Tightly Secure Hierarchical Identity-Based Encryption

Roman Langrehr  Jiaxin Pan*

July 3, 2019

Karlsruhe Institute of Technology, Karlsruhe, Germany
roman.langrehr@student.kit.edu, jiaxin.pan@kit.edu

Abstract

We construct the first tightly secure hierarchical identity-based encryption (HIBE) scheme based on standard assumptions, which solves an open problem from Blazy, Kiltz, and Pan (CRYPTO 2014). At the core of our constructions is a novel randomization technique that enables us to randomize user secret keys for identities with flexible length.

The security reductions of previous HIBEs lose at least a factor of \( Q \), which is the number of user secret key queries. Different to that, the security loss of our schemes is only dependent on the security parameter. Our schemes are adaptively secure based on the Matrix Diffie-Hellman assumption, which is a generalization of standard Diffie-Hellman assumptions such as \( k \)-Linear. We have two tightly secure constructions, one with constant ciphertext size, and the other with tighter security at the cost of linear ciphertext size. Among other things, our schemes imply the first tightly secure identity-based signature scheme by a variant of the Naor transformation.

Keywords: Hierarchical identity-based encryption, tight security, affine message authentication codes.

1 Introduction

1.1 Motivation

Tight security. Reductions are useful tools for proving the security of public-key cryptographic schemes. Asymptotically, a reduction shows that if there is an efficient adversary \( \mathcal{A} \) that breaks the security of a scheme then we can have another adversary \( \mathcal{R} \) that solves the underlying computationally hard problem. Concretely, a reduction provides a security bound for the scheme, \( \varepsilon_{\mathcal{A}} \leq \ell \cdot \varepsilon_{\mathcal{R}}, \)

1 where \( \varepsilon_{\mathcal{A}} \) is the success probability of \( \mathcal{A} \) and \( \varepsilon_{\mathcal{R}} \) is that of \( \mathcal{R} \). Ideally, it is more desirable to have \( \ell \) as small as a constant. We say a reduction is tight if \( \ell \) is a small constant and the running time of \( \mathcal{A} \) is approximately the same as that of \( \mathcal{R} \). Most of the current works have considered the tightness notion called “almost tight security”, where \( \ell \) may linearly (or, even better, logarithmically) depend on the security parameter, but not on the size of \( \mathcal{A} \).

Recently, tightly secure cryptographic schemes drew a large amount of attention (e.g. [18, 8, 3, 11, 16, 1, 12, 17]), since tightly secure schemes do not need to compensate for any security loss.

(Hierarchical) identity-based encryption. The concept of identity-based encryption (IBE) was proposed by Shamir [32] to simplify the management of public keys and certificates. With an IBE scheme, one can encrypt a message under a recipient’s identity \( \text{id} \) (for instance, email address or ID card number), and this encrypted message can be decrypted with user \( \text{id} \)'s secret key from a trusted authority. The first constructions of IBE were given in 2001 [4, 9, 31] in the random oracle model.

*Supported by DFG grant HO 4534/4-1.
1Here we ignore the additive negligible terms for simplicity.
2In this paper, we do not distinguish almost tight security from tight security, but we will detail the security loss in the security proof and comparison of our schemes.
A hierarchical IBE (HIBE) scheme [22, 14] generalizes the concept of IBE and provides more functionality by forming levels of a hierarchy. In an $L$-level HIBE, a hierarchical identity is a vector of maximal $L$ identities, and a user at level $i$ can delegate a secret key for its descendants at level $i'$ (where $i < i' \leq L$). Moreover, a user at level $i$ is not supposed to decrypt any encryption from a recipient which is not amongst its descendants. HIBE schemes not only are more general than IBE schemes (for instance, an IBE is simply a 1-level HIBE), but also provide numerous applications. Most famous ones are CCA-secureIBEs [5] and identity-based signatures [24] from HIBE. Both implications are tight.

Adaptive security is a widely accepted security notion for (H)IBE schemes, where an adversary is allowed to adaptively choose a challenge identity id" after it sees the (master) public key and $Q$-many user secret keys for adversarial chosen identities. To achieve adaptive security in the standard model, the early IBE constructions require either non-tight reductions to the hardness of the underlying assumptions [34, 7, 28, 23], or $Q$-type, non-static assumptions [13].

In 2013, Chen and Wee constructed the first tightly secure IBE based on static assumptions in the standard model [8]. After that, several works have been done to improve its efficiency and achieve stronger security [3, 21, 16, 19]. However, constructing an $L$-level HIBE for $L > 1$ with a tight (i.e., independent of $Q$) security reduction to a standard assumption remains open.

HIBE MEET TIGHTNESS: DIFFICULTIES AND THE HOPE. Before analyzing the difficulties of achieving tightly secure HIBE, we consider the security loss of the current state-of-the-art HIBE schemes. The $L$-level HIBE from [34] has a relatively large security loss, $Q^2$, which depends on both $Q$ and $L$. Although the security loss of more recent HIBE schemes [33, 28, 8, 3, 15] does not depend on the number of maximal levels $L$, they are still not tight and lose a factor of $Q$.

In general, it is harder to construct HIBE schemes than IBEs, since HIBE allows public delegation of user secret keys, given the corresponding ancestor’s secret key. Hence, given a tightly secure IBE, there is no (tight) black-box transformation to HIBE. The works of Lewko and Waters [29] show the potential difficulty of constructing HIBE with tight reductions. More precisely, [29] proves that it is hard to have an HIBE scheme with security loss less than exponential in $L$, if the HIBE has rerandomizable user secret keys (over all “functional” user secret keys).

The first attempt of constructing tightly secure HIBE is due to Blazy, Kiltz, and Pan (cf. the proceeding version and the first full version of [3]), where they tightly transform algebraic message authentication code (MAC) schemes with affine structures to (H)IBE schemes. As long as the algebraic MAC has tight security, the resulting (H)IBE is tightly secure. The first version of their paper contains a tightly secure delegatable MAC, which results in a tightly secure HIBE. The resulting HIBE has bypassed the impossibility result of [29] and their user secret keys are only rerandomizable over all keys generated by the user secret key generation algorithm, which is only a subspace of all “functional” keys. However, shortly after its publication, a flaw was found in a proof step of the delegatable MAC, and they remove this tightly secure delegatable MAC from their paper. The flaw is basically due to the fact that the BKP randomization technique failed to randomize MAC tags (which is an important part of user secret keys) for hierarchical identities.

The hope of achieving tight security for HIBE schemes lies in developing a novel method that enables randomization of user secret keys for identities with flexible level.

1.2 Our contributions

We answer the aforementioned open question affirmatively with two tightly secure hierarchical identity-based encryption schemes with identity space $ID := \{0, 1\}^\leq L$: One with constant ciphertext size (in terms of the number of group elements) and $O(\alpha L^2)$ security loss, and the other with ciphertext size linear in $L$ but $O(\alpha L)$ security loss. Both schemes are the first tightly secure HIBE schemes. We compare our schemes with the existing HIBE schemes in prime-order pairing groups in Table 1.

Furthermore, via the known tight transformations from [24] and [5], our HIBEs imply the first tightly secure identity-based signature and tightly CCA-secure HIBE almost for free. We note that an $(L + 1)$-level HIBE tightly implies an $L$-level CCA-secure HIBE via the CHK transformation [5] in the single-challenge setting.

Core idea. In a nutshell, the technical novelty of our constructions is a new randomization technique that enables us to randomize user secret keys with flexible identity length. This technique is motivated by the recent tightly CCA-secure public-key encryption of Gay et al. [11].
<table>
<thead>
<tr>
<th>Scheme</th>
<th>[mpk]</th>
<th>[usk]</th>
<th>[C]</th>
<th>Loss</th>
<th>Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wat05 [34]</td>
<td>O(αL)[G₁]</td>
<td>O(αL)[G₂]</td>
<td>(1 + L)[G₁]</td>
<td>O(αL)^L</td>
<td>DBDH</td>
</tr>
<tr>
<td>Wat09 [33]</td>
<td>O(L)[G₁]</td>
<td>O(L)[G₂]</td>
<td>O(L)[G₁] +</td>
<td>O(Q)</td>
<td>2-LIN</td>
</tr>
<tr>
<td>Lew12 [28]</td>
<td>60[G₁ + 2[G₂]</td>
<td>(60 + 10L)[G₁]</td>
<td>10L</td>
<td>O(Q)</td>
<td>2-LIN</td>
</tr>
<tr>
<td>CW13 [8]</td>
<td>O(αL²)[G₁] +</td>
<td>O(αL)[G₂]</td>
<td>(2L + 2)[G₁]</td>
<td>O(Q)</td>
<td>k-LIN</td>
</tr>
<tr>
<td>BKP14 [3]</td>
<td>O(αL²)[G₁] +</td>
<td>O(αL)[G₂]</td>
<td>(2L + 2)[G₁]</td>
<td>O(Q)</td>
<td>k-LIN</td>
</tr>
<tr>
<td>HIBKEM₁ (Fig. 15)</td>
<td>O(αL²)[G₁] +</td>
<td>O(αL²)[G₂]</td>
<td>5[G₁]</td>
<td>O(αL²)</td>
<td>SXDH</td>
</tr>
<tr>
<td>HIBKEM₂¹ (Fig. 7)</td>
<td>O(αL)[G₁] +</td>
<td>O(αL)[G₂]</td>
<td>5[G₁]</td>
<td>O(αL)</td>
<td>SXDH</td>
</tr>
<tr>
<td>HIBKEM₂ (Fig. 16)</td>
<td>O(αL²)[G₁] +</td>
<td>O(αL)[G₂]</td>
<td>(3L + 2)[G₁]</td>
<td>O(αL)</td>
<td>SXDH</td>
</tr>
<tr>
<td>HIBKEM₂² (Fig. 11)</td>
<td>O(αL)[G₁] +</td>
<td>O(αL)[G₂]</td>
<td>(3L + 2)[G₁]</td>
<td>O(αL)</td>
<td>SXDH</td>
</tr>
</tbody>
</table>

Table 1: Comparison of L-level HIBEs with identity-space ID = (0, 1)^L in prime-order pairing groups. Schemes in gray are new ones from this paper. In particular, HIBKEM₁ is an improvement of HIBKEM with collision-resistant hash functions, namely, implementing the generic construction (cf. Fig. 12) with the MAC scheme from Fig. 7. Similarly, HIBKEM₂¹ is an improvement of HIBKEM₂ with the MAC scheme from Fig. 11. "[mpk]", "[usk]" and "[C]" stand for the size of master public key, user secret key and ciphertext. We count the number of group elements in G₁, G₂, and G₇. For a scheme that works in symmetric pairing groups, we write G := G₁ = G₂. Q is the number of user secret key queries by the adversary. γ is the bit length of the range of a collision resistant hash functions.

At the core of our constructions lie two new pseudorandom message authentication code (MAC) schemes for messages with flexible length. Their pseudorandomness can be proven with tight reductions to the Matrix Decisional Diffie-Hellman (MDDH) assumption [10]. The MDDH assumption is a generalization of the known standard Diffie-Hellman assumptions, such as the k-linear (k-LIN) assumption. Our MAC schemes have algebraic structures compatible with the BKP transformation. In the end, together with a variant of the BKP framework [3], we can tightly randomize user secret keys with hierarchical identities and we have tightly secure HIBEs.

A CLOSER LOOK AT THE BKP FRAMEWORK. The BKP framework proposes the notion of affine MACs and transforms it to an (H)IBE scheme with pairings. Their transformation is tightness-preserving. Under the MDDH assumption, if the affine MAC is tightly secure, then the (H)IBE is also tightly secure. It is worth mentioning that the BKP transformation and its variants are widely used in constructing identity-based encryption [19] with multi-challenge CCA security, predicate encryption [35, 6], quasi-adaptive NIZK [26], and structure-preserving signature [25, 12] based on standard, static assumptions.

We recall their tightly secure MAC, MACR, based on the Naor-Reingold pseudorandom function [30], which is implicitly in the Chen-Wee (CW) IBE [8] as well. We observe in this paper that MACR is tightly secure in the semi-adaptive sense, which implies a tightly semi-adaptively secure HIBE. For completeness, we provide a detailed proof for that in Appendix A. However, until now, we do not know any tight security proof for the adaptive security of MACR. In the following, we give more detailed analysis for that.

MACR is defined over an additive prime-order group G₂ := ⟨P₂⟩ and its message space is corresponding to the identity space of the resulting IBE. We use the implicit notation [x]₂ := xP₂ from [10]. MACR chooses B ∈ Z_q^{k+1}×k according to the underlying assumption. For message space M := {0, 1}^α, its secret key is defined as

\[ sk_{MAC} := \left( (x_{i,b})_{1 \leq i \leq \alpha, b = 0,1}, x_0^i \right) \in (\mathbb{Z}_q^{k+2})^\alpha \times \mathbb{Z}_q \]

and its MAC tag contains a message-independent vector [t]₂ and a message-dependent value [u]₂ in the form of

\[ t = Bs \in \mathbb{Z}_q^k \quad \text{for} \quad s \in \mathbb{Z}_q^k \]
\[ u = \sum_i x_{i,m}^T t + x_0 \in \mathbb{Z}_q \],

where B denotes the first k rows of B. The BKP transformation requires the MAC scheme has pseudorandomness against chosen-message attacks (PR-CMA security), which is a decisional variant of the standard existential unforgeability against chosen-message attacks (EUF-CMA security). In order to
provide a simpler and more intuitive discussion, we consider the standard EUF-CMA security of MAC_{NR}, where an adversary $A$ is allowed to see many MAC tags $\tau_m := ([t_m]_2, [u_m]_2)$ on messages $m$ of its choice and tries to forge a fresh and valid forgery $(m', \tau')$ which satisfies Equation (1).

Following the CW argument [8], by a hybrid argument on the bit length of $m$, one can show that the value $[u]_2$ is pseudorandom such that it is hard for an adversary to forge. By embedding the problem challenge in $t$ and $x_{i+1,1-b}$, the CW argument can manage to develop the following random function $RF_{i+1}$ for $(i+1)$-bit messages from a random function $RF_i$ for $i$-bit messages on-the-fly:

$$RF_{i+1}(m_{i+1}) = \begin{cases} RF_i(m_i) & (\text{if } m_{i+1} = b) \\ RF_i(m_i) + RF'_i(m_i) & (\text{if } m_{i+1} = 1 - b) \end{cases}$$

where $b$ is the guess for the $(i+1)$-th bit of $m'$ and $m_i$ is the first $i$ bits of $m$. Such an argument works well if messages have fixed length. For messages $m$ with fixed length, an adversary can see the output of either $RF_i$ (in Hybrid $i$) or $RF_{i+1}$ (in Hybrid $i+1$), but not both. However, that is not the case for messages $m'$ with flexible length.

Concretely, identities for HIBEs are messages with flexible level. If we follow the CW and BKP arguments, we first need to develop a random function at the 2-level based on that at the 1-level. The critical case happens when we switch from Hybrid $\alpha$ (namely, the end of randomization at the 1-level) to Hybrid $\alpha + 1$ (namely, the beginning of randomization at the 2-level). If we define $RF_{\alpha+1}$ (with message space $\{0,1\}^\alpha \cup \{0,1\}^{\alpha+1}$) via Equation (2) based on random functions $RF_\alpha, RF'_\alpha$ (with message space $\{0,1\}^\alpha$), then we have $RF_{\alpha+1}(m) = RF_{\alpha+1}(m\|b)$ for a $m \in \{0,1\}^\alpha$ and that means the resulting $RF_{\alpha+1}$ is not a random function for messages with flexible levels.

### 1.3 Our approach: independent randomization

To circumvent the aforementioned problem, we propose a suitable pseudorandom MAC, which isolates the tag randomization for messages with different levels. Our strategy is to randomize tags for messages with only one level first, and then for those with two levels, and so on. By a novel use of the recent subspace randomization refined from [11], tags for messages with different levels are randomized independently.

**Affine MACs with levels.** We consider a new notion of affine MACs, called *affine MACs with levels*, and we give two constructions of it. This new notion considers messages with flexible levels and enable us to develop independent random functions $RF_\alpha$ for messages with only one level (i.e., in $\{0,1\}^\alpha$), and $RF'_\alpha$ for messages with only two levels (i.e., in $\{0,1\}^{2\alpha}$), and so on. For simplicity, we present an overview of our technique in terms of 2-level HIBEs ($L = 2$), namely, the hierarchical identity space $ID := \{(0,1)^i\}^{\leq 2}$. We denote 1-level messages as $m \in \{0,1\}^\alpha$ and 2-level messages as $m' \in \{0,1\}^{2\alpha}$.

Our first MAC construction $MAC_1$’s secret keys have the form of

$$sk_{MAC_1} := (x_{i,b})_{i,b} \biggl[ (x_{j,b})_{1 \leq j \leq 2\alpha,b} \biggr] x'_{0} \in \left( \mathbb{Z}_q^{k} \right)^{\alpha} \times \left( \mathbb{Z}_q^{k} \right)^{2\alpha} \times \mathbb{Z}_q.$$ 

Value $u$ in the MAC tags for $m \in \{0,1\}^\alpha$ and $m' \in \{0,1\}^{2\alpha}$ has the form of

$$u_m := \sum_{i=1}^{\alpha} x_{i,m}^T t + x'_{0} \in \mathbb{Z}_q$$

$$u_{m'} := \sum_{i=1}^{2\alpha} x_{i,m'}^T t + \sum_{j=1}^{\alpha} x_{j,m'}^T t + x'_{0} \in \mathbb{Z}_q.$$ 

By a similar argument as in the BKP we can randomize all the $u_m$ for 1-level messages $m$ and, after the first level messages randomization, $u_m$ has the form

$$u_m := \sum_{i=1}^{\alpha} x_{i,m}^T t + RF_\alpha (m) ,$$
namely, we replace $x'_m$ with $RF_α(m)$, but this affects the $u_{m'}$ for 2-level messages $m'$ as well. More precisely, $u_{m'}$ carries the random function $RF_α$ and has the form
\[ u_{m'} := \left( \sum_{i=1}^{α} x_{i,m'}^T + \sum_{j=1}^{2α} x_{j,m'}^T \right) t + RF_α(m'_α). \]
If we continue to randomize $u_{m'}$, we will run into the exact same problem as in the CW or BKP randomization.

Motivated by [11], we hide $RF_α$ in some orthogonal space. By switching $t$ into the “right” span, $RF_α$ appears in $u_m$, but gets canceled in $u_{m'}$. Concretely, we choose $B \xleftarrow{\$} Z_q^{3k \times k}$ and $B^⊥ \xleftarrow{\$} Z_q^{3k \times 2k}$ is a kernel matrix of $B$ such that $(B^⊥)^TB = 0$. We replace $t \xleftarrow{\$} Z_q^k$ with larger $t \xleftarrow{\$} Z_q^k$. We embed the random function $RF_α$ into the kernel of $B$ and $u_y (y \in \{m, m'\})$ has the form
\[ u_y := \left( \sim + RF_α(y_α)(B^⊥) \right) t + x_0', \]
where “$\sim$” denotes corresponding summation terms. During the randomization for 1-level messages, if we choose $t \in \text{Span}(B) := \{v \mid \exists s \in Z_q^k : v = Bs\}$ for 2-level messages $m'$, then $RF_α$ will get canceled out; and if we choose $t \notin \text{Span}(B)$ for 1-level messages $m$, then $RF_α$ will appear but $u_y$ gets randomized. After the randomization for 1-level messages, $u_{m'}$ for 2-level messages $m'$ is distributed the same as in Equation (3) so that we can start 2-level randomization from a constant random function $RF_0(\epsilon)$ multiplying with $(B^⊥)^T$, where $\epsilon$ denotes the empty string.

The way of developing $RF_α$ (or $RF_{2,α}$, respectively) from $RF_0$ (or $RF_{0,α}$, respectively) is similar to [11]. Roughly, we choose two random matrices $B_0, B_1 \xleftarrow{\$} Z_q^{3k \times k}$ and decompose $Z_q^{2k}$ into the span of $B, B_0, B_1$. The span of $B^⊥$ is decomposed into that of $B_0^* \in Z_q^{3k \times k}$ and $B_1^* \in Z_q^{3k \times k}$. An overview of the orthogonal relations between all these matrices is given in Figure 1. After the decomposition of linear

\[ \text{basis for } \text{Span}(B^⊥) \]
\[ \text{basis for } Z_q^{3k} \]

\[ B \quad B_0^* \quad B_1^* \]
\[ B_0 \quad B_1 \]

Figure 1: Solid lines mean orthogonal: $B^TB_0^* = B_1^TB_0^* = 0 = B^TB_1^* = B_0^TB_1^* \in Z_q^{k \times k}$.

spaces, $RF_{i}(m_{1i})(B^⊥)^T = RF_{i}^{(0)}(m_{1i})(B_0^*)^T + RF_{i}^{(1)}(m_{1i})(B_1^*)^T$. By using the MDDH assumption, we can switch $[t]_2$ to the right span and develop $RF_{i+1}(m_{1i+1})(B^⊥)^T$ from $RF_{i}(m_{1i})(B^⊥)^T$ in a tight fashion.

In order to have public delegation, the user secret keys at level 1 contain delegation terms $[X_{1b}^T t]_2$. Since our randomization at different levels are isolated, the published terms will not affect our randomization strategy. Details are given in Section 3.1. In the end, our security reduction loses a factor of $O(\alpha L^2)$ due to $L$-many randomization loops and the fact that in each loop a additional factor of $O(\alpha L)$ is required. Applying a variant of the BKP transformation (cf. Section 4), we obtain the first HIBE scheme with tight security.

Achieving tighter security. Our second MAC construction (MAC$_2$ in Section 3.2) parallelizes the above randomization strategy and it has a scheme with security loss $O(\alpha L)$. The cost of doing this is to have different $t_i$ at different level for a message with $L$ levels, which results in an HIBE with $O(L)$-size ciphertext via the BKP transformation.

1.4 More related work and open problems

Bader et al. [2] use some idea from the BKP HIBE to construct digital signature schemes with corruptions, but it does not involve any randomization for messages with flexible length, and thus it does not have the same issue as the BKP.

Very recently, Hofheinz, Jia, and Pan [19] extend the BKP construction with the information-theoretical Cramer-Shoup-like argument of [11] to answer multiple challenge ciphertext queries for IBE. However, we do not think that their technique and the one from [16] can work in a straightforward way here to
construct tightly multi-challenge secure HIBE. To give more details, the main idea in [11, 19] is to have the secret keys as matrices instead of vectors in the BKP construction such that they can create subspaces to answer multiple challenge queries. Since our randomization technique is quite different to BKP, these subspace creating technique does not work here. Essentially, the information-theoretic arguments in Lemmata 3.10, 3.11 and 3.14 (for our first MAC) and Lemmata 3.25 and 3.26 (for our second MAC) will fail in the multi-challenge setting even with larger secret keys as matrices. Thus, we leave achieving tight multi-challenge security for HIBE as an open problem.

1.5 Publication Information

An extended abstract of this work appeared in the proceedings of PKC 2019 [27]. Shortly after the publication, we got invited to submit our full version to the Journal of Cryptology based on the recommendation of the program chairs. This is the full version of it, which includes security proofs of our second MAC scheme (cf. Theorem 3.19) and our generic HIBKEM construction (cf. Theorem 4.1), and a concrete instantiation of our generic construction based on the SDXH assumption. More than 25% of this version is new.

2 Preliminaries

NOTATIONS. We use $x \in \mathcal{S}$ to denote the process of sampling an element $x$ from $\mathcal{S}$ uniformly at random if $\mathcal{S}$ is a set. For positive integers $k > 1$, $\eta \in \mathbb{Z}^+$ and a matrix $A \in \mathbb{Z}_q^{(k+\eta)\times k}$, we denote the upper square matrix of $A$ by $\overline{A} \in \mathbb{Z}_q^{k\times k}$ and the lower $\eta$ rows of $A$ by $\underline{A} \in \mathbb{Z}_q^{\eta\times k}$. Similarly, for a column vector $v \in \mathbb{Z}_q^{k+n}$, we denote the upper $k$ elements by $\overline{v} \in \mathbb{Z}_q^k$ and the lower $\eta$ elements of $v$ by $\underline{v} \in \mathbb{Z}_q^\eta$. For a string $m \in \Sigma^n$, $m_i$ denotes the $i$-th component of $m$ ($1 \leq i \leq n$) and $m_i$ denotes the prefix of length $i$ of $m$.

Furthermore for a $p$-tuple of bit strings $m \in (\{0,1\}^n)^p$, we use $[m]_i$ to denote the string $m_1||\ldots||m_p$. Thus for $1 \leq i \leq np$, $[m]_i$ denotes the $i$-th bit of $m_1||\ldots||m_p$ and $[m]_{i1}$ denotes the $i$-bit-long prefix of $m_1||\ldots||m_p$.

All our algorithms are probabilistic polynomial time unless we stated otherwise. If $\mathcal{A}$ is an algorithm, then we write a $\ell^k$-$\mathcal{A}(b)$ to denote the random variable that outputted by $\mathcal{A}$ on input $b$.

GAMES. Following [3], we use code-based games to define and prove security. A game $G$ contains procedures Init and Finalize, and some additional procedures $P_1, \ldots, P_n$, which are defined in pseudocode. Initially all variables in a game are undefined (denoted by $\bot$), all sets are empty (denoted by $\emptyset$), and all partial maps (denoted by $f: A \rightarrow B$) are totally undefined. An adversary $\mathcal{A}$ is executed in game $G$ (denote by $G^{\mathcal{A}}$) if it first calls Init, obtaining its output. Next, it may make arbitrary queries to $P_1$ (according to their specification), again obtaining their output. Finally, it makes one single call to Finalize$(\cdot)$ and stops. We use $G^{\mathcal{A}} \Rightarrow d$ to denote that $G$ outputs $d$ after interacting with $\mathcal{A}$, and $d$ is the output of Finalize.

$T(\mathcal{A})$ denotes the running time of $\mathcal{A}$.

2.1 Pairing groups and matrix Diffie-Hellman assumptions

Let $G_{\text{Gen}}$ be a probabilistic polynomial time (PPT) algorithm that on input $1^\lambda$ returns a description $G := (G_1, G_2, G_T, q, P_1, P_2, e)$ of asymmetric pairing groups where $G_1, G_2, G_T$ are cyclic groups of order $q$ for a $\lambda$-bit prime $q$, $P_1$ and $P_2$ are generators of $G_1$ and $G_2$, respectively, and $e: G_1 \times G_2$ is an efficient computable (non-degenerated) bilinear map. Define $P_T := e(P_1, P_2)$, which is a generator in $G_T$. In this paper, we only consider Type III pairings, where $G_1 \neq G_2$ and there is no efficient homomorphism between them. All our constructions can be easily instantiated with Type I pairings by setting $G_1 = G_2$ and defining the dimension $k$ to be greater than 1.

We use implicit representation of group elements as in [10]. For $s \in \{1, 2, T\}$ and $a \in \mathbb{Z}_q$ define $[a]_s = aP_s \in G_s$ as the implicit representation of $a$ in $G_s$. Similarly, for a matrix $A = (a_{ij}) \in \mathbb{Z}_q^{n \times m}$ we define $[A]_{s}$ as the implicit representation of $A$ in $G_{s}$. $\text{Span}(A) := \{Ar \mid r \in \mathbb{Z}_q^m\} \subset \mathbb{Z}_q^n$ denotes the linear span of $A$, and similarly $\text{Span}([A]_s) := \{[Ar]_s \mid r \in \mathbb{Z}_q^m\} \subset G_{s}^n$. Note that it is efficient to compute $[AB]_s$. 
given \([A], B\) or \((A, [B])\) with matching dimensions. We define \([A]_1 \circ [B]_2 := e([A]_1, [B]_2) = [AB]_T\), which can be efficiently computed given \([A]_1\) and \([B]_2\).

Next we recall the definition of the matrix Diffie-Hellman (MDDH) and related assumptions [10].

**Definition 2.1 (Matrix distribution).** Let \(k, \ell \in \mathbb{N}\) with \(\ell > k\). We call \(D_{\ell, k}\) a matrix distribution if it outputs matrices in \(\mathbb{Z}_{q}^{\ell \times k}\) of full rank \(k\) in polynomial time.

Without loss of generality, we assume the first \(k\) rows of \(A \in D_{\ell, k}\) form an invertible matrix. The \(D_{\ell, k}\)-matrix Diffie-Hellman problem is to distinguish the two distributions \(([A], [AW])\) and \(([A], [U])\) where \(A \in D_{\ell, k}, w \in \mathbb{Z}_{q}^{k}\) and \(u \in \mathbb{Z}_{q}^{\ell}\).

**Definition 2.2 (\(D_{\ell, k}\)-matrix Diffie-Hellman assumption).** Let \(D_{\ell, k}\) be a matrix distribution and \(s \in \{1, 2, T\}\). We say that the \(D_{\ell, k}\)-matrix Diffie-Hellman \((D_{\ell, k}\text{-MDDH})\) assumption holds relative to \(G_{Gen}\) in group \(G_s\) if for all PPT adversaries \(A\), it holds that

\[
\text{Adv}_{D_{\ell, k}\text{-Gen}, s}^{\text{MDDH}}(A) := |\Pr[A(G, [A]_{s}, [AW]_{s}) = 1] - \Pr[A(G, [A]_{s}, [U]_{s}) = 1]| \approx \frac{1}{q - 1} ,
\]

is negligible where the probability is taken over \(G \in G(1^\lambda), A \in D_{\ell, k}, w \in \mathbb{Z}_{q}^{k}\) and \(u \in \mathbb{Z}_{q}^{\ell}\).

The uniform distribution is a particular matrix distribution that deserves special attention, as an adversary breaking the \(U_{\ell, k}\) assumption can also distinguish between real MDDH tuples and random tuples for all other possible matrix distributions. For uniform distributions, they stated in [11] that \(U_{\ell}\)-MDDH and \(U_{\ell, k}\)-MDDH assumptions are equivalent.

**Definition 2.3 (Uniform distribution).** Let \(k, \ell \in \mathbb{N}\) with \(\ell > k\). We call \(U_{\ell, k}\) a uniform distribution if it outputs uniformly random matrices in \(\mathbb{Z}_{q}^{\ell \times k}\) of rank \(k\) in polynomial time. Let \(U_{\ell} := U_{\ell+1, k}\).

**Lemma 2.4 \((U_{\ell, k}\text{-MDDH} \iff U_{\ell}\text{-MDDH})\).** Let \(\ell, k \in \mathbb{N}_+\) with \(\ell > k\). An \(U_{\ell, k}\)-MDDH instance is as hard as an \(U_{\ell}\)-MDDH instance. Precisely, for each adversary \(A\) there exists an adversary \(B\) and vice versa with

\[
\text{Adv}_{U_{\ell, k}\text{-Gen}, s}^{\text{MDDH}}(A) = \text{Adv}_{U_{\ell}\text{-Gen}, s}^{\text{MDDH}}(B) ,
\]

and \(T(A) \approx T(B)\).

**Lemma 2.5 \((D_{\ell, k}\text{-MDDH} \Rightarrow U_{\ell}\text{-MDDH})\).** Let \(\ell, k \in \mathbb{N}_+\) with \(\ell > k\) and let \(D_{\ell, k}\) be a matrix distribution. A \(U_{\ell}\)-MDDH instance is at least as hard as an \(D_{\ell, k}\) instance. Precisely, for each adversary \(A\) there exists an adversary \(B\) with

\[
\text{Adv}_{U_{\ell}\text{-Gen}, s}^{\text{MDDH}}(A) \leq \text{Adv}_{D_{\ell, k}\text{-Gen}, s}^{\text{MDDH}}(B) ,
\]

and \(T(A) \approx T(B)\).

For \(Q \in \mathbb{N}, W \in \mathbb{Z}_{q}^{k \times Q}, U \in \mathbb{Z}_{q}^{\ell \times Q}\), consider the \(Q\)-fold \(D_{\ell, k}\)-MDDH problem which is distinguishing the distributions \(([A], [AW])\) and \(([A], [U])\). That is, the \(Q\)-fold \(D_{\ell, k}\)-MDDH problem contains \(Q\) independent instances of the \(D_{\ell, k}\)-MDDH problem (with the same \(A\) but different \(w_i\)). By a hybrid argument one can show that the two problems are equivalent, where the reduction loses a factor \(Q\). The following lemma gives a tight reduction. For the uniform distribution \(U_{\ell, k}\), the security loss \(\ell - k\) can be avoided by applying Lemma 2.6 to the \(U_{\ell}\) distribution and then use Lemma 2.4 on each of the \(U_{\ell}\) instances to get a \(U_{\ell, k}\) instance.

**Lemma 2.6 (Random self-reducibility [10]).** For \(\ell > k\) and any matrix distribution \(D_{\ell, k}\), the \(D_{\ell, k}\)-MDDH assumption is random self-reducible. In particular, for any \(Q \geq 1\) and any adversary \(A\) there exists a adversary \(B\) with

\[
(\ell - k)\text{Adv}_{D_{\ell, k}\text{-Gen}, s}^{\text{MDDH}}(A) + \frac{1}{q - 1} \geq \text{Adv}_{D_{\ell, k}\text{-Gen}, s}^{\text{MDDH}, Q}(B) := |\Pr[B(G, [A], [AW] \Rightarrow 1)] - \Pr[B(G, [A], [U] \Rightarrow 1)]| ,
\]

where \(G \in G(1^\lambda), A \in D_{\ell, k}, W \in \mathbb{Z}_{q}^{k \times Q}, U \in \mathbb{Z}_{q}^{(k+1) \times Q}\), and \(T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)\), where \(\text{poly}\) is a polynomial independent of \(A\).
We recall syntax and security of a hierarchical identity-based key encapsulation mechanism (HIBKEM). We define indistinguishability (to the one from real). Moreover, we also require the distribution of $\text{usr}[\text{id}]$, $\text{udk}[\text{id}]$ generated by $\text{Ext}($sk, id$)$ and all $(K, c)$ generated by $\text{Enc}(pk, id)$:

$$\Pr[\text{Dec}(\text{usr}[\text{id}], id, C) = K] = 1.$$ 

Moreover, we also require the distribution of $\text{usr}[\text{id}][\text{id}^{p+1}]$ from $\text{Del}(\text{usr}[\text{id}], \text{udk}[\text{id}], \text{id}, \text{id}^{p+1})$ is identical to the one from $\text{Ext}(\text{sk}, \text{id}[^{\text{id}^{p+1}}]).$

In our HIBKEM definition we make the delegation key $\text{dk}$ explicit to make our constructions more readable. We define indistinguishability (IND-HID-CPA) against adaptively chosen identity and plaintext attacks for a HIBKEM via games IND-HID-CPA_{real} and IND-HID-CPA_{rand} from Figure 2.

**Definition 2.8 (IND-HID-CPA security).** A hierarchical identity-based key encapsulation scheme HIBKEM is IND-HID-CPA-secure if for all PPT $\mathcal{A}$,

$$\text{Adv}_{\text{HIBKEM}}^{\text{ind-hid-cpa}}(\mathcal{A}) := |\Pr[\text{IND-HID-CPA}_{\text{real}}^{\mathcal{A}} = 1] - \Pr[\text{IND-HID-CPA}_{\text{rand}}^{\mathcal{A}}]|$$

is negligible.

**2.3 Collision resistant hash functions**

For some constructions we make use of collision resistant hash functions.
Definition 2.9 (Hash function). A family of hash functions is a tuple $\mathcal{H} = (\text{HGen}, \text{HEval})$ of polynomial time algorithms with:

- $\text{HGen}$ is a probabilistic algorithm that gets the security parameter $1^\lambda$ and returns a (public) hash key $k$.
- $\text{HEval}$ is a deterministic algorithm that gets a hash key $k$ and an input $X \in \mathcal{D}_k$ and outputs a hash $\text{HEval}(k, X) \in \mathcal{R}_k$, where $\mathcal{D}_k$ is the domain set and $\mathcal{R}_k$ is the finite range set.

The security notion we require for our hash functions is collision resistance.

Definition 2.10 (Collision resistance). A family of hash functions $\mathcal{H} = (\text{HGen}, \text{HEval})$ is collision-resistant if for all PPT adversaries $A$,

$$\text{Adv}_\mathcal{H}^{\text{cma}}(A) := \Pr[\text{HEval}(k, X_1) = \text{HEval}(k, X_2) | (X_1, X_2) \in \mathcal{D}_k, k \in \mathcal{K}] - \Pr[A(1^\lambda, k, k) \neq \text{HGen}(1^\lambda)]$$

is negligible in $\lambda$.

3 Affine MAC with levels

The core of our HIBE constructions is a Message Authentication Code with suitable algebraic structures and we call it affine MAC with levels. This is a generalization of the delegatable, affine MAC used in [3], namely, a delegatable, affine MAC is affine MAC with levels with $\ell(p) = 1$ for all $p \in \{1, \ldots, L\}$.

Definition 3.1 (Affine MAC with levels). An affine MAC with levels MAC consists of three PPT algorithms $(\text{GenMAC}, \text{Tag}, \text{VerMAC})$ with the following properties:

- $\text{GenMAC}((G_2, q, P_2))$ gets a description of a prime-order group $(G_2, q, P_2)$ and returns a secret key $sk_{\text{MAC}} := (B, (x_{l,i,j})_{1 \leq l \leq L, 1 \leq i \leq \ell(l), 1 \leq j \leq \ell(l, i), l,i,j})$ where $B \in \mathbb{Z}_q^{n \times n'}$, $x_{l,i,j} \in \mathbb{Z}_q^p$.
- $\text{Tag}(sk_{\text{MAC}}, m) \in \mathcal{S}^{\leq L}$ returns a tag $\tau := ((t_{l,2})_{1 \leq l \leq \ell(p)}, [u]_2)$ where

$$t_{l} := B_{s_l} \quad \text{for} \quad s_l \in \mathbb{Z}_q^{n'} \quad (1 \leq l \leq \ell(p))$$

$$u := \sum_{l=1}^{\ell(p)} \left( \sum_{i=1}^{p} \sum_{j=1}^{\ell'(l, i)} f_{l,i,j}(m_i) x_{l,i,j}^T \right) t_{l} + x_{0}'$$

- $\text{VerMAC}(sk_{\text{MAC}}, m, \tau = ([t]_2, [u]_2))$ checks whether Equation (4) holds.

The messages of MAC have the form $m = (m_1, \ldots, m_p)$ where $p \leq L$ and $m_i \in \mathcal{S}$. After the transformation to an HIBE, $\mathcal{S}$ will be the base set of the identity space and $L$ will be the maximum number of levels. The functions $f_{l,i,j} : \mathcal{S}^l \rightarrow \mathbb{Z}_q$ must be public, efficiently computable functions. The parameters $\ell : \{1, \ldots, p\} \rightarrow \mathbb{N}_+$, $n, n' \in \mathbb{N}$, and $\ell' : \{1, \ldots, p\} \times \{1, \ldots, L\} \rightarrow \mathbb{N}_+$ ($1 \leq i \leq L$) are arbitrary, scheme-depending parameters. The function $\ell$ must be monotonous increasing.

Security. We require hierarchical pseudorandomness against chosen-message attacks (HPR$_0$-CMA-security) for affine MACs with levels. This is a generalization of the HPR$_0$-CMA-security for delegatable affine MACs defined in [3]. The security is defined by games in Figure 3.

Definition 3.2 (HPR$_0$-CMA security). An affine MAC with levels is HPR$_0$-CMA secure in $G_2$ if for all PPT adversaries $A$ the function

$$\text{Adv}_{\text{MAC}_{G_2}}^{\text{hpr,cma}}(A) := \Pr[\text{HPR}_0-\text{CMA}_{\text{real}}^A \Rightarrow 1] - \Pr[\text{HPR}_0-\text{CMA}_{\text{rand}}^A \Rightarrow 1]$$

is negligible.

3.1 Our first construction

Let $(G_2, q, P_2)$ be a group of prime order $q$. Our first affine MAC with levels $\text{MAC}_{\ell(\mathbb{R}, k)} := (\text{GenMAC}, \text{Tag}, \text{VerMAC})$ with message space $\mathcal{T}D := \mathcal{S}^{\leq L} := ([0, 1]^\alpha)^{\leq L}$ is defined in Figure 4. The identity vectors...
INIT: \[ sk_{\text{MAC}} \triangleq \text{Gen}_{\text{MAC}}(G_2, q, P_2) \]
\[ \text{Parse } sk_{\text{MAC}} = ([B, (x_{i,j})]_{1 \leq i \leq L, 1 \leq j \leq L}, y_0) \]
\[ dk := ([x_{i,j}]_{B}^T)_{1 \leq i \leq L, 1 \leq j \leq L} \]
\[ \text{return } ([B]_2, dk) \]

EVAL(m ∈ \Sigma^\ell): \[ Q_M = Q_M \cup \{m\} \]
\[ ([t]_2, [u]_2) \triangleq \text{Tag}(sk, m) \]
\[ \text{for } l \in \{1, \ldots, \ell(p)\}, i \in \{p+1, \ldots, L\}, j \in \{1, \ldots, \ell'(l, i)\} \text{ do} \]
\[ [d_{l,i,j}] = x_{i,j} \cdot \ell' \]
\[ \text{tdk} := ([d_{l,i,j}]_{B}^T)_{1 \leq p+1 \leq L, 1 \leq j \leq \ell'(l, i)} \]
\[ \text{return } (([t]_2)_{1 \leq \ell', \{u\}_2}, \text{tdk}) \]

CHAL(m∗ ∈ \Sigma^p): // one query
\[ h \leftarrow Z_q \]
\[ \text{for } l \in \{1, \ldots, \ell(p)\} \text{ do} \]
\[ h_{0,l} := \left( \sum_{j=1}^{l(i)} f_{i,j}^t(m_{i,j})_{x_{i,j}} \right) \cdot h \]
\[ h_1 = x_i \cdot h \in Z_q \]
\[ \text{return } ([h_1, (h_{0,l})_{1 \leq \ell(p)}, [h_1]_T) \]

FINALIZE(β ∈ \{0, 1\}): \[ \text{return } \beta \wedge (\text{Prefix}(m^*) \cap Q_M = \emptyset) \]

Figure 3: Games HPR_0-CMA_{real} and HPR_0-CMA_{rand} for defining HPR_0-CMA security for affine MACs with levels.

bit-length α and the maximum length L of the identity vectors can be chosen freely.\(^3\) The resulting HIBE from this MAC has constant ciphertext length.

MAC\(1[\mathcal{U}_{3k,k}]\) has n := 3k and n′ := k where k ∈ \mathbb{N}^+ can be chosen arbitrary. To match the formal definition, x_{i,j} should be renamed to x_{i,j−b} and f_{i,j−b}(m_{i,j}) := \left( \left[ m_{i,j} \right]_j^T \right) = b. Then we get \ell(p) = 1 and \ell'(1, i) = 2α.

**Theorem 3.3 (Security of MAC\(1[\mathcal{U}_{3k,k}]\)).** MAC\(1[\mathcal{U}_{3k,k}]\) is tightly HPR_0-CMA secure in G_2 under the \mathcal{U}_k-MDDH assumption for G_2. Precisely, for all adversaries A there exists an adversary B with

\[ \text{Adv}^{\text{HPR}_{0-CMA}}_{\text{MAC}([\mathcal{U}_{3k,k}], G_2)}(A) \leq \left( 4(\alpha + 1)L + 4\alpha L^2 \right) \left( \text{Adv}^{\text{MDDH}}_{\mathcal{U}_k, \text{Gen}_{G_2}}(B) + \frac{1}{q-1} \right) + \frac{Q}{q^{2k}} \]

\[ \text{and } T(B) \approx T(A) + Q \cdot \text{poly}(\lambda), \text{ where } Q \text{ denotes the number of EVAL queries of } A \text{ and } \text{poly is a polynomial independent of } A. \]

**Proof.** The proof uses a hybrid argument with the hybrids G_0 (the HPR_0-CMA_{real} game), G_1, G_2,i,0, G_2,i,1, G_2,i,2,0,G_2,i,2,1, G_2,i,3,1, G_2,i,4, and G_2,i,5 for i ∈ \{1, \ldots, L\} and j ∈ \{1, \ldots, \alpha\}, and finally G_3. The hybrids are given in Figure 5 and 6. A summary can be found in Table 2. They make use of random functions RF_{i,j} : \{0, 1\}^* → Z_q^{k \times 2k}, RF_{i,j}^{(0)} : \{0, 1\}^* → Z_q^{2k}, and RF_{i,j}^{(1)} : \{0, 1\}^* → Z_q^{1 \times k}, defined on-the-fly.

**Lemma 3.4 (G_0 ↔ G_i).**
\[ \text{Pr}[G_0^4 \Rightarrow 1] = \text{Pr}[G_i^4 \Rightarrow 1] \]

**Proof.** In game G_1 each time the adversary queries a tag for a message m where he queried a tag for m before, the adversary will get a rerandomized version of the first tag he queried. The rerandomized tag is identically distributed to a fresh tag: t′ := t + Bs′ is uniformly random in \text{Span}(B), when s′ is uniformly random in Z_q^{k}. Together with u′ := u + \sum_{i=1}^{p} \left( \sum_{j=1}^{\alpha} x_{i,j}^T m_{i,j} \right) Bs′ we get a valid message tag for m, when ([t]_2, [u]_2) is a valid tag for m.

Note that the rerandomization uses only the “public key” returned by the INIT oracle, so it could actually be carried out by the adversary herself. To put it in a nutshell, repeated EVAL queries for a message m will leak no information, that is not already leaked by the first EVAL query or m by the “public key.”\(^4\)

\(^3\)A different bitlength on each level is possible as well, but we assume it is α on each level to ease notation.
\(^4\)The same technique can be used to prove the IBE of [3] secure with duplicated EVAL-queries. Thus they work without a pseudorandom function.
Lemma 3.5 (G_1 \rightsquigarrow G_{2,1,0})

Pr[G_1^A \Rightarrow 1] = Pr[G_{2,1,0}^A \Rightarrow 1]

Proof. These two games are equivalent.

Legend 3.6 (G_{2,1,0} \rightsquigarrow G_{2,1,1}). For all adversaries \( A \) there exists an adversary \( B \) with

\[
|Pr[G_{2,1,0}^A \Rightarrow 1] - Pr[G_{2,1,1}^A \Rightarrow 1]| \leq \text{Adv}_{\text{mdth}}^{U_3,Gen,G_2}(B) + \frac{1}{q - 1}
\]

and \( T(B) \approx T(A) + Q \cdot \text{poly}(\lambda) \).

Proof. These two games are equivalent except that in EVAL-queries with \( p = i \) the value \( t \) is chosen uniformly random from \( \text{Span}(B) \) in \( G_{2,1,0} \) and uniformly random from \( \mathbb{Z}^{3k} \) in game \( G_{2,1,1} \). Since for all computed values it is enough to have \( |B|_2 \) instead of \( B \), this leads to a straightforward reduction to the QL-fold \( U_{3,k} \)-MDDH assumption. Remember that by Lemma 2.4, the \( U_{3,k} \)-MDDH assumption is equivalent to the \( U_k \)-MDDH assumption.

The running time of \( B \) is dominated by the running time of \( A \) plus some (polynomial) overhead that is independent of \( T(A) \) for the group operations in each oracle query.

Lemma 3.7 (G_{2,1,1} \rightsquigarrow G_{2,1,3}). For all \( i \in \{1, \ldots, L\}, j \in \{1, \ldots, \alpha - 1\} \) and all adversaries \( A \) there exists an adversary \( B \) with

\[
|Pr[G_{2,1,1}^A \Rightarrow 1] - Pr[G_{2,1,3}^A \Rightarrow 1]| \leq 4i\alpha \left( \text{Adv}_{\text{mdth}}^{U_3,Gen,G_2}(B) + \frac{1}{q - 1} \right)
\]

and \( T(B) \approx T(A) + Q \cdot \text{poly}(\lambda) \).

Proof. To prove this transition, we introduce new hybrids \( G_{2,1,2,j,1}, G_{2,1,2,j,2} \) and \( G_{2,1,2,j,3} \) for \( i \in \{1, \ldots, L\} \) and \( j \in \{1, \ldots, \alpha - 1\} \). The hybrids are given in Figure 6.

Lemma 3.7 follows directly from Lemma 3.8-3.13.

Lemma 3.8 (G_{2,1,1} \rightsquigarrow G_{2,1,2,0,0}).

Pr[G_{2,1,1}^A \Rightarrow 1] = Pr[G_{2,1,2,0,0}^A \Rightarrow 1]
The algorithm \texttt{RerandomizeTag} is only helper function and not an oracle for the adversary.
Figure 6: Hybrids for the transition from $G_{2,i,1}$ to $G_{2,i,j+1}$. The notion $a := b$ is shorthand for $a := a + b$. The algorithm RerandomizeTag is defined in Figure 5.
Table 2: Summary of the hybrids of Figure 5 and 6. Eval queries with \( p = i \) draw \( t \) from the set described by the second column and add the randomness \( r_u(m)t \) to \( u \) or choose \( u \) uniform random. The Chal query adds the term \( r_{h_0}(m^*)^T \) to \( h_0 \) if \( m^* \) has length \( \geq i \). The column “Transition” displays how we can switch to this hybrid from the previous one. The background colors indicate repeated transitions.

![Table](image)

**Proof.** These two games are equivalent. When changing in \( G_{2,i,1} \) the secret values \( x_{i,1,b} \) to \( x_{i,1,b} + B^T(\text{RF}_{i,0}(c)) \) (for \( b \in \{0,1\} \), we get game \( G_{2,i,2,0,0} \). The distribution of \( x_{i,1,b} \) and \( x_{i,1,b} + B^T(\text{RF}_{i,0}(c)) \) is identical. Note that the term \( B^T(\text{RF}_{i,0}(c)) \) cancels out in the master public key and in the user delegation keys of Eval-queries with \( p < i \).

**Lemma 3.9** \((G_{2,i,2,j,0} \rightsquigarrow G_{2,i,2,j,1})\). For all adversaries \( A \) there exists an adversary \( B \) with

\[
|\Pr[G_{2,i,2,j,0}^A \Rightarrow 1] - \Pr[G_{2,i,2,j,1}^A \Rightarrow 1]| \leq 2 \left( \text{Adv}_{\text{MDDH}}^{\text{mddh}}(G_{\text{Gen},G_0},B) + \frac{1}{q-1} \right)
\]

and \( T(B) \approx T(A) + Q \cdot \text{poly}(\lambda) \).

**Proof.** These two games are equivalent except that the value \( t \) is generated uniformly random from \( \mathbb{Z}_q^{3k} \) in game \( G_{2,i,2,j,0} \) and from either \( \text{Span}(B|B_0) \) or \( \text{Span}(B|B_1) \) depending on the bit \([m]_{j+1} \) in game \( G_{2,i,2,j,1} \). We can switch from \( G_{2,i,2,j,0} \) to \( G_{2,i,2,j,1} \) with two \( Q \)-fold \( U_{3k,k} \)-MDDH challenges. Remember that the \( U_{3k,k} \)-MDDH assumption is equivalent to the \( U_k \)-MDDH assumption by Lemma 2.4.

To achieve that, we first switch \( t \) for \([m]_{j+1} = 0 \) from a random vector in \( \mathbb{Z}_q^{3k} \) to \( t := Bs_1 + s_2 \) where \( s_1 \overset{\$}{\in} \mathbb{Z}_q^{3k} \) and \( s_2 \overset{\$}{\in} \mathbb{Z}_q^{3k} \). This change is only conceptual. Then we change \( s_2 \) from a random vector in \( \mathbb{Z}_q^{3k} \) to a random vector in the span of \( B_0 \) via the MDDH assumption. More precisely, let \((B_0|_2, |Z|_2) \in G_2^{3k \times (k+q)} \) be a \( Q \)-fold \( U_{3k,k} \)-MDDH challenge. For the \( i \)-th Eval query with \([m]_{j+1} = 0 \), the reduction \( B \) computes \([t]_2 := [Bs_1 + |Z|_2]_2 \), where \( s_1 \overset{\$}{\in} \mathbb{Z}_q^{3k} \) and \( |Z|_i \) is the \( i \)-th column vector of \( Z \). Furthermore, in order to make sure that the column vectors of \((B|B_0|B_1) \) form a random basis of \( \mathbb{Z}_q^{3k} \),
the reduction $\mathcal{B}$ chooses $B, B_1 \in U_{\ell b, k}$ such that $(B|B_1)$ has rank $2k$ and $(B|B_1)^\perp b = 0$ for all column vectors $b$ of $B_0$. We note that the latter one can be done over group $G_2$ by knowing $B$ and $B_1$ over $\mathbb{Z}_q$.

Until now, if $Z$ is uniform then $\mathcal{B}$ simulates the game $G_{2, i, 2, j, 0}$, else if $Z$ is from Span$(B_0)$ then $\mathcal{B}$ simulates the game $G_{2, i, 2, j, 1}$ for messages with $[m]_{j+1} = 0$.

By using the same argument, we can switch $t$ for $[m]_{j+1} = 1$ from a random vector in $\mathbb{Z}_q^{3k}$ to a random vector in Span$(B_1)$.

The running time of $\mathcal{B}$ is dominated by the running time of $A$ plus some (polynomial) overhead that is independent of $T(A)$ for the group operations in each oracle query.

Lemma 3.10 ($G_{2, i, 2, j, 1} \approx G_{2, i, 2, j, 2}$).

$\Pr[G_{2, i, 2, j, 1} \Rightarrow 1] = \Pr[G_{2, i, 2, j, 2} \Rightarrow 1]$.

Proof. First of all, we replace in game $G_{2, i, 2, j, 1}$ the term $RF_{i,j}((m)_{j}) (B^\perp)^\top$ with $RF_{i,j}((m)_{j}) (B_0^\perp)^\top + RF_{i,j}^\perp((m)_{j}) (B_1^\perp)^\top$. This does not change the distribution, since $B_0^\perp, B_1^\perp$ is a basis for Span$(B^\perp)$.

We define

$$RF_{i,j}^\perp((m)_{j+1}) := \begin{cases} RF_{i,j}((m)_{j}) & \text{if } [m]_{j+1} = 0 \\ RF_{i,j}((m)_{j}) + RF_{i,j}^\perp((m)_{j}) & \text{if } [m]_{j+1} = 1 \end{cases}$$

where $RF_{i,j}^\perp : \{0, 1\}^{j+1} \to \mathbb{Z}_q^{1 \times k}$ is another independent random function. Since $RF_{i,j}^\perp$ does not appear in game $G_{2, i, 2, j, 2}$ anymore, $RF_{i,j}^\perp$ is a random function.

EVAL queries with $p \neq i$ use the same code in both games and EVAL queries with $p = i$ and $[m]_{j+1} = 0$ are distributed identically in both games, by definition of $RF_{i,j}^\perp$.

EVAL queries with $p = i$ and $[m]_{j+1} = 1$ are distributed identically in both games, since for those queries $t \in \text{Span}(B_1)$ and both $B$ and $B_1$ are orthogonal to $B_0^\perp$ and thus $RF_{i,j}^\perp((m)_{j+1}) (B_0^\perp)^\top t = 0$.

The CHAL query uses the same code if $p \neq i$ and otherwise it is distributed identically if $[m^*]_{j+1} = 0$. For the case $[m^*]_{j+1} = 1$ note that $x_{i,j+1,1}$ is identically distributed as $x_{i,j+1,1} + B_i^\perp w$ for $w \in \mathbb{Z}_q^{k}$ and $w$ is hidden from the adversary except for the CHAL query: In all EVAL queries with $p \neq i$ only $x_{i,j+1,1}$ is used and thus the $B_i^\perp$-part cancels out. In the EVAL queries with $p = i$ there is either $[m]_{j+1} = 0$ which means that $x_{i,j+1,1}$ is not used to compute the tag or there is $[m]_{j+1} = 1$ which means that $t \in \text{Span}(B_i|B_1)$ and thus the $B_i^\perp$-part of $x_{i,j+1,1}$ cancels out. All in all this means that the value $h_0$ is the only one in the game that depends on $w$ and thus the $B_i^\perp$-part of $h_0$ is uniformly random to the adversary. Especially $h_0$ is distributed identically in both games.

Lemma 3.11 ($G_{2, i, 2, j, 2} \approx G_{2, i, 2, j, 3}$).

$\Pr[G_{2, i, 2, j, 2} \Rightarrow 1] = \Pr[G_{2, i, 2, j, 3} \Rightarrow 1]$.

Proof. We define

$$RF_{i,j}^{(1)}((m)_{j+1}) := \begin{cases} RF_{i,j}^{(1)}((m)_{j}) & \text{if } [m]_{j+1} = 0 \\ RF_{i,j}^{(1)}((m)_{j}) + RF_{i,j}^{(1)}((m)_{j}) & \text{if } [m]_{j+1} = 1 \end{cases}$$

where $RF_{i,j}^{(1)} : \{0, 1\}^{j+1} \to \mathbb{Z}_q^{1 \times k}$ is another independent random function. Since $RF_{i,j}^{(1)}$ is not used in game $G_{2, i, 2, j, 3}$, $RF_{i,j}^{(1)}$ is a random function.

The argument, that the games $G_{2, i, 2, j, 2}$ and $G_{2, i, 2, j, 3}$ are identically distributed, is the same as in Lemma 3.10, just with the roles of 0 and 1 swapped.

Lemma 3.12 ($G_{2, i, 2, j, 3} \approx G_{2, i, 2, j+1, 0}$). For $j < i \alpha$ and all adversaries $A$ there exists an adversary $B$ with

$$|\Pr[G_{2, i, 2, j, 3} \Rightarrow 1] - \Pr[G_{2, i, j+1} \Rightarrow 1]| \leq 2A_{\text{adv}}^{\text{adv}}_{i, k, G_{2, i, j+1}} + \frac{1}{q - 1}$$

and $T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)$.
Proof. In game $G_{2,i,2,i,3}$, replace the term $RF_{i,j+1}^{(0)} \left( \|m\|_{i+1} \right) (B_0^i)^\top + RF_{i,j+1}^{(1)} \left( \|m\|_{i+1} \right) (B_1^i)^\top$ with $RF_{i,j} \left( \|m\|_j \right) (B^j)$ to avoid computing $B_0^i$ and $B_1^i$. This does not change the distribution, since $B_0^i, B_1^i$ is a basis for $\text{Span}(B^j)$. The remaining transition is the reverse of Lemma 3.9.

Lemma 3.13 ($G_{2,i,2,i,0,3} \leadsto G_{2,i,3}$). Let $Q_i$ denote the number of $\text{Eval}$ queries with $p = i$.

$$|\Pr[G_{2,i,2,i,0,3} \Rightarrow 1] - \Pr[G_{2,i,3} \Rightarrow 1]| \leq \frac{Q_i}{q^2}.$$  

Proof. In game $G_{2,i,2,i,0,3}$ the $CHAL$-query evaluates $RF_{i,io}$ only for the input value $m^*_i$ of the $i$th adversary (if $p \geq i$, otherwise it does not use $RF_{i,io}$ at all). Assume $\text{Prefix}(m^*) \cap \mathcal{Q}_{A_4} = \emptyset$, otherwise the adversary has lost the $\text{Eval}$ query anyway. In each user secret key query with $p = i$ the value $RF_{i,io}(m)(B^j)^i \cdot t$ is part of $u$. This is the only place where $RF_{i,io}(m)$ is used, since only the first $\text{Eval}$-query for each message evaluates the random function. Thus each query outputs a uniformly random value for $u$ when $t_p \not\in \text{Span}(B)$, which happens with probability $1 - 1/(q^{2k})$. In this case the games are distributed identically.

Lemma 3.14 ($G_{2,i,3} \leadsto G_{2,i,4}$).

$$\Pr[G_{2,i,3} \Rightarrow 1] = \Pr[G_{2,i,4} \Rightarrow 1]$$  

Proof. The games execute the same code if $p < i$ and otherwise we can argue that $x_{i,1,[m^*]} \cdot B_2 - (RF_{i,io}({m^*}^i))^\top$ are identical distributed. All $\text{Eval}$ queries and the “public key” returned by $\text{INIT}$ make only use of $x_{i,1,[m^*]} B$, so the $B^j(RF_{i,io}(\cdot))^i$ part cancels out.

Lemma 3.15 ($G_{2,i,4} \leadsto G_{2,i,5}$). For all adversaries $A$ there exists an adversary $B$ with

$$|\Pr[G_{2,i,4} \Rightarrow 1] - \Pr[G_{2,i,5} \Rightarrow 1]| \leq \frac{\lambda d_{\text{mdih}}}{\mathbf{Adv}_{U_{hs},G_{Gen},G_{2}}(B)} + \frac{1}{q-1}$$  

and $T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)$.

Proof. The transition is the reverse of Lemma 3.6.

Lemma 3.16 ($G_{2,i,5} \leadsto G_{2,i+1,0}$). For $i < L$

$$\Pr[G_{2,i,5} \Rightarrow 1] = \Pr[G_{2,i+1,0} \Rightarrow 1].$$  

Proof. These two games are equivalent.

Lemma 3.17 ($G_{2,L,5} \leadsto G_{3}$).

$$\Pr[G_{2,L,5} \Rightarrow 1] = \Pr[G_{3} \Rightarrow 1]$$  

Proof. In game $G_{2,L,5}$ the value $x_0'$ is only used to compute $h_1$, thus $h_1$ is a uniform random value to $A$ and the games are distributed identical.

Summary. To prove Theorem 3.3, we combine Lemmas 3.4–3.17 to change $h_1$ from real to random and then apply all Lemmas in reverse order to get to the $\text{HPR}_0$-$\text{CMA}_{\text{rand}}$ game.

Optimization. $\text{MAC}_1[\mathcal{U}_{sk,k}]$ can be improved with a collision-resistant hash function. In $\text{MAC}_1[\mathcal{U}_{sk,k}]$ in a tag for $m$ we add to the value $u$ for the $i$-th level $N_i f_j(m_i) x_j^i$. The idea is to replace this by $\sum_j f_j(H(m_j)) x_j^i$ for a collision resistant hash function $H$.

Formally, we need a family of hash functions $H = (H_{GGen},H_{\text{Eval}})$ with domain $\mathcal{S} \leq L$ and range $\{0,1\}^\gamma$ for all hash keys. The affine MAC $\text{MAC}_1^H[\mathcal{U}_{sk,k}]$ is shown in Figure 7.

Theorem 3.18 (Security of $\text{MAC}_1^H[\mathcal{U}_{sk,k}]$. $\text{MAC}_1^H[\mathcal{U}_{sk,k}]$ is tightly $\text{HPR}_0$-$\text{CMA}$ secure in $G_2$ when $H$ is collision resistant and the $\mathcal{U}_{sk}$-MDDH assumption holds for $G_2$. Precisely, for all adversaries $A$ there exist $B$ and $C$ with

$$\lambda d_{\text{mdih}} \mathbf{Adv}_{\text{HPR}_0,\text{CMA}}(A) \leq (4L + 8\gamma L) \left( \mathbf{Adv}_{\text{mdih}}(H_{GGen},G_{2}) + \frac{1}{q-1} \right) + L \cdot \mathbf{Adv}_{\text{CMA}}(C) + \frac{Q}{q^{2k}}$$  

and $T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)$ and $T(C) \approx T(A) + Q \cdot \text{poly}(\lambda)$, where $Q$ denotes the number of $\text{Eval}$ queries of $A$ and $\text{poly}$ is a polynomial independent of $A$. 

16
Table 3. They make use of random functions $\mathbf{V}_{i,j} : \{0,1\}^j \rightarrow \mathbb{Z}_q^{1 \times 2k}$, $\mathbf{R}^{(0)}_{i,j} : \{0,1\}^j \rightarrow \mathbb{Z}_q^{1 \times k}$ and $\mathbf{R}^{(1)}_{i,j} : \{0,1\}^j \rightarrow \mathbb{Z}_q^{1 \times k}$, defined on-the-fly.

Figure 7: Our first affine MAC improved with a hash function.

We omit the proof of this theorem, since it is very similar to the proof of Theorem 3.3. The main idea is rather standard: We use the collision resistance of $H$ to reject the collision and then we apply the hybrid argument on the hash output.

3.2 Our second construction

Let $(\mathcal{G}_2, q, P_2)$ be a group of prime order $q$. Our second affine MAC with levels $\text{MAC}_2[\mathcal{U}_{3,k}, k] := (\text{Gen}_{\text{MAC}}, \text{Tag}, \text{Ver}_{\text{MAC}})$ with message space $TD := S^{\leq L} := \{(0,1)^{\leq k}\}$ is defined in Figure 8. The identity vectors bit-length $\alpha$ and the maximum length $L$ of the identity vectors can be chosen freely. The difference to the first construction is that this MAC uses a different $t_i$ on each level $(\ell(p) = p)$ and thus needs no delegation keys. This leads to shorter user secret keys and allows a more efficient reduction. However, this comes at the price of larger ciphertexts. Formally, this MAC uses $\ell'(l, i) = 0$ for $i < p$ and $\ell'(l, i) = 2i\alpha$ for $i = p$.

Theorem 3.19 (Security of $\text{MAC}_2[\mathcal{U}_{3,k}, k]$). $\text{MAC}_2[\mathcal{U}_{3,k}, k]$ is tightly HPR$_0$-CMA secure in $\mathcal{G}_2$ under the $\mathcal{U}_k$-MDDH assumption for $\mathcal{G}_2$. Precisely, for all adversaries $\mathcal{A}$ there exists an adversary $B$ with

$$\text{Adv}_{\text{MAC}_2[\mathcal{U}_{3,k}, k], \mathcal{G}_2}^\text{hpr-cma}(\mathcal{A}) \leq (2 + 8\alpha L) \left(\text{Adv}_{\mathcal{U}_k, \mathcal{G}_2}^\text{mdh}(\mathcal{B}) + \frac{1}{q - 1}\right) + \frac{Q}{q^{2k}}$$

and $T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)$, where $Q$ denotes the number of $\text{Eval}$ queries of $\mathcal{A}$ and $\text{poly}$ is a polynomially independent of $\mathcal{A}$.

Proof. The proof uses a hybrid argument with the hybrids $G_0$ (the HPR$_0$-CMA$_{\text{real game}}$), $G_1$, $G_2$, $G_{3,j}$ for $j \in \{1, \ldots, L_o\}$, $G_{3,1} - G_{3,3,j}$ for $j \in \{1, \ldots, L_o - 1\}$, and finally $G_4$. The hybrids are given in Figure 9 and Figure 10. A summary can be found in Table 3.

The proof uses a hybrid argument with the hybrids $G_0$ (the HPR$_0$-CMA$_{\text{real game}}$), $G_1$, $G_2$, $G_{3,j}$ for $j \in \{1, \ldots, L_o\}$ and finally $G_4$. The hybrids are given in Figure 9. A summary can be found in Table 3. They make use of random functions $\mathbf{R}^{(0)}_{i,j} : \{0,1\}^j \rightarrow \mathbb{Z}_q^{1 \times 2k}$, $\mathbf{R}^{(1)}_{i,j} : \{0,1\}^j \rightarrow \mathbb{Z}_q^{1 \times k}$ and $\mathbf{R}^{(1)}_{i,j} : \{0,1\}^j \rightarrow \mathbb{Z}_q^{1 \times k}$, defined on-the-fly.
These two games are equivalent except that the values
Lemma 3.22 have
Span
queries for one message
Lemma 3.20 The games are equivalent. When changing in
Proof. In game \( G_1 \) each time the adversary queries a tag for a message \( m \) and he queried a tag for \( m \) before, the adversary will get a rerandomized version of the first tag he queried. The rerandomized tag is identically distributed to a fresh tag: \( t'_i := t_i + B s_i \) is uniformly random in \( \text{Span}(B) \), when \( s_i' \) is uniform random in \( \mathbb{Z}_q^v \). Together with \( u' := u + \sum_{i=1}^p \left( \sum_{j=1}^\alpha x_{i,j,[m]} \right) t_i + x_0' \) we get a valid message tag for \( m \), when \( \left( (t_i)_{i \leq i \leq p} | u \right) \) is a valid tag for \( m \).

Note that the rerandomization can be carried out by just using the “public key” returned by the \text{INIT} oracle, so it could actually be carried out by the adversary herself. To put it in a nutshell, repeated \text{EVAL} queries for one message \( m \) will leak no information, that is not already leaked by the first \text{EVAL} query for \( m \) or by the “public key”.

Lemma 3.21 (\( G_1 \to G_2 \)). For all adversaries \( \mathcal{A} \) there exists an adversary \( \mathcal{B} \) with

\[
\Pr[ \mathcal{G}_1^A \Rightarrow 1 ] - \Pr[ \mathcal{G}_2^A \Rightarrow 1 ] \leq \text{Adv}_\text{mddh}^{\mathcal{A}, \mathcal{G}_1^A, \mathcal{G}_2^A}(\mathcal{B}) + \frac{1}{q-1}
\]

and \( T(\mathcal{B}) \approx T(\mathcal{A}) + Q \cdot \text{poly}(\lambda) \).

Proof. These two games are equivalent except that the values \( t_i \) are generated uniformly random from \( \text{Span}(B) \) in \( G_1 \) and uniformly random from \( \mathbb{Z}_q^v \) in game \( G_2 \). Since for all computed values it is enough to have \( [B]_2 \) instead of \( B \), this leads to a straightforward reduction to the QL-fold \( U_{k,k}^-\text{-MDDH} \) assumption. Remember that by Lemma 2.4, the \( U_{k,k}^-\text{-MDDH} \) assumption is equivalent to the \( U_k \) assumption.

The running time of \( \mathcal{B} \) is dominated by the running time of \( \mathcal{A} \) plus some (polynomial) overhead that is independent of \( T(\mathcal{A}) \) for the group operations in each oracle query.

Lemma 3.22 (\( G_2 \to G_{3,0} \)).

\[
\Pr[ \mathcal{G}_3^A \Rightarrow 1 ] = \Pr[ \mathcal{G}_{3,0}^A \Rightarrow 1 ]
\]

Proof. The games are equivalent. When changing in \( G_2 \) the secret values \( x_{i,1,b} \) to \( x_{i,1,b} + B^\top(\text{RF}_{i,0}(\varepsilon))^\top \) (for all \( i \in \{1, \ldots, L\} \) and \( b \in \{0,1\} \), we get game \( G_{3,0} \). The distribution of \( x_{i,1,b} \) and \( x_{i,1,b} + B^\top(\text{RF}_{i,0}(\varepsilon))^\top \) is identical. Note that the term \( B^\top(\text{RF}_{i,0}(\varepsilon))^\top \) cancels out in the public key.
The hybrids are given in Figure 10.

Proof. The first challenge is used to change the distribution of $(B, B_0)$. For all adversaries $A$ there exists an adversary $B$ with

$$\Pr[A_{\mathcal{G}_j} \Rightarrow 1] - \Pr[A_{\mathcal{G}_{j+1}} \Rightarrow 1] \leq 4 \left( \text{Adv}_{\mathcal{U}_k, \mathcal{G}, \mathcal{G}_2}^{\text{mddh}}(B) + \frac{1}{q - 1} \right)$$

and $T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)$.

**Proof.** To prove this transition, we introduce new hybrids $\mathcal{G}_{3,j,1}$, $\mathcal{G}_{3,j,2}$ and $\mathcal{G}_{3,j,3}$ for $j \in \{1, \ldots, \hat{i}_0 - 1\}$. The hybrids are given in Figure 10. Lemma 3.23 follows directly from Lemma 3.24-3.27.

**Lemma 3.24** ($G_{3,j} \leadsto G_{3,j+1}$). For all adversaries $A$ there exists an adversary $B$ with

$$\Pr[A_{\mathcal{G}_{3,j}} \Rightarrow 1] - \Pr[A_{\mathcal{G}_{3,j+1}} \Rightarrow 1] \leq 2 \left( \text{Adv}_{\mathcal{U}_k, \mathcal{G}, \mathcal{G}_2}^{\text{mddh}}(B) + \frac{1}{q - 1} \right)$$

and $T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)$.

**Proof.** These two games are equivalent except that the values $t_i$ are generated uniformly random from $\mathbb{Z}_{q}^{3k}$ in game $\mathcal{G}_{3,j}$ and from $\text{Span}(B|B_0)$ respectively $\text{Span}(B|B_1)$ in game $\mathcal{G}_{3,j,1}$ for $i \in \{i, \ldots, p\}$. We can switch from $\mathcal{G}_{3,j}$ to $\mathcal{G}_{3,j,1}$ with two $QL$-fold $\mathcal{U}_{k,k}$-$\text{MDDH}$ challenges. Remember that the $\mathcal{U}_{k,k}$-$\text{MDDH}$ assumption is equivalent to the $\mathcal{U}_k$ assumption by Lemma 2.4.

The first challenge is used to change the distribution of $t_i$ from $\mathbb{Z}_{q}^{3k}$ to $\text{Span}(B|B_0)$ for the $\text{Eval}$-queries with $[m]_{j+1} = 0$. Therefore, on input of a $QL$-fold $\mathcal{U}_{k,k}$-$\text{MDDH}$ challenge $([B_0]_2, [Z]_2)$ where $B_0 \in \mathbb{Z}_q^{3k \times k}$ and the column vectors of $Z \in \mathbb{Z}_q^{3k \times QL}$ are either uniform random from $\mathbb{Z}_q^{3k}$ or uniform random from $\text{Span}(B_0)$. We can now switch from $\mathcal{G}_{3,j}$ to an intermediate hybrid, where the $\text{Eval}$ queries with $[m]_{j+1} = 0$ are distributed as in $\mathcal{G}_{3,j,1}$ and everything else is distributed as in game $\mathcal{G}_{3,j}$. Therefore, first change in game $\mathcal{G}_{3,j}$ the generation of those $t_i$ with $i \geq 1$ in $\text{Eval}$ queries with $[m]_{j+1} = 0$ to $s_1 \leftarrow \mathbb{Z}_q^2$; $s_2 \leftarrow \mathbb{Z}_q^{3k}$; $t := Bs_1 + s_2$. Adversary $B$ now derives $B_0$ from the $\mathcal{U}_k$-$\text{MDDH}$ challenge and generates $B, B_1 \leftarrow \mathcal{U}_{k,k}$ such that $(B, B_1)$ has rank $2k$ and $(B|B_1)^\top b = 0$ for all column vectors $b$ of $B_0$ (this can be checked over group $G_2$). Like this, the column vectors of $B, B_0$, and $B_1$ form a random basis of $\mathbb{Z}_q^{3k}$.
Proof. First of all, replace in game \( G_{3,j,1} \) the term \( \text{RF}_{i,j} \left( \left[ m \right]_{ij} \right) \left( B^\perp \right)^T \) with \( \text{RF}_{i,j}^{(0)} \left( \left[ m \right]_{ij} \right) \left( B^*_0 \right)^T + \text{RF}_{i,j}^{(1)} \left( \left[ m \right]_{ij} \right) \left( B^*_1 \right)^T \). This doesn’t change the distribution, since \( B^*_0, B^*_1 \) is a basis for \( \text{Span}(B^\perp) \).

We define

\[
\text{RF}_{i,j}^{(0)} \left( \left[ m \right]_{ij} \right) := \begin{cases} 
\text{RF}_{i,j}^{(0)} \left( \left[ m \right]_{ij} \right) & \text{if } [m]_{ij} = 0 \\
\text{RF}_{i,j}^{(0)} \left( \left[ m \right]_{ij} \right) + \text{RF}_{i,j}^{(0)} \left( \left[ m \right]_{ij} \right) & \text{if } [m]_{ij} = 1
\end{cases}
\]

Figure 9: Hybrids for the security proof of \( \text{MAC}_2[U_{6k,k}] \). The notation \( a + b \) is shorthand for \( a := a + b \).

Now set in the \( \nu \)-th \textsc{Eval} query \( t_i := B_{s_i} + Z_i[iL + i] \) where \( s_i \in \mathbb{Z}_q^k \) and \( Z_i[i] \) is the \( i \)-th column vector of \( Z \). If the column vectors of \( Z \) are uniform random from \( \mathbb{Z}_q^k \), \( B \) is simulating game \( G_{3,j,j} \). Otherwise, if the column vectors of \( Z \) are uniform random from \( \text{Span}(B_0) \), \( t \) is uniformly random from \( \text{Span}(B_0) \) and \( B \) is simulating the intermediate hybrid.

We proceed analogous to switch from the intermediate hybrid to game \( G_{3,j,1} \).

The running time of \( B \) is dominated by the running time of \( A \) plus some (polynomial) overhead that is independent of \( T(A) \) for the group operations in each oracle query. \( \square \)

Lemma 3.25 \( (G_{3,j,1} \Rightarrow G_{3,j,2}) \).

\[
\Pr[G_{3,j,1}^A \Rightarrow 1] = \Pr[G_{3,j,2}^A \Rightarrow 1]
\]

Proof. First of all, replace in game \( G_{3,j,1} \) the term \( \text{RF}_{i,j} \left( \left[ m \right]_{ij} \right) \left( B^\perp \right)^T \) with \( \text{RF}_{i,j+1}^{(0)} \left( \left[ m \right]_{ij} \right) \left( B^*_0 \right)^T + \text{RF}_{i,j}^{(1)} \left( \left[ m \right]_{ij} \right) \left( B^*_1 \right)^T \). This doesn’t change the distribution, since \( B^*_0, B^*_1 \) is a basis for \( \text{Span}(B^\perp) \).

We define

\[
\text{RF}_{i,j+1}^{(0)} \left( \left[ m \right]_{ij} \right) := \begin{cases} 
\text{RF}_{i,j}^{(0)} \left( \left[ m \right]_{ij} \right) & \text{if } [m]_{ij} = 0 \\
\text{RF}_{i,j}^{(0)} \left( \left[ m \right]_{ij} \right) + \text{RF}_{i,j}^{(0)} \left( \left[ m \right]_{ij} \right) & \text{if } [m]_{ij} = 1
\end{cases}
\]
Figure 10: Hybrids for the transition from $G_{3,j}$ to $G_{3,j+1}$. The notion $a + b$ is shorthand for $a := a + b$. 

\[ G_{3,j} \xrightarrow{\mathbf{h}} G_{3,j+1} \]

**INIT:**

\[
\begin{align*}
\mathbf{B} &\colon [B_0, B_1, B_0^*, B_1^*] \triangleq \mathbb{U}_{6,k} \\
\end{align*}
\]

with the constrains:

- $\mathbf{B}, \mathbf{B}_0, \mathbf{B}_1$ is a basis for $\mathbb{Z}^k_q$
- $\mathbf{B}_0^*, \mathbf{B}_1^*$ is a basis for $\text{Span}(\mathbf{B}^\perp)$

- $\mathbf{B}_1^* = \mathbf{B}_0^* \mathbf{B}_1$

for $i \in \{1, \ldots, L\}, j \in \{1, \ldots, i\alpha\}, b \in \{0, 1\}$ do

\[
\begin{align*}
\mathbf{x}_{i,j,b} &\triangleq \mathbb{Z}^k_q \\
K &\colon \mathcal{S}^L \rightarrow ((\mathbb{Z}^k_q)^e \times \mathbb{G}_2) \\
\mathbf{R}_{\mathbf{f}_{i,j}} &\colon \mathbb{Z}^k_q \\
\end{align*}
\]

return $\left( [\mathbf{B}]_{2^i}, ([\mathbf{x}_{i,j,b}]_{2^i})_{1 \leq i \leq L, 1 \leq j \leq i, b \in \{0, 1\}} \right)$

**CHAL($m^* \in \mathcal{S}^p$):**

\[ h \triangleq \mathbb{Z}^k_q \]

for $i \in \{1, \ldots, p\}$ do

\[ h_{0,i} := \sum_{j=1}^{i} \mathbf{x}_{i,j,[m^*]} ] / h \]

for $i \in \{1, \ldots, \min(p, i-1)\}$ do

\[ h_{0,i} := RF_{i-\alpha} \left( \left[ \mathbf{m}^*_j \right]_i \right) (\mathbf{B}^\perp)^\top h \]

\[ i := \lfloor (j+1)/\alpha \rfloor \]

for $i \in \{1, \ldots, p\}$ do

\[ h_{0,i} := h_0, i \]

\[ + \mathbf{B}^\perp RF_{i,j} \left( \left[ \mathbf{m}^*_j \right]_i \right)^\top h \]

\[ h_{1,i} := h_0, i \]

return $\{ h_{1,1}, (h_{0,i})_{1 \leq i \leq p}, h_1 \}$

**EVAL($m \in \mathcal{S}^p$):**

Parse $m := \langle m_1, \ldots, m_p \rangle$

if $m \in \mathcal{Q}_M$ then return RerandomizeTag($K(m)$)

$\mathcal{Q}_M := \mathcal{Q}_M \cup \{ m \}$

for $i \in \{1, \ldots, p\}$ do

\[ t := \lfloor (j+1)/\alpha \rfloor \]

for $i \in \{1, \ldots, p\}$ do

\[ t_i := (\langle \mathbf{B} \mathbf{B}_0 \rangle s_i) \]

else

\[ t_i := (\langle \mathbf{B} \mathbf{B}_1 \rangle s_i) \]

\[ u_i := u_i + RF_{i,\alpha} \left( \left[ \mathbf{m}^*_j \right]_{i+1} \right) (\mathbf{B}^\perp)^\top t_i, i \leq \alpha \]

for $i \in \{1, \ldots, p\}$ do

\[ u_i := u_i + RF_{i,\alpha} \left( \left[ \mathbf{m}^*_j \right]_{i+1} \right) (\mathbf{B}^\perp)^\top t_i, i \leq \alpha \]

\[ u := \sum_{i=1}^{p} u_i + h_0 \]

$K(m) := \left( \sum_{i=0}^{n-1} [t_i] \right) \mathbb{Z}^k_q \cup \mathcal{Q}_M = \emptyset \wedge \beta$

return $\text{Prefix}(m^* \cap \mathcal{Q}_M = \emptyset) \wedge \beta$
where $\text{RF}_{i,j}^{(0)} : \{0,1\}^{j+1} \rightarrow \mathbb{Z}_q^{1 \times k}$ is another independent random function. Since $\text{RF}_{i,j}^{(0)}$ in not used in game $G_{3, j, 2}$, $\text{RF}_{i,j}^{(1)}$ is a random function.

Eval queries with $[m]_{j+1} = 0$ are distributed identically in both games, by definition of $\text{RF}_{i,j+1}^{(0)}$.

Eval queries with $[m]_{j+1} = 1$ are distributed identically in both games, since for those queries (for all $i \in \{i, \ldots, p\}$) $t_i \in \text{span}(B_1 B_2)$ and both $B$ and $B_1$ are orthogonal to $B_0$ and thus $\text{RF}_{i,j+1}^{(0)}([m]_{j+1})(B_0)^\top t_i = 0$.

The Chal query is distributed identically if $[m^*]_{j+1} = 0$. For the case $[m^*]_{j+1} = 1$ note that for all $i \in \{i, \ldots, p\}$, $x_{i,j+1,1}$ is identically distributed as $x_{i,j+1,1} + B_0^* w_i$ for $w_i \stackrel{s} \in \mathbb{Z}_q^k$ and $w_i$ is hidden to the adversary since in all Eval queries (with $p \geq i$) there is either $[m]_{j+1} = 0$ which means that $x_{i,j+1,1}$ (for all $i \in \{i, \ldots, p\}$) is not used to compute the tag or there is $[m]_{j+1} = 1$ which means that $t_i \in \text{span}(B_1 B_2)$ and thus the $B_0^*$-part of $x_{i,j+1,1}$ cancels out. All in all this means that the value $h_{0,i}$ is the only one in the game that depends on $w_i$ and thus the $B_0^*$-part of $h_{0,i}$ is uniformly random to the adversary. Especially it is distributed identically in both games.

**Lemma 3.26** ($G_{3, j, 2} \leadsto G_{3, j, 3}$).

$$\Pr[G_{3, j, 2} \Rightarrow 1] = \Pr[G_{3, j, 3} \Rightarrow 1]$$

**Proof.** We define

$$\text{RF}_{i,j+1}^{(1)}([m]_{j+1}) := \begin{cases} \text{RF}_{i,j}^{(1)}([m]_{j}) + \text{RF}_{i,j}^{(1)}([m]_{j}) & \text{if } [m]_{j+1} = 0 \\ \text{RF}_{i,j}^{(1)}([m]_{j}) & \text{if } [m]_{j+1} = 1 \end{cases},$$

where $\text{RF}_{i,j}^{(1)} : \{0,1\}^{j+1} \rightarrow \mathbb{Z}_q^{1 \times k}$ is another independent random function. Since $\text{RF}_{i,j}^{(1)}$ in not used in game $G_{3, j, 3}$, $\text{RF}_{i,j+1}^{(1)}$ is a random function.

The argument, that the games $G_{3, j, 2}$ and $G_{3, j, 3}$ are identically distributed, is the same as in Lemma 3.25, just with the roles of 0 and 1 swapped.

**Lemma 3.27** ($G_{3, j, 3} \leadsto G_{3, j+1}$). For all adversaries $A$ there exists an adversary $B$ with

$$|\Pr[G_{3, j, 3}^A \Rightarrow 1] - \Pr[G_{3, j+1}^A \Rightarrow 1]| \leq 2 \left( \text{Adv}_{\text{ddh}}^{\text{mddh}}(B_m, G_{3, 3}) + \frac{1}{q^3} \right),$$

and $T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)$.

**Proof.** In game $G_{3, j, 3}$, replace all occurrences of the term $\text{RF}_{i,j+1}^{(0)}([m]_{j+1})(B_0^*)^\top + \text{RF}_{i,j+1}^{(1)}([m]_{j+1})(B_1^*)^\top$ with $\text{RF}_{i,j}([m]_{j})(B_1^*)^\top$ to avoid computing $B_0$ and $B_1$. This does not change the distribution, since $B_0, B_1$ is a basis for $\text{span}(B_1^*)$. The remaining transition is the reverse of Lemma 3.24.

**Lemma 3.28** ($G_{3, j, 3} \leadsto G_4$).

$$|\Pr[G_{3, j, 3}^A \Rightarrow 1] - \Pr[G_{4}^A \Rightarrow 1]| \leq \frac{Q}{q^{2k}}$$

**Proof.** The challenge query evaluates $\text{RF}_{i,\alpha}$ only for the input value $m^*_i$. Assume Request($m^*_i$) $\cap Q_M = \emptyset$, otherwise the adversary has lost the game anyway. In each user secret key the value $\text{RF}_{p, \text{pub}}(m)(B_1^*)^\top t_p$ is part of $u$. This is the only place where $\text{RF}_{p, \text{pub}}(m)$ is used, since only the first Eval query for each evaluates the random function. Each query outputs a uniformly random value for $u$ when $t_p \not\in \text{span}(B)$, which happens with probability $\geq 1 - 1/q^{2k}$. In this case $h_1$ is the only value depending on $x_0^t$ and thus uniform random as well.

**Summary.** To prove Theorem 3.19, we combine Lemmas 3.20–3.28 to change $h_1$ from real to random and then apply all Lemmas in reverse order to get to the HPR0-CMA random game.
exist adversaries queries of and $T$ is collision resistant and the

Theorem 4.1 (Security of the HIBKEM transformation).

Again we replace in the equation for IND

public-key encryption in [20].

an HIBE scheme with a (one-time secure) symmetric cipher by adapting a similar transformation for

mechanism (HIBKEM) under the

4 Transformation to HIBE

The term $\text{MAC} \leftarrow (B, (x_{i,j,k})_{1 \leq i \leq L, 1 \leq j \leq \gamma, b \in \{0,1\}}, x'_0, k)$ is collision resistant hash function.

Optimization. MAC$_2[U_{\mathcal{A}, k}]$ can be improved, like MAC$_1[U_{\mathcal{A}, k}]$ with a collision-resistant hash function. Again we replace in the equation for $u$ the term $\sum_j f_j(m_i) x_j^\top$ with $\sum_j f_j(H(m_i)) x_j^\top$, where $H$ is a collision resistant hash function.

Formally, we need a family of hash functions $\mathcal{H} = (\text{Gen}, \text{Eval})$ with domain $S^L$ and range $\{0,1\}^\gamma$ for all hash keys. The affine MAC MAC$_2^H[U_{\mathcal{A}, k}]$ is shown in Figure 11.

Figure 11: Our second affine MAC improved with a hash function.

Theorem 3.29 (Security of MAC$_2^H[U_{\mathcal{A}, k}]$). MAC$_2^H[U_{\mathcal{A}, k}]$ is tightly HPR$_0$-CMA secure in $G_2$ when $\mathcal{H}$ is collision resistant and the $U_{\mathcal{A}}$-MDDH assumption holds for $G_2$. Precisely, for all adversaries $A$ there exist adversaries $B$ and $C$ with

\[
\text{Adv}_{\text{MAC}_2^H[U_{\mathcal{A}, k}, G_2]}^{\text{HPR}_0, \text{CMA}}(A) \leq (2 + 8\gamma) \left( \text{Adv}_{\text{MAC}_2^H[U_{\mathcal{A}, k}, G_2]}^{\text{MDDH}}(B) + \frac{1}{q - 1} \right) + \text{Adv}^\gamma_{\mathcal{H}}(C) + \frac{LQ}{q2^k}
\]

and $T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)$ and $T(C) \approx T(A) + Q \cdot \text{poly}(\lambda)$, where $Q$ denotes the number of Eval queries of $A$ and $\text{poly}$ is a polynomial independent of $A$.

We omit the proof of this theorem because it is very similar to the proof of Theorem 3.3, just with the changes we already mentioned for theorem Theorem 3.18.

4 Transformation to HIBE

Any affine MAC with levels can be transformed tightly to a hierarchical identity-based key encapsulation mechanism (HIBKEM) under the $D_k$-MDDH assumption. The transformation is shown in Figure 12. It is a generalization of the transformation from delegatable, affine MACs to HIBKEMs in [3]. We only consider HIBKEM here and one can easily prove that every HIBKEM can be transformed (tightly) into an HIBE scheme with a (one-time secure) symmetric cipher by adapting a similar transformation for public-key encryption in [20].

Theorem 4.1 (Security of the HIBKEM transformation). The HIBKEM HIBKEM[MAC, $D_k$] is IND-HID-CPA secure in $G$ under the $D_k$-MDDH assumption for $G_1$ if MAC is HPR$_0$-CMA secure in $G_2$. Precisely, for all adversaries $A$ there exists adversaries $B_1$ and $B_2$ with

\[
\text{Adv}_{\text{HIBKEM}[MAC, D_k], G}^{\text{HID}-\text{CPA}}(A) \leq \text{Adv}_{\text{MAC}, G_2}^{\text{HPR}_0, \text{CMA}}(B_1) + \text{Adv}_{\text{MAC}, G_2}^{\text{MDDH}}(B_2)
\]
\[
\text{Gen}(G_1, G_2, G_T, q, P_1, P_2, \varepsilon):
\]
\[
\text{skMAC} := (B, (x_{i,j})_{1 \leq i \leq \ell(L), 1 \leq j \leq \ell^p(l, i)} : x_0^T) \\
A := D_k
\]
\[
\text{for } l \in \{1, \ldots, \ell(L)\}, i \in \{1, \ldots, L\}, j \in \{1, \ldots, \ell^p(l, i)\} \text{ do}
\]
\[
\begin{align*}
Y_{i,j,l} &:= (Y_{i,j,l}^T | x_{i,j,l}) \cdot A \\
d_{i,j,l} &:= x_{i,j,l} \cdot B; E_{i,j,l} := Y_{i,j,l} \cdot B \\
y_0^* &:= x_0^T \cdot y_0^* \cdot A \\
pk &:= \left( G, [A]_1, \left([Z_{l,i,j}]_{1 \leq i \leq \ell(L), 1 \leq j \leq \ell^p(l, i)} \right) \right) \\
dk &:= \left( [B]_2, \left([d_{l,i,j}]_2, [E_{l,i,j}]_2 \right)_{1 \leq i \leq \ell(L), 1 \leq j \leq \ell^p(l, i)} \right) \\
sk &:= \left( \text{skMAC}, (Y_{i,j,l})_{1 \leq i \leq \ell(L), 1 \leq j \leq \ell^p(l, i)}, y_0^* \right)
\end{align*}
\]
\[
\text{return } (pk, dk, sk)
\]
\[
\text{Enc}(pk, id \in \mathcal{S}^n): \\
r \leftarrow \mathcal{Z}_q; c_0 := Ar \\
\text{for } l \in \{1, \ldots, \ell(p)\} \text{ do}
\]
\[
\begin{align*}
c_{l,i} &:= \sum_{j=1}^{\ell^p(l, i)} f_{i,j}(m_{i,j}) \cdot Z_{l,i,j} \cdot r \\
C &:= \left(c_{0,1}, \left(c_{l,i,1} \right)_{1 \leq i \leq \ell(p)} \right) \\
K &:= z_0^* \cdot r \\
\text{return } ([K], C)
\end{align*}
\]
\[
\text{Dec}(usk[id], id \in \mathcal{S}^n, C): \\
\text{usk}[id] := \left( \left([t_{i,j}]_{1 \leq i \leq \ell(p)}, [u_{i,j}], [v_{i,j}] \right) \right) \\
\text{Parse } C := \left( c_{0,1}, \left(c_{l,i,1} \right)_{1 \leq i \leq \ell(p)} \right) \\
[K]_T := r \cdot \left( [c_{0,1}], \left[ \begin{array}{c} v \\ u \end{array} \right] \right) - \sum_{l=1}^{\ell(p)} \sum_{i=1}^{\ell(L)} \sum_{j=1}^{\ell^p(l, i)} c_{l,i,j} - [t_{i,j}]_2 \\
\text{return } [K]_T
\]

\[
\text{Ext}(sk, id \in \mathcal{S}^n): \\
\left( \left([t_{i,j}]_{1 \leq i \leq \ell(p)}, [u_{i,j}], [v_{i,j}] \right) \right) \leftarrow \text{Tag}(skMAC, id) \\
v := \left( \sum_{l=1}^{\ell(p)} \sum_{i=1}^{\ell^p(l, i)} f_{i,j}(m_{i,j}) \cdot Y_{i,j,l} \right) \cdot t_i + y_0' \\
\text{for } l \in \{1, \ldots, \ell(p)\}, i \in \{p+1, \ldots, L\}, j \in \{1, \ldots, \ell^p(l, i)\} \text{ do}
\]
\[
\begin{align*}
d_{i,j,l} &:= x_{i,j,l} \cdot t_i \\
\end{align*}
\]
\[
\text{return } \left( \text{usk}', \text{udk}' \right)
\]

Figure 12: Transformation of an affine MAC with levels to a HIBKEM.
and \( T(B_1) \approx T(A) + Q \cdot \text{poly}(\lambda) \) and \( T(B_2) \approx T(A) + Q \cdot \text{poly}(\lambda) \), where \( Q \) denotes the number of Ext queries of \( A \) and poly is a polynomial independent of \( A \).

The detailed proof of Theorem 4.1 can be found in Appendix B.

### 4.1 Instantiations

#### 4.1.1 MDDH

The result of applying the HIBKEM transformation to \( \text{MAC}_1[\mathcal{U}_{4k,k}] \) is shown in Figure 13. The scheme has \( \alpha(L^2 + L)(4k^2 + k) + 3k^2 + 2k \) group elements in the public key and \( 4k + 1 \) group elements in the ciphertext. The user secret keys have at most \( \alpha(L^2/2 + L/2 - 1)(k + 1) + 4k + 1 \) group elements. Identities that are deeper in the hierarchy have smaller secret keys, since the user secret key size is dominated by the size of the delegation keys. On the last level, the user secret keys consist of only \( 4k + 1 \) keys.

The result of applying the HIBKEM transformation to \( \text{MAC}_2[\mathcal{U}_{4k,k}] \) is shown in Figure 14. The scheme has \( \alpha(L^2 + L)(4k^2 + k) + 3k^2 + 2k \) group elements in the public key and \( 3Lk + k + 1 \) group elements in the ciphertext. The user secret keys have at most \( 3Lk + k + 1 \) group elements. Identities that are deeper in the hierarchy have larger secret keys.

The schemes have both the same public key. The first scheme has smaller ciphertexts, while the second has a more efficient reduction and smaller user secret keys in the worst case.

#### 4.1.2 SXDH

With a type III pairing, both of our schemes can be instantiated with the SXDH assumption.
Figure 14: The resulting scheme HIBKEM[MAC2[U_{ik,k}], D_k].

The result (HIBKEM_1) of instantiating scheme HIBKEM[MAC_1[U_{i,1}], U_{2,1}] with the SXDH assumption is shown in Figure 15. The scheme has \(\alpha(L^2 + L)5 + 5\) group elements in the public key and 5 group elements in the ciphertext. The user secret keys have at most \(\alpha(L^2 + L - 2) + 5\) group elements.

The result (HIBKEM_2) of instantiating scheme HIBKEM[MAC2[U_{i,1}], U_{2,1}] with the SXDH assumption is shown in Figure 16. The scheme has \(\alpha(L^2 + L)5 + 5\) group elements in the public key and \(3L + 2\) group elements in the ciphertext. The user secret keys have at most \(3L + 2\) group elements.

References


Gen(1^λ):
B := U_{3,1}; A := U_{2,1}
for i ∈ {1, ..., L}, j ∈ {1, ..., iα}, b ∈ {0, 1} do
    x_{i,j,b} := Z_q^3 \ Y_{i,j,b} := Z_q^{1 \times 3}
Z_{i,j,b} := (Y_{i,j,b} | x_{i,j,b}) \cdot A
    d_{i,j,b} := x_{i,j,b} \cdot B; E_{i,j,b} := Y_{i,j,b} \cdot B
x_0 := Z_q; y_0 := Z_q; z_0 := (y_0^T | x_0^T) \cdot A
pk := (G, [A]_1, ([Z_{i,j,b}]_1)_{1 \leq i \leq L, 1 \leq j \leq iα, b \in \{0, 1\}}
    \cdot [z_0^T]_1)
dk := ([B]_2, ([d_{i,j,b}]_2, [E_{i,j,b}]_2)_{1 \leq i \leq L, 1 \leq j \leq iα, b \in \{0, 1\}}
    \cdot [y_0^T, z_0^T]_2)
sk := (sk_{MAC}, ([Y_{i,j,b}]_1)_{1 \leq i \leq L, 1 \leq j \leq iα, b \in \{0, 1\}}, [y_0^T]_2)
return (pk, dk, sk)

Ext(sk, id ∈ S^n):
s := Z_q; t := Bs
u := \left(\sum_{i=1}^{p} \sum_{j=1}^{iα} \chi_{i,j,[u]} \cdot t \right) + x_0
v := \left(\sum_{i=1}^{p} \sum_{j=1}^{iα} \chi_{i,j,[v]} \cdot t \right) + y_0
for i ∈ {p + 1, ..., L}, j ∈ {1, ..., iα}, b ∈ {0, 1} do
    x_{i,j,b} := x_{i,j,b} \cdot t; e_{i,j,b} := Y_{i,j,b} \cdot t
us := ([t]_2, [u]_2, [v]_2)
ud := ([d_{i,j,b}]_2, [e_{i,j,b}]_2)_{p+1 \leq i \leq L, 1 \leq j \leq iα}
return (us, ud)

Del(dk, usk, udk, id ∈ S^n, id_{p+1} ∈ S):
Parse usk := ([t]_2, [u]_2, [v]_2)
s := Z_q; t := t + Bs
u := u + \left(\sum_{i=1}^{p} \sum_{j=1}^{iα} d_{i,j,[u]} \cdot t \right)
v := v + \left(\sum_{i=1}^{p} \sum_{j=1}^{iα} E_{i,j,[v]} \cdot u \right)
for i ∈ {p + 1, ..., L}, j ∈ {1, ..., iα}, b ∈ {0, 1} do
    d_{i,j,b} := d_{i,j,b} + d_{i,j,[b]}
    e_{i,j,b} := e_{i,j,b} + E_{i,j,[b]}
us := ([t']_2, [u']_2, [v']_2)
ud := ([d_{i,j,b}']_2, [e_{i,j,b}']_2)_{p+2 \leq i \leq L, 1 \leq j \leq iα}
return (usk', udk')

Enc(pk, id ∈ S^n):
r := Z_q; c_0 := Ar
c_1 := \sum_{i=1}^{p+1} \sum_{j=1}^{iα} Z_{l,j,[id]} \cdot r
C := ([c_0]_1, [c_1]_1)
K := z_0 \cdot r
return ([K]_T, C)

Dec(usk, id ∈ S^n, C):
Parse usk := ([t]_2, [u]_2, [v]_2)
Parse C := ([c_0]_1, [c_1]_1)
[K]_T := [c_0]_1 \circ \left[\sum_{i=1}^{p+1} \sum_{j=1}^{iα} C_{l,j,[id]} \cdot t \right]
return [K]_T

Figure 15: The resulting scheme HIBKEM_1 := HIBKEM[MAC_1, U_{3,1}, U_{2,1}].
Figure 16: The resulting scheme $\text{HIBKEM}_2 := \text{HIBKEM}[\text{MAC}_2[U_{d,1}], U_{d,1}]$.


A Semi-adaptive Security of the BKP MAC

Blazy, Kiltz and Pan proposed in [3] an affine MAC with a tight security reduction in the non-hierarchical setting. This MAC can be generalized to the hierarchical setting in the semi-adaptive security model, as shown in Figure 17. In the semi-adaptive security model, the adversary has to send the challenge identity setting. This MAC can be generalized to the hierarchical setting in the semi-adaptive security model, as Blazy, Kiltz and Pan proposed in [3] an affine MAC with a tight security reduction in the non-hierarchical setting.

Definition A.1 (IND-sahID-CPA security). A hierarchical identity-based key encapsulation scheme HIBKEM is IND-sahID-CPA-secure if for all PPT A,

$$\text{Adv}^{\text{IND-sahID-CPA}}_{\text{HIBKEM}}(A) := \text{Pr}[\text{IND-sahID-CPA}^A \text{real} \Rightarrow 1] - \text{Pr}[\text{IND-sahID-CPA}^A \text{rand}]$$

is negligible.

For MACs we define saHPR0-CMA security similar to HPR0-CMA security, just with the difference that the adversary has to send the challenge message before any Tag queries. Such a MAC can be transformed to a IND-sahID-CPA secure HIBE in the same way as in the full security setting.

The main difference to the original, non-hierarchical MAC is that the randomness in the Tag algorithm $s$ is random and not pseudo-random. Thus the tags are not deterministic and we need to take care of duplicated tag queries for a single message. We do this in our proof by storing for each message $m$ the
Theorem A.2 (Security of MAC\textsubscript{NR}[D\textsubscript{k}]. MAC\textsubscript{NR}[D\textsubscript{k}] is tightly saHPR\textsubscript{0}-CMA secure in G\textsubscript{2} under the D\textsubscript{k}-MDDH assumption for G\textsubscript{2}. Precisely, for all adversaries A there exists an adversary B with

$$\text{Adv}_{\text{MAC}_{\text{NR}}[D\textsubscript{k}],G\textsubscript{2}}^{\text{sa-hpr0-cma}}(A) \leq 2\alpha L \text{Adv}_{D\textsubscript{k},G\text{Gen},G\textsubscript{2}}^{\text{mddh}}(B)$$

and $T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)$.

Proof. The proof uses a hybrid argument with hybrids $G_0, G_1, G_2, i$ for $i \in \{0, \ldots, \alpha L\}$ and $G_3$. $G_0$ is the saPR-CMA real game, the other games are defined in Figure 19. They make use of a random functions $R\text{F}_i : \{0, 1\} \rightarrow Z_{q^k}$, for $i \in \{0, \ldots, \alpha L\}$, defined on-the-fly.

Lemma A.3 ($G_0 \sim G_1$).

$$\text{Pr}[G_0 \Rightarrow 1] = \text{Pr}[G_1 \Rightarrow 1]$$

Figure 19: Hybrids for the security proof of MAC\textsubscript{NR}[D\textsubscript{k}]. The algorithm Delegate is only helper function and not an oracle for the adversary.

first tag we computed and return a randomized version of this tag for each follow-up Tag query for m. This randomization only requires knowledge of the public key.

We call messages $(m_1, \ldots, m_p)$ critical, if $(m_1, \ldots, m_{p-1})$ is a prefix of the challenge message $m^*$. For each message m the critical prefix is the prefix of $m$, that is a critical message. The critical prefix exists for all messages, that are not a prefix of the challenge message $m^*$. Our overall proof strategy is to randomize $u$ in all Tag oracle queries for critical messages and for Tag oracle queries of non-critical messages we simulate a Tag oracle query the critical prefix and use the key delegation mechanism to obtain a tag for the actual message.

Theorem A.2 (Security of MAC\textsubscript{NR}[D\textsubscript{k}]). MAC\textsubscript{NR}[D\textsubscript{k}] is tightly saHPR\textsubscript{0}-CMA secure in G\textsubscript{2} under the D\textsubscript{k}-MDDH assumption for G\textsubscript{2}. Precisely, for all adversaries A there exists an adversary B with

$$\text{Adv}_{\text{MAC}_{\text{NR}}[D\textsubscript{k}],G\textsubscript{2}}^{\text{sa-hpr0-cma}}(A) \leq 2\alpha L \text{Adv}_{D\textsubscript{k},G\text{Gen},G\textsubscript{2}}^{\text{mddh}}(B)$$

and $T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)$.

Proof. The proof uses a hybrid argument with hybrids $G_0, G_1, G_2, i$ for $i \in \{0, \ldots, \alpha L\}$ and $G_3$. $G_0$ is the saPR-CMA real game, the other games are defined in Figure 19. They make use of a random functions $R\text{F}_i : \{0, 1\} \rightarrow Z_{q^k}$, for $i \in \{0, \ldots, \alpha L\}$, defined on-the-fly.

Lemma A.3 ($G_0 \sim G_1$).

$$\text{Pr}[G_0 \Rightarrow 1] = \text{Pr}[G_1 \Rightarrow 1]$$

32
Proof. In game $G_1$ the queried message tags are not computed directly, instead we compute a tag for the prefix $m'$ of $m$, that is one component longer than the longest common prefix of $m$ and $m^*$. This prefix exists when $m$ is not a prefix of $m^*$. The tag for $m'$ is then delegated to a tag for $m$ and re-randomized. This requires only the public key. The distribution of a delegated and re-randomized tag is the same as the distribution of a fresh tag, thus the games are identical. □

Lemma A.4 ($G_1 \leadsto G_{2,0}$).

$$\Pr[G_1^A \Rightarrow 1] = \Pr[G_{2,0}^A \Rightarrow 1]$$

Proof. In game $G_{2,0}^A$ we replace $x_i^A$ with $RF_0(c)$, which is equivalent. □

Lemma A.5 ($G_{2,i} \leadsto G_{2,i+1}$). For all adversaries $A$ there exists an adversary $B$ with

$$|\Pr[G_{2,i}^A \Rightarrow 1] - \Pr[G_{2,i+1}^A \Rightarrow 1]| \leq \text{Adv}_D^{\text{DDH}}_{D_k,G\text{Gen},G_2}(B)$$

and $T(B) \approx T(A) + Q \cdot \text{poly}(\lambda)$.

Proof. We define

$$RF_{i+1}(\lfloor m \rfloor) := \begin{cases} RF_i(\lfloor m \rfloor) & \text{if } \lfloor m \rfloor = \lfloor m^* \rfloor + 1, \\ RF_i(\lfloor m \rfloor) + RF'_i(\lfloor m \rfloor) & \text{if } \lfloor m \rfloor = 1 - \lfloor m^* \rfloor + 1, \\ \end{cases}$$

where $RF'_i : \{0,1\}^* \rightarrow \mathbb{Z}_q$ is another independent random function. Note that whenever $RF_{i+1}$ is evaluated, $RF_i$ is not evaluated. This is because the adversary is not allowed to query the tags for any prefix of $m'$ since each such message would be a prefix of the challenge message $m^*$.

The adversary $B$ uses the random self reducibility to get a $Q$-fold $D_k$-$\text{DDH}$ challenge $(\lfloor A^2 \rfloor, \lfloor H \rfloor)$, where $Q$ denotes the number of tag queries. The reduction uses a function $\phi : \{0,1\}^* \rightarrow \{1, \ldots, Q\}$ that has to be injective on all prefixes of queried messages. It can be computed on-the-fly. The reduction is given in Figure 20.

The reduction computes the public key honestly except for $x_{i^*,j^*,1-(m^*_{i^*})^*}$. The public key is distributed correctly since $r^T A$ is a uniform random $k$ dimensional row vector, just like $x^T_{i^*,j^*,1-(m^*_{i^*})^*} B$ would be for a uniform random $x^T_{i^*,j^*,1-(m^*_{i^*})^*}$.

CHAL, FINALIZE and EVAL queries with $K(m') \neq \perp$ are the same as in $G_{2,i}$ and $G_{2,i+1}$. The EVAL queries with $p' < i^* \lor \lfloor m' \rfloor \neq \lfloor m^* \rfloor$ remain also unchanged. Note that for these queries, the tag delegation keys $[d_{i,j},k]_{q^2}$ can be computed from $\lfloor x_{i,j},k \rfloor^2$ and $s$.

For the EVAL queries with $p' \geq i^* \land \lfloor m' \rfloor \neq \lfloor m^* \rfloor$, write $H_c := A W_c + R_c$ for some $W_c \in \mathbb{Z}_q$ and $R_c = 0$. If $H_c$ was drawn from $D_k$ or uniform random $R_c \in \mathbb{Z}_q^{k+1}$ if $H_c$ was chosen uniformly random. Then

$$u = \sum_{(i,j)=(1,1)}^{(p',\alpha)} x^T_{i,j,(m')_{j}} t + r^T A(s + W_c) + r^T R_c + RF_i(\lfloor m' \rfloor)$$

$$= \sum_{(i,j)=(1,1)}^{(p',\alpha)} x^T_{i,j,(m')_{j}} t + x^T_{i^*,j^*,1-(m^*_{i^*})} B (s + W_c) + r^T R_c + RF_i(\lfloor m' \rfloor)$$

$$= \sum_{(i,j)=(1,1)}^{(p',\alpha)} x^T_{i,j,(m')_{j}} t + r^T R_c + RF_i(\lfloor m' \rfloor).$$

If $R_c = 0$ the reduction is simulating $G_{2,i}$, if $R_c$ is uniform random, define $RF'_i(\lfloor m' \rfloor) := r^T R_c$ and the reduction is simulating $G_{2,i}$. Note that for these queries, the tag delegation keys $[d_{i,j},k]_{q^2}$ can be computed from $x_{i,j,k}$ and $\lfloor t \rfloor$ since $x_{i,j,k}$ is known to the adversary for all $i > p' \geq i^*$.

□
Figure 20: Adversary $B$ for the proof of Lemma A.5. The helper algorithm Delegate is given in Figure 19.

Lemma A.6 ($G_{2, La} \leadsto G_3$).

\[ \Pr[G_{2, La} \Rightarrow 1] = \Pr[G_3 \Rightarrow 1] \]

Proof. In game $G_{2, La}$ all the $u$s in tags for a message prefix $m'$ are masked by $RF_{La}$ evaluated at the unique value $\llbracket m' \rrbracket$. So they are uniformly random as in $G_3$. \hfill \square

Summary. To prove Theorem A.2, we combine Lemmas A.3–A.6 to change $h_1$ from real to random and then apply all Lemmas in reverse order to get to the saHP$\sigma$-CMA$_{rand}$ game. \hfill \square
B Security of the HIBKEM transformation

Proof (of Theorem 4.1). The proof makes use of the hybrids $G_0 - G_4$ defined in Figure 21. $G_0$ is the IND-HID-CPA\textsubscript{real} game.

Lemma B.1 ($G_0 \leadsto G_1$).
\[ \Pr[G_0^A \Rightarrow 1] = \Pr[G_4^A \Rightarrow 1] \]

Proof. The only difference between these games is that $c_{i,j}^*$ is computed with the public value $z_{l,i,j}$ in game $G_0$ and with the secret key $x_{l,i,j}$ and $y_{l,i,j}$ in $G_1$.

Lemma B.2 ($G_1 \leadsto G_2$). For all adversaries $A$ there exists an adversary $B_2$ with
\[ |\Pr[G_1^A \Rightarrow 1] - \Pr[G_2^A \Rightarrow 1]| \leq \text{Adv}^\text{mdh}_{D_k,G\text{Gen},G_1}(B_2) \]
and $T(B_2) \approx T(A) + Q \cdot \text{poly}(\lambda)$.

Proof. The only difference between these is that $c_{i,j}^*$ is chosen from $\text{Span}(A)$ in $G_1$ and from $\mathbb{Z}_q^{k+1}$ in $G_2$.

The running time of $B_2$ is dominated by the running time of $A$ plus some (polynomial) overhead that is independent of $T(A)$ for the group operations in each oracle query.

Lemma B.3 ($G_2 \leadsto G_3$).
\[ \Pr[G_2^A \Rightarrow 1] = \Pr[G_3^A \Rightarrow 1] \]

Proof. These two games are equivalent. First notice that the values $z_{l,i,j}$ and $y'_0$ are uniform random when $y_{l,i,j}$ and $z'_0$ are hidden, so $Z_{l,i,j}$ and $z_0'$ are distributed identical in both games. Second notice
\[ Z_{l,i,j} := (y_{l,i,j}^T | x_{l,i,j}) \cdot A \iff y_{l,i,j}^T = (z_{l,i,j} - x_{l,i,j} A) A^{-1} \]
and similarly
\[ z'_0 := (y'_0^T | x'_0) \cdot A \iff y'_0^T = (z'_0 - x'_0 A) A^{-1}. \]

Game $G_3$ is obtained from $G_2$ by choosing $Z_{l,i,j}$ and $z'_0$ uniform random and replacing all occurrences of the values $y_{l,i,j}$ and $y'_0$ with the above equation. Thus the games are equally distributed.

Lemma B.4 ($G_3 \leadsto G_4$). For all adversaries $A$ there exists an adversary $B_1$ with
\[ |\Pr[G_3^A \Rightarrow 1] - \Pr[G_4^A \Rightarrow 1]| \leq \text{Adv}^\text{hpr\textsubscript{cma}}_{\text{MAC},c_2}(B_1) \]
and $T(B_1) \approx T(A) + Q \cdot \text{poly}(\lambda)$.

Proof. The adversary $B$ is given in Figure 22. When $B$ plays the PR-CMA\textsubscript{real} game with the affine MAC with levels challenger, he simulates the game $G_4$ for $A$. On the other hand, when $B$ plays the PR-CMA\textsubscript{rand} game with the MAC challenger, he simulates the game $G_4$ for $A$.

The running time of $B_1$ is dominated by the running time of $A$ plus some (polynomial) overhead that is independent of $T(A)$ for the group operations in each oracle query.

Summary. To prove Theorem 4.1, we combine Lemmas B.1–B.4 to change the key $K$ from real to random and then apply all Lemmas in reverse order to get to the IND-HID-CPA\textsubscript{rand} game.
$G_0 \xrightarrow{1} G_1 \xrightarrow{1} G_3 \xrightarrow{1} G_4$

**Init:**

$sk_{MAC} \leftarrow Gen_{MAC}(G_2, q, P_2)$

$sk_{MAC} = (B, (x_{l,i,j})_{1 \leq i \leq L, 1 \leq j \leq \ell(l), 1 \leq \ell \leq \ell'} , x_0^\prime)$

$A \leftarrow D_0$

For $l \in \{1, \ldots, \ell(L), i \in \{1, \ldots, L\}, j \in \{1, \ldots, \ell(l, i)\}$ do

$Y_{l,i,j} \leftarrow Z_q^{\ell(l, i)}$, $Z_{l,i,j} := (Y_{l,i,j}^T \mid x_{l,i,j}) \cdot A$

$d_{l,i,j} := x_{l,i,j} \cdot B; E_{l,i,j} := Y_{l,i,j} \cdot B$

$\text{Tag} := \overrightarrow{\left(Z_{l,i,j} \cdot B - A^T d_{l,i,j}\right)}$

$y_0^\prime \leftarrow Z_q^0$

$z_0^\prime := (y_0^\prime \mid x_0^\prime) \cdot A$

$pk := (G, [A]_z, ([Z_{l,i,j}]_{1 \leq i \leq L, 1 \leq j \leq \ell(l)}, [Z_0])_{1 \leq i \leq \ell(l)}, [Z_{l,i,j}])_{1 \leq i \leq \ell(l)}$

$dk := (B_{l,i,j}, [d_{l,i,j}]_z, [E_{l,i,j}]_z)_{1 \leq i \leq L, 1 \leq j \leq \ell(l)}$

return $(pk, dk)$

**Ext(id $\in S^p$):**

$\mathcal{Q}_{ID} \leftarrow \mathcal{Q}_{ID} \cup \{id\}$

$(\{t_{l,i,j}\}_{1 \leq i \leq \ell(l), 1 \leq l \leq \ell'}) \leftarrow \text{Tag}(sk_{MAC}, id)$

$v := \sum_{l=1}^{\ell(p)} \sum_{i=1}^{\ell(l)} \sum_{j=1}^{\ell(l)} f_{l,i,j}(m_i) Y_{l,i,j} \cdot t_{l,i,j}$

$v^T := \sum_{l=1}^{\ell(p)} t_{l}^T \sum_{i=1}^{\ell(l)} \sum_{j=1}^{\ell(l)} f_{l,i,j}(m_i) Z_{l,i,j}^T + z_0^\prime - u \cdot A^T$

For $l \in \{1, \ldots, \ell(p), i \in \{p + 1, \ldots, L\}, j \in \{1, \ldots, \ell(l, i)\}$ do

$d_{l,i,j} := x_{l,i,j} \cdot t_{l,i,j}; e_{l,i,j} := Y_{l,i,j} \cdot t_{l,i,j}$

$e_{l,i,j} := (t_l \mid Z_{l,i,j} + z_0^\prime - d_{l,i,j} A) / A^T$

$udk(id) := ([d_{l,i,j}]_z, [e_{l,i,j}]_z)_{1 \leq i \leq \ell(p), 1 \leq L, 1 \leq \ell(l, i)}$

return $(\{t_{l,i,j}\}_{1 \leq i \leq \ell(p)} \mid [u]_2, [v]_2, udk(id))$

**Exc(id $\in S^p$):**

$r \leftarrow Z_q; c_0 := A \cdot r$

$c_0^\prime := Z_q^0$

$h := Z_q$

$c_0^\prime \leftarrow Z_q^0; c_0^\prime := h + A \cdot A^{-1} c_0^\prime$

For $l \in \{1, \ldots, \ell(p)\}$ do

$c_{l,i} := \sum_{i=1}^{\ell(p)} \sum_{j=1}^{\ell(l)} f_{l,i,j}(m_i) Z_{l,i,j}^T \cdot c_0^\prime$

$c_{l,i} := \sum_{i=1}^{\ell(p)} \sum_{j=1}^{\ell(l)} f_{l,i,j}(m_i) (Z_{l,i,j}^T \cdot c_0^\prime + x_{l,i,j} \cdot h)$

$K^* := z_0^\prime \cdot r$

$\overrightarrow{K^*} := (y_0^\prime \mid x_0^\prime) \cdot c_0^\prime$

$\overrightarrow{K^*} := z_0^\prime \cdot A \cdot c_0^\prime + z_0^\prime \cdot h$

$\overrightarrow{K^*} := z_0^\prime$

$C_i := (\{c_{l,i}\}_{1 \leq i \leq \ell(p)})$

return $(\overrightarrow{K^*}, [c_0^\prime], C_i)$

**Finalize($\beta \in \{0, 1\}$):**

return $\beta \land \text{Prefix(id$^\ast$) } \cap \mathcal{Q}_{ID} = \emptyset$

---

Figure 21: Hybrids for the security proof of the HIBKEM transformation.
\textbf{INIT:}\n
\begin{align*}
\text{pk}_\text{MAC} & \leftarrow \text{INIT}_\text{MAC} \\
\text{Parse } \text{pk}_\text{MAC} & =: \left( [B]_2, ([d]_{i,j})_{1 \leq i \leq \ell(l), 1 \leq j \leq L} \right) \\
A & \leftarrow \mathcal{D}_h \\
\text{for } l \in \{1, \ldots, \ell(L)\}, i \in \{1, \ldots, L\}, j \in \{1, \ldots, \ell'(l, i)\} & \text{ do} \\
Z_{l,i,j} & \leftarrow \mathbb{Z}^{k \times k}_q \\
E_{l,i,j} & := A^T (Z_{l,i,j}^T B - A^T d_{l,i,j}) \\
z_0 & \leftarrow \mathbb{Z}^{1 \times k}_q \\
pk & := \left( \mathcal{G}, [A]_1, \left( [Z_{l,i,j}]_{1 \leq i \leq \ell(L), 1 \leq j \leq L}; [z_0]_1 \right) \right)_{1 \leq j \leq \ell'(l, i)} \\
dk & := \left( [B]_2, ([d]_{i,j})_{2}, [E_{l,i,j}]_{2} \right)_{1 \leq i \leq \ell(L), 1 \leq j \leq L} \\
\text{return } (pk, dk) \\
\text{ENC}(\text{id}^* \in \mathcal{S}^p): & \quad /\!/ \text{one query} \\
C & \leftarrow \text{CHAL}(\text{id}^*) \\
\text{Parse } C = & : \left( [h]_1, ([h_{0,i}]_1)_{1 \leq i \leq \ell(p)}; [h]_T \right) \\
c_0 & \leftarrow \mathbb{Z}^{k}_q; \quad c_{i,j}^* := h + A \cdot A^{-1} c_0 \\
\text{for } l \in \{1, \ldots, \ell(p)\} & \text{ do} \\
c_{l,i} := & \sum_{j=0}^{\ell'(l, i)} \sum_{j=0}^{\ell'(l, i)} (f_{l,i,j}(\text{id}_{i,j})) \left( Z_{l,i,j} A^{-1} c_0 + x_{l,i,j} h \right) \\
K^* & := Z_0 A^{-1} c_0 + h_T \\
\text{return } ([K^*]_T, [c_{i,j}]_1, ([c_{i,j}]_1)_{1 \leq i \leq \ell(p)})
\end{align*}

\textbf{EXT(id \in \mathcal{S}^p):}\n
\begin{align*}
\mathcal{Q}_{\mathcal{D}} & \leftarrow \mathcal{Q}_{\mathcal{D}} \cup \{\text{id}\} \\
\tau & \leftarrow \text{EVAL}(\text{id}) \\
\text{Parse } \tau = & : \left( ([t]_1)_{1 \leq i \leq \ell(p)}; [u]_2, tdk \right) \\
tdk := & \left( ([d]_{i,j})_2 \right)_{1 \leq i \leq \ell(p), 1 \leq j \leq \ell'(l, i)} \\
v_T := & \left( \sum_{i=1}^{\ell(p)} \left( t_{i,j}^* \sum_{j=1}^{\ell'(l, i)} f_{l,i,j}(\text{id}_{i,j}) \right) \right) \\
& + z_0 - u \cdot A \cdot A^{-1} \\
\text{for } l \in \{1, \ldots, \ell(p)\}, i \in \{p + 1, \ldots, L\}, j \in \{1, \ldots, \ell'(l, i)\} & \text{ do} \\
\vec{a}_{l,i,j} := & \left( t_{l,i,j}^* Z_{l,i,j} + z_0 - d_{l,i,j} \right) \cdot A^{-1} \\
udk := & \left( ([d]_{i,j})_2; [e_{l,i,j}]_2 \right)_{1 \leq l \leq \ell(p), 1 \leq i \leq L, 1 \leq j \leq \ell'(l, i)} \\
\text{return } \left( ([t]_1)_{1 \leq i \leq \ell(p)}; [u]_2, [v]_T, udk \right) \\
\text{FINALIZE(} \beta \in \{0, 1\} \text{):} \\
\text{return } \text{FINALIZE}_\text{MAC}(\beta)
\end{align*}

Figure 22: Adversary \( B \) for Lemma B.4