^\lambda, \text{hk})$, and returns the state (hk, st) and the offsets $(0, (2^k a_j^i)_{(j,k) \in [d] \times [0..w-1]})$. On input $\perp$, the distinguisher $\mathcal{D}_i$ outputs 0, and on input $(N, (g_0, \mathbf{g}), (\text{hk}, \text{st}), (K_0, \mathbf{K}))$, it sets $\text{ek} \leftarrow (N, d, w, L_{\text{exp}}, K_0, \mathbf{K})$ and outputs $\mathcal{A}(\text{hk}, \text{st}, \text{ek})$. Both algorithms are nonuniform PPT because S and $\mathcal{A}$ are. We write $\mathfrak{D}_{b,\lambda}^{\text{off}}$ for the ensemble of LEMMA 4.11 at $dw + 1$ coordinates with the algorithm $\mathcal{B}_i$, and

$$\varepsilon_i(\lambda) := \left| \Pr \left[\mathcal{D}_i(X) = 1 \mid X \xleftarrow{\$} \mathfrak{D}_{1,\lambda}^{\text{off}} \right] - \Pr \left[\mathcal{D}_i(X) = 1 \mid X \xleftarrow{\$} \mathfrak{D}_{0,\lambda}^{\text{off}} \right] \right|,$$

which is negligible by LEMMA 4.11.

A sample of $\mathfrak{D}_{b,\lambda}^{\text{off}}$ is $\perp$ exactly when DJGen or one of the $dw + 1$ samples from $\mathbb{G}_N$ fails, which has probability $f_{\text{full}}(\lambda)$, and then $\mathcal{D}_i$ outputs 0. Conditioned on success, the input of $\mathcal{A}$ is distributed as the output of $\text{Hyb}_{2i}^{\text{fp}}$ for $b = 1$ and as the output of Hyb_1^{fp} for $b = 0$, each conditioned on the success of all sampling calls, except for the generator: it is z^{2N} with z uniform in $\mathbb{Z}/N^2\mathbb{Z}$, whereas

the hybrids use the generator of DJGen. Conditioned on N , both generators are independent of the bases, and the generator of DJGen is uniform among the generators of $\mathbb{G}_N$. By [LEMMA 2.10](#), the element z^{2N} is within statistical distance $\Pr[\gcd(z, N) \neq 1] < 2^{-\ell(\lambda)+2}$ of uniform on $\mathbb{G}_N$, and a uniform element of $\mathbb{G}_N$ fails to generate $\mathbb{G}_N$ with probability at most $1/p' + 1/q' \leq 2^{-\ell(\lambda)+3}$. Hence, conditioned on N , the two generators are within statistical distance $\eta(\lambda) := 2^{-\ell(\lambda)+4}$. Hence, each of the two acceptance probabilities of $\mathcal{D}_i$ differs from the corresponding q by at most $(1 - f_{\text{full}}(\lambda))\eta(\lambda)$, and we obtain

$$|q_i - q_{i+1}| \leq \varepsilon_i(\lambda) + 2(1 - f_{\text{full}}(\lambda))\eta(\lambda) \quad (i \in \{0, 1\}).$$

THE BOUND. By the triangle inequality and $\ell(\lambda) \geq \lambda + 2$,

$$\begin{aligned} |p_0 - p_1| &\leq f_{\text{full}}(\lambda) + \varepsilon_0(\lambda) + \varepsilon_1(\lambda) + 4(1 - f_{\text{full}}(\lambda))\eta(\lambda) \\ &\leq f_{\text{full}}(\lambda) + \varepsilon_0(\lambda) + \varepsilon_1(\lambda) + 2^{-\lambda+4}. \end{aligned}$$

In [§4.1](#), we bounded f_{full} by a negligible function, so $|p_0 - p_1|$ is negligible. ■

For mode indistinguishability, we change only the distribution of the hash bases, from bounded powers to uniform elements, with one application of [LEMMA 3.9](#) to all $dw + 1$ bases. Since $\bar{r} \leq 2^\lambda$ at every nonfull parameter, we have $\gamma < 2^\lambda$, so γ is smaller than the order of every Damgård–Jurik subgroup $\mathbb{G}_N$, as that lemma requires. The same inequality places these parameters within the scope of [THEOREM 3.11](#), since $\ell(\lambda) \geq \lambda + 2$ and the smallest prime factor π_q of $|\mathbb{G}_N| = p'q'$ is at least $2^{\ell(\lambda)-2}$, so $\gamma < 2^\lambda \leq \pi_q$.

THEOREM 4.13 (Mode Indistinguishability). Let $d: \mathbb{N} \rightarrow \mathbb{N}_{>0}$ be polynomially bounded. If DJGen($\cdot$) satisfies tunable DSEDL, then ELTDH satisfies the mode-indistinguishability clause of [DEFINITION 4.1](#) at the family d . ┘

Proof. We change all $dw + 1$ hash bases at once with [LEMMA 3.9](#), choose the image threshold so that every nonfull parameter above it has an exponent bound above the threshold of that lemma, and compare two arbitrary parameters through the full one.

We fix a polynomial t and a positive polynomial δ , write $d = d(\lambda)$ and $w = 2\ell(\lambda)$, and choose a polynomial p_d with $p_d(\lambda) \geq d(\lambda)$ for every λ . We write $D(\lambda) := d(\lambda)w(\lambda) + 1$ and choose a polynomial p_{hk} that bounds the size of a circuit computing the map Ψ that sends $\perp$ to $\text{hk}_\perp$ and a challenge $(1^\lambda, (N, g), \gamma, (G_0, (G_{j,k})_{j,k}))$ to the hash key $((N, g), d, w, \lceil \log_2 N \rceil + \lambda, G_0, (G_{j,k})_{j,k})$. We apply [LEMMA 3.9](#) to the cyclic group sampler DJGen of [LEMMA 2.10](#) with dimension D , size bound $t + p_{\text{hk}}$, and target polynomial 2δ , and obtain a polynomial Q_{vec} and a threshold $\lambda_{\text{vec}} \in \mathbb{N}$. We choose an integer-valued polynomial $\bar{Q}$ with $\bar{Q}(\lambda) \geq \max\{1, \lceil Q_{\text{vec}}(\lambda) \rceil\}$ for every λ and define the polynomial

$$Q(\lambda) := 2p_d(\lambda)\ell(\lambda)\bar{Q}(\lambda) + 1.$$

Since Q is a polynomial, there is $\lambda_1 \in \mathbb{N}$ with $Q(\lambda) \leq 2^\lambda$ for every $\lambda \geq \lambda_1$, and we take $\lambda_0 := \max\{\lambda_{\text{vec}}, \lambda_1\}$.

We fix $\lambda \geq \lambda_0$. Since $d \geq 1$ and $2\ell(\lambda) \geq 2\lambda + 4$, we have $R_{\text{full}}(\lambda, d) = 2^{dw} \geq 2^{2\ell(\lambda)} > 2^\lambda \geq Q(\lambda)$, so the interval $[Q(\lambda), R_{\text{full}}(\lambda, d)]$ is nonempty. For $r \in [Q(\lambda), R_{\text{full}}(\lambda, d) - 1]$, we write $\bar{r} = \min\{r, 2^\lambda\}$, so $\bar{r} \geq Q(\lambda)$, and the exponent bound of the setup at r satisfies

$$\gamma = \left\lfloor \frac{\bar{r} - 1}{dw} \right\rfloor \geq \left\lfloor \frac{p_d(\lambda)w\bar{Q}(\lambda)}{dw} \right\rfloor \geq \bar{Q}(\lambda) \geq \max\{1, Q_{\text{vec}}(\lambda)\}.$$

Moreover, $\gamma \leq \bar{r} - 1 < 2^\lambda$, and every modulus output by $\text{DJGen}(1^\lambda)$ satisfies $|\mathbb{G}_N| = p'q' > 2^{2\ell(\lambda)-4} \geq 2^{2\lambda}$ by [LEMMA 2.10](#), so γ is smaller than the order of every group in the support of $\text{DJGen}(1^\lambda)$. Since also $\lambda \geq \lambda_{\text{vec}}$, [LEMMA 3.9](#) implies

$$\mathfrak{D}_{0,\lambda,\gamma}^{\text{vDSEDL}} \equiv_{t+p_{\text{hk}},1/(2\delta)} \mathfrak{D}_{1,\lambda,\gamma}^{\text{vDSEDL}}$$

at dimension $D(\lambda)$.

The map Ψ sends $\mathfrak{D}_{0,\lambda,\gamma}^{\text{vDSEDL}}$ to $\mathfrak{D}_{\lambda,r}^{\text{mode}}$, because the setup at r returns $\text{hk}_\perp$ when DJGen fails and otherwise publishes $g_0 = g^{s_0}$ and $g_{j,k} = g^{s_{j,k}}$ for independent $s_0, s_{j,k} \xleftarrow{\$} [\gamma]$, with this $\gamma \geq 1$. It sends $\mathfrak{D}_{1,\lambda,\gamma}^{\text{vDSEDL}}$ to $\mathfrak{D}_{\lambda,R_{\text{full}}(\lambda,d)}^{\text{mode}}$, because the setup at $R_{\text{full}}(\lambda,d)$ returns $\text{hk}_\perp$ when DJGen or one of the $dw+1$ samples from $\mathbb{G}_N$ fails and otherwise publishes $dw+1$ independent uniform elements of $\mathbb{G}_N$, and the challenge of $\mathfrak{D}_{1,\lambda,\gamma}^{\text{vDSEDL}}$ is $\perp$ in the same event and otherwise consists of the same elements. For every circuit C of size at most $t(\lambda)$, the composition of C with Ψ has size at most $t(\lambda) + p_{\text{hk}}(\lambda)$, so

$$\mathfrak{D}_{\lambda,r}^{\text{mode}} \equiv_{t,1/(2\delta)} \mathfrak{D}_{\lambda,R_{\text{full}}(\lambda,d)}^{\text{mode}}.$$

For $r_0, r_1 \in [Q(\lambda) \dots R_{\text{full}}(\lambda,d)]$, we compare each nonfull one with $R_{\text{full}}(\lambda,d)$, and by the triangle inequality, $\mathfrak{D}_{\lambda,r_0}^{\text{mode}} \equiv_{t,1/\delta} \mathfrak{D}_{\lambda,r_1}^{\text{mode}}$. This holds for every $\lambda \geq \lambda_0$, which is the mode-indistinguishability clause of [DEFINITION 4.1](#). $\blacksquare$

We combine [THEOREMS 4.5, 4.7, 4.8, 4.12](#) and [4.13](#) and [LEMMA 4.9](#) to verify every clause of [DEFINITION 4.1](#).

THEOREM 4.14 (Extremely Lossy Trapdoor Hash). Suppose that $\text{DJGen}(\cdot)$ satisfies DCR against nonuniform PPT adversaries and tunable DSEDL. Then, for every polynomially bounded $d: \mathbb{N} \rightarrow \mathbb{N}_{>0}$, the construction of [FIGURE 4.1](#) is a perfectly correct, statistically input-private, function-private, rate-one extremely lossy trapdoor-hashing scheme for residue-linear predicates, with $\nu_{\text{os}} = 0$. It also satisfies the perfect simultaneous identity of [COROLLARY 4.6](#) at every image parameter. $\lrcorner$

Proof. We obtain compact descriptions, the rate, and the running-time bounds from [LEMMA 4.9](#). Perfect correctness follows from [COROLLARY 4.6](#), statistical input privacy from [THEOREM 4.7](#), and lossiness from [THEOREM 4.8](#). Function privacy follows from [THEOREM 4.12](#) and mode indistinguishability from [THEOREM 4.13](#). Hence, ELTDH satisfies every clause of [DEFINITION 4.1](#). $\blacksquare$

At nonfull parameters with polynomially bounded γ , we can recover the offsets from the encoding key, so we require function privacy only at $R_{\text{full}}(\lambda,d)$.

REMARK 4.15 (Offset Recovery for Polynomially Bounded γ). At $\text{hk}_\perp$ and in the branch $\gamma = 0$, the encoding-key entries are $K_{j,k} = \exp(2^k a_j)$, which determine $\mathbf{a}$. We suppose that the setup returned a hash key other than $\text{hk}_\perp$ at a nonfull parameter with polynomially bounded $\gamma \geq 1$. With baby-step-giant-step [[Sha71](#)], we recover s_0 from $g_0 = g^{s_0}$ and $s_{j,0}$ from $g_{j,0} = g^{s_{j,0}}$, for every $j \in [d]$, with $\mathcal{O}(\sqrt{\gamma})$ group operations each. The corresponding encoding-key entries are $K_0 = g^{s_0\sigma}$ and $K_{j,0} = g^{s_{j,0}\sigma} \exp(a_j)$, so $K_{j,0} K_0^{-s_{j,0}} = \exp(s_0 a_j)$. Since $s_0 \leq \gamma < 2^\lambda$ and the prime factors of N exceed $2^{\ell(\lambda)-1} \geq 2^{\lambda+1}$, the integer s_0 is invertible modulo N , and we recover $a_j = s_0^{-1} \log(\exp(s_0 a_j))$ exactly. Hence, at such a parameter, a selector that returns two distinct coefficient vectors $\mathbf{a}^0 \neq \mathbf{a}^1$ and a distinguisher that recovers the coefficient vector from the encoding key have advantage one, so function privacy fails there. $\lrcorner$

An ELF that is injective in one mode has at least 2^n outputs on $\{0, 1\}^n$, and since fewer than 2^n bit strings have length less than n , some output has length at least n . In contrast, the public map of ELTDH has outputs of length $\text{poly}(\lambda)$ for every arity d , and with the trapdoor, we recover the predicate value from each output at a lossy parameter.

COROLLARY 4.16 (Image and Output Length of the Public Map). Let $\lambda \in \mathbb{N}$ and $d \in \mathbb{N}_{>0}$, let $r \in [R_{\text{full}}(\lambda, d)]$ and $\text{hk} \in \text{supp}(\text{ELTDH.Setup}(1^\lambda, 1^d, r))$ with modulus N , let $\mathbf{a} \in (\mathbb{Z}/N\mathbb{Z})^d$ and $(\text{ek}, \text{td}) \in \text{supp}(\text{ELTDH.Gen}(\text{hk}, \mathbf{a}))$, and let $F_{\text{hk}, \text{ek}}(\mathbf{x}) := (\Phi_{\text{hk}}(\mathbf{x}), \text{ELTDH.Enc}(\text{ek}, \mathbf{x}; 0))$ be the public map of [REMARK 4.2](#). Then $|F_{\text{hk}, \text{ek}}(\mathcal{D})| \leq r \cdot |L_{\mathbf{a}}(\mathcal{D})|$ for every $\mathcal{D} \subseteq (\mathbb{Z}/N\mathbb{Z})^d$, every $\mathbf{x} \in (\mathbb{Z}/N\mathbb{Z})^d$ satisfies $e - \text{ELTDH.Dec}(\text{td}, h) = L_{\mathbf{a}}(\mathbf{x})$ at $(h, e) = F_{\text{hk}, \text{ek}}(\mathbf{x})$, and every output of $F_{\text{hk}, \text{ek}}$ has at most $3 \lceil \log_2 N \rceil$ bits. $\lrcorner$

Proof. Since the share identity determines the second coordinate from the digest and the predicate value, the image has at most the product of their two ranges. By [COROLLARY 4.6](#), every $\mathbf{x}$ satisfies $e - \text{ELTDH.Dec}(\text{td}, h) = L_{\mathbf{a}}(\mathbf{x})$ at the all-zero tape, so e is determined by h and $L_{\mathbf{a}}(\mathbf{x})$. Hence, $F_{\text{hk}, \text{ek}}$ factors through $\mathbf{x} \mapsto (\Phi_{\text{hk}}(\mathbf{x}), L_{\mathbf{a}}(\mathbf{x}))$, and $|F_{\text{hk}, \text{ek}}(\mathcal{D})| \leq |\Phi_{\text{hk}}(\mathcal{D})| \cdot |L_{\mathbf{a}}(\mathcal{D})| \leq r \cdot |L_{\mathbf{a}}(\mathcal{D})|$ by [THEOREM 4.8](#). By [LEMMA 4.9](#), a digest has at most $2 \lceil \log_2 N \rceil$ bits and an encoding has $\lceil \log_2 N \rceil$ bits, so every output has at most $3 \lceil \log_2 N \rceil$ bits. $\blacksquare$

We take $\mathcal{D} = \{0, 1\}^d$ and $\mathbf{a} = (1, \dots, 1)$, so that $L_{\mathbf{a}}(\mathbf{x})$ is the Hamming weight of $\mathbf{x}$, and we take $6\ell(\lambda) < d < 2^{2\ell(\lambda)-2}$. Every modulus N of a hash key satisfies $2^{2\ell(\lambda)-2} < N < 2^{2\ell(\lambda)}$, so $3 \lceil \log_2 N \rceil \leq 6\ell(\lambda) < d$ and $d < N$. Since $d < N$, the $d+1$ possible weights are distinct in $\mathbb{Z}/N\mathbb{Z}$, so $|L_{\mathbf{a}}(\mathcal{D})| = d+1$, and since $d > 3 \lceil \log_2 N \rceil$, every output of $F_{\text{hk}, \text{ek}}$ is shorter than the d -bit input. Hence, at a polynomial image parameter r , the map $F_{\text{hk}, \text{ek}}$ has at most $r(d+1)$ outputs on a domain of 2^d points, and with the trapdoor, we recover the Hamming weight from each output.

4.4 Rate-One Trapdoor Hashing

The predicate value of ELTDH and its encoding both lie in $\mathbb{Z}/N\mathbb{Z}$, so ELTDH has rate one in the sense of the rate clause of [DEFINITION 4.1](#) by [LEMMA 4.9](#). Trapdoor hashing in the sense of [DEFINITION 2.2](#) has bit-valued predicates, so we restrict ELTDH to bit inputs, $\mathbb{F}_2$ -linear predicates, and one-bit encodings, and we recover the rate-one trapdoor hash of [\[RS21\]](#). For this restriction, we measure the rate with the definition of [\[DGI⁺19\]](#), which we state for the syntax of [DEFINITION 2.2](#).

DEFINITION 4.17 (Rate). Let $\mu: \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$. A trapdoor-hashing scheme THF for $\mathcal{C}$ has rate at least μ if $\text{len}(f(x)) / \text{len}(\text{TDH.Enc}(\text{ek}, x; \rho)) \geq \mu(\lambda)$ for every $\lambda, n \in \mathbb{N}$, every $x \in \{0, 1\}^n$, every $\text{hk} \in \text{supp}(\text{TDH.Setup}(1^\lambda, 1^n))$, every $f \in \mathcal{C}_{\lambda, n, \text{hk}}$, every $(\text{ek}, \text{td}) \in \text{supp}(\text{TDH.Gen}(\text{hk}, f))$, and every random tape ρ , where $\text{len}(u)$ denotes the length of the canonical representation of u . The scheme has rate one if $\text{len}(f(x)) = \text{len}(\text{TDH.Enc}(\text{ek}, x; \rho))$ for all these objects. $\lrcorner$

To obtain one-bit encodings, we add a uniform mask $\alpha \in \mathbb{Z}/N\mathbb{Z}$ to both shares of [THEOREM 4.5](#) and keep the parities of their representatives. We adapt the rate-one trapdoor hash of [\[RS21\]](#) in this way to the bounded-exponent hash keys of ELTDH, and the parities differ by the predicate value except with the probability of a wraparound.

COROLLARY 4.18 (Rate-One Trapdoor Hash for $\mathbb{F}_2$ -Linear Predicates). Let $n: \mathbb{N} \rightarrow \mathbb{N}$ be polynomially bounded with $n(\lambda) \geq 1$ for every λ , and suppose that $\text{DJGen}(\cdot)$ satisfies DCR against nonuniform PPT adversaries. Take ELTDH at arity $d = n(\lambda)$ and at the full parameter, restrict the input coordinates and the coefficient vector $\mathbf{a}$ to $\{0, 1\}$, and take $\text{rep}(L_{\mathbf{a}}(\mathbf{x})) \bmod 2$ as the $\mathbb{F}_2$ -linear

predicate. Generation additionally samples $\alpha \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$ and places it in both **ek** and **td**. The encoder returns $t := \text{rep}(e + \alpha) \bmod 2$. Given $e' \leftarrow \text{ELTDH.Dec}(\text{td}, h)$, the decoder sets $t_0 := \text{rep}(e' + \alpha) \bmod 2$ and returns the candidate pair $(t_0, t_0 \oplus 1)$. The resulting trapdoor-hashing scheme is δ_{par} -correct, statistically input-private, function-private, and rate one, where $\delta_{\text{par}}(\lambda) := \min\{1, n(\lambda)2^{2-2\ell(\lambda)}\}$. Its digest is one element of $(\mathbb{Z}/N^2\mathbb{Z})^\times$. $\lrcorner$

Proof. The only correctness error comes from a wraparound when we add the integer predicate value to the masked decoded share. Since the mask is uniform, the masked share is uniform, so we first bound this event and then obtain privacy and rate from [THEOREMS 4.7](#) and [4.12](#).

For a bit input $\mathbf{x}$ and a bit coefficient vector $\mathbf{a}$, we write

$$z := \sum_{j=1}^{n(\lambda)} a_j x_j \in [0..n(\lambda)],$$

so that $\text{rep}(L_{\mathbf{a}}(\mathbf{x})) = z$. By [THEOREM 4.5](#), $e = e' + z$ in $\mathbb{Z}/N\mathbb{Z}$. Since α is uniform in $\mathbb{Z}/N\mathbb{Z}$ and independent of e' , the element $e' + \alpha$ is uniform in $\mathbb{Z}/N\mathbb{Z}$. If $\text{rep}(e' + \alpha) + z < N$, then $\text{rep}(e + \alpha) = \text{rep}(e' + \alpha) + z$, so $t = t_0 \oplus (z \bmod 2)$, which is the candidate indexed by $z \bmod 2 = \text{rep}(L_{\mathbf{a}}(\mathbf{x})) \bmod 2$, and the two candidates differ. The complementary event $\text{rep}(e' + \alpha) + z \geq N$ has probability $z/N \leq n(\lambda)/N$. Every supported modulus exceeds $2^{2\ell(\lambda)-2}$, so the failure probability is at most $\delta_{\text{par}}(\lambda)$.

The mask α is independent of the predicate and of the remaining key material, and the digest is the digest of ELTDH. Therefore, statistical input privacy follows from [THEOREM 4.7](#), and function privacy follows from [THEOREM 4.12](#), since a reduction can append a fresh mask to the encoding key of ELTDH. The predicate value and the encoding are both one bit, so the rate is one. $\blacksquare$

The error δ_{par} is negligible, whereas the composite-residuosity variant of [\[DGI⁺19\]](#) has an inverse-polynomial error. From quadratic residuosity, [\[DGI⁺19\]](#) constructs a fully correct rate-one trapdoor hash for $\mathbb{F}_2$ -linear predicates with a message group of order two. The composite-residuosity variant of [\[DGI⁺19\]](#) has a message group of order N , and the compilation of its group-valued encoding to one bit in [\[DGI⁺19\]](#) has an inverse-polynomial error. The work [\[RS21\]](#) replaces the distance function of [\[DGI⁺19\]](#) by DDLog_N , which reduces the error to the probability of a wraparound, and for the scheme of [COROLLARY 4.18](#), this probability is at most $n(\lambda)/N$.

5 Extremely Lossy Functions

From the extremely lossy trapdoor hash of [§4](#) at residue arity one, we obtain a trapdoor ELF on the domain $[m]$ by hashing the single residue $x = u - 1$ of an input $u \in [m]$, with the identity predicate in the injective mode and the zero predicate in the lossy mode.

The injective generator of a trapdoor extremely lossy function also returns an inversion trapdoor. We use the trapdoor extremely lossy functions of [\[Zha19\]](#) with four changes. We write m and $N_{\text{out}}(m)$ for the domain size and the bound on the range, which [\[Zha19\]](#) calls M and N , because N denotes a modulus in this paper. We replace the inverse function by an inversion trapdoor and a deterministic inversion algorithm, and we give evaluation as a separate deterministic algorithm. We bound the lengths of descriptions, trapdoors, and outputs by $\text{poly}(\log m)$. We state mode indistinguishability for probabilistic circuits of size t , where [\[Zha19\]](#) quantifies over adversaries whose running time is at most a polynomial p .

DEFINITION 5.1 (Trapdoor ELF). For every $m \geq 2$, write $n_m := \lceil \log_2 m \rceil$. A trapdoor extremely lossy function $\text{TELF} = (\text{TELF.GenInj}, \text{TELF.GenLos}, \text{TELF.Eval}, \text{TELF.Inv})$ consists of a function N_{out} with $\log N_{\text{out}}(m) = \text{poly}(n_m)$ and the following four algorithms.

- $(f, \text{td}) \xleftarrow{\$} \text{TELF.GenInj}(m)$. On input $m \geq 2$, the probabilistic injective-generation algorithm samples a public description f of a function $f: [m] \rightarrow [N_{\text{out}}(m)]$ and an inversion trapdoor td .
- $f \xleftarrow{\$} \text{TELF.GenLos}(m, r)$. On input $m \geq 2$ and $r \in [m]$, the probabilistic lossy-generation algorithm samples a public description f of a function $f: [m] \rightarrow [N_{\text{out}}(m)]$.
- $y \leftarrow \text{TELF.Eval}(f, u)$. On input a public description f and $u \in [m]$, the deterministic evaluation algorithm outputs $y = f(u)$.
- $u \leftarrow \text{TELF.Inv}(f, \text{td}, y)$. On input a public description f , a trapdoor td , and $y \in [N_{\text{out}}(m)]$, the deterministic inversion algorithm outputs a candidate $u \in [m]$ or $\perp$.

We identify a public description f with the function $u \mapsto \text{TELF.Eval}(f, u)$. The tuple TELF satisfies the following four properties.

- **Compact Descriptions.** Public descriptions, trapdoors, and outputs have length $\text{poly}(n_m)$, and every algorithm of TELF runs in time polynomial in the length of its input.
- **Injective Mode.** If $(f, \text{td}) \xleftarrow{\$} \text{TELF.GenInj}(m)$, then except with probability negligible in n_m , every $u \in [m]$ satisfies $\text{TELF.Inv}(f, \text{td}, \text{TELF.Eval}(f, u)) = u$.
- **Lossy Mode.** For every $r \in [m]$, if $f \xleftarrow{\$} \text{TELF.GenLos}(m, r)$, then $|\text{Im}(f)| \leq r$ except with probability negligible in n_m .
- **Mode Indistinguishability.** For every polynomial t and every positive polynomial δ , there exists a polynomial Q such that for every sufficiently large m and every $r \in [Q(n_m)..m]$,

$$\left\{ f \mid (f, \text{td}) \xleftarrow{\$} \text{TELF.GenInj}(m) \right\} \equiv_{t, 1/\delta} \left\{ f \mid f \xleftarrow{\$} \text{TELF.GenLos}(m, r) \right\},$$

where t and δ are evaluated at n_m . J

For some applications of extremely lossy functions, [Zha16] also requires an efficient algorithm that outputs the lossy image, and it distinguishes an algorithm that outputs a superset of the image from one that outputs the image exactly. We use these two notions of [Zha16], with the overwhelming probability written as $1 - \varepsilon(\lceil \log_2 m \rceil)$ for a negligible function ε .

DEFINITION 5.2 (Enumerable Image). Let $\text{ELF} = (\text{ELF.Gen}, \text{ELF.Eval})$ be an extremely lossy function. For a randomized algorithm Enumerate , integers $m \geq 2$, and $r \in [m]$, define

$$p_{\subseteq}^{\text{Enumerate}}(m, r) := \Pr \left[\text{Im}(f) \subseteq S \mid \begin{array}{l} f \xleftarrow{\$} \text{ELF.Gen}(m, r), \\ S \xleftarrow{\$} \text{Enumerate}(f, r) \end{array} \right],$$

and define $p_{=}^{\text{Enumerate}}(m, r)$ in the same way, with the event $\text{Im}(f) = S$. The image of ELF is weakly efficiently enumerable if there exist a randomized algorithm Enumerate running in time $\text{poly}(r, \log m)$ and a negligible function ε such that $p_{\subseteq}^{\text{Enumerate}}(m, r) \geq 1 - \varepsilon(\lceil \log_2 m \rceil)$ for every $m \geq 2$ and $r \in [m]$. It is strongly efficiently enumerable if the same holds with $p_{=}^{\text{Enumerate}}$ in place of $p_{\subseteq}^{\text{Enumerate}}$. J

Following [Zha16], we apply the same two notions to the lossy generator TELF.GenLos of a trapdoor extremely lossy function.

DEFINITION 5.3 (Enumerable Trapdoor ELF). A trapdoor extremely lossy function TELF has a weakly (respectively, strongly) efficiently enumerable image if DEFINITION 5.2 holds for TELF with $\text{ELF.Gen}(m, r)$ replaced by $\text{TELF.GenLos}(m, r)$. J

5.1 The Trapdoor Construction

For every $m \geq 2$, we choose the security parameter of the hash as a function of $n := \lceil \log_2 m \rceil$. We fix an efficiently computable, strictly increasing, polynomially bounded function $\lambda_{\text{grp}}: \mathbb{N} \rightarrow \mathbb{N}$ with $\lambda_{\text{grp}}(n) \geq n$ for every $n \in \mathbb{N}$, and we set $\lambda := \lambda_{\text{grp}}(n)$, $w_\lambda := 2\ell(\lambda)$, and $R_\lambda := 2^{w_\lambda}$. Every modulus N of a hash key in the support of $\text{ELTDH.Setup}(1^\lambda, 1^1, \cdot)$, including the fallback modulus, satisfies $m \leq 2^n < N$, so for every $u \in [m]$, the least nonnegative representative of the residue $u - 1 \in \mathbb{Z}/N\mathbb{Z}$ is $u - 1$.

For a public description fk of this construction, we write $f_{\text{fk}}: [m] \rightarrow (\mathbb{Z}/N^2\mathbb{Z})^\times \times \mathbb{Z}/N\mathbb{Z}$ for the function described by fk . We encode the first coordinate of an output in exactly $4\ell(\lambda)$ bits and the second in exactly $2\ell(\lambda)$ bits, with leading zeros, and we interpret the resulting $6\ell(\lambda)$ -bit string as its nonnegative integer value plus one. Hence, under this fixed-width encoding, every output lies in $[N_T(m)]$ for $N_T(m) := 2^{6\ell(\lambda_{\text{grp}}(\lceil \log_2 m \rceil))}$, and $\log N_T(m) = \text{poly}(\log m)$, as required in [DEFINITION 5.1](#).

Throughout this subsection, ELTDH denotes the construction of [THEOREM 4.14](#) at residue arity $d = 1$, and we write $\text{id}(x) = x$ and $0(x) = 0$ for the two linear predicates of the construction. In injective generation, we take a hash key at the full image parameter R_λ and an encoding key for id , so that the trapdoor recovers $x = u - 1$ from one decoded share. In lossy generation with $r < m$, we take a hash key at the image parameter r and an encoding key for 0 .

TELF.GenInj(m) 00 assert $m \geq 2$ 01 $n \leftarrow \lceil \log_2 m \rceil$; $\lambda \leftarrow \lambda_{\text{grp}}(n)$ 02 $R_\lambda \leftarrow 2^{2\ell(\lambda)}$ 03 $\text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^1, R_\lambda)$ 04 $(\text{ek}, \text{td}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (1))$ 05 $\text{fk} \leftarrow (m, n, \lambda, \text{hk}, \text{ek})$ 06 return (fk, td)	TELF.Eval(fk, u) 15 parse $\text{fk} = (m, n, \lambda, \text{hk}, \text{ek})$ 16 parse $\text{hk} = ((N, g), 1, w_\lambda, L_{\text{exp}}, g_0, (g_{1,k})_k)$ 17 assert $u \in [m]$ 18 $x \leftarrow u - 1 \in \mathbb{Z}/N\mathbb{Z}$ 19 $h \leftarrow \text{ELTDH.Hash}(\text{hk}, (x); 0)$ 20 $e \leftarrow \text{ELTDH.Enc}(\text{ek}, (x); 0)$ 21 return $y \leftarrow (h, e)$
TELF.GenLos(m, r) 07 assert $m \geq 2 \wedge r \in [m]$ 08 if $r = m$ 09 $(\text{fk}, \text{td}) \xleftarrow{\$} \text{TELF.GenInj}(m)$ 10 return fk 11 $n \leftarrow \lceil \log_2 m \rceil$; $\lambda \leftarrow \lambda_{\text{grp}}(n)$ 12 $\text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^1, r)$ 13 $(\text{ek}, \text{td}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (0))$ 14 return $\text{fk} \leftarrow (m, n, \lambda, \text{hk}, \text{ek})$	TELF.Inv(fk, td, y) 22 parse $\text{fk} = (m, n, \lambda, \text{hk}, \text{ek})$ 23 parse $\text{hk} = ((N, g), 1, w_\lambda, L_{\text{exp}}, g_0, (g_{1,k})_k)$ 24 if $y \notin (\mathbb{Z}/N^2\mathbb{Z})^\times \times \mathbb{Z}/N\mathbb{Z}$ return $\perp$ 25 parse $y = (h, e)$ 26 $v \leftarrow e - \text{ELTDH.Dec}(\text{td}, h)$; $x \leftarrow \text{rep}(v)$ 27 if $x \geq m$ return $\perp$ 28 return $u \leftarrow x + 1$

FIGURE 5.1. Trapdoor ELF from ELTDH at residue arity one, for $m \geq 2$ and $r \in [m]$. Here $\ell(\lambda)$ is the bit length of the prime factors of N , and the hash key carries the exponent length L_{exp} and the bases $g_{1,k}$ for $k \in [0..w_\lambda - 1]$ of [FIGURE 4.1](#).

For the compact-description condition of [DEFINITION 5.1](#) and for the costs compared in [§1.4](#), we count the group elements in public descriptions, trapdoors, and outputs.

LEMMA 5.4 (Efficiency). Let $m \geq 2$ and $r \in [m]$. The algorithms in [FIGURE 5.1](#) run in time $\text{poly}(\log m)$. Excluding the modulus, the generator, and integer metadata, a public description contains $2w_\lambda + 2$ elements of $(\mathbb{Z}/N^2\mathbb{Z})^\times$, a trapdoor contains one exponent, and an output contains

one element of $(\mathbb{Z}/N^2\mathbb{Z})^\times$ and one element of $\mathbb{Z}/N\mathbb{Z}$. $\lrcorner$

Proof. The underlying ELTDH construction has residue arity one, so by [LEMMA 4.9](#) each of hk and ek contains $w_\lambda + 1$ group elements, the trapdoor contains one exponent, and an output contains one digest and one residue. Since $w_\lambda = \text{poly}(\lambda)$ and $\lambda = \text{poly}(n)$, all descriptions and algorithms have size and running time $\text{poly}(n) = \text{poly}(\log m)$. For $u \in [m]$, the bits of $u - 1$ at the positions $k \in [n..w_\lambda - 1]$ of its padded binary expansion are zero, so we compute each of the products in ELTDH.Hash and ELTDH.Enc with $\mathcal{O}(n)$ group multiplications, and ELTDH.Enc adds one evaluation of DDLog_N . $\blacksquare$

With the single trapdoor td , we invert f_{fk} on all of $[m]$ by the identity $e - \text{ELTDH.Dec}(\text{td}, h) = x$ of [COROLLARY 4.6](#) at coefficient vector (1).

THEOREM 5.5 (Perfect Inversion). Let $m \geq 2$ and $u \in [m]$. For every pair (fk, td) in the support of $\text{TELF.GenInj}(m)$ and for $y = \text{TELF.Eval}(\text{fk}, u)$, we have $\text{TELF.Inv}(\text{fk}, \text{td}, y) = u$. In particular, f_{fk} is injective. $\lrcorner$

Proof. We fix $u \in [m]$ and write $x = u - 1$ and $(h, e) = \text{TELF.Eval}(\text{fk}, u)$, so that $h \in (\mathbb{Z}/N^2\mathbb{Z})^\times$ and $e \in \mathbb{Z}/N\mathbb{Z}$. By [COROLLARY 4.6](#) at coefficient vector (1), we have $e - \text{ELTDH.Dec}(\text{td}, h) = x$ in $\mathbb{Z}/N\mathbb{Z}$. Since $0 \leq x < m \leq 2^n < N$, the least nonnegative representative of this residue is $x < m$, so TELF.Inv returns $x + 1 = u$. Since TELF.Inv is deterministic, $f_{\text{fk}}(u) = f_{\text{fk}}(u')$ for $u, u' \in [m]$ implies

$$u = \text{TELF.Inv}(\text{fk}, \text{td}, f_{\text{fk}}(u)) = \text{TELF.Inv}(\text{fk}, \text{td}, f_{\text{fk}}(u')) = u'.$$

$\blacksquare$

In [FIGURE 5.1](#), we recover the complete n -bit input from one decoded residue in $\mathbb{Z}/N\mathbb{Z}$, which is possible because $N > 2^n$. When every decoded value lies in a set of at most B elements, we need at least $n/\log_2 B$ of them, by a count of the possible tuples of decoded values.

LEMMA 5.6 (Lower Bound on the Number of Decoded Coordinates). Let $f: \{0, 1\}^n \rightarrow \mathcal{Y}$ be injective, let $B \geq 2$ and $k \geq 1$ be integers, and let $d_i: \mathcal{Y} \rightarrow \mathcal{V}_i$ be maps with $|\mathcal{V}_i| \leq B$ for $i \in [k]$ such that $x \mapsto (d_i(f(x)))_{i \in [k]}$ is injective on $\{0, 1\}^n$. Then $k \geq n/\log_2 B$. $\lrcorner$

Proof. The map $x \mapsto (d_i(f(x)))_{i \in [k]}$ is an injection of $\{0, 1\}^n$ into the product of the sets $\mathcal{V}_i$, so

$$2^n \leq \prod_{i=1}^k |\mathcal{V}_i| \leq B^k,$$

and we take logarithms to obtain $k \geq n/\log_2 B$. $\blacksquare$

By [LEMMA 5.6](#), $k = \Omega(n/\log n)$ when $B = \text{poly}(n)$, whereas in [FIGURE 5.1](#), we decode with $k = 1$ and $\mathcal{V}_1 = \mathbb{Z}/N\mathbb{Z}$.

In lossy generation with $r < m$, we use the predicate 0, so by [COROLLARY 4.6](#) the encoding coordinate of $f_{\text{fk}}(u)$ is a function of the digest $\Phi_{\text{hk}}(u - 1)$, which is the first coordinate of $f_{\text{fk}}(u)$. Hence, $|\text{Im}(f_{\text{fk}})| \leq |\text{Im}(\Phi_{\text{hk}})|$, and for $r = m$, the domain has $m = r$ points.

THEOREM 5.7 (Lossy Image Bound). Let $m \geq 2$ and $r \in [m]$. Every $\text{fk} \in \text{supp}(\text{TELF.GenLos}(m, r))$ satisfies $|\text{Im}(f_{\text{fk}})| \leq r$. $\lrcorner$

Proof. If $r = m$, then $|\text{Im}(f_{\text{fk}})| \leq |[m]| = r$. We suppose that $r < m$ and fix $\text{fk} = (m, n, \lambda, \text{hk}, \text{ek}) \in \text{supp}(\text{TELF.GenLos}(m, r))$. By the definition of TELF.GenLos , there is a trapdoor td with $(\text{ek}, \text{td}) \in \text{supp}(\text{ELTDH.Gen}(\text{hk}, (0)))$, and we define $J_{\text{td}}(h) := (h, \text{ELTDH.Dec}(\text{td}, h))$ for $h \in (\mathbb{Z}/N^2\mathbb{Z})^\times$. For every $u \in [m]$ and $(h, e) = f_{\text{fk}}(u)$, we have $h = \Phi_{\text{hk}}(u-1)$, and by [COROLLARY 4.6](#) at coefficient vector (0) , we have $e = \text{ELTDH.Dec}(\text{td}, h)$, so $f_{\text{fk}}(u) = J_{\text{td}}(\Phi_{\text{hk}}(u-1))$. Since $\{u-1 \mid u \in [m]\} = [0..m-1]$, we obtain

$$|\text{Im}(f_{\text{fk}})| = |J_{\text{td}}(\Phi_{\text{hk}}([0..m-1]))| \leq |\text{Im}(\Phi_{\text{hk}})| \leq r,$$

where the last inequality is [THEOREM 4.8](#). ■

Under the predicate 0, the first coordinate of $f_{\text{fk}}(u)$ is $g^{A(u)}$ for a bounded subset sum $A(u)$ of the exponents of the hash bases, and since $A(u)$ is smaller than the order of g , we count the image of f_{fk} exactly through the values of $A(u)$.

LEMMA 5.8 (Exact Lossy Image). Let $m \geq 2$, $r \in [m-1]$, and $\text{fk} = (m, n, \lambda, \text{hk}, \text{ek}) \in \text{supp}(\text{TELF.GenLos}(m, r))$, and write $\gamma := \lfloor (r-1)/w_\lambda \rfloor$. If $\text{hk} = \text{hk}_\perp$ or $\gamma = 0$, let $s_{1,k} := 0$ for $k \in [0..w_\lambda-1]$. Otherwise, let $s_{1,k} \in [\gamma]$ be the exponents with $g_{1,k} = g^{s_{1,k}}$ for the generator g and the bases $g_{1,k}$ in hk . For $u \in [m]$, define

$$A(u) := \sum_{j=1}^n s_{1,j-1} x_j(u).$$

Then $|\text{Im}(f_{\text{fk}})| = |\{A(u) \mid u \in [m]\}|$. ┘

Proof. We identify the digest of every input with $g^{A(u)}$ and then use that $A(u)$ is smaller than the order of g .

We fix a trapdoor td with $(\text{ek}, \text{td}) \in \text{supp}(\text{ELTDH.Gen}(\text{hk}, (0)))$. As in the proof of [THEOREM 5.7](#), every $u \in [m]$ satisfies $f_{\text{fk}}(u) = J_{\text{td}}(\Phi_{\text{hk}}(u-1))$ for $J_{\text{td}}(h) := (h, \text{ELTDH.Dec}(\text{td}, h))$, and J_{td} is injective because its first coordinate is h . Hence,

$$|\text{Im}(f_{\text{fk}})| = |\{\Phi_{\text{hk}}(u-1) \mid u \in [m]\}|.$$

If $\text{hk} = \text{hk}_\perp$ or $\gamma = 0$, then every base $g_{1,k}$ equals 1, so Φ_{hk} is constant, every $A(u)$ is zero, and both sides of the claim equal one. Otherwise, setup sampled a Damgård–Jurik group and the bounded bases. Since $r < m \leq 2^n \leq 2^\lambda$, the capped parameter $\min\{r, 2^\lambda\}$ of [FIGURE 4.1](#) equals r , so $\gamma \geq 1$ is the interval bound of setup and $g_{1,k} = g^{s_{1,k}}$ with $s_{1,k} \in [\gamma]$. By [LEMMA 2.10](#), the order $M = p'q'$ of g satisfies $M > 2^{2\ell(\lambda)-4} \geq 2^{2\lambda} \geq 2^{2n} > m$. Since $\gamma \leq r-1 < M$, each $s_{1,k}$ is the unique exponent in $[\gamma]$ with $g_{1,k} = g^{s_{1,k}}$. For $u \in [m]$, we have $u-1 < 2^n$, so the bit of $u-1$ at position k is $x_{k+1}(u)$ for $k \in [0..n-1]$ and zero for $k \in [n..w_\lambda-1]$, and at the all-zero tape,

$$\Phi_{\text{hk}}(u-1) = \prod_{j=1}^n g_{1,j-1}^{x_j(u)} = g^{A(u)}.$$

Since $0 \leq A(u) \leq n\gamma \leq w_\lambda\gamma \leq r-1 < M$, the map $a \mapsto g^a$ is injective on $[0..r-1]$. Hence, $\Phi_{\text{hk}}(u-1) = \Phi_{\text{hk}}(u'-1)$ holds exactly when $A(u) = A(u')$, and the two sets $\{\Phi_{\text{hk}}(u-1) \mid u \in [m]\}$ and $\{A(u) \mid u \in [m]\}$ have the same cardinality. ■

5.2 Mode Indistinguishability

By [THEOREMS 5.5](#) and [5.7](#), each generator has the property required of its mode, and it remains to show that the two distributions of public descriptions are indistinguishable. The two generators differ in the predicate of the encoding key and in the image parameter of the hash key, and we change these one at a time.

THEOREM 5.9 (Mode Indistinguishability). Suppose that $\text{DJGen}(\cdot)$ satisfies DCR against nonuniform PPT adversaries and tunable DSED. Then the generators in [FIGURE 5.1](#) satisfy the mode-indistinguishability condition of [DEFINITION 5.1](#). $\square$

Proof. We pass from injective to lossy generation through one intermediate hybrid. In the first transition, we replace the identity predicate by the zero predicate at the full image parameter R_λ , using function privacy, and in the second, we lower the image parameter from R_λ to r with the zero predicate, using ELTDH mode indistinguishability.

We fix a polynomial t and a positive polynomial δ , and we choose constants $C > 0$ and $D \in \mathbb{N}$ such that $t(z) \leq C(z+1)^D$ and $2\delta(z) \leq C(z+1)^D$ for every $z \in \mathbb{N}$. We also fix a polynomial p such that, for every λ and every hash key hk in the support of $\text{ELTDH.Setup}(1^\lambda, 1^1, \cdot)$, a circuit of size at most $p(\lambda)$ samples $(\text{ek}, \text{td}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (0))$ and assembles $(m, n, \lambda, \text{hk}, \text{ek})$ from hk , ek , and hardwired m and n with $\lambda_{\text{grp}}(n) = \lambda$. We set

$$\hat{t}(\lambda) := C(\lambda + 1)^D + p(\lambda), \quad \hat{\delta}(\lambda) := C(\lambda + 1)^D.$$

Since $\lambda_{\text{grp}}(n) \geq n$, every $n \in \mathbb{N}$ satisfies $t(n) + p(\lambda_{\text{grp}}(n)) \leq \hat{t}(\lambda_{\text{grp}}(n))$ and $\hat{\delta}(\lambda_{\text{grp}}(n)) \geq 2\delta(n)$.

For $m \geq 2$ and $r \in [m-1]$, with $n = \lceil \log_2 m \rceil$ and $\lambda = \lambda_{\text{grp}}(n)$, each of the three hybrids samples a hash key and an encoding key and outputs the public description $(m, n, \lambda, \text{hk}, \text{ek})$.

<u>Hyb₀</u>	* TELF.GenInj(m) *	<u>Hyb₁</u>	* zero predicate *	<u>Hyb₂</u>	* TELF.GenLos(m, r) *
00 $\text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^1, R_\lambda)$		03 $\text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^1, R_\lambda)$		06 $\text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^1, r)$	
01 $(\text{ek}, \text{td}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (1))$		04 $(\text{ek}, \text{td}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (0))$		07 $(\text{ek}, \text{td}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (0))$	
02 return $(m, n, \lambda, \text{hk}, \text{ek})$		05 return $(m, n, \lambda, \text{hk}, \text{ek})$		08 return $(m, n, \lambda, \text{hk}, \text{ek})$	

FIGURE 5.2. Hybrids for mode indistinguishability of the trapdoor ELF, at $m \geq 2$ and $r \in [m-1]$. The highlighted line of Hyb_i is the line in which it differs from Hyb_{i-1} .

By the definitions in [FIGURE 5.1](#), Hyb_0 in [FIGURE 5.2](#) is distributed as the public description output by $\text{TELF.GenInj}(m)$, and Hyb_2 is distributed as $\text{TELF.GenLos}(m, r)$. For a circuit $\mathcal{D}$ of size at most $t(n)$, we write p_i for the probability that $\mathcal{D}$ outputs 1 on a sample of Hyb_i , so that

$$|p_0 - p_2| \leq |p_0 - p_1| + |p_1 - p_2|.$$

At [Line 4](#), we replace the identity predicate by zero at the full image parameter. We take the selector in the function-privacy condition of [DEFINITION 4.1](#) at residue arity $d \equiv 1$ that outputs the coefficient vectors (1) and (0) and the empty state, and we recall that $R_{\text{full}}(\lambda, 1) = R_\lambda$. By [THEOREM 4.12](#) and [LEMMA 2.1](#) with size bound $\hat{t}$ and target polynomial $\hat{\delta}$, there is $\lambda_1 \in \mathbb{N}$ such that, for every $\lambda \geq \lambda_1$, every circuit of size at most $\hat{t}(\lambda)$ distinguishes the two function-privacy ensembles at λ with advantage at most $1/\hat{\delta}(\lambda)$. Our reduction $\mathcal{B}_1$ receives $(\text{hk}, \text{st}, \text{ek})$ from one of

these ensembles and outputs $\mathcal{D}(m, n, \lambda, \text{hk}, \text{ek})$. Its input to $\mathcal{D}$ is distributed as Hyb_0 when ek is generated for (1) and as Hyb_1 when ek is generated for (0), and its size is at most $t(n) + p(\lambda) \leq \hat{t}(\lambda)$. Hence, $|p_0 - p_1| \leq 1/\hat{\delta}(\lambda)$ whenever $\lambda \geq \lambda_1$.

At [Line 6](#), we lower the image parameter from R_λ to r with the zero predicate. We apply [THEOREM 4.13](#) at residue arity $d \equiv 1$ with size bound $\hat{t}$ and target polynomial $\hat{\delta}$, and we obtain a polynomial Q_{hash} and $\lambda_2 \in \mathbb{N}$ such that, for every $\lambda \geq \lambda_2$ and all $r_0, r_1 \in [Q_{\text{hash}}(\lambda)..R_\lambda]$, every circuit of size at most $\hat{t}(\lambda)$ distinguishes $\text{ELTDH.Setup}(1^\lambda, 1^1, r_0)$ from $\text{ELTDH.Setup}(1^\lambda, 1^1, r_1)$ with advantage at most $1/\hat{\delta}(\lambda)$. Our reduction $\mathcal{B}_2$ receives a hash key hk , samples $(\text{ek}, \text{td}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (0))$, and outputs $\mathcal{D}(m, n, \lambda, \text{hk}, \text{ek})$. Its input to $\mathcal{D}$ is distributed as Hyb_1 when hk is sampled at the image parameter R_λ and as Hyb_2 when hk is sampled at r , and its size is at most $\hat{t}(\lambda)$. Since λ_{grp} is polynomially bounded, there is a polynomial Q_{TELF} with $Q_{\text{TELF}}(n) \geq Q_{\text{hash}}(\lambda_{\text{grp}}(n))$ for every $n \in \mathbb{N}$. For $r \geq Q_{\text{TELF}}(n)$, we have $r \geq Q_{\text{hash}}(\lambda)$ and $r < m \leq 2^n \leq 2^\lambda < R_\lambda$, so $|p_1 - p_2| \leq 1/\hat{\delta}(\lambda)$ whenever $\lambda \geq \lambda_2$.

We take Q_{TELF} as the polynomial of the mode-indistinguishability condition and consider every $m \geq 2$ with $n = \lceil \log_2 m \rceil \geq \max\{\lambda_1, \lambda_2\}$, which holds for every sufficiently large m and implies $\lambda = \lambda_{\text{grp}}(n) \geq \max\{\lambda_1, \lambda_2\}$, and every $r \in [Q_{\text{TELF}}(n)..m]$. If $r = m$, then $\text{TELF.GenLos}(m, m)$ outputs the public description of a run of $\text{TELF.GenInj}(m)$, so the two distributions are identical. If $r < m$, we add the bounds of the two transitions and use $\hat{\delta}(\lambda) \geq 2\delta(n)$ to obtain

$$|p_0 - p_2| \leq \frac{2}{\hat{\delta}(\lambda)} \leq \frac{1}{\delta(n)}.$$

Hence, every circuit of size at most $t(n)$ distinguishes the two distributions of public descriptions with advantage at most $1/\delta(n)$, which is the condition of [DEFINITION 5.1](#). $\blacksquare$

We have proved the four conditions of [DEFINITION 5.1](#) for the construction in [LEMMA 5.4](#) and [THEOREMS 5.5, 5.7 and 5.9](#).

THEOREM 5.10 (Trapdoor Extremely Lossy Function). Suppose that $\text{DJGen}(\cdot)$ satisfies tunable DSED and DCR against nonuniform PPT adversaries. Then the algorithms in [FIGURE 5.1](#) form a trapdoor extremely lossy function, which has perfect inversion and satisfies the lossy image bound at every lossy public description in the support. $\lrcorner$

Proof. We verify the conditions of [DEFINITION 5.1](#) for the interval $[N_{\text{T}}(m)]$. Compact descriptions follow from [LEMMA 5.4](#) and $\log N_{\text{T}}(m) = 6\ell(\lambda_{\text{grp}}(\lceil \log_2 m \rceil)) = \text{poly}(\log m)$. The injective mode holds for every $(\text{fk}, \text{td}) \in \text{supp}(\text{TELF.GenInj}(m))$ by [THEOREM 5.5](#), and the lossy mode holds for every $\text{fk} \in \text{supp}(\text{TELF.GenLos}(m, r))$ by [THEOREM 5.7](#). Mode indistinguishability is [THEOREM 5.9](#). $\blacksquare$

5.3 Enumerating the Lossy Image

For an efficiently enumerable image in the sense of [\[Zha16\]](#), we compute the lossy image $\text{Im}(f_{\text{fk}})$ exactly from fk and r in time $\text{poly}(r, \log m)$.

Since every output of f_{fk} is a product of public bases selected by the bits of $u - 1$, we enumerate the image exactly from the public description, without the secret exponents. Working downwards from the most significant bit of $m - 1$, we keep the products for the processed prefixes of $u - 1$ that are equal to the corresponding prefix of $m - 1$ or smaller than it.

We state the procedure over an arbitrary finite abelian group $\mathbb{A}$ written multiplicatively, since its correctness depends only on the group operation. For $z, C \in \mathbb{A}$, we write $z \odot C$ for their product, and for $S \subseteq \mathbb{A}$, we write $S \odot C := \{z \odot C \mid z \in S\}$. For a finite set S of elements with canonical representations, $\text{Deduplicate}(S)$ is the list of the distinct elements of S in increasing order of the integers that their canonical representations encode, so that this list is uniquely determined by S .

```

Enumerate(fk, r)
00 parse fk = (m, n, λ, hk, ek)
01 parse hk = ((N, g), 1, w_λ, L_exp, g_0, (g_{1,k})_k)
02 parse ek = (N, 1, w_λ, L_exp, K_0, (K_{1,k})_k)
03 for j ∈ [n]
04   C_j ← (g_{1,j-1}, K_{1,j-1}); b_j ← x_j(m)
05   E ← {(1, 1)}; L ← ∅
06   for j = n, n-1, ..., 1
07     if b_j = 0
08       E' ← E; L' ← Deduplicate(L ∪ (L ⊙ C_j))
09     else:
10       E' ← E ⊙ C_j
11       L' ← Deduplicate(L ∪ (L ⊙ C_j) ∪ E)
12   (E, L) ← (E', L')
13   if |E| + |L| > r + 1 return ∅
14   return Deduplicate{(h, DDLog_N(c)) | (h, c) ∈ E ∪ L}

```

FIGURE 5.3. Exact enumeration of the image of a lossy public description of the trapdoor ELF, over $\mathbb{A} = (\mathbb{Z}/N^2\mathbb{Z})^\times \times (\mathbb{Z}/N^2\mathbb{Z})^\times$ with coordinatewise multiplication. Here $b_j = x_j(m)$ is the bit of $m-1$ at position j , E holds the product for the processed prefix of $u-1$ equal to the corresponding prefix of $m-1$, and L holds the products for smaller processed prefixes.

For a lossy public description, every product in $E \cup L$ is determined by a partial sum of the bounded exponents of the hash bases, so that $|E| + |L| \leq r + 1$ at every position. On every input, one iteration of FIGURE 5.3 forms at most $2(r + 1)$ products, because we return $\emptyset$ as soon as $|E| + |L| > r + 1$.

THEOREM 5.11 (Strong Enumeration for the Trapdoor ELF). There is a deterministic algorithm `Enumerate` such that, for every $m \geq 2$, $r \in [m]$, and $\text{fk} \in \text{supp}(\text{TELF.GenLos}(m, r))$, `Enumerate(fk, r)` returns $\text{Im}(f_{\text{fk}})$ in time $\text{poly}(r, \log m)$. In particular, the construction in FIGURE 5.1 has a strongly efficiently enumerable image in the sense of DEFINITION 5.3. $\square$

Proof. We prove an invariant of the state sets of the algorithm `Enumerate` in FIGURE 5.3 by induction over the processed positions, and we bound their sizes through the bounded subset sums of LEMMA 5.8.

We write $\mathbb{A} := (\mathbb{Z}/N^2\mathbb{Z})^\times \times (\mathbb{Z}/N^2\mathbb{Z})^\times$ and take $C_j \in \mathbb{A}$ and $b_j = x_j(m)$ as in FIGURE 5.3. For $u \in [2^n]$ and $j \in [n+1]$, we write $\pi_j(u) := \lfloor (u-1)/2^{j-1} \rfloor$ for the prefix of $u-1$ formed by the bits at positions $n, n-1, \dots, j$, and

$$P_j(u) := \prod_{i=j}^n C_i^{x_i(u)} \in \mathbb{A}$$

for the corresponding product, so that $\pi_{n+1}(u) = 0$ and $P_{n+1}(u) = (1, 1)$. We show that after the

iteration for j , and before the first iteration for $j = n + 1$, the state sets satisfy

$$E = \{P_j(m)\}, \quad L = \{P_j(u) \mid u \in [2^n], \pi_j(u) < \pi_j(m)\}.$$

For $j = n + 1$, we have $E = \{(1, 1)\}$ and $L = \emptyset$. We suppose that the invariant holds after the iteration for $j + 1$. Every $u \in [2^n]$ satisfies $\pi_j(u) = 2\pi_{j+1}(u) + x_j(u)$ and $P_j(u) = P_{j+1}(u) \odot C_j^{x_j(u)}$, and $P_{j+1}(u)$ depends only on $\pi_{j+1}(u)$. Hence, $\pi_j(u) = \pi_j(m)$ holds exactly when $\pi_{j+1}(u) = \pi_{j+1}(m)$ and $x_j(u) = b_j$, and $\pi_j(u) < \pi_j(m)$ holds exactly when $\pi_{j+1}(u) < \pi_{j+1}(m)$, or when $\pi_{j+1}(u) = \pi_{j+1}(m)$ and $x_j(u) < b_j$. Every value of π_{j+1} occurs with both values of the bit at position j , so the prefixes of the first kind contribute the products $L \cup (L \odot C_j)$. If $b_j = 0$, the prefixes of the second kind do not occur, and the new state is $E' = E$ and $L' = L \cup (L \odot C_j)$. If $b_j = 1$, the equal prefix has the product $P_{j+1}(m) \odot C_j$, and the only prefix of the second kind has the product $P_{j+1}(m)$, so the new state is $E' = E \odot C_j$ and $L' = L \cup (L \odot C_j) \cup E$. These are the two update rules of [FIGURE 5.3](#), and **Deduplicate** preserves the set of elements, so the invariant holds after the iteration for j .

After the iteration for $j = 1$, we have $\pi_1(u) = u - 1$, so $E \cup L = \{P_1(u) \mid u \in [m]\}$. For $u \in [m]$, the bits of $u - 1$ at the positions $k \in [n..w_\lambda - 1]$ are zero, so by the definitions of **ELTDH.Hash** and **ELTDH.Enc** in [FIGURE 4.1](#) at the all-zero tape, $P_1(u) = (h, c)$ for the digest $h = \Phi_{\text{hk}}(u - 1)$ and the product c formed in **ELTDH.Enc**(ek, $(u - 1); 0$), which returns $\text{DDLog}_N(c)$. Hence, if the guard condition $|E| + |L| > r + 1$ is false at every position, the last line of [FIGURE 5.3](#) returns the list of the distinct elements of $\{f_{\text{fk}}(u) \mid u \in [m]\} = \text{Im}(f_{\text{fk}})$.

It remains to show that $|E| + |L| \leq r + 1$ after every iteration. If $r = m$, then $|E| = 1$ and $|L| \leq \pi_j(m) \leq m - 1$. If $r < m$, we write $\gamma := \lfloor (r - 1)/w_\lambda \rfloor$ and take the exponents $s_{1,k}$ of [LEMMA 5.8](#). If $\text{hk} = \text{hk}_\perp$ or $\gamma = 0$, every base $g_{1,k}$ equals 1, and the entries of the encoding key for the zero predicate are $K_{1,k} = g_{1,k}^\sigma = 1$, so every product equals $(1, 1)$ and $|E| + |L| \leq 2 \leq r + 1$. Otherwise, $g_{1,k} = g^{s_{1,k}}$ and $K_{1,k} = g^{\sigma s_{1,k}}$, so $P_j(u) = (g^a, g^{\sigma a})$ for the partial sum

$$a := \sum_{i=j}^n s_{1,i-1} x_i(u) \in [0..n\gamma].$$

Hence, $E \cup L$ has at most $n\gamma + 1 \leq w_\lambda \gamma + 1 \leq r$ elements, and $|E| + |L| \leq |E \cup L| + 1 \leq r + 1$.

Therefore, for every supported public description, the guard condition is false at every position, and **Enumerate**(fk, r) returns $\text{Im}(f_{\text{fk}})$. On every input, $|E| + |L| \leq r + 1$ holds before each iteration, so each iteration forms at most $2(r + 1)$ products in $\mathbb{A}$ and deduplicates at most $2(r + 1)$ elements. Over the n iterations, the algorithm uses $\mathcal{O}(nr)$ multiplications in $\mathbb{A}$, sorts lists of at most $2(r + 1)$ canonical representations, and ends with at most $r + 1$ evaluations of DDLog_N , each operation in time $\text{poly}(\lambda) = \text{poly}(\log m)$. Hence, it runs in time $\text{poly}(r, \log m)$. ■

Uniform sampling from the lossy image, as considered in [\[Zha16\]](#), follows from exact enumeration, since a uniform element of a list without repetitions is uniform on the set of its elements for every distribution of preimage sizes.

COROLLARY 5.12 (Uniform Image Sampling). Let $m \geq 2$ and $r \in [m]$. For every $\text{fk} \in \text{supp}(\text{TELF.GenLos}(m, r))$, we can sample uniformly from $\text{Im}(f_{\text{fk}})$ in time $\text{poly}(r, \log m)$. ▮

Proof. We run the algorithm of [THEOREM 5.11](#) on (fk, r) and return a uniformly random element of the returned list. Since this list consists of the distinct elements of $\text{Im}(f_{\text{fk}})$, every element of $\text{Im}(f_{\text{fk}})$ is returned with probability $1/|\text{Im}(f_{\text{fk}})|$, and the running time is $\text{poly}(r, \log m)$. ■

[ABG⁺26] constructs anamorphic-resistant encryption, in which every anamorphic triplet that is correct on average and semi-adaptively secure has an anamorphic message space of polynomial size, in two ways. The first construction combines a perfectly correct public-key encryption scheme, a universal hash family, fully unique non-interactive zero-knowledge arguments, which are known from indistinguishability obfuscation, and an ELF with inputs of length 3λ and outputs of length polynomial in λ , and it uses the ELF only through its two modes and mode indistinguishability, where injective generation must always return an injective function and the lossy image must be smaller than the image bound. The second construction avoids indistinguishability obfuscation. It combines public-key encryption and a lossy trapdoor function with a variant of the trapdoor ELF of [Zha19], and [ABG⁺26] proves its anamorphic resistance by a direct reduction to exponential matrix DDH for that variant. Without its trapdoor, our trapdoor ELF is an ELF in the sense of the first construction.

COROLLARY 5.13 (Always-Injective ELF). Suppose that $\text{DJGen}(\cdot)$ satisfies DCR against nonuniform PPT adversaries and tunable DSEDL. For $m \geq 2$ and $r \in [2..m]$, let $\text{ELF.Gen}^*(m, r)$ return the public description of $\text{TELF.GenInj}(m)$ if $r = m$, and the public description of $\text{TELF.GenLos}(m, r - 1)$ if $r < m$, and let TELF.Eval evaluate it. Then the following hold.

- **Injective Mode.** Every output of $\text{ELF.Gen}^*(m, m)$ describes an injective function.
- **Lossy Mode.** For every $r \in [2..m - 1]$, every output fk of $\text{ELF.Gen}^*(m, r)$ satisfies $|\text{Im}(f_{\text{fk}})| < r$.
- **Mode Indistinguishability.** With evaluation by TELF.Eval , the generator ELF.Gen^* satisfies the mode-indistinguishability condition of [DEFINITION 2.6](#). J

Proof. The injective mode holds by [THEOREM 5.5](#), and the lossy mode holds by [THEOREM 5.7](#) at the image bound $r - 1 \in [m]$. For mode indistinguishability, we fix t and δ and take the polynomial Q_{TELF} that [THEOREM 5.9](#) provides for them, and we set $Q := Q_{\text{TELF}} + 1$. For $r = m$, the two distributions are equal. For $r \in [Q(n_m)..m - 1]$, the output of $\text{ELF.Gen}^*(m, r)$ is that of $\text{TELF.GenLos}(m, r - 1)$ with $r - 1 \geq Q_{\text{TELF}}(n_m)$, so [THEOREM 5.9](#) bounds the advantage of every circuit of size $t(n_m)$ by $1/\delta(n_m)$ for every sufficiently large m . ■

By [COROLLARY 5.13](#) at $m = 2^{3\lambda}$, where $\log N_{\text{out}}(m)$ is polynomial in λ , the first construction of [ABG⁺26] is anamorphic-resistant and IND-CPA-secure with our ELF in place of an ELF from exponential DDH, under tunable DSEDL and DCR together with the other building blocks of that construction. The same holds for the result of [ABG⁺26] that the first construction, applied to a public-key encryption scheme without anamorphic bits in the adaptive setting, returns such a scheme.

REMARK 5.14 (Partial Trapdoors). The direct reduction for the second construction publishes one layer as $[\mathbf{A}]$ with $\mathbf{A} = \mathbf{BC}$ for a uniform matrix $\mathbf{B}$, which serves as a partial trapdoor: with $\mathbf{B}$, the reduction decides membership in $[\text{Im } \mathbf{B}]$, a superset of the image of that layer in both modes. Soundness of this test in the lossy mode uses the fact that the group of the layer has polynomial order, so every element that an adversary computes from $[\mathbf{BC}]$ with $\mathbf{C}$ of rank one lies in a set of polynomial size. In our construction, the lossy image is small because the exponents are small in $\mathbb{G}_N$, whose order is exponential. From the bases $g_{1,k}$ and the keys $K_{1,k}$, every party computes $h = \prod_k g_{1,k}^{y_k}$ and its consistent share $\text{DDLog}_N(\prod_k K_{1,k}^{y_k})$ for arbitrary integers y_k . A public test that accepts these values for $y \in \{0, 1\}^w$ and rejects them for large y_k separates g^z for z in a short interval from g^z for a large z , which tunable DSEDL assumes hard for observers of bounded size, and the trapdoor σ ,

which rejects them in the injective mode, determines the encoded predicate. Our proof of mode indistinguishability changes the predicate under DCR, where σ is unknown to the reduction, and the bases under tunable DSEDL, where the size of the reduction is bounded independently of γ , so the reduction for the second construction does not carry over, and whether our construction admits a partial trapdoor is open. $\lrcorner$

5.4 Linear-Size Plain ELF

With the same small-exponent subset products and without a trapdoor, we obtain an ELF whose public description consists of n group elements in addition to the group parameters. Its mode indistinguishability depends only on tunable DSEDL, whereas in the trapdoor construction we use DCR to hide the predicate in the encoding key, and its injective and lossy modes hold unconditionally.

In the construction and its proofs, we use only uniform sampling in the group, exponentiation of the generator, and a lower bound on the group order, so we state the construction over an arbitrary cyclic group sampler CGen in the sense of [DEFINITION 2.7](#). We fix such a sampler and a positive rational constant c , and for $m \geq 2$, we set $n := \lceil \log_2 m \rceil$ and, in this subsection, $\lambda := \lceil cn \rceil$ in place of $\lambda_{\text{grp}}(n)$. The first two operations are part of [DEFINITION 2.7](#), and the lower bound on the order determines the admissible range of c , because we use it to show that, except with negligible probability, the uniform exponents of the injective bases satisfy only the trivial relation with coefficients in $\{-1, 0, 1\}$. We count group elements, and each of them has a representation of at most $\ell_{\text{grp}}(\lambda)$ bits, so the public description is linear in n as a count of elements and superlinear in bits.

To identify outputs with elements of an interval as in [DEFINITION 2.6](#), we encode a group element by the length of its canonical representation, written in exactly $\lceil \log_2(\ell_{\text{grp}}(\lambda) + 1) \rceil$ bits, followed by the representation padded with zeros to length $\ell_{\text{grp}}(\lambda)$. This encoding is injective, has the fixed length $L(\lambda) := \ell_{\text{grp}}(\lambda) + \lceil \log_2(\ell_{\text{grp}}(\lambda) + 1) \rceil$, and is computable and invertible in polynomial time. We interpret the resulting $L(\lambda)$ -bit string as its nonnegative binary value plus one, which is an element of $[2^{L(\lambda)}]$, and we take $N_{\text{out}}(m) := 2^{L(\lceil c \lceil \log_2 m \rceil \rceil)}$, so that $\log N_{\text{out}}(m) = \text{poly}(\log m)$.

We replace every failure of CGen or of the base sampler by the fixed description $f_{\perp} := (1, m, n, \lambda, \perp, (1)_{j \in [n]})$. Since its first entry is 1, $\text{ELF.Eval}(f_{\perp}, u) = 1 \in [N_{\text{out}}(m)]$ for every $u \in [m]$, and $f_{\perp}$ can lie in the support of $\text{ELF.Gen}(m, r)$. Both generators output $f_{\perp}$ when CGen fails, and $\text{ELF.Gen}(m, m)$ also outputs $f_{\perp}$ when one of its n base samples fails. The function $\text{ELF.Eval}(f_{\perp}, \cdot)$ satisfies every lossy image bound, and since $f_{\perp}$ is output with negligible probability, its probability is a negligible term of the injectivity error, which [DEFINITION 2.6](#) requires to be negligible.

For mode indistinguishability, we identify each generator with one distribution of [LEMMA 3.9](#), since a public description differs from a vector of bounded or uniform powers only in deterministic entries.

THEOREM 5.15 (Generators as Vector DSEDL Samples). Let $m \geq 2$ and $r \in [m - 1]$, write $n = \lceil \log_2 m \rceil$, $\lambda = \lceil cn \rceil$, and $\gamma = \lfloor (r - 1)/n \rfloor$, and suppose that $1 \leq \gamma < \text{ord}(\text{pp})$ for every $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$. Let $\mathfrak{D}_{0, \lambda, \gamma}^{\text{vDSEDL}}$ and $\mathfrak{D}_{1, \lambda, \gamma}^{\text{vDSEDL}}$ be the distributions of [LEMMA 3.9](#) for the constant coordinate count n , and define the deterministic polynomial-time map ψ by $\psi(\perp) := f_{\perp}$ and

$$\psi(1^\lambda, \text{pp}, \gamma, (G_j)_{j \in [n]}) := (0, m, n, \lambda, \text{pp}, (G_j)_{j \in [n]}).$$

<pre> ELF.Gen(m, r) 00 assert $m \geq 2 \wedge r \in [m]$ 01 $n \leftarrow \lceil \log_2 m \rceil$; $\lambda \leftarrow \lceil cn \rceil$ 02 $\mathbf{pp} \xleftarrow{\\$} \mathbf{CGen}(1^\lambda)$ 03 if $\mathbf{pp} = \perp$ return $f \leftarrow f_\perp$ 04 $g \leftarrow g(\mathbf{pp})$ 05 if $r = m$ 06 for $j \in [n]$ 07 $g_j \xleftarrow{\\$} \mathbb{G}(\mathbf{pp})$ 08 if $\exists j: g_j = \perp$ return $f \leftarrow f_\perp$ 09 else: 10 $\gamma \leftarrow \lfloor (r-1)/n \rfloor$ 11 for $j \in [n]$ 12 if $\gamma = 0$ $g_j \leftarrow 1$ 13 else: 14 $s_j \xleftarrow{\\$} [\gamma]$ 15 $g_j \leftarrow g^{s_j}$ 16 $f \leftarrow (0, m, n, \lambda, \mathbf{pp}, (g_j)_{j \in [n]})$ 17 return f </pre>	<pre> ELF.Eval(f, u) 18 parse $f = (\chi, m, n, \lambda, \mathbf{pp}, (g_j)_{j \in [n]})$ 19 assert $u \in [m]$ 20 if $\chi = 1$ return 1 21 return $\prod_{j=1}^n g_j^{x_j(u)}$ </pre>
--	--

FIGURE 5.4. Linear-size ELF over a cyclic group sampler $\mathbf{CGen}$, for $m \geq 2$ and $r \in [m]$, with $n = \lceil \log_2 m \rceil$ and $\lambda = \lceil cn \rceil$. The first entry χ of a description equals 1 exactly for $f_\perp$.

Then

$$\begin{aligned} \left\{ f \mid f \xleftarrow{\$} \mathbf{ELF.Gen}(m, m) \right\} &= \left\{ \psi(X) \mid X \xleftarrow{\$} \mathfrak{D}_{1, \lambda, \gamma}^{\text{vDSEDL}} \right\}, \\ \left\{ f \mid f \xleftarrow{\$} \mathbf{ELF.Gen}(m, r) \right\} &= \left\{ \psi(X) \mid X \xleftarrow{\$} \mathfrak{D}_{0, \lambda, \gamma}^{\text{vDSEDL}} \right\}, \end{aligned}$$

and the polynomial-time map $(0, m, n, \lambda, \mathbf{pp}, (G_j)_{j \in [n]}) \mapsto (1^\lambda, \mathbf{pp}, \gamma, (G_j)_{j \in [n]})$ inverts ψ on the supports of both vector distributions without $\perp$. $\lrcorner$

Proof. We compare the samplings of the generators and of the vector distributions step by step.

Both generators and both vector distributions first sample $\mathbf{pp} \xleftarrow{\$} \mathbf{CGen}(1^\lambda)$, and on $\mathbf{pp} = \perp$ the generators output $f_\perp = \psi(\perp)$, while the vector distributions output $\perp$. For $\mathbf{pp} \neq \perp$, $\mathbf{ELF.Gen}(m, m)$ samples $g_1, \dots, g_n \xleftarrow{\$} \mathbb{G}(\mathbf{pp})$ and outputs $f_\perp$ if one of these samples fails and $(0, m, n, \lambda, \mathbf{pp}, (g_j)_{j \in [n]})$ otherwise, and $\mathfrak{D}_{1, \lambda, \gamma}^{\text{vDSEDL}}$ samples the same n elements and outputs $\perp$ if one of them fails and $(1^\lambda, \mathbf{pp}, \gamma, (g_j)_{j \in [n]})$ otherwise, which implies the first identity. Since $r < m$ and $\lfloor (r-1)/n \rfloor = \gamma \geq 1$, $\mathbf{ELF.Gen}(m, r)$ samples $(s_j)_{j \in [n]} \xleftarrow{\$} [\gamma]^n$ and outputs $(0, m, n, \lambda, \mathbf{pp}, (g^{s_j})_{j \in [n]})$, and $\mathfrak{D}_{0, \lambda, \gamma}^{\text{vDSEDL}}$ samples the same exponents and outputs $(1^\lambda, \mathbf{pp}, \gamma, (g^{s_j})_{j \in [n]})$, which implies the second identity. Outside $\perp$, the map ψ replaces the entries 1^λ and γ by $(0, m, n)$, which depend only on m , and keeps λ , so the map in the statement inverts ψ . $\blacksquare$

From [THEOREM 5.15](#) and [LEMMA 3.9](#), a union bound over relations among uniform exponents, and a count of bounded subset sums, we obtain an ELF for every rational $c > \log_2 3$, where we need $c > \log_2 3$ to compare the $3^n - 1$ nonzero relations with the group order $\text{ord}(\mathbf{pp}) \geq 2^{\lceil cn \rceil}$.

THEOREM 5.16 (Linear-Size ELF). Let $c > \log_2 3$, and suppose that $\mathbf{CGen}(\cdot)$ satisfies tunable DSEDL. Then the construction in [FIGURE 5.4](#) is an extremely lossy function. Excluding the group parameters $\mathbf{pp}$, its public description contains n group elements, and evaluation returns one group element using $\mathcal{O}(n)$ group operations. $\lrcorner$

Proof. We rule out nonzero relations with coefficients in $\{-1, 0, 1\}$ among the uniform exponents with a union bound, we count bounded subset sums for the lossy mode, and for mode indistinguishability we combine [THEOREM 5.15](#) with [LEMMA 3.9](#).

EFFICIENCY AND OUTPUT ENCODING. The parameters satisfy $n = \Theta(\log m)$ and $\lambda = \Theta(n)$. In generation, we sample n group elements or compute g^{s_j} for n exponents $s_j \leq \gamma < m$, and in evaluation, we multiply the bases g_j with $x_j(u) = 1$, using at most $n - 1$ group multiplications. By [DEFINITION 2.7](#), these operations take time $\text{poly}(\lambda) = \text{poly}(\log m)$, and every group element has a representation of at most $\ell_{\text{grp}}(\lambda)$ bits, so under the padded encoding, the output lies in $[2^{L(\lambda)}] = [N_{\text{out}}(m)]$ with $\log N_{\text{out}}(m) = L(\lambda) = \text{poly}(\log m)$.

INJECTIVE MODE. We fix $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$ and write $M := \text{ord}(\text{pp})$, so that $g = g(\text{pp})$ generates the cyclic group $\mathbb{G}(\text{pp})$ of order M . Conditioned on the event that all n base samples succeed, the bases are independent and uniform in $\mathbb{G}(\text{pp})$ by [DEFINITION 2.7](#), so $g_j = g^{a_j}$ for independent uniform $a_j \in \mathbb{Z}/M\mathbb{Z}$. We write E_{inj} for the event that two distinct inputs $u, u' \in [m]$ have the same output. From such a collision, we obtain distinct $x(u), x(u') \in \{0, 1\}^n$ and a nonzero vector $v := x(u) - x(u') \in \{-1, 0, 1\}^n$ with

$$\sum_{j=1}^n v_j a_j = 0$$

in $\mathbb{Z}/M\mathbb{Z}$. We fix a nonzero $v \in \{-1, 0, 1\}^n$ and choose $j^* \in [n]$ with $v_{j^*} \in \{-1, 1\}$. Since v_{j^*} is a unit modulo M , after we condition on $(a_j)_{j \neq j^*}$, exactly one value of a_{j^*} satisfies this equation, namely

$$a_{j^*} = -v_{j^*}^{-1} \sum_{j \neq j^*} v_j a_j.$$

Since a_{j^*} is uniform in $\mathbb{Z}/M\mathbb{Z}$ and independent of $(a_j)_{j \neq j^*}$, the equation holds with probability $1/M$. We take a union bound over the $3^n - 1$ nonzero vectors v , use $M \geq 2^\lambda \geq 2^{cn}$ from [DEFINITION 2.7](#), and obtain

$$\begin{aligned} \Pr \left[E_{\text{inj}} \mid (a_j)_{j \in [n]} \stackrel{\$}{\leftarrow} (\mathbb{Z}/M\mathbb{Z})^n \right] &\leq \sum_{v \in \{-1, 0, 1\}^n \setminus \{0\}} \frac{1}{M} = \frac{3^n - 1}{M} \\ &< \frac{3^n}{2^{cn}} = 2^{-(c - \log_2 3)n}. \end{aligned}$$

This bound holds for every pp , and it is negligible in n because $c > \log_2 3$. Generation outputs $f_\perp$ with probability at most

$$\Pr \left[\text{pp} = \perp \mid \text{pp} \stackrel{\$}{\leftarrow} \text{CGen}(1^\lambda) \right] + n\nu_{\mathbb{G}}(\lambda),$$

which is negligible in n because $cn \leq \lambda \leq cn + 1$. Hence, f is injective except with probability at most the sum of these two negligible terms.

LOSSY MODE. If $r = m$, the domain has r points, and $|\text{Im}(f)| \leq r$. We suppose that $r < m$. The fallback description $f_\perp$ describes a constant function, and on every other description in the support, $\gamma = \lfloor (r - 1)/n \rfloor$. If $\gamma = 0$, every base is 1, and the function is constant. If $\gamma \geq 1$, every output equals $g^{A(u)}$ for

$$A(u) := \sum_{j=1}^n s_j x_j(u) \in [0..n\gamma].$$

In all cases, $|\text{Im}(f)| \leq n\gamma + 1 = n \lfloor (r - 1)/n \rfloor + 1 \leq r$.

MODE INDISTINGUISHABILITY. We fix a polynomial t and a positive polynomial δ , and we choose

constants $C > 0$ and $D \in \mathbb{N}$ such that $t(z) \leq C(z+1)^D$ and $2\delta(z) \leq C(z+1)^D$ for every $z \in \mathbb{N}$. We write $\kappa(\lambda) := \lceil \lambda/c \rceil$, and since $\lambda = \lceil cn \rceil \geq cn$, we have $n \leq \lambda/c \leq \kappa(\lambda)$. We fix a polynomial p such that, for every λ , a circuit of size at most $p(\lambda)$ maps a challenge of [LEMMA 3.9](#) with $\kappa(\lambda)$ coordinates to the image under ψ of its first n coordinates, for hardwired m and n , and we set

$$\hat{t}(\lambda) := C(\lambda/c + 1)^D + p(\lambda), \quad \hat{\delta}(\lambda) := C(\lambda/c + 1)^D.$$

Both are polynomials because c is a positive rational and $D \in \mathbb{N}$, and from $n \leq \lambda/c$, we obtain $t(n) + p(\lambda) \leq \hat{t}(\lambda)$ and $\hat{\delta}(\lambda) \geq 2\delta(n)$.

We apply [LEMMA 3.9](#) with coordinate count κ , size bound $\hat{t}$, and target polynomial $\hat{\delta}$, and we obtain a polynomial Q and $\lambda_0 \in \mathbb{N}$ such that, for every $\lambda \geq \lambda_0$ and every integer γ with $Q(\lambda) \leq \gamma < \text{ord}(\text{pp})$ for all $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$, every circuit of size at most $\hat{t}(\lambda)$ distinguishes $\mathfrak{D}_{0,\lambda,\gamma}^{\text{vDSEDL}}$ from $\mathfrak{D}_{1,\lambda,\gamma}^{\text{vDSEDL}}$ with $\kappa(\lambda)$ coordinates with advantage at most $1/\hat{\delta}(\lambda)$. We choose an integer-valued polynomial $\bar{Q}$ with $\bar{Q}(\lambda) \geq \max\{1, \lceil Q(\lambda) \rceil\}$ for every λ , and an integer-valued polynomial Q_{plain} with $Q_{\text{plain}}(n) \geq n\bar{Q}(\lceil cn \rceil) + 1$ for every $n \in \mathbb{N}$. Since κ is polynomially bounded and $\nu_{\mathbb{G}}$ is negligible, $\varepsilon_{\text{unused}}(\lambda) := \kappa(\lambda)\nu_{\mathbb{G}}(\lambda)$ is negligible, so there is $\lambda_1 \in \mathbb{N}$ with $\varepsilon_{\text{unused}}(\lambda) \leq 1/\hat{\delta}(\lambda)$ for every $\lambda \geq \lambda_1$.

We fix $m \geq 2$ with $\lambda = \lceil cn \rceil \geq \max\{\lambda_0, \lambda_1\}$, which holds for every sufficiently large m , and $r \in [Q_{\text{plain}}(n)..m]$ and a circuit $\mathcal{D}$ of size at most $t(n)$, and we write p_0 and p_1 for the probabilities that $\mathcal{D}$ outputs 1 on a sample of $\text{ELF.Gen}(m, m)$ and of $\text{ELF.Gen}(m, r)$, respectively. If $r = m$, then $p_0 = p_1$. We suppose that $r < m$ and set $\gamma := \lfloor (r-1)/n \rfloor$. From $r \geq Q_{\text{plain}}(n) \geq n\bar{Q}(\lambda) + 1$, we obtain $\gamma \geq \bar{Q}(\lambda) \geq \max\{1, Q(\lambda)\}$. Since $c > \log_2 3 > 1$, [DEFINITION 2.7](#) implies $\gamma < r < m \leq 2^n < 2^{cn} \leq 2^\lambda \leq \text{ord}(\text{pp})$ for every $\text{pp} \in \text{supp}(\text{CGen}(1^\lambda))$. Hence, the hypotheses of [THEOREM 5.15](#) hold.

Hyb ₀	* ELF.Gen(m, m) *	Hyb ₁	* ELF.Gen(m, r) *
00 $\text{pp} \xleftarrow{\$} \text{CGen}(1^\lambda)$		06 $\text{pp} \xleftarrow{\$} \text{CGen}(1^\lambda)$	
01 if $\text{pp} = \perp$ return $f_\perp$		07 if $\text{pp} = \perp$ return $f_\perp$	
02 $g \leftarrow g(\text{pp})$		08 $g \leftarrow g(\text{pp})$	
03 $(g_j)_{j \in [n]} \xleftarrow{\$} \mathbb{G}(\text{pp})^n$		09 $(s_j)_{j \in [n]} \xleftarrow{\$} [\gamma]^n; g_j \leftarrow g^{s_j}$	
04 if $\exists j: g_j = \perp$ return $f_\perp$		10 return $(0, m, n, \lambda, \text{pp}, (g_j)_{j \in [n]})$	
05 return $(0, m, n, \lambda, \text{pp}, (g_j)_{j \in [n]})$			

FIGURE 5.5. Hybrids for mode indistinguishability of the plain ELF, at $r \in [m-1]$ with $\gamma = \lfloor (r-1)/n \rfloor \geq 1$. The highlighted line of Hyb₁ is the line in which it differs from Hyb₀.

In [FIGURE 5.5](#), Hyb₀ and Hyb₁ are $\text{ELF.Gen}(m, m)$ and $\text{ELF.Gen}(m, r)$ for this $\gamma \geq 1$, and at [Line 9](#), we replace the uniform bases by bounded powers. Our reduction $\mathcal{B}$ receives a challenge X from $\mathfrak{D}_{b,\lambda,\gamma}^{\text{vDSEDL}}$ with $\kappa(\lambda)$ coordinates. If $X = \perp$, it outputs $\mathcal{D}(f_\perp)$, and otherwise, for $X = (1^\lambda, \text{pp}, \gamma, (G_j)_{j \in [\kappa(\lambda)]})$, it outputs $\mathcal{D}(0, m, n, \lambda, \text{pp}, (G_j)_{j \in [n]})$. Its size is at most $t(n) + p(\lambda) \leq \hat{t}(\lambda)$, and we write q_b for the probability that it outputs 1. For $b = 0$, the first n coordinates of X , with $\perp$ kept as $\perp$, are distributed as $\mathfrak{D}_{0,\lambda,\gamma}^{\text{vDSEDL}}$ with n coordinates, because a bounded challenge equals $\perp$ exactly when CGen fails, so $q_0 = p_1$ by [THEOREM 5.15](#). For $b = 1$, the first n coordinates of X are distributed as $\mathfrak{D}_{1,\lambda,\gamma}^{\text{vDSEDL}}$ with n coordinates except on the event that CGen and the first n coordinate samples succeed while one of the last $\kappa(\lambda) - n$ coordinate samples fails, whose probability is at most

$\varepsilon_{\text{unused}}(\lambda)$, so $|q_1 - p_0| \leq \varepsilon_{\text{unused}}(\lambda)$ by [THEOREM 5.15](#). By the choice of Q , λ_0 , and λ_1 , we obtain

$$|p_0 - p_1| \leq |q_0 - q_1| + \varepsilon_{\text{unused}}(\lambda) \leq \frac{2}{\delta(\lambda)} \leq \frac{1}{\delta(n)}.$$

Hence, every circuit of size at most $t(n)$ distinguishes $\text{ELF.Gen}(m, m)$ from $\text{ELF.Gen}(m, r)$ with advantage at most $1/\delta(n)$, and with the efficiency and the injective and lossy modes established in this proof, the construction satisfies every condition of [DEFINITION 2.6](#). $\blacksquare$

In the proof of [THEOREM 5.16](#), we use $c > \log_2 3$ only through the lower bound $\text{ord}(\text{pp}) \geq 2^\lambda$ of [DEFINITION 2.7](#), in the union bound of the injective mode and in the inequality $2^n < \text{ord}(\text{pp})$ of the mode transition. For $\text{CGen} = \text{DJGen}$, the order $M = p'q'$ exceeds $2^{2\ell(\lambda)-4} \geq 2^{2\lambda}$ by [LEMMA 2.10](#), so a smaller constant suffices, and since $\lambda = \lceil cn \rceil$, group elements are shorter for smaller c at the same m .

COROLLARY 5.17 (Linear-Size ELF over Damgård–Jurik Groups). Let $c > (\log_2 3)/2$, and suppose that $\text{DJGen}(\cdot)$ satisfies tunable DSEDL. Then the construction in [FIGURE 5.4](#) at $\text{CGen} = \text{DJGen}$ is an extremely lossy function, with the description size, output size, and evaluation cost of [THEOREM 5.16](#). $\lrcorner$

Proof. We repeat the proof of [THEOREM 5.16](#) with $\text{CGen} = \text{DJGen}$, which is a cyclic group sampler by [LEMMA 2.10](#), and we replace the lower bound $\text{ord}(\text{pp}) \geq 2^\lambda$ by $M = p'q' > 2^{2\ell(\lambda)-4} \geq 2^{2\lambda} \geq 2^{2cn}$, which holds by [LEMMA 2.10](#) because $\ell(\lambda) \geq \lambda + 2$. In the injective mode, the union bound over the $3^n - 1$ nonzero vectors v becomes

$$\Pr \left[\text{E}_{\text{inj}} \mid (a_j)_{j \in [n]} \xleftarrow{\$} (\mathbb{Z}/M\mathbb{Z})^n \right] \leq \frac{3^n - 1}{M} < \frac{3^n}{2^{2cn}} = 2^{-(2c - \log_2 3)n},$$

which is negligible because $2c > \log_2 3$. Since $2c > \log_2 3 > 1$, we have $2^n < 2^{2cn} < M$ for every $n \geq 1$, so $\gamma < r < m \leq 2^n < M$ in the mode transition. The remaining steps of the proof of [THEOREM 5.16](#) hold for every rational $c > 0$, and $f_\perp$ is the same for every sampler. $\blacksquare$

For $c < 1$, the interval bound $\gamma = \lfloor (r - 1)/n \rfloor < 2^n$ of the lossy mode can exceed the smallest prime factor $\min\{p', q'\}$ of the group order $M = p'q'$, since $p', q' < 2^{\ell(\lambda)-1}$ and $\lambda = \lceil cn \rceil$. The generic-group bound of [THEOREM 3.11](#) holds for every γ smaller than the group order q , with the term $1/\gamma + 2/\pi_q$ for each pair of forms, and hence also when γ exceeds the smallest prime factor π_q of q .

For the plain construction, we specialize [FIGURE 5.3](#) to one coordinate, since the factors of every output are the bases g_j and, in the lossy mode, a partial product is determined by a bounded subset sum.

THEOREM 5.18 (Strong Enumeration for the Plain ELF). The construction in [FIGURE 5.4](#) has strongly efficiently enumerable images in the sense of [DEFINITION 5.2](#). More precisely, there is a deterministic algorithm that, on input (f, r) with $m \geq 2$, $r \in [m]$, and $f \in \text{supp}(\text{ELF.Gen}(m, r))$, returns $\{\text{ELF.Eval}(f, u) \mid u \in [m]\}$ using $\mathcal{O}(nr)$ group multiplications and runs in time $\text{poly}(r, \log m)$. $\lrcorner$

Proof. We specialize the prefix dynamic program of [FIGURE 5.3](#) to one coordinate and omit its final DDLog_N step. If $f = f_\perp$, then $\text{ELF.Eval}(f_\perp, \cdot)$ is constant, and we return the list (1). Otherwise, we write $f = (0, m, n, \lambda, \text{pp}, (g_j)_{j \in [n]})$, take $\mathbb{A} := \mathbb{G}(\text{pp})$ and $C_j := g_j$, and run the updates of [FIGURE 5.3](#) from $E = \{1\}$ and $L = \emptyset$. By the prefix invariant in the proof of [THEOREM 5.11](#), which holds over

every finite abelian group, the final deduplicated set is

$$\left\{ \prod_{j=1}^n g_j^{x_j(u)} \mid u \in [m] \right\} = \{\text{ELF.Eval}(f, u) \mid u \in [m]\}.$$

If $r = m$, then $|E| = 1$ and at most $m - 1$ smaller processed prefixes occur at any position, so $|E| + |L| \leq r + 1$. If $r < m$, we write $\gamma = \lfloor (r - 1)/n \rfloor$, and we set $s_j := 0$ for $j \in [n]$ when $\gamma = 0$. When $\gamma \geq 1$, the definition of ELF.Gen gives $g_j = g^{s_j}$ with $s_j \in [\gamma]$. Every partial product is then g^a for a partial sum $a \in [0..n\gamma]$, so at most $n\gamma + 1 \leq r$ elements remain after deduplication, and $|E| + |L| \leq r + 1$. As in [FIGURE 5.3](#), we return $\emptyset$ when $|E| + |L| > r + 1$, and by these bounds, this condition is false at every position for every f in the support. Therefore, each iteration forms at most $2(r + 1)$ products, and the algorithm uses $\mathcal{O}(nr)$ group multiplications and runs in time $\text{poly}(r, \log m)$. ■

Uniform sampling from the lossy image, as considered in [\[Zha16\]](#), follows from the exact image list, since a uniform position of a list without repetitions is a uniform element of the image for every distribution of preimage sizes.

COROLLARY 5.19 (Uniform Sampling for the Plain ELF). For every $m \geq 2$, every $r \in [m]$, and every $f \in \text{supp}(\text{ELF.Gen}(m, r))$, we can sample uniformly from $\{\text{ELF.Eval}(f, u) \mid u \in [m]\}$ in time $\text{poly}(r, \log m)$. ┘

Proof. We enumerate the image with the algorithm of [THEOREM 5.18](#) and return a uniformly random element of the returned list. Since this list consists of the distinct elements of the image, each element of the image is returned with probability one over the size of the image, and the running time is $\text{poly}(r, \log m)$. ■

6 All-But-Some and All-But-One Trapdoor ELFs

Following [\[Zha19\]](#), we give trapdoor ELFs branches: the functions on a selected set of polynomially many branches are lossy, the functions on every other branch are invertible, and the two generation modes are indistinguishable even given inversion queries on the nonlossy branches. If we evaluate several lossy functions in parallel, as in all-but-one constructions of lossy trapdoor functions [\[PW08\]](#), the lossy image size is the product of their image sizes, which is not polynomially bounded [\[Zha19\]](#). We use one Damgård–Jurik hash key of [§4](#) for all branches and subtract a branch term from the encodings, so that on branch b the input is multiplied by $P(\beta(b))$ for an integer $\beta(b)$ and a polynomial P whose roots are the integers of the selected branches.

6.1 All-But-Some and All-But-One Syntax

We write $a \geq 1$ for the branch length, so that a branch is an element of $\{0, 1\}^a$. In the selective notion of [\[Zha19\]](#), the adversary chooses the lossy branches before it receives the public description, and its second stage receives its first-stage state and an inversion oracle on every remaining branch. Our definition generalizes the all-but-one trapdoor ELF of [\[Zha19\]](#), which has one lossy branch, to a selected set $\mathcal{S}^*$ of s lossy branches: both generators receive 1^s , the lossy generator receives $\mathcal{S}^*$, the length bounds and the failure probabilities are polynomial and negligible in $n_m + a + s$, and the selective-security threshold depends on polynomial bounds A and S for the branch length and for s .

DEFINITION 6.1 (All-But-Some Trapdoor ELF). For every $m \geq 2$, write $n_m := \lceil \log_2 m \rceil$. An all-but-some trapdoor extremely lossy function consists of two probabilistic polynomial-time generation algorithms and two deterministic polynomial-time algorithms, $\text{ABS-TELF} = (\text{ABS-TELF.GenInj}, \text{ABS-TELF.GenLos}, \text{ABS-TELF.Eval}, \text{ABS-TELF.Inv})$. Every public description fk determines a finite output space $\mathcal{Y}_{\text{fk}}$ with an injective canonical encoding of length $\text{poly}(n_m + a + s)$, and membership testing, encoding, and decoding run in time $\text{poly}(n_m + a + s)$. The algorithms have the following syntax.

- $(\text{fk}, \text{td}) \xleftarrow{\$} \text{ABS-TELF.GenInj}(m, 1^a, 1^s)$. On input integers $m \geq 2$ and $a, s \geq 1$ with $s \leq 2^a$, the injective generator samples a public description fk and an inversion trapdoor td for the branches in $\{0, 1\}^a$.
- $(\text{fk}, \text{td}) \xleftarrow{\$} \text{ABS-TELF.GenLos}(m, 1^a, 1^s, r, \mathcal{S}^*)$. On input m, a , and s as for ABS-TELF.GenInj , $r \in [m]$, and $\mathcal{S}^* \subseteq \{0, 1\}^a$ with $|\mathcal{S}^*| = s$, the lossy generator samples a public description fk and an inversion trapdoor td for the branches in $\{0, 1\}^a \setminus \mathcal{S}^*$.
- $y \leftarrow \text{ABS-TELF.Eval}(\text{fk}, b, u)$. On input a public description fk and $(b, u) \in \{0, 1\}^a \times [m]$, the evaluation algorithm returns the value $y \in \mathcal{Y}_{\text{fk}}$ of branch b at u . We write $f_{\text{fk}, b}(u) := \text{ABS-TELF.Eval}(\text{fk}, b, u)$.
- $u \leftarrow \text{ABS-TELF.Inv}(\text{fk}, \text{td}, b, y)$. On input a public description fk , a trapdoor td , a branch b , and a string y , the inversion algorithm returns an element of $[m] \cup \{\perp\}$, and it returns $\perp$ whenever $(b, y) \notin \{0, 1\}^a \times \mathcal{Y}_{\text{fk}}$.

We call ABS-TELF an all-but-some trapdoor ELF if it has the following properties.

- **Compact descriptions.** Public descriptions, trapdoors, branches, and outputs have length $\text{poly}(n_m + a + s)$, and every algorithm runs in time polynomial in its input length.
- **Injective mode.** There is a negligible function $\nu_{\text{inj}}: \mathbb{N} \rightarrow [0, 1]$ such that for all $m \geq 2$ and $a, s \geq 1$ with $s \leq 2^a$, with $z := n_m + a + s$, the event $\text{E}_{\text{inj}}(\text{fk}, \text{td})$ that

$$\forall (b, u) \in \{0, 1\}^a \times [m]: \text{ABS-TELF.Inv}(\text{fk}, \text{td}, b, f_{\text{fk}, b}(u)) = u$$

satisfies

$$\Pr \left[\text{E}_{\text{inj}}(\text{fk}, \text{td}) \mid (\text{fk}, \text{td}) \xleftarrow{\$} \text{ABS-TELF.GenInj}(m, 1^a, 1^s) \right] \geq 1 - \nu_{\text{inj}}(z).$$

- **Lossy mode.** There is a negligible function $\nu_{\text{los}}: \mathbb{N} \rightarrow [0, 1]$ such that for all $m \geq 2$ and $a, s \geq 1$ with $s \leq 2^a$, every $r \in [m]$, and every $\mathcal{S}^* \subseteq \{0, 1\}^a$ with $|\mathcal{S}^*| = s$, with $z := n_m + a + s$, the event $\text{E}_{\text{los}}(\text{fk}, \text{td})$ that

$$\begin{aligned} & \forall b \in \mathcal{S}^*: |\text{Im}(f_{\text{fk}, b})| \leq r \\ & \wedge \forall (b, u) \in (\{0, 1\}^a \setminus \mathcal{S}^*) \times [m]: \text{ABS-TELF.Inv}(\text{fk}, \text{td}, b, f_{\text{fk}, b}(u)) = u \end{aligned}$$

satisfies

$$\Pr \left[\text{E}_{\text{los}}(\text{fk}, \text{td}) \mid (\text{fk}, \text{td}) \xleftarrow{\$} \text{ABS-TELF.GenLos}(m, 1^a, 1^s, r, \mathcal{S}^*) \right] \geq 1 - \nu_{\text{los}}(z).$$

- **Selective security.** For $c \in \{0, 1\}$, integers $m \geq 2, a, s \geq 1$ with $s \leq 2^a, r \in [m]$, and a two-stage nonuniform probabilistic circuit $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ whose second stage returns a bit, the experiment $\text{G}_{\text{ABS-TELF}, \mathcal{A}}^c(m, a, s, r)$ with $n := n_m$ runs as follows.
 - (1) It samples $(\mathcal{S}^*, \text{st}) \xleftarrow{\$} \mathcal{A}_1(1^n, m, 1^a, 1^s, r)$ and returns 0 unless $\mathcal{S}^* \subseteq \{0, 1\}^a$ and $|\mathcal{S}^*| = s$.

- (2) If $c = 0$, it samples $(\text{fk}, \text{td}) \xleftarrow{\$} \text{ABS-TELF.GenInj}(m, 1^a, 1^s)$. If $c = 1$, it samples $(\text{fk}, \text{td}) \xleftarrow{\$} \text{ABS-TELF.GenLos}(m, 1^a, 1^s, r, \mathcal{S}^*)$.
- (3) It returns $\mathcal{A}_2^{\mathcal{O}_{\text{inv}}}(\text{st}, \text{fk})$ for the total oracle

$$\mathcal{O}_{\text{inv}}(b, y) := \begin{cases} \perp, & (b, y) \notin (\{0, 1\}^a \setminus \mathcal{S}^*) \times \mathcal{Y}_{\text{fk}}, \\ \text{ABS-TELF.Inv}(\text{fk}, \text{td}, b, y), & (b, y) \in (\{0, 1\}^a \setminus \mathcal{S}^*) \times \mathcal{Y}_{\text{fk}}. \end{cases}$$

Let A and S be positive integer-valued polynomials, the bounds on the branch length and on the selected-set size. For every polynomial p and every positive polynomial δ , there is a polynomial Q_{ABS} such that for every sufficiently large m , with $n := n_m$, every $a \in [A(n)]$, every $s \in [S(n+a)]$ with $s \leq 2^a$, every $r \in [Q_{\text{ABS}}(n+a)..m]$, and every two-stage circuit $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ of size at most $p(n+a)$,

$$p_c := \Pr \left[G_{\text{ABS-TELF}, \mathcal{A}}^c(m, a, s, r) = 1 \right] \quad (c \in \{0, 1\}), \quad |p_0 - p_1| \leq \frac{1}{\delta(n+a)}.$$

┘

For $s = 1$, the selected set is one branch b^* , and we obtain the all-but-one notion of [Zha19], with signatures that omit the unary input 1^1 and take b^* in place of $\{b^*\}$.

DEFINITION 6.2 (All-But-One Trapdoor ELF). An all-but-one trapdoor ELF is a tuple of algorithms $\text{ABO-TELF} = (\text{ABO-TELF.GenInj}, \text{ABO-TELF.GenLos}, \text{ABO-TELF.Eval}, \text{ABO-TELF.Inv})$ such that the tuple defined by $\text{ABS-TELF.Eval} := \text{ABO-TELF.Eval}$, $\text{ABS-TELF.Inv} := \text{ABO-TELF.Inv}$, and

$$\begin{aligned} \text{ABS-TELF.GenInj}(m, 1^a, 1^1) &:= \text{ABO-TELF.GenInj}(m, 1^a), \\ \text{ABS-TELF.GenLos}(m, 1^a, 1^1, r, \{b^*\}) &:= \text{ABO-TELF.GenLos}(m, 1^a, r, b^*) \end{aligned}$$

satisfies **DEFINITION 6.1** with every quantifier over s restricted to $s = 1$ and with $S \equiv 1$. We write Q_{ABO} for the polynomial Q_{ABS} of its selective-security clause, and $G_{\text{ABO-TELF}, \mathcal{A}}^c(m, a, r)$ for the experiment $G_{\text{ABS-TELF}, \mathcal{A}}^c(m, a, 1, r)$ in which $\mathcal{A}_1$ receives $(1^n, m, 1^a, r)$ and returns b^* in place of $\{b^*\}$. ┘

For $a = 1$ and $b^* = 0$, we keep only branch 0 and obtain a trapdoor ELF in the sense of **DEFINITION 5.1**: except with negligible probability, under ABO-TELF.GenInj , ABO-TELF.Inv inverts branch 0 on every input, and under ABO-TELF.GenLos , branch 0 has image size at most r , and selective security for adversaries that make no oracle query implies mode indistinguishability.

6.2 Branch Polynomials

On every nonselected branch b , we invert by dividing by the value of a branch polynomial at $\beta(b)$, so we need these values to be units modulo every honestly sampled modulus. For $b = (b_1, \dots, b_a) \in \{0, 1\}^a$ and $\tau := 2^a$, we write

$$\beta(b) := \sum_{k=1}^a 2^{k-1} b_k.$$

Hence, β is a bijection from $\{0, 1\}^a$ to $[0, \tau - 1]$, and $\tau \notin [0, \tau - 1]$. In the branch constructions, every ELTDH input is a single residue $x \in \mathbb{Z}/N\mathbb{Z}$, which we use only as an input of ELTDH.Hash and ELTDH.Enc , so its bit decomposition appears only inside **FIGURE 4.1**. Whenever we pass an integer such as τ , $\beta(b)$, or a coefficient of a branch polynomial to ELTDH.Gen , we pass its residue

class modulo the modulus of hk .

For a set $B^* = \{\beta_1^*, \dots, \beta_s^*\} \subseteq [0.. \tau - 1]$ of size s , we define the integer polynomials

$$P_{B^*}(T) := \prod_{j=1}^s (\beta_j^* - T), \quad P_\tau(T) := (\tau - T)^s,$$

both of degree s with leading coefficient $(-1)^s$. On the branches where we divide by them, both polynomials take values that are nonzero products of s integers of absolute value at most 2^a , and at the group parameter $\lambda_{\text{grp}}(n + as)$ both prime factors of the modulus exceed 2^{as} .

LEMMA 6.3 (Invertible Branch Multipliers). Let $n, a, s \in \mathbb{N}_{>0}$ with $s \leq 2^a$, let $\tau := 2^a$ and $\lambda := \lambda_{\text{grp}}(n + as)$, and let $(N, g) \in \text{supp}(\text{DJGen}(1^\lambda))$. Then $\gcd(P_\tau(\beta), N) = 1$ for every $\beta \in [0.. \tau - 1]$, and $\gcd(P_{B^*}(\beta), N) = 1$ for every $B^* \subseteq [0.. \tau - 1]$ with $|B^*| = s$ and every $\beta \in [0.. \tau - 1] \setminus B^*$. $\square$

Proof. Every factor of either product is a nonzero integer of absolute value at most 2^a : we have $1 \leq \tau - \beta \leq \tau = 2^a$ because $0 \leq \beta \leq \tau - 1$, and $1 \leq |\beta_j^* - \beta| \leq \tau - 1$ because $\beta, \beta_j^* \in [0.. \tau - 1]$ and $\beta \notin B^*$. Hence, $P_\tau(\beta)$ and $P_{B^*}(\beta)$ are nonzero integers of absolute value at most 2^{as} . We write $N = pq$ as in [DEFINITION 2.8](#), so $p, q > 2^{\ell(\lambda)-1}$ with $\ell(\lambda) \geq \lambda + 2$, and $\lambda = \lambda_{\text{grp}}(n + as) \geq n + as$ by the choice of λ_{grp} in [§5](#). Hence, $p, q > 2^{\lambda+1} > 2^{as}$, so neither prime divides either product, and both greatest common divisors equal one. $\blacksquare$

6.3 All-But-Some Construction

The polynomial of the construction is $P := P_\tau$ in the injective mode and $P := P_{\beta(\mathcal{S}^*)}$ in the lossy mode, where $\beta(\mathcal{S}^*) = \{\beta(b) \mid b \in \mathcal{S}^*\}$ has size s because β is injective. We write its coefficients as

$$P(T) = (-1)^s T^s + \sum_{i=0}^{s-1} c_i T^i, \quad c_0, \dots, c_{s-1} \in \mathbb{Z}.$$

We generate two independent bundles $q \in \{0, 1\}$ of s encoding keys each, where key (q, i) encodes the coefficient c_i , and on branch b we combine the encodings of bundle q with the weights $1, \beta(b), \dots, \beta(b)^{s-1}$. For a public description with modulus N , we take $\mathcal{Y}_{\text{fk}} := (\mathbb{Z}/N^2\mathbb{Z})^\times \times (\mathbb{Z}/N\mathbb{Z})^2$ with $\lambda := \lambda_{\text{grp}}(n + as)$. Following the padding convention of [§5](#), we encode the group coordinate in exactly $4\ell(\lambda)$ bits and each residue coordinate in exactly $2\ell(\lambda)$ bits, and we read the concatenated $8\ell(\lambda)$ -bit string as its nonnegative integer value plus one. The resulting encoding is an injective map from $\mathcal{Y}_{\text{fk}}$ to $[2^{8\ell(\lambda)}]$, and membership testing, encoding, and decoding take time $\text{poly}(\lambda)$, which is polynomial in $n + a + s$ because $n + as \leq (n + a + s)^2$. We define the construction in [FIGURE 6.1](#).

From the decoded values of each bundle, we reconstruct the value of its polynomial at the branch. We fix a public description in [FIGURE 6.1](#) whose hash key $\text{hk} \in \text{supp}(\text{ELTDH.Setup}(1^\lambda, 1^1, r'))$, for some r' , has modulus N , a bundle $q \in \{0, 1\}$, and integers $c_0, \dots, c_{s-1}$ with $(\text{ek}_{q,i}, \text{td}_{q,i}) \in \text{supp}(\text{ELTDH.Gen}(\text{hk}, (c_i)))$ for every $i \in [0..s-1]$, and we let P be the polynomial with these coefficients and leading term $(-1)^s T^s$. For a branch b with $\beta := \beta(b)$ and an input $u \in [m]$ with $x = u - 1$, let $(h, E_0, E_1) = \text{ABS-TELF.Eval}(\text{fk}, b, u)$. By [COROLLARY 4.6](#), $e_{q,i} - \text{ELTDH.Dec}(\text{td}_{q,i}, h) = c_i x$ for every i , and we weight these s identities by $1, \beta, \dots, \beta^{s-1}$, add $(-1)^s \beta^s x$, and obtain

$$E_q - \sum_{i=0}^{s-1} \beta^i \text{ELTDH.Dec}(\text{td}_{q,i}, h) = P(\beta)x. \quad (6.1)$$

<pre> <u>ABS-TELF.GenInj($m, 1^a, 1^s$)</u> 00 assert $m \geq 2 \wedge a \geq 1 \wedge 1 \leq s \leq 2^a$ 01 $n \leftarrow \lceil \log_2 m \rceil$; $\lambda \leftarrow \lambda_{\text{grp}}(n + as)$; $\tau \leftarrow 2^a$ 02 $P \leftarrow P_\tau$ 03 $R \leftarrow 2^{2\ell(\lambda)}$; $\text{hk} \xleftarrow{\\$} \text{ELTDH.Setup}(1^\lambda, 1^1, R)$ 04 for $(q, i) \in \{0, 1\} \times [0..s-1]$ 05 $(\text{ek}_{q,i}, \text{td}_{q,i}) \xleftarrow{\\$} \text{ELTDH.Gen}(\text{hk}, (c_i))$ 06 $\text{fk} \leftarrow (m, n, a, s, \lambda, \text{hk}, (\text{ek}_{q,i})_{q,i})$ 07 return $(\text{fk}, (P, (\text{td}_{q,i})_{q,i}))$ <u>ABS-TELF.GenLos($m, 1^a, 1^s, r, \mathcal{S}^*$)</u> 08 assert $m \geq 2 \wedge a \geq 1 \wedge 1 \leq s \leq 2^a \wedge r \in [m] \wedge \mathcal{S}^* \subseteq \{0, 1\}^a \wedge \mathcal{S}^* = s$ 09 if $r = m$ return <u>ABS-TELF.GenInj</u>($m, 1^a, 1^s$) 10 $n \leftarrow \lceil \log_2 m \rceil$; $\lambda \leftarrow \lambda_{\text{grp}}(n + as)$ 11 $P \leftarrow P_{\beta(\mathcal{S}^*)}$ 12 $\text{hk} \xleftarrow{\\$} \text{ELTDH.Setup}(1^\lambda, 1^1, r)$ 13 for $(q, i) \in \{0, 1\} \times [0..s-1]$ 14 $(\text{ek}_{q,i}, \text{td}_{q,i}) \xleftarrow{\\$} \text{ELTDH.Gen}(\text{hk}, (c_i))$ 15 $\text{fk} \leftarrow (m, n, a, s, \lambda, \text{hk}, (\text{ek}_{q,i})_{q,i})$ 16 return $(\text{fk}, (P, (\text{td}_{q,i})_{q,i}))$ </pre>	<pre> <u>ABS-TELF.Eval(fk, b, u)</u> 17 parse $\text{fk} = (m, n, a, s, \lambda, \text{hk}, (\text{ek}_{q,i})_{q,i})$ 18 parse $\text{hk} = ((N, g), 1, w, L_{\text{exp}}, g_0, (g_{1,k})_k)$ 19 assert $b \in \{0, 1\}^a \wedge u \in [m]$ 20 $x \leftarrow u - 1 \in \mathbb{Z}/N\mathbb{Z}$; $\beta \leftarrow \beta(b)$ 21 $h \leftarrow \text{ELTDH.Hash}(\text{hk}, (x); 0)$ 22 for $q \in \{0, 1\}$ 23 $E_q \leftarrow (-1)^s \beta^s x$ 24 for $i \in [0..s-1]$ 25 $e_{q,i} \leftarrow \text{ELTDH.Enc}(\text{ek}_{q,i}, (x); 0)$; 26 $E_q \leftarrow E_q + \beta^i e_{q,i}$ 27 return (h, E_0, E_1) <u>ABS-TELF.Inv($\text{fk}, \text{td}, b, y$)</u> 27 parse $\text{fk} = (m, n, a, s, \lambda, \text{hk}, (\text{ek}_{q,i})_{q,i})$; $\text{td} = (P, (\text{td}_{q,i})_{q,i})$ 28 parse $\text{hk} = ((N, g), 1, w, L_{\text{exp}}, g_0, (g_{1,k})_k)$ 29 if $b \notin \{0, 1\}^a \vee y \notin \mathcal{Y}_{\text{fk}}$ return $\perp$ 30 parse $y = (h, E_0, E_1)$; $\beta \leftarrow \beta(b)$; $d \leftarrow P(\beta) \bmod N$ 31 if $\gcd(\text{rep}(d), N) \neq 1$ return $\perp$ 32 $D \leftarrow 0$ 33 for $i \in [0..s-1]$ 34 $D \leftarrow D + \beta^i \text{ELTDH.Dec}(\text{td}_{0,i}, h)$ 35 $x \leftarrow \text{rep}(d^{-1}(E_0 - D))$ 36 if $x \geq m$ return $\perp$ 37 $u \leftarrow x + 1$ 38 if <u>ABS-TELF.Eval</u>(fk, b, u) $\neq y$ return $\perp$ 39 return u </pre>
--	--

FIGURE 6.1. All-but-some trapdoor extremely lossy function with selected-set size s . The ELTDH algorithms are those of FIGURE 4.1, and $c_0, \dots, c_{s-1}$ are the coefficients of $T^0, \dots, T^{s-1}$ in P .

Hence, if $P(\beta)$ is a unit modulo N , then we recover x from bundle q , and if $P(\beta) = 0$, then E_q is a function of the digest h .

In the security proof, we answer inversion queries through whichever bundle encodes a polynomial that is a unit at the queried branch, so we need the answer to be the same for every such bundle. For $q \in \{0, 1\}$, we write ABS-TELF.Inv_q for the algorithm ABS-TELF.Inv of FIGURE 6.1 in which we compute D from the trapdoors $\text{td}_{q,i}$ and use E_q in place of E_0 , so that $\text{ABS-TELF.Inv}_0 = \text{ABS-TELF.Inv}$, and for a polynomial P , we write $\text{td}^P := (P, (\text{td}_{q,i})_{q,i})$.

LEMMA 6.4 (Inversion Through One Bundle). Let fk , N , q , $c_0, \dots, c_{s-1}$, and P be defined as in EQUATION (6.1), let $b \in \{0, 1\}^a$, and let y be any string. If $\gcd(P(\beta(b)), N) \neq 1$, then $\text{ABS-TELF.Inv}_q(\text{fk}, \text{td}^P, b, y) = \perp$. If $\gcd(P(\beta(b)), N) = 1$, then

$$\text{ABS-TELF.Inv}_q(\text{fk}, \text{td}^P, b, y) = \begin{cases} u, & u \in [m] \wedge \text{ABS-TELF.Eval}(\text{fk}, b, u) = y, \\ \perp, & y \notin \text{Im}(f_{\text{fk}, b}), \end{cases} \quad (6.2)$$

and the preimage u in the first case is unique. J

Proof. We decode bundle q and divide by $P(\beta(b))$, and after the reevaluation test we return only a

genuine preimage. The first claim is the test on $\gcd(\text{rep}(d), N)$ in [FIGURE 6.1](#). For the second, if ABS-TELF.Inv_q returns some $u \neq \perp$, then $\text{ABS-TELF.Eval}(\text{fk}, b, u) = y$ by its last test. Conversely, let $u \in [m]$ satisfy $\text{ABS-TELF.Eval}(\text{fk}, b, u) = y = (h, E_0, E_1)$, and let $x = u - 1$. Every supported modulus, including the fallback modulus, satisfies $N > 2^{2\ell(\lambda)-2} \geq 2^\lambda \geq 2^n \geq m$ by [DEFINITION 4.3](#), so $x \in [0..m-1]$ is its own representative. By [EQUATION \(6.1\)](#), $E_q - D = P(\beta(b))x$, so ABS-TELF.Inv_q computes $\text{rep}(d^{-1}(E_q - D)) = u - 1 < m$, passes the reevaluation test, and returns u . Any two preimages u, u' satisfy $P(\beta(b))(u - 1) = P(\beta(b))(u' - 1)$ in $\mathbb{Z}/N\mathbb{Z}$ by the same identity, and dividing by the unit $P(\beta(b))$, we obtain $u = u'$. ■

Functionality follows from [EQUATION \(6.1\)](#) and [LEMMA 6.3](#), and selective security follows from the function privacy and the mode indistinguishability of ELTDH.

THEOREM 6.5 (Selective All-But-Some Trapdoor ELF). Suppose that DJGen satisfies DCR against nonuniform PPT adversaries and tunable DSED. Then the construction in [FIGURE 6.1](#) is a selectively secure all-but-some trapdoor ELF in the sense of [DEFINITION 6.1](#). For all $m \geq 2$ and $a, s \geq 1$ with $s \leq 2^a$, with $n := \lceil \log_2 m \rceil$ and $w := 2\ell(\lambda_{\text{grp}}(n + as))$, and excluding the modulus, the generator, and integer metadata, its public description contains $(2s + 1)(w + 1)$ group elements, its trapdoor contains $2s$ ELTDH exponents, and its output contains one group element and two residues. ┘

Proof. First, we derive functionality from [EQUATION \(6.1\)](#) according to whether $P(\beta)$ is a unit or zero, and we check compactness for growing s . After that, we prove selective security with $2s$ function-privacy transitions, one change of the bundle that answers the oracle, and one hash-key mode transition. For the functionality, we fix $m \geq 2$ and $a, s \geq 1$ with $s \leq 2^a$, and we write $n := \lceil \log_2 m \rceil$ and $\lambda := \lambda_{\text{grp}}(n + as)$. In the injective mode, $P = P_\tau$, and setup samples hk at the full parameter $R = 2^{2\ell(\lambda)} = R_{\text{full}}(\lambda, 1)$. Outside the fallback event E_{full} , the hash key has a modulus N with $(N, g) \in \text{supp}(\text{DJGen}(1^\lambda))$, so $\gcd(P_\tau(\beta(b)), N) = 1$ for every $b \in \{0, 1\}^a$ by [LEMMA 6.3](#), and by [EQUATION \(6.2\)](#) with $q = 0$, we have $\text{ABS-TELF.Inv}(\text{fk}, \text{td}, b, f_{\text{fk}, b}(u)) = u$ for every $(b, u) \in \{0, 1\}^a \times [m]$. If $r = m$, then ABS-TELF.GenLos returns the output of ABS-TELF.GenInj , so the inversion guarantee is the injective one, and every branch has image size at most $m = r$. If $r < m$, then $P = P_{\beta(\mathcal{S}^*)}$, and for $b \in \mathcal{S}^*$, we have $P(\beta(b)) = 0$, so by [EQUATION \(6.1\)](#), both residue coordinates of $f_{\text{fk}, b}(u)$ are functions of the digest $h = \text{ELTDH.Hash}(\text{hk}, (u - 1); 0)$, which is the first coordinate. Hence, $|\text{Im}(f_{\text{fk}, b})| \leq |\text{Im}(\Phi_{\text{hk}})| \leq r$ by [THEOREM 4.8](#), for every hash key in the support, the fallback key included. Outside the event that setup returns the fallback key, $\gcd(P_{\beta(\mathcal{S}^*)}(\beta(b)), N) = 1$ for every $b \in \{0, 1\}^a \setminus \mathcal{S}^*$ by [LEMMA 6.3](#) with $B^* = \beta(\mathcal{S}^*)$, so by [EQUATION \(6.2\)](#) with $q = 0$, we invert every input on these branches.

In both modes, inversion can fail only if setup returns the fallback key. At the full parameter this is the event E_{full} , of probability $f_{\text{full}}(\lambda, 1)$, and at a nonfull parameter it is the failure of $\text{DJGen}(1^\lambda)$, on which the full setup also returns the fallback key, so its probability is at most $f_{\text{full}}(\lambda, 1)$. Since $n + a + s - 1 \leq n + as \leq \lambda$, both failure probabilities are at most

$$\nu_{\text{inj}}(k) = \nu_{\text{los}}(k) := \sup_{\lambda' \geq k-1} f_{\text{full}}(\lambda', 1)$$

at $k = n + a + s$, and this function is negligible because $f_{\text{full}}(\cdot, 1)$ is negligible by the analysis in [§4.1](#) and the supremum over the tail of a negligible function is negligible.

COMPACTNESS. Every root of P_τ or $P_{\beta(\mathcal{S}^*)}$ is an integer of absolute value at most 2^a , and each

coefficient c_i is, up to sign, a sum of $\binom{s}{i} \leq 2^s$ products of $s - i$ roots. Hence, $|c_i| \leq 2^{s(a+1)}$, and every coefficient has $\mathcal{O}(s(a+1))$ bits. We compute all coefficients by multiplying the s linear factors one at a time, with $\mathcal{O}(s^2)$ integer operations on integers of $\mathcal{O}(s(a+1))$ bits, so both generators run in polynomial time. Since $n + as \leq (n + a + s)^2$, the parameters $\lambda = \lambda_{\text{grp}}(n + as)$ and $w = 2\ell(\lambda)$ are polynomially bounded in $n + a + s$. By [LEMMA 4.9](#), the hash key and each of the $2s$ encoding keys contain $w + 1$ group elements and each trapdoor one exponent, and evaluation and inversion make $\mathcal{O}(s)$ calls to ELTDH algorithms. Hence, the compactness clause of [DEFINITION 6.1](#) holds, with the element counts of the statement.

HYBRIDS. We fix polynomials A , S , and p and a positive polynomial δ as in [DEFINITION 6.1](#). Since the conclusion for nondecreasing polynomial upper bounds of p , δ , and S implies the conclusion for p , δ , and S , we assume that all three are nondecreasing. We fix m , with $n := \lceil \log_2 m \rceil$, and $a \in [A(n)]$, $s \in [S(n + a)]$ with $s \leq 2^a$, $r \in [Q_{\text{ABS}}(n + a)..m]$, where Q_{ABS} is a polynomial that depends only on p , δ , and S , and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ of size at most $p(n + a)$, and we write $z := n + a$ and $\lambda := \lambda_{\text{grp}}(n + as)$. If $r = m$, then ABS-TELF.GenLos returns the output of ABS-TELF.GenInj and the two experiments are identical, so we suppose henceforth that $r < m$. Since $\mathcal{A}_1$ returns $\mathcal{S}^*$ before key generation, the coefficient lists

$$P_\tau(T) = (-1)^s T^s + \sum_{i=0}^{s-1} c_i^\tau T^i, \quad P_{\beta(\mathcal{S}^*)}(T) = (-1)^s T^s + \sum_{i=0}^{s-1} c_i^* T^i$$

are functions of the output of $\mathcal{A}_1$.

We define the hybrids $\text{Hyb}_0, \dots, \text{Hyb}_{2s+1}$ and Hyb'_s . As in the experiments, each of them samples $(\mathcal{S}^*, \text{st}) \xleftarrow{\$} \mathcal{A}_1(1^n, m, 1^a, 1^s, r)$ and returns 0 unless $\mathcal{S}^*$ is a valid selected set. Otherwise, it generates fk as in [FIGURE 6.2](#), where j ranges over $[s]$ and Hyb'_s generates fk as Hyb_s does, and it returns $\mathcal{A}_2^\mathcal{O}(\text{st}, \text{fk})$. The oracle $\mathcal{O}$ returns $\perp$ on every query (b, y) with $b \in \mathcal{S}^*$. On every other query, it returns $\text{ABS-TELF.Inv}_1(\text{fk}, \text{td}^{P_\tau}, b, y)$ in $\text{Hyb}_0, \dots, \text{Hyb}_s$, and it returns $\text{ABS-TELF.Inv}_0(\text{fk}, \text{td}^{P_{\beta(\mathcal{S}^*)}}, b, y)$ in Hyb'_s and in $\text{Hyb}_{s+1}, \dots, \text{Hyb}_{2s+1}$. In Hyb_0 , both bundles encode P_τ , and ABS-TELF.Inv_0 and ABS-TELF.Inv_1 return the same answer for a common polynomial on every hash key, by the common test on $\text{gcd}(\text{rep}(d), N)$ and by [EQUATION \(6.2\)](#). Hence, Hyb_0 is the experiment $G_{\text{ABS-TELF}, \mathcal{A}}^0(m, a, s, r)$, and Hyb_{2s+1} , in which both bundles encode $P_{\beta(\mathcal{S}^*)}$ under a hash key at parameter r and the oracle is ABS-TELF.Inv , is the experiment $G_{\text{ABS-TELF}, \mathcal{A}}^1(m, a, s, r)$.

KEY TRANSITIONS. For $j \in [s]$, we change key $(0, j - 1)$ at [Line 9](#) from c_{j-1}^τ to c_{j-1}^* in the transition from Hyb_{j-1} to Hyb_j , and we bound this transition by a function-privacy reduction at residue arity one. Its selector receives $(1^\lambda, \text{hk})$, runs $(\mathcal{S}^*, \text{st}) \xleftarrow{\$} \mathcal{A}_1(1^n, m, 1^a, 1^s, r)$, and, if $\mathcal{S}^*$ is invalid, returns two equal predicates and a state on which the distinguisher returns 0. Otherwise, the selector computes $(c_i^\tau)_i$ and $(c_i^*)_i$, samples the keys $(0, i)$ with $i \neq j - 1$ and all keys of bundle 1 as in Hyb_j , with their trapdoors, and returns the predicates with coefficient vectors (c_{j-1}^τ) and (c_{j-1}^*) and the state st' consisting of $\mathcal{S}^*$, st , and these keys and trapdoors. The distinguisher receives $(\text{hk}, \text{st}', \text{ek})$ for the state st' of the selector, sets $\text{ek}_{0,j-1} := \text{ek}$, assembles fk , runs $\mathcal{A}_2(\text{st}, \text{fk})$, answers each permitted query with ABS-TELF.Inv_1 and the trapdoors of bundle 1, which encodes P_τ , and returns the output bit of $\mathcal{A}_2$. For every hash key, the fallback key included, the output of the distinguisher is distributed as the output of Hyb_{j-1} for the first predicate and as the output of Hyb_j for the second. With two bundles, we keep a complete bundle and its trapdoors while the other bundle is under the function-privacy challenge, whose trapdoor the reduction does not know. For $j \in [s]$, we change key

<u>Hyb₀</u>	<i>* injective key *</i>	<u>Hyb_j</u>	<i>* bundle 0, key j - 1 *</i>
00 $n \leftarrow \lceil \log_2 m \rceil$; $\lambda \leftarrow \lambda_{\text{grp}}(n + as)$; $R \leftarrow 2^{2\ell(\lambda)}$		05 $n \leftarrow \lceil \log_2 m \rceil$; $\lambda \leftarrow \lambda_{\text{grp}}(n + as)$; $R \leftarrow 2^{2\ell(\lambda)}$	
01 $\text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^1, R)$		06 $\text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^1, R)$	
02 for $(q, i) \in \{0, 1\} \times [0..s - 1]$		07 for $i \in [0..j - 2]$	
03 $(\text{ek}_{q,i}, \text{td}_{q,i}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (c_i^*))$		08 $(\text{ek}_{0,i}, \text{td}_{0,i}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (c_i^*))$	
04 return $\text{fk} \leftarrow (m, n, a, s, \lambda, \text{hk}, (\text{ek}_{q,i})_{q,i})$		09 $(\text{ek}_{0,j-1}, \text{td}_{0,j-1}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (c_{j-1}^*))$	
		10 for $i \in [j..s - 1]$	
		11 $(\text{ek}_{0,i}, \text{td}_{0,i}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (c_i^*))$	
		12 for $i \in [0..s - 1]$	
		13 $(\text{ek}_{1,i}, \text{td}_{1,i}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (c_i^*))$	
		14 return $\text{fk} \leftarrow (m, n, a, s, \lambda, \text{hk}, (\text{ek}_{q,i})_{q,i})$	
			<i>* lossy key *</i>
<u>Hyb_{s+j}</u>	<i>* bundle 1, key j - 1 *</i>	<u>Hyb_{2s+1}</u>	
15 $n \leftarrow \lceil \log_2 m \rceil$; $\lambda \leftarrow \lambda_{\text{grp}}(n + as)$; $R \leftarrow 2^{2\ell(\lambda)}$		25 $n \leftarrow \lceil \log_2 m \rceil$; $\lambda \leftarrow \lambda_{\text{grp}}(n + as)$	
16 $\text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^1, R)$		26 $\text{hk} \xleftarrow{\$} \text{ELTDH.Setup}(1^\lambda, 1^1, r)$	
17 for $i \in [0..s - 1]$		27 for $(q, i) \in \{0, 1\} \times [0..s - 1]$	
18 $(\text{ek}_{0,i}, \text{td}_{0,i}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (c_i^*))$		28 $(\text{ek}_{q,i}, \text{td}_{q,i}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (c_i^*))$	
19 for $i \in [0..j - 2]$		29 return $\text{fk} \leftarrow (m, n, a, s, \lambda, \text{hk}, (\text{ek}_{q,i})_{q,i})$	
20 $(\text{ek}_{1,i}, \text{td}_{1,i}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (c_i^*))$			
21 $(\text{ek}_{1,j-1}, \text{td}_{1,j-1}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (c_{j-1}^*))$			
22 for $i \in [j..s - 1]$			
23 $(\text{ek}_{1,i}, \text{td}_{1,i}) \xleftarrow{\$} \text{ELTDH.Gen}(\text{hk}, (c_i^*))$			
24 return $\text{fk} \leftarrow (m, n, a, s, \lambda, \text{hk}, (\text{ek}_{q,i})_{q,i})$			

FIGURE 6.2. Key generation in the hybrids for selective all-but-some security, for $j \in [s]$. The ELTDH algorithms are those of FIGURE 4.1, and the highlighted line is the line that changes from the preceding hybrid. The hybrid Hyb'_s generates its keys as Hyb_s does.

$(1, j - 1)$ at Line 21 in the transition from Hyb_{s+j-1} to Hyb_{s+j} , with $\text{Hyb}_{s+0} := \text{Hyb}'_s$, and we use the same reduction with the two bundles exchanged, where the distinguisher answers each permitted query with ABS-TELF.Inv_0 and the trapdoors of bundle 0, which encodes $P_{\beta(\mathcal{S}^*)}$.

From Hyb_s to Hyb'_s , we change only the bundle that answers the oracle. Outside $\mathbf{E}_{\text{full}}$, the hash key of Hyb_s has a modulus N with $(N, g) \in \text{supp}(\text{DJGen}(1^\lambda))$, so by LEMMA 6.3, $P_\tau(\beta(b))$ and $P_{\beta(\mathcal{S}^*)}(\beta(b))$ are units modulo N for every $b \in \{0, 1\}^a \setminus \mathcal{S}^*$, and by EQUATION (6.2), both oracles return the unique preimage or $\perp$. Hence, the two hybrids differ only on $\mathbf{E}_{\text{full}}$, and

$$|\Pr[\text{Hyb}_s = 1] - \Pr[\text{Hyb}'_s = 1]| \leq \Pr[\mathbf{E}_{\text{full}}] = f_{\text{full}}(\lambda, 1).$$

MODE TRANSITION. At Line 26, both bundles encode $P_{\beta(\mathcal{S}^*)}$, so our reduction receives a hash key hk from the distribution $\mathfrak{D}_{\lambda, r'}^{\text{mode}}$ of DEFINITION 4.1 at residue arity one, runs $\mathcal{A}_1$, samples every encoding key with its trapdoor for the coefficients c_i^* , runs $\mathcal{A}_2(\text{st}, \text{fk})$, answers each permitted query with ABS-TELF.Inv , and returns the output bit of $\mathcal{A}_2$. For every hash key in either distribution, its output is distributed as the output of Hyb_{2s} for $r' = R_{\text{full}}(\lambda, 1)$ and as the output of Hyb_{2s+1} for $r' = r$.

THRESHOLDS. Since $a, s \geq 1$ and $\lambda_{\text{grp}}(v) \geq v$, we have $z \leq n + as \leq \lambda$ and $s \leq \lambda$. Each reduction of this proof runs $\mathcal{A}$, of size at most $p(z) \leq p(\lambda)$, generates at most $2s$ further ELTDH keys, and answers each query with $\mathcal{O}(s)$ ELTDH decodings and one evaluation, so there is one polynomial $\hat{p}$ such that every reduction, for every admissible choice of $m, a, s, r, \mathcal{A}$, and the changed key, has size at most $\hat{p}(\lambda)$. We write $\hat{\delta}(v) := 2(2v + 1)\delta(v)$. There is λ_{fp} such that for every $\lambda \geq \lambda_{\text{fp}}$ and every

admissible choice with $\lambda_{\text{grp}}(n + as) = \lambda$, the function-privacy reduction for this choice has advantage at most $1/\hat{\delta}(\lambda)$. Otherwise, for infinitely many λ , we fix an admissible choice whose reduction has advantage greater than $1/\hat{\delta}(\lambda)$, and we let $\mathcal{S}'$ and $\mathcal{D}'$ be the nonuniform families that equal the selector and the distinguisher of this reduction at these λ and that return two equal predicates and 0 at every other λ . Since $\mathcal{S}'$ is a nonuniform selector of size at most $\hat{p}(\lambda)$, its two ensembles are computationally indistinguishable by [THEOREM 4.12](#), and by [LEMMA 2.1](#) with size bound $\hat{p}$ and target $\hat{\delta}$, the advantage of $\mathcal{D}'$ is at most $1/\hat{\delta}(\lambda)$ for every sufficiently large λ , which contradicts the advantages of the fixed reductions.

We apply [THEOREM 4.13](#) at residue arity one with size bound $\hat{p}$ and target $\hat{\delta}$, and we obtain a polynomial Q_{hash} and λ_{mode} such that $\mathfrak{D}_{\lambda, r_0}^{\text{mode}} \equiv_{\hat{p}, 1/\hat{\delta}} \mathfrak{D}_{\lambda, r_1}^{\text{mode}}$ for every $\lambda \geq \lambda_{\text{mode}}$ and all $r_0, r_1 \in [Q_{\text{hash}}(\lambda) \dots R_{\text{full}}(\lambda, 1)]$. Since $r \leq m \leq 2^n < 2^\lambda < R_{\text{full}}(\lambda, 1)$, the parameter r lies in this interval once $r \geq Q_{\text{hash}}(\lambda)$. We take a polynomial upper bound G on the increasing function λ_{grp} and a nondecreasing polynomial upper bound $\overline{Q_{\text{hash}}}$ on Q_{hash} . From $n, a \leq z$ and $s \leq S(z)$, we obtain $n + as \leq z + zS(z)$ and $\lambda \leq G(z + zS(z))$, and we choose a polynomial Q_{ABS} with $Q_{\text{ABS}}(z) \geq \overline{Q_{\text{hash}}}(G(z + zS(z)))$. Then $r \geq Q_{\text{ABS}}(z) \geq Q_{\text{hash}}(\lambda)$ for every admissible a and s . Finally, since $f_{\text{full}}(\cdot, 1)$ is negligible, there is λ_{full} with $f_{\text{full}}(\lambda, 1) \leq 1/(2\delta(\lambda))$ for every $\lambda \geq \lambda_{\text{full}}$.

TOTAL. Since $\lambda \geq n = \lceil \log_2 m \rceil$, for every sufficiently large m , we have $\lambda \geq \max\{\lambda_{\text{fp}}, \lambda_{\text{mode}}, \lambda_{\text{full}}\}$. For such m , we sum the $2s$ key transitions, the oracle change, and the mode transition, and we use $s, z \leq \lambda$ and that δ is nondecreasing:

$$\begin{aligned} |\Pr[\text{Hyb}_0 = 1] - \Pr[\text{Hyb}_{2s+1} = 1]| &\leq \frac{2s+1}{\hat{\delta}(\lambda)} + f_{\text{full}}(\lambda, 1) \\ &\leq \frac{2\lambda+1}{2(2\lambda+1)\delta(\lambda)} + \frac{1}{2\delta(\lambda)} \leq \frac{1}{\delta(z)}. \end{aligned}$$

Since Hyb_0 and Hyb_{2s+1} are the two selective-security experiments, the selective-security clause of [DEFINITION 6.1](#) holds. ■

6.4 Enumerating Selected-Branch Images

For $r < m$ and every $b^* \in \mathcal{S}^*$, the integer $\beta(b^*)$ is a root of P , so by [EQUATION \(6.1\)](#), both residue coordinates on branch b^* are functions of the digest, and with the digest enumeration of [THEOREM 5.11](#), we obtain the exact selected-branch image, with an additional polynomial dependence on s .

COROLLARY 6.6 (Strong Enumeration on Selected Branches). There is a deterministic algorithm $\mathcal{E}$ such that, for all $m \geq 2$, $a, s \geq 1$ with $s \leq 2^a$, $r \in [m]$, $\mathcal{S}^* \subseteq \{0, 1\}^a$ with $|\mathcal{S}^*| = s$, $(\text{fk}, \text{td}) \in \text{supp}(\text{ABS-TELF.GenLos}(m, 1^a, 1^s, r, \mathcal{S}^*))$, and $b^* \in \mathcal{S}^*$, the output of $\mathcal{E}(\text{fk}, b^*, r)$ is $\text{Im}(\text{ABS-TELF.Eval}(\text{fk}, b^*, \cdot))$ and its running time is $\text{poly}(r, \log m + a + s)$. ┘

Proof. We list one representative input for every digest and evaluate the branch on these representatives. If $r = m$, we evaluate $\text{ABS-TELF.Eval}(\text{fk}, b^*, u)$ for all $u \in [m]$ and deduplicate the outputs, in time $m \cdot \text{poly}(\log m + a + s) = \text{poly}(r, \log m + a + s)$. We suppose henceforth that $r < m$. Then $P = P_{\beta(\mathcal{S}^*)}$ and $P(\beta(b^*)) = 0$, so by [EQUATION \(6.1\)](#), $\text{ABS-TELF.Eval}(\text{fk}, b^*, u)$ is a function of the digest $\text{ELTDH.Hash}(\text{hk}, (u-1); 0)$.

We enumerate the distinct digests of the inputs $u-1$ for $u \in [m]$ with the prefix dynamic program of [FIGURE 5.3](#), run on the hash bases of hk , and we attach to each deduplicated state one

processed prefix that produces it, keeping either prefix when two products coincide. By the bounded subset-sum argument in the proof of [THEOREM 5.11](#), at most r digests occur at every position, so the state bound and the running time are unchanged, and we know an input for every final digest. We evaluate $\text{ABS-TELF.Eval}(\text{fk}, b^*, \cdot)$ on these representatives and deduplicate. Every output on b^* arises from some digest, and equal digests have equal outputs, so the resulting list is the exact image, computed in time $\text{poly}(r, \lambda) \cdot \mathcal{O}(s) = \text{poly}(r, \log m + a + s)$ because $\lambda = \lambda_{\text{grp}}(n + as)$ and $n + as \leq (n + a + s)^2$. ■

With the exact image list of [COROLLARY 6.6](#), we can sample uniformly from every selected-branch image by choosing a uniform position in the deduplicated list, as for the list of [THEOREM 5.11](#), in time $\text{poly}(r, \log m + a + s)$, which is $\text{poly}(r, \log m + a)$ for $s = 1$.

6.5 The All-But-One Case

For $s = 1$ and $\mathcal{S}^* = \{b^*\}$, we obtain the all-but-one trapdoor ELF of [\[Zha19\]](#), whose inversion oracle is available on every branch other than b^* . Each bundle then consists of one encoding key, and in the proof of [THEOREM 6.5](#), we answer inversion queries with the trapdoor of the second key while we change the first key under function privacy, and then we exchange the two keys.

COROLLARY 6.7 (Selective All-But-One Trapdoor ELF). Suppose that DJGen satisfies DCR against nonuniform PPT adversaries and tunable DSED, and let $\text{ABO-TELF.Eval} := \text{ABS-TELF.Eval}$, $\text{ABO-TELF.Inv} := \text{ABS-TELF.Inv}$, and

$$\begin{aligned} \text{ABO-TELF.GenInj}(m, 1^a) &:= \text{ABS-TELF.GenInj}(m, 1^a, 1^1), \\ \text{ABO-TELF.GenLos}(m, 1^a, r, b^*) &:= \text{ABS-TELF.GenLos}(m, 1^a, 1^1, r, \{b^*\}) \end{aligned}$$

for the algorithms of [FIGURE 6.1](#). Then ABO-TELF is a selectively secure all-but-one trapdoor ELF in the sense of [DEFINITION 6.2](#). For all $m \geq 2$ and $a \geq 1$, with $n := \lceil \log_2 m \rceil$ and $w := 2\ell(\lambda_{\text{grp}}(n + a))$, and excluding the modulus, the generator, and integer metadata, its public description contains $3(w + 1)$ group elements, its trapdoor contains two ELTDH exponents, and its output contains one group element and two residues. There is a deterministic algorithm $\mathcal{E}$ such that, for all $m \geq 2$, $a \geq 1$, $r \in [m]$, $b^* \in \{0, 1\}^a$, and $(\text{fk}, \text{td}) \in \text{supp}(\text{ABO-TELF.GenLos}(m, 1^a, r, b^*))$, the output of $\mathcal{E}(\text{fk}, b^*, r)$ is $\text{Im}(\text{ABO-TELF.Eval}(\text{fk}, b^*, \cdot))$ and its running time is $\text{poly}(r, \log m + a)$. ┘

Proof. For $s = 1$, the polynomials in [FIGURE 6.1](#) are $P_\tau(T) = \tau - T$ and $P_{\{\beta^*\}}(T) = \beta^* - T$ with $\beta^* := \beta(b^*)$, so each bundle consists of one key encoding τ or β^* , and $E_q = e_{q,0} - \beta(b)x$. The selected set $\{b^*\}$ determines its unique branch, so the tuple that [DEFINITION 6.2](#) derives from ABO-TELF is the construction of [FIGURE 6.1](#) at $s = 1$. Hence, the security, functionality, and size claims are [THEOREM 6.5](#) at $s = 1$ and $\mathcal{S} \equiv 1$, with $Q_{\text{ABO}} := Q_{\text{ABS}}$, and the enumeration claim is [COROLLARY 6.6](#) at $s = 1$, since $\text{poly}(r, \log m + a + 1) = \text{poly}(r, \log m + a)$. ■

AI Disclosure and Provenance. Commercial AI tools, including ChatGPT and Claude, were used to assist with grammar, proofreading, expository polishing, and L^AT_EX formatting. These tools were not used in developing the scientific content, results, proofs, analyses, or conclusions, all of which were produced by the authors.

Acknowledgements. A. Datta and A. Dutta thank Nilabha Saha for discussions on the presentation of this work. A. Datta and D. Abram were supported by Input Output (iog.io) through their funding

of the Edinburgh Blockchain Technology Lab. A. Dutta was supported by Cyberagentur under the project “Multi-Party Computation in the Confidential Cloud (MPCC)”. L. Roy was supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme under grant agreement number 101124977 (DECRYPSIS), and by the Swiss National Science Foundation (SNSF) under the Starting grant “Low-Entropy Cryptography for the Quantum Era”.

REFERENCES

- [ABG⁺26] Gennaro Avitabile, Vincenzo Botta, Emanuele Giunta, Marcin Mielniczuk, and Francesco Migliaro. The malice of ELF: Practical anamorphic-resistant encryption without random oracles. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026, Part V*, volume 16545 of *Lecture Notes in Computer Science*, pages 303–333, Rome, Italy, May 10–14, 2026. Springer, Cham, Switzerland. doi:10.1007/978-3-032-25330-9_11. (Cited on pp. 3, 6, and 59).
- [ACK23] Thomas Attema, Pedro Capitão, and Lisa Kohl. On homomorphic secret sharing from polynomial-modulus LWE. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023: 26th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 13941 of *Lecture Notes in Computer Science*, pages 3–32, Atlanta, GA, USA, May 7–10, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-31371-4_1. (Cited on p. 9).
- [ADOS22] Damiano Abram, Ivan Damgård, Claudio Orlandi, and Peter Scholl. An algebraic framework for silent preprocessing with trustless setup and active security. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part IV*, volume 13510 of *Lecture Notes in Computer Science*, pages 421–452, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-15985-5_15. (Cited on p. 7).
- [AFLT12] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 572–590, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-29011-4_34. (Cited on pp. 10 and 23).
- [AFLT16] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly secure signatures from lossy identification schemes. *Journal of Cryptology*, 29(3):597–631, July 2016. doi:10.1007/s00145-015-9203-7. (Cited on p. 10).
- [AMR25a] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Slightly sublinear trapdoor hash functions and PIR from low-noise LPN. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025: 23rd Theory of Cryptography Conference, Part I*, volume 16268 of *Lecture Notes in Computer Science*, pages 280–300, Aarhus, Denmark, December 1–5, 2025. Springer, Cham, Switzerland. doi:10.1007/978-3-032-12287-2_10. (Cited on p. 9).
- [AMR25b] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Succinct oblivious tensor evaluation and applications: Adaptively-secure laconic function evaluation and trapdoor hashing for all circuits. In Michal Koucký and Nikhil Bansal, editors, *57th Annual ACM Symposium on Theory of Computing*, pages 1875–1886, Prague, Czechia, June 23–27, 2025. ACM Press. doi:10.1145/3717823.3718164. (Cited on p. 9).
- [BCD⁺25] Pedro Branco, Arka Rai Choudhuri, Nico Döttling, Abhishek Jain, Giulio Malavolta, and Akshayaram Srinivasan. Black-box non-interactive zero knowledge from vector trapdoor hash. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology –*

- EUROCRYPT 2025, Part IV*, volume 15604 of *Lecture Notes in Computer Science*, pages 64–92, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland. doi:[10.1007/978-3-031-91134-7_3](https://doi.org/10.1007/978-3-031-91134-7_3). (Cited on p. 9).
- [BCG⁺17] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, and Michele Orrù. Homomorphic secret sharing: Optimizations and applications. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 2105–2122, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. doi:[10.1145/3133956.3134107](https://doi.org/10.1145/3133956.3134107). (Cited on p. 9).
- [BDHL25] Rishabh Bhaduria, Nico Döttling, Carmit Hazay, and Chuanwei Lin. Laconic cryptography with preprocessing. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology – ASIACRYPT 2025, Part V*, volume 16249 of *Lecture Notes in Computer Science*, pages 228–261, Melbourne, VIC, Australia, December 8–12, 2025. Springer, Singapore, Singapore. doi:[10.1007/978-981-95-5116-3_8](https://doi.org/10.1007/978-981-95-5116-3_8). (Cited on p. 9).
- [BDSZ24] Pedro Branco, Nico Döttling, Akshayaram Srinivasan, and Riccardo Zanotto. Rate-1 fully local somewhere extractable hashing from DDH. In Qiang Tang and Vanessa Teague, editors, *PKC 2024: 27th International Conference on Theory and Practice of Public Key Cryptography, Part III*, volume 14603 of *Lecture Notes in Computer Science*, pages 356–386, Sydney, NSW, Australia, April 15–17, 2024. Springer, Cham, Switzerland. doi:[10.1007/978-3-031-57725-3_12](https://doi.org/10.1007/978-3-031-57725-3_12). (Cited on p. 9).
- [Ber80] Geoffrey C. Berresford. The Uniformity Assumption in the Birthday Problem. *Mathematics Magazine*, 53(5):286–288, 1980. doi:[10.1080/0025570x.1980.11976872](https://doi.org/10.1080/0025570x.1980.11976872). (Cited on pp. 19 and 20).
- [BFO08] Alexandra Boldyreva, Serge Fehr, and Adam O’Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 335–359, Santa Barbara, CA, USA, August 17–21, 2008. Springer Berlin Heidelberg, Germany. doi:[10.1007/978-3-540-85174-5_19](https://doi.org/10.1007/978-3-540-85174-5_19). (Cited on p. 13).
- [BG10] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 1–20, Santa Barbara, CA, USA, August 15–19, 2010. Springer Berlin Heidelberg, Germany. doi:[10.1007/978-3-642-14623-7_1](https://doi.org/10.1007/978-3-642-14623-7_1). (Cited on pp. 16, 18, 43, and 44).
- [BG25] Nir Bitansky and Rachit Garg. Succinct randomized encodings from laconic function evaluation, faster and simpler. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025, Part VII*, volume 15607 of *Lecture Notes in Computer Science*, pages 406–436, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland. doi:[10.1007/978-3-031-91098-2_15](https://doi.org/10.1007/978-3-031-91098-2_15). (Cited on p. 9).
- [BGI15] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 337–367, Sofia, Bulgaria,

- April 26–30, 2015. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-46803-6_12. (Cited on p. 9).
- [BGI16a] Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the circuit size barrier for secure computation under DDH. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 509–539, Santa Barbara, CA, USA, August 14–18, 2016. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-53018-4_19. (Cited on pp. 9 and 17).
- [BGI16b] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing: Improvements and extensions. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pages 1292–1303, Vienna, Austria, October 24–28, 2016. ACM Press. doi:10.1145/2976749.2978429. (Cited on p. 9).
- [BGI⁺18] Elette Boyle, Niv Gilboa, Yuval Ishai, Huijia Lin, and Stefano Tessaro. Foundations of homomorphic secret sharing. In Anna R. Karlin, editor, *ITCS 2018: 9th Innovations in Theoretical Computer Science Conference*, volume 94, pages 21:1–21:21, Cambridge, MA, USA, January 11–14, 2018. Leibniz International Proceedings in Informatics (LIPIcs). doi:10.4230/LIPIcs.ITCS.2018.21. (Cited on p. 9).
- [BHK13] Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Instantiating random oracles via UCEs. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 398–415, Santa Barbara, CA, USA, August 18–22, 2013. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-40084-1_23. (Cited on p. 3).
- [BK04] Mihir Bellare and Tadayoshi Kohno. Hash function balance and its impact on birthday attacks. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 401–418, Interlaken, Switzerland, May 2–6, 2004. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-540-24676-3_24. (Cited on p. 19).
- [BKPW12] Mihir Bellare, Eike Kiltz, Chris Peikert, and Brent Waters. Identity-based (lossy) trapdoor functions and applications. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 228–245, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-29011-4_15. (Cited on p. 9).
- [BKS19] Elette Boyle, Lisa Kohl, and Peter Scholl. Homomorphic secret sharing from lattices without FHE. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part II*, volume 11477 of *Lecture Notes in Computer Science*, pages 3–33, Darmstadt, Germany, May 19–23, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-17656-3_1. (Cited on p. 9).
- [BL17] Xavier Boyen and Qinyi Li. All-but-many lossy trapdoor functions from lattices and applications. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 298–331, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-63697-9_11. (Cited on p. 9).

- [BW10] Xavier Boyen and Brent Waters. Shrinking the keys of discrete-log-type lossy trapdoor functions. In Jianying Zhou and Moti Yung, editors, *ACNS 2010: 8th International Conference on Applied Cryptography and Network Security*, volume 6123 of *Lecture Notes in Computer Science*, pages 35–52, Beijing, China, June 22–25, 2010. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-13708-2_3. (Cited on p. 9).
- [CDG⁺17] Chongwon Cho, Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Antigoni Polychroniadou. Laconic oblivious transfer and its applications. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 33–65, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-63715-0_2. (Cited on p. 9).
- [CDG18] Sandro Coretti, Yevgeniy Dodis, and Siyao Guo. Non-uniform bounds in the random-permutation, ideal-cipher, and generic-group models. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 693–721, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-96884-1_23. (Cited on pp. 10, 31, 32, and 33).
- [CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, TX, USA, May 23–26, 1998. ACM Press. doi:10.1145/276698.276741. (Cited on p. 3).
- [CHP25] Geoffroy Couteau, Aditya Hegde, and Sihang Pu. Enhanced trapdoor hashing from DDH and DCR. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025, Part VI*, volume 15606 of *Lecture Notes in Computer Science*, pages 123–151, Madrid, Spain, May 4–8, 2025. Springer, Cham, Switzerland. doi:10.1007/978-3-031-91095-1_5. (Cited on p. 9).
- [CHW26] Henry Corrigan-Gibbs, Alexandra Henzinger, and David J. Wu. The structured generic-group model. In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026, Part V*, volume 16545 of *Lecture Notes in Computer Science*, pages 95–124, Rome, Italy, May 10–14, 2026. Springer, Cham, Switzerland. doi:10.1007/978-3-032-25330-9_4. (Cited on pp. 10 and 34).
- [CK18] Henry Corrigan-Gibbs and Dmitry Kogan. The discrete-logarithm problem with preprocessing. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 415–447, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-78375-8_14. (Cited on p. 10).
- [CS06] An Commeine and Igor Semaev. An algorithm to solve the discrete logarithm problem with the number field sieve. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *PKC 2006: 9th International Conference on Theory and Practice of Public Key Cryptography*, volume 3958 of *Lecture Notes in Computer Science*, pages 174–190, New York, NY, USA, April 24–26, 2006. Springer Berlin Heidelberg, Germany. doi:10.1007/11745853_12. (Cited on p. 34).

- [DGGM19] Nico Döttling, Sanjam Garg, Vipul Goyal, and Giulio Malavolta. Laconic conditional disclosure of secrets and applications. In David Zuckerman, editor, *60th Annual Symposium on Foundations of Computer Science*, pages 661–685, Baltimore, MD, USA, November 9–12, 2019. IEEE Computer Society Press. doi:10.1109/FOCS.2019.00046. (Cited on p. 9).
- [DGI⁺19] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 3–32, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-26954-8_1. (Cited on pp. 3, 5, 7, 9, 10, 11, 12, 13, 15, 35, 43, 44, 49, and 50).
- [DGKV22] Lalita Devadas, Rishab Goyal, Yael Kalai, and Vinod Vaikuntanathan. Rate-1 non-interactive arguments for batch-NP and applications. In *63rd Annual Symposium on Foundations of Computer Science*, pages 1057–1068, Denver, CO, USA, October 31 – November 3, 2022. IEEE Computer Society Press. doi:10.1109/FOCS54457.2022.00103. (Cited on p. 9).
- [DIJL23] Quang Dao, Yuval Ishai, Aayush Jain, and Huijia Lin. Multi-party homomorphic secret sharing and sublinear MPC from sparse LPN. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part II*, volume 14082 of *Lecture Notes in Computer Science*, pages 315–348, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-38545-2_11. (Cited on p. 9).
- [DJ01] Ivan Damgård and Mats Jurik. A generalisation, a simplification and some applications of Paillier’s probabilistic public-key system. In Kwangjo Kim, editor, *PKC 2001: 4th International Workshop on Theory and Practice in Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136, Cheju Island, South Korea, February 13–15, 2001. Springer Berlin Heidelberg, Germany. doi:10.1007/3-540-44586-2_9. (Cited on p. 15).
- [DJ24] Quang Dao and Aayush Jain. Lossy cryptography from code-based assumptions. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part III*, volume 14922 of *Lecture Notes in Computer Science*, pages 34–75, Santa Barbara, CA, USA, August 18–22, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-68382-4_2. (Cited on p. 9).
- [DKK18] Itai Dinur, Nathan Keller, and Ohad Klein. An optimal distributed discrete log protocol with applications to homomorphic secret sharing. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 213–242, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-96878-0_8. (Cited on p. 9).
- [DM13] Nico Döttling and Jörn Müller-Quade. Lossy codes and a new variant of the learning-with-errors problem. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 18–34, Athens, Greece, May 26–30, 2013. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-38348-9_2. (Cited on p. 9).

- [Fel68] William Feller. *An Introduction to Probability Theory and Its Applications*, volume 1 of *Wiley Series in Probability and Statistics*. John Wiley & Sons, New York, 3 edition, 1968. (Cited on p. 19).
- [FGK⁺10] David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 279–295, Paris, France, May 26–28, 2010. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-13013-7_17. (Cited on pp. 8 and 9).
- [FR23] Marc Fischlin and Felix Rohrbach. Searching for ELFs in the cryptographic forest. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023: 21st Theory of Cryptography Conference, Part III*, volume 14371 of *Lecture Notes in Computer Science*, pages 207–236, Taipei, Taiwan, November 29 – December 2, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-48621-0_8. (Cited on p. 9).
- [GI14] Niv Gilboa and Yuval Ishai. Distributed point functions and their applications. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 640–658, Copenhagen, Denmark, May 11–15, 2014. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-55220-5_35. (Cited on p. 9).
- [GOR11] Vipul Goyal, Adam O’Neill, and Vanishree Rao. Correlated-input secure hash functions. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 182–200, Providence, RI, USA, March 28–30, 2011. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-19571-6_12. (Cited on p. 3).
- [GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 99–108, San Jose, CA, USA, June 6–8, 2011. ACM Press. doi:10.1145/1993636.1993651. (Cited on p. 25).
- [HLOV11] Brett Hemenway, Benoît Libert, Rafail Ostrovsky, and Damien Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 70–88, Seoul, South Korea, December 4–8, 2011. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-25385-0_4. (Cited on p. 9).
- [HO13] Brett Hemenway and Rafail Ostrovsky. Building lossy trapdoor functions from lossy encryption. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 241–260, Bangalore, India, December 1–5, 2013. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-42045-0_13. (Cited on p. 9).
- [Hoe63] Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi:10.1080/01621459.1963.10500830. (Cited on p. 25).

- [Hof12] Dennis Hofheinz. All-but-many lossy trapdoor functions. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 209–227, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-29011-4_14. (Cited on p. 9).
- [HW15] Pavel Hubacek and Daniel Wichs. On the communication complexity of secure function evaluation with long output. In Tim Roughgarden, editor, *ITCS 2015: 6th Conference on Innovations in Theoretical Computer Science*, pages 163–172, Rehovot, Israel, January 11–13, 2015. Association for Computing Machinery. doi:10.1145/2688073.2688105. (Cited on p. 9).
- [IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 145–161, Santa Barbara, CA, USA, August 17–21, 2003. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-540-45146-4_9. (Cited on p. 3).
- [IL25] Rahul Ilango and Alex Lombardi. Cryptography meets worst-case complexity: Optimal security and more from iO and worst-case assumptions. In *66th Annual Symposium on Foundations of Computer Science*, pages 585–614, Sydney, Australia, December 14–17, 2025. IEEE Computer Society Press. doi:10.1109/FOCS63196.2025.00032. (Cited on p. 9).
- [ILM21] Yuval Ishai, Russell W. F. Lai, and Giulio Malavolta. A geometric approach to homomorphic secret sharing. In Juan Garay, editor, *PKC 2021: 24th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 12711 of *Lecture Notes in Computer Science*, pages 92–119, Virtual Event, May 10–13, 2021. Springer, Cham, Switzerland. doi:10.1007/978-3-030-75248-4_4. (Cited on p. 9).
- [KK04] Takeshi Koshihara and Kaoru Kurosawa. Short exponent Diffie-Hellman problems. In Feng Bao, Robert Deng, and Jianying Zhou, editors, *PKC 2004: 7th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2947 of *Lecture Notes in Computer Science*, pages 173–186, Singapore, March 1–4, 2004. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-540-24632-9_13. (Cited on p. 10).
- [LMS18] Russell W. F. Lai, Giulio Malavolta, and Dominique Schröder. Homomorphic secret sharing for low degree polynomials. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 279–309, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-030-03332-3_11. (Cited on p. 9).
- [LSSS17] Benoît Libert, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 332–364, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-63697-9_12. (Cited on p. 9).

- [Mac29] Colin MacLaurin. IV. A Second Letter from Mr. Colin McLaurin, Professor of Mathematicks in the University of Edinburgh and F. R. S. to Martin Folkes, Esq.; Concerning the Roots of Equations, with the Demonstration of Other Rules in Algebra; Being the Continuation of the Letter Published in the Philosophical Transactions, No. 394. *Philosophical Transactions of the Royal Society of London*, 36(408):59–96, 1729. [doi:10.1098/rstl.1729.0011](https://doi.org/10.1098/rstl.1729.0011). (Cited on p. 20).
- [Mau05] Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *Lecture Notes in Computer Science*, pages 1–12, Cirencester, UK, December 19–21, 2005. Springer Berlin Heidelberg, Germany. [doi:10.1007/11586821_1](https://doi.org/10.1007/11586821_1). (Cited on p. 10).
- [MMN06] Ilya Mironov, Anton Mityagin, and Kobbi Nissim. Hard instances of the constrained discrete logarithm problem. Cryptology ePrint Archive, Report 2006/253, 2006. URL: <https://eprint.iacr.org/2006/253>. (Cited on p. 10).
- [Mun77] A. G. Munford. A Note on the Uniformity Assumption in the Birthday Problem. *The American Statistician*, 31(3):119, 1977. [doi:10.1080/00031305.1977.10479214](https://doi.org/10.1080/00031305.1977.10479214). (Cited on p. 19).
- [MY10] Petros Mol and Scott Yilek. Chosen-ciphertext security from slightly lossy trapdoor functions. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 296–311, Paris, France, May 26–28, 2010. Springer Berlin Heidelberg, Germany. [doi:10.1007/978-3-642-13013-7_18](https://doi.org/10.1007/978-3-642-13013-7_18). (Cited on p. 9).
- [Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109, Santa Barbara, CA, USA, August 17–21, 2003. Springer Berlin Heidelberg, Germany. [doi:10.1007/978-3-540-45146-4_6](https://doi.org/10.1007/978-3-540-45146-4_6). (Cited on p. 25).
- [Nec94] Vassiliy Ilyich Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, February 1994. [doi:10.1007/BF02113297](https://doi.org/10.1007/BF02113297). (Cited on pp. 10, 27, and 28).
- [OPWW15] Tatsuaki Okamoto, Krzysztof Pietrzak, Brent Waters, and Daniel Wichs. New realizations of somewhere statistically binding hashing and positional accumulators. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part I*, volume 9452 of *Lecture Notes in Computer Science*, pages 121–145, Auckland, New Zealand, November 30 – December 3, 2015. Springer Berlin Heidelberg, Germany. [doi:10.1007/978-3-662-48797-6_6](https://doi.org/10.1007/978-3-662-48797-6_6). (Cited on p. 9).
- [OSY21] Claudio Orlandi, Peter Scholl, and Sophia Yakubov. The rise of paillier: Homomorphic secret sharing and public-key silent OT. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 678–708, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland. [doi:10.1007/978-3-030-77870-5_24](https://doi.org/10.1007/978-3-030-77870-5_24). (Cited on pp. 9, 17, and 40).

- [Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT’99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238, Prague, Czech Republic, May 2–6, 1999. Springer Berlin Heidelberg, Germany. doi:10.1007/3-540-48910-X_16. (Cited on pp. 15, 18, and 43).
- [Pol78] John M. Pollard. Monte Carlo Methods for Index Computation (Mod P). *Mathematics of Computation*, 32(143):918–924, July 1978. doi:10.2307/2006496. (Cited on pp. 10, 22, and 23).
- [Pol00] John M. Pollard. Kangaroos, monopoly and discrete logarithms. *Journal of Cryptology*, 13(4):437–447, September 2000. doi:10.1007/s001450010010. (Cited on pp. 10, 22, and 23).
- [PRS12] Krzysztof Pietrzak, Alon Rosen, and Gil Segev. Lossy functions do not amplify well. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 458–475, Taormina, Sicily, Italy, March 19–21, 2012. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-28914-9_26. (Cited on p. 9).
- [PS98] Sarvar Patel and Ganapathy S. Sundaram. An efficient discrete log pseudo random generator. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO’98*, volume 1462 of *Lecture Notes in Computer Science*, pages 304–317, Santa Barbara, CA, USA, August 23–27, 1998. Springer Berlin Heidelberg, Germany. doi:10.1007/BFb00055737. (Cited on pp. 10, 22, 23, and 24).
- [PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 187–196, Victoria, BC, Canada, May 17–20, 2008. ACM Press. doi:10.1145/1374376.1374406. (Cited on pp. 5, 9, 13, and 65).
- [QWW18] Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. In Mikkel Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 859–870, Paris, France, October 7–9, 2018. IEEE Computer Society Press. doi:10.1109/FOCS.2018.00086. (Cited on p. 9).
- [QWW21] Willy Quach, Brent Waters, and Daniel Wichs. Targeted lossy functions and applications. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part IV*, volume 12828 of *Lecture Notes in Computer Science*, pages 424–453, Virtual Event, August 16–20, 2021. Springer, Cham, Switzerland. doi:10.1007/978-3-030-84259-8_15. (Cited on p. 9).
- [RS21] Lawrence Roy and Jaspal Singh. Large message homomorphic secret sharing from DCR and applications. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part III*, volume 12827 of *Lecture Notes in Computer Science*, pages 687–717, Virtual Event, August 16–20, 2021. Springer, Cham, Switzerland. doi:10.1007/978-3-030-84252-9_23. (Cited on pp. 7, 9, 15, 16, 17, 35, 40, 41, 44, 45, 49, and 50).
- [Sha71] Daniel Shanks. Class Number, a Theory of Factorization, and Genera. In *1969 Number Theory Institute*, volume 20 of *Proceedings of Symposia in Pure Mathematics*, pages

- 415–440. American Mathematical Society, 1971. doi:10.1090/pspum/020/0316385. (Cited on pp. 10, 22, 23, 24, 30, 34, and 48).
- [Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT’97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer Berlin Heidelberg, Germany. doi:10.1007/3-540-69053-0_18. (Cited on pp. 10, 27, and 28).
- [vOW96] Paul C. van Oorschot and Michael J. Wiener. On Diffie-Hellman key agreement with short exponents. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT’96*, volume 1070 of *Lecture Notes in Computer Science*, pages 332–343, Saragossa, Spain, May 12–16, 1996. Springer Berlin Heidelberg, Germany. doi:10.1007/3-540-68339-9_29. (Cited on pp. 10, 22, 23, 24, and 30).
- [vzGS13] Joachim von zur Gathen and Igor E. Shparlinski. Generating Safe Primes. *Journal of Mathematical Cryptology*, 7(4):333–365, 2013. doi:10.1515/jmc-2013-5011. (Cited on p. 15).
- [Web95] Damian Weber. An implementation of the general number field sieve to compute discrete logarithms mod p . In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology – EUROCRYPT’95*, volume 921 of *Lecture Notes in Computer Science*, pages 95–105, Saint-Malo, France, May 21–25, 1995. Springer Berlin Heidelberg, Germany. doi:10.1007/3-540-49264-X_8. (Cited on p. 34).
- [Wee12] Hoeteck Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 246–262, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-29011-4_16. (Cited on p. 9).
- [YYHK16] Takashi Yamakawa, Shota Yamada, Goichiro Hanaoka, and Noboru Kunihiro. Adversary-dependent lossy trapdoor function from hardness of factoring semi-smooth RSA subgroup moduli. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 3–32, Santa Barbara, CA, USA, August 14–18, 2016. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-53008-5_1. (Cited on p. 9).
- [Zha16] Mark Zhandry. The magic of ELFs. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 479–508, Santa Barbara, CA, USA, August 14–18, 2016. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-53018-4_18. (Cited on pp. 3, 5, 6, 9, 10, 13, 19, 24, 25, 35, 51, 56, 58, and 65).
- [Zha19] Mark Zhandry. On ELFs, deterministic encryption, and correlated-input security. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 3–32, Darmstadt, Germany, May 19–23, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-17659-4_1. (Cited on pp. 3, 6, 8, 9, 13, 50, 59, 65, 67, and 74).