Nonmalleable Digital Lockers and Robust Fuzzy Extractors in the Plain Model

Daniel Apon[1(✉)], Chloe Cachet[2(✉)], Benjamin Fuller[2], Peter Hall[3], and Feng-Hao Liu[4]

[1] MITRE, McLean, VA, USA
dapon.crypto@gmail.com
[2] University of Connecticut, Mansfield, CT, USA
{chloe.cachet,benjamin.fuller}@uconn.edu
[3] New York University, New York, USA
pf2184@nyu.edu
[4] Florida Atlantic University, Boca Raton, FL, USA
liuf@fau.edu

Abstract. We give the first constructions in the plain model of 1) nonmalleable digital lockers (Canetti and Varia, TCC 2009) and 2) robust fuzzy extractors (Boyen et al., Eurocrypt 2005) that secure sources with entropy below 1/2 of their length. Constructions were previously only known for both primitives assuming random oracles or a common reference string (CRS).

Along the way, we define a new primitive called a nonmalleable point function obfuscation with associated data. The associated data is public but protected from all tampering. We use the same paradigm to then extend this to digital lockers. Our constructions achieve nonmalleability over the output point by placing a CRS into the associated data and using an appropriate non-interactive zero-knowledge proof. Tampering is protected against the input point over low-degree polynomials and over any tampering to the output point and associated data. Our constructions achieve virtual black box security.

These constructions are then used to create robust fuzzy extractors that can support low-entropy sources in the plain model. By using the geometric structure of a syndrome secure sketch (Dodis et al., SIAM Journal on Computing 2008), the adversary's tampering function can always be expressed as a low-degree polynomial; thus, the protection provided by the constructed nonmalleable objects suffices.

Keywords: Point obfuscation · Digital lockers · Nonmalleability · Virtual black box obfuscation · Fuzzy extractors

1 Introduction

The random oracle (RO) paradigm [9] allows one to analyze cryptographic primitives/protocols with an idealized random function, significantly simplifying the

© International Association for Cryptologic Research 2022
S. Agrawal and D. Lin (Eds.): ASIACRYPT 2022, LNCS 13794, pp. 353–383, 2022.
https://doi.org/10.1007/978-3-031-22972-5_13

designs and analyses. Since instantiating RO with a real-life object is impossible for the general case [23], it is important to identify useful RO properties that are achievable under specific hard problems.

Initial Efforts – Point Obfuscation. Canetti [20] initiated a study on an important property of random oracles called *oracle hashing*—or *point obfuscation*—and realized it in the plain model. More specifically, a point function I_{val} is indexed by a string val and acts as follows:

$$I_{\mathsf{val}}(\mathsf{val}') = \begin{cases} 1 & \mathsf{val} = \mathsf{val}' \\ 0 & \text{otherwise} \end{cases}.$$

An obfuscated point function should reveal nothing beyond the input/output behavior of the function $I_{\mathsf{val}}(\cdot)$. This security notion is called virtual black-box (VBB) security. Constructions are known from multiple assumptions [4,20,27,46].

VBB secure obfuscation of point functions captures the idea that the output of the RO is independent of its input, and that one can verify whether the output (for now, up to one bit) of an RO is correctly generated from a specific input. While VBB security is impossible for general functions [6], VBB secure obfuscation appears possible for point functions. (Similar techniques are used to obfuscate wildcards, conjunctions, and hyperplanes [7,12,18,24,31,39].)

Next Step – Nonmalleability. However, there are many other properties of the RO that make it a desirable object. For example, given an RO output value on input x, it should be infeasible to obtain another output of RO on any related input point (e.g., $x+1$). Applied to our setting, this is known as *nonmalleable point obfuscation*. The nonmalleability of random oracles enables many other objects that resist active attack. For example, this work considers robust fuzzy extractors [16] as an application, which were first constructed from random oracles.

Canetti and Varia [25] defined a nonmalleable point function and realized it in the common reference string (CRS) model. However, as one of the most valuable properties of the RO is that no trusted setup is required, an ideal instantiation would not require a CRS.

To tackle this, Komargodski and Yogev [44] proposed a construction of a nonmalleable point obfuscation in the plain model.[1] Prior work in plain model point obfuscation considers a limiting tampering class of low-degree polynomials where the degree relates to the hardness of the underlying number-theoretic assumption.

Another Step Forward – Digital Lockers. An obfuscated point function only outputs one bit. However, we are generally interested in the RO outputting a random string for a given input. To emulate this functionality, a natural

[1] Unfortunately, their underlying cryptographic assumption was broken by Bartusek, Ma, and Zhandry [8]. An alternative assumption was posed in [45], but this did not suffice to show security. Fortunately, Bartusek, Ma, and Zhandry introduced their own assumption and accompanying construction, showing their assumption holds in a strong variant of the generic group model [8].

extension is the *multi-bit* point function, where each function $I_{\mathsf{val},\mathsf{key}}$ is indexed by a pair of strings $(\mathsf{val}, \mathsf{key})$ and works as follows:

$$I_{\mathsf{val},\mathsf{key}}(\mathsf{val}') = \begin{cases} \mathsf{key} & \mathsf{val} = \mathsf{val}' \\ \perp & \text{otherwise} \end{cases}.$$

An obfuscation of a function of this class is called a *digital locker*, which is useful in password [20] and biometric authentication [1,22].

Though we know how to build digital lockers in the plain model [21], the only existing nonmalleable constructions require a CRS. Fenteany and Fuller [38] achieved half of the goal, constructing a digital locker that nonmalleabile against tampering only on val in the standard model. However, while the work [38] pointed out a technique to additionally protect key, it required a CRS, similarly to the original work [25]. As an ideal instantiation of RO does not require a trusted setup, this naturally motivates our main question:

Can one build a nonmalleable digital locker in the plain model without setup?

Our Technical Contributions We answer the main question in the affirmative, constructing a nonmalleable digital locker in the plain model. We present the following contributions:

1. **Point Obfuscation with Associated Data** We define a new primitive called a *nonmalleable point obfuscator with associated data*. We then instantiate this object using group assumptions introduced by Bartusek, Ma, and Zhandry [8].
2. **Creating a Multibit Output** We then integrate this construction with the real or random construction [21], yielding a nonmalleable digital locker that prevents tampering on the input and associated data only. This step is not black box in the point obfuscations. Instead, it is created from scratch using similar techniques from the same group assumptions as the constructed point obfuscation.
3. **Protecting the Multibit Output** By putting the CRS of a true simulation extractable non-interactive zero-knowledge proof (NIZK) [32] into the associated data, we can protect the output of the digital locker. Conceptually, our new tool protects the NIZK $\mathtt{crs}$, which (if intact) can be used to derive nonmalleability for the other parts of the construction. This step is black box from an appropriate variant of a digital locker.

In all of the above steps, the prevented tampering class for the input point, val is low-degree polynomials, rather than the desired complete tamper resistance. However, this class is still meaningful in many applications where a RO was previously used.

1.1 Low Entropy Robust Fuzzy Extractors in the Plain Model

Despite a limited tampering class, our nonmalleable objects suffice to construct the first plain model robust fuzzy extractors [16] that support sources whose

entropy is less than half their length 1/2, a known barrier for information-theoretically secure constructions [35]. We notice that all prior computationally secure constructions relied on some form of a CRS, and our work shows that this component is not required.

A fuzzy extractor is a pair of algorithms $(\mathsf{Gen}, \mathsf{Rep})$ with two properties:

Correctness. Let w, w' be values that are close in some distance metric, and define $(\mathsf{key}, \mathsf{pub}) \leftarrow \mathsf{Gen}(w)$. Then it is true that $\mathsf{Rep}(w', \mathsf{pub}) = \mathsf{key}$.

Security. The value key is computationally indistinguishable from a uniform value given pub.

Digital lockers have been used to construct reusable fuzzy extractors, as in [50] [1,22], i.e., one can derive multiple keys from the same entropy source. An additional desirable property is robustness [33], which prevents an adversary from modifying pub in an attempt to force Rep to produce a different key.

Robust fuzzy extractors are notoriously difficult to construct – we show various limitations of the prior constructions in Table 1. Dodis and Wichs [35] showed that it is only possible information-theoretically if the entropy of w is at least half its length. Feng and Tang [37] showed this barrier exists in the CRS model, as well. Feng and Tang construct a robust fuzzy extractor with computational security for entropy sources that can depend on the CRS.

We construct the first robust fuzzy extractor in the plain model that supports entropy for w that is less than half its length. We combine our nonmalleable digital locker with a specific error-correction component, the syndrome construction [11,30,34]. The syndrome construction allows the reduction to extract a low-degree polynomial that is consistent with the adversary's tampering. Similar techniques were used to construct CRS model robust fuzzy extractors from algebraic-manipulation detection codes [29]. We present a second construction directly from the nonmalleable point function from associated data which is able to extract a limited length key.[2]

To the best of our knowledge, our work and that of Cramer et al. [29] are the only two approaches to building a robust fuzzy extractor that do not build a robust extractor first. This is because our nonmalleable tools only prevent limited tampering classes; both works use the secure sketch component to guarantee the adversary's tampering is in this low complexity class.

1.2 Technical Overview

In this section, we present an overview of our techniques. In the CRS model, nonmalleability of point functions can be achieved as [25], by using a nonmalleable NIZK system – in addition to generating a regular $C \leftarrow \mathsf{DL}(I_{\mathsf{val},\mathsf{key}})$, one also appends a zero-knowledge proof π to the output showing knowledge of the pair $(\mathsf{val}, \mathsf{key})$ inside C. However, any non-trivial nonmalleable NIZK system would require a trusted (nontamperable) CRS for security of the proof system, so the

[2] We also show that a nonmalleable point function (without associated data) suffices to construct a robust secure sketch [34].

Table 1. Comparison of Robust Fuzzy Extractors. The CRS* model means that the distribution of W can depend on the CRS, however, the CRS is still assumed not to be modified. For a distribution W, $\mathrm{H}_\infty(W)$ represents min-entropy (see Sect. 2) and $|W|$ represents its length. IT corresponds to information-theoretic security and Comp. represents security against computationally bounded adversaries. Syn. is the syndrome or null space of an appropriate error correcting code. The column of SS errors indicates the error tolerance of the underlying secure sketch. This parameter is related to the information leakage of the secure sketch. This work and prior computational works require a secure sketch that corrects $2t$ errors, which leads to more leakage.

Scheme	Model	Security	SS errors	$\mathrm{H}_\infty(W) < \|W\|/2$?
[14,15]	RO	IT	t	✓
[33]	Plain	IT	t	X
[29]	CRS	IT	t	X
[53–55]	CRS	Comp	$2t$	✓
[37]	CRS*	Comp	$2t$	✓
Syn. + NM Point Obf w/Assoc. Data	Plain	Comp	$2t$	✓
Syn. + NM Digital Locker	Plain	Comp	$2t$	✓

overall obfuscation would be $(\mathtt{crs}, C, \pi)$. Without trusted setup, an adversary may simply replace the $\mathtt{crs}$, rendering the NIZK ineffective and breaking nonmalleability. So, this trusted setup required immediately fails at achieving our goal.

Point Obfuscation with Associated Data To achieve our goal, we formalize a notion that blends any public string with point obfuscation in a meaningful way, called *point obfuscation with associated data*. More specifically, the obfuscator $\mathsf{Obf}(I_{\mathsf{val}}, \mathtt{ad})$ takes as input the point function I_{val} and an additional public string $\mathtt{ad}$ (e.g., $\mathtt{crs}$) and then outputs an obfuscated program C along with $\mathtt{ad}$. The output program C should be VBB secure, and $\mathtt{ad}$ is treated as public information.

We formulate nonmalleability properties that treat the two inputs quite differently. The adversary outputs $(C', \mathtt{ad}', f)$ and wins if C' is consistent with the values $f(\mathsf{val})$ and $\mathtt{ad}'$, and one of the following hold:

1. The function belongs to some targeted function class, i.e., $f \in \mathcal{F}$, or
2. The function f is the identity and $\mathtt{ad}' \neq \mathtt{ad}$.

Nonmalleability requires that the adversary has only a negligible winning probability, meaning that they cannot replace $\mathtt{ad}$ by any other string, nor tamper val consistently by any function in the class $\mathcal{F}$.

Remark 1. It is undesirable that in the definition the adversary output their tampering function. The desired notion is that the adversary cannot output $(C', \mathtt{ad}')$ that is consistent with any f. This notion is impossible to achieve in the plain model if f contains linear shifts. Essentially, given an obfuscation of

point x, an adversary not required to output their mauling function may simply create an obfuscation of independent point y. It is clear that if the function $f(z) = z - x + y$ is in $\mathcal{F}$, this would be a valid tampering, but it is impossible to prevent without requiring the adversary shows some awareness of its specific tampering. The definition where the adversary chooses and outputs f after seeing C does imply that all fixed functions f are prevented [45]. See the full version for more details [3, Appendix A].

How to Construct this Object. Before instantiating such an object, we recall some notations and related constructions of nonmalleable point obfuscations in prior work [8,38,44]. We note that all of these constructions rely on groups that only efficiently admit linear operations.

Suppose that g is a generator of a prime order group whose order is p. Throughout this paper, $[x]_g$ will be used to represent g^x (called implicit notation in [36]) so as to highlight the behavior in the exponent. We treat val as an element in $\mathbb{Z}_p$. Let the class of tampering functions $\mathcal{F}$ correspond to low degree polynomials over $\mathbb{Z}_p$. Previous constructions [8] use a set of polynomial encodings, denoted as $\mathcal{P}$, and compute the following for $\mathsf{Obf}(I_{\mathsf{val}})$:

1. Sample some $P \leftarrow \mathcal{P}$,
2. Output $P, [P(\mathsf{val})]_g$.

The intuition for security[3] is twofold: 1) that $\mathcal{P}$ is sufficiently randomized to argue virtual black box security [6], and that 2) for all instances of $P \in \mathcal{P}$ no fixed affine functions of P - i.e., $\alpha P(\mathsf{val}) + \beta$ - correspond to any $P'(f(\mathsf{val}))$ for $P' \in \mathcal{P}$ and low degree polynomial f. Prior work achieves these two properties jointly by randomizing the low degree coefficients of P and fixing some higher powers to have a coefficient of 1. For example, Bartusek et al. [8] consider $P_a(x) = ax + x^2 + x^3 + x^4 + x^5$.

Our construction builds such a function class $\mathcal{P}$, parameterizing $P \in \mathcal{P}$ by both a random a and ad, so that ad and val can be blended in a secure way. Let $\rho := |\mathsf{ad}|$. Then, we have:

$$P_{a,\mathsf{ad}}(x) \stackrel{\text{def}}{=} ax + \sum_{i=2}^{\rho+1} \mathsf{ad}_i x^i + \sum_{i=\rho+2}^{\rho+6} x^i.$$

In the above, the random a corresponds to the lowest degree coefficient of P and the bits of ad set intermediate coefficients of the polynomial P. We can prove security using the same group assumption used in prior nonmalleable point obfuscation works [8,38].

While the construction has a similar structure to prior work, analysis of nonmalleability is significantly more complicated by the fact that the adversary

[3] The actual constructions are more complicated to ensure correctness holds, using other points of randomness and group elements to check correctness. These are not used in arguing nonmalleability. For simplicity, we do not discuss correctness in this section.

Algorithm 1: Augmented real-or-random construction that provides nonmalleability over input point and associated data. Obf is an obfuscator and NMObf is a nonmalleable obfuscator.

Input $(\mathsf{val}, \mathsf{key}, \mathtt{ad})$.
Sample random z.
for *each bit* i *of* key **do**
  if $\mathsf{key}_i = 1$ **then**
    $C_i \leftarrow \mathsf{Obf}(I_{\mathsf{val}})$
  else
    $C_i \leftarrow \mathsf{Obf}(I_z)$
  end
end
Let $C_0 \leftarrow \mathsf{NMObf}(I_{\mathsf{val},\mathtt{ad}})$. `// To distinguish an all-zero key and provide nonmalleability`
Output $C = (C_0, C_1, \ldots, C_{|\mathsf{key}|}, \mathtt{ad})$.

1) knows ad, 2) can output any value for ad′, and 3) doesn't have to explain how ad′ arose from ad. This gives the adversary more flexibility, and proving nonmalleability becomes a careful multi-step procedure.

To give some intuition for the algebraic structure, it is important that the powers multiplied by the bits of ad are below the powers with coefficients 1. If these were switched, one could apply a polynomial tampering function to x and change the associated data to compensate for the resulting changes in the higher powers.

Extending to the Multibit Setting. Next, we integrate the above with the real-or-random approach of Canetti and Dakdouk [21]. The modified algorithm is summarized in Algorithm 1.

On the technical side, this approach requires the polynomials in the group to have more randomized powers, similar to the prior work of Fenteany and Fuller [38]. However, unlike their work, we only use one nonmalleable point obfuscation, the rest simply provide privacy. That is, only C_0 in Algorithm 1 is nonmalleable. As we show, this is sufficient to ensure nonmalleability over the resultant digital locker.

Protecting the Multibit Output. The above instantiation of the real-or-random construction prevents tampering of the input point and associated data but provides no protection over key. Our protection of the associated data allows us to upgrade the NIZK construction of [25] to the plain model. Our technique protects the associated data, which is set as crs, and the security of NIZK protects everything else, so long as crs cannot be tampered with. As we discuss in Sect. 5.3, we are also able to use a weaker NIZK system, specifically true simulation extractible NIZKs, which may be instantiable in pairing-free groups.

1.3 Discussion and Open Questions

This work presents the first constructions in the plain model of nonmalleable digital lockers and low-entropy robust fuzzy extractors. The integration of the nonmalleable point function with associated data with the real-or-random construction is technical and non-black box. Ideally, one would be able to define some necessary condition such that general black box composition of our point obfuscation with associated data and any other point obfuscation or digital locker is possible. One can view our construction as evidence that our particular nonmalleable point obfuscator with associated data is safe under composition with a specific point function.

There are known barriers to constructing digital lockers secure against auxiliary data that is hard to invert (such as a point function) if indistinguishability obfuscation exists [10,19]. Security in the presence of auxiliary data is the standard method for arguing composition.

In this work, we focus on nonmalleability of digital lockers. Obfuscating wildcards, conjunctions, and hyperplanes use similar techniques [7,12,18,24,31,39], so our techniques may apply. We note that some of these objects directly yield non-robust fuzzy extractors [31,39], so it may be possible to provide robustness by making the obfuscation nonmalleable. It seems less likely the techniques can be used to protect obfuscation of general evasive functions [5], compute-and-compare programs [17,43,56] and general obfuscation [2,40–42,48,49].

We generically use (true simulation extractible) NIZKs. Optimizing this construction is important, since this object will likely represent the dominant computational cost.

2 Preliminaries

Logarithms are base 2. Let $X_i \in \mathcal{Z}$ be random variables. We denote by $X = X_1, ..., X_n$ the tuple $(X_1, \ldots, X_n)$. For a discrete random variable X, the *min-entropy* of X is $\mathrm{H}_\infty(X) = -\log(\max_x \Pr[X = x])$. For a pair of discrete random variables X, Y, the average min-entropy of $X|Y$ is

$$\tilde{\mathrm{H}}_\infty(X|Y) = -\log\left(\mathop{\mathbb{E}}_{y \in Y}\left(2^{-\mathrm{H}_\infty(X|Y)}\right)\right).$$

The notation id is used to denote the identity function: $\forall x, \mathsf{id}(x) = x$. Capitalized letters are used for random variables and lowercase letters for samples. Let $\{\mathsf{D}_\lambda\}$ be an ensemble of sets. Two circuits, C and C', with inputs in D^λ are *functionally equivalent*, denoted $C \equiv C'$, if $\forall x \in \mathsf{D}^\lambda, C(x) = C'(x)$. For a matrix $\mathbf{A}$, let $\mathbf{A}_i$ denote the ith row and $\mathbf{A}_{i,j}$ to denote the entry in the i row and jth column.

Definition 1. *An ensemble of distributions $\mathcal{X} = \{X_\lambda\}_{\lambda \in \mathbb{N}}$, where $\mathcal{X}_\lambda$ is over D^λ, is* well-spread *if the function $\mathrm{H}_\infty(X_\lambda)$ mapping λ to non negative reals grows faster than $\omega(\log \lambda)$. That is, $\mathrm{H}_\infty(X_\lambda) = \omega(\log \lambda)$.*

Definition 2. *An ensemble of distributions* $\mathcal{X} = \{X_\lambda\}_{\lambda\in\mathbb{N}}$*, where* $\mathcal{X}_\lambda$ *is over* D^λ*, is* efficiently sampleable *if exists a PPT algorithm given* 1^λ *as input whose output is identically distributed as* X_λ*.*

Throughout this work, we will use λ to represent the security parameter, ρ to represent the length of the associated data, ℓ to represent the length of the output key, and τ to represent the maximum degree of the polynomial the adversary uses for mauling.

3 Obfuscation Definitions

All obfuscation definitions include require only polynomial slowdown, which is easily verifiable for all presented constructions. The main object we introduce in this work is a nonmalleable point function with associated data. A traditional point function $I_{\mathsf{val}} : \mathbb{Z}_p \mapsto \{0,1\}$ takes a single input $\mathsf{val} \in \mathbb{Z}_p$ and returns 1 if and only if the input x to the function is val. An obfuscator is designed to preserve this functionality while hiding val. The definition of a nonmalleable point function with associated data adds a second input to I denoted as $\mathsf{ad} \in \{0,1\}^\rho$. This input does not need to be hidden by the obfuscator but should be nonmalleable. So the raw functionality is just a point function of the pair $\mathsf{val}, \mathsf{ad}$. That is,

$$I_{\mathsf{val},\mathsf{ad}}(x,y) = \begin{cases} 1 & x = \mathsf{val} \wedge y = \mathsf{ad} \\ 0 & \text{otherwise.} \end{cases}$$

Note that, since in our use cases ad is public, an honest user may just use the given ad in using the obfuscated point function. In our further sections, we use $\mathsf{lockPoint}(\cdot)$ to denote point obfuscation algorithm and $\mathsf{unlockPoint}$ as the obfuscated program. As prior work [25,38,44], we first present the notion of an obfuscation verifier:

Definition 3 (Obfuscation Verifier). *Let* $\lambda \in \mathbb{N}$ *be a security parameter and let* $\mathcal{O}$ *input* $x \in \mathsf{D}^\lambda$ *and output a program* $\mathcal{P}$*. An algorithm* $\mathsf{V}_{\mathtt{obf}}$ *is a value verifier if* $\forall x \in \mathsf{D}^\lambda$ *it is true that* $\Pr_{\mathsf{V}_{\mathtt{obf}},\mathcal{O}}[\mathsf{V}_{\mathtt{obf}}(\mathcal{P}) = 1 | \mathcal{P} \leftarrow \mathcal{O}(x)] = 1$*.*

Definition 4 (Nonmalleable Point Function with Associated Data). *For security parameter* $\lambda \in \mathbb{N}$ *parameter* $\rho \in \mathbb{N}$*, let* D_λ *be a sequence of input domains and* $\mathcal{F} : \mathsf{D}_\lambda \to \mathsf{D}_\lambda$ *be a family of functions. Let* $\mathcal{X}$ *be a family of distributions over* D_λ*. A* $(\mathcal{F}, \mathcal{X}, \rho)$-nonmalleable point function obfuscation with associated data $\mathsf{lockPoint}$ *is a PPT algorithm that inputs a point* $\mathsf{val} \in \mathsf{D}_\lambda$ *and* $\mathsf{ad} \in \{0,1\}^\rho$*, and outputs a circuit* $\mathsf{unlockPoint}$*. Let* $\mathsf{V}_{\mathtt{obf}}$ *be an obfuscation verifier for* $\mathsf{lockPoint}$ *as defined in Definition 3. The following properties must hold:*

1. ***Completeness****: For all* $\mathsf{val} \in \mathsf{D}^\lambda, \mathsf{ad} \in \{0,1\}^\rho$*, it holds that*

$$\Pr[\mathsf{unlockPoint}(\cdot,\cdot) \equiv I_{\mathsf{val},\mathsf{ad}}(\cdot,\cdot) | \mathsf{unlockPoint} \leftarrow \mathsf{lockPoint}(\mathsf{val},\mathsf{ad})] \geq 1 - \mathtt{ngl}(\lambda),$$

where the probability is over the randomness of $\mathsf{lockPoint}$*.*

2. ***Virtual Black Box Security****: For every PPT $\mathcal{A}$ and any polynomial function p, there exists a simulator $\mathcal{S}$ and a polynomial function $q(\cdot)$ such that, for all large enough $\lambda \in \mathbb{N}$, all $\mathsf{val} \in \mathsf{D}^\lambda, \mathsf{ad} \in \{0,1\}^\rho$ and for any predicate $\mathcal{P} : \mathsf{D}^\lambda \times \{0,1\}^\rho \mapsto \{0,1\}$,*

$$\begin{aligned} &|\Pr[\mathcal{A}(\mathsf{unlockPoint}, \mathsf{ad}) = \mathcal{P}(\mathsf{val}, \mathsf{ad}) | \mathsf{unlockPoint} \leftarrow \mathsf{lockPoint}(\mathsf{val}, \mathsf{ad})] \\ &- \Pr[\mathcal{S}^{I_{\mathsf{val},\mathsf{ad}}(\cdot)}(1^\lambda, \mathsf{ad}) = \mathcal{P}(\mathsf{val}, \mathsf{ad})]| \leq \frac{1}{p(\lambda)}, \end{aligned}$$

where S is allowed $q(\lambda)$ oracle queries total to $I_{\mathsf{val},\mathsf{ad}}$ and the probabilities are over the internal randomness of $\mathcal{A}$ and lockPoint*, and of $\mathcal{S}$, respectively. Here $I_{\mathsf{val},\mathsf{ad}}(\cdot)$ is an oracle that returns 1 when provided input $(\mathsf{val}, \mathsf{ad})$ and 0 otherwise.*

3. ***Nonmalleability:*** *For any $X \in \mathcal{X}$, for all $\mathsf{ad} \in \{0,1\}^\rho$, for any PPT $\mathcal{A}$, there exists $\epsilon = \mathtt{ngl}(\lambda)$, such that defining*

$$\begin{aligned} \mathsf{unlockPoint} &\leftarrow \mathsf{lockPoint}(\mathsf{val}, \mathsf{ad}), \\ (C, f, \mathsf{ad}^*) &\leftarrow \mathcal{A}\,(\mathsf{unlockPoint}, \mathsf{ad}) \end{aligned}$$

it is true that :

$$\Pr_{\mathsf{val} \leftarrow X}\left[\begin{array}{l} \mathsf{V}_{\mathtt{obf}}(C) = 1, (I_{f(\mathsf{val}),\mathsf{ad}^*} \equiv C) \\ f \in \mathcal{F} \ \vee (f = \mathsf{id} \wedge \mathsf{ad}^* \neq \mathsf{ad}) \end{array}\right] \leq \epsilon.$$

3.1 Nonmalleable Digital Locker

We recall the definition of a nonmalleable digital locker. To distinguish this from the case of point obfuscation, we use $\mathsf{lock}()$ to denote the multi-bit point obfuscation algorithm and unlock as the (obfuscated) digital locker. In our construction, all tampering of the output key is prevented, so we remove the notion of a key verifier that was used in [38].

Definition 5 (Nonmalleable Digital Locker). *For security parameter $\lambda \in \mathbb{N}$, let D_λ be a sequence of domains, let*

1. *$\mathcal{F} : \mathsf{D}_\lambda \to \mathsf{D}_\lambda$ be a function family,*
2. *$\mathcal{X}$ be a family of distributions over D^λ,*
3. *lock be a PPT algorithm that maps points $\mathsf{val} \in \mathsf{D}^\lambda, \mathsf{key} \in \{0,1\}^n$ to a circuit* unlock*, and*
4. *$\mathsf{V}_{\mathtt{obf}}$ be an obfuscation verifier.*

The algorithm lock *is a $(\mathcal{F}, \mathcal{X}, n)$-nonmalleable* digital locker *if all of the below are satisfied:*

1. ***Completeness*** *For all $\mathsf{val} \in \mathsf{D}^\lambda, \mathsf{key} \in \{0,1\}^n$ it holds that*

$$\Pr[\mathsf{unlock}(\cdot) \equiv I_{\mathsf{val},\mathsf{key}}(\cdot) | \mathsf{unlock} \leftarrow \mathsf{lock}(\mathsf{val}, \mathsf{key})] \geq 1 - \mathtt{ngl}(\lambda),$$

where the probability is over the randomness of lock*. Here $I_{\mathsf{val},\mathsf{key}}$ is a function that returns* key *when provided input* val*, otherwise $I_{\mathsf{val},\mathsf{key}}$ returns $\perp$.*

2. ***Virtual Black Box Security***: *For all PPT* $\mathcal{A}$ *and* $p = \mathtt{poly}(\lambda)$, $\exists \mathcal{S}$ *and* $q(\lambda) = \mathtt{poly}(\lambda)$ *such that for all large enough* $\lambda \in \mathbb{N}$, $\forall \mathsf{val} \in \mathsf{D}_\lambda, \mathsf{key} \in \{0,1\}^n, \mathcal{P} : \mathsf{D}_\lambda \times \{0,1\}^n \mapsto \{0,1\}$,

$$\left|\Pr[\mathcal{A}(\mathsf{lock}(\mathsf{val},\mathsf{key})) = \mathcal{P}(\mathsf{val},\mathsf{key})] - \Pr[\mathcal{S}^{I_{\mathsf{val},\mathsf{key}}}(1^\lambda) = \mathcal{P}(\mathsf{val},\mathsf{key})]\right| \leq \frac{1}{p(\lambda)},$$

where S *is allowed* $q(\lambda)$ *oracle queries to* $I_{\mathsf{val},\mathsf{key}}$ *and the probabilities are over the internal randomness of* $\mathcal{A}$ *and* lock*, and of* $\mathcal{S}$*, respectively.*

3. ***Nonmalleability*** $\forall X \in \mathcal{X}, PPT\ \mathcal{A}, \mathsf{key} \in \{0,1\}^n$*, there exists* $\epsilon = \mathtt{ngl}(\lambda)$ *such that:*

$$\Pr_{\mathsf{val} \leftarrow X}\left[\begin{array}{r|r} \mathsf{V}_{\mathtt{obf}}(C) = 1, & \mathsf{unlock}_{\mathsf{val},\mathsf{key}} \leftarrow \mathsf{lock}(\mathsf{val},\mathsf{key}) \\ \left(\begin{array}{r} (f \in \mathcal{F} \wedge \mathsf{key}' \neq \perp) \vee \\ (\mathsf{key}' \notin \{\perp, \mathsf{key}\} \wedge f = \mathsf{id}) \end{array}\right) & \begin{array}{r} (C,f) \leftarrow \mathcal{A}(\mathsf{unlock}_{\mathsf{val},\mathsf{key}}) \\ \mathsf{key}' \leftarrow C(f(\mathsf{val})) \end{array} \end{array}\right] \leq \epsilon.$$

recall id *is the identity function.*

Remark 2. As mentioned in the Introduction, there are alternative notions of nonmalleability. We formally define fixed nonmalleability, a weaker definition which was used in [25], and oblivious nonmalleability, which does not require the adversary to output the targeted function f in the full version [3, Appendix A]. There we show that oblivious nonmalleability is impossible in general. One can bypass this result by using cryptographic tools that extract the tampering function, such as a random oracle or non-falsifiable assumptions.

3.2 Same Point Definitional Equivalences

The soundness in Definitions 4 and 5 are virtual black box security [6]. In the majority of this work, we will be using distributional indistinguishability, which says that obfuscations of all well spread distributions X are indistinguishable from obfuscations of random points. Bitanski and Canetti [13] showed that this definition is equivalent to virtual black box obfuscation for point functions (see also [20,52]). Furthermore, they showed this equivalence holds when given a constant number of obfuscations on related points. Fenteany and Fuller [38] show that this equivalence holds if given a polynomial number of copies $\mathsf{unlockPoint}_1 \leftarrow \mathsf{lockPoint}(X), ..., \mathsf{unlockPoint}_\ell \leftarrow \mathsf{lockPoint}(X)$ as long as the same value is locked in each call to $\mathsf{lockPoint}$. In the full version [3], we generalize these results showing that a vector of obfuscations that have output on a single input point are secure when composed with associated data. That is, define the circuit class

$$\mathbf{Point}_{\mathsf{val},\mathsf{key},\mathsf{ad}}(\mathsf{val}', \mathsf{ad}') = \begin{cases} \mathsf{key} & \mathsf{val}' = \mathsf{val} \wedge \mathsf{ad}' = \mathsf{ad} \\ \perp & \text{otherwise} \end{cases}.$$

Note that point functions and digital lockers both with and without associated data variants fall into this class by adjusting whether $\mathtt{ad}$ and key are of length 0. These proofs are straightforward extensions of the proofs in [38]. There are presented for completeness in the full version of this work [3].

Definition 6 (Distributional Indistinguishability). *A* **Point** *obfuscator is called a good distributional indistinguishable (DI) obfuscator if for any PPT $\mathcal{A}$ with binary output and any well-spread distribution $\mathcal{X}$ over points in D^λ, for all vectors $\vec{\mathsf{key}}, \vec{\mathsf{ad}}$ then there exists some negligible function ϵ such that*

$$\begin{aligned} &|\Pr_{\mathsf{val}\leftarrow\mathcal{X}}[\mathcal{A}(\{C_i, \vec{\mathsf{ad}}_i\}_{i=1}^{\ell}) = 1 | \{C_i \leftarrow \mathbf{Point}(\mathsf{val}, \vec{\mathsf{ad}}_i, \vec{\mathsf{key}}_i)\}_{i=1}^{\ell}] \\ &- \Pr_{u\xleftarrow{\$}\mathsf{D}^\lambda}[\mathcal{A}(\{C_i, \vec{\mathsf{ad}}_i\}_{i=1}^{\ell}) = 1 | \{C_i \leftarrow \mathbf{Point}(u, \vec{\mathsf{ad}}_i, \vec{\mathsf{key}}_i)\}_{i=1}^{\ell}]| \leq \epsilon. \end{aligned}$$

Theorem 1. *For the class* **Point** *under $\ell = \mathtt{poly}(\lambda)$ composition where the same* val *is used in each obfuscation, distributional indistinguishability and virtual black box security (in Definition 4) are equivalent.*

3.3 Group Theoretic Assumptions

We present our underlying group-theoretic assumptions here. As a reminder, we use the implicit notation [36] to denote encoding in a group with generator g (where $[x]_g$ denotes g^x).

Assumption 1 [8, Assumption 3]**.** *Fix some $\psi \in \mathbb{Z}^+$. Let $\mathcal{G} = \{\mathbb{G}_\lambda\}_{\lambda\in\mathbb{N}}$ be a group ensemble with efficient representation and operations where each $\mathbb{G}_\lambda$ is a group of prime order $p(\lambda) \in (2^\lambda, 2^{\lambda+1})$. Let $\{\mathcal{X}_\lambda\}$ be a family of well-spread distributions over $\mathsf{D}^\lambda = \mathbb{Z}_{p(\lambda)}$. Then for any PPT $\mathcal{A}$:*

$$\left|\Pr[\mathcal{A}(\{k_i, [k_i x + x^i]_g\}_{i\in[2,\ldots,\psi]} = 1] - \Pr[\mathcal{A}(\{k_i, [k_i r + r^i]_g\}_{i\in[2,\ldots,\psi]}\right| = \mathtt{ngl}(\lambda).$$

where $x \leftarrow \mathcal{X}_\lambda, r \leftarrow \mathbb{Z}_{p(\lambda)}, k_i \leftarrow \mathbb{Z}_{p(\lambda)}$.

Bartusek, Ma, and Zhandry [8] justified Assumption 1 by showing it holds in the generic group model even if $\mathcal{X}_\lambda$ depends on g. This model of allowing a distribution to depend on g is related to the non-uniform generic group model [26]. Such an assumption is crucial to arguing plain model security (rather than treating $\mathcal{X}_\lambda$ as independent of g). The second assumption can be proved from Assumption 1, see [8, Lemma 8], and is useful for arguing nonmalleability:

Assumption 2 [8, Assumption 4]**.** *Fix some $\psi \in \mathbb{Z}^+$. Let $\mathcal{G}$ and $\mathcal{X}_\lambda$ be defined as in Assumption 1. For any PPT $\mathcal{A}$,*

$$\Pr[[x]_g \leftarrow \mathcal{A}(\{k_i, [k_i x + x^i]_g\}_{i\in[2,..,\psi]})] = \mathtt{ngl}(\lambda).$$

where $x \leftarrow \mathcal{X}_\lambda$ and $k_i \leftarrow \mathbb{Z}_{p(\lambda)}$.

4 Nonmalleable Point Functions with Associated Data

We begin by instantiating a nonmalleable point obfuscation satisfying Definition 4.

Construction 1. *Let $\lambda \in \mathbb{N}$ be a security parameter, let $\rho \in \mathbb{N}$ be a parameter. Let $\mathcal{G} = \{\mathbb{G}_\lambda\}_{\lambda\in\mathbb{N}}$ be a group ensemble with efficient representation and operations where each $\mathbb{G}_\lambda$ is a group of prime order $p(\lambda) \in (2^\lambda, 2^{\lambda+1})$. Define five polynomials $p_1, ..., p_5$ as follows:*

$$\begin{aligned} p_{1,\mathsf{ad},c_1}(\mathsf{val}) &= c_1\mathsf{val} + \sum_{i=1}^{\rho} \mathsf{ad}_i\mathsf{val}^{i+1} + \sum_{i=\rho+2}^{\rho+6} \mathsf{val}^i, \\ p_{2,c_2}(\mathsf{val}) &= c_2\mathsf{val} + \mathsf{val}^{\rho+7}, \\ p_{3,c_3}(\mathsf{val}) &= c_3\mathsf{val} + \mathsf{val}^{\rho+8}, \\ p_{4,c_4}(\mathsf{val}) &= c_4\mathsf{val} + \mathsf{val}^{\rho+9}, \\ p_{5,c_5}(\mathsf{val}) &= c_5\mathsf{val} + \mathsf{val}^{\rho+10}. \end{aligned}$$

In the above, all calculations are conducted modulo $\mathbb{Z}_{p(\lambda)}$.

Let g be a generator of the group $\mathbb{G}_\lambda$. Let $c_1, c_2, c_3, c_4, c_5 \xleftarrow{\$} \mathbb{Z}_{p(\lambda)}$ be input randomness. Let $\rho \stackrel{def}{=} |\mathsf{ad}|$. Consider the following construction:

$$\mathsf{lockPoint}(\mathsf{val}, \mathsf{ad}; a, b, c) \stackrel{def}{=} \begin{pmatrix} c_1, & [p_{1,\mathsf{ad},c_1}(\mathsf{val})]_g \\ c_2, & [p_{2,c_2}(\mathsf{val})]_g \\ c_3, & [p_{3,c_3}(\mathsf{val})]_g \\ c_4, & [p_{4,c_4}(\mathsf{val})]_g \\ c_5, & [p_{5,c_5}(\mathsf{val})]_g \end{pmatrix}$$

$\mathsf{V}_{\mathtt{obf}}$ *is the circuit that checks that* unlockPoint *consists of the appropriate number of values and group elements. If not, it outputs* 0*. Given a program* unlockPoint *consisting of five pairs $\{(c'_i, g'_i)\}_{i=1}^5$* [4] *and inputs val', ad' compute:*

$$\begin{gathered} \left[p_{1,\mathsf{ad}',c'_1}(\mathsf{val}')\right]_g \stackrel{?}{=} g'_1, \\ \left\{\left[p_{i,c'_i}(\mathsf{val}')\right]_g \stackrel{?}{=} g'_i\right\}_{i=2}^5. \end{gathered}$$

If all of these checks pass, output 1*. Otherwise, output* 0*.*

Theorem 2. *Let all parameters be as in Construction 1, let $\rho \in \mathbb{N}$ be a parameter. Define $\mathcal{F} : \mathbb{Z}_{p(\lambda)} \to \mathbb{Z}_{p(\lambda)}$ as the set of non-constant, non-identity polynomials of maximum power τ. Suppose that*

1. *Assumption 1 holds for $\psi = \max\{\tau(\rho+6), \rho+10\}$ and*
2. *$(\rho+6)2^{2\rho}/p(\lambda)^3 = \mathtt{ngl}(\lambda)$.*

Then, Construction 1 is a $(\mathcal{F}, \mathcal{X}, \rho)$-nonmalleable point function obfuscation with associated data.

[4] g is a generator that is a global system parameter along with the group description. Note that it is efficiently checkable 1) whether the order of a group is prime and 2) whether an element g is a generator of the known order group.

Remark 3. In the above, the size of associated data is limited to be $\rho \approx \log(p(\lambda))$, which is linear in the security parameter λ. Our primary application has the associated data as the CRS of some NIZK. Such strings can be quite long. In Sect. 5.4, we show that it suffices to include a short value in the associated data whose size is $\Theta(\log \lambda)$.

In order to prove that Construction 1 satisfies Definition 4, we must prove correctness, virtual black box security, and nonmalleability.

Correctness: We present the following lemma proving correctness. Its proof is deferred to the full version [3].

Lemma 1. *For any ρ such that $(\rho+6)2^{2\rho}/p(\lambda)^3 + \rho/p(\lambda) = \mathtt{ngl}(\lambda)$, Construction 1 satisfies completeness.*

Security: We present the following theorem proving security. Its proof is deferred to the full version [3]. Within the proof, Lemma 2 presents a general approach to creating valid point obfuscations from Assumption 1, which will be used in later constructions, as well.

Theorem 3. *Let ρ be the length of $\mathtt{ad}$. Suppose that Assumption 1 holds for highest power $\psi = \rho + 10$. Then, Construction 1 satisfies virtual black box security.*

Nonmalleability: Finally, we must prove nonmalleability. We give the main theorem below. The proof strategy for it is as follows:

1. **Lemma 2.** We first prove that any method of incorporating associated data suffices for keeping val from being changed as long as there are enough large powers of val that are not affected by associated data. We show this holds even for adversaries that may arbitrarily tamper with the associated data.
2. **Lemma 3.** We then prove that, if the value val is not tampered, then for Construction 1 it is difficult to change $\mathtt{ad} \in \{0,1\}^\rho$ to any distinct $\mathtt{ad}' \in \{0,1\}^\rho$.

The aggregate of both of these results yields the desired nonmalleabillity property. We include the statements of Lemma 2 and Lemma 3 below, as well. Their proofs are deferred to the full version [3].

Theorem 4. *Let λ be a security parameter. Let $\{\mathcal{X}_\lambda\}$ be a well-spread distribution ensemble and let $\tau, \rho \in \mathbb{Z}^+$ be parameters that are both $\mathtt{poly}(\lambda)$. Let $\mathcal{F}_{poly}$ be the ensemble of functions f_λ where f_λ is the set of non-constant, non-identity polynomials in $\mathbb{Z}_{p(\lambda)}[x]$ with degree at most τ. Suppose that Assumption 1 holds for $\psi = \max\{\rho + 10, \tau(\rho+6)\}$. Then, the obfuscator in Construction 1 is nonmalleable over $\mathcal{F}_{poly}$ with distribution ensemble $\{\mathcal{X}_\lambda\}$, and $\mathcal{AD} = \{0,1\}^\rho$.*

Lemma 2. *Let λ be a security parameter. Let $\{\mathcal{X}_\lambda\}$ be a well-spread distribution ensemble and let $\tau, \ell \in \mathbb{Z}^+$ be $\mathtt{poly}(\lambda)$. Let $\mathcal{F}_{poly}$ be the ensemble of functions f_λ where f_λ is the set of non-constant, non-identity polynomials in $\mathbb{Z}_{p(\lambda)}[x]$ with degree at most τ.*

Let $P(x) = r_1 x + \ldots + r_{\rho-1}x^{\rho-1} + r_\rho x^\rho$ with $r_i \in \mathbb{Z}_{p(\lambda)}$, and let $\vec{P} = \{r_1, \ldots, r_\rho\}$ where any or all of the r_i may be 0. Suppose that Assumption 1 holds for $\psi = \max\{\rho + 10, \tau(\rho + 6)\}$. Define as obfuscation (with c_1, c_2, c_3, c_4, c_5 uniformly distributed in $\mathbb{Z}_{p(\lambda)}$)

$$\mathsf{lockPoint}_P(\mathsf{val}, \vec{P}; c_1, c_2, c_3, c_4, c_5) \stackrel{def}{=} \vec{P}, \begin{bmatrix} c_1, & \left[c_1\mathsf{val} + \mathsf{val}P(\mathsf{val}) + \sum_{i=\rho+2}^{\rho+6} \mathsf{val}^i\right]_g \\ c_2, & \left[c_2\mathsf{val} + \mathsf{val}^{\rho+7}\right]_g \\ c_3, & \left[c_3\mathsf{val} + \mathsf{val}^{\rho+8}\right]_g \\ c_4, & \left[c_4\mathsf{val} + \mathsf{val}^{\rho+9}\right]_g \\ c_5, & \left[c_5\mathsf{val} + \mathsf{val}^{\rho+10}\right]_g \end{bmatrix}.$$

Consider $\mathcal{F}_{poly}$ and distribution ensemble $\{\mathcal{X}_\lambda\}$. For any nonmalleability PPT adversary $\mathcal{A}$ in Definition 4, $\mathcal{A}$ outputs a valid $f, P', \mathsf{unlockPoint}_{P'}$ with negligible probability.

Lemma 3. *Let λ be a security parameter. Let $\{\mathcal{X}_\lambda\}$ be a well-spread distribution ensemble and let $\tau, \rho \in \mathbb{Z}^+$ be $\mathtt{poly}(\lambda)$. Let $\mathcal{F}_{poly}$ be the ensemble of functions f_λ where f_λ is the set of non-constant, non-identity polynomials in $\mathbb{Z}_{p(\lambda)}[x]$ with degree at most τ.*

Let $P_{\vec{b}}(x) = b_\rho x^\rho + b_{\rho-1}x^{\rho-1} + \ldots + b_1 x$ where $b_i \in \{0, 1\}$. Suppose that Assumption 2 holds for $\psi = \max\{\rho + 10, \tau(\rho + 6)\}$. Define as an obfuscation (with c_1, c_2, c_3, c_4, c_5 uniformly distributed in $p(\lambda)$):

$$\mathsf{lockPoint}(\mathsf{val}, \vec{b}; c_1, c_2, c_3, c_4, c_5) \stackrel{def}{=} \vec{b}, \begin{bmatrix} c_1, & \left[c_1\mathsf{val} + \mathsf{val}P_{\vec{b}}(\mathsf{val}) + \sum_{i=\rho+2}^{\rho+6} \mathsf{val}^i\right]_g \\ c_2, & \left[c_2\mathsf{val} + \mathsf{val}^{\rho+7}\right]_g \\ c_3, & \left[c_3\mathsf{val} + \mathsf{val}^{\rho+8}\right]_g \\ c_4, & \left[c_4\mathsf{val} + \mathsf{val}^{\rho+9}\right]_g \\ c_5, & \left[c_5\mathsf{val} + \mathsf{val}^{\rho+10}\right]_g \end{bmatrix}.$$

Consider $\mathcal{F}_{poly}$ and distribution ensemble $\{\mathcal{X}_\lambda\}$. The probability that any PPT algorithm outputs a valid obfuscation with the identity function f and some $P_{\vec{b}'}$ with $\vec{b}' \in \{0, 1\}^\rho, \vec{b}' \neq \vec{b}$ is negligible.

5 Standard Model Digital Lockers

We will now construct a nonmalleable digital locker in two steps.

- In Sect. 5.1 we amend our previous construction of a $\mathtt{NMPO}_{\mathsf{ad}}$ to instead output a predetermined key rather than a single bit. Nonmalleability of the input val and ad must still be preserved, but *no* nonmalleability is guaranteed for key.

– In Sect. 5.2, we use this intermediate digital locker with a non-interactive zero knowledge proof, to guarantee complete nonmalleability over key.

Of course, correctness and security must hold for val, key as well. The end result of these efforts (Construction 3) will be a digital locker with: 1) input val nonmalleable over low-degree polynomials, 2) public helper string ad nonmalleable over any tampering, and 3) output key nonmalleable over any tampering. As we will see in Sect. 6, these tampering classes have meaningful applications.

5.1 Digital Lockers Nonmalleable Over Val and Ad

We integrate our $\mathtt{NMPO}_{\mathsf{ad}}$ with the real-or-random construction [21] in Fig. 1. The essential idea is that we may encode each bit of key as a real (encoding val) or random (encoding a random point) point obfuscation, with an additional obfuscation of val to ensure that is the point being tested. We encode the attestation of ad in this additional obfuscation.

In order to adapt our techniques to a real-or-random digital locker with $|\mathsf{key}| = \ell$, then, it is clear that we must ensure that each point obfuscation retains security in the presence of up to ℓ other copies of the same point (i.e., if $\mathsf{key} = 1^\ell$). The previous construction is clearly not sufficient, providing two copies of the obfuscation breaks security (see discussion in [38]), but we may use similar techniques as so. We begin by defining the intermediate cryptographic object.

Definition 7 (Input Nonmalleable Digital Locker with Associated Data). *For security parameter $\lambda \in \mathbb{N}$, let $\{\mathsf{D}^\lambda\}$ be an ensemble of finite sets, let $\rho \in \mathbb{N}$ be a parameter. Let*

1. *$\mathcal{F} : \mathsf{D}^\lambda \to \mathsf{D}^\lambda$ be a function family,*
2. *$\mathcal{X}$ be a family of distributions over D^λ,*
3. *iLock be a PPT algorithm that maps points $\mathsf{val} \in \mathsf{D}^\lambda, \mathsf{ad} \in \{0,1\}^\rho, \mathsf{key} \in \{0,1\}^n$ to a circuit $\mathsf{iUnlock}$, and*
4. *$\mathsf{V}_{\mathtt{obf}}$ be an obfuscation verifier.*

The algorithm iLock is a $(\mathcal{F}, \mathcal{X}, \rho, n)$-input nonmalleable digital locker *with associated data if all of the below are satisfied:*

1. ***Completeness*** *For all $\mathsf{val} \in \mathsf{D}^\lambda, \mathsf{ad} \in \{0,1\}^\rho, \mathsf{key} \in \{0,1\}^n$ it holds that*

$$\Pr[\mathsf{iUnlock}(\cdot) \equiv I_{\mathsf{val},\mathsf{ad},\mathsf{key}}(\cdot) | \mathsf{iUnlock} \leftarrow \mathsf{iLock}(\mathsf{val}, \mathsf{ad}, \mathsf{key})] \geq 1 - \mathtt{ngl}(\lambda),$$

   where the probability is over the randomness of iLock. Here $I_{\mathsf{val},\mathsf{ad},\mathsf{key}}$ is a function that returns key when provided input $(\mathsf{val}, \mathsf{ad})$, otherwise $I_{\mathsf{val},\mathsf{ad},\mathsf{key}}$ returns $\perp$.
2. ***Virtual Black Box Security****: For all PPT $\mathcal{A}$ and $p(\lambda) = \mathtt{poly}(\lambda)$, $\exists \mathcal{S}$ and $q(\lambda) = \mathtt{poly}(\lambda)$ such that for all large enough $\lambda \in \mathbb{N}$, $\forall \mathsf{val} \in \mathsf{D}^\lambda, \mathsf{ad} \in$*

$\{0,1\}^\rho, \mathsf{key} \in \{0,1\}^n, \mathcal{P} : \mathsf{D}^\lambda \times \{0,1\}^{\rho+n} \mapsto \{0,1\}$,

$$\Big| \Pr[\mathcal{A}(\mathsf{iLock}(\mathsf{val},\mathsf{ad},\mathsf{key}),\mathsf{ad}) = \mathcal{P}(\mathsf{val},\mathsf{ad},\mathsf{key})] - \Pr[\mathcal{S}^{I_{\mathsf{val},\mathsf{ad},\mathsf{key}}}(1^\lambda,\mathsf{ad}) = \mathcal{P}(\mathsf{val},\mathsf{ad},\mathsf{key})]\Big| \leq \frac{1}{p(\lambda)},$$

where S is allowed $q(\lambda)$ oracle queries to $I_{\mathsf{val},\mathsf{ad},\mathsf{key}}$ and the probabilities are over the internal randomness of $\mathcal{A}$ and lock*, and of $\mathcal{S}$, respectively.*

3. ***Input Nonmalleability*** *For all $X \in \mathcal{X}$, PPT $\mathcal{A}, \mathsf{ad} \in \{0,1\}^\rho, \mathsf{key} \in \{0,1\}^n$, there exists $\epsilon = \mathtt{ngl}(\lambda)$ such that:*

$$\Pr_{\mathsf{val}\leftarrow X}\left[\begin{array}{r} \mathsf{V}_{\mathsf{obf}}(C) = 1, \\ f \in \mathcal{F} \vee (f = \mathsf{id} \wedge \mathsf{ad}' \neq \mathsf{ad}) \\ C(f(\mathsf{val}),\mathsf{ad}') \neq \perp \end{array} \middle| \begin{array}{l} \mathsf{unlock}_{\mathsf{val},\mathsf{key}} \leftarrow \mathsf{iLock}(\mathsf{val},\mathsf{ad},\mathsf{key}) \\ (C,f,\mathsf{ad}') \leftarrow \mathcal{A}(\mathsf{unlock}_{\mathsf{val},\mathsf{key}},\mathsf{ad}) \end{array}\right] \leq \epsilon.$$

Remark 4. Note that input nonmalleability does not protect against key tampering. In fact, an adversary that arbitrarily mauls key to $\mathsf{key}' \in \{0,1\}^n$ is allowed for this object, so long as val and ad are not tampered.

Before introducing the construction, we define some polynomials that will be used in the construction as follows:

$$p_{0,\mathsf{ad},\vec{c_0}}(\mathsf{val}) = c_{0,1}\mathsf{val} + \sum_{i=1}^{\ell} c_{0,i+1}\mathsf{val}^{i+1} + \sum_{i=1}^{\rho} \mathsf{ad}_i\mathsf{val}^{\ell+1+i} + \sum_{i=1}^{5} \mathsf{val}^{\ell+\rho+1+i}, \tag{1}$$

$$p_{0,1,c_{0,\ell+2}}(\mathsf{val}) = c_{0,\ell+2}\mathsf{val} + \mathsf{val}^{\ell+\rho+7}, \tag{2}$$

$$p_{0,2,c_{0,\ell+3}}(\mathsf{val}) = c_{0,\ell+3}\mathsf{val} + \mathsf{val}^{\ell+\rho+8}, \tag{3}$$

$$p_{0,3,c_{0,\ell+4}}(\mathsf{val}) = c_{0,\ell+4}\mathsf{val} + \mathsf{val}^{\ell+\rho+9}, \tag{4}$$

$$p_{0,4,c_{0,\ell+5}}(\mathsf{val}) = c_{0,\ell+5}\mathsf{val} + \mathsf{val}^{\ell+\rho+10}, \tag{5}$$

$$p^*_{\vec{c}}(\mathsf{val}) = c_{j,1}\mathsf{val} + \sum_{i=1}^{\ell} c_{j,i+1}\mathsf{val}^{i+1}. \tag{6}$$

Construction 2. *Let $\lambda \in \mathbb{N}$ be a security parameter, let $\rho, \ell \in \mathbb{N}$ be parameters. Let $\mathcal{G} = \{\mathbb{G}_\lambda\}$ be a group ensemble with efficient representation and operations where each $\mathbb{G}_\lambda$ is a group of prime order $p(\lambda) \in (2^\lambda, 2^{\lambda+1})$. Let $\mathsf{D}^\lambda = \mathbb{Z}_{p(\lambda)}$. Let g be a generator of $\mathbb{G}_\lambda$. Let $\rho, \ell \in \mathbb{Z}^+$ such that $\rho = O(\log\lambda)$ and $\ell = \mathtt{poly}(\lambda)$. Define the Construction of* (iLock, iUnlock) *as in Fig. 1.*

Theorem 5. *Let all parameters be as in Construction 2. Let $\tau \in \mathbb{N}$ and $\rho \in \mathbb{N}$ be parameters.*

1. *Suppose that Assumption 1 holds for maximum power* $\max\{\ell+\rho+10, \tau(\ell+\rho+6)\}$,

$\mathsf{iLock}(\mathsf{val}, \mathsf{ad}, \mathsf{key})$:

1. Define $\ell = |\mathsf{key}|$,
2. Sample $z \leftarrow \mathbb{Z}_{p(\lambda)}$,
3. Sample $\mathbf{C} \leftarrow \mathbb{Z}_{p(\lambda)}^{(\ell+1)\times(\ell+1)}$, $c_{0,\ell+2}, c_{0,\ell+3}, c_{0,\ell+4}, c_{0,\ell+5} \leftarrow \mathbb{Z}_{p(\lambda)}$,
4. Compute $\mathsf{unlockPoint}_{\mathsf{ad}} =$

$$\begin{bmatrix} [p_{0,\mathsf{ad},\mathbf{C}_0}(\mathsf{val})]_g \\ c_{0,\ell+2},\ [p_{0,1,c_{0,\ell+2}}(\mathsf{val})]_g \\ c_{0,\ell+3},\ [p_{0,2,c_{0,\ell+3}}(\mathsf{val})]_g \\ c_{0,\ell+4},\ [p_{0,3,c_{0,\ell+4}}(\mathsf{val})]_g \\ c_{0,\ell+5},\ [p_{0,4,c_{0,\ell+5}}(\mathsf{val})]_g \end{bmatrix}$$

5. For $i = 1$ to ℓ:

$$\mathsf{unlockPoint}_i = \begin{cases} [p^*_{\mathbf{C}_i}(\mathsf{val})]_g & \mathsf{key}_i = 1 \\ [p^*_{\mathbf{C}_i}(z)]_g & \mathsf{key}_i = 0 \end{cases}.$$

6. Output $\mathbf{C}, \mathsf{unlockPoint}_{\mathsf{ad}}, \{\mathsf{unlockPoint}_i\}_{i=1}^{\ell}$.

$\mathsf{iUnlock}(\mathbf{C}', \mathsf{unlockPoint}'_{\mathsf{ad}}, \{\mathsf{unlockPoint}'_i\}_{i=1}^{\ell}, \mathsf{val}', \mathsf{ad}')$:

1. Parse $\mathsf{unlockPoint}'_{\mathsf{ad}}$ as $c'_{0,\ell+2}, c'_{0,\ell+3}, c'_{0,\ell+4}, c'_{0,\ell+5}, g'_{0,1}, g'_{0,2}, g'_{0,3}, g'_{0,4}, g'_{0,5}$.
2. Verify

$$\left[p_{1,\mathsf{ad}',\mathbf{C}'_0}(\mathsf{val}')\right]_g = g'_{0,1},$$

$$\left\{\left[p_{i,c'_{0,i}}(\mathsf{val}')\right]_g = g'_{0,i}\right\}_{i=2}^{5}.$$

   If one checks does not pass output $\perp$.
3. Initialize $\mathsf{key} = \vec{0}^{\ell}$.
4. For $i = 1$ to ℓ:
   (a) If $[p^*_{\mathbf{C}'_i}(\mathsf{val}')]_g = \mathsf{unlockPoint}'_i$ set $\mathsf{key}_i = 1$.
5. Output key.

Fig. 1. Real-or random-instantiation of input nonmalleable digital locker with associated data.

2. *Let* $\mathcal{F}_{\mathtt{poly}}$ *be the family of polynomials over* $\mathbb{Z}_{p(\lambda)}$ *with maximum degree* τ, *and*
3. $(\ell + \rho + 10)2^{2\rho}/p(\lambda)^3 = \mathtt{ngl}(\lambda)$.

Then, Construction 2 is a $(\mathcal{F}_{\mathtt{poly}}, \mathcal{X}, \rho, \ell)$*-input nonmalleable digital locker with associated data.*

The proof of this statement is deferred to the full version [3]. The technical details behind Theorem 5 follow the same structure as Theorem 2 — we prove correctness, virtual black box security, and input nonmalleability separately, each following a similar proof structure as the respective property of Construction 1.

5.2 Adding Key Nonmalleability

We now show that the input nonmalleable digital locker with associated data suffices to build a fully nonmalleable digital locker for the same function class. Let iLock be such an object and $\Pi = (\mathsf{Setup}, P, V)$ be some appropriate non-interactive proof system (described in Sect. 5.3) using a $\mathtt{crs}$ of length ρ for the following language that proves well-formness of iLock:

$$\mathcal{L} = \{\mathsf{iUnlock} : \exists(\mathsf{val}, \mathtt{crs}, \mathsf{key}, r) \text{ such that } \mathsf{iUnlock} = \mathsf{iLock}(\mathsf{val}, \mathtt{crs}, \mathsf{key}; r)\}$$

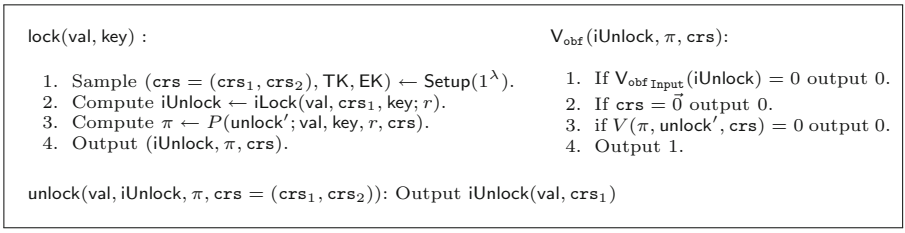

Fig. 2. Digital locker construction.

Construction 3. *For security parameter $\lambda \in \mathbb{N}$, let $\mathcal{F} : \mathsf{D}^\lambda \to \mathsf{D}^\lambda$ be a family of functions, let $\rho, \ell \in \mathbb{N}$ be parameters, $\mathcal{X}$ be a family of distributions over D^λ. Suppose that*

1. iLock *is a $(\mathcal{F}_{\mathtt{poly}}, \mathcal{X}, \rho, \ell)$-input-nonmalleable digital locker with associated data with associated obfuscation verifier $\mathsf{V}_{\mathtt{obf\,Input}}$, and*
2. *$\Pi = (\mathsf{Setup}, P, V)$ is an NIZK system for the language $\mathcal{L}$ with short non-tamperable CRS.*[5] *We formally define this property and show a generic construction in Sect. 5.3.*

Then define $(\mathsf{lock}, \mathsf{unlock}, \mathsf{V}_{\mathtt{obf}})$ as in Fig. 2.

Theorem 6. *Let notation be as in Construction 3. Suppose that*

1. iLock *is a $(\mathcal{F}_{\mathtt{poly}}, \mathcal{X}, \rho, \ell)$-input-nonmalleable digital locker with associated data with associated obfuscation verifier $\mathsf{V}_{\mathtt{obf\,Input}}$, and*
2. *$\Pi = (\mathsf{Setup}, P, V)$ is a true simulation extractable non-interactive zero knowledge proof system as described in Sect. 5.3,*
3. *That every function $f \in \mathcal{F}$ is entropy preserving; i.e., for any well-spread X, $f(X)$ is also well-spread.*

Then $\mathsf{lock}, \mathsf{unlock}$ is a $(\mathcal{F}, \mathcal{X}, n)$-nonmalleable digital locker.

Proof (Proof of Theorem 6). Following Definition 5, we need to prove completeness, soundness, and nonmalleability. Completeness can be easily verified, so we just focus on the non-trivial parts, i.e., proof of soundness and nonmalleability.

Soundness. To prove soundness, we first observe that according to Theorem 1, for this class of circuits being obfuscated DI is equivalent to VBB, so for the rest of the proof, we focus on proving the DI. We prove soundness by contradiction. Suppose there exists a PPT adversary $\mathcal{A}$, a key $\mathsf{key} \in \{0,1\}^\ell$, and a well-spread distribution X such that

$$|\Pr_{\mathsf{val} \leftarrow \mathcal{X}}[\mathcal{A}(\mathsf{lock}(\mathsf{val}, \mathsf{key})) = 1] - \Pr_{r \xleftarrow{\$} \mathsf{D}^\lambda}[\mathcal{A}(\mathsf{lock}(r, \mathsf{key})) = 1] > \epsilon$$

[5] That is, $\mathtt{crs}$ can be split into $(\mathtt{crs}_1, \mathtt{crs}_2)$ where $\mathtt{crs}_1$ has length independent of the language, i.e., $O(\lambda)$, and only $\mathtt{crs}_1$ is required to be non-tamperable. $\mathtt{crs}_2$ cannot be modified (computationally infeasible) given the original $\mathtt{crs}_1$.

for some non-negligible ϵ, then there exists an adversary $\mathcal{B}$ that breaks the DI security of the input-nonmalleable digital locker. This reaches a contradiction.

$\mathcal{B}$ receives the distribution X samples $(\mathtt{crs}_1, \mathtt{crs}_2, \mathsf{TK}, \mathsf{EK}) \leftarrow \mathsf{Setup}(1^\lambda)$ for the proof system, and sets associated data as $\mathtt{crs}_1$. $\mathcal{B}$ sends this distribution to the reduction for iLock with the input distribution the same X, and associated data, $\mathtt{crs}_1$. The reduction samples some $\mathsf{val} \leftarrow X$ or uniform r. $\mathcal{B}$ receives $\mathsf{iUnlock}$. Next $\mathcal{B}$ creates a simulated π. It sends $\mathsf{iUnlock}, \pi, \mathtt{crs}$ to $\mathcal{A}$ and outputs $\mathcal{A}$'s decision.

Clearly, if val is from the distribution X, then the reduction has simulated an indistinguishable $\mathsf{lock}(\mathsf{val}, \mathsf{key})$ (assuming the simulated proof π is indistinguishable), or otherwise, $\mathsf{lock}(r, \mathsf{key})$. That is, in both cases, the obfuscation is properly prepared assuming the indistinguishability of the simulated proof. Thus, the advantage of the adversary $\mathcal{A}$ translates to the advantage of $\mathcal{B}$ in breaking the DI of the nonmalleable point obfuscation. By the equivalence of DI and VBB of point obfuscation, this breaks the soundness of the nonmalleable point obfuscation.

Nonmalleability. Now we prove nonmalleability. As before, we prove by contradiction. Suppose there exists a PPT adversary $\mathcal{A}$ and $\mathsf{key} \in \{0,1\}^\ell$, a well-spread distribution X such that $\mathcal{A}$ breaks the nonmalleability experiment with non-negligible probability ϵ. Then there exists an adversary $\mathcal{B}$ that breaks the nonmalleability of the underlying $\mathsf{iLock}(\cdot)$.

$\mathcal{B}$ follows exactly the same procedure in preparing the input to the adversary $\mathcal{A}$ as in soundness proof above. Now $\mathcal{A}$ would return a triple $(C, f, \mathtt{crs}^* = (\mathtt{crs}_1^*, \mathtt{crs}_2^*))$ that passes the checking conditions with a non-negligible probability ϵ. Assume C is different from the original obfuscation given to $\mathcal{A}$ (as we don't allow identity tampering). $\mathcal{B}$ does the following:

- If the $\mathtt{crs}_1$ is modified to a different $\mathtt{crs}_1^*$, then the reduction just outputs the $C, f, \mathtt{crs}_1^*$ which correspond to a tamper according to nonmalleability of $\mathsf{iLock}(\cdot)$.
- If the $\mathtt{crs}_1$ is kept intact but $\mathtt{crs}_2$ is modified to a different $\mathtt{crs}_2^*$, then this breaks the underlying NIZK as it is computationally infeasible to obtain a consistent but different $\mathtt{crs}_2^*$.
- If the $\mathtt{crs} = \mathtt{crs}^*$ in C is intact yet the statement-proof pair is modified, then $\mathcal{B}$ runs the witness extractor to extract a valid witness, i.e., val' used to generate C. As the input obfuscated circuits received by $\mathcal{B}$ are properly prepared by the challenger, the simulated proof given to the adversary $\mathcal{A}$ is with respect to a true statement. In this case, the notion of true simulation extractability allows $\mathcal{B}$ to extract a valid witness by running the extractor. Thus, given $\mathsf{val}' = f(\mathsf{val})$. $\mathcal{B}$ can prepare an obfuscation (with an arbitrary associated data of val'), breaking the nonmalleability of $\mathsf{iLock}(\cdot)$.

Since $\mathcal{A}$ wins the nonmalleable experiment with a non-negligible probability, one of the above case must happen with a non-negligible probability. This would imply the contradiction we expect. The above two arguments complete the proof of Theorem 6.

5.3 The Building Block – True Simulation Extractable NIZK

In this section, we present the building block used in Construction 3 – true simulation extractable NIZK. The notion was introduced by Dodis et al. [32] as a relaxation of *all* simulation extractable NIZK. We describe the notion in what follows.

Definition 8. *Let R be an NP relation on pairs (x, w) with corresponding language $L_R = \{x : \exists w \text{ such that } (x, w) \in R\}$. A true-simulation extractable non-interactive zero-knowledge (NIZK) argument for a relation R consists of three algorithms* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *with the following syntax:*

- $(\mathtt{crs}, \mathsf{TK}, \mathsf{EK}) \leftarrow \mathsf{Setup}(1^\lambda)$*: creates a common reference string* $\mathtt{crs}$*, a trapdoor* TK*, and an extraction key* EK*.*
- $\pi \leftarrow \mathsf{Prove}(\mathtt{crs}, x, w)$*: creates an argument π that $R(x, w) = 1$.*
- $0/1 \leftarrow \mathsf{Verify}(\mathtt{crs}, x, \pi)$*: verifies whether or not the argument π is correct.*

For presentation simplicity, we omit $\mathtt{crs}$ *in the* Prove *and* Verify*. We require that the following three basic properties hold:*

- ***Completeness.*** *For any $(x, w) \in R$, if $(\mathtt{crs}, \mathsf{TK}, \mathsf{EK}) \leftarrow \mathsf{Setup}(1^\lambda)$, $\pi \leftarrow \mathsf{Prove}(x, w)$, then $\mathsf{Verify}(x, \pi) = 1$.*
- ***Soundness.*** *For any PPT adversary $\mathcal{A}$, the following probability is negligible: for $(\mathtt{crs}, \mathsf{TK}, \mathsf{EK}) \leftarrow \mathsf{Setup}(1^\lambda)$, $(x^*, \pi^*) \leftarrow \mathcal{A}(\mathtt{crs})$ such that $x^* \notin L_R$ but $\mathsf{Verify}(x^*, \pi^*) = 1$.*
- ***Composable Zero-knowledge.*** *There exists a PPT simulator $\mathcal{S}$ such that for any PPT $\mathcal{A}$, the advantage (the probability $\mathcal{A}$ wins minus one half) is negligible in the following game.*
  - *The challenger samples $(\mathtt{crs}, \mathsf{TK}, \mathsf{EK}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends $(\mathtt{crs}, \mathsf{TK})$ to $\mathcal{A}$*
  - *$\mathcal{A}$ chooses $(x, w) \in R$ and sends to the challenger.*
  - *The challenger generates $\pi_0 \leftarrow \mathsf{Prove}(x, w)$, $\pi_1 \leftarrow \mathsf{Sim}(x, \mathsf{TK})$, and then samples a random bit $b \leftarrow \{0, 1\}$. Then he sends π_b to $\mathcal{A}$.*
  - *$\mathcal{A}$ outputs a guess bit b', and wins if $b' = b$.*
- ***Extractibility.*** *Additionally, true simulation extractability requires that there exists a PPT extractor* Ext *such that for any PPT adversary $\mathcal{A}$, the probability $\mathcal{A}$ wins is negligible in the following game:*
  - *The challenger samples $(\mathtt{crs}, \mathsf{TK}, \mathsf{EK}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends $\mathtt{crs}$ to $\mathcal{A}$.*
  - *$\mathcal{A}$ is allowed to make oracle queries to the simulation algorithm $\mathsf{Sim}'((x, w), \mathsf{TK})$ adaptively. Sim' first checks if $(x, w) \in R$ and returns $\mathsf{Sim}(x, TK)$ if that is the case.*
  - *$\mathcal{A}$ outputs a tuple x^*, L^*, π^*.*
  - *The challenger runs the extractor $w^* \leftarrow \mathsf{Ext}(L^*, (x^*, \pi^*), \mathsf{EK})$.*
  - *$\mathcal{A}$ wins if (1) the pair (x^*, L^*) was not part of the simulator query, (2) the proof π^* verifies, and (3) $R(x^*, w^*) = 0$.*

Briefly speaking, a true simulation extractable NIZK requires that the adversary can only query the simulation oracle only on true statements, whereas all simulation extractability allows the adversary to query on any (perhaps false) statement. As shown by the work [32], the true simulation extractable NIZK can be constructed in a fairly simple way as summarized by the following theorem.

Theorem 7 ([32]). *Assume that there exists a CCA2 encryption and a regular NIZK argument for NP languages, then there exists a true simulation extractable NIZK for NP languages.*

The work [32] showed how to instantiate the building blocks under the SXDH assumption over bilinear groups. There is plausible evidence that the regular NIZK can be constructed without the need of pairing groups, c.f. [28], under some non-standard assumptions.

5.4 NIZK with Short Non-tamperable CRS

The generic use of the NIZK from Dodis et al. [32] requires long CRS that would depend on the language being proved, and this is a general fact for NIZKs. In our application, however, this poses a challenge when we combine this with our non-malleable obfuscation with associate data. Particularly, the correctness of Theorem 2 requires a group that has a length larger than that of associated data. We notice that the language $\mathcal{L}$ used in Construction 3 requires a long CRS, as the statement and the witness are long. So, putting CRS as the associated data in the non-malleable digital locker would require a significantly larger group, which is undesirable.

To handle this technical subtlety, we present a simple transformation from any NIZK into one whose CRS has the following structure: $\mathtt{crs} = (\mathtt{crs}_1, \mathtt{crs}_2)$, where only $\mathtt{crs}_1$ is short and non-tamperable, $\mathtt{crs}_2$ can be arbitrarily long but cannot be tampered consistently (computationally infeasible) as long as $\mathtt{crs}_1$ is kept intact. In this way, we can put $\mathtt{crs}_1$ as the associated data into our non-malleable digital locker, and keep $\mathtt{crs}_2$ public, as we presented in the prior section. Thus, the underlying group of the non-malleable obfuscation can be significantly smaller.

To achieve this, given any $\mathtt{crs}'$ from the underlying NIZK, we define a new NIZK which is essentially the same as the original one, except in the CRS generation: first it samples a collision resistant hash function h and computes $z = h(\mathtt{crs})$. It outputs $\mathtt{crs} = (\mathtt{crs}_1 = (h, z), \mathtt{crs}_2 = \mathtt{crs}')$ as the new CRS. The verifier will always check whether $h(\mathtt{crs}_2) = z$ and rejects immediately if it does not hold. The security (zero-knowledge, soundness) is not affected by $\mathtt{crs}_1$, as it can be generated just given $\mathtt{crs}'$.

6 Application to Fuzzy Extractors

In this section, we show that a nonmalleable digital locker suffices to build a robust fuzzy extractor [14–16,33] when combined with a standard secure sketch.

We note information-theoretic robust fuzzy extractor in the plain model or CRS models requires the source to have an entropy of at least half its length [35]. In this work, we consider computational robust fuzzy extractors in the plain model. We begin with a few definitions.

Definition 9 (Secure Sketch). *Let λ be a security parameter. Let $\mathcal{W} = \mathcal{W}_\lambda$ be a family of random variables over metric space $(\mathcal{M}, \mathsf{dis}) = (\mathcal{M}_\lambda, \mathsf{dis}_\lambda)$. Then $\mathsf{SS}, \mathsf{Rec}$ is a $(\mathcal{M}, \mathcal{W}, t, \delta)$-secure sketch if the following hold:*

Correctness. *For all $w, w' \in \mathcal{M}$ such that $\mathsf{dis}(w, w') \leq t$, $\Pr[\mathsf{Rec}(w', \mathsf{SS}(w)) = w] \geq 1 - \delta$.*
Security. *For all distributions $W \in \mathcal{W}$ it is true that $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) \geq \omega(\log \lambda)$.*

Definition 10 (Robust Fuzzy extractor). *An $(\mathcal{M}, \mathcal{W}, \ell, t)$-computationally robust fuzzy extractor is a pair of PPT algorithms $(\mathsf{Gen}, \mathsf{Rep})$ where for all $w, w' \in \mathcal{M}$,*

- $(\mathsf{key}, \mathsf{pub}) \leftarrow \mathsf{Gen}(w)$, *where* $\mathsf{key} \in \{0,1\}^\ell$ *and* $\mathsf{pub} \in \{0,1\}^*$
- $\mathsf{key}' \leftarrow \mathsf{Rep}(\mathsf{pub}, w')$

such that the following properties are true:

- ***Correctness :*** *For all $w, w' \in \mathcal{M}$ such that $\mathsf{dist}(w, w') \leq t$,*

$$\Pr\left[\mathsf{key}' = \mathsf{key} \;\middle|\; (\mathsf{key}, \mathsf{pub}) \leftarrow \mathsf{Gen}(w), \mathsf{key}' \leftarrow \mathsf{Rep}(\mathsf{pub}, w')\right] \geq 1 - \mathtt{ngl}(\lambda).$$

- ***Security :*** *For any distribution $W \in \mathcal{W}$, and for $(\mathsf{key}, \mathsf{pub}) \leftarrow \mathsf{Gen}(W)$, for all PPT $\mathcal{A}$ there exists some $\mathtt{ngl}(\lambda)$ function such that*

$$|\Pr[\mathcal{A}(\mathsf{key}, \mathsf{pub}) = 1] - \Pr[\mathcal{A}(U_\ell, \mathsf{pub}) = 1] \leq \mathtt{ngl}(\lambda).$$

  where U_ℓ is a uniformly distributed random variable on $\{0,1\}^\ell$.
- ***Robustness:*** *Let $W, W' \in \mathcal{M}$ be (correlated) distributions such that*

$$\Pr_{(w,w') \leftarrow (W,W')}[\mathsf{dis}(w, w') \leq t] = 1$$

  and $W, W' \in \mathcal{W}$. For all $W, W' \in \mathcal{W}$ and for all adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following experiment is at most $\mathtt{ngl}(\lambda)$:
  1. *Sample $(w, w') \leftarrow (W, W')$.*
  2. *Compute $(\mathsf{key}, \mathsf{pub}) \leftarrow \mathsf{FE.Gen}(w)$ and send it to $\mathcal{A}$.*
  3. *$\mathcal{A}$ outputs pub' and wins if $\mathsf{pub}' \neq \mathsf{pub}$ and $\mathsf{FE.Rep}(\mathsf{pub}', w') \notin \{\perp, \mathsf{key}\}$.*

Before introducing a common secure sketch which uses code syndromes we introduce the notation of $\mathsf{Wgt}(x) = \mathsf{dis}(x, 0)$ as the Hamming weight of x.

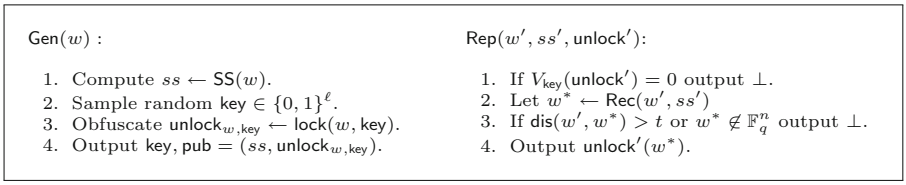

Fig. 3. Robust Fuzzy Extractor from nonmalleable digital locker and syndrome secure sketch.

Definition 11 (Syndrome). *Let* $\mathbf{A} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ *be a* $(n, k, d = 2t + 1)$*-linear error code, then there exists a matrix* $\mathsf{Syn} : \mathbb{F}_q^n \to \mathbb{F}_q^{n-k}$ *with two properties:*

1. *For all values* x *where* $\mathsf{Wgt}(x) \leq t$ *the value* $\mathsf{Syn}(x)$ *is unique.*
2. *There is an efficient mapping from* $s \in \mathbb{F}_q^{n-k}$ *to the value* x *of weight at most* t *if one exists. Let* Invert *denote this mapping. If no such value exists then the output of* Invert *is* $\perp$*.*
3. *For any two values* s, s' *where* $\mathsf{Wgt}(s), \mathsf{Wgt}(s'), \mathsf{Wgt}(s - s') \leq t$ *it is true that*

$$\begin{aligned}\mathsf{Invert}(\mathsf{Syn}(s - s')) &= \mathsf{Invert}(\mathsf{Syn}(s) - \mathsf{Syn}(s')) \\ &= \mathsf{Invert}(\mathsf{Syn}(s)) - \mathsf{Invert}(\mathsf{Syn}(s')) = s - s'.\end{aligned}$$

Definition 12 (Syndrome Secure Sketch [11,30,34]). *Let* $\mathcal{W} \in \mathbb{F}_q^n$ *be the set of all distributions* W *where* $\mathrm{H}_\infty(W) = (n - k)\log q + \omega(\log \lambda)$*. Let* Syn *be the Syndrome of an* $(n, k, d = 2t+1)$*-error correcting code. Then define* $\mathsf{SS}(w) = \mathsf{Syn}(w)$ *and* $\mathsf{Rec}(w', s) = w' - \mathsf{Invert}(\mathsf{Syn}(w') - s) = w' - \mathsf{Invert}(\mathsf{Syn}(w' - w)) = w$*. Then* $(\mathsf{SS}, \mathsf{Rec})$ *is a* $(\mathbb{F}_q^n, \mathcal{W}, t, 0)$*-secure sketch.*

Theorem 8. *Assume the following:*

1. $(\mathsf{SS}, \mathsf{Rec})$ *be a syndrome secure sketch for distance* $2t$*, that is,* $d = 4t + 1$*,*
2. $\mathcal{W}$ *is the set of all efficiently sampleable distributions* W *where*

$$\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) \geq \omega(\log \lambda),$$

3. $(\mathsf{lock}, \mathsf{unlock}, V_{\mathsf{key}})$ *is a nonmalleable digital locker for* $(\mathcal{F}, \mathcal{X})$ *where* $\mathcal{F}$ *includes all functions* $f : \mathbb{F}_q^n \to \mathbb{F}_q^n$ *of the form* $f(x) = x + a$ *and where* $\mathcal{X}$ *is the set of all distributions* X *where* $\mathrm{H}_\infty(X) = \omega(\log \lambda)$*.*

Then $(\mathsf{Gen}, \mathsf{Rep})$ *described in Fig. 3 is a* $(\mathcal{M}, \mathcal{W}, \ell, t)$*-robust fuzzy extractor (Definition 10).*

See the full version [3] for the full proof. The intuition behind the robustness proof is that the adversary will be able to extract the function f from the robust fuzzy extractor adversary's output by computing $\mathsf{Invert}(ss' - ss)$.

Aligning Tampering Functions. There is a subtlety when we instantiate the fuzzy extractor of Theorem 8 – the digital locker in Theorem 8 requires a function class of the domain $\mathbb{F}_q^n$, whereas the digital locker constructed in Fig. 2 works in $\mathbb{Z}_{p(\lambda)}$. It is unclear whether there is an additively homomorphic mapping between these spaces for arbitrary p, q, n. Therefore, a trivial plug-in of the digital locker of Fig. 2 does not work. In Sect. 6.1, we show that how to align readings in a simple way at the cost of increased leakage of the secure sketch.

An Alternative to Efficiently Sampleable W. Theorem 8 required W to be efficiently sampleable. This is because in the proof the reduction samples a $w \leftarrow W$ to compute a secure sketch ss and create the conditional distribution $W|\mathsf{SS}(w)$. An alternative approach is to define all of the objects throughout our main technical sections to be nonmalleable in the presence of auxiliary information Z such that $\mathrm{H}_\infty(W|Z) \geq \omega(\log \lambda)$. In this case, $\mathcal{A}'$ can receive ss as auxiliary information and directly forward it to $\mathcal{A}$.

All of the proofs contained in this work naturally extend to the setting of auxiliary information. The major work needed to have confidence in the auxiliary input approach is to show that [8, Assumption 3] holds in the non-uniform generic group model [26] in the presence of auxiliary information. Importantly, the distribution W has average min-entropy conditioned on $\mathsf{SS}(W)$. There are strong impossibility results on digital lockers that are secure against hard to invert auxiliary information [19].

Applications of Nonmalleable Point Function Obfuscation. Nonmalleable point obfuscation and nonmalleable point obfuscation with associated data (Definition 4) can be used to build robust secure sketches and robust fuzzy extractors, respectively.

- **Robust secure sketch:** Robustness for secure sketches is defined in a similar fashion as for fuzzy extractors. For correlated distributions W, W', the adversary receives $\mathsf{SS}(w)$ from the challenger and outputs SS'. The adversary wins the robustness game if he succeeds in finding a value SS' such that $\mathsf{Rec}(\mathsf{SS}', w') \notin \{\perp, w\}$. Informally, suppose $(\mathsf{lockPoint}, \mathsf{unlockPoint})$ is a nonmalleable point obfuscation and $(\mathsf{SS}, \mathsf{Rec})$ is a syndrome-based secure sketch. Then Fig. 4 describes a robust secure sketch. The formal theorem and proof can be found in the full version [3].
- **Robust fuzzy extractor:** Let $(\mathsf{lockPoint}, \mathsf{unlockPoint})$ be a nonmalleable point obfuscation with associated data, $(\mathsf{SS}, \mathsf{Rec})$ be a syndrome-based secure sketch and $\mathtt{ext}$ be a randomness extractor. Then Fig. 5 describes a robust fuzzy extractor. We stress that this construction requires the remaining entropy of W to be high conditioned on both the produced key which is produced using a randomness extractor [47,51] and $\mathsf{SS}(w)$. There is no limitation on the key length in the robust fuzzy extractor from the nonmalleable digital locker (in Theorem 8). The formal theorem and proof can be found in the full version [3].

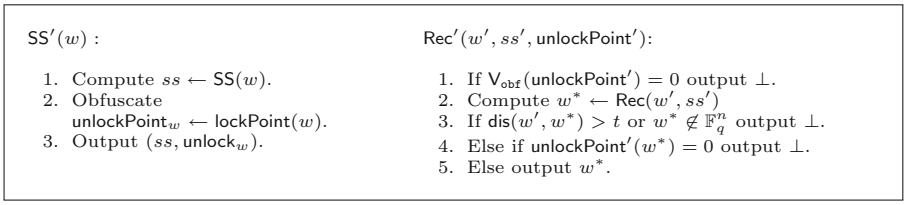

Fig. 4. Robust Secure Sketch from nonmalleable point obfuscation and syndrome secure sketch.

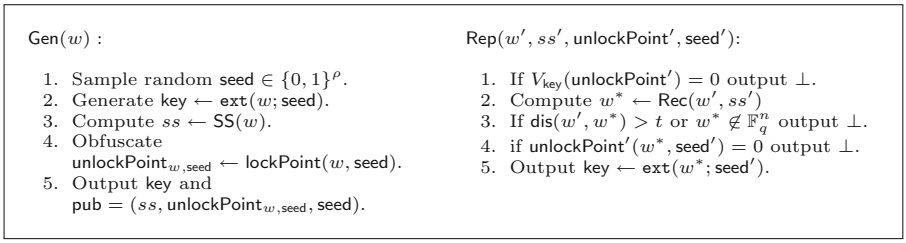

Fig. 5. Robust Fuzzy Extractor from nonmalleable point obfuscation with associated data, syndrome secure sketch and randomness extractor.

6.1 Instantiations – Aligning the Tampering Function Classes

In this section, we show how to align the tampering function classes required by the fuzzy extractor of Theorem 8 and the construction of Fig. 2. This deals with the mismatch in input domain for the syndrome (which takes inputs in $\mathbb{F}_q^n$) and the nonmalleable digital locker (which takes inputs in $\mathbb{Z}_p$).

Assume that the input readings w, w' are q-ary strings of length n. Instead of using a q-ary error correcting code ($\mathbf{A} \in \mathbb{F}_q^{n\times k}$ and $\mathsf{Syn} : \mathbb{F}_q^{n\times(n-k)}$), we consider an error correcting code with entries in $\mathbb{F}_{q'}$ for some prime $q' \geq 2(q-1)+1$. That is, let $\mathbf{A}' \in \mathbb{F}_q^{n\times k}$ be a $(n, k, d = 4t+1)$ linear error correcting code, and let Syn' be the corresponding syndrome. Furthermore, we make the restriction $p \geq q^n$, where $\mathbb{Z}_p$ is the input domain of the digital locker of Fig. 2. In the construction of Rep, note there is a check if the recovered value, w^*, is not q-ary, in which case we output $\perp$. Thus, for the adversary to successfully break robustness they must produce a q-ary output.

Now we encode every string $x \in \mathbb{F}_q^n$ as the natural q-ary representation, i.e., $x \mapsto \sum_{i\in[n]} x_i q^{i-1} \in \mathbb{Z}_p$, denoted as $\mathsf{Enc}(x)$. Moreover, the digital locker takes input an encoded version of w, i.e.,

$$\mathsf{lockPoint}(\mathsf{Enc}(w), \mathsf{seed}) \text{ and } \mathsf{unlockPoint}(\mathsf{Enc}(w^*), \mathsf{seed}').$$

By setting things up in this way, Theorem 8 holds even if the underlying digital locker is non-malleable for shift functions in $\mathbb{Z}_p$.

In Theorem 8 the reduction extracts a tampering function $f : \mathbb{F}_q^n \to \mathbb{F}_q^n$ where $f(w) = w + \mathsf{Invert}(ss' - ss)$. With the modified syndrome construction,

the function $f : \mathbb{F}_q^n \to \mathbb{F}_q^n$, as above, the reduction can extract a function $f(w) = w + \mathsf{Invert}(ss' - ss)$. By the check of $w^* \in \mathbb{F}_q^n$ and the initial condition that $w \in \mathbb{F}_q^n$, this implies that $w_i + \mathsf{Invert}(ss' - ss)_i \in \mathbb{F}_q$, we can first conclude that $\mathsf{Invert}(ss - ss')$ can be represented in $\{-(q-1), ..., (q-1)\}^n$. Under this representation, we conclude that for each i, $w_i + \mathsf{Invert}(ss' - ss)_i \in \mathbb{F}_q$ using standard integer addition. So, for each i, we are guaranteed an element in $\mathbb{F}_q$ (i.e., $\mathsf{Enc}(w^*) = \mathsf{Enc}(w) + \mathsf{Enc}(\mathsf{Invert}(ss' - ss))$), which corresponds exactly to a shift tampering function in $\mathbb{Z}_p$, and thus the reduction can break the underlying non-malleable digital locker.

The effect of this transform is to increase the required entropy on the distribution W. The standard analysis of the secure sketch assumes that $\mathsf{SS}(W)$ leaks $(n-k)\log q$ bits of information about W. By increasing the syndrome from q to q' this increases the leakage of the secure sketch by $(n-k)\log(q'/q) \approx (n-k)\log 2$. This transform applies to the constructions in Figs. 4 and 5 as well. We do not include it in our proofs to show the general connection between syndrome secure sketches and nonmalleable point obfuscation variants.

Acknowledgements. The authors thank the reviewers for their helpful comments and suggestions. The authors wish to thank Leonid Reyzin for important discussions. The work of D.A. was done while at NIST. C.C. is supported by NSF Awards #1849904 and 2141033 and a fellowship from Synchrony Inc. B.F. is supported by NSF Awards #1849904, and 2141033. The work of F.L. is supported by the NSF Award #1942400.

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19020700008. This material is based upon work supported by the Defense Advanced Research Projects Agency under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA, ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

1. Alamélou, Q., et al.: Pseudoentropic isometries: a new framework for fuzzy extractor reusability. In: AsiaCCS (2018)
2. Ananth, P., Jain, A.: Indistinguishability obfuscation from compact functional encryption. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 308–326. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_15
3. Apon, D., Cachet, C., Fenteany, P., Fuller, B., Liu, F.-H.: Nonmalleable digital lockers and robust fuzzy extractors in the plain model. Cryptology ePrint Archive, Report (2022). https://eprint.iacr.org/2022/1108
4. Bahler, L., Di Crescenzo, G., Polyakov, Y., Rohloff, K., Cousins, D.B.: Practical implementation of lattice-based program obfuscators for point functions. In: 2017 International Conference on High Performance Computing & Simulation (HPCS), pp. 761–768. IEEE (2017)

5. Barak, B., Bitansky, N., Canetti, R., Kalai, Y.T., Paneth, O., Sahai, A.: Obfuscation for evasive functions. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 26–51. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_2
6. Barak, B., et al.: On the (im) possibility of obfuscating programs. J. ACM (JACM) **59**(2), 6 (2012)
7. Bartusek, J., Lepoint, T., Ma, F., Zhandry, M.: New techniques for obfuscating conjunctions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11478, pp. 636–666. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17659-4_22
8. Bartusek, J., Ma, F., Zhandry, M.: The distinction between fixed and random generators in group-based assumptions. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11693, pp. 801–830. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26951-7_27
9. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: CCS 1993 - Proceedings of the First ACM Conference on Computer and Communications Security (1993)
10. Bellare, M., Stepanovs, I., Tessaro, S.: Contention in cryptoland: obfuscation, leakage and UCE. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9563, pp. 542–564. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49099-0_20
11. Bennett, C.H., Brassard, G., Crépeau, C., Skubiszewska, M.-H.: Practical quantum oblivious transfer. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 351–366. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_29
12. Bishop, A., Kowalczyk, L., Malkin, T., Pastro, V., Raykova, M., Shi, K.: A simple obfuscation scheme for pattern-matching with wildcards. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10993, pp. 731–752. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96878-0_25
13. Bitansky, N., Canetti, R.: On strong simulation and composable point obfuscation. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 520–537. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_28
14. Boyen, X.: Reusable cryptographic fuzzy extractors. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, pp. 82–91 (2004)
15. Boyen, X.: Robust and reusable fuzzy extractors. In: Security with Noisy Data, pp. 101–112. Springer, Heidelberg (2007). https://doi.org/10.1007/978-1-84628-984-2_6
16. Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., Smith, A.: Secure remote authentication using biometric data. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 147–163. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_9
17. Brakerski, Z., Rothblum, G.N.: Obfuscating conjunctions. J. Cryptol. **30**(1), 289–320 (2017)
18. Brakerski, Z., Vaikuntanathan, V., Wee, H., Wichs, D.: Obfuscating conjunctions under entropic ring LWE. In: Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, pp. 147–156 (2016)
19. Brzuska, C., Farshim, P., Mittelbach, A.: Indistinguishability obfuscation and UCEs: the case of computationally unpredictable sources. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 188–205. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_11
20. Canetti, R.: Towards realizing random oracles: hash functions that hide all partial information. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 455–469. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052255

21. Canetti, R., Dakdouk, R.R.: Obfuscating point functions with multibit output. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 489–508. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_28
22. Canetti, R., Fuller, B., Paneth, O., Reyzin, L., Smith, A.: Reusable fuzzy extractors for low-entropy distributions. J. Cryptol. **34**(1), 1–33 (2021)
23. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM **51**(4), 557–594 (2004)
24. Canetti, R., Rothblum, G.N., Varia, M.: Obfuscation of hyperplane membership. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 72–89. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_5
25. Canetti, R., Varia, M.: Non-malleable obfuscation. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 73–90. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00457-5_6
26. Coretti, S., Dodis, Y., Guo, S.: Non-uniform bounds in the random-permutation, ideal-cipher, and generic-group models. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10991, pp. 693–721. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96884-1_23
27. Cousins, D.B., et al.: Implementing conjunction obfuscation under entropic ring LWE. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 354–371. IEEE (2018)
28. Couteau, G., Katsumata, S., Ursu, B.: Non-interactive zero-knowledge in pairing-free groups from weaker assumptions. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020. LNCS, vol. 12107, pp. 442–471. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45727-3_15
29. Cramer, R., Dodis, Y., Fehr, S., Padró, C., Wichs, D.: Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 471–488. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_27
30. Crépeau, C.: Efficient cryptographic protocols based on noisy channels. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 306–317. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-69053-0_21
31. Demarest, L., Fuller, B., Russell, A.: Code offset in the exponent. In: 2nd Conference on Information-Theoretic Cryptography (ITC 2021). Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2021)
32. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient public-key cryptography in the presence of key leakage. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 613–631. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_35
33. Dodis, Y., Kanukurthi, B., Katz, J., Reyzin, L., Smith, A.: Robust fuzzy extractors and authenticated key agreement from close secrets. IEEE Trans. Inf. Theory **58**(9), 6207–6222 (2012)
34. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008)
35. Dodis, Y., Wichs, D.: Non-malleable extractors and symmetric key cryptography from weak secrets. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, pp. 601–610. ACM (2009)
36. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. J. Cryptol. **30**(1), 242–288 (2017)

37. Feng, H., Tang, Q.: Computational robust (fuzzy) extractors for CRS-dependent sources with minimal min-entropy. In: Nissim, K., Waters, B. (eds.) TCC 2021. LNCS, vol. 13043, pp. 689–717. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90453-1_24
38. Fenteany, P., Fuller, B.: Same point composable and nonmalleable obfuscated point functions. In: Conti, M., Zhou, J., Casalicchio, E., Spognardi, A. (eds.) ACNS 2020. LNCS, vol. 12147, pp. 124–144. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57878-7_7
39. Galbraith, S.D., Zobernig, L.: Obfuscated fuzzy hamming distance and conjunctions from subset product problems. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019. LNCS, vol. 11891, pp. 81–110. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36030-6_4
40. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, pp. 40–49. IEEE (2013)
41. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. SIAM J. Comput. **45**(3), 882–929 (2016)
42. Gentry, C., Lewko, A.B., Sahai, A., Waters, B.: Indistinguishability obfuscation from the multilinear subgroup elimination assumption. In: 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS), pp. 151–170. IEEE (2015)
43. Goyal, R., Koppula, V., Waters, B.: Lockable obfuscation. In: 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), pp. 612–621. IEEE (2017)
44. Komargodski, I., Yogev, E.: Another step towards realizing random oracles: non-malleable point obfuscation. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 259–279. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_10
45. Komargodski, I., Yogev, E.: Another step towards realizing random oracles: Non-malleable point obfuscation. Cryptology ePrint Archive, Report 2018/149 (2018). Version 20190226:074205, https://eprint.iacr.org/2018/149
46. Lynn, B., Prabhakaran, M., Sahai, A.: Positive results and techniques for obfuscation. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 20–39. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_2
47. Nisan, N., Zuckerman, D.: Randomness is linear in space. J. Comput. Syst. Sci. **52**(1), 43–52 (1996)
48. Pass, R., Seth, K., Telang, S.: Indistinguishability obfuscation from semantically-secure multilinear encodings. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 500–517. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_28
49. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, pp. 475–484. ACM (2014)
50. Simhadri, S., Steel, J., Fuller, B.: Cryptographic authentication from the iris. In: Lin, Z., Papamanthou, C., Polychronakis, M. (eds.) ISC 2019. LNCS, vol. 11723, pp. 465–485. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30215-3_23

51. Vadhan, S.P., et al.: Pseudorandomness, vol. 7. Now Delft (2012)
52. Varia, M.H.: Studies in program obfuscation. PhD thesis, Massachusetts Institute of Technology (2010)
53. Wen, Y., Liu, S.: Robustly reusable fuzzy extractor from standard assumptions. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018. LNCS, vol. 11274, pp. 459–489. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03332-3_17
54. Wen, Y., Liu, S., Gu, D.: Generic constructions of robustly reusable fuzzy extractor. In: Lin, D., Sako, K. (eds.) PKC 2019. LNCS, vol. 11443, pp. 349–378. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17259-6_12
55. Wen, Y., Liu, S., Hu, Z., Han, S.: Computational robust fuzzy extractor. Comput. J. **61**(12), 1794–1805 (2018)
56. Wichs, D., Zirdelis, G.: Obfuscating compute-and-compare programs under LWE. In: 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), pp. 600–611. IEEE (2017)