Relaxed Two-to-one Recoding Schemes

Omkant Pandey* Kim Ramchen† Brent Waters ‡

Abstract

A *two-to-one recoding* (TOR) scheme is a new cryptographic primitive, proposed in the recent work of Gorbunov, Vaikuntanathan, and Wee (GVW), as a means to construct attribute-based encryption (ABE) schemes for all boolean circuits. GVW show that TOR schemes can be constructed assuming the hardness of the learning-with-errors (LWE) problem.

We propose a slightly weaker variant of TOR schemes called *correlation-relaxed two-to-one recoding* (CR-TOR). Unlike the TOR schemes, our weaker variant does not require an encoding function to be pseudorandom on correlated inputs. We instead replace it with an indistinguishability property that states a ciphertext is hard to decrypt without access to a certain encoding. The primary benefit of this relaxation is that it allows the construction of ABE for circuits using the TOR paradigm from a broader class of cryptographic assumptions.

We show how to construct a CR-TOR scheme from the noisy cryptographic multilinear maps of Garg, Gentry, and Halevi as well as those of Coron, Lepoint, and Tibouchi. Our framework leads to an instantiation of ABE for circuits that is conceptually different from the existing constructions.

1 Introduction

Encrypting data using traditional public-key encryption results in a very coarse-grained access to the data, since only those who possess an appropriate secret-key can decrypt the resulting ciphertext. Attribute-based encryption (ABE), introduced by Sahai and Waters [26] is an emerging class of cryptosystems which allow for significantly more fine-grained access to data. There are two variants of ABE cryptosystems [16]: Key-Policy ABE and Ciphertext-Policy ABE. In Key-Policy ABE, the secret-keys SK_f have an associated boolean-function f called the policy. The messages are encrypted under an assignment x of boolean variables called the attributes. A secret-key SK_f can decrypt a message M encrypted under assignment x if and only if $f(x) = 1$. In Ciphertext-Policy ABE, these roles are reversed: secret-keys are associated with assignments x and ciphertexts are associated with policies f.

Recently, two independent works due to Garg, Gentry, Halevi, Sahai, and Waters [13], and Gorbunov, Vaikuntanathan, and Wee [15] showed how to construct ABE schemes for general circuits. More specifically, these works show how to realize the class of access policies f that can be expressed as a boolean circuit of depth d and input length n; both d and n are fixed at the system setup and can be polynomial in the security

*University of Illinois, Urbana-Champaign. Email: omkant@uiuc.edu. Part of this work was done while the author was at The University of Texas at Austin.

†The University of Texas at Austin. Email: kramchen@cs.utexas.edu

‡The University of Texas at Austin. Email: bwaters@cs.utexas.edu Supported by NSF CNS-0915361 and CNS-0952692, CNS-1228599 DARPA through the U.S. Office of Naval Research under Contract N00014-11-1-0382, DARPA N11AP20006, Google Faculty Research award, the Alfred P. Sloan Fellowship, Microsoft Faculty Fellowship, and Packard Foundation Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Defense or the U.S. Government.

parameter; the size of ciphertexts and public-parameters is at most polynomial in d and n but independent of the size of the circuits in the class. The construction of [13] uses noisy cryptographic multilinear maps of Garg, Gentry, and Halevi [12], and is based on a new assumption in ideal lattices. The construction of [15] is based on the (standard) learning-with-errors assumption [23]. Prior to these works, the construction of [16] supported the largest class of access policies until now; namely the policies corresponding to polynomial sized boolean formulas, or equivalently circuits in the complexity class $\mathbf{NC^1}$.

Two-to-one recoding schemes. The work of GVW on attribute-based encryption introduces an interesting new framework called *two-to-one recoding* (TOR) schemes. Roughly speaking, a TOR scheme resembles a proxy re-encryption scheme [4]: it has an "encoding" mechanism with the following functionality. Given the encodings of a message m under two different public-keys pk_1 and pk_2, and an appropriate trapdoor t, it is possible to obtain the encoding of m under a third public-key pk_3. The trapdoor t, called the recoding key, can be generated using any one of the secret-keys corresponding to pk_1 or pk_2. GVW show that if such a primitive satisfies several additional simulatability and indistinguishability properties (described later), then circuit ABE can be constructed in a black box manner.

TOR schemes are intriguing primitive that we find interesting at least for two reasons. First, because it immediately yields a (black-box) construction of circuit ABE. And second, *how* it yields ABE construction. Roughly speaking, the TOR encodings and recoding-keys are used imitate the circuit-computation along the lines of garbled-circuits [29]. This ability to execute a circuit computation "securely and in a tamper-proof manner" makes TOR a powerful primitive.

Relaxing the requirements of TOR schemes. In this work, we take a closer look at TOR as an independent primitive. In particular, we investigate the possibility of building TOR schemes from assumptions that are different from LWE.

Our focal point is the *correlated pseudorandomness* property [24] which states that "the output of the encoding function on several correlated inputs looks pseudorandom.' While this property follows naturally from the LWE construction of GVW, it proves to be significantly more difficult to achieve in other contexts. For instance, we found that it was possible to achieve TOR in generic multilinear maps using a natural generalization of the "matrix DDH" assumption [21]. However, this assumption assumption is actually false in the framework of GGH [12]. In addition, while it remains plausible in the framework of CLT [11], the resulting construction encumbers significant additional overhead to the existing multilinear construction of Garg, Gentry, Halevi, Sahai, and Waters [13]. Ideally, we would like an abstraction that both leads to circuit ABE from a broader range of assumptions and one which naturally leads to competitive constructions.

With this goal in mind we reexamine how correlated pseudorandomness was used in the GVW construction. In the GVW Circuit ABE construction multiple TOR primitives for each input gate for each interior gate in a private key circuit for f. However, in their proof the correlated randomness property is actually not needed or used at any of these gates except the final output gate. This follows from the fact that in the circuit ABE construction there is no real reason to hide from an attacker that two encodings are generated from the same randomness — the attacker naturally knows this anyway for a well formed ciphertext. The correlated randomness property is used in combination with a one time encryption property at the output gate to show that if an attacker cannot derive the encoding, then he cannot decrypt a message.

Our goal is to present a relaxed formulation of two-to-one reencoding that more directly meets this intuitive security goal. We aim to replace correlated pseudorandomness with a security goal that more tightly meets what is needed to construct Circuit ABE systems.

Our contributions. We first present a relaxed formulation called *correlation-relaxed two-to-one recoding* (CR-TOR) schemes. In this framework the encodings are not required to be pseudorandom on correlated inputs. Rather, we capture the corresponding security requirement by an indistinguishability game which specifies only that there exists an encryption function producing an indistinguishable ciphertexts which can be decrypted by "appropriately computed" encodings. In terms of circuit ABE, an "appropriately computed" encoding will be the recoding corresponding to the output of the circuit. After presenting our formulation of CR-TOR, we show that it is sufficient to build circuit ABE in a black-box manner.

Next, we consider the question of constructing CR-TOR and TOR schemes from assumptions different from LWE. For this purpose, we turn to the framework of idealized multilinear maps [6, 12, 11, 13], and show how to construct a:

- CR-TOR scheme based on a natural generalization of the DDH assumption;
- TOR scheme based on the "matrix DDH" assumption (in groups with multilinear maps)

We note that the construction of CR-TOR is much more efficient compared to the corresponding construction of TOR (which requires us to use matrix DDH assumption). This indicates that correlated pseudorandomness property of TOR comes at a price in efficiency.

As of today, no constructions of idealized multilinear maps are known. However, the breakthrough work of Garg, Gentry, and Halevi [12], as well as the recent followup work of Coron, Lepoint, and Tibouchi [11], constructs randomized encoding schemes which can be seen as candidate constructions for "approximate" multilinear maps. These constructions are based on new cryptographic assumptions on ideal lattices.

We show that our construction of CR-TOR scheme can be easily adapted to work in the framework of both GGH [12] and CLT [11]. Moreover, the performance of the resulting constructions is roughly on par with the GGHSW constructions [13].

However, our construction of the TOR scheme can only work with the framework of CLT. This is because the matrix DDH assumption does *not* hold in the GGH setting; but it remains plausible in the CLT setting. Furthermore, the overhead is significantly increased compared to the CR-TOR systems. The additional overhead can be directly attributed to achieving the stronger (and unused) correlation resistance property.

Finally, we note that since CR-TOR suffices to obtain circuit ABE in a black-box manner, we obtain a new construction of circuit ABE that is distinct from both GVW[15] and GGHSW [13]. At a conceptual level, this construction resembles the GVW construction since it is obtained from CR-TOR; on the other hand, it uses multilinear maps as its internal mechanism for computation, resulting in the same underlying assumption as the GGHSW construction. We remark that the construction of circuit ABE in all of these works, including ours, are in the selective-security model [5, 16].

Goals and Non-Goals. One of our main objectives in this work is to understand which properties of TOR are crucial to build circuit-ABE and eliminate the unnecessary ones. Specifically, we have investigated the correlated pseudorandomness property and find that it might be unwarranted for circuit-ABE, resulting in unnecessary inefficiencies. This argument is supported by constructing a circuit-ABE scheme which compares favourably to existing schemes [13, 15] in terms of efficiency. However, *building a new and more efficient circuit-ABE scheme is* not *a goal* of this paper. We do so only to demonstrate that correlated pseudorandomness property is not required for circuit-ABE; relaxed-TOR is sufficient and enables a better construction.

Related works. After the introduction of ABE, while limited progress was made on expanding the class of access policies f, significant progress was made in many directions on ABE. New proof techniques were

developed in [27, 18, 22, 8, 1, 3, 20, 2] to diversify the underlying security assumptions based on both bilinear pairings as well as lattices. New constructions for decentralizing trust in the key-issuing authority were proposed in [9, 10, 19]. In addition, schemes supporting policies of different flavors were also developed such as: inner-product policy [17], regular languages [28], branching programs [7], and more expressive schemes in the (much weaker) "bounded collusion" model [25, 14].

Paper organization. We will start by recalling the setting of idealized multilinear maps and attribute-based encryption in Section 2. We provide a definition of our correlation-relaxed TOR in the next section 3, followed by a black-box construction of ABE from CR-TOR in Section 4. We conclude by presenting a construction of our CR-TOR in Section 5. Due to space constraints, the construction of the original TOR (with strong correlation-psuedorandomness property) is given in appendix A. Finally, in Appendix B we describe how to translate our construction in the framework of graded encoding schemes of GGH.

2 Preliminaries

In this section we recall the setting of multilinear maps, hardness assumptions, and definitions for circuit ABE. We follow the conventions established in [15, 13].

2.1 Multilinear maps

We first recall the setting of ideal multilinear maps. Following [13], we assume the existence of a group generator $\mathcal{G}$, which takes as input a security parameter λ and a positive integer d to indicate the number of allowed pairing operations. $\mathcal{G}(1^\lambda, d)$ outputs a sequence of groups $\vec{\mathbb{G}} = (\mathbb{G}_1, \ldots, \mathbb{G}_d)$ each of large prime order $p > 2^\lambda$. Let g_i be a canonical generator of $\mathbb{G}_i$ publicly known from group's description, and let $g = g_1$.

We assume the existence of a set of efficiently computable bilinear maps $\{e_{i,j} : \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j} | i, j \geq 1; i + j \leq d\}$. The map $e_{i,j}$ satisfies the following relation:

$$e_{i,j}(g_i^a, g_j^b) = g_{i+j}^{ab} : \forall a, b \in \mathbb{Z}_p.$$

A consequence of this is that $e_{i,j}(g_i, g_j) = g_{i+j}$. When the context is obvious, we will sometimes abuse notation and drop the subscripts i, j. For example, we may simply write:

$$e(g_i^a, g_j^b) = g_{i+j}^{ab}.$$

Assumption 1. (d-Multilinear Decisional Diffie-Hellman (d-MDDH) assumption) Suppose that a challenger runs $\mathcal{G}(1^\lambda, d)$ and generates groups $(\mathbb{G}_1, \ldots, \mathbb{G}_d)$ of prime order p with generators $(g_1, \ldots, g_d)$. Then, the d-MDDH assumption states that the advantage $Adv_{\mathcal{A}}(\lambda)$ of every polynomial time adversary $\mathcal{A}$, defined below, is at most negligible in λ:

$$|\Pr[\mathcal{A}(g, g^s, g^{c_1}, \ldots, g^{c_d}, g_d^{sc_1 \ldots c_d}) = 1] - \Pr[\mathcal{A}(g, g^s, g^{c_1}, \ldots, g^{c_d}, g_d^u) = 1]|$$

where $s, c_1, \ldots, c_k$ and u are uniformly distributed in $\mathbb{Z}_p$.

This is a natural generalization of the DDH assumption in the multilinear setting. Intuitively, this assumption is plausible because there are $d + 1$ element multiplications in the exponent, which cannot be computed using a d-linear map.

We will describe our constructions in this ideal setting first. However, later we will show how to adapt them to the noisy settings of GGH and CLT [12, 11].

2.2 Attribute Based Encryption

The definition of ABE provided here is for the key-policy variant of ABE, where the secret-keys are generated for a circuit C, and the ciphertexts are encrypted under a "set of attributes" denoted by an *index* $\mathtt{ind} \in \{0,1\}^l$.

ABE for circuits. An ABE scheme for a class of circuits $\mathcal{C}$ is a tuple of algorithms $\mathcal{ABE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ where:

- $\mathsf{Setup}(1^\lambda, l, n)$ The setup algorithm takes as input the security parameter λ, the length l of the index $\mathtt{ind}$, and a bound n on circuit depth; it outputs public parameters pp and the master key msk.
- $\mathsf{Enc}(pp, \mathtt{ind} \in \{0,1\}^l, m)$ The encryption algorithm takes as input the public parameters pp, a bit string $\mathtt{ind} \in \{0,1\}^l$ representing the assignment of boolean variables (a.k.a. "attributes"), and a message m. It outputs a ciphertext ct.
- $\mathsf{KeyGen}(msk, C)$ The key generation algorithm takes as input the master key msk and the description of a circuit C of maximum depth n. It outputs a secret-key sk_C.
- $\mathsf{Decrypt}(sk_C, ct)$. The decryption algorithm takes as input a secret key sk_C and ciphertext ct. The algorithm attempts to decrypt and outputs a message m if successful; otherwise it outputs a special symbol $\perp$.

Correctness. It is required that for all pp and msk produced by algorithm Setup, for all $\mathtt{ind} \in 0,1^l$, all messages m, for all appropriate circuits C such that $C(\mathtt{ind}) = 1$, if $\mathsf{KeyGen}(msk, C) \rightarrow sk_C$ and $\mathsf{Enc}(pp, \mathtt{ind}, m) \rightarrow ct$ then: $\mathsf{Dec}(sk_C, ct) = m$.

Selective security game for ABE. The selective-security game [16, 13, 5] for ABE proceeds in following stages between an adversary $\mathcal{A}$ and a challenger:

- INIT The adversary declares an index $\mathtt{ind}^*$
- SETUP The challenger runs the Setup algorithm and gives the public-parameters to the adversary.
- PHASE 1 The adversary adaptively makes secret-key queries for several circuit C_j such that $C_j(\mathtt{ind}^*) = 0$ for every j. The challenger answers each query by running the KeyGen algorithm using the master secret-key.
- CHALLENGE The adversary submits two challenge messages m_0 and m_1 of equal length. The challenger flips a bit b and sends an encryption of m_b under the index $\mathtt{ind}^*$ to the adversary.
- PHASE 2 Phase 1 is repeated.
- GUESS The adversary outputs a guess b'.

The *advantage* of the adversary $\mathcal{A}$ in the selective-security game is defined as $\left|\Pr\left[b' = b\right] - \frac{1}{2}\right|$. We say that an ABE scheme is selectively-secure if the advantage of every polynomial time adversary $\mathcal{A}$ in the above game is at most negligible.

3 Correlation-relaxed Two-to-one Recoding Schemes

In this section we will define our relaxation of the original TOR scheme of [15]. Let us first recall some salient features of the scheme. A TOR scheme defines a probabilistic algorithm $\mathsf{Encode}(\cdot,\cdot)$ whose first input is a public key, and whose second input is a tag, from some tag set $\mathcal{S}$. Additionally there is a "two-to-one" recoding algorithm with the following property: for any tuple of public keys (pk_0, pk_1, pk_{tgt}) and any $s \in \mathcal{S}$, there exists a recoding key rk such that the recoding algorithm performs the following transformation

$$(\mathsf{Encode}(pk_0, s), \mathsf{Encode}(pk_1, s)) \xrightarrow{rk} \mathsf{Encode}(pk_{tgt}, s)$$

There is an algorithm to generate the recoding key using either sk_0 or sk_1, such that the key has the same distribution in either case. Additionally there is an algorithm to simulate a fake recoding key/public key pair for any input keys pk_0 and pk_1. The fake pair (rk, pk_{tgt}) should be indistinguishable from that generated honestly by the recode key generation algorithm for a random pk_{tgt}. Finally "correlated pseudorandomness" states that given polynomially many encodings of tag s under distinct public keys, an encoding under a fresh public key is indistinguishable from random.

Our relaxation. We now describe the core features of our relaxation. Firstly we remove the requirement for "correlated pseudorandomness", paving the way for construction of secure ABE from new assumptions. In doing so we introduce a message encryption function whose random input is precisely the tag s, i.e. the function is deterministic once s is picked. Additionally our scheme also generates encodings deterministically.

Looking ahead to our ABE scheme in the next section, we will see that the encryption function only uses randomness when sampling a tag. Therefore ABE from correlation relaxed TOR can use a reduced entropy pool, which is useful when encryption is performed on embedded systems. However one consequence is that our key generation algorithm must generate "levelled" public keys. Intuitively the reason is that in the original TOR scheme, encodings under distinct public keys are unnrelated, whereas in the relaxed scheme encodings at given level are all re-randomized versions of a specific encoding.

Finally, we capture security of correlation relaxed TOR by an indistinguishability experiment; *indistinguishability of encoding derived ciphertexts* (IND-EDC). The game specifies that the encrypted messages are indistinguishable given polynomially many encodings of the tag.

The definition. A *correlation-relaxed two-to-one recoding* (CR-TOR) scheme over an input space $\mathcal{S} = \mathcal{S}_\lambda$ is a tuple of eight polynomial time algorithms (Params, Keygen, Encode, $\mathsf{ReKeyGen}$, $\mathsf{SimReKeyGen}$, Recode, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$). The first three algorithms define a mechanism for encoding the input as follows:

- $\mathsf{Params}(1^\lambda, d)$ is a probabilistic algorithm that takes as input the security parameter λ and an upper bound d on the number of recoding operations; it outputs the global public parameters pp.
- $\mathsf{Keygen}(pp, i)$ is a probabilistic algorithm that takes as input the public parameters pp, an index i called the *level index*; it outputs a public/secret key pair (pk, sk). When $i = d$ only, the algorithm is deterministic and outputs a unique public/secret key pair.
- $\mathsf{Encode}(pk, s)$ is a deterministic algorithm that takes as input a public-key pk and an input $s \in \mathcal{S}_\lambda$ to be encoded; it outputs ψ which is called an encoding of s. Input s is sometimes referred to as the *tag* or the *secret*.

The next three algorithms provide two different mechanisms to generate recoding-keys, and a recoding mechanism as follows:

- $\mathsf{ReKeyGen}(pp, i, pk_0, pk_1, sk_0, pk_{\text{tgt}})$ is a probabilistic algorithm that takes as input the public parameters pp, a level index i, a key pair (pk_0, sk_0), another public key pk_1, and a "target" public key pk_{tgt}; it outputs a trapdoor rk called the *recoding key*.
- $\mathsf{SimReKeyGen}(pp, i, pk_0, pk_1)$ is a probabilistic algorithm that takes as input public parameters pp, a level index i, and two public-keys pk_0, pk_1; it outputs a recoding-key rk together with a "target" public key pk_{tgt}.
- $\mathsf{Recode}(rk, \psi_0, \psi_1)$ is a deterministic algorithm that takes as input a recoding key rk, and two encodings ψ_0, ψ_1; it outputs an encoding ψ_{tgt}.

Finally, the last two algorithms define a symmetric encryption scheme with the following properties:

- $\mathsf{Encrypt}(pp, m; s)$ is a probabilistic algorithm which takes as input the public parameters pp, a message m (from a well-defined message space $\mathcal{M}$) and a tag $s \in \mathcal{S}$ as random coins; it outputs a ciphertext τ.
- $\mathsf{Decrypt}(pp, \psi_{\text{out}}, \tau)$ is a deterministic algorithm which takes as input the public parameters pp, an encoding ψ_{out}, and a ciphertext τ; it produces a message $m \in \mathcal{M}$.

In addition, the following requirements must be satisfied.

Correctness. At a high level, correctness states that each properly generated recoding-key works correctly for input encodings ψ_0, ψ_1. Since encodings are generated under public-keys, and public-keys are generated for a given level-index i,[1] stating this requirement is somewhat notation-heavy. In addition, we will have the correctness requirement on the encrypt and decrypt algorithms.

Formally, the first requirement is stated as follows. For every λ, d, every $pp \leftarrow \mathsf{Params}(1^\lambda, d)$, and every pk generated for index $i < d$ (i.e. $(pk, sk) \leftarrow \mathsf{Keygen}(pp, i)$), and every tag $s \in \mathcal{S}$ there exists a set $\Psi_{pk,s}$ satisfying the following condition. Suppose that (pk_0, sk_0) and (pk_1, sk_1) are generated by $\mathsf{Keygen}(pp, i)$ for index i, and $(pk_{\text{tgt}}, sk_{\text{tgt}})$ by $\mathsf{Keygen}(pp, i+1)$ for the index $i+1$. Then, for all $\psi_0 \in \Psi_{pk_0,s}, \psi_1 \in \Psi_{pk_1,s}$ and $rk \leftarrow \mathsf{ReKeyGen}(pp, i, pk_0, pk_1, sk_0, pk_{\text{tgt}})$, it holds that $\mathsf{Recode}(rk, \psi_0, \psi_1) \in \Psi_{pk_{\text{tgt}},s}$.

The second requirement is as follows. Let $(pk_{\text{out}}, sk_{\text{out}}) \leftarrow \mathsf{Keygen}(pp, i = d)$. Then, for all $m \in \mathcal{M}$, $s \in \mathcal{S}$, $\psi_{\text{out}} \in \Psi_{pk_{\text{out}},s}$, it holds that $\mathsf{Decrypt}\,(pp, \psi_{\text{out}}, \mathsf{Encrypt}(pp, m; s)) = m$.

Key indistinguishability. Let $i < d$, and $(pk_b, sk_b) \leftarrow \mathsf{Keygen}(pp, i)$, and $(pk_{\text{tgt}}, sk_{\text{tgt}}) \leftarrow \mathsf{Keygen}(pp, i+1)$. Then, the following two ensembles must be statistically close:[2]

$$\begin{aligned}&\big[Aux, \mathsf{ReKeyGen}(pp, i, pk_0, pk_1, sk_0, pk_{\text{tgt}})\big] \equiv_s\\ &\big[Aux, \mathsf{ReKeyGen}(pp, i, pk_0, pk_1, sk_1, pk_{\text{tgt}})\big]\end{aligned}$$

where $Aux = ((pk_0, sk_0), (pk_1, sk_1), (pk_{\text{tgt}}, sk_{\text{tgt}}))$.

[1] This is another minor deviation in our definition from original TOR; it can be seen as an additional weakening. We will avoid subscripting each pk with its level index i when clear from the context.

[2] Computational indistinguishability may also be sufficient.

Recoding Simulation. Let $i < d$. Let $(pk_b, sk_b) \leftarrow \mathsf{Keygen}(pp, i)$ for $b = 0, 1$. Then the following two ensembles are statistically close:

$$\left[Aux, pk_{\text{tgt}}, rk : (pk_{\text{tgt}}, sk_{\text{tgt}}) \leftarrow \mathsf{Keygen}(pp, i+1), rk \leftarrow \mathsf{ReKeyGen}(pp, i, pk_0, pk_1, sk_0, pk_{\text{tgt}})\right] \equiv_s$$
$$\left[Aux, pk_{\text{tgt}}, rk : (pk_{\text{tgt}}, rk) \leftarrow \mathsf{SimReKeyGen}(pp, i, pk_0, pk_1)\right]$$

where $Aux = ((pk_0, sk_0), (pk_1, sk_1))$.

The above two properties are statistical properties and identical to the properties of original TOR scheme. We now describe the third property called *indistinguishability of encoding derived ciphertexts* or IND-EDC. This is a computational property; recall that the original TOR formulation had *correlated pseudorandomness* which is stronger than IND-EDC.

Indistinguishability of Encoding Derived Ciphertexts (IND-EDC). We require that the advantage of every polynomial time adversary $\mathcal{A}$ in the IND-EDC game is at most negligible where the IND-EDC game proceeds as follows and the *advantage* of $\mathcal{A}$ is defined as $\left|\Pr\left[b' = b\right] - \frac{1}{2}\right|$ (see below):

- The challenger sends $(pp, pk_1, \ldots, pk_\ell)$ to the adversary where: $pp \leftarrow \mathsf{Params}(1^\lambda, d)$, $(pk_j, sk_j) \leftarrow \mathsf{Keygen}(pp, 1)$ for $j = 1, \ldots, \ell = \text{poly}(\lambda)$
- Adversary sends two equal length messages m_0, m_1.
- Challenger samples a random bit b and secret tag $s \in \mathcal{S}$. It sends $(\psi_1, \ldots, \psi_\ell, \tau_b)$ where $\psi_j \leftarrow \mathsf{Encode}(pk_j, s)$ for every $j \in [\ell]$, and $\tau_b \leftarrow \mathsf{Encrypt}(pp, m_b; s)$.
- Adversary outputs a bit b' and halts.

4 Circuit ABE from correlation-relaxed TOR

In this section we construct ABE for circuits from correlation-relaxed TOR. The construction is very similar to the GVW construction of ABE from TOR [15] except that in proving security, instead of using correlation pseudorandomness, we will use IND-EDC property.

Circuits are described using the same convention as in [15], which as follows. Without loss of generality, we consider the class of circuits $\mathcal{C} = \{\mathbf{C}_\lambda\}_{\lambda \in \mathbb{N}}$ where each circuit $C \in \mathbf{C}_\lambda$ is a *layered* circuit consisting of input wires, gates, internal wires, and a single output wire. Recall that in a layered circuits gates are arranged in layers where every gate at a given layer has a pre-specified depth. The lowest row has depth 1 and depth increases by one as we go up. A gate at depth i receives both of its inputs from wires at depth $i - 1$. The circuit has $l = l(\lambda)$ input wires, numbered from 1 to l. The size of the circuit is denoted by $|C|$, and all internal wires are indexed from $l + 1, \ldots, |C| - 1$; the output wire has index $|C|$. Every gate is a boolean-gate with exactly two input wires and one output wire.

Our construction of ABE from a CR-TOR scheme follows.

The construction. Suppose that the algorithms of the given CR-TOR scheme are: (Params, Keygen, Encode, ReKeyGen, SimReKeyGen, Recode, Encrypt, Decrypt). The algorithms of our ABE scheme $\mathcal{ABE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ are as follows.

- $\mathsf{Setup}(1^\lambda, l, d)$: The setup algorithm for ABE first runs the parameter generation algorithm of CR-TOR to obtain global public-parameters: $pp \leftarrow \mathsf{Params}(1^\lambda, d)$. Then, for each input wire $i \in [l]$, it generates two fresh public and secret key pairs, and an additional pair for the output wire:

$$(pk_{i,b}, sk_{i,b}) \leftarrow \mathsf{Keygen}(pp, 1) \text{ for } i \in [l], b \in \{0,1\}$$
$$(pk_{\text{out}}, sk_{\text{out}}) \leftarrow \mathsf{Keygen}(pp, d)$$

  It outputs the master public-key and master secret-key pair (mpk, msk) as follows (note that secret-key sk_{out} is not used):

$$mpk := pp, pk_{\text{out}}, \{pk_{i,b}\}_{i\in[l],b\in\{0,1\}}, msk := \{sk_{i,b}\}_{i\in[l],b\in\{0,1\}}.$$

- $\mathsf{Enc}(mpk, \mathtt{ind}, m)$: Let $\mathtt{ind} = (\mathtt{ind}_1, \ldots, \mathtt{ind}_l) =\in \{0,1\}^l$. The algorithm chooses a uniform $s \xleftarrow{\$} S$, encodes it under the public-keys specified by the bits of $\mathtt{ind}$, and finally encrypts m under pp and s. That is,

$$\psi_i \leftarrow \mathsf{Encode}(pk_{i,\mathtt{ind}_i}, s), \forall i \in [l], \text{ and } \tau \leftarrow \mathsf{Encrypt}(pp, m; s),$$

  The algorithm outputs $ct_{\mathtt{ind}} = (\mathtt{ind}, \psi_1, \ldots, \psi_l, \tau)$ as the ciphertext.

- $\mathsf{KeyGen}(msk, C)$: The algorithm proceeds in two steps:

  1. For every non-input wire $w \in \{l+1, \ldots, |C|\}$ of the circuit C, it generates two public-secret key pairs denoting two possible values $b \in \{0,1\}$ for this wire. However, the public-key corresponding to the circuit-output 1 is (always) set to pk_{out}. That is, for every $w \in \{l+1, \ldots, |C|\}$ and every $b \in \{0,1\}$ such that $(w,b) \neq (|C|, 1)$, generate: $(pk_{w,b}, sk_{w,b}) \leftarrow \mathsf{Keygen}(pp, i)$, where i is the depth of wire w; then set $pk_{|C|,1} = pk_{\text{out}}$.
  2. For every gate $g := (u, v, w)$ at level i—where (u, v) are two incoming wires of g and w is its outgoing wire—compute four recoding-keys $rk^w_{b,c}$ for wire w as follows:

$$rk^w_{b,c} \leftarrow \mathsf{ReKeyGen}(pp, i, pk_{u,b}, pk_{v,c}, sk_{u,b}, pk_{w,g_w(b,c)})$$

     where $g_w(b,c)$ denotes the output of g on input (b,c).

  The algorithm outputs the secret key sk_C which is a collection of all $4(|C| - l)$ recoding keys it has computed (along with the circuit C).

$$sk_C := C, \left(rk^w_{b,c} : w \in [l+1, |C|], b \in \{0,1\}, c \in \{0,1\}\right).$$

- $\mathsf{Dec}(sk_C, ct_{\mathtt{ind}})$: If $C(\mathtt{ind}) = 0$, algorithm outputs $\perp$. Otherwise, $C(\mathtt{ind}) = 1$ defines a computation of the circuit where each wire carries a well defined value in $\{0,1\}$. In particular, an input wire $w \in \{1, \ldots, l\}$ carries the bit $\mathtt{ind}_w$, and every other wire $w \in \{l+1, \ldots, |C|\}$ carries a bit as follows. Suppose w is the outgoing wire of (uniquely defined) gate $g := (u, v, w)$, and wires u and v carry values b^* and c^* respectively; then w carries the value $d^* = g_w(b^*, c^*)$. For every wire, the decryption algorithm computes:

$$\psi_{w,d^*} \leftarrow \mathsf{Recode}\left(rk^w_{b^*,c^*}, \psi_{u,b^*}, \psi_{v,c^*}\right)$$

  using appropriate values from the ciphertext $ct_{\mathtt{ind}}$ and the key sk_C. Note that since $C(\mathtt{ind}) = 1$, the algorithm must have also computed an encoding $\psi_{\text{out}} \in \Psi_{pk_{\text{out}},s}$ corresponding to the output wire. The decrypted message is: $m \leftarrow \mathsf{Decrypt}(pp, \psi_{\text{out}}, \tau)$.

Theorem 1. *Scheme $\mathcal{ABE}$ described above is a selectively-secure ABE scheme for all polynomial size circuits as per the definition in section 2.*

Proof. To prove the theorem, we show that if there exists a PPT adversary $\mathcal{A}$ breaking the selective security of $\mathcal{ABE}$ with noticeable advantage, then there exists a PPT $\mathcal{B}$ winning the IND-EDC game with noticeable advantage (against the underlying CR-TOR scheme). The construction of $\mathcal{B}$, called the simulator, proceeds as follows.

Simulator $\mathcal{B}$. The simulator participates in the IND-EDC game with an outside challenger. At the same time, internally, it plays the selective-security game with $\mathcal{A}$ as follows. $\mathcal{B}$ runs $\mathcal{A}$ answering its queries in various stages as follows.

INIT. It receives an index $\mathtt{ind}^*$ from $\mathcal{A}$.

SETUP. In this phase, first the simulator $\mathcal{B}$ asks the challenger of IND-EDC game to send $(pp, pk_1, \ldots, pk_l, pk_{\text{out}})$. Then, it prepares the parameters for $\mathcal{A}$ as follows. It defines $pk_{i,\mathtt{ind}_i^*} = pk_i$, and generates the remaining keys as: $(pk_{i,1-ind_i^*}, sk_{i,1-ind_i^*}) \leftarrow \mathsf{Keygen}(pp, 1)$. It sends mpk to $\mathcal{A}$ where:

$$mpk := pp, pk_{\text{out}}, \{pk_{i,b}\}_{i\in[l],b\in\{0,1\}}\,.$$

Note that the mpk is well defined and distributed identically to the output of the actual setup algorithm of $\mathcal{ABE}$.

PHASE 1. In this phase $\mathcal{A}$ submits polynomially many secret-key queries for various circuits. Let C be one such query, then by definition of the game, $C(\mathtt{ind}^*) = 0$. The computation $C(\mathtt{ind}^*)$ defines a unique value carried by each wire of C. $\mathcal{B}$ generates the simulated-key for C as follows.

- For each wire $w \in [l+1, |C|-1]$ generate $(pk_{w,1-b^*}, sk_{w,1-b^*}) \leftarrow \mathsf{Keygen}(pp, i)$, where i is the depth of w and b^* is the bit it carries in computation $C(\mathtt{ind}^*)$. Define $pk_{|C|,1} = pk_{\text{out}}$.
- For every gate $g = (u, v, w)$ do the following (here i is the depth of g, and b^*, c^* are the bits carried by its incoming wires u, v in computation $C(\mathtt{ind}^*)$):
  1. $pk_{w,g(b^*,c^*)}, rk^w_{b^*,c^*} \leftarrow \mathsf{SimReKeyGen}(pp, i, pk_{u,b^*}, pk_{v,c^*})$. Note that at this point, two public-keys for each wire in C have been fixed including the output wire.[3] This step also fixes one recode-key for each wire corresponding to the computation $C(\mathtt{ind}^*)$. The remaining 3 recode-keys for each wire are sampled in the next step.
  2. For $(b, c) \in \{0,1\}^2 \backslash (b^*, c^*)$, sample:

     $$rk^w_{b,c} \leftarrow \mathsf{ReKeyGen}(pp, i, sk^*, pk_{u,b}, pk_{v,c}, pk_{w,g(b,c)}),$$

     where sk^* is any one of the two secret-keys $sk_{u,b}$ or $sk_{v,c}$; note that at least one of them is always known.
- Output $sk_C =: \Big(rk^w_{b,c} : w \in [l+1, |C|], b \in \{0,1\}, c \in \{0,1\}\Big)$.

[3] While $pk_{|C|,0}$ is obtained in this step, the key $pk_{|C|,1} = pk_{\text{out}}$, always (and hence never sampled once pk_{out} is fixed).

Observe that this indeed fixes all recode keys as desired, and that the distribution of sk_C is statistically close to the output of KeyGen of $\mathcal{ABE}$ due to the statistical properties of recoding simulation and key indistinguishability.

CHALLENGE. When $\mathcal{A}$ sends (m_0, m_1), the simulator forwards them to the outside challenger, and recieves $(\psi_1, \ldots, \psi_l, \tau_b)$ where $\psi_i = \mathsf{Encode}(pk_i, s) : i \in [l]$ and $\tau_b = \mathsf{Encrypt}(pp, m_b; s)$ for a random bit b. The simulator forwards this response to $\mathcal{A}$.
PHASE 2. $\mathcal{B}$ answers the queries of $\mathcal{A}$ as in phase 1.

GUESS. $\mathcal{A}$ outputs a guess bit b'. The simulator also outputs b' and halts.

By construction $(\psi_1, \ldots, \psi_l, \tau_b)$ is a correctly distributed $\mathcal{ABE}$ encryption of m_b. Therefore $\mathcal{B}$ wins the IND-EDC game if $\mathcal{A}$ wins the selective security game. □

5 Correlation-relaxed TOR from multilinear maps

In this section we provide an instantiation of our CR-TOR scheme. For convenience we first describe our construction using idealized multilinear maps under the d-MDDH assumption (see Section 2). We will then describe how to adapt this construction to the noisy multilinear maps of GGH in appendix B.

5.1 Overview

At a high level our construction works as follows. Let $\vec{\mathbb{G}} = (\mathbb{G}_1, \ldots, \mathbb{G}_d)$ be a tuple of groups equipped with a multilinear map e (Section 2). Let $h_1, \ldots, h_d$ be random elements in $\mathbb{G}_1$, which will be public parameters. A public key at level $i < d$ is formed by powering h_i to a random exponent $z \xleftarrow{\$} \mathbb{Z}_q$. The corresponding public key/secret key pair is (h_i^z, z). The unique public key at level $i = d$ is simply h_d and we take the corresponding secret key[4] to be $z = 1$. Let $y_1 = h_1$ and define recursively $y_{i+1} = e(y_i, h_{i+1})$ for $i < d$. Note that y_i is an element in $\mathbb{G}_i$ for all $i \geq 1$.

Encoding and Recoding. We take $\mathcal{S} = \mathbb{Z}_q$ to be the set of tags. Let $pk = h_i^z$ be a level i public key. Then the set of encodings of a tag s under pk is simply the singleton set $\Psi_{pk,s} = \{y_i^{zs}\}$. Generating a recode key for a pair of public keys $(h_i^{z_0}, h_i^{z_1})$ to a target public key $h_{i+1}^{z_{\text{tgt}}}$ consists of constructing a pair of elements (ρ_0, ρ_1) such that $\rho_0^{z_0} \cdot \rho_1^{z_1} = h_{i+1}^{z_{\text{tgt}}}$. Given encodings $\psi_0 = y_i^{z_0 s}$, $\psi_1 = y_i^{z_1 s}$, one recodes by computing $e(\psi_0, \rho_0) \cdot e(\psi_1, \rho_1) = \psi_{\text{tgt}}$; this calculation is detailed below.

The encoding produced under the output public key is indistinguishable from random if d-MDDH assumption holds. Therefore, we can use it encrypt/blind a message. These are the core ideas, the full scheme follows.

5.2 Construction

- $\mathsf{Params}(1^\lambda, d)$: Output a description of a tuple of groups $\vec{\mathbb{G}} = (\mathbb{G}_1, \ldots, \mathbb{G}_d)$ together with a multilinear map $e(\mathbb{G}_i, \mathbb{G}_1) \to \mathbb{G}_{i+1}$ for $i < d$. Each group has prime order q. Let $g = g_1$ be a canonical generator of $\mathbb{G}_1$. Choose $h_1, \ldots, h_d \xleftarrow{\$} \mathbb{G}_1$. Let $y_1 = h_1$ and define $y_{i+1} = e(y_i, h_{i+1})$ for $i < d$.

[4] Recall from the definition of correlation relaxed TOR that the secret key at level $i = d$ plays no role in the actual computation.

- $\mathsf{Keygen}(pp, i)$: If $i < d$ choose $z \xleftarrow{\$} \mathbb{Z}_q$, let $pk = h_i^z$ and let $sk = z$. If $i = d$, let $pk = h_d$ and let $sk = 1$. Output the pair (pk, sk).
- $\mathsf{Encode}(pk, s)$: Let $pk = h_1^z$ be a level one public key. Compute $\psi = (h_1^z)^s = h_1^{zs}$.
- $\mathsf{ReKeyGen}(pp, i, sk_0, pk_0, pk_1, pk_{\text{tgt}})$: Let $pk_0 = h_i^{z_0}, pk_1 = h_i^{z_1}, sk_0 = z_0$. Compute $rk = (\rho_0, \rho_1)$ as follows:
  1. Choose $r_1 \xleftarrow{\$} \mathbb{Z}_q$ and let $\rho_1 = h_i^{r_1}$.
  2. Compute $\rho_0 = (pk_{\text{tgt}}/(pk_1)^{r_1})^{z_0^{-1}}$.

  Note that the above samples (ρ_0, ρ_1) according to the relation $\rho_0^{z_0} \cdot \rho_1^{z_1} = h_{i+1}^{z_{\text{tgt}}}$, but does so knowing only secret key z_0.
- $\mathsf{Recode}(rk_{0,1}^{\text{tgt}}, \psi_0, \psi_1) = e(\psi_0, \rho_0) \cdot e(\psi_1, \rho_1) = e(y_i^{z_0 s}, \rho_0) \cdot e(y_i^{z_1 s}, \rho_1)$
$= e(y_i^s, \rho_0^{z_0}) \cdot e(y_i^s, \rho_0^{z_1}) = e(y_i^s, \rho_0^{z_0} \cdot \rho_1^{z_1})$
$= e(y_i^s, pk_{\text{tgt}}) = e(y_i^s, h_{i+1}^{z_{\text{tgt}}})$
$= e(y_i, h_{i+1})^{z_{\text{tgt}} s} = y_{i+1}^{z_{\text{tgt}} s} = \psi_{i+1}^{\text{tgt}}$.
- $\mathsf{SimReKeyGen}(pp, i, pk_0, pk_1)$: Let $pk_0 = h_i^{z_0}, pk_1 = h_i^{z_1}$.
  1. Choose $r_0, r_1 \xleftarrow{\$} \mathbb{Z}_q$, set $\rho_0 = h_i^{r_0}$ and $\rho_1 = h_i^{r_1}$. Output recode key $rk = (\rho_0, \rho_1)$.
  2. Let $pk_{\text{tgt}} = (pk_0)^{r_0} \cdot (pk_1)^{r_1}$. Output pk_{tgt}.
- $\mathsf{Encrypt}(pp, m; s)$: We have $pp = (h_1, \ldots, h_d)$. Output $\tau = m \cdot e(\ldots e(e(h_1, h_2), h_3) \ldots, h_d)^s = m \cdot y_d^s$.
- $\mathsf{Decrypt}(pp, \psi_{\text{out}}, \tau)$: Compute $m = \tau/\psi_{\text{out}}$.

The correctness properties are easy to verify. We now show that other properties hold as well if the d-MDDH assumption holds.

Key indistinguishability. Let $(pk_b, sk_b) \leftarrow \mathsf{Keygen}(pp, i)$ for $b = 0, 1$ and $(pk_{\text{tgt}}, sk_{\text{tgt}}) \leftarrow \mathsf{Keygen}(pp, i+1)$. Let $pk_b = h_i^{z_b}$, $sk_b = z_b$ and $pk_{\text{tgt}} = h_{i+1}^{z_{\text{tgt}}}$. The distributions

$$(\rho_0, \rho_1) : \rho_0 = h_i^{r_0}, \rho_1 = (pk_{\text{tgt}}/(pk_0)^{r_0})^{z_1^{-1}}, r_0 \xleftarrow{\$} \mathbb{Z}_q$$
$$(\rho_0, \rho_1) : \rho_1 = h_i^{r_1}, \rho_0 = (pk_{\text{tgt}}/(pk_1)^{r_1})^{z_0^{-1}}, r_1 \xleftarrow{\$} \mathbb{Z}_q$$

are statistically indstinguishable since both experiments sample uniformly from the set $S_{z_0, z_1, pk_{\text{tgt}}} = \{(\rho_0, \rho_1) : \rho_0^{z_0} \cdot \rho_1^{z_1} = pk_{\text{tgt}}\}$.

Recoding simulation. Let $(pk_b, sk_b) \leftarrow \mathsf{Keygen}(pp, i)$ for $b = 0, 1$. Let $pk_b = h_i^{z_b}, sk_b = z_b$. The distributions:

$$pk_{\text{tgt}}, (\rho_0, \rho_1) : pk_{\text{tgt}} = h_{i+1}^{z_{\text{tgt}}}, \rho_0 = h_i^{r_0}, \rho_1 = (pk_{\text{tgt}}/(pk_0)^{r_0})^{z_1^{-1}}, z_{\text{tgt}}, r_0, r_1 \xleftarrow{\$} \mathbb{Z}_q$$
$$pk_{\text{tgt}}, (\rho_0, \rho_1) : pk_{\text{tgt}} = (pk_0)^{r_0} \cdot (pk_1)^{r_1}, \rho_0 = h_i^{r_0}, \rho_1 = h_i^{r_1}, r_0, r_1 \xleftarrow{\$} \mathbb{Z}_q$$

are statistically indistinguishable since in both experiments pk_{tgt} is uniform over $\mathbb{G}_1$ and (ρ_0, ρ_1) sampled uniformly from the set $S_{z_0, z_1, y_{\text{tgt}}}$ defined above.

Indistinguishability of Encoding Derived Ciphertexts. We prove the following claim.

Claim 1. *The above scheme is IND-EDC if the d-Multilinear Decisional Diffie-Hellman assumption holds.*

Proof. Suppose there exists an IND-EDC adversary $\mathcal{A}$ against the above scheme with advantage ϵ. Then there exists an adversary $\mathcal{B}$ which breaks the d-MDDH problem with the same advantage. $\mathcal{B}$ is passed an instance $(g^s, g^{c_1}, \ldots, g^{c_d}, T)$ and runs as follows:

1. Generates $x_1, \ldots x_l \xleftarrow{\$} \mathbb{Z}_q$. Lets $pp = (g^{c_1}, \ldots, g^{c_d})$. Lets $pk_j = g^{x_j}$ for $j \in [l]$. Lets $\psi_j = (g^s)^{x_j}$ for $j \in [l]$
2. Sends $(pp, pk_1, \ldots, pk_l)$ to $\mathcal{A}$.
3. Receives (m_0, m_1) from $\mathcal{A}$.
4. Chooses $b \xleftarrow{\$} 0, 1$ and sends $(\psi_1, \ldots, \psi_l, \tau_b = m_b \cdot T)$ to $\mathcal{A}$.
5. Receives guess b' from $\mathcal{A}$.
6. Outputs 1 if $b' = b$.

Let E_T be the event that T is a multilinear Diffie-Hellman element, while E_F be the event that T is a random element of $\mathbb{G}_d$. Note that $x_j \xleftarrow{\$} \mathbb{Z}_q$ has the same distribution as $c_1 \cdot z_j : z_j \xleftarrow{\$} \mathbb{Z}_q$, thus pk_j are simulated correctly. If E_T occurs, then τ_b is exactly equivalent to the output of $\mathsf{Encrypt}(pp, m_b; s)$, thus $b' = b$ holds exactly when $\mathcal{A}$ wins the IND-EDC game. But if E_F occurs, then τ_b is statistically independent of b, thus $b' = b$ with probability $1/2$. So $\mathcal{B}$ has advantage $|\Pr[b' = b|E_T] - \Pr[b' = b|E_F]| = 1/2 + \epsilon - 1/2 = \epsilon$. □

Corollary 1. *Assume the existence of multilinear maps and the validity of d-MDDH assumption. Then, there exists a selectively-secure ABE scheme for all polynomial-size circuits of depth at most $d - 1$.*

References

[1] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (h)ibe in the standard model. In *EUROCRYPT*, pages 553–572, 2010.

[2] Shweta Agrawal, Xavier Boyen, Vinod Vaikuntanathan, Panagiotis Voulgaris, and Hoeteck Wee. Functional encryption for threshold functions (or fuzzy ibe) from lattices. In *Public Key Cryptography*, pages 280–297, 2012.

[3] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In *ASIACRYPT*, pages 21–40, 2011.

[4] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *EUROCRYPT*, pages 127–144, 1998.

[5] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.

[6] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *IACR Cryptology ePrint Archive*, 2002:80, 2002.

[7] Xavier Boyen. Attribute-based functional encryption on lattices. In *TCC*, pages 122–142, 2013.

[8] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, pages 523–552, 2010.

[9] Melissa Chase. Multi-authority attribute based encryption. In *TCC*, pages 515–534, 2007.

[10] Melissa Chase and Sherman S. M. Chow. Improving privacy and security in multi-authority attribute-based encryption. In *ACM Conference on Computer and Communications Security*, pages 121–130, 2009.

[11] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *CRYPTO (1)*, pages 476–493, 2013.

[12] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, pages 1–17, 2013.

[13] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In *CRYPTO 2013*, volume 8043, pages 479–499. Springer Berlin Heidelberg, 2013.

[14] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In *CRYPTO*, pages 162–179, 2012.

[15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, STOC '13, pages 545–554, New York, NY, USA, 2013. ACM.

[16] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS '06, pages 89–98, New York, NY, USA, 2006. ACM.

[17] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, pages 146–162, 2008.

[18] Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, pages 62–91, 2010.

[19] Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In *EUROCRYPT*, pages 568–588, 2011.

[20] Allison B. Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO*, pages 180–198, 2012.

[21] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 18–35. Springer Berlin Heidelberg, 2009.

[22] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pages 191–208, 2010.

[23] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.

[24] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *TCC*, pages 419–436, 2009.

[25] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In *ACM Conference on Computer and Communications Security*, pages 463–472, 2010.

[26] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[27] Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *CRYPTO*, pages 619–636, 2009.

[28] Brent Waters. Functional encryption for regular languages. In *CRYPTO*, pages 218–235, 2012.

[29] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.

A Construction of TOR

In this section we construct a TOR recoding [15] which is secure under the matrix d-linear assumption [21]. The construction is described in the ideal multilinear setting. By following the description in the next section, this construction is easily adapted to the setting of noisy multilinear maps of both GGH and CLT.

An important remark is that the matrix d-linear assumption cannot hold in the GGH framework [12]. Nevertheless, it remains plausible in the framework of CLT [11]. Therefore, the resulting instantiation of TOR scheme only makes sense in the CLT framework.

Notation For matrices $M = (m_{ij}) \in \mathbb{Z}_q^{a\times b}, N = (n_{ij}) \in \mathbb{Z}_q^{b\times c}$ define $g^M \otimes N = (\prod_{k=1}^n (g^{m_{ik}})^{n_{kj}})_{ij} = (\prod_{k=1}^n g^{m_{ik}n_{kj}})_{ij} = g^{MN}$ and $M \otimes g^N = (\prod_{k=1}^n (g^{n_{kj}})^{m_{ik}}) = (\prod_{k=1}^n g^{m_{ik}n_{kj}})_{ij} = g^{MN}$.

Assumption 2 (Matrix d-linear assumption [21])**.** For any integers a and b, and for any $d \le i < j \le \min(a,b)$ the ensembles $(g, g^R) : R \xleftarrow{\$} \mathsf{Rk}_i(\mathbb{Z}_q^{a\times b})$ and $(g, g^R) : R \xleftarrow{\$} \mathsf{Rk}_j(\mathbb{Z}_q^{a\times b})$ are computationally indistinguishable.

A.1 Definition of TOR [15]

A TOR scheme over an input space $\mathcal{S} = \{\mathcal{S}_\lambda\}$ consists of six polynomial time algorithms (Params, Keygen, Encode, $\mathsf{ReKeyGen}$, $\mathsf{SimReKeyGen}$, Recode) and a symmetric-key encryption scheme (E, D) with the following properties:

- $\mathsf{Params}(1^\lambda, d_{max})$ is a probabilistic algorithm that takes as input security parameter λ and an upper bound d_{max} on the number of nested recoding operations, outputs public parameters pp.
- $\mathsf{Keygen}(\mathsf{pp})$ is a probabilistic algorithm that outputs a public/secret key pair $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Encode}(\mathsf{pk}, s)$ is a probabilistic algorithm that takes pk and an input $s \in \mathcal{S}$, and outputs an encoding ψ.

- $\mathsf{ReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{sk}_0, \mathsf{sk}_1)$ is a probabilistic algorithm that takes a key pair $(\mathsf{pk}_0, \mathsf{sk}_0)$, another public key pk_1, a target public key pk_{tgt} and outputs a recoding key rk.
- $\mathsf{SimReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1)$ is a probabilistic algorithm that tkaes two public keys $\mathsf{pk}_0, \mathsf{pk}_1$ and outputs a recoding key rk together with a target public key pk_{tgt}.
- $\mathsf{Recode}(\mathsf{rk}, \psi_0, \psi_1)$ is a deterministic algorithm that takes the recoding key rk, two encodings ψ_0 and ψ_1, and outputs an encoding ψ_{tgt}.

Correctness For every pk and $s \in \mathcal{S}$ there exists a family of sets $\Psi_{\mathsf{pk},s,j}, j = 0, 1, \ldots, d_{max}$:

- $\Pr[\mathsf{Encode}(\mathsf{pk}, s) \in \Psi_{\mathsf{pk},s,0}] = 1$, where the probability is taken over the coin tosses of Encode;
- $\Psi_{\mathsf{pk},s,0} \subseteq \Psi_{\mathsf{pk},s,1} \subseteq \ldots \subseteq \Psi_{\mathsf{pk},s,d_{max}}$.
- for all $\psi, \psi' \in \Psi_{\mathsf{pk},s,d_{max}}$ and all $m \in \mathcal{M}$, $\mathsf{D}(\psi', \mathsf{E}(\psi, m)) = m$.

Additionally for any triple of key pairs $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1), (\mathsf{pk}_{tgt}, \mathsf{sk}_{tgt})$ and any encodings $\psi_0 \in \Psi_{\mathsf{pk}_0,s,j_0}$ and $\psi_1 \in \Psi_{\mathsf{pk}_1,s,j_1}$,

$$\mathsf{Recode}(\mathsf{rk}, \psi_0, \psi_1) \in \Psi_{\mathsf{pk}_{tgt},s,\max(j_0,j_1)+1}$$

Key Indistinguishablity Let $(\mathsf{pk}_b, \mathsf{sk}_b) \leftarrow \mathsf{Keygen}(\mathsf{pp})$ for $b = 0, 1$ and $(\mathsf{pk}_{tgt}, \mathsf{sk}_{tgt}) \leftarrow \mathsf{Keygen}(\mathsf{pp})$. The following two ensembles must be statistically indistinguishable:

$$[\mathsf{Aux}, \mathsf{ReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{sk}_0, \mathsf{pk}_{tgt})] \equiv_s$$
$$[\mathsf{Aux}, \mathsf{ReKeyGen}(\mathsf{pk}_1, \mathsf{pk}_0, \mathsf{sk}_1, \mathsf{pk}_{tgt})]$$

where $\mathsf{Aux} = ((\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1), (\mathsf{pk}_{tgt}, \mathsf{sk}_{tgt}))$.

Recoding Simulation Let $(\mathsf{pk}_b, \mathsf{sk}_b) \leftarrow \mathsf{Keygen}(\mathsf{pp})$ for $b = 0, 1$. Then the following two ways of sampling the tuple $[(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1), \mathsf{pk}_{tgt}, \mathsf{rk}]$ must be statistically indistinguishable:

$$[(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1), \mathsf{pk}_{tgt}, \mathsf{rk}_{tgt} : (\mathsf{pk}_{tgt}, \mathsf{sk}_{tgt}) \leftarrow \mathsf{Keygen}(\mathsf{pp}); \mathsf{rk} \leftarrow \mathsf{ReKeyGen}(\mathsf{pk}_0, \mathsf{pk}_1, \mathsf{sk}_0, \mathsf{pk}_{tgt})] \equiv_s$$
$$[(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1), \mathsf{pk}_{tgt}, \mathsf{rk}_{tgt} : (\mathsf{pk}_{tgt}, \mathsf{rk}) \leftarrow \mathsf{SimReKeyGen}(\mathsf{pp});]$$

One-time Semantic Security For all $m_0, m_1 \in \mathcal{M}$, the following two distributions must be statistically indistinguishable:

$$[\mathsf{E}(\psi, m_0) : \psi \xleftarrow{\$} \mathcal{K}] \equiv_s [\mathsf{E}(\psi, m_1) : \psi \xleftarrow{\$} \mathcal{K}]$$

Correlated Pseudorandomness For every polynomial $l = l(\lambda)$, let $(pk_i, sk_i) \leftarrow \mathsf{Keygen}(pp)$ for $i \in [l+1]$. Let $s \xleftarrow{\$} \mathcal{S}$, and let $\psi_i \leftarrow \mathsf{Encode}(pk_i, s)$ for $i \in [l+1]$. Then the following two ensembles must be computationally indistinguishable:

$$\left[(pk_i, \psi_i)_{i\in[l]}, pk_{l+1}, \psi_{l+1}\right] \equiv_c$$
$$\left[(pk_i, \psi_i)_{i\in[l]}, pk_{l+1}, \psi : \psi \xleftarrow{\$} \mathcal{K}\right].$$

A.2 TOR from matrix decision linear assumption.

We now describe our construction. Recall that in the (original) TOR scheme, instead of IND-EDC game, one requires correlated pseudorandomness property. All other properties and algorithms remain the same as in the definition of CR-TOR (see section 3). The algorithms of the TOR scheme are as follows.

- $\mathsf{Params}(1^\lambda, d)$: Output a description of a tuple of groups $\vec{\mathbb{G}} = (\mathbb{G}_1, \ldots, \mathbb{G}_d)$ together with a multilinear map $e(\mathbb{G}_i, \mathbb{G}_1) \to \mathbb{G}_{i+1}$ for $i < d$. Each group has prime order q. Let g_i be a canonical generator of $\mathbb{G}_i$ and let $g = g_1$.
- $\mathsf{Keygen}(pp)$: Sample $A \xleftarrow{\$} \mathbb{Z}_q^{d\times d}$. Set $pk = g^A$.
- $\mathsf{Encode}(pk, \vec{s}) = pk \otimes \vec{s} = (g^A)^{\vec{s}} = g^{A\vec{s}}$.
- $\mathsf{ReKeyGen}(pk_0, pk_1, sk_b, pk_{tgt})$: Let $pk_0 = g^{A_0}, pk_1 = g^{A_1}, pk_{tgt} = g^{A_{tgt}}$, so $sk_b = A_b$. Compute $rk = (\rho_0, \rho_1) = (g^{R_0}, g^{R_1})$ as follows:
  1. Sample $R_{1-b} \xleftarrow{\$} \mathbb{Z}_q^{d\times d}$ and let $\rho_{1-b} = g^{R_{1-b}}$.
  2. Compute $\rho_b = (g^{A_{tgt}}/(\rho_{1-b} \otimes A_{1-b})) \otimes A_b^{-1}$.
- $\mathsf{Recode}(rk_{0,1}^{tgt}, \psi_0, \psi_1) = e(\rho_0, g_i^{A_0\vec{s}}) \times e(\rho_1, g_i^{A_1\vec{s}}) = g_{i+1}^{R_0A_0\vec{s}} \times g_{i+1}^{R_1A_1\vec{s}} = g_{i+1}^{(R_0A_0+R_1A_1)\vec{s}} = g_{i+1}^{R_t\vec{s}} = \psi_{tgt}$.
- $\mathsf{SimReKeyGen}(pk_0, pk_1)$: Let $pk_0 = g^{A_0}$ and $pk_1 = g^{A_1}$
  1. Sample $R_0, R_1 \xleftarrow{\$} \mathbb{Z}_q^{d\times d}$, set $\rho_0 = g^{R_0}, \rho_1 = g^{R_1}$ output $rk = (\rho_0, \rho_1)$.
  2. Let $pk = R_0 \otimes g^{A_0} \times R_1 \otimes g^{A_1} = g^{(R_0A_0+R_1A_1)}$.

The correctness properties of our scheme are easy to verify. We show that it satisfies all other properties of TOR as well.

Key indistinguishability. We require that for all $(pk_0 = g^{A_0}, pk_1 = g^{A_1}, pk_{tgt} = g^{A_t})$ the following distributions are indistinguishable:

- Choose $R_0 \xleftarrow{\$} \mathbb{G}_1^{d\times d}$, compute $\rho_0 = g^{R_0}$ and compute $\rho_1 = (g^{A_t}/(R_0 \otimes g^{A_0})) \otimes A_1^{-1}$. Output (ρ_0, ρ_1).
- Choose $R_1 \xleftarrow{\$} \mathbb{G}_1^{d\times d}$, compute $\rho_1 = g^{R_1}$ and compute $\rho_0 = (g^{A_t}/(R_1 \otimes g^{A_1})) \otimes A_0^{-1}$. Output (ρ_0, ρ_1).

However this follows from the fact that $f : \mathbb{Z}_q^{d\times d} \to \mathbb{Z}_q^{d\times d}$ satisfying $f(X) = (g^{A_t}/(X \otimes g^{A_0})) \otimes A_1^{-1}$ is injective.

Recoding simulation. This follows from the fact that $(pk_{tgt} = R_0 \otimes g^{A_0} \times R_1 \otimes g^{A_1}, g^{R_0}, g^{R_1}) : A_0 \xleftarrow{\$} \mathbb{Z}_q^{d\times d}, A_1 \xleftarrow{\$} \mathbb{Z}_q^{d\times d}, R_0 \xleftarrow{\$} \mathbb{Z}_q^{d\times d}, R_1 \xleftarrow{\$} \mathbb{Z}_q^{d\times d}$ is statistically close to $(g^{A_{tgt}}, g^{R_0}, g^{R_1}) : A_{tgt} \xleftarrow{\$} \mathbb{Z}_q^{d\times d}, R_0 \xleftarrow{\$} \mathbb{Z}_q^{d\times d}, R_1 \xleftarrow{\$} \mathbb{Z}_q^{d\times d}$.

Correlated pseudorandomness. We prove the following lemma.

Lemma 1. *The TOR construction achieves correlated pseudorandomness (Section 4.1 [15]) if the matrix d-linear assumption [21] holds.*

Proof. It suffices to prove that the following distributions are indistinguishable:

$$\left[(g, g^{A_1}, \ldots, g^{A_{l+1}}, g^{A_1\vec{s}}, \ldots, g^{A_{l+1}\vec{s}}) : A_i \xleftarrow{\$} \mathbb{Z}_q^{d\times d}, \vec{s} \xleftarrow{\$} \mathbb{Z}_q^d\right] \equiv_c$$
$$\left[(g, g^{A_1}, \ldots, g^{A_{l+1}}, g^{A_1\vec{s}}, \ldots, g^{A_l\vec{s}}, g^{\vec{u}}) : A_i \xleftarrow{\$} \mathbb{Z}_q^{d\times d}, \vec{s} \xleftarrow{\$} \mathbb{Z}_q^d, \vec{u} \xleftarrow{\$} \mathbb{Z}_q^d\right].$$

Suppose the simulator is given a tuple $(g, W = g^A, x = g^{A\vec{s}}, Y = g^B, z = g^{\vec{u}})$ where $A, B \xleftarrow{\$} {\mathbb{Z}_q}^{d\times d}$ and $\vec{u}$ is either equal to $B\vec{s}$ or is random in $\mathbb{Z}_q^d$. Observe that with overwhelming probability, the matrix $[A\ B]$ has rank d in which case the matrix $[[W\ x]; [Y\ z]]$ defines a random $2d \times (d+1)$ matrix of rank either d or $d+1$. Thus the tuple is an instance of the matrix d-Linear assumption. Then the simulator chooses $C_1, \ldots, C_l \xleftarrow{\$} \mathbb{Z}_q^{d\times d}$ and outputs $C_i \otimes W = g^{C_iA}$ and $C_i \otimes x = g^{C_iA\vec{s}}$ for $i = 1, \ldots, l$. Taking $A_i = C_iA$ and $A_{l+1} = B$, one obtains an instance of $(g, g^{A_1}, \ldots, g^{A_{l+1}}, g^{A_1\vec{s}}, \ldots, g^{A_l\vec{s}}, g^{\vec{u}})$ where $\vec{u}$ is either equal to $A_{l+1}\vec{s}$ or is random in $\mathbb{Z}_q^d$. The claim follows. □

One-time semantic security. Let $\mathcal{M} = \mathbb{G}_d^d$, define

$$\mathsf{E}(\psi, \vec{\mu}) = \psi \odot \vec{\mu}$$

where $\odot$ denotes the component-wise product. One-time semantic security follows from the fact that ψ is computationally indistinguishable from a vector of random group elements.

B Mapping our constructions to graded encoding systems

In this section we describe how to translate our constructions using multilinear maps to the graded encoding system of Garg et al. [12]. For simplicity we focus on mapping our construction of CR-TOR from generic multilinear maps in Section 5.

B.1 Graded encoding systems

In the framework of Garg et al. [12] an element g_i^α in a mutlinear group family is an encoding of α at level i. The encoding permits the following operations: equality testing, addition and a bounded number of multiplications. At a high level a d-graded encoding system is a ring R and system of sets $\mathcal{S} = S_i^{(\alpha)} \subset \{0,1\}^* : \alpha \in R, 0 \le i \le d$ such that for every i, the sets $\{S_i^{(\alpha)} : \alpha \in R\}$ are disjoint and form a partition of $S_i = \cup_\alpha S_i^{(\alpha)}$. The GGH system is equipped with the following additional procedures for manipulating encodings:

- **Instance Generation:** $\mathsf{InstGen}(1^\lambda, 1^d)$ takes as inputs the parameters λ, d and outputs $(\mathsf{params}, \mathbf{p}_{zt})$ where params is a description of a d-Graded Encoding System and $\mathbf{p}_{zt}$ is a zero-test parameter for level d.
- **Ring Sampler:** $\mathsf{samp}(\mathsf{params})$ is a randomized algorithm which outputs a level zero encoding $a \in S_0^{(\alpha)}$ for nearly uniform element $\alpha \xleftarrow{\$} R$. The encoding a is not necessarily uniform in $S_0^{(\alpha)}$.

- **Encoding:** $\mathsf{enc}(\mathsf{params}, i, a)$ takes a level zero encoding $a \in S_0^{(\alpha)}$ for some $\alpha \in R$ and index $i \leq d$ and outputs level-i encoding $u \in S_i^{(\alpha)}$ for the same α.
- **Addition and subtraction:** For $u_1 \in S_i^{(\alpha_1)}$, $u_2 \in S_i^{(\alpha_2)}$, we have $\mathsf{add}(\mathsf{params}, i, u_1, u_2) = u_1 + u_2 \in S_i^{(\alpha_1+\alpha_2)}$ and $\mathsf{neg}(\mathsf{params}, i, u_1) = -u_1 \in S_i^{(-\alpha_1)}$.
- **Multiplication:** For $u_1 \in S_{i_1}^{(\alpha_1}$, $u_2 \in S_{i_2}^{(\alpha_2)}$ such that $i_1 + i_2 \leq d$, we have $\mathsf{mul}(\mathsf{params}, i_1, u_1, i_2, u_2) = u_1 \times u_2 \in S_{i_1+i_2}^{(\alpha_1 \cdot \alpha_2)}$.
- **Re-randomization:** For $u \in S_i^{(\alpha)}$, algorithm $\mathsf{reRand}(\mathsf{params}, i, u)$ outputs another encoding $u' \in S_i^{(\alpha)}$. For any two $u_1, u_2 \in S_i^{(\alpha)}$, the distributions of $\mathsf{reRand}(\mathsf{params}, i, u_1)$ and $\mathsf{reRand}(\mathsf{params}, i, u_2)$ are nearly the same.
- **Zero-test:** $\mathsf{isZero}(\mathsf{params}, u)$ outputs 1 if $u \in S_d^{(0)}$ and 0 otherwise. In conjuction with neg, one can test if $u_1, u_2 \in S_d$ encoding the same element $\alpha \in R$.
- **Extraction:** $\mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, u)$ outputs $s \in \{0,1\}^\lambda$ such that
  1. For any $\alpha \in R$ and $u_1, u_2 \in S_d^{(\alpha)}$, $\mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, u_1) = \mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, u_2)$
  2. The distribution : $\{\mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, u_1) : \alpha \xleftarrow{\$} R, u \in S_d^{(\alpha)}\}$ is nearly uniform over $\{0,1\}^\lambda$.

In practice with some negligible probability the zero-test may produce false positives or the extraction may produce differing outputs for encodings of the same element.

B.1.1 Graded Multilinear Decisonal Diffie Hellman assumption

We will require the following analogue of the d-Multilinear Decision Diffie Hellman assumption for d-graded encoding systems.

Assumption 3. (d-Graded Multilinear Decisional Diffie-Hellman (d-GMDDH) assumption) Suppose that a challenger runs $\mathsf{InstGen}(1^\lambda, 1^d)$ generating $(\mathsf{params}, \mathbf{p}_{zt})$. Let $s, c_1, \ldots c_d \leftarrow \mathsf{samp}(\mathsf{params})$. Define $\tilde{s} = \mathsf{cenc}_1(\mathsf{params}, 1, s), \tilde{c}_1 = \mathsf{cenc}_1(\mathsf{params}, 1, c_1), \ldots, \tilde{c}_d = \mathsf{cenc}_1(\mathsf{params}, 1, c_d)$. Then, the d-GMDDH assumption states that the advantage $Adv_{\mathcal{A}}(\lambda)$ of every polynomial time adversary $\mathcal{A}$, defined below, is at most negligible in λ:

$$|\Pr[\mathcal{A}(\tilde{s}, \tilde{c}_1, \ldots, \tilde{c}_d, v) = 1] - \Pr[\mathcal{A}(\tilde{s}, \tilde{c}_1, \ldots, \tilde{c}_d, w) = 1]|$$

where $v = \mathsf{cenc}_1(\mathsf{params}, d, s \cdot c_1 \ldots c_d)$ and $w = \mathsf{cenc}_1(\mathsf{params}, d, \mathsf{samp}(\mathsf{params}))$.

B.2 Our correlation-relaxed TOR using graded encodings

The canonicalizing algorithm $\mathsf{cenc}_l(\mathsf{params}, i, \mathbf{u})$ defined in Remark 2 [12] takes an encoding $\mathbf{u}$ and produces another encoding $\mathbf{u}'$ which is equivalent to l re-randomizations of $\mathbf{u}$. For our purposes l will always be a small constant. For convenience we suppress the params argument when making repeated calls to samp and cenc_l.

- $\mathsf{Params}(1^\lambda, d)$: Run $\mathsf{InstGen}(1^\lambda, 1^d)$ to generate $(\mathsf{params}, \mathbf{p}_{zt})$ where params is a description of a d-Graded Encoding System $\mathcal{S} = (S_1, \ldots, S_d)$. Let $c_1, \ldots, c_d \leftarrow \mathsf{samp}()$. Let $h_1 = \mathsf{cenc}_1(1, c_1), \ldots, h_d = \mathsf{cenc}_1(1, c_d)$. Define $y_i = \prod_{i=1}^{i} h_i \in S_i$ for $i = 1 \ldots d$.
- $\mathsf{Keygen}(pp, i)$: If $i < d$ sample $z \leftarrow \mathsf{samp}()$, let $pk = \mathsf{cenc}_2(1, h_i \cdot z)$ and let $sk = z$. If $i = d$, let $pk = h_d$ and $sk = 1$. Output the pair (pk, sk).

- $\mathsf{Encode}(pk, s)$: Let $pk = \mathsf{cenc}_2(1, h_1 \cdot z)$ be a level one public key. Compute $\psi = \mathsf{cenc}_3(1, pk \cdot s)$.
- $\mathsf{ReKeyGen}(pp, i, sk_0, pk_0, pk_1, pk_{tgt})$: Let $pk_0 = \mathsf{cenc}_2(1, h_i \cdot z_0), pk_1 = \mathsf{cenc}_2(1, h_i \cdot z_1), sk_0 = z_0$. Compute $rk = (\rho_0, \rho_1)$ as follows:
  1. Sample $r_1 \leftarrow \mathsf{samp}()$ and let $\rho_1 = \mathsf{cenc}_3(1, h_i \cdot r_1)$.
  2. Compute $\rho_0 = \mathsf{cenc}_3(1, (pk_{tgt} - pk_1 \cdot r_1) \cdot z_0^{-1})$ where z_0^{-1} is computed over R_q.

  Note that the above samples (ρ_0, ρ_1) according to the relation $\rho_0 \cdot z_0 + \rho_1 \cdot z_1 = h_{i+1} \cdot z_{tgt}$, but does so knowing only secret key z_0.
- $\mathsf{Recode}(rk_{0,1}^{tgt}, \psi_0, \psi_1) = \psi_0 \cdot \rho_0 + \psi_1 \cdot \rho_1 = (y_i \cdot (z_0 \cdot s)) \cdot \rho_0 + (y_i \cdot (z_1 \cdot s)) \cdot \rho_1$
$= (y_i \cdot s) \cdot \rho_0 \cdot z_0 + (y_i \cdot s) \cdot \rho_0 \cdot z_1 = (y_i \cdot s) \cdot (\rho_0 \cdot z_0 + \rho_1 \cdot z_1)$
$= (y_i \cdot s) \cdot pk_{tgt} = (y_i \cdot s) \cdot (h_{i+1} \cdot z_{tgt})$
$= (y_i \cdot h_{i+1}) \cdot (z_{tgt} \cdot s) = y_{i+1} \cdot (z_{tgt} \cdot s) = \psi_{i+1}^{tgt}$.
- $\mathsf{SimReKeyGen}(pp, i, pk_0, pk_1)$: Let $pk_0 = \mathsf{cenc}_2(1, h_i \cdot z_0), pk_1 = \mathsf{cenc}_2(1, h_i \cdot z_1)$.
  1. Sample $r_0, r_1 \leftarrow \mathsf{samp}()$, set $\rho_0 = \mathsf{cenc}_3(1, h_i \cdot r_0)$ and $\rho_1 = \mathsf{cenc}_3(1, h_i \cdot r_1)$. Output recode key $rk = (\rho_0, \rho_1)$.
  2. Let $pk_{tgt} = \mathsf{cenc}_3(1, pk_0 \cdot r_0 + pk_1 \cdot r_1)$. Output pk_{tgt}.
- $\mathsf{Encrypt}(pp, m; s)$: We have $pp = (h_1, \ldots, h_d)$. Let $P = \mathsf{ext}(\mathbf{p}_{zt}, s \cdot \prod_{i=1}^{d} h_i)$. Output $\tau = m \oplus P$.
- $\mathsf{Decrypt}(pp, \psi_{out}, \tau)$: Compute $m = \tau \oplus \mathsf{ext}(\mathbf{p}_{zt}, \psi_{out})$.

Once again, correctness follows easily. We prove the other properties now.

Key indistinguishability. Let $(pk_b, sk_b) \leftarrow \mathsf{Keygen}(pp, i)$ for $b = 0, 1$ and $(pk_{tgt}, sk_{tgt}) \leftarrow \mathsf{Keygen}(pp, i+1)$. Let $pk_b = \mathsf{cenc}_2(1, h_i \cdot z_b), sk_b = z_b$ and $pk_{tgt} = \mathsf{cenc}_2(1, h_{i+1} \cdot z_{tgt})$. The distributions

$$(\rho_0, \rho_1) : \rho_0 = \mathsf{cenc}_3(1, h_i \cdot r_0), \rho_1 = \mathsf{cenc}_3(1, (pk_{tgt} - pk_0 \cdot r_0) \cdot z_1^{-1}), r_0 \leftarrow \mathsf{samp}()$$
$$(\rho_0, \rho_1) : \rho_1 = \mathsf{cenc}_3(1, h_i \cdot r_1), \rho_0 = \mathsf{cenc}_3(1, (pk_{tgt} - pk_1 \cdot r_1) \cdot z_0^{-1}), r_1 \leftarrow \mathsf{samp}()$$

are statistically indstinguishable since both experiments sample uniformly from the set $S_{z_0,z_1,pk_{tgt}} = \{(\rho_0, \rho_1) : \rho_0 \cdot z_0 + \rho_1 \cdot z_1 = pk_{tgt}\}$.

Recoding simulation. Let $(pk_b, sk_b) \leftarrow \mathsf{Keygen}(pp, i)$ for $b = 0, 1$. Let $pk_b = \mathsf{cenc}_2(1, h_i \cdot z_b), sk_b = z_b$. The distributions:

$$\begin{aligned} pk_{tgt}, (\rho_0, \rho_1) :& pk_{tgt} = \mathsf{cenc}_3(1, h_{i+1} \cdot z_{tgt})), \rho_0 = \mathsf{cenc}_3(1, h_i \cdot r_0), \\ & \rho_1 = \mathsf{cenc}_3(1, (pk_{tgt} - (pk_0) \cdot r_0) \cdot z_1^{-1}), z_{tgt}, r_0, r_1 \leftarrow \mathsf{samp}() \\ pk_{tgt}, (\rho_0, \rho_1) :& pk_{tgt} = \mathsf{cenc}_3(1, (pk_0) \cdot r_0 + (pk_1) \cdot r_1), \rho_0 = \mathsf{cenc}_3(1, h_i \cdot r_0), \\ & \rho_1 = \mathsf{cenc}_3(1, h_i \cdot r_1), r_0, r_1 \leftarrow \mathsf{samp}() \end{aligned}$$

are statistically indistinguishable since in both experiments pk_{tgt} is nearly uniform over S_1 and (ρ_0, ρ_1) sampled uniformly from the set $S_{z_0,z_1,pk_{tgt}}$ defined above.

Indistinguishability of Encoding Derived Ciphertexts. We prove the following claim.

Claim 2. *The above scheme is IND-EDC if the d-GMDDH assumption holds.*

Proof. Suppose there exists an IND-EDC adversary $\mathcal{A}$ against the above scheme with advantage ϵ. Then there exists an adversary $\mathcal{B}$ which breaks the d-GMDDH problem with the same advantage. $\mathcal{B}$ is passed an instance $(\tilde{s} = \mathsf{cenc}_1(1, s), \tilde{c}_1 = \mathsf{cenc}_1(1, c_1), \ldots, \tilde{c}_d = \mathsf{cenc}_1(1, c_d), T)$ and runs as follows:

1. Samples $x_1, \ldots x_l \leftarrow \mathsf{samp}()$. Lets $pp = \tilde{s}, \tilde{c_1}, \ldots, \tilde{c}_d$. Lets $pk_j = \mathsf{cenc}_2(1, x_j)$ for $j \in [l]$. Lets $\psi_j = \mathsf{cenc}_3(1, \tilde{s} \cdot x_j)$ for $j \in [l]$.
2. Sends $(pp, pk_1, \ldots, pk_l)$ to $\mathcal{A}$.
3. Receives (m_0, m_1) from $\mathcal{A}$.
4. Chooses $b \xleftarrow{\$} 0, 1$ and sends $(\psi_1, \ldots, \psi_l, \tau_b = M_b \cdot T)$ to $\mathcal{A}$.
5. Receives guess b' from $\mathcal{A}$.
6. Outputs 1 if $b' = b$.

Let E_T be the event that T is a multilinear Graded Diffie-Hellman element, while E_F be the event that T is a random element of S_d. Note that $\mathsf{cenc}_2(1, x_j) : x_j \leftarrow \mathsf{samp}()$ has the same distribution as $\mathsf{cenc}_2(1, \tilde{c}_1 \cdot z_j) : z_j \leftarrow \mathsf{samp}()$, thus pk_j are simulated correctly. If E_T occurs, then τ_b is exactly equivalent to the output of $\mathsf{Encrypt}(pp, m_b; s)$, thus $b' = b$ holds exactly when $\mathcal{A}$ wins the IND-EDC game. But if E_F occurs, then τ_b is statistically independent of b, thus $b' = b$ with probability $1/2$. So $\mathcal{B}$ has advantage $|\Pr[b' = b|E_T] - \Pr[b' = b|E_F]| = 1/2 + \epsilon - 1/2 = \epsilon$. □