Des. Codes Cryptogr. (2018) 86:517–554
https://doi.org/10.1007/s10623-017-0339-3

Tightly CCA-secure identity-based encryption with ciphertext pseudorandomness

Shuai Han[1,3] · **Shengli Liu**[1,3,4] · **Baodong Qin**[2,3] · **Dawu Gu**[1]

Received: 10 August 2016 / Revised: 15 December 2016 / Accepted: 24 January 2017 /
Published online: 21 March 2017

Abstract Affine message authentication code (MAC) and delegatable affine MAC turn out to be useful tools for constructing identity-based encryption (IBE) and hierarchical IBE (HIBE), as shown in Blazy, Kiltz and Pan's (BKP) creative work in CRYPTO (2014). An important result obtained by BKP is IBE of tight PR-ID-CPA security, i.e., tight IND-ID-CPA security together with ciphertext pseudorandomness (PR). However, the problem of designing tightly PR-ID-CCA2 secure IBE remains open. We note that the CHK transformation does not preserve ciphertext pseudorandomness when converting IND-ID-CPA secure 2-level HIBE to IND-ID-CCA2 secure IBE. In this paper, we solve this problem with a new approach. We introduce a new concept called *De-randomized delegatable affine MAC* and define for it *weak APR-CMA security*. We construct such a MAC with a tight security reduction to the Matrix DDH assumption, which includes the k-Linear and DDH assumptions. We present a paradigm for constructing PR-ID-CCA2 secure IBE, which enjoys both ciphertext pseudorandomness and IND-ID-CCA2 security, from De-randomized delegatable affine MAC and Chameleon

Communicated by K. Matsuura.

✉ Shengli Liu
slliu@sjtu.edu.cn

Shuai Han
dalen17@sjtu.edu.cn

Baodong Qin
qinbaodong@swust.edu.cn

Dawu Gu
dwgu@sjtu.edu.cn

1 Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

2 School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China

3 State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China

4 Westone Cryptologic Research Center, Beijing 100070, China

hashing. The security reduction is tightness preserving. It provides another approach to IND-ID-CCA2 security besides the CHK transformation. By instantiating the paradigm with our specific De-randomized delegatable affine MAC, we obtain the first IBE of tight PR-ID-CCA2 security from the Matrix DDH assumption over pairing groups of prime order. Our IBE also serves as the first tightly IND-ID-CCA2 secure IBE with anonymous recipient (ANON-ID-CCA2) from the Matrix DDH assumption. Our IBE further implies the first tightly IND-ID-CCA2 secure extractable IBE based on the Matrix DDH assumption. The latter can be used to get IBE of simulation-based selective opening CCA2 (SIM-SO-CCA2) security (due to Lai et al. in EUROCRYPT, 2014). The tight security of our IBE leads to a tighter reduction of the SIM-SO-CCA2 security.

Keywords Tight security reduction · Identity-based encryption · Ciphertext pseudorandomness · CCA2 security · Affine message authentication code

Mathematics Subject Classification 68P25 · 94A60

1 Introduction

Identity-based encryption (IBE) is a public-key encryption that enables one to encrypt a message using a recipient's identity, rather than its public key. It simplifies public key and certificate distribution and management. The concept of IBE was first proposed by Shamir in [30]. The traditional security notion for IBE, denoted by IND-ID-CPA, is the indistinguishability of ciphertexts under a target identity chosen by the adversary, who can obtain the secret keys of identities (other than the target identity) of its choice. The first constructions of IBE satisfying this notion were proposed by Boneh and Franklin [8] and Cocks [13], however the proofs were based on the random oracle model. The first IBE with IND-ID-CPA security in the standard model was presented by Boneh and Boyen [7], and later Waters [31] simplified the scheme of [7], substantially improving its efficiency.

1.1 Tight security reduction

Modern research on IBE pursues tight security reductions to standard cryptographic assumptions. It is not only an interesting theoretical problem but also has practical significance. A loose reduction makes the instantiations augment larger security parameter, hence the IBE scheme will be less efficient. A typical loose security reduction of IBE, for instance [31,32], is related to Q, the number of user secret key queries. Then implementing such an IBE has to set a larger parameter to compensate reduction's security loss. Recently, Chen and Wee [12] proposed the first (almost) tightly IND-ID-CPA secure IBE, which only loses a factor λ, independent of Q. Here λ is the bit-length of the identities. More works about tightly secure IBEs were done in [3,6,19,24].

1.2 Recipient-anonymity and ciphertext pseudorandomness

Informally, an IBE is recipient-anonymous, denoted by ANON-ID-CPA, if a ciphertext does not leak any information about the identity of the recipient to probabilistic polynomial-time (PPT) adversaries. Boneh et al. [9] observed that anonymous IBE can be used to construct searchable public-key encryption, and later Abdalla et al. [1] gave a formalization. Constructions of anonymous IBE were found in [10,28], both of which have loose security reductions.

Recently, Blazy et al. [6] proposed the first (almost) tightly ANON-ID-CPA secure IBE, and more work can be found in [3].

A stronger notion is called *ciphertext pseudorandomness*, denoted by PR-ID-CPA [2,6], which means that the ciphertext generated by IBE is indistinguishable from a random element in the ciphertext space to PPT adversaries. This notion implies that the ciphertext hides the plaintext (thus IND-ID-CPA), the identity (thus ANON-ID-CPA), and also the public key used to create it. Therefore, a PR-ID-CPA secure IBE even protects anonymity of authorities that issue user secret keys [2]. PR-ID-CPA security implies ANON-ID-CPA security, but not vice verse, i.e., an anonymous IBE does not necessarily have pseudorandom ciphertexts. Meanwhile, ciphertext pseudorandomness turned out to be very useful and it was implicitly used in PKE and IBE to achieve simulation-based selective opening CPA (SIM-SO-CPA) security, see [5,16].

1.3 CCA2 security

Security against adaptive chosen-ciphertext attacks is a de facto security notion. Active adversaries might also be able to obtain the decryptions of ciphertexts under any identity of its choice. Thus it is necessary to consider the stronger security notion for IBE, i.e., IND-ID-CCA2 security. Similarly, imposing anonymous property/ciphertext pseudorandomness to IND-ID-CCA2 gives ANON/PR-ID-CCA2 security. We stress again that PR-ID-CCA2 security was implicitly used in achieving simulation-based selective opening CCA2 (SIM-SO-CCA2) security for IBE in [27].

1.4 Tightly secure IBE with recipient-anonymity/ciphertext pseudorandomness

Gentry [17] proposed tightly ANON-ID-CPA/CCA2 secure IBEs. However, the IBE schemes rely on non-standard Q-assumptions, which again depend on the number Q of user secret key queries. Recently, Attrapadung et al. [3] used broadcast encoding to construct a tightly ANON-ID-CPA secure IBE. Blazy, Kiltz and Pan (BKP) [6] introduced the notion of affine message authentication code (MAC) and delegatable affine MAC, which were used in constructing tightly secure (H)IBEs in a novel way, including: (1) a tightly PR-ID-CPA secure IBE; (2) a tightly IND-ID-CPA secure but not anonymous HIBE.[1]

There is no tightly ANON-ID-CCA2 secure IBE from standard assumptions yet, to the best of our knowledge. Of course, a tightly ANON-ID-CCA2 secure IBE can be obtained from a tightly ANON-ID-CPA secure 2-level HIBE with the help of the CHK transformation [11]. Unfortunately, 2-level HIBE of tight ANON-ID-CPA security is still missing.

PR-ID-CCA2 security provides stronger privacy than ANON-ID-CCA2 security, since the ciphertexts are completely random. Meanwhile, PR-ID-CCA2 secure IBE plays an important role in building IBE of SIM-SO-CCA2 security. These observations motivate us to pursuit tight PR-ID-CCA2 security for IBE.

However, tight PR-ID-CCA2 security is much harder to achieve from standard assumptions. We stress that the CHK transformation does not work even if a tightly PR-ID-CPA secure 2-level HIBE is available, since the CHK transformation does not preserve ciphertext pseudorandomness. We have to resort to an alternative approach to solve the challenging problem:

How to construct PR-ID-CCA2 secure IBE possessing both security reduction tightness and ciphertext pseudorandomness from a standard assumption?

[1] As far as we know, this is the only HIBE with a tight security reduction.

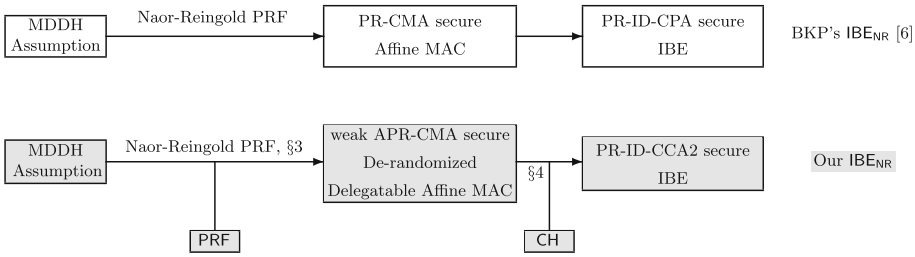

Fig. 1 Schematic overview of BKP's construction [6] (the *top row*) and our construction (the *bottom row*) of IBE. All implications have a tight security reduction. In our construction, we need a pseudorandom function PRF as well as a chameleon hashing CH, both of which have tightly secure instantiations

1.5 Our contributions

In this paper, we answer the above question affirmatively. We propose a new IBE over pairing groups of prime order. Our IBE enjoys (almost) tight security reduction, IND-ID-CCA2 security and ciphertext pseudorandomness simultaneously. The security of our IBE is based on the Matrix Decisional Diffie-Hellman (MDDH) assumption [15], which includes the standard k-Linear and DDH assumptions. More precisely,

- We introduce a new concept, namely *De-randomized Delegatable Affine MAC*, and define a security notion for the MAC, i.e., *the weak APR-CMA security*.[2] We build a de-randomized delegatable affine MAC with weak APR-CMA security based on the Naor-Reingold PRF (NR-PRF) and a traditional pseudorandom function, and the security is tightly reduced to the MDDH assumption.
- We propose a paradigm for constructing PR-ID-CCA2 secure IBE from de-randomized delegatable affine MAC and chameleon hashing. The security reduction is tightness preserving. When instantiating MAC with the NR-PRF-based one,
  - we obtain the first tightly PR-ID-CCA2 secure IBE, which is also the first tightly ANON-ID-CCA2 secure IBE, based on the MDDH assumption;
  - we obtain the first tightly IND-ID-CCA2 secure extractable IBE based on the MDDH assumption, when our tightly PR-ID-CCA2 secure IBE is converted to an extractable IBE (the conversion is shown in Appendix 3 as a minor contribution);
  - we obtain a SIM-SO-CCA2 secure IBE from the MDDH assumption with a much tighter reduction, when following the black-box construction of SIM-SO-CCA2 secure IBE from extractable IBE [27].

Our results are illustrated on the bottom of Fig. 1. We summarize known tightly secure IBEs, their securities and parameters in Table 1.

1.6 Our approach

Firstly, we recall BKP's approach to tightly PR-ID-CPA secure IBE [6]. Then we introduce our approach to tightly PR-ID-CCA2 secure IBE. We will explain the intuitions behind our construction, the problems arising along the way and the methods to solve them.

[2] The APR-CMA security was originally defined by Blazy et al. [6] for delegatable affine MAC, but no constructions are available with tight security reduction.

Table 1 Comparison between the known tightly-secure IBEs with identity space $\mathcal{ID} = \{0,1\}^\lambda$ in prime order groups based on standard assumptions

IBE Scheme	Loss	CCA2?	ANON?	PR?	Assumption	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{usk}\rvert$	$\lvert\mathsf{C}\rvert$
CW13 [12]	$O(\lambda)$	–	–	–	k-LIN	$2\lambda k^2$	$4k$	$4k$
BKP14's $\mathsf{IBE}_{\mathsf{NR}}$ [6]	$O(\lambda)$	–	√	√	$\mathcal{D}_k$-MDDH	$2\lambda k^2$	$2k$	$2k$
BKP14's $\mathsf{HIBE}_{\mathsf{NR}}$ [6] + CHK	$O(\lambda)$	√	–	–	$\mathcal{D}_k$-MDDH	$8\lambda k^2$	$2\lambda k$	$2k$
AHY15[a] [3]	$O(\lambda)$	–	√	–	2-LIN	16λ	8	8
GCDCT16 [19]	$O(\lambda)$	–	–	–	k-LIN	$6\lambda k^2$	$6k$	$6k$
Ours: $\mathsf{IBE}_{\mathsf{NR}}$ (§4)	$O(\lambda)$	√	√	√	$\mathcal{D}_k$-MDDH	$4\lambda k^2$	$2\lambda k$	$2k$

"CHK" stands for the general transformation proposed by [11], which converts IND-ID-CPA secure 2-level HIBE to IND-ID-CCA2 secure IBE. "CCA2" asks whether the scheme is IND-ID-CCA2 secure, "ANON" anonymous, and "PR" pseudorandom ciphertexts which is stronger than recipient-anonymity. "k-LIN" is short for the k-Linear assumption, and "$\mathcal{D}_k$-MDDH" the $\mathcal{D}_k$-Matrix DDH assumption. $\mathcal{D}_k$-MDDH is more general than k-LIN. $\lvert\mathsf{pk}\rvert$, $\lvert\mathsf{usk}\rvert$ and $\lvert\mathsf{C}\rvert$ denote the size per public key, user secret key and ciphertext, respectively. Here we count the number of group elements in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, and we only show the leading term due to the lack of space (a detailed efficiency comparison table is in Appendix 1)

[a] AHY15 [3] also proposed two non-anonymous schemes, here we only show the anonymous one

We will sketch the approach in terms of an identity-based key encapsulation mechanism (IBKEM). To get a full-fledged IBE, one can simply combine an IBKEM with a (one-time secure) authenticated encryption scheme.

1.6.1 BKP's approach to tightly PR-ID-CPA secure IBE

In [6], BKP proposed the concept of affine MAC which was used to build IBE. They defined PR-CMA security for affine MAC, which was dedicated to PR-ID-CPA security of IBE. See the top row of Fig. 1.

Roughly speaking, an affine MAC uses secret key $\mathsf{sk}_{\mathsf{MAC}} = (\mathbf{B}, \mathbf{x}_0, \ldots, \mathbf{x}_\ell, x'_0)$ to compute a tag $([\mathbf{t}]_2, [u]_2)$ for a message m as follows:

$$\mathbf{s} \leftarrow\!\!\$\ \mathbb{Z}_q^{n'}, \quad \mathbf{t} := \mathbf{B} \cdot \mathbf{s} \in \mathbb{Z}_q^n, \quad u := \sum_{i=0}^{l(\mathsf{m})} f_i(\mathsf{m}) \cdot \mathbf{x}_i^\top \cdot \mathbf{t} + x'_0 \in \mathbb{Z}_q,$$

where f_i mapping m to $\mathbb{Z}_q$ and l mapping m to $\{0, 1, \ldots, \ell\}$ are some public defining functions, and $[\mathbf{t}]_j \in \mathbb{G}_j^n$ ($j \in \{1, 2, T\}$) is an implicit expression of $\mathbf{t} \in \mathbb{Z}_q^n$ over a group $\mathbb{G}_j$ of prime order q.

The PR-CMA security of affine MAC requires pseudorandomness of token $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$ for a target message id^*, given multiple tags $([\mathbf{t}]_2, [u]_2)$ of messages of adversaries' choice, where

$$h \leftarrow\!\!\$\ \mathbb{Z}_q, \qquad \mathbf{h}_0 := \sum_{i=0}^{l(\mathsf{id}^*)} f_i(\mathsf{id}^*) \cdot \mathbf{x}_i \cdot h \in \mathbb{Z}_q^n, \qquad h_1 := x'_0 \cdot h \in \mathbb{Z}_q.$$

BKP constructed an IBKEM from an affine MAC. The high-level idea behind their approach is the Bellare–Goldwasser transformation [4] from MAC, commitment and NIZK to digital signature.[3] Let us briefly recall their IBKEM scheme. The public key pk consists of $[\mathbf{A}]_1 \in \mathbb{G}_1^{(k+1)\times k}$, $[\mathbf{Z}_i]_1 = [(\mathbf{Y}_i^\top \mid \mathbf{x}_i) \cdot \mathbf{A}]_1$ and $[\mathbf{z}'_0]_1 = [(\mathbf{y}'^\top_0 \mid x'_0) \cdot \mathbf{A}]_1$ which can be seen as perfect hiding commitment of $(\{\mathbf{x}_i\}, x'_0)$. In the key encapsulation algorithm $\mathsf{Encap}(\mathsf{pk}, \mathsf{id}^*)$, a ciphertext $([\mathbf{c}_0]_1, [\mathbf{c}_1]_1)$ encapsulates a symmetric key $[K]_T$ with randomness $\mathbf{r} \leftarrow\!\!\$\ \mathbb{Z}_q^k$:

[3] Recall that the user secret key generation algorithm of IBKEM corresponds to the signing algorithm of digital signature scheme.

$$\mathbf{c}_0 := \mathbf{A} \cdot \mathbf{r} \in \mathbb{Z}_q^{k+1}, \quad \mathbf{c}_1 := \sum_{i=0}^{l(\mathsf{id}^*)} f_i(\mathsf{id}^*) \cdot \mathbf{Z}_i \cdot \mathbf{r} \in \mathbb{Z}_q^n, \quad K := \mathbf{z}_0' \cdot \mathbf{r} \in \mathbb{Z}_q, \quad (1)$$

which can also be computed with master secret key $(\{\mathbf{x}_i\}, x_0', \{\mathbf{Y}_i\}, \mathbf{y}_0')$ of IBKEM:

$$\mathbf{c}_0 := \mathbf{A} \cdot \mathbf{r} \in \mathbb{Z}_q^{k+1}, \quad \mathbf{c}_1 := \sum_{i=0}^{l(\mathsf{id}^*)} f_i(\mathsf{id}^*) \cdot (\mathbf{Y}_i^\top \mid \mathbf{x}_i) \cdot \mathbf{c}_0, \quad K := (\mathbf{y}_0'^\top \mid x_0') \cdot \mathbf{c}_0. \quad (2)$$

In the PR-ID-CPA security proof of IBKEM, some entropy can always be introduced to the last row of $\mathbf{c}_0$, according to the MDDH assumption (cf. Definition 1):

$$\mathbf{c}_0 := \mathbf{A} \cdot \mathbf{r} + (\mathbf{0} \mid h)^\top, \quad \text{with } h \leftarrow_\$ \mathbb{Z}_q. \quad (3)$$

According to (2), we have that

$$\mathbf{c}_1 := \sum_{i=0}^{l(\mathsf{id}^*)} f_i(\mathsf{id}^*) \cdot \mathbf{Z}_i \cdot \mathbf{r} + \underbrace{\sum_{i=0}^{l(\mathsf{id}^*)} f_i(\mathsf{id}^*) \cdot \mathbf{x}_i \cdot h}_{\mathbf{h}_0}, \quad K := \mathbf{z}_0' \cdot \mathbf{r} + \underbrace{x_0' \cdot h}_{h_1}.$$

Finally, the PR-ID-CPA security of IBKEM can be tightly reduced to the PR-CMA security of affine MAC, where $([\mathbf{c}_0]_1, [\mathbf{c}_1]_1, [K]_T)$ is pseudorandom due to the pseudorandomness of token $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$, and the user secret key generation can be implemented by tag generation of affine MAC.

1.6.2 Our approach to tightly PR-ID-CCA2 secure IBE

BKP's IBKEM is not PR-ID-CCA2 secure. According to (1), it is clear that the ciphertext is linearly homomorphic in the sense that, if $([\mathbf{c}_0]_1, [\mathbf{c}_1]_1)$ encapsulates a key $[K]_T$, then the ciphertext $([a \cdot \mathbf{c}_0]_1, [a \cdot \mathbf{c}_1]_1)$ will encapsulate the key $[a \cdot K]_T$. This implies a trivial CCA2 attack.

To circumvent this attack and achieve CCA2 security, we make use of chameleon hashing CH to eliminate this linear homomorphism. In the key encapsulation algorithm $\mathsf{Encap}(\mathsf{pk}, \mathsf{id}^*)$ of our IBKEM, we bind a (second-level) identity id'^* with $[\mathbf{c}_0]_1$ (and a randomness R_{CH}) via $\mathsf{id}'^* = \mathsf{CH.Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1; R_{\mathsf{CH}})$, and compute $[\mathbf{c}_1]_1$ with respect to the hierarchical identity $\mathsf{id}^* | \mathsf{id}'^*$ (instead of id^*), i.e.,

$$\mathbf{c}_0 := \mathbf{A} \cdot \mathbf{r}, \qquad \mathsf{id}'^* = \mathsf{CH.Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1; R_{\mathsf{CH}}),$$
$$\mathbf{c}_1 := \sum_{i=0}^{l(\mathsf{id}^* | \mathsf{id}'^*)} f_i(\mathsf{id}^* | \mathsf{id}'^*) \cdot \mathbf{Z}_i \cdot \mathbf{r}, \qquad K := \mathbf{z}_0' \cdot \mathbf{r}.$$

The above CCA2 attack no longer works for our IBKEM. In our scheme, a transformed ciphertext $([a \cdot \mathbf{c}_0]_1, [a \cdot \mathbf{c}_1]_1, \tilde{R}_{\mathsf{CH}})$ is not able to encapsulate the key $[a \cdot K]_T$ any more, since the second-level identity $\widetilde{\mathsf{id}}'^*$ bound with $[a \cdot \mathbf{c}_0]_1$ (using randomness $\tilde{R}_{\mathsf{CH}}$) is totally different from the second-level identity id'^* bound with $[\mathbf{c}_0]_1$ (using randomness R_{CH}), unless a CH-collision occurs.

1.6.3 New problem and our solution

We note that a new problem arises in the proof of the ciphertext pseudorandomness of our scheme, when embedding the pseudorandom token $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$ w.r.t. $\mathsf{id}^* | \mathsf{id}'^*$ to the challenge ciphertext $([\mathbf{c}_0]_1, [\mathbf{c}_1]_1)$ and the encapsulated key $[K]_T$, where

$$h \leftarrow_\$ \mathbb{Z}_q, \qquad \mathbf{h}_0 := \sum_{i=0}^{l(\mathsf{id}^* | \mathsf{id}'^*)} f_i(\mathsf{id}^* | \mathsf{id}'^*) \cdot \mathbf{x}_i \cdot h \in \mathbb{Z}_q^n, \qquad h_1 := x_0' \cdot h \in \mathbb{Z}_q.$$

On the one hand, since $\mathsf{id}'^* = \mathsf{Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1; R_{\mathsf{CH}})$, the simulator needs to compute $[\mathbf{c}_0]_1$ before obtaining the token w.r.t. target $\mathsf{id}^*|\mathsf{id}'^*$. On the other hand, in order to embed $[h]_1$ into $[\mathbf{c}_0]_1$ according to (3), the simulator has to compute $[\mathbf{c}_0]_1$ after getting the token $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$.

To break this deadlock, we make use of the *equivocation* property of chameleon hashing CH (cf. Definition 4). More precisely, the simulator first picks dummy $[\tilde{\mathbf{c}}_0]_1$, $\tilde{R}_{\mathsf{CH}}$ randomly, and compute $\mathsf{id}'^* := \mathsf{Eval}(ek_{\mathsf{CH}}, [\tilde{\mathbf{c}}_0]_1; \tilde{R}_{\mathsf{CH}})$. Then the simulator submits $\mathsf{id}^*|\mathsf{id}'^*$ as the target message and obtains the token $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$. After embedding the challenge $[h]_1$ into $[\mathbf{c}_0]_1$, the simulator can reopen $R_{\mathsf{CH}} \leftarrow \mathsf{Equiv}(td_{\mathsf{CH}}, [\tilde{\mathbf{c}}_0]_1, \tilde{R}_{\mathsf{CH}}, [\mathbf{c}_0]_1)$ using the trapdoor td_{CH}. By the equivocation property of CH, it holds that $\mathsf{id}'^* = \mathsf{Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1; R_{\mathsf{CH}})$ and R_{CH} is uniformly distributed and independent of $[\mathbf{c}_0]_1$.

1.6.4 Another problem

In order to support second-level identity id'^* in our IBKEM, we need affine MAC supporting hierarchical message space, i.e., Delegatable affine MAC. The syntax of delegatable affine MAC and corresponding security notion, i.e., APR-CMA security, were proposed by BKP. The APR-CMA security requires pseudorandom token for delegatable affine MAC, which is used to construct PR-ID-CPA secure HIBE in [6].

However, it is much more difficult for MAC to get APR-CMA security than PR-CMA. In the definition of APR-CMA security of delegatable affine MAC, when the adversary adaptively queries a message, it gets not only a tag, but also some extra elements served for delegation and re-randomization in HIBE. Those elements for re-randomization might information-theoretically reveal the critical part of the secret key to the adversary. This makes APR-CMA security very hard to achieve.

In [6], only one construction of APR-CMA secure delegatable affine MAC was proposed, but with a loose security reduction. As far as we know, no MAC is available possessing tight APR-CMA security yet.

1.6.5 Our solution: new security notion and new syntax

Fortunately, delegatable affine MAC is used to construct IBE (instead of HIBE) in our scenario, hence those elements used for re-randomization are not necessary and can be safely discarded in the APR-CMA security game.

Based on this observation, we define a weak version of APR-CMA security, i.e., the *weak APR-CMA security*, which still stipulates pseudorandom token, but no re-randomization elements are revealed to the adversary.

In order to get a tightly secure MAC, we further introduce the de-randomization technique and define a new primitive called *De-randomized delegatable affine MAC*. Roughly speaking, we require that the tag generation algorithm of MAC always employs the same "randomness" for the same message, i.e., the tag generation algorithm is deterministic. This can further restrict the information on secret key that the adversary may obtain in the security game. The de-randomization technique plays an essential role in achieving weak APR-CMA security with a tight security reduction.

The tuned security requirement and syntax make it possible for us to obtain a tightly weakly APR-CMA secure De-randomized delegatable affine MAC from the MDDH assumption.

Using this MAC in our IBKEM construction, together with an information-theoretically secure authenticated encryption, we obtain an IBE of tight PR-ID-CCA2 security. To reduce

the PR-ID-CCA2 security of our IBE to the weak APR-CMA security of MAC, we need to show that one can simulate the decryption oracle of IBE with the help of the tag generation oracle of MAC. This part is fairly involved and technical, and we leave the detailed analysis to Sect. 4.

2 Preliminaries

2.1 Notations

Throughout this paper, $\lambda \in \mathbb{N}$ denotes the security parameter. For integers $i, j \in \mathbb{N}$ with $i < j$, $[i, j]$ denotes the set $\{i, i+1, \ldots, j\}$ and $[j]$ denotes the set $\{1, 2, \ldots, j\}$. For a finite set $\mathcal{S}$, we denote by $s \leftarrow_\$ \mathcal{S}$ the operation of picking an element s from $\mathcal{S}$ uniformly at random. For an algorithm $\mathcal{A}$, we denote by $y \leftarrow_\$ \mathcal{A}(x)$ the operation of running $\mathcal{A}$ with input x, and assigning y as the result. The symbol ε denotes the empty string. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{(k+1)\times n}$, denote the upper k rows of $\mathbf{A}$ by $\overline{\mathbf{A}} \in \mathbb{Z}_q^{k\times n}$ and the last row by $\underline{\mathbf{A}} \in \mathbb{Z}_q^{1\times n}$. For a string $\mathsf{s} \in \{0,1\}^*$, $|\mathsf{s}|$ always denotes the bit-length of s. Let $\mathsf{s}||\mathsf{t}$ denote the concatenation of two strings s and t.

2.2 Games

We use games for our security reductions, as in [6]. A game G consists of an INITIALIZE procedure and a FINALIZE procedure, as well as some optional (public) procedures $\text{PROC}_1, \ldots, \text{PROC}_n$ and private procedures $\text{PRIVPROC}_1, \ldots, \text{PRIVPROC}_m$. All procedures are described using pseudo-code, where initially all variables are empty strings and all sets are empty. An adversary is executed in game G if it first calls INITIALIZE, obtaining its output. Then it may make arbitrary queries to (public) procedures PROC_i according to their specification, and obtain their output. The adversary is not allowed to query private procedures PRIVPROC_i directly. Finally it makes one single call to FINALIZE. By $\mathsf{G}^{\mathcal{A}} \Rightarrow b$ we means that the game G outputs b after interacting with $\mathcal{A}$, and b is in fact the output of FINALIZE. We denote by $\Pr_i[\cdot]$ the probability of a particular event occurring in game G_i. By $a \overset{\mathsf{G}}{=} b$ we mean that a equals b or is computed as b in game G.

2.3 Pairing groups and implicit representation of group elements

Let $\mathsf{PGGen}(1^\lambda)$ be a PPT algorithm that on input the security parameter 1^λ outputs a description $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, g_1, g_2)$ of asymmetric pairing groups, where $(\mathbb{G}_1, \cdot)$, $(\mathbb{G}_2, \cdot)$, $(\mathbb{G}_T, \cdot)$ are cyclic groups of a λ-bit prime order q and equipped with a non-degenerated bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$, and g_1, g_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Denote $g_T := e(g_1, g_2)$, which is a generator of $\mathbb{G}_T$. The pairing e is required to be efficiently computable.

We recall the implicit representation of group elements [15]. Let $s \in \{1, 2, T\}$. For a matrix $\mathbf{A} = (a_{i,j})_{i,j}$ over $\mathbb{Z}_q$, denote by $[\mathbf{A}]_s := g_s^{\mathbf{A}} = (g_s^{a_{i,j}})_{i,j}$ the matrix over $\mathbb{G}_s$, which is the implicit representation of $\mathbf{A}$ in $\mathbb{G}_s$.

The above implicit representation has the following properties : (1) given $[\mathbf{A}]_s$ and $[\mathbf{B}]_s$ with appropriate dimensions, we can efficiently compute $[\mathbf{A} \pm \mathbf{B}]_s$; (2) given $\mathbf{A}$ and $[\mathbf{B}]_s$, or

$[\mathbf{A}]_s$ and $\mathbf{B}$ with appropriate dimensions, we can efficiently compute $[\mathbf{A}\cdot\mathbf{B}]_s$; (3) given $[\mathbf{A}]_1$ and $[\mathbf{B}]_2$ with appropriate dimensions, we can efficiently compute $[\mathbf{A}\cdot\mathbf{B}]_T$.

2.4 MDDH assumption, PRF, chameleon hashing and universal hashing

Let $k \in \mathbb{N}$. A probabilistic distribution $\mathcal{D}_k$ is called a *matrix distribution*, if it outputs matrices in $\mathbb{Z}_q^{(k+1)\times k}$ of full rank k in polynomial time. Without loss of generality, we assume that the first k rows of $\mathbf{A} \leftarrow\!\!\$\ \mathcal{D}_k$ are linearly independent.

We recall the definition of the Matrix Decisional Diffie-Hellman (MDDH) assumption from [15].

Definition 1 ($\mathcal{D}_k$*-MDDH assumption*) Let $\mathcal{D}_k$ be a matrix distribution and $s \in \{1, 2, T\}$. The $\mathcal{D}_k$-Matrix DDH ($\mathcal{D}_k$-MDDH) Assumption holds w.r.t. PGGen for $\mathbb{G}_s$, if for any PPT adversary $\mathcal{A}$, the following is negligible in λ:

$$\mathsf{Adv}^{\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_s,\mathcal{A}}(\lambda) := \left|\Pr\left[\mathcal{A}\big(\mathcal{PG}, [\mathbf{A}]_s, [\mathbf{A}\cdot\mathbf{w}]_s\big) = 1\right] - \Pr\left[\mathcal{A}\big(\mathcal{PG}, [\mathbf{A}]_s, [\mathbf{r}]_s\big) = 1\right]\right|,$$

where $\mathcal{PG} \leftarrow\!\!\$\ \mathsf{PGGen}(1^\lambda)$, $\mathbf{A} \leftarrow\!\!\$\ \mathcal{D}_k$, $\mathbf{w} \leftarrow\!\!\$\ \mathbb{Z}_q^k$, $\mathbf{r} \leftarrow\!\!\$\ \mathbb{Z}_q^{k+1}$.

The $\mathcal{D}_k$-MDDH assumption covers many well-known assumptions, such as the DDH and k-Linear assumptions [15].

Definition 2 (Q*-fold* $\mathcal{D}_k$*-MDDH assumption*) Let $Q \geq 1$. The Q-fold $\mathcal{D}_k$-MDDH Assumption holds w.r.t. PGGen for $\mathbb{G}_s$, if for any PPT adversary $\mathcal{A}$, the following is negligible in λ:

$$\mathsf{Adv}^{Q,\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_s,\mathcal{A}}(\lambda) := \left|\Pr\left[\mathcal{A}\big(\mathcal{PG}, [\mathbf{A}]_s, [\mathbf{A}\cdot\mathbf{W}]_s\big) = 1\right] - \Pr\left[\mathcal{A}\big(\mathcal{PG}, [\mathbf{A}]_s, [\mathbf{R}]_s\big) = 1\right]\right|,$$

where $\mathcal{PG} \leftarrow\!\!\$\ \mathsf{PGGen}(1^\lambda)$, $\mathbf{A} \leftarrow\!\!\$\ \mathcal{D}_k$, $\mathbf{W} \leftarrow\!\!\$\ \mathbb{Z}_q^{k\times Q}$, $\mathbf{R} \leftarrow\!\!\$\ \mathbb{Z}_q^{(k+1)\times Q}$.

The following lemma gives a tight reduction from the Q-fold $\mathcal{D}_k$-MDDH assumption to the (1-fold) $\mathcal{D}_k$-MDDH assumption.

Lemma 1 (Random self-reducibility [15]) *For any matrix distribution $\mathcal{D}_k$, the $\mathcal{D}_k$-MDDH assumption is random self-reducible.*

More precisely, for any $Q, k \geq 1$, suppose that $\mathcal{A}$ is a PPT adversary against the Q-fold $\mathcal{D}_k$-MDDH assumption w.r.t. PGGen *for group $\mathbb{G}_s$ of order q, then there exists a PPT adversary $\mathcal{B}$ against the (1-fold) $\mathcal{D}_k$-MDDH assumption w.r.t.* PGGen *for $\mathbb{G}_s$, such that*

$$\mathsf{Adv}^{Q,\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_s,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_s,\mathcal{B}}(\lambda) + 1/q.$$

Let $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{X} \longrightarrow \mathcal{Y}$ be a polynomial-time computable function with key space $\mathcal{K}_{\mathsf{PRF}}$, domain $\mathcal{X}$ and range $\mathcal{Y}$. Roughly speaking, PRF is pseudorandom if its outputs are computationally indistinguishable from those of a truly random function, even the inputs are adaptively chosen by PPT adversaries.

Definition 3 (*Pseudorandom function*) PRF is a pseudorandom function, if for any PPT adversary $\mathcal{A}$, which has oracle access to a function from $\mathcal{X}$ to $\mathcal{Y}$, the following is negligible in λ:

$$\mathsf{Adv}^{prf}_{\mathsf{PRF},\mathcal{A}}(\lambda) := \left|\Pr\left[\mathcal{A}^{\mathsf{PRF}(\mathsf{k}_{\mathsf{PRF}},\cdot)}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{TRF}(\cdot)}(1^\lambda) = 1\right]\right|,$$

where $\mathsf{k}_{\mathsf{PRF}} \leftarrow\!\!\$\ \mathcal{K}_{\mathsf{PRF}}$ and TRF is a truly random function chosen uniformly from the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$.

A chameleon hashing function CH [26] is associated with an evaluation key and a trapdoor. Its collision-resistant property holds when only the evaluation key is known. With the trapdoor, however, collision can be easily found. We recall the definition of chameleon hashing from [22].

Definition 4 (*Chameleon hashing*) A chameleon hashing function $\mathsf{CH} = (\mathsf{CH.Gen}, \mathsf{Eval}, \mathsf{Equiv})$ consists of three PPT algorithms:

- The key generation algorithm $\mathsf{CH.Gen}(1^\lambda)$ outputs an evaluation key ek_{CH} and a trapdoor td_{CH}.
- The evaluation algorithm $\mathsf{Eval}(ek_{\mathsf{CH}}, X; R_{\mathsf{CH}})$ takes as input an evaluation key ek_{CH}, $X \in \{0,1\}^*$ and a randomness $R_{\mathsf{CH}} \in \mathcal{R}_{\mathsf{CH}}$, and outputs $Y \in \mathcal{Y}$. We require that for any possible ek_{CH} and $X \in \{0,1\}^*$, if R_{CH} is uniformly distributed over $\mathcal{R}_{\mathsf{CH}}$, then so is Y over $\mathcal{Y}$.
- The equivocation algorithm $\mathsf{Equiv}(td_{\mathsf{CH}}, X, R_{\mathsf{CH}}, X')$ takes as input a trapdoor td_{CH}, $X, X' \in \{0,1\}^*$ and $R_{\mathsf{CH}} \in \mathcal{R}_{\mathsf{CH}}$, and outputs $R'_{\mathsf{CH}} \in \mathcal{R}_{\mathsf{CH}}$ satisfying
$$\mathsf{Eval}(ek_{\mathsf{CH}}, X; R_{\mathsf{CH}}) = \mathsf{Eval}(ek_{\mathsf{CH}}, X'; R'_{\mathsf{CH}}) \tag{4}$$
for the corresponding key ek_{CH}. We require that for any possible td_{CH} and $X, X' \in \{0,1\}^*$, if R_{CH} is uniformly distributed over $\mathcal{R}_{\mathsf{CH}}$, then so is R'_{CH}.

CH is called *collision-resistant*, if for any PPT adversary $\mathcal{A}$, the following advantage is negligible in λ:

$$\mathsf{Adv}^{cr}_{\mathsf{CH},\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{c} (ek_{\mathsf{CH}}, td_{\mathsf{CH}}) \leftarrow\$ \mathsf{CH.Gen}(1^\lambda), \\ (X, R_{\mathsf{CH}}; X', R'_{\mathsf{CH}}) \leftarrow\$ \mathcal{A}(ek_{\mathsf{CH}}) \end{array} : \begin{array}{c} (X, R_{\mathsf{CH}}) \neq (X', R'_{\mathsf{CH}}) \\ \wedge \text{ Eq. (4) holds} \end{array}\right].$$

Definition 5 (*Universal hash* [33]) A family of functions $\mathcal{H} = \{\mathsf{H} : \mathcal{X} \longrightarrow \mathcal{Y}\}$ is universal, if all distinct $x, x' \in \mathcal{X}$, it follows that

$$\Pr\left[\mathsf{H} \leftarrow\$ \mathcal{H} \ : \ \mathsf{H}(x) = \mathsf{H}(x')\right] \leq 1/|\mathcal{Y}|.$$

We will sometimes abuse notation and say that a function H is universal if H is randomly chosen from a universal family of functions $\mathcal{H}$.

We state a simplified version of Leftover Hash Lemma [21] with uniform input.

Lemma 2 (Leftover Hash Lemma) *Let $\mathcal{H} = \{\mathsf{H} : \mathcal{X} \longrightarrow \mathcal{Y}\}$ be a family of universal hash functions. Let X be the uniform distribution over $\mathcal{X}$. Then for $\mathsf{H} \leftarrow\$ \mathcal{H}$, where H and X are independent, it holds that*

$$\Delta\big((\mathsf{H}, \mathsf{H}(X)),\ (\mathsf{H}, \mathsf{U}_{\mathcal{Y}})\big) \leq \frac{1}{2} \cdot \sqrt{|\mathcal{Y}|/|\mathcal{X}|},$$

where $\mathsf{U}_{\mathcal{Y}}$ is the uniform distribution over $\mathcal{Y}$. In particular, if $|\mathcal{Y}|/|\mathcal{X}| \leq 2^{-\Omega(\ell)}$, $(\mathsf{H}, \mathsf{H}(X))$ is statistically close to the uniform distribution over $\mathcal{H} \times \mathcal{Y}$.

2.5 Delegatable affine MAC and APR-CMA security

A message authentication code (MAC) $\mathsf{MAC} = (\mathsf{MAC.Gen}, \mathsf{Tag}, \mathsf{Vrfy})$ consists of a tuple of PPT algorithms: (1) $\mathsf{MAC.Gen}(\mathsf{params})$ takes as input a system parameter params, and outputs an authentication key $\mathsf{sk}_{\mathsf{MAC}}$. (2) $\mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m})$ is a randomized algorithm. It takes as input $\mathsf{sk}_{\mathsf{MAC}}$ and a message m, and outputs a tag t. (3) $\mathsf{Vrfy}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m}, \mathsf{t})$ takes as input

$\mathsf{sk}_{\mathsf{MAC}}$, a message m and a tag t, and outputs a verification bit $b \in \{0, 1\}$. Correctness of MAC requires that for all possible params and $\mathsf{sk}_{\mathsf{MAC}} \leftarrow\!\!\$\ \mathsf{MAC.Gen}(\mathsf{params})$, all messages m, we have that $\mathsf{Vrfy}\big(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m}, \mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m})\big) = 1$. Delegatable affine MAC is group-based MAC with specific algebraic structures.

Definition 6 (*Delegatable affine MAC* [6]) Let q be a prime number, and $n, L \in \mathbb{N}$. We say that $\mathsf{MAC} = (\mathsf{MAC.Gen}, \mathsf{Tag}, \mathsf{Vrfy})$ is a delegatable affine MAC over $\mathbb{Z}_q^n$, if the following holds:

1. The system parameters params contains a group description $(\mathbb{G}_2, q, g_2)$.
2. The secret key $\mathsf{sk}_{\mathsf{MAC}}$ contains $(\mathbf{B}, \mathbf{x}_0, \ldots, \mathbf{x}_\ell, x'_0)$, where $\mathbf{B} \in \mathbb{Z}_q^{n \times n'}$, $\mathbf{x}_i \in \mathbb{Z}_q^n$, $x'_0 \in \mathbb{Z}_q$ for some $n', \ell \in \mathbb{N}$, and $\mathbf{B}$ has rank at least one.
3. The message space is $\mathcal{M} = \mathcal{B}^{\leq L}$ $(= \cup_{p \in [L]} \mathcal{B}^p)$ for some finite base set $\mathcal{B}$. For messages $\mathsf{m} = (\mathsf{m}_1, \ldots, \mathsf{m}_p) \in \mathcal{B}^p$ and $\mathsf{m}' = (\mathsf{m}'_1, \ldots, \mathsf{m}'_{p'}) \in \mathcal{B}^{p'}$, m is called a prefix of m', denoted by $\mathsf{m} \sqsubseteq \mathsf{m}'$, if $p \leq p'$, and for all $i \in [1, p]$, $\mathsf{m}_i = \mathsf{m}'_i$. For a message $\mathsf{m} \in \mathcal{M}$, denote by $\mathsf{prefix}(\mathsf{m}) := \{\mathsf{m}' \in \mathcal{M} \mid \mathsf{m}' \sqsubseteq \mathsf{m}\}$ the set of all prefixes of m.
4. $\mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m})$ computes a tag $([\mathbf{t}]_2, [u]_2) \in \mathbb{G}_2^n \times \mathbb{G}_2$ as

$$\mathbf{s} \leftarrow\!\!\$\ \mathbb{Z}_q^{n'}, \quad \mathbf{t} := \mathbf{B} \cdot \mathbf{s} \in \mathbb{Z}_q^n, \quad u := \sum\nolimits_{i=0}^{l(\mathsf{m})} f_i(\mathsf{m}) \cdot \mathbf{x}_i^\top \cdot \mathbf{t} + x'_0 \in \mathbb{Z}_q, \tag{5}$$

   where $f_i : \mathcal{M} \longrightarrow \mathbb{Z}_q$ and $l : \mathcal{M} \longrightarrow [0, \ell]$ are some public defining functions satisfying

   (a) For any message $\mathsf{m} \in \mathcal{M}$, we have $f_i(\mathsf{m}) = 0$ for all $i \in [l(\mathsf{m}) + 1, \ell]$.
   (b) For any two messages $\mathsf{m} \sqsubseteq \mathsf{m}' \in \mathcal{M}$, we have $l(\mathsf{m}) \leq l(\mathsf{m}')$, and $f_i(\mathsf{m}) = f_i(\mathsf{m}')$ for all $i \in [0, l(\mathsf{m})]$.
5. $\mathsf{Vrfy}\big(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m}, ([\mathbf{t}]_2, [u]_2)\big)$ verifies (5) via $[u]_2 \stackrel{?}{=} \left[\sum_{i=0}^{l(\mathsf{m})} f_i(\mathsf{m}) \cdot \mathbf{x}_i^\top \cdot \mathbf{t} + x'_0\right]_2$.

In [6], APR-CMA (anonymity-preserving pseudorandomness against chosen-message attacks) security is defined for delegatable affine MAC, which is dedicated to PR-ID-CPA secure HIBE. The APR-CMA security is reviewed as follows. Let $\mathcal{PG} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, g_1, g_2)$ be an asymmetric pairing group such that $(\mathbb{G}_2, q, g_2)$ is contained in params. Consider the $\mathsf{APR\text{-}CMA}$ game in Fig. 2, where μ is the (publicly known) rank of matrix $\mathbf{B}$ output by $\mathsf{MAC.Gen}(\mathsf{params})$.

Definition 7 (*APR-CMA security*) A delegatable affine MAC over $\mathbb{Z}_q^n$ is APR-CMA secure, if for any PPT adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{apr\text{-}cma}_{\mathsf{MAC},\mathcal{A}}(\lambda) := |\Pr[\mathsf{APR\text{-}CMA}^{\mathcal{A}} \Rightarrow 1] - 1/2|$ is negligible in λ, where game $\mathsf{APR\text{-}CMA}$ is specified in Fig. 2.

3 De-randomized delegatable affine MAC with weak APR-CMA security

We will give the formal definitions of De-randomized delegatable affine MAC and its weak APR-CMA security, and present a tightly secure instantiation of de-randomized delegatable affine MAC satisfying our new security notion.

3.1 De-randomized delegatable affine MACs and its weak APR-CMA security

Definition 8 (*De-randomized delegatable affine MAC*) A delegatable affine $\mathsf{MAC} = (\mathsf{MAC.Gen}, \mathsf{Tag}, \mathsf{Vrfy})$ over $\mathbb{Z}_q^n$ is de-randomized, if the following property of de-randomization holds:

Procedure $\textsc{Initialize}_{\mathsf{MAC}}$:
$\mathsf{sk}_{\mathsf{MAC}} \leftarrow\!\!\$\ \mathsf{MAC.Gen}(\mathsf{params})$,
where $\mathsf{sk}_{\mathsf{MAC}} = (\boxed{\mathsf{k}_{\mathsf{PRF}},} \mathbf{B}, \mathbf{x}_0, \cdots, \mathbf{x}_\ell, x'_0)$.
$\beta \leftarrow\!\!\$\ \{0,1\}$. // challenge bit
Return ε.

Procedure $\textsc{Chal}(\mathsf{m}^*)$: // one query
If $\mathsf{prefix}(\mathsf{m}^*) \cap \mathcal{QEVA} \neq \emptyset$, Return $\perp$.
$h \leftarrow\!\!\$\ \mathbb{Z}_q$.
$\mathbf{h}_0 := \sum_{i=0}^{l(\mathsf{m}^*)} f_i(\mathsf{m}^*) \cdot \mathbf{x}_i \cdot h \in \mathbb{Z}_q^n$.
$h_1 := x'_0 \cdot h \in \mathbb{Z}_q$.
If $\beta = 1$
  $\mathbf{h}_0 \leftarrow\!\!\$\ \mathbb{Z}_q^n$, $h_1 \leftarrow\!\!\$\ \mathbb{Z}_q$.
Return $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$.

Procedure $\textsc{Eval}(\mathsf{m})$:
If $\mathsf{m} \in \mathsf{prefix}(\mathsf{m}^*)$, Return $\perp$.
$\mathcal{QEVA} := \mathcal{QEVA} \cup \{\mathsf{m}\}$.
$([\mathbf{t}]_2, [u]_2) \leftarrow\!\!\$\ \mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m})$.
$\boxed{([\mathbf{t}]_2, [u]_2) \leftarrow \mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m}).}$
$\mathbf{S} \leftarrow\!\!\$\ \mathbb{Z}_q^{n' \times \mu}$, $\mathbf{T} := \mathbf{B} \cdot \mathbf{S} \in \mathbb{Z}_q^{n \times \mu}$.
$\mathbf{u}^\top := \sum_{i=0}^{l(\mathsf{m})} f_i(\mathsf{m}) \cdot \mathbf{x}_i^\top \cdot \mathbf{T} \in \mathbb{Z}_q^{1 \times \mu}$.
For $i \in [l(\mathsf{m})+1, \ell]$,
  $[d_i]_2 := [\mathbf{x}_i^\top \cdot \mathbf{t}]_2$. $[\mathbf{d}_i]_2 := [\mathbf{x}_i^\top \cdot \mathbf{T}]_2$.
Return $\big([\mathbf{t}]_2, [u]_2, [\mathbf{T}]_2, [\mathbf{u}^\top]_2,$
  $\{[d_i]_2, [\mathbf{d}_i]_2\}_{i \in [l(\mathsf{m})+1, \ell]}\big)$.

Procedure $\textsc{Finalize}(\beta')$:
Return $(\beta' = \beta)$.

Fig. 2 Games APR-CMA and weak − APR − CMA for defining securities of MAC. The shadowed parts appear only in the description of game APR-CMA, while the framed parts appear only in the game weak-APR-CMA

- The secret key $\mathsf{sk}_{\mathsf{MAC}}$ also contains a key $\mathsf{k}_{\mathsf{PRF}}$ for some pseudorandom function $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times \mathcal{M} \longrightarrow \mathbb{Z}_q^{n'}$, where $\mathcal{M}$ is the message space and n' is the column dimension of $\mathbf{B} \in \mathbb{Z}_q^{n \times n'}$ in $\mathsf{sk}_{\mathsf{MAC}}$.
- $\mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m})$ is a deterministic algorithm. For a message $\mathsf{m} \in \mathcal{M}$, it computes a tag $([\mathbf{t}]_2, [u]_2) \in \mathbb{G}_2^n \times \mathbb{G}_2$ as

$$\mathbf{s} := \mathsf{PRF}(\mathsf{k}_{\mathsf{PRF}}, \mathsf{m}) \in \mathbb{Z}_q^{n'}, \quad \mathbf{t} := \mathbf{B} \cdot \mathbf{s} \in \mathbb{Z}_q^n, \quad u := \sum_{i=0}^{l(\mathsf{m})} f_i(\mathsf{m}) \cdot \mathbf{x}_i^\top \cdot \mathbf{t} + x'_0 \in \mathbb{Z}_q,$$

  i.e., $\mathbf{s}$ is the pseudorandom value of message m under PRF.

Definition 9 (*Weak APR-CMA security*) A de-randomized delegatable affine MAC over $\mathbb{Z}_q^n$ is weakly APR-CMA secure, if for any PPT adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{\textit{weak-apr-cma}}_{\mathsf{MAC},\mathcal{A}}(\lambda) := |\Pr[\textsf{weak-APR-CMA}^{\mathcal{A}} \Rightarrow 1] - 1/2|$ is negligible in λ, where game weak-APR-CMA is specified in Fig. 2.

3.2 A de-randomized delegatable affine MAC from NR-PRF

Let $\mathsf{PRF} : \mathcal{K}_{\mathsf{PRF}} \times (\{0,1\}^\lambda)^{\leq L} \longrightarrow \mathbb{Z}_q^k$ be a pseudorandom function. Let $\mathcal{D}_k$ be a matrix distribution that outputs matrices $\mathbf{A} \in \mathbb{Z}_q^{(k+1) \times k}$. Our NR-PRF-based de-randomized delegatable affine MAC over $\mathbb{Z}_q^k$, $\mathsf{MAC}_{\mathsf{NR}}[\mathcal{D}_k]$, is defined in Fig. 3. The message space is $\mathcal{M} = (\{0,1\}^\lambda)^{\leq L}$ for the base set $\mathcal{B} = \{0,1\}^\lambda$. For messages $\mathsf{m} \in \mathcal{M}$, the bit-length $|\mathsf{m}|$ is a multiple of λ, and denote by $p(\mathsf{m})$ the number of blocks in m, i.e., $p(\mathsf{m}) = |\mathsf{m}|/\lambda$. We express messages $\mathsf{m} \in \mathcal{M}$ as $\mathsf{m} = (\mathsf{m}_1, \ldots, \mathsf{m}_{p(\mathsf{m})}) \in (\{0,1\}^\lambda)^{p(\mathsf{m})}$, where $\mathsf{m}_i \in \{0,1\}^\lambda$ is the ith block. We also express each block $\mathsf{m}_i = (\mathsf{m}_{i,1}, \ldots, \mathsf{m}_{i,\lambda})$ as a bit string of length λ with $\mathsf{m}_{i,j} \in \{0,1\}$.

It is easy to check that our $\mathsf{MAC}_{\mathsf{NR}}[\mathcal{D}_k]$ is a de-randomized delegatable affine MAC. Renaming $\mathbf{x}_{i,j}^{(b)}$ to $\mathbf{x}_{2((i-1)\cdot\lambda+j)+b}$, we have that $n = n' = k$, $\ell = 2L\lambda + 1$, $f_0(\mathsf{m}) = f_1(\mathsf{m}) =$

$\underline{\mathsf{sk}_{\mathsf{MAC}} \leftarrow\!\!\$\ \mathsf{MAC.Gen}(\mathsf{params}):}$
$\mathsf{k}_{\mathsf{PRF}} \leftarrow\!\!\$\ \mathcal{K}_{\mathsf{PRF}}$.
$\mathbf{A} \leftarrow\!\!\$\ \mathcal{D}_k$, $\mathbf{B} := \overline{\mathbf{A}} \in \mathbb{Z}_q^{k\times k}$.
For $i \in [L]$, $j \in [\lambda]$
$\quad \mathbf{x}_{i,j}^{(0)}, \mathbf{x}_{i,j}^{(1)} \leftarrow\!\!\$\ \mathbb{Z}_q^k$.
$x'_0 \leftarrow\!\!\$\ \mathbb{Z}_q$.
Return $\mathsf{sk}_{\mathsf{MAC}} := \big(\mathsf{k}_{\mathsf{PRF}}, \mathbf{B}, \{\mathbf{x}_{i,j}^{(0)}, \mathbf{x}_{i,j}^{(1)}\}_{i\in[L],j\in[\lambda]}, x'_0\big)$.

$\underline{([\mathbf{t}]_2, [u]_2) \leftarrow \mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m}):}$
Parse $\mathsf{m} = (\mathsf{m}_{i,j})_{i\in[p(\mathsf{m})],j\in[\lambda]} \in \mathcal{M}$.
$\mathbf{s} := \mathsf{PRF}(\mathsf{k}_{\mathsf{PRF}}, \mathsf{m}) \in \mathbb{Z}_q^k$, $\mathbf{t} := \mathbf{B}\cdot\mathbf{s} \in \mathbb{Z}_q^k$.
$u := \sum_{i=1}^{p(\mathsf{m})} \sum_{j=1}^{\lambda} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \mathbf{t} + x'_0 \in \mathbb{Z}_q$.
Return $([\mathbf{t}]_2, [u]_2) \in \mathbb{G}_2^k \times \mathbb{G}_2$.

$\underline{0/1 \leftarrow \mathsf{Vrfy}\big(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{m}, ([\mathbf{t}]_2, [u]_2)\big):}$
Parse $\mathsf{m} = (\mathsf{m}_{i,j})_{i\in[p(\mathsf{m})],j\in[\lambda]} \in \mathcal{M}$.
If $[u]_2 = \left[\sum_{i=1}^{p(\mathsf{m})} \sum_{j=1}^{\lambda} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \mathbf{t} + x'_0\right]_2$, Return 1.
Else, Return 0.

Fig. 3 Construction of $\mathsf{MAC}_{\mathsf{NR}}[\mathcal{D}_k]$

0, $f_{2((i-1)\cdot\lambda+j)+b}(\mathsf{m}) = (\mathsf{m}_{i,j} = b)$ for $i \in [p(\mathsf{m})]$, $j \in [\lambda]$, $f_{2((i-1)\cdot\lambda+j)+b}(\mathsf{m}) = 0$ for $i \in [p(\mathsf{m})+1, L]$, $j \in [\lambda]$, and $l(\mathsf{m}) = 2p(\mathsf{m})\lambda + 1$.

Our construction bears a superficial resemblance to the NR-PRF-based one proposed in [6, Section 3.2], but with two essential differences.

- Our base set of message space is $\mathcal{B} = \{0,1\}^\lambda$ while in [6] the base set is $\mathcal{B} = \{0,1\}$. In our MAC, if the bit-length of message m is not a multiple of λ, m will be regarded as an invalid message and algorithms Tag and Vrfy will output $\perp$ immediately.
- Our Tag algorithm is deterministic, which always uses the same pseudorandom value $\mathbf{s}$ (hence the same $\mathbf{t}$) for the same message m, while the Tag algorithm of their proposal is a randomized one, which samples a fresh randomness $\mathbf{s}$ each time.

Theorem 1 *If* PRF *is a pseudorandom function and the* $\mathcal{D}_k$*-MDDH assumption holds w.r.t.* PGGen *for* $\mathbb{G}_2$*, then the de-randomized delegatable affine* $\mathsf{MAC}_{\mathsf{NR}}[\mathcal{D}_k]$ *over* $\mathbb{Z}_q^k$ *in Fig. 3 is weakly APR-CMA secure.*

More precisely, suppose that $\mathcal{A}$ *is a PPT adversary against the weak APR-CMA security of* $\mathsf{MAC}_{\mathsf{NR}}[\mathcal{D}_k]$*, that makes at most* Q *times of* EVAL *queries, then there exists a PPT adversary* $\mathcal{B}_{prf}$ *against the pseudorandomness of* PRF *and a PPT adversary* $\mathcal{B}_{mddh}$ *against the* $\mathcal{D}_k$*-MDDH assumption, such that*

$$\mathsf{Adv}^{weak\text{-}apr\text{-}cma}_{\mathsf{MAC}_{\mathsf{NR}}[\mathcal{D}_k],\mathcal{A}}(\lambda) \le \mathsf{Adv}^{prf}_{\mathsf{PRF},\mathcal{B}_{prf}}(\lambda) + 8L\lambda \cdot \left(\mathsf{Adv}^{\mathcal{D}_k\text{-}mddh}_{\mathsf{PGGen},\mathbb{G}_2,\mathcal{B}_{mddh}}(\lambda) + 1/q\right) + Q/2^\lambda.$$

Note that there are constructions of pseudorandom functions with an (almost) tight security reduction to generic primitives such as pseudorandom generators (i.e., the tree-based GGM-construction) [18] and specific assumptions such as the DDH assumption [14,29]. Thus according to Theorem 1, the weak APR-CMA security of our de-randomized delegatable affine MAC enjoys an (almost) tight security reduction.

Proof of Theorem 1 The proof proceeds with a sequence of games illustrated in Fig. 4. Let us first fix some notations. For a bit string $\mathsf{x} \in \{0,1\}^{\le L\lambda}$, we represent the length $|\mathsf{x}|$ uniquely as $|\mathsf{x}| = (i_\mathsf{x} - 1)\cdot\lambda + j_\mathsf{x}$ for integers $1 \le i_\mathsf{x} \le L$ and $1 \le j_\mathsf{x} \le \lambda$, and parse x block-wisely as $\mathsf{x}_1||\cdots||\mathsf{x}_{i_\mathsf{x}-1}||(\mathsf{x}_{i_\mathsf{x},j})_{j\in[j_\mathsf{x}]}$, where $\mathsf{x}_1, \ldots, \mathsf{x}_{i_\mathsf{x}-1} \in \{0,1\}^\lambda$ are the first $i_\mathsf{x}-1$ (completed) blocks of length λ and $(\mathsf{x}_{i_\mathsf{x},j})_{j\in[j_\mathsf{x}]} \in \{0,1\}^{j_\mathsf{x}}$ denote the i_xth block. For integers $\zeta \in [1, L]$ and $\eta \in [0, \lambda]$, if $|\mathsf{x}| > (\zeta-1)\lambda + \eta$, let $\mathsf{x}_{|\zeta,\eta}$ denote the first $(\zeta-1)$ blocks (of length λ) concatenating the first η bits of the ζth block, i.e., $\mathsf{x}_{|\zeta,\eta} := \mathsf{x}_1||\cdots||\mathsf{x}_{\zeta-1}||(\mathsf{x}_{\zeta,j})_{j\in[\eta]} \in \{0,1\}^{(\zeta-1)\lambda+\eta}$; otherwise, let $\mathsf{x}_{|\zeta,\eta}$ denote x itself. Note that $\mathsf{x}_{|\zeta,\lambda} = \mathsf{x}_{|\zeta+1,0}$.

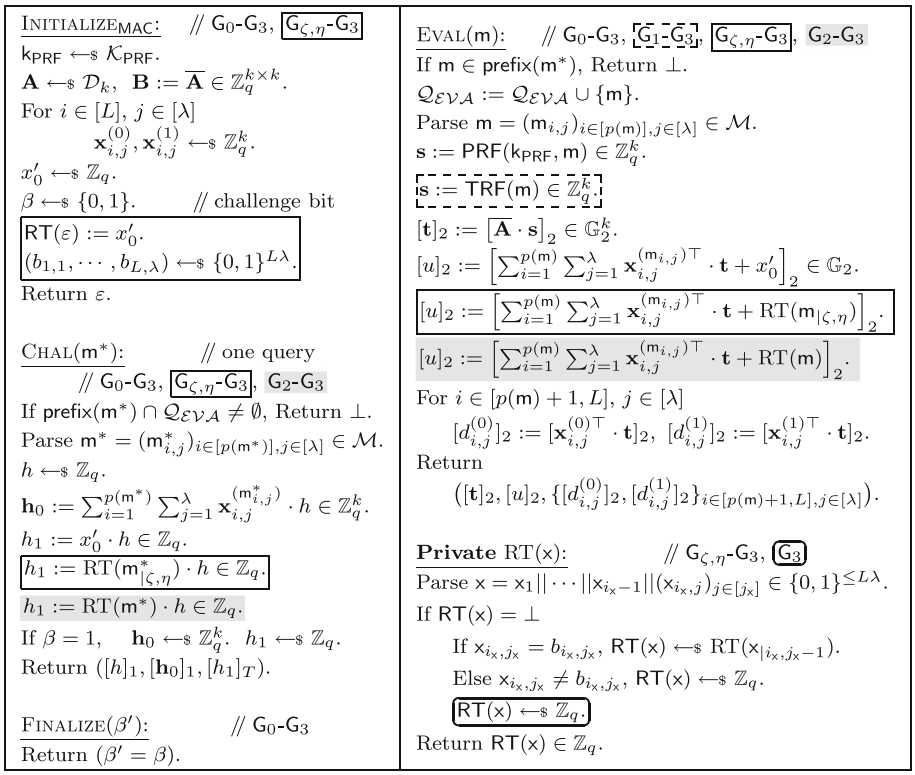

Fig. 4 Games G_0, G_1, $\{\mathsf{G}_{\zeta,\eta}\}_{\zeta\in[1,L],\eta\in[0,\lambda]}$, G_2, G_3 for the proof of Theorem 1

- Game G_0: This is the original weak-APR-CMA game. Let Win denote the event that $\beta' = \beta$. Then by definition, $\mathsf{Adv}^{weak\text{-}apr\text{-}cma}_{\mathsf{MAC}_{\mathsf{NR}}[\mathcal{D}_k],\mathcal{A}}(\lambda) = \left|\Pr_0[\mathsf{Win}] - \frac{1}{2}\right|$.
- Game G_1: It is the same as game G_0, except that, when answering EVAL(m), the challenger uses a truly random function $\mathsf{TRF} : \mathcal{M} \longrightarrow \mathbb{Z}_q^k$ to compute $\mathbf{s}$ with $\mathbf{s} := \mathsf{TRF}(\mathsf{m})$. Any difference between G_0 and G_1 results in a PPT adversary $\mathcal{B}_{pr\,f}$ breaking the pseudorandomness of PRF, i.e.,

$$\left|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\right| \leq \mathsf{Adv}^{pr\,f}_{\mathsf{PRF},\mathcal{B}_{prf}}(\lambda).$$

- Game $\mathsf{G}_{\zeta,\eta}$, $\zeta \in [1, L]$, $\eta \in [0, \lambda]$: This game is the same as game G_1, except that, the challenger chooses random $(b_{1,1}, \ldots, b_{L,\lambda}) \leftarrow\$ \{0,1\}^{L\lambda}$ and sets $\mathsf{RT}(\varepsilon) := x'_0$ beforehand (in INITIALIZE$_{\mathsf{MAC}}$). In addition, the challenger will implement a "random" table $\mathsf{RT} : \{0,1\}^{\leq L\lambda} \longrightarrow \mathbb{Z}_q$ recursively, via the private procedure RT($\cdot$) shown in Fig. 4. The "random" table RT has the special property: for any $\mathsf{x} = \mathsf{x}_1||\cdots||\mathsf{x}_{i_\mathsf{x}-1}||(\mathsf{x}_{i_\mathsf{x},j})_{j\in[j_\mathsf{x}]} \in \{0,1\}^{\leq L\lambda}$,

$$\mathsf{RT}(\mathsf{x}) = \begin{cases} \mathsf{RT}(\mathsf{x}_{|i_\mathsf{x},j_\mathsf{x}-1}), & \text{if } \mathsf{x}_{i_\mathsf{x},j_\mathsf{x}} = b_{i_\mathsf{x},j_\mathsf{x}} \\ \text{random element}, & \text{if } \mathsf{x}_{i_\mathsf{x},j_\mathsf{x}} = 1 - b_{i_\mathsf{x},j_\mathsf{x}} \end{cases}, \tag{6}$$

where $\mathsf{x}_{|i_\mathsf{x},j_\mathsf{x}-1}$ is the first $|\mathsf{x}| - 1$ bits of x and $\mathsf{x}_{i_\mathsf{x},j_\mathsf{x}}$ is the last bit of x.

Now the challenger uses $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta})$ instead of x'_0 to compute $[u]_2$ in EVAL(m) and uses $\mathsf{RT}(\mathsf{m}^*_{|\zeta,\eta})$ instead of x'_0 to compute h_1 in CHAL(m^*), where $\mathsf{m}_{|\zeta,\eta}$ denotes the first $(\zeta-1)\lambda + \eta$ bits of m if $|\mathsf{m}| > (\zeta-1)\lambda + \eta$ and denotes m itself otherwise:

$$[u]_2 := \left[\sum_{i=1}^{p(\mathsf{m})}\sum_{j=1}^{\lambda}\mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top}\cdot\mathbf{t}+\mathsf{RT}(\mathsf{m}_{|\zeta,\eta})\right]_2,\quad h_1 := \mathsf{RT}(\mathsf{m}^*_{|\zeta,\eta})\cdot h.$$

In game $\mathsf{G}_{1,0}$, the challenger will use $\mathsf{RT}(\mathsf{m}_{|1,0}) = \mathsf{RT}(\mathsf{m}^*_{|1,0}) = \mathsf{RT}(\varepsilon) = x'_0$ to compute $[u]_2$ in $\textsc{Eval}(\mathsf{m})$ and compute h_1 in $\textsc{Chal}(\mathsf{m}^*)$. Thus $\mathsf{G}_{1,0}$ is identical to G_1, and we have $\Pr_{1,0}[\mathsf{Win}] = \Pr_1[\mathsf{Win}]$.

For any $\zeta \in [L]$ and $\eta \in [\lambda]$, the only difference between games $\mathsf{G}_{\zeta,\eta-1}$ and $\mathsf{G}_{\zeta,\eta}$ is that $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1})$ is used in $\mathsf{G}_{\zeta,\eta-1}$ while $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta})$ is used in $\mathsf{G}_{\zeta,\eta}$. Note that $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1})$ and $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta})$ are the same when $|\mathsf{m}| < (\zeta-1)\lambda+\eta$ or $\mathsf{m}_{\zeta,\eta} = b_{\zeta,\eta}$, while they are independent when $|\mathsf{m}| \geq (\zeta-1)\lambda+\eta$ and $\mathsf{m}_{\zeta,\eta} = 1-b_{\zeta,\eta}$. By the $\mathcal{D}_k$-MDDH assumption, we will show that it is infeasible for the adversary to notice whether $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1})$ is used or $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta})$ is used via the following lemma. The reduction can be done by embedding the $\mathcal{D}_k$-MDDH instance into $\mathbf{x}_{\zeta,\eta}^{(1-b_{\zeta,\eta})^\top}$ in the secret key (this trick is similar to that in [6]). Its proof is provided in Appendix 2.

Lemma 3 *For any $\zeta \in [L]$ and $\eta \in [\lambda]$, there exist PPT adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ against the Q-fold $\mathcal{D}_k$-MDDH assumption, such that*

$$\left|\Pr_{\zeta,\eta-1}[\mathsf{Win}] - \Pr_{\zeta,\eta}[\mathsf{Win}]\right| \leq 4\cdot\left(\mathsf{Adv}^{Q,\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_2,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{Q,\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_2,\mathcal{B}_2}(\lambda)\right).$$

For any $\zeta \in [L-1]$, the same value $\mathsf{RT}(\mathsf{m}_{|\zeta,\lambda}) = \mathsf{RT}(\mathsf{m}_{|\zeta+1,0})$ is used in both games $\mathsf{G}_{\zeta,\lambda}$ and $\mathsf{G}_{\zeta+1,0}$, since $\mathsf{m}_{|\zeta,\lambda} = \mathsf{m}_{|\zeta+1,0}$ for any m. Thus $\mathsf{G}_{\zeta,\lambda}$ and $\mathsf{G}_{\zeta+1,0}$ are essentially the same, and $\Pr_{\zeta,\lambda}[\mathsf{Win}] = \Pr_{\zeta+1,0}[\mathsf{Win}]$.

– Game G_2: This game is identical to game $\mathsf{G}_{L,\lambda}$, thus $\Pr_2[\mathsf{Win}] = \Pr_{L,\lambda}[\mathsf{Win}]$. More precisely, the challenger will use $\mathsf{RT}(\mathsf{m}_{|L,\lambda}) = \mathsf{RT}(\mathsf{m})$ to compute $[u]_2$ in $\textsc{Eval}(\mathsf{m})$ and use $\mathsf{RT}(\mathsf{m}^*_{|L,\lambda}) = \mathsf{RT}(\mathsf{m}^*)$ to compute h_1 in $\textsc{Chal}(\mathsf{m}^*)$:

$$[u]_2 := \left[\sum_{i=1}^{p(\mathsf{m})}\sum_{j=1}^{\lambda}\mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top}\cdot\mathbf{t}+\mathsf{RT}(\mathsf{m})\right]_2,\quad h_1 := \mathsf{RT}(\mathsf{m}^*)\cdot h.$$

– Game G_3: It is identical to game G_2, except that the challenger implements the table RT as a truly random table without the special property (6). That is, as long as $\mathsf{x} \neq \mathsf{x}'$, $\mathsf{RT}(\mathsf{x})$ and $\mathsf{RT}(\mathsf{x}')$ are independently distributed.

Denote by Hit the event that $\mathcal{A}$ ever queries two messages m, m' to $\textsc{Eval}$ or $\textsc{Chal}$, such that $|\mathsf{m}| < |\mathsf{m}'|$ and $\mathsf{m}' = \mathsf{m}||b_{p(\mathsf{m})+1,1}||\cdots||b_{p(\mathsf{m})+1,\lambda}||\cdots||b_{p(\mathsf{m}'),1}||\cdots||b_{p(\mathsf{m}'),\lambda}$. If the event Hit does not occur, then the table RT implemented in game G_2 also behaves like a truly random function. Thus G_2 and G_3 are identical from the point of view of $\mathcal{A}$ until Hit occurs, and it holds that

$$\left|\Pr_2[\mathsf{Win}] - \Pr_3[\mathsf{Win}]\right| \leq \Pr_2[\mathsf{Hit}] = \Pr_3[\mathsf{Hit}].$$

We give upper bounds on $\Pr_3[\mathsf{Hit}]$ and $\Pr_3[\mathsf{Win}]$ via the following claims.

Claim 1 $\Pr_3[\mathsf{Hit}] \leq Q/2^\lambda$.

Proof of Claim 1 Note that in G_3, RT is implemented as a truly random table, and the challenger never uses $(b_{1,1},\ldots,b_{L,\lambda})$ to answer the queries. If the event Hit occurs, i.e., $\mathcal{A}$ queries two messages m, m' such that $|\mathsf{m}| < |\mathsf{m}'|$ and $\mathsf{m}' = \mathsf{m}||b_{p(\mathsf{m})+1,1}||\cdots||b_{p(\mathsf{m}'),\lambda}$, it implies that $\mathcal{A}$ guesses the values of $(b_{p(\mathsf{m})+1,1},\ldots,b_{p(\mathsf{m}'),\lambda})$ correctly, which can happen with probability $1/2^{|\mathsf{m}'|-|\mathsf{m}|} \leq 1/2^\lambda$. Thus by the union bound, $\Pr_3[\mathsf{Hit}] \leq Q/2^\lambda$. □

Claim 2 $\Pr_3[\mathsf{Win}] = 1/2$.

Proof of Claim 2 In game G_3, to answer $\textsc{Eval}(\mathsf{m})$, the challenger computes $[u]_2$ and $\{[d_{i,j}^{(0)}]_2, [d_{i,j}^{(1)}]_2\}_{i\in[p(\mathsf{m})+1,L],j\in[\lambda]}$ as follows:

$$u := \sum_{i=1}^{p(\mathsf{m})}\sum_{j=1}^{\lambda} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top}\cdot\mathbf{t} + \mathsf{RT}(\mathsf{m}),\quad d_{i,j}^{(0)} = \mathbf{x}_{i,j}^{(0)\top}\cdot\mathbf{t},\quad d_{i,j}^{(1)} = \mathbf{x}_{i,j}^{(1)\top}\cdot\mathbf{t},$$

where $\mathbf{t} = \overline{\mathbf{A}}\,\mathbf{s}$ for $\mathbf{s} = \mathsf{TRF}(\mathsf{m})$. Since the term $\sum_{i=1}^{p(\mathsf{m})}\sum_{j=1}^{\lambda}\mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top}\cdot\mathbf{t}$ is completely determined by m, it is totally hidden by $\mathsf{RT}(\mathsf{m})$, which is implemented as a truly random function. Also $\{[d_{i,j}^{(0)}]_2, [d_{i,j}^{(1)}]_2\}$ does not involve $(\mathbf{x}_{1,j}^{(0)}, \mathbf{x}_{1,j}^{(1)})_{j\in[\lambda]}$. Thus the $\textsc{Eval}$ oracle leaks no information about $(\mathbf{x}_{1,j}^{(0)}, \mathbf{x}_{1,j}^{(1)})_{j\in[\lambda]}$ at all. Then in $\textsc{Chal}(\mathsf{m}^*)$, if $\beta = 0$:

$$\mathbf{h}_0 := \sum_{j=1}^{\lambda}\mathbf{x}_{1,j}^{(\mathsf{m}^*_{1,j})}\cdot h + \sum_{i=2}^{p(\mathsf{m}^*)}\sum_{j=1}^{\lambda}\mathbf{x}_{i,j}^{(\mathsf{m}^*_{i,j})}\cdot h,\quad h_1 := \mathsf{RT}(\mathsf{m}^*)\cdot h.$$

$\mathbf{h}_0$ is uniformly distributed due to the randomness of $\sum_{j=1}^{\lambda}\mathbf{x}_{1,j}^{(\mathsf{m}^*_{1,j})}$, and h_1 is uniformly distributed due to the fresh randomness of $\mathsf{RT}(\mathsf{m}^*)$. Therefore, $\mathbf{h}_0$ and h_1 are uniformly random no matter $\beta = 0$ or $\beta = 1$, then $\mathcal{A}$ can guess β with probability $1/2$, i.e., $\Pr_3[\mathsf{Win}] = 1/2$. □

Taking all things together and by Lemma 1 (Random Self-Reducibility),

$$\begin{aligned}
&\mathsf{Adv}^{weak-apr-cma}_{\mathsf{MAC}_{\mathsf{NR}}[\mathcal{D}_k],\mathcal{A}}(\lambda)\\
&\le \mathsf{Adv}^{prf}_{\mathsf{PRF},\mathcal{B}_{prf}}(\lambda) + 4L\lambda\cdot\left(\mathsf{Adv}^{Q,\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_2,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{Q,\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_2,\mathcal{B}_2}(\lambda)\right) + Q/2^\lambda\\
&\le \mathsf{Adv}^{prf}_{\mathsf{PRF},\mathcal{B}_{prf}}(\lambda) + 8L\lambda\cdot\left(\mathsf{Adv}^{\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_2,\mathcal{B}_{mddh}}(\lambda) + 1/q\right) + Q/2^\lambda,
\end{aligned}$$

thus Theorem 1 follows. □

Remark We stress that the property of de-randomization (cf. Definition 8) plays an essential role in the proof of Claim 2. If we do not stipulate the property of de-randomization and employ a fresh randomness $\mathbf{t}$ to compute $[u]_2$ each time, we can hardly prove Claim 2. The reason is as follows.

If a fresh randomness $\mathbf{t}$ is used each time, then in the oracle $\textsc{Eval}(\mathsf{m})$,

$$u = \sum_{i=1}^{p(\mathsf{m})}\sum_{j=1}^{\lambda}\mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top}\cdot\mathbf{t} + \mathsf{RT}(\mathsf{m}),$$

the term $\sum_{i=1}^{p(\mathsf{m})}\sum_{j=1}^{\lambda}\mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top}\cdot\mathbf{t}$ will vary according to $\mathbf{t}$. In this case, we cannot expect to use the same (fixed) randomness $\mathsf{RT}(\mathsf{m})$ to hide many different terms $\left(\sum_{i=1}^{p(\mathsf{m})}\sum_{j=1}^{\lambda}\mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top}\cdot\mathbf{t}\right)_{\mathbf{t}}$. To demonstrate the problem clearly, suppose that the adversary $\mathcal{A}$ queries the oracle $\textsc{Eval}$ with the same m twice, we denote the two responses by $([\mathbf{t}]_2, [u]_2, \ldots)$ and $([\mathbf{t}']_2, [u']_2, \cdots)$ respectively, where

$$u = \sum_{i=1}^{p(\mathsf{m})}\sum_{j=1}^{\lambda}\mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top}\cdot\mathbf{t} + \mathsf{RT}(\mathsf{m}),\qquad u' = \sum_{i=1}^{p(\mathsf{m})}\sum_{j=1}^{\lambda}\mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top}\cdot\mathbf{t}' + \mathsf{RT}(\mathsf{m}).$$

Then $\mathcal{A}$ can compute $([\Delta\mathbf{t}]_2, [\Delta u]_2)$, where $\Delta\mathbf{t} = \mathbf{t} - \mathbf{t}'$ and $\Delta u = u - u'$, from $([\mathbf{t}]_2, [u]_2)$ and $([\mathbf{t}']_2, [u']_2)$. By this trick, $\mathcal{A}$ knows that $([\Delta\mathbf{t}]_2, [\Delta u]_2)$ satisfies

$$\Delta u = \sum_{i=1}^{p(\mathsf{m})} \sum_{j=1}^{\lambda} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \Delta \mathbf{t}, \tag{7}$$

which gets rid of the mask of $\mathsf{RT}(\mathsf{m})$ successfully.

Consequently, as the adversary $\mathcal{A}$ collects many different pairs $([\Delta \mathbf{t}]_2, [\Delta u]_2)$ of form (7), some information about $(\mathbf{x}_{1,j}^{(0)}, \mathbf{x}_{1,j}^{(1)})_{j\in[\lambda]}$ might be leaked through the oracle EVAL. As a result, $\sum_{j=1}^{\lambda} \mathbf{x}_{1,j}^{(\mathsf{m}^*_{1,j})}$ cannot be used to randomize $\mathbf{h}_0$ in $\textsc{Chal}(\mathsf{m}^*)$ any more, and it is hard to claim the pseudorandomness of $\mathbf{h}_0$.

4 IBE from de-randomized delegatable affine MAC and chameleon hashing

We present a paradigm for constructing PR-ID-CCA2 secure IBE from De-randomized delegatable affine MAC and Chameleon hashing. Our paradigm is tightness preserving.

4.1 Identity-based encryption and its PR-ID-CCA2 security

Definition 10 (*Identity-based encryption*) An identity-based encryption (IBE) scheme $\mathsf{IBE} = (\mathsf{Gen}, \mathsf{USKGen}, \mathsf{Enc}, \mathsf{Dec})$ consists of a tuple of PPT algorithms:

- The key generation algorithm $\mathsf{Gen}(1^\lambda)$ outputs a pair of public key pk and master secret key msk. We assume that pk publicly defines an identity space $\mathcal{ID}$, a message space $\mathcal{M}$ and a ciphertext space $\mathcal{C}$.
- The user secret key generation algorithm $\mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$ takes as input a master secret key msk and an identity $\mathsf{id} \in \mathcal{ID}$, and outputs a user secret key $\mathsf{usk}[\mathsf{id}]$ for id.
- The encryption algorithm $\mathsf{Enc}(\mathsf{pk}, \mathsf{id}, m)$ takes as input a public key pk, an identity $\mathsf{id} \in \mathcal{ID}$ and a message $m \in \mathcal{M}$, and outputs a ciphertext $\mathsf{ct} \in \mathcal{C}$.
- The decryption algorithm $\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{ct})$ takes as input a user secret key $\mathsf{usk}[\mathsf{id}]$, an identity $\mathsf{id} \in \mathcal{ID}$ and a ciphertext $\mathsf{ct} \in \mathcal{C}$, and outputs a message $m \in \mathcal{M}$ or the reject symbol $\perp$.

Correctness of IBE requires that, for all $\lambda \in \mathbb{N}$, all $(\mathsf{pk}, \mathsf{msk}) \leftarrow_\$ \mathsf{Gen}(1^\lambda)$, all $\mathsf{id} \in \mathcal{ID}$, all $m \in \mathcal{M}$, all $\mathsf{ct} \leftarrow_\$ \mathsf{Enc}(\mathsf{pk}, \mathsf{id}, m)$ and all $\mathsf{usk}[\mathsf{id}] \leftarrow_\$ \mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$, it holds that $\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{ct}) = m$.

The traditional security requirements for IBE are indistinguishability and recipient-anonymity against adaptively chosen-identity and chosen-ciphertext attacks (IND-ID-CCA2 and ANON-ID-CCA2). Here we define a stronger security notion according to [2,6], namely, *ciphertext pseudorandomness against adaptively chosen-identity and chosen-ciphertext attacks* (PR-ID-CCA2). PR-ID-CCA2 trivially implies IND-ID-CCA2 and ANON-ID-CCA2.

Definition 11 (*PR-ID-CCA2 security for IBE*) An identity-based encryption scheme IBE is PR-ID-CCA2 secure, if for any PPT adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{pr\text{-}id\text{-}cca2}_{\mathsf{IBE},\mathcal{A}}(\lambda) := |\Pr[\mathsf{PR\text{-}ID\text{-}CCA2}^{\mathcal{A}} \Rightarrow 1] - 1/2|$ is negligible in λ, where game $\mathsf{PR\text{-}ID\text{-}CCA2}$ is specified in Fig. 5.

4.2 IBE from de-randomized delegatable affine MAC and chameleon hashing

Let $\mathcal{D}_k$ be a matrix distribution that outputs matrices $\mathbf{A} \in \mathbb{Z}_q^{(k+1)\times k}$. Let $\mathsf{MAC} = (\mathsf{MAC.Gen}, \mathsf{Tag}, \mathsf{Vrfy})$ be a de-randomized delegatable affine MAC over $\mathbb{Z}_q^n$ with mes-

Procedure INITIALIZE: $(\mathsf{pk}, \mathsf{msk}) \leftarrow\!\$\ \mathsf{Gen}(1^\lambda)$. $\beta \leftarrow\!\$\ \{0,1\}$. // challenge bit Return pk. **Procedure** ENC(id^*, m^*): // one query If $\mathsf{id}^* \in \mathcal{Q}_{\mathcal{USK}}$, Return $\perp$. $\mathsf{ct}^* \leftarrow\!\$\ \mathsf{Enc}(\mathsf{pk}, \mathsf{id}^*, m^*)$. If $\beta = 1$, $\mathsf{ct}^* \leftarrow\!\$\ \mathcal{C}$. Return ct^*.	**Procedure** USKGEN(id): If $\mathsf{id} = \mathsf{id}^*$, Return $\perp$. $\mathcal{Q}_{\mathcal{USK}} := \mathcal{Q}_{\mathcal{USK}} \cup \{\mathsf{id}\}$. Return $\mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$. **Procedure** DEC$(\mathsf{id}, \mathsf{ct})$: If $(\mathsf{id}, \mathsf{ct}) = (\mathsf{id}^*, \mathsf{ct}^*)$, Return $\perp$. $\mathsf{usk}[\mathsf{id}] \leftarrow\!\$\ \mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$. Return $\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{ct})$. **Procedure** FINALIZE$(\beta')$: Return $(\beta' = \beta)$.

Fig. 5 PR-ID-CCA2 security game for IBE

sage space $\mathcal{ID} \cup \mathcal{ID}^2$ (the base set is $\mathcal{B} = \mathcal{ID}$). Let $\mathsf{CH} = (\mathsf{CH.Gen}, \mathsf{Eval}, \mathsf{Equiv})$ be a chameleon hashing. Let $\mathsf{H} : \mathbb{G}_T \longrightarrow \{0,1\}^{3\lambda}$ be a hash function. The proposed $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k] = (\mathsf{Gen}, \mathsf{USKGen}, \mathsf{Enc}, \mathsf{Dec})$ with identity space $\mathcal{ID}$ and message space $\mathcal{M} = \{0,1\}^\lambda$ is defined in Fig. 6.

Our construction can be viewed as a combination of a (PR-ID-CCCA2[4] secure) IBKEM and a (one-time secure) authenticated encryption scheme.

To show the correctness of our IBE, we denote the output of $\mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$ by $\mathsf{usk}[\mathsf{id}] = ([\mathbf{t}]_2, [u]_2, [\mathbf{v}]_2, \{[d_i]_2, [\mathbf{e}_i]_2\}_{i \in [l(\mathsf{id})+1, \ell]})$, then in $\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \langle \mathsf{C}, \chi \rangle)$,

$$
\begin{aligned}
[u']_2 &= \Big[\underbrace{\sum_{i=0}^{l(\mathsf{id})} f_i(\mathsf{id}) \mathbf{x}_i^\top \mathbf{t} + x'_0}_{u} + \sum_{i=l(\mathsf{id})+1}^{l(\mathsf{id}|\mathsf{id}')} f_i(\mathsf{id}|\mathsf{id}') \cdot \underbrace{\mathbf{x}_i^\top \mathbf{t}}_{d_i} \Big]_2 \\
&= \left[\sum_{i=0}^{l(\mathsf{id}|\mathsf{id}')} f_i(\mathsf{id}|\mathsf{id}') \mathbf{x}_i^\top \mathbf{t} + x'_0 \right]_2, \\
[\mathbf{v}']_2 &= \Big[\underbrace{\sum_{i=0}^{l(\mathsf{id})} f_i(\mathsf{id}) \mathbf{Y}_i \mathbf{t} + \mathbf{y}'_0}_{\mathbf{v}} + \sum_{i=l(\mathsf{id})+1}^{l(\mathsf{id}|\mathsf{id}')} f_i(\mathsf{id}|\mathsf{id}') \cdot \underbrace{\mathbf{Y}_i \mathbf{t}}_{\mathbf{e}_i} \Big]_2 \\
&= \left[\sum_{i=0}^{l(\mathsf{id}|\mathsf{id}')} f_i(\mathsf{id}|\mathsf{id}') \mathbf{Y}_i \mathbf{t} + \mathbf{y}'_0 \right]_2, \qquad (8) \\
\mathsf{K} &= \left[(\mathbf{v}'^\top \mid u') \cdot \mathbf{c}_0 \right]_T / \left[\mathbf{t}^\top \cdot \mathbf{c}_1 \right]_T \\
&= \left[\mathbf{t}^\top \cdot \underbrace{\left(\sum_{i=0}^{l(\mathsf{id}|\mathsf{id}')} f_i(\mathsf{id}|\mathsf{id}') \cdot (\mathbf{Y}_i^\top \mid \mathbf{x}_i) \cdot \mathbf{c}_0 - \mathbf{c}_1 \right)}_{(*)} + (\mathbf{y}_0'^\top \mid x'_0) \cdot \mathbf{c}_0 \right]_T. \qquad (9)
\end{aligned}
$$

If $\langle \mathsf{C} = ([\mathbf{c}_0]_1, [\mathbf{c}_1]_1, R_{\mathsf{CH}}), \chi = (\chi_1, \chi_2) \rangle$ is an output of $\mathsf{Enc}(\mathsf{pk}, \mathsf{id}, m)$, then $(*) = 0$, and $\mathsf{K} = \left[(\mathbf{y}_0'^\top \mid x'_0) \cdot \mathbf{c}_0 \right]_T = \left[\mathbf{z}'_0 \cdot \mathbf{r} \right]_T$. Therefore it will derive the same $(k_1, k_2, k_3) := \mathsf{H}(\mathsf{K})$, and the correctness follows.

[4] CCCA2 is short for Constrained CCA2 security, which is a weaken security notion proposed by [23]. PR-ID-CCCA2 secure IBKEM can be converted to PR-ID-CCA2 secure IBE if combined with a one-time secure authenticated encryption.

$\underline{(\mathsf{pk}, \mathsf{msk}) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)}$:
$(ek_{\mathsf{CH}}, td_{\mathsf{CH}}) \leftarrow\!\!\$\ \mathsf{CH.Gen}(1^\lambda)$.
$\mathbf{A} \leftarrow\!\!\$\ \mathcal{D}_k$, with $\mathbf{A} \in \mathbb{Z}_q^{(k+1)\times k}$.
$\mathsf{sk}_{\mathsf{MAC}} \leftarrow\!\!\$\ \mathsf{MAC.Gen}(\mathsf{params})$,
where $\mathsf{sk}_{\mathsf{MAC}} = (\mathsf{k}_{\mathsf{PRF}}, \mathbf{B}, \mathbf{x}_0, \cdots, \mathbf{x}_\ell, x'_0)$.
For $i \in [0, \ell]$,
$\mathbf{Y}_i \leftarrow\!\!\$\ \mathbb{Z}_q^{k\times n}$, $\mathbf{Z}_i := (\mathbf{Y}_i^\top \mid \mathbf{x}_i)\mathbf{A} \in \mathbb{Z}_q^{n\times k}$.
$\mathbf{y}'_0 \leftarrow\!\!\$\ \mathbb{Z}_q^k$, $\mathbf{z}'_0 := (\mathbf{y}_0'^\top \mid x'_0)\cdot\mathbf{A} \in \mathbb{Z}_q^{1\times k}$.
$\mathsf{pk} := \big(ek_{\mathsf{CH}}, \mathcal{PG}, [\mathbf{A}]_1, \{[\mathbf{Z}_i]_1\}_{i\in[0,\ell]}, [\mathbf{z}'_0]_1\big)$.
$\mathsf{msk} := \big(\mathsf{sk}_{\mathsf{MAC}}, \{\mathbf{Y}_i\}_{i\in[0,\ell]}, \mathbf{y}'_0\big)$.
Return $(\mathsf{pk}, \mathsf{msk})$.

$\underline{\langle \mathsf{C}, \chi\rangle \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, \mathsf{id}, m)}$: // $m \in \{0,1\}^\lambda$
$\mathbf{r} \leftarrow\!\!\$\ \mathbb{Z}_q^k$. $[\mathbf{c}_0]_1 := [\mathbf{A}\cdot\mathbf{r}]_1 \in \mathbb{G}_1^{k+1}$.
$R_{\mathsf{CH}} \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{CH}}$.
$\mathsf{id}' := \mathsf{Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1; R_{\mathsf{CH}}) \in \mathcal{ID}$.
$[\mathbf{c}_1]_1 := \Big[\sum_{i=0}^{l(\mathsf{id}|\mathsf{id}')} f_i(\mathsf{id}|\mathsf{id}')\cdot\mathbf{Z}_i\cdot\mathbf{r}\Big]_1 \in \mathbb{G}_1^n$.
$\mathsf{K} := \big[\mathbf{z}'_0\cdot\mathbf{r}\big]_T \in \mathbb{G}_T$.
$(k_1, k_2, k_3) := \mathsf{H}(\mathsf{K}) \in \{0,1\}^{3\lambda}$.
$\chi_1 := k_1 + m$, $\chi_2 := k_2\cdot\chi_1 + k_3 \in \{0,1\}^\lambda$,
where the operations "+" and "·" are in the field $GF(2^\lambda)$.
Return
$\mathsf{C} := ([\mathbf{c}_0]_1, [\mathbf{c}_1]_1, R_{\mathsf{CH}})$ and $\chi := (\chi_1, \chi_2)$.

$\underline{\mathsf{usk}[\mathsf{id}] \leftarrow \mathsf{USKGen}(\mathsf{msk}, \mathsf{id})}$:
$([\mathbf{t}]_2, [u]_2) \leftarrow \mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{id})$.
// $\mathbf{t} := \mathbf{B}\cdot\mathbf{s} \in \mathbb{Z}_q^n$ for $\mathbf{s} := \mathsf{PRF}(\mathsf{k}_{\mathsf{PRF}}, \mathsf{id})$,
// $u := \sum_{i=0}^{l(\mathsf{id})} f_i(\mathsf{id})\cdot\mathbf{x}_i^\top\cdot\mathbf{t} + x'_0 \in \mathbb{Z}_q$.
$[\mathbf{v}]_2 := \Big[\sum_{i=0}^{l(\mathsf{id})} f_i(\mathsf{id})\cdot\mathbf{Y}_i\cdot\mathbf{t} + \mathbf{y}'_0\Big]_2 \in \mathbb{G}_2^k$.
For $i \in [l(\mathsf{id})+1, \ell]$,
$[d_i]_2 := [\mathbf{x}_i^\top\cdot\mathbf{t}]_2 \in \mathbb{G}_2$.
$[\mathbf{e}_i]_2 := [\mathbf{Y}_i\cdot\mathbf{t}]_2 \in \mathbb{G}_2^k$.
Return $\mathsf{usk}[\mathsf{id}] :=$
$\big([\mathbf{t}]_2, [u]_2, [\mathbf{v}]_2, \{[d_i]_2, [\mathbf{e}_i]_2\}_{i\in[l(\mathsf{id})+1,\ell]}\big)$.

$\underline{m/\perp \leftarrow \mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \langle \mathsf{C}, \chi\rangle)}$:
Parse $\mathsf{usk}[\mathsf{id}] =$
$\big([\mathbf{t}]_2, [u]_2, [\mathbf{v}]_2, \{[d_i]_2, [\mathbf{e}_i]_2\}_{i\in[l(\mathsf{id})+1,\ell]}\big)$.
Parse $\mathsf{C} = ([\mathbf{c}_0]_1, [\mathbf{c}_1]_1, R_{\mathsf{CH}})$, $\chi = (\chi_1, \chi_2)$.
$\mathsf{id}' := \mathsf{Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1; R_{\mathsf{CH}}) \in \mathcal{ID}$.
// delegation of $[u]_2$ and $[\mathbf{v}]_2$
$[u']_2 := \Big[u + \sum_{i=l(\mathsf{id})+1}^{l(\mathsf{id}|\mathsf{id}')} f_i(\mathsf{id}|\mathsf{id}')d_i\Big]_2 \in \mathbb{G}_2$.
$[\mathbf{v}']_2 := \Big[\mathbf{v} + \sum_{i=l(\mathsf{id})+1}^{l(\mathsf{id}|\mathsf{id}')} f_i(\mathsf{id}|\mathsf{id}')\mathbf{e}_i\Big]_2 \in \mathbb{G}_2^k$.
$\mathsf{K} := \big[(\mathbf{v}'^\top \mid u')\cdot\mathbf{c}_0\big]_T / \big[\mathbf{t}^\top\cdot\mathbf{c}_1\big]_T \in \mathbb{G}_T$.
$(k_1, k_2, k_3) := \mathsf{H}(\mathsf{K}) \in \{0,1\}^{3\lambda}$.
If $\chi_2 = k_2\cdot\chi_1 + k_3$, Return $m := \chi_1 - k_1$.
Else $\chi_2 \neq k_2\cdot\chi_1 + k_3$, Return $\perp$.

Fig. 6 Paradigm for constructing $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k]$

Theorem 2 *If the $\mathcal{D}_k$-MDDH assumption holds w.r.t. PGGen for $\mathbb{G}_1$, the underlying de-randomized delegatable affine MAC over $\mathbb{Z}_q^n$ is weakly APR-CMA secure, CH is a collision-resistant chameleon hashing and H is a universal hash function, then the $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k]$ in Fig. 6 is PR-ID-CCA2 secure.*

More precisely, suppose that $\mathcal{A}$ is a PPT adversary against the PR-ID-CCA2 security of $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k]$, that makes at most Q_d times of Dec *queries, then there exist a PPT adversary $\mathcal{B}_1$ against the $\mathcal{D}_k$-MDDH assumption, a PPT adversary $\mathcal{B}_2$ against the collision resistance property of CH, and PPT adversaries $\mathcal{B}_3$ and $\mathcal{B}_4$ against the weak APR-CMA security of MAC, such that*

$$\mathsf{Adv}^{pr\text{-}id\text{-}cca2}_{\mathsf{IBE}[\mathsf{MAC},\mathsf{CH},\mathcal{D}_k],\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_1,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{cr}_{\mathsf{CH},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{weak\text{-}apr\text{-}cma}_{\mathsf{MAC},\mathcal{B}_3}(\lambda) + \mathsf{Adv}^{weak\text{-}apr\text{-}cma}_{\mathsf{MAC},\mathcal{B}_4}(\lambda) + 4Q_d/2^\lambda + 2Q_d/q^n.$$

Note that there are constructions of chameleon hashing with a tight security reduction to specific assumptions such as the RSA [25], the factoring [26] and the discrete logarithm [26] assumptions. When instantiating the de-randomized delegatable affine MAC with $\mathsf{MAC}_{\mathsf{NR}}[\mathcal{D}_k]$ described in Fig. 3, whose weak APR-CMA security is tightly reduced to the

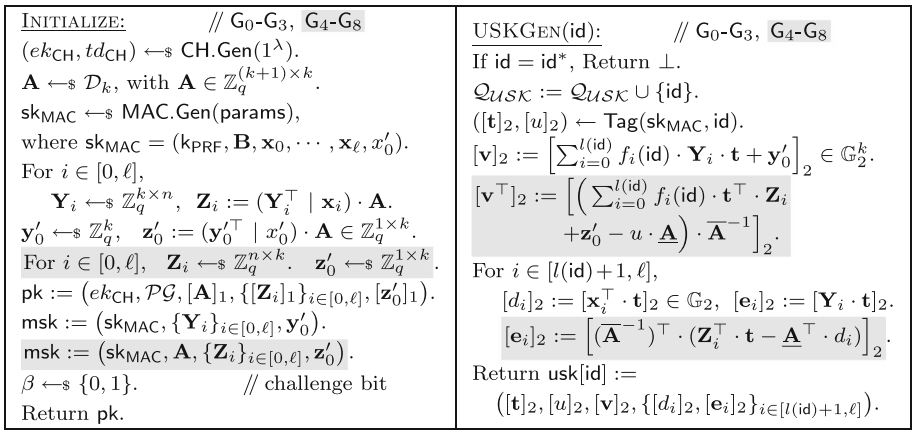

Fig. 7 Games G_0–G_8 for the proof of Theorem 2 (also see Fig. 8)

$\mathcal{D}_k$-MDDH assumption, we immediately obtain the first IBE with tight PR-ID-CCA2 security based on the $\mathcal{D}_k$-MDDH assumption, according to Theorem 2.

Proof of Theorem 2 We prove it through a sequence of games illustrated in Figs. 7 and 8. A rough description of difference between adjacent games is summarized in Table 2. Before presenting the full detailed proof, we first give a high-level description how the PR-ID-CCA2 security of our IBE[MAC, CH, $\mathcal{D}_k$] is reduced to the weak APR-CMA security of the underlying MAC.

(a) In the PR-ID-CCA2 game (cf. Fig. 5), the adversary can make three kinds of oracle queries, namely USKGEN(id), ENC(id*, m^*) and DEC(id, $\langle \mathsf{C}, \chi \rangle$). Our goal is to simulate these oracles by using the oracles $\text{EVAL}_{\mathsf{MAC}}$, $\text{CHAL}_{\mathsf{MAC}}$[5] in the weak-APR-CMA game of MAC (cf. Fig. 2), instead of using $\mathsf{sk}_{\mathsf{MAC}} = (\mathsf{k}_{\mathsf{PRF}}, \mathbf{B}, \mathbf{x}_0, \ldots, \mathbf{x}_\ell, \mathbf{x}'_0)$ directly.

(b) To achieve this goal, we use a "tuned" Groth-Sahai NIZK proof [20] technique, similar to the intuition in [6]. Loosely speaking, $\mathbf{Z}_i = (\mathbf{Y}_i^\top \mid \mathbf{x}_i) \cdot \mathbf{A}$ and $\mathbf{z}'_0 = (\mathbf{y}_0'^\top \mid x'_0) \cdot \mathbf{A}$ can be viewed as perfect hiding commitments of $\mathsf{sk}_{\mathsf{MAC}}$. In INITIALIZE, we do not choose $\mathbf{Y}_i$ and $\mathbf{y}'_0$ directly, but choose $\mathbf{Z}_i$ and $\mathbf{z}'_0$ uniformly instead. We will use $\mathbf{A}$, $\mathbf{Z}_i$ and $\mathbf{z}'_0$ (in group $\mathbb{Z}_q$) as trapdoor to simulate the oracles without using $\mathsf{sk}_{\mathsf{MAC}}$. This simulation can be viewed as a perfect simulation.

(c) Using $\mathbf{A}$, $\mathbf{Z}_i$ and $\mathbf{z}'_0$ (in group $\mathbb{Z}_q$) as trapdoor, we can simulate USKGEN(id) ($\mathsf{id} \neq \mathsf{id}^*$) through oracle access to $\text{EVAL}_{\mathsf{MAC}}(\mathsf{id})$, without using $\mathsf{sk}_{\mathsf{MAC}}$.

(d) We change how ENC(id*, m^*) works so that we can embed the challenge $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$, which is the output of $\text{CHAL}_{\mathsf{MAC}}(\mathsf{id}^* | \mathsf{id}'^*)$ in the weak-APR-CMA game of MAC (cf. Fig. 2), into the challenge ciphertext $\big(\mathsf{C}^* = ([\mathbf{c}_0^*]_1, [\mathbf{c}_1^*]_1, R_{\mathsf{CH}}^*), \chi^* = (\chi_1^*, \chi_2^*)\big)$ of the PR-ID-CCA2 game. In addition, we can simulate ENC(id*, m^*) without using $\mathsf{sk}_{\mathsf{MAC}}$. We stress that a subtle problem exists. Recall that $\mathsf{id}'^* = \mathsf{Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1^*; R_{\mathsf{CH}}^*)$, so before submitting $\mathsf{id}^* | \mathsf{id}'^*$ to $\text{CHAL}_{\mathsf{MAC}}$, we need to compute $[\mathbf{c}_0]_1^*$ first. On the other hand, to embed the challenge $[h]_1$ into $[\mathbf{c}_0]_1^*$, we have to compute $[\mathbf{c}_0]_1^*$ after the $\text{CHAL}_{\mathsf{MAC}}$ query. To break this deadlock, we make use of the Equivocation property of chameleon hashing CH.

(e) In contrast to the PR-ID-CPA security proof of IBE in [6], we need to handle the decryption queries. Using $\mathbf{A}$, $\mathbf{Z}_i$ and $\mathbf{z}'_0$ (in group $\mathbb{Z}_q$) as trapdoor, we can simulate DEC(id, $\langle \mathsf{C},$

[5] For clarity, hereafter we use $\mathcal{O}_{\mathsf{MAC}}$ to indicate oracles in the security game of MAC.

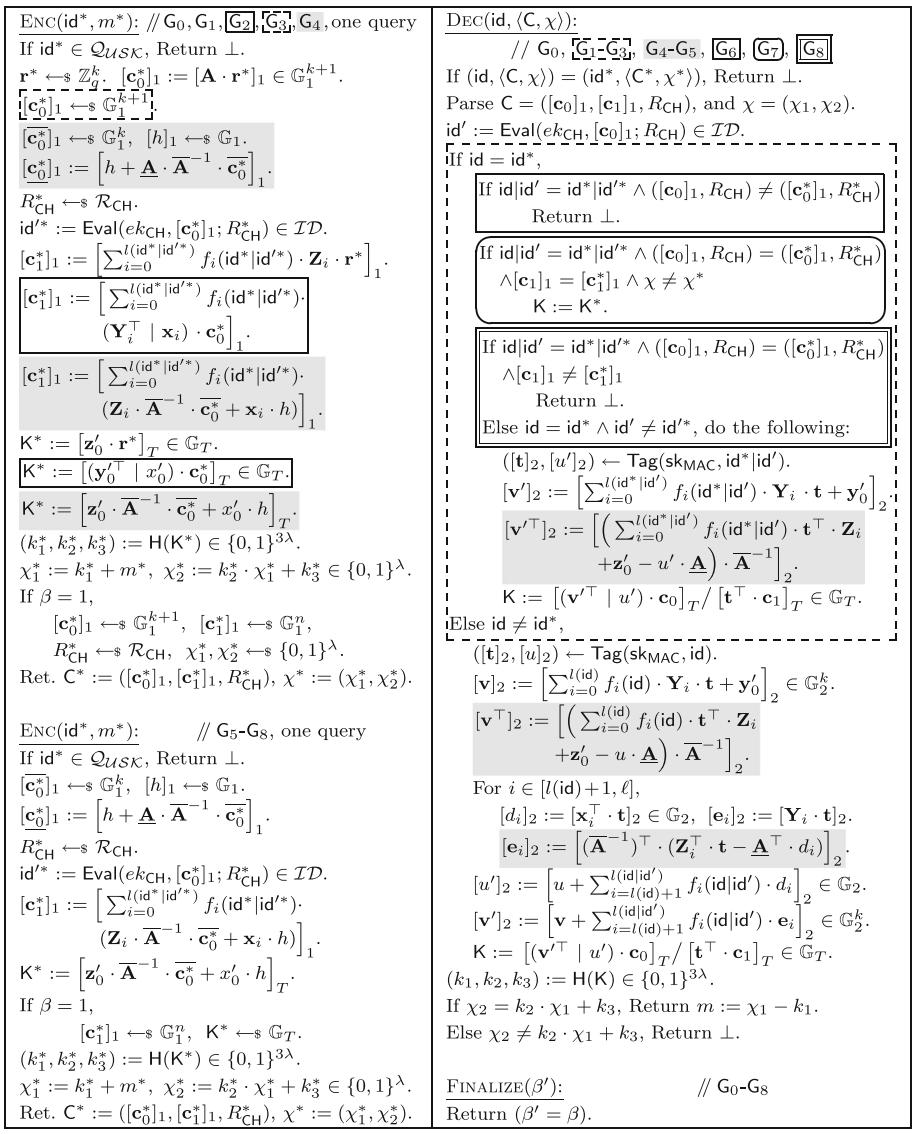

Fig. 8 Games G_0–G_8 for the proof of Theorem 2 (also see Fig. 7)

$\chi\rangle$) through oracle access to $\textsc{Eval}_{\mathsf{MAC}}(\mathsf{id})$ when $\mathsf{id} \neq \mathsf{id}^*$ or access to $\textsc{Eval}_{\mathsf{MAC}}(\mathsf{id}^*|\mathsf{id}')$ when $\mathsf{id} = \mathsf{id}^*$ but $\mathsf{id}' \neq \mathsf{id}'^*$, without using $\mathsf{sk}_{\mathsf{MAC}}$, as long as $\mathsf{id}|\mathsf{id}' \neq \mathsf{id}^*|\mathsf{id}'^*$. In the case of $\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^*$, we are not allowed to invoke $\textsc{Eval}_{\mathsf{MAC}}$ any more. So we have to change how $\textsc{Dec}(\mathsf{id}, \langle \mathsf{C}, \chi\rangle)$ works such that neither $\mathsf{sk}_{\mathsf{MAC}}$ nor $\textsc{Eval}_{\mathsf{MAC}}$ is needed. This is the most difficult part in our proof. We divide the case $\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^*$ into three sub-cases:

– If $\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^* \wedge ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) \neq ([\mathbf{c}_0^*]_1, R_{\mathsf{CH}}^*)$, we expect $\textsc{Dec}$ to return $\perp$, due to the collision-resistance property of the chameleon hashing CH. Recall that $\mathsf{id}' = \mathsf{Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1; R_{\mathsf{CH}})$, $\mathsf{id}'^* = \mathsf{Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1^*; R_{\mathsf{CH}}^*)$.

Table 2 Brief description of the security proof of Theorem 2

	Changes between adjacent games	Assumptions
G_0	The original PR-ID-CCA2 security game	–
G_1	DEC(id, $\langle \mathsf{C}, \chi \rangle$): if $\mathsf{id} = \mathsf{id}^*$, compute $[u']_2$ and $[\mathbf{v}']_2$ from $\mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{id}^* \vert \mathsf{id}')$ directly, instead of that first compute $[u]_2$ and $[\mathbf{v}]_2$ from $\mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{id}^*)$ and then delegate to get $[u']_2$ and $[\mathbf{v}']_2$	$\mathsf{G}_0 \approx_s \mathsf{G}_1$ by entropy of the $\mathbf{t}$ related to id^* and $\mathbf{t}$ related to $\mathsf{id}^* \vert \mathsf{id}'$
G_2	ENC(id*, m^*): use the master secret key msk to compute $[\mathbf{c}_1^*]_1$ and K^* from $[\mathbf{c}_0^*]_1$ directly	$\mathsf{G}_1 = \mathsf{G}_2$
G_3	ENC(id*, m^*): sample $[\mathbf{c}_0^*]_1$ uniformly over $\mathbb{G}_1^k$ instead of computing as $[\mathbf{c}_0^*]_1 = [\mathbf{A} \cdot \mathbf{r}^*]_1$	$\mathsf{G}_2 \approx_c \mathsf{G}_3$ by $\mathcal{D}_k$-MDDH
G_4	INITIALIZE: sample $\mathbf{Z}_i$ and $\mathbf{z}_0'$ uniformly, instead of sampling $\mathbf{Y}_i$ and $\mathbf{y}_0'$ directly USKGEN(id): use $\mathbf{A}$, $\mathbf{Z}_i$ and $\mathbf{z}_0'$ as trapdoor to compute $[\mathbf{v}]_2$ from $[u]_2$ and compute $[\mathbf{e}_i]_2$ from $[d_i]_2$ DEC(id, $\langle \mathsf{C}, \chi \rangle$): use $\mathbf{A}$, $\mathbf{Z}_i$ and $\mathbf{z}_0'$ as trapdoor to compute $[\mathbf{v}']_2$ from $[u']_2$ ENC(id*, m^*): compute $[\mathbf{c}_0^*]_1$ with $[\mathbf{c}_0^*]_1 = \left[h + \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*}\right]_1$ Now $[\mathbf{c}_1^*]_1 =$ $\left[\sum f_i(\mathsf{id}^* \vert \mathsf{id}'^*) \cdot \mathbf{Z}_i \overline{\mathbf{A}}^{-1} \overline{\mathbf{c}_0^*} + \underbrace{\sum f_i(\mathsf{id}^* \vert \mathsf{id}'^*) \mathbf{x}_i h}_{\text{(I)}}\right]_1$, and $\mathsf{K}^* = \left[\mathbf{z}_0' \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + \overbrace{x_0' \cdot h}^{\text{(II)}}\right]_T$ in the case of $\beta = 0$	$\mathsf{G}_3 = \mathsf{G}_4$ since it essentially is a perfect simulation
G_5	ENC(id*, m^*): in the case of $\beta = 1$, sample K^* uniformly and compute χ_1^*, χ_2^* from K^*, instead of sampling χ_1^*, χ_2^* directly	$\mathsf{G}_4 = \mathsf{G}_5$ since H is universal
G_6	DEC(id, $\langle \mathsf{C}, \chi \rangle$): return $\perp$ if $\mathsf{id} \vert \mathsf{id}' = \mathsf{id}^* \vert \mathsf{id}'^* \wedge ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) \neq ([\mathbf{c}_0^*]_1, R_{\mathsf{CH}}^*)$	$\mathsf{G}_5 \approx_c \mathsf{G}_6$ since CH is coll.-resist
G_7	DEC(id, $\langle \mathsf{C}, \chi \rangle$): set $\mathsf{K} := \mathsf{K}^*$ if $\mathsf{id} \vert \mathsf{id}' = \mathsf{id}^* \vert \mathsf{id}'^* \wedge ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) = ([\mathbf{c}_0^*]_1, R_{\mathsf{CH}}^*) \wedge [\mathbf{c}_1]_1 = [\mathbf{c}_1^*]_1 \wedge \chi \neq \chi^*$	$\mathsf{G}_6 = \mathsf{G}_7$ if $\beta = 0$, $\mathsf{G}_6 \approx_s \mathsf{G}_7$ if $\beta = 1$
G_8	DEC(id, $\langle \mathsf{C}, \chi \rangle$): return $\perp$ if $\mathsf{id} \vert \mathsf{id}' = \mathsf{id}^* \vert \mathsf{id}'^* \wedge ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) = ([\mathbf{c}_0^*]_1, R_{\mathsf{CH}}^*) \wedge [\mathbf{c}_1]_1 \neq [\mathbf{c}_1^*]_1$ A weak APR-CMA adversary of MAC can simulate G_8 with $\mathcal{A}$ perfectly by embedding $[\mathbf{h}_0]_1$ to (I) and embedding $[h_1]_T$ to (II)	$\mathsf{G}_7 \approx_c \mathsf{G}_8$ by weak APR-CMA security of MAC

– If $\mathsf{id} \vert \mathsf{id}' = \mathsf{id}^* \vert \mathsf{id}'^* \wedge ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) = ([\mathbf{c}_0^*]_1, R_{\mathsf{CH}}^*) \wedge [\mathbf{c}_1]_1 = [\mathbf{c}_1^*]_1 \wedge \chi \neq \chi^*$, we change DEC to use K^* directly to decrypt $\chi = (\chi_1, \chi_2)$. This is trivially correct when $([\mathbf{c}_0^*]_1, [\mathbf{c}_1^*]_1, R_{\mathsf{CH}}^*)$ is indeed an encapsulation of K^*, i.e., when $\beta = 0$. However, the situation of $\beta = 1$ (where the challenge ciphertext is randomly chosen) is more subtle. Nevertheless, we use an information-theoretic argument to show that this change is still undetectable. Roughly speaking, given χ^*, the remaining entropy of K^* is large enough to make the decryption of χ failure. Using the real key encapsulated in

$([\mathbf{c}_0^*]_1, [\mathbf{c}_1^*]_1, R_{\mathsf{CH}}^*)$ has the same effect as K^*, since the real key can be proved to be randomly distributed conditioned on $([\mathbf{c}_0^*]_1, [\mathbf{c}_1^*]_1, R_{\mathsf{CH}}^*)$. (See Lemma 4 for details.)

- If $\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^* \wedge ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) = ([\mathbf{c}_0^*]_1, R_{\mathsf{CH}}^*) \wedge [\mathbf{c}_1]_1 \neq [\mathbf{c}_1^*]_1$, we expect that Dec will return $\perp$. We prove this by another time of reduction to the weak APR-CMA security of MAC. (This proof is highly non-trivial. See Lemma 5 for details.) The essential idea is: if $[\mathbf{c}_1]_1$ from the adversary's query hits some specific value, we develop a new computational argument and reduce the "hit" event to a successful weak APR-CMA attack to MAC. Otherwise, $[\mathbf{c}_1]_1$ does not hit the value, then we give an information-theoretical analysis that Dec will return $\perp$ except with negligible probability.

(f) Consequently, we can simulate the game using the oracles $\text{Eval}_{\mathsf{MAC}}$, $\text{Chal}_{\mathsf{MAC}}$ in the weak-APR-CMA game of MAC, instead of using $\mathsf{sk}_{\mathsf{MAC}}$ directly. Then the weak APR-CMA security of MAC implies the PR-ID-CCA2 security of $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k]$.

- Game G_0: This is the original PR-ID-CCA2 game. Let Win denote the event that $\beta' = \beta$. Then by definition, $\mathsf{Adv}^{pr\text{-}id\text{-}cca2}_{\mathsf{IBE}[\mathsf{MAC},\mathsf{CH},\mathcal{D}_k],\mathcal{A}}(\lambda) = \left|\Pr_0[\mathsf{Win}] - \frac{1}{2}\right|$.
- Game G_1: This game is the same as game G_0, except that, when answering Dec query $\big(\mathsf{id}, \langle \mathsf{C} = ([\mathbf{c}_0]_1, [\mathbf{c}_1]_1, R_{\mathsf{CH}}), \chi = (\chi_1, \chi_2)\rangle\big)$ with $\mathsf{id} = \mathsf{id}^*$, the challenger changes the way it computes $[u']_2$ and $[\mathbf{v}']_2$ as follows:
  - compute $\mathsf{id}' := \mathsf{Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1; R_{\mathsf{CH}}) \in \mathcal{ID}$,
  - invoke $([\mathbf{t}]_2, [u']_2) \leftarrow \mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{id}^*|\mathsf{id}')$ directly using the secret key, where $[u']_2 = \left[\sum_{i=0}^{l(\mathsf{id}^*|\mathsf{id}')} f_i(\mathsf{id}^*|\mathsf{id}') \cdot \mathbf{x}_i^\top \cdot \mathbf{t} + x_0'\right]_2$,
  - compute $[\mathbf{v}']_2 := \left[\sum_{i=0}^{l(\mathsf{id}^*|\mathsf{id}')} f_i(\mathsf{id}^*|\mathsf{id}') \cdot \mathbf{Y}_i \cdot \mathbf{t} + \mathbf{y}_0'\right]_2$.

  According to Eq. (8), $[u']_2$ and $[\mathbf{v}']_2$ are the same functions of $\mathbf{t}$ in G_0 and G_1. Thus the only difference between G_0 and G_1 is the distribution of $\mathbf{t}$ itself. In game G_0, $\mathbf{t}$ is generated via $([\mathbf{t}]_2, [u]_2) \leftarrow \mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{id}^*)$, while in game G_1, it is generated via $([\mathbf{t}]_2, [u']_2) \leftarrow \mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{id}^*|\mathsf{id}')$. Similar to Eq. (9), we have

$$\begin{aligned}\mathsf{K} &= \left[(\mathbf{v}'^\top \mid u') \cdot \mathbf{c}_0\right]_T \Big/ \left[\mathbf{t}^\top \cdot \mathbf{c}_1\right]_T \\ &\overset{(9)}{=} \Big[\mathbf{t}^\top \cdot \underbrace{\Big(\sum_{i=0}^{l(\mathsf{id}^*|\mathsf{id}')} f_i(\mathsf{id}^*|\mathsf{id}') \cdot (\mathbf{Y}_i^\top \mid \mathbf{x}_i) \cdot \mathbf{c}_0 - \mathbf{c}_1\Big)}_{(*)} + (\mathbf{y}_0'^\top \mid x_0') \cdot \mathbf{c}_0\Big]_T.\end{aligned}$$

  If the Dec query satisfies $(*) = 0$, then the challenger will answer the decryption query with $\mathsf{K} = \left[(\mathbf{y}_0'^\top \mid x_0') \cdot \mathbf{c}_0\right]_T$, which is the same both in games G_0 and G_1. In this case, the $\mathbf{t}$ related to id^* or the $\mathbf{t}$ related to $\mathsf{id}^*|\mathsf{id}'$ is not used at all. If the Dec query satisfies $(*) \neq 0$, note that the challenger never uses the value of the $\mathbf{t}$ related to id^* or the $\mathbf{t}$ related to $\mathsf{id}^*|\mathsf{id}'$ in other procedures, thus $\mathsf{K} = [\mathbf{t}^\top \cdot (*) + (\mathbf{y}_0'^\top \mid x_0') \cdot \mathbf{c}_0]_T$ will be uniformly distributed over $\mathbb{G}_T$ from the point of view of $\mathcal{A}$, due to the randomness of $\mathbf{t}$,[6] both in games G_0 and G_1. Then in the following steps of Dec, by the Leftover Hash Lemma, $(k_1, k_2, k_3) := \mathsf{H}(\mathsf{K}) \in \{0,1\}^{3\lambda}$ is statistically close to the uniform distribution, thus $\chi_2 \neq k_2 \cdot \chi_1 + k_3$ holds except with a negligible probability $2^{-\lambda}$. In this case, Dec outputs $\perp$ both in games G_0 and G_1, and in addition, it does not leak the value of the $\mathbf{t}$

[6] Actually, $\mathbf{t}$ is only pseudorandom here, since $\mathbf{t} = \mathbf{B}\,\mathbf{s}$ and $\mathbf{s} = \mathsf{PRF}(\mathsf{k}_{\mathsf{PRF}}, \mathsf{id}^*)$ (or $\mathbf{s} = \mathsf{PRF}(\mathsf{k}_{\mathsf{PRF}}, \mathsf{id}^*|\mathsf{id}')$). Nevertheless, it is easy to add a game so that $\mathbf{s} = \mathsf{TRF}(\mathsf{id}^*)$ (or $\mathbf{s} = \mathsf{TRF}(\mathsf{id}^*|\mathsf{id}')$), where TRF is a truly random function. For the sake of simplicity, we forgo making this explicit in our proof.

related to id^* or the $\mathbf{t}$ related to $\mathsf{id}^*|\mathsf{id}'$ to $\mathcal{A}$. By the union bound, G_0 and G_1 are essentially the same except with probability $Q_d/2^\lambda$, i.e., $\left|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\right| \leq Q_d/2^\lambda$.

– Game G_2: This game is the same as game G_1, except that, when answering $\textsc{Enc}(\mathsf{id}^*, m^*)$, the challenger uses the master secret key $\mathsf{msk} = \left(\mathsf{sk}_{\mathsf{MAC}}, \{\mathbf{Y}_i\}_{i\in[0,\ell]}, \mathbf{y}'_0\right) = (\mathsf{k}_{\mathsf{PRF}}, \mathbf{B}, \{\mathbf{x}_i\}_{i\in[0,\ell]}, x'_0, \{\mathbf{Y}_i\}_{i\in[0,\ell]}, \mathbf{y}'_0)$ to compute $[\mathbf{c}^*_1]_1$ and K^* as follows:
  - compute $[\mathbf{c}^*_1]_1 := \left[\sum_{i=0}^{l(\mathsf{id}^*|\mathsf{id}'^*)} f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot (\mathbf{Y}_i^\top \mid \mathbf{x}_i) \cdot \mathbf{c}^*_0\right]_1$,
  - compute $\mathsf{K}^* := \left[(\mathbf{y}_0'^\top \mid x'_0) \cdot \mathbf{c}^*_0\right]_T$.

  Observe that

$$[\mathbf{c}^*_1]_1 \overset{\mathsf{G}_1}{=} \left[\sum_{i=0}^{l(\mathsf{id}^*|\mathsf{id}'^*)} f_i(\mathsf{id}^*|\mathsf{id}'^*)\mathbf{Z}_i\mathbf{r}^*\right]_1 = \left[\sum_{i=0}^{l(\mathsf{id}^*|\mathsf{id}'^*)} f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot (\mathbf{Y}_i^\top \mid \mathbf{x}_i)\mathbf{A} \cdot \mathbf{r}^*\right]_1$$
$$\overset{\mathsf{G}_2}{=} \left[\sum_{i=0}^{l(\mathsf{id}^*|\mathsf{id}'^*)} f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot (\mathbf{Y}_i^\top \mid \mathbf{x}_i) \cdot \mathbf{c}^*_0\right]_1,$$
$$\mathsf{K}^* \overset{\mathsf{G}_1}{=} \left[\mathbf{z}'_0 \cdot \mathbf{r}^*\right]_T = \left[(\mathbf{y}_0'^\top \mid x'_0) \cdot \mathbf{A} \cdot \mathbf{r}^*\right]_T \overset{\mathsf{G}_2}{=} \left[(\mathbf{y}_0'^\top \mid x'_0) \cdot \mathbf{c}^*_0\right]_T.$$

  Thus G_2 is identical to G_1, and $\Pr_1[\mathsf{Win}] = \Pr_2[\mathsf{Win}]$.

– Game G_3: This game is the same as game G_2, except that, when answering $\textsc{Enc}(\mathsf{id}^*, m^*)$, the challenger samples $[\mathbf{c}^*_0]_1$ uniformly from $\mathbb{G}_1^{k+1}$, instead of computing $[\mathbf{c}^*_0]_1 = [\mathbf{A}\cdot\mathbf{r}^*]_1$ with $\mathbf{r}^* \leftarrow_\$ \mathbb{Z}_q^k$.
  The only difference between G_2 and G_3 is the computation of $[\mathbf{c}^*_0]_1$ in $\textsc{Enc}$. In game G_2, the joint distribution of $(\mathcal{PG}, [\mathbf{A}]_1, [\mathbf{c}^*_0]_1)$ is identical to the real $\mathcal{D}_k$-MDDH distribution, while in game G_3, it is identical to the random $\mathcal{D}_k$-MDDH distribution. It is straightforward to construct a PPT adversary $\mathcal{B}_1$ against the $\mathcal{D}_k$-MDDH assumption with respect to PGGen for $\mathbb{G}_1$. Note that it is enough for $\mathcal{B}_1$ to use $[\mathbf{A}]_1$ (instead of $\mathbf{A}$ in $\mathbb{Z}_q$) to perfectly simulate G_2 or G_3 for $\mathcal{A}$. Thus $\left|\Pr_2[\mathsf{Win}] - \Pr_3[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_1,\mathcal{B}_1}(\lambda)$.

– Game G_4: This game is the same as game G_3, except that, in $\textsc{Initialize}$, the challenger does not choose $\mathbf{Y}_i$ and $\mathbf{y}'_0$ directly, but chooses $\mathbf{Z}_i$ and $\mathbf{z}'_0$ uniformly instead, and regards them as part of the master secret key. By $\mathbf{Z}_i = (\mathbf{Y}_i^\top \mid \mathbf{x}_i) \cdot \mathbf{A}$ and $\mathbf{z}'_0 = (\mathbf{y}_0'^\top \mid x'_0) \cdot \mathbf{A}$, we have

$$\mathbf{Y}_i^\top = (\mathbf{Z}_i - \mathbf{x}_i \cdot \underline{\mathbf{A}}) \cdot \overline{\mathbf{A}}^{-1} \text{ and } \mathbf{y}_0'^\top = (\mathbf{z}'_0 - x'_0 \cdot \underline{\mathbf{A}}) \cdot \overline{\mathbf{A}}^{-1}. \tag{10}$$

  Consequently, procedures $\textsc{USKGen}$, $\textsc{Dec}$ and $\textsc{Enc}$ now can proceed by using $(\mathbf{Z}_i, \mathbf{z}'_0)$ via (10) instead of using $(\mathbf{Y}_i, \mathbf{y}'_0)$ directly. More precisely, to answer $\textsc{USKGen}(\mathsf{id})$ and answer $\textsc{Dec}(\mathsf{id}, \langle \mathsf{C}, \chi\rangle)$ with $\mathsf{id} \neq \mathsf{id}^*$, the challenger computes $[\mathbf{v}^\top]_2$ and $[\mathbf{e}_i]_2$ as follows:

$$\begin{aligned}
[\mathbf{v}^\top]_2 &\overset{\mathsf{G}_3}{=} \left[\sum f_i(\mathsf{id}) \cdot \mathbf{t}^\top \cdot \mathbf{Y}_i^\top + \mathbf{y}_0'^\top\right]_2 \\
&= \left[\sum f_i(\mathsf{id}) \cdot \mathbf{t}^\top \cdot (\mathbf{Z}_i - \mathbf{x}_i \cdot \underline{\mathbf{A}}) \cdot \overline{\mathbf{A}}^{-1} + (\mathbf{z}'_0 - x'_0 \cdot \underline{\mathbf{A}}) \cdot \overline{\mathbf{A}}^{-1}\right]_2 \\
&\overset{\mathsf{G}_4}{=} \Bigg[\Bigg(\sum f_i(\mathsf{id}) \cdot \mathbf{t}^\top \cdot \mathbf{Z}_i + \mathbf{z}'_0 - \underbrace{\left(\sum f_i(\mathsf{id}) \cdot \mathbf{t}^\top \cdot \mathbf{x}_i + x'_0\right)}_{u} \cdot \underline{\mathbf{A}}\Bigg) \cdot \overline{\mathbf{A}}^{-1}\Bigg]_2,
\end{aligned}$$

$$[\mathbf{e}_i]_2 \overset{\mathsf{G}_3}{=} [\mathbf{Y}_i \cdot \mathbf{t}]_2 = \left[(\overline{\mathbf{A}}^{-1})^\top(\mathbf{Z}_i^\top - \underline{\mathbf{A}}^\top\mathbf{x}_i^\top)\mathbf{t}\right]_2 \overset{\mathsf{G}_4}{=} \left[(\overline{\mathbf{A}}^{-1})^\top(\mathbf{Z}_i^\top\mathbf{t} - \underline{\mathbf{A}}^\top d_i)\right]_2,$$

and similarly, to answer $\text{DEC}(\mathsf{id}, \langle \mathsf{C}, \chi \rangle)$ with $\mathsf{id} = \mathsf{id}^*$, it computes $[\mathbf{v}'^{\top}]_2$ as follows:

$$[\mathbf{v}'^{\top}]_2 \overset{\mathsf{G}_3}{=} \left[\sum f_i(\mathsf{id}^*|\mathsf{id}')\mathbf{t}^{\top}\mathbf{Y}_i^{\top} + \mathbf{y}_0'^{\top}\right]_2 \overset{\mathsf{G}_4}{=} \left[\left(\sum f_i(\mathsf{id}^*|\mathsf{id}')\mathbf{t}^{\top}\mathbf{Z}_i + \mathbf{z}_0' - u'\underline{\mathbf{A}}\right)\overline{\mathbf{A}}^{-1}\right]_2.$$

As for $\text{ENC}(\mathsf{id}^*, m^*)$, the challenger now computes $[\mathbf{c}_0^*]_1$ in a different way:

$$[\mathbf{c}_0^*]_1 = \begin{pmatrix} [\overline{\mathbf{c}_0^*}]_1 \\ [\underline{\mathbf{c}_0^*}]_1 \end{pmatrix} := \begin{pmatrix} [\overline{\mathbf{c}_0^*}]_1 \\ \left[h + \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*}\right]_1 \end{pmatrix}$$

with random $[\overline{\mathbf{c}_0^*}]_1 \leftarrow\!\$\ \mathbb{G}_1^k$ and $[h]_1 \leftarrow\!\$\ \mathbb{G}_1$. Then $[\mathbf{c}_0^*]_1$ is uniformly random over $\mathbb{G}_1^{k+1}$ as in G_3. Also, the challenger computes $[\mathbf{c}_1^*]_1$ and K^* by applying (10) and by the fact that $\underline{\mathbf{c}_0^*} = h + \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*}$:

$$\begin{aligned}
[\mathbf{c}_1^*]_1 &\overset{\mathsf{G}_3}{=} \left[\sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot (\mathbf{Y}_i^{\top} \mid \mathbf{x}_i)\mathbf{c}_0^*\right]_1 = \left[\sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot (\mathbf{Y}_i^{\top}\overline{\mathbf{c}_0^*} + \mathbf{x}_i\underline{\mathbf{c}_0^*})\right]_1 \\
&= \left[\sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \left((\mathbf{Z}_i - \mathbf{x}_i \cdot \underline{\mathbf{A}}) \cdot \overline{\mathbf{A}}^{-1}\overline{\mathbf{c}_0^*} + \mathbf{x}_i(h + \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*})\right)\right]_1 \\
&\overset{\mathsf{G}_4}{=} \left[\sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot (\mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + \mathbf{x}_i \cdot h)\right]_1,
\end{aligned}$$

and similarly, $\mathsf{K}^* \overset{\mathsf{G}_3}{=} \left[(\mathbf{y}_0'^{\top} \mid x_0') \cdot \mathbf{c}_0^*\right]_T \overset{\mathsf{G}_4}{=} \left[\mathbf{z}_0' \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + x_0' \cdot h\right]_T$.

Thus these changes are conceptual, and G_4 is identical to G_3. Then $\Pr_3[\mathsf{Win}] = \Pr_4[\mathsf{Win}]$.

– Game G_5: This game is the same as game G_4, except that, when answering $\text{ENC}(\mathsf{id}^*, m^*)$, in the case of $\beta = 1$, the challenger does not choose $\chi_1^*, \chi_2^* \leftarrow\!\$\ \{0,1\}^{\lambda}$ directly, but instead, it chooses a random $\mathsf{K}^* \leftarrow\!\$\ \mathbb{G}_T$, computes $(k_1^*, k_2^*, k_3^*) := \mathsf{H}(\mathsf{K}^*) \in \{0,1\}^{3\lambda}$ and sets $\chi_1^* := k_1^* + m^*,\ \chi_2^* := k_2^* \cdot \chi_1^* + k_3^*$.
By the Leftover Hash Lemma, since K^* is uniformly distributed over $\mathbb{G}_T$, (k_1^*, k_2^*, k_3^*) will be statistically close to the uniform distribution over $\{0,1\}^{3\lambda}$. Therefore χ_1^*, χ_2^* are also uniformly random in G_5, the same as in G_4. Then G_5 is identical to G_4, and $\Pr_4[\mathsf{Win}] = \Pr_5[\mathsf{Win}]$.
– Game G_6: This game is the same as game G_5, except that, when answering DEC query $\left(\mathsf{id}, \langle \mathsf{C} = ([\mathbf{c}_0]_1, [\mathbf{c}_1]_1, R_{\mathsf{CH}}), \chi = (\chi_1, \chi_2)\rangle\right)$, the challenger returns $\perp$ directly if the following condition holds

$$\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^* \wedge ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) \neq ([\mathbf{c}_0^*]_1, R_{\mathsf{CH}}^*).$$

Since $\mathsf{id}' = \mathsf{Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1; R_{\mathsf{CH}})$ and $\mathsf{id}'^* = \mathsf{Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1^*; R_{\mathsf{CH}}^*)$, any difference between G_5 and G_6 will imply a CH-collision. Thus $\left|\Pr_5[\mathsf{Win}] - \Pr_6[\mathsf{Win}]\right| \leq \mathsf{Adv}^{cr}_{\mathsf{CH},\mathcal{B}_2}(\lambda)$ for a PPT adversary $\mathcal{B}_2$.
– Game G_7: This game is the same as game G_6, except that, when answering DEC query $\left(\mathsf{id}, \langle \mathsf{C} = ([\mathbf{c}_0]_1, [\mathbf{c}_1]_1, R_{\mathsf{CH}}), \chi = (\chi_1, \chi_2)\rangle\right)$, the challenger sets $\mathsf{K} := \mathsf{K}^*$ directly if the following condition holds

$$\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^* \ \wedge\ ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) = ([\mathbf{c}_0^*]_1, R_{\mathsf{CH}}^*) \ \wedge\ [\mathbf{c}_1]_1 = [\mathbf{c}_1^*]_1 \ \wedge\ \chi \neq \chi^*. \quad (11)$$

We analyze the difference between G_6 and G_7 via the following lemma.

Lemma 4 $\left|\Pr_6[\mathsf{Win}] - \Pr_7[\mathsf{Win}]\right| \leq Q_d/q^n + 2 \cdot Q_d/2^{\lambda}$.

Proof of Lemma 4 If $\mathcal{A}$ submits a DEC query satisfies Condition (11), i.e., submits $\left(\mathsf{id}^*, \langle \mathsf{C}^* = ([\mathbf{c}_0^*]_1, [\mathbf{c}_1^*]_1, R_{\mathsf{CH}}^*), \chi = (\chi_1, \chi_2)\rangle\right)$ with $\chi \neq \chi^*$, then in G_7 the challenger will set $\mathsf{K} := \mathsf{K}^*$, while in G_6 it will compute K as follows:

– invoke $([\mathbf{t}]_2, [u']_2) \leftarrow \mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{id}^*|\mathsf{id}'^*)$, where $\mathbf{t}$ is related to $\mathsf{id}^*|\mathsf{id}'^*$ and $u' = \sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \mathbf{t}^\top \cdot \mathbf{x}_i + x'_0$,
– compute $[\mathbf{v}'^\top]_2 = \left[\left(\sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \mathbf{t}^\top \cdot \mathbf{Z}_i + \mathbf{z}'_0 - u' \cdot \underline{\mathbf{A}}\right) \cdot \overline{\mathbf{A}}^{-1}\right]_2$,
– compute $\mathsf{K} = \left[(\mathbf{v}'^\top \mid u') \cdot \mathbf{c}_0^*\right]_T \big/ \left[\mathbf{t}^\top \cdot \mathbf{c}_1^*\right]_T$.

By the fact that $\underline{\mathbf{c}}_0^* = h + \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*}$, in G_6 we have that

$$
\begin{aligned}
\mathsf{K} &= \left[(\mathbf{v}'^\top \mid u') \cdot \mathbf{c}_0^*\right]_T \Big/ \left[\mathbf{t}^\top \cdot \mathbf{c}_1^*\right]_T = \left[\mathbf{v}'^\top \cdot \overline{\mathbf{c}_0^*} + u' \cdot \underline{\mathbf{c}}_0^* - \mathbf{t}^\top \cdot \mathbf{c}_1^*\right]_T \\
&= \left[\left(\mathbf{t}^\top \sum f_i(\mathsf{id}^*|\mathsf{id}'^*)\mathbf{Z}_i + \mathbf{z}'_0 - u'\underline{\mathbf{A}}\right)\overline{\mathbf{A}}^{-1}\overline{\mathbf{c}_0^*} + u'\left(h + \underline{\mathbf{A}}\overline{\mathbf{A}}^{-1}\overline{\mathbf{c}_0^*}\right) - \mathbf{t}^\top\mathbf{c}_1^*\right]_T \\
&= \left[\left(\mathbf{t}^\top \sum f_i(\mathsf{id}^*|\mathsf{id}'^*)\mathbf{Z}_i + \mathbf{z}'_0\right)\overline{\mathbf{A}}^{-1}\overline{\mathbf{c}_0^*} + u'h - \mathbf{t}^\top\mathbf{c}_1^*\right]_T \\
&= \left[\left(\mathbf{t}^\top \sum f_i(\mathsf{id}^*|\mathsf{id}'^*)\mathbf{Z}_i + \mathbf{z}'_0\right)\overline{\mathbf{A}}^{-1}\overline{\mathbf{c}_0^*} + \left(\mathbf{t}^\top \sum f_i(\mathsf{id}^*|\mathsf{id}'^*)\mathbf{x}_i + x'_0\right)h - \mathbf{t}^\top\mathbf{c}_1^*\right]_T \\
&= \Bigg[\mathbf{t}^\top \cdot \underbrace{\left(\sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \left(\mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + \mathbf{x}_i \cdot h\right) - \mathbf{c}_1^*\right)}_{(**)} + \mathbf{z}'_0\overline{\mathbf{A}}^{-1}\overline{\mathbf{c}_0^*} + x'_0 h\Bigg]_T . \qquad (12)
\end{aligned}
$$

If $\beta = 0$, then in $\textsc{Enc}(\mathsf{id}^*, m^*)$, $\mathbf{c}_1^*$ and K^* are honestly computed:

$$
\mathbf{c}_1^* = \sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot (\mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + \mathbf{x}_i \cdot h), \quad \mathsf{K}^* = \left[\mathbf{z}'_0 \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + x'_0 \cdot h\right]_T .
$$

Consequently, $(**) = 0$ and $\mathsf{K} = \mathsf{K}^*$ in G_6, which is the same as in G_7.

Whereas if $\beta = 1$, then in $\textsc{Enc}(\mathsf{id}^*, m^*)$, $\mathbf{c}_1^*$ and K^* are uniformly chosen. In this case, we analyze the difference between G_6 and G_7 as follows:

- In game G_6, the challenger will compute K according to (12) to carry subsequent computations of $\textsc{Dec}$. Since $\mathbf{c}_1^*$ is randomly chosen in $\textsc{Enc}$, it holds that $(**) \neq 0$ except with probability $1/q^n$. Note that the challenger never leaks to the adversary any information of the $\mathbf{t}$ related to $\mathsf{id}^*|\mathsf{id}'^*$ in other procedures,[7] thus K is uniformly distributed over $\mathbb{G}_T$ due to the randomness of $\mathbf{t}$. Consequently, by the Leftover Hash Lemma, $(k_1, k_2, k_3) = \mathsf{H}(\mathsf{K}) \in \{0,1\}^{3\lambda}$ is statistically close to the uniform distribution, and $\chi_2 \neq k_2 \cdot \chi_1 + k_3$ holds except with probability $2^{-\lambda}$. In this case, $\textsc{Dec}$ outputs $\perp$ in G_6, and in addition, no information of the $\mathbf{t}$ related to $\mathsf{id}^*|\mathsf{id}'^*$ is leaked to $\mathcal{A}$.
- In game G_7, the challenger will set $\mathsf{K} = \mathsf{K}^*$ directly, where K^* is uniformly chosen in $\textsc{Enc}$, compute $(k_1^*, k_2^*, k_3^*) = \mathsf{H}(\mathsf{K}^*) \in \{0,1\}^{3\lambda}$, and output $m := \chi_1 - k_1^*$, if the following condition holds

$$
\chi_2 = k_2^* \cdot \chi_1 + k_3^*. \qquad (13)
$$

  Note that the only information about $(k_1^*, k_2^*, k_3^*) = \mathsf{H}(\mathsf{K}^*)$ leaked to the adversary is contained in $\chi^* = (\chi_1^*, \chi_2^*)$ via $\textsc{Enc}(\mathsf{id}^*, m^*)$, where $\chi_1^* = k_1^* + m^*$ and $\chi_2^* = k_2^* \cdot \chi_1^* + k_3^*$. Since $\mathcal{A}$ submits $\chi = (\chi_1, \chi_2)$ with $(\chi_1, \chi_2) \neq (\chi_1^*, \chi_2^*)$ in our discussion (i.e., Condition

[7] In the analysis of game G_1, we show that no information of the $\mathbf{t}$ related to id^* or the $\mathbf{t}$ related to $\mathsf{id}^*|\mathsf{id}'$ (for any $\mathsf{id}' \in \mathcal{ID}$) is leaked to $\mathcal{A}$ except with negligible probability. As our games move on, $\textsc{Dec}$, which is the only possible oracle that may leak that value, is much more restricted, thus cannot leak information about that $\mathbf{t}$ except with negligible probability.

(11) holds), (13) will not hold except with probability at most $2^{-\lambda}$. In this case, DEC outputs $\perp$ in G_7.

In summary, if $\mathcal{A}$ submits a DEC query satisfies Condition (11), then in the case of $\beta = 0$, both DEC in G_6 and G_7 will set $\mathsf{K} := \mathsf{K}^*$, while in the case of $\beta = 1$, both DEC in G_6 and G_7 will output $\perp$ except with probability at most $1/q^n + 2/2^{\lambda}$. The lemma follows from the union bound. □

– Game G_8: This game is the same as game G_7, except that, when answering DEC query $\big(\mathsf{id}, \langle \mathsf{C} = ([\mathbf{c}_0]_1, [\mathbf{c}_1]_1, R_{\mathsf{CH}}), \chi = (\chi_1, \chi_2)\rangle\big)$, the challenger returns $\perp$ directly if the following condition holds

$$\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^* \ \wedge\ ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) = ([\mathbf{c}_0^*]_1, R^*_{\mathsf{CH}}) \ \wedge\ [\mathbf{c}_1]_1 \neq [\mathbf{c}_1^*]_1. \tag{14}$$

Note that in G_8, for DEC queries satisfying $\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^*$, the challenger can answer them without using the secret key.

We analyze the difference between G_7 and G_8 via the following lemma.

Lemma 5 *There exists a PPT adversary $\mathcal{B}_3$ against the weak APR-CMA security of the de-randomized delegatable affine* MAC*, such that*

$$\big|\Pr_7[\mathsf{Win}] - \Pr_8[\mathsf{Win}]\big| \leq Q_d/2^{\lambda} + \mathsf{Adv}^{weak\text{-}apr\text{-}cma}_{\mathsf{MAC},\mathcal{B}_3}(\lambda) + Q_d/q^n.$$

Proof of Lemma 5 If $\mathcal{A}$ submits a DEC query satisfies Condition (14), i.e., submits $\big(\mathsf{id}^*, \langle \mathsf{C} = ([\mathbf{c}_0^*]_1, [\mathbf{c}_1]_1, R^*_{\mathsf{CH}}), \chi = (\chi_1, \chi_2)\rangle\big)$ with $[\mathbf{c}_1]_1 \neq [\mathbf{c}_1^*]_1$, then in G_8 the challenger will return $\perp$, while in G_7 it will proceed as follows:

– invoke $([\mathbf{t}]_2, [u']_2) \leftarrow \mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{id}^*|\mathsf{id}'^*)$, where $\mathbf{t}$ is related to $\mathsf{id}^*|\mathsf{id}'^*$ and $u' = \sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \mathbf{t}^\top \cdot \mathbf{x}_i + x'_0$,
– compute $[\mathbf{v}'^\top]_2 = \Big[\big(\sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \mathbf{t}^\top \cdot \mathbf{Z}_i + \mathbf{z}'_0 - u' \cdot \underline{\mathbf{A}}\big) \cdot \overline{\mathbf{A}}^{-1}\Big]_2$,
– compute $\mathsf{K} = \big[(\mathbf{v}'^\top \mid u') \cdot \mathbf{c}_0^*\big]_T \,/\, \big[\mathbf{t}^\top \cdot \mathbf{c}_1\big]_T$.

Similar to (12), in G_7 we have that

$$\begin{aligned}\mathsf{K} &= \Big[(\mathbf{v}'^\top \mid u') \cdot \mathbf{c}_0^*\Big]_T \,/\, \Big[\mathbf{t}^\top \cdot \mathbf{c}_1\Big]_T \\ &\overset{(12)}{=} \Big[\mathbf{t}^\top \cdot \underbrace{\Big(\sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \big(\mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + \mathbf{x}_i \cdot h\big) - \mathbf{c}_1\Big)}_{(**)} + \mathbf{z}'_0 \overline{\mathbf{A}}^{-1} \overline{\mathbf{c}_0^*} + x'_0 h\Big]_T.\end{aligned} \tag{15}$$

If $\beta = 0$, then $\mathbf{c}_1^*$ is honestly computed by ENC(id^*, m^*):

$$\mathbf{c}_1^* = \sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot (\mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + \mathbf{x}_i \cdot h).$$

Consequently, $[\mathbf{c}_1]_1 \neq [\mathbf{c}_1^*]_1$ implies that $(**) = \big(\mathbf{c}_1^* - \mathbf{c}_1\big) \neq 0$.

If $\beta = 1$, then in ENC(id^*, m^*), $\mathbf{c}_1^*$ is uniformly chosen from $\mathbb{Z}_q^n$.

Let Hit denote the event that the challenge bit $\beta = 1$ and $\mathcal{A}$ makes a DEC query $\big(\mathsf{id}, \langle \mathsf{C} = ([\mathbf{c}_0]_1, [\mathbf{c}_1]_1, R_{\mathsf{CH}}), \chi = (\chi_1, \chi_2)\rangle\big)$, such that

$$\mathbf{c}_1 = \sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot (\mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + \mathbf{x}_i \cdot h).$$

Then if Hit does not happen, we also have $(**) \neq 0$.

In summary, if Hit does not happen, then $(**) \neq 0$ in G_7. Note that the challenger never leaks to $\mathcal{A}$ any information of the $\mathbf{t}$ related to $\mathsf{id}^*|\mathsf{id}'^*$ in other procedures (cf. Footnote 7), thus K computed by (15) is uniformly distributed over $\mathbb{G}_T$ due to the randomness of $\mathbf{t}$. Consequently, $(k_1, k_2, k_3) := \mathsf{H}(\mathsf{K}) \in \{0,1\}^{3\lambda}$ is statistically close to the uniform distribution, by the Leftover Hash Lemma. It follows that $\chi_2 \neq k_2 \cdot \chi_1 + k_3$ except with probability $2^{-\lambda}$. In this case, DEC outputs $\perp$ in G_7, which is the same as in G_8. In addition, it does not leak the value of the $\mathbf{t}$ related to $\mathsf{id}^*|\mathsf{id}'^*$. Therefore, G_7 and G_8 are the same except with probability $Q_d/2^{\lambda}$, unless Hit occurs. We have that

$$\left|\Pr_7[\mathsf{Win}] - \Pr_8[\mathsf{Win}]\right| \leq Q_d/2^{\lambda} + \Pr_8[\mathsf{Hit}].$$

To give an upper bound on $\Pr_8[\mathsf{Hit}]$, we construct a PPT adversary $\mathcal{B}_3$ in Fig. 9 against the weak APR-CMA security of the de-randomized delegatable affine MAC. According to the weak-APR-CMA security game (see Fig. 2), $\mathcal{B}_3$ has access to $\textsc{Eval}_{\mathsf{MAC}}$ oracle and one access to $\textsc{Chal}_{\mathsf{MAC}}$ oracle, and aims to tell the output $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$ of $\textsc{Chal}_{\mathsf{MAC}}$ is properly computed or randomly chosen. In INITIALIZE, $\mathcal{B}_3$ does not choose $\mathsf{sk}_{\mathsf{MAC}} = (\mathsf{k}_{\mathsf{PRF}}, \mathbf{B}, \mathbf{x}_0, \ldots, \mathbf{x}_\ell, x'_0)$, and implicitly sets $\mathsf{sk}_{\mathsf{MAC}}$ to be the secret key used by its weak APR-CMA challenger. It invokes $(ek_{\mathsf{CH}}, td_{\mathsf{CH}}) \leftarrow\!\!\$\ \mathsf{CH.Gen}(1^\lambda)$, picks $\mathbf{A}, \mathbf{Z}_i, \mathbf{z}'_0$ randomly, and sets $\mathsf{td} := \left(td_{\mathsf{CH}}, \mathbf{A}, \{\mathbf{Z}_i\}_{i\in[0,\ell]}, \mathbf{z}'_0\right)$ as its trapdoor. $\mathcal{B}_3$ simulates the scenario of $\beta = 1$ in G_8 for $\mathcal{A}$ as follows.

- For $\textsc{Enc}(\mathsf{id}^*, m^*)$, it picks $[\tilde{\mathbf{c}}_0^*]_1 \leftarrow\!\!\$\ \mathbb{G}_1^{k+1}$, $\tilde{R}_{\mathsf{CH}}^* \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{CH}}$, and computes $\mathsf{id}'^* := \mathsf{Eval}(ek_{\mathsf{CH}}, [\tilde{\mathbf{c}}_0^*]_1; \tilde{R}_{\mathsf{CH}}^*)$. Then $\mathcal{B}_3$ submits $\mathsf{id}^*|\mathsf{id}'^*$ to its own $\textsc{Chal}_{\mathsf{MAC}}$ oracle, and obtains $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$. Then it picks $[\overline{\mathbf{c}_0^*}]_1 \leftarrow\!\!\$\ \mathbb{G}_1^k$, computes $[\underline{\mathbf{c}_0^*}]_1 := \left[h + \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*}\right]_1$, and reopens $R_{\mathsf{CH}}^* \leftarrow \mathsf{Equiv}\left(td_{\mathsf{CH}}, [\tilde{\mathbf{c}}_0^*]_1, \tilde{R}_{\mathsf{CH}}^*, [\mathbf{c}_0^*]_1\right)$ using the trapdoor td_{CH}. Finally, it picks $[\mathbf{c}_1^*]_1$, K^* randomly, as in the scenario of $\beta = 1$ in G_8.
  Note that, h is chosen from $\mathbb{Z}_q$ with fresh randomness, thus $[\mathbf{c}_0^*]_1$ is independent of $\tilde{R}_{\mathsf{CH}}^*$. Since $\tilde{R}_{\mathsf{CH}}^*$ is uniformly distributed and independent of $[\tilde{\mathbf{c}}_0^*]_1$ and $[\mathbf{c}_0^*]_1$, then by the Equivocation property of CH, R_{CH}^* is uniformly distributed over $\mathcal{R}_{\mathsf{CH}}$ and independent of $[\mathbf{c}_0^*]_1$, same as G_8.
- For $\textsc{USKGen}(\mathsf{id})$, $\mathsf{id} \neq \mathsf{id}^*$, $\mathcal{B}_3$ submits id to its own $\textsc{Eval}_{\mathsf{MAC}}$ oracle, and obtains $\left([\mathbf{t}]_2, [u]_2, \{[d_i]_2\}_{i\in[l(\mathsf{id})+1,\ell]}\right)$. It then computes $[\mathbf{v}]_2$ from $[u]_2$ and $[\mathbf{e}_i]_2$ from $[d_i]_2$ with the trapdoor $\mathbf{A}, \mathbf{Z}_i, \mathbf{z}'_0$, as in G_8.
- For $\textsc{Dec}(\mathsf{id}, \langle \mathsf{C}, \chi \rangle)$, $\mathcal{B}_3$ first checks whether the following holds

$$[\mathbf{c}_1]_1 = \left[\sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + \mathbf{h}_0\right]_1. \tag{16}$$

  If (16) holds, $\mathcal{B}_3$ outputs 1 to its weak APR-CMA challenger. In the case of $\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^*$, $\mathcal{B}_3$ responds without using the secret key, as in G_8. In the case of $\mathsf{id} = \mathsf{id}^* \wedge \mathsf{id}' \neq \mathsf{id}'^*$, $\mathcal{B}_3$ submits $\mathsf{id}^*|\mathsf{id}'$ to its own $\textsc{Eval}_{\mathsf{MAC}}$ oracle, and obtains $\left([\mathbf{t}]_2, [u']_2, \{[d'_i]_2\}_{i\in[l(\mathsf{id}^*|\mathsf{id}')+1,\ell]}\right)$. It then computes $[\mathbf{v}']_2$ from $[u']_2$ with the trapdoor $\mathbf{A}, \mathbf{Z}_i, \mathbf{z}'_0$, as in G_8. In the case of $\mathsf{id} \neq \mathsf{id}^*$, $\mathcal{B}_3$ submits id to its own $\textsc{Eval}_{\mathsf{MAC}}$ oracle, and obtains $\left([\mathbf{t}]_2, [u]_2, \{[d_i]_2\}_{i\in[l(\mathsf{id})+1,\ell]}\right)$. It then computes $[\mathbf{v}]_2$ from $[u]_2$ and $[\mathbf{e}_i]_2$ from $[d_i]_2$ with the trapdoor $\mathbf{A}, \mathbf{Z}_i, \mathbf{z}'_0$, as in G_8.

Therefore $\mathcal{B}_3$ perfectly simulates the scenario of $\beta = 1$ in G_8 for $\mathcal{A}$, and outputs 1 if and only if (16) holds.

INITIALIZE:
$(ek_{\mathsf{CH}}, td_{\mathsf{CH}}) \leftarrow\!\$\ \mathsf{CH.Gen}(1^\lambda)$.
$\mathbf{A} \leftarrow\!\$\ \mathcal{D}_k$, with $\mathbf{A} \in \mathbb{Z}_q^{(k+1)\times k}$.
$\varepsilon \leftarrow\!\$\ \text{INITIALIZE}_{\mathsf{MAC}}$.
For $i \in [0,\ell]$, $\mathbf{Z}_i \leftarrow\!\$\ \mathbb{Z}_q^{n\times k}$. $\mathbf{z}'_0 \leftarrow\!\$\ \mathbb{Z}_q^{1\times k}$.
$\mathsf{pk} := \big(ek_{\mathsf{CH}}, \mathcal{PG}, [\mathbf{A}]_1, \{[\mathbf{Z}_i]_1\}_{i\in[0,\ell]}, [\mathbf{z}'_0]_1\big)$.
$\mathsf{td} := \big(td_{\mathsf{CH}}, \mathbf{A}, \{\mathbf{Z}_i\}_{i\in[0,\ell]}, \mathbf{z}'_0\big)$.
Return pk.

ENC(id^*, m^*): // one query
If $\mathsf{id}^* \in \mathcal{Q}_{\mathcal{USK}}$, Return $\perp$.
$[\tilde{\mathbf{c}}_0^*]_1 \leftarrow\!\$\ \mathbb{G}_1^{k+1}$. $\tilde{R}^*_{\mathsf{CH}} \leftarrow\!\$\ \mathcal{R}_{\mathsf{CH}}$.
$\mathsf{id}'^* := \mathsf{Eval}(ek_{\mathsf{CH}}, [\tilde{\mathbf{c}}_0^*]_1; \tilde{R}^*_{\mathsf{CH}}) \in \mathcal{ID}$.
$\big([h]_1, [\mathbf{h}_0]_1, [h_1]_T\big) \leftarrow\!\$\ \text{CHAL}_{\mathsf{MAC}}(\mathsf{id}^*|\mathsf{id}'^*)$.
$[\underline{\mathbf{c}_0^*}]_1 \leftarrow\!\$\ \mathbb{G}_1^k$, $[\underline{\mathbf{c}_0^*}]_1 := \Big[h + \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*}\Big]_1$.
$R^*_{\mathsf{CH}} \leftarrow \mathsf{Equiv}\big(td_{\mathsf{CH}}, [\tilde{\mathbf{c}}_0^*]_1, \tilde{R}^*_{\mathsf{CH}}, [\mathbf{c}_0^*]_1\big)$.
$[\mathbf{c}_1^*]_1 \leftarrow\!\$\ \mathbb{G}_1^n$, $\mathsf{K}^* \leftarrow\!\$\ \mathbb{G}_T$.
$[\mathbf{c}_1^*]_1 := \Big[\sum_{i=0}^{l(\mathsf{id}^*|\mathsf{id}'^*)} f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \mathbf{Z}_i \overline{\mathbf{A}}^{-1} \overline{\mathbf{c}_0^*} + \mathbf{h}_0\Big]_1$.
$\mathsf{K}^* := \Big[\mathbf{z}'_0 \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + h_1\Big]_T$.
$(k_1^*, k_2^*, k_3^*) := \mathsf{H}(\mathsf{K}^*) \in \{0,1\}^{3\lambda}$.
$\chi_1^* := k_1^* + m^*$, $\chi_2^* := k_2^* \cdot \chi_1^* + k_3^*$.
Ret. $\mathsf{C}^* := ([\mathbf{c}_0^*]_1, [\mathbf{c}_1^*]_1, R^*_{\mathsf{CH}})$, $\chi^* := (\chi_1^*, \chi_2^*)$.

USKGEN(id):
If $\mathsf{id} = \mathsf{id}^*$, Return $\perp$.
$\mathcal{Q}_{\mathcal{USK}} := \mathcal{Q}_{\mathcal{USK}} \cup \{\mathsf{id}\}$.
$\big([\mathbf{t}]_2, [u]_2, \{[d_i]_2\}_{i\in[l(\mathsf{id})+1,\ell]}\big) \leftarrow \text{EVAL}_{\mathsf{MAC}}(\mathsf{id})$.
$[\mathbf{v}^\top]_2 := \Big[\Big(\sum_{i=0}^{l(\mathsf{id})} f_i(\mathsf{id}) \cdot \mathbf{t}^\top \cdot \mathbf{Z}_i + \mathbf{z}'_0 - u \cdot \underline{\mathbf{A}}\Big) \cdot \overline{\mathbf{A}}^{-1}\Big]_2$.
For $i \in [l(\mathsf{id})+1, \ell]$,
$[\mathbf{e}_i]_2 := \Big[(\overline{\mathbf{A}}^{-1})^\top \cdot (\mathbf{Z}_i^\top \cdot \mathbf{t} - \underline{\mathbf{A}}^\top \cdot d_i)\Big]_2$.
Return $\mathsf{usk}[\mathsf{id}] := \big([\mathbf{t}]_2, [u]_2, [\mathbf{v}]_2, \{[d_i]_2, [\mathbf{e}_i]_2\}_{i\in[l(\mathsf{id})+1,\ell]}\big)$.

DEC($\mathsf{id}, \langle \mathsf{C}, \chi\rangle$):
If $(\mathsf{id}, \langle \mathsf{C}, \chi\rangle) = (\mathsf{id}^*, \langle \mathsf{C}^*, \chi^*\rangle)$, Return $\perp$.
Parse $\mathsf{C} = ([\mathbf{c}_0]_1, [\mathbf{c}_1]_1, R_{\mathsf{CH}})$, $\chi = (\chi_1, \chi_2)$.
If $[\mathbf{c}_1]_1 = \Big[\sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + \mathbf{h}_0\Big]_1$
$\mathcal{B}_3$ outputs 1 to its challenger.
$\mathsf{id}' := \mathsf{Eval}(ek_{\mathsf{CH}}, [\mathbf{c}_0]_1; R_{\mathsf{CH}}) \in \mathcal{ID}$.
If $\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^* \wedge ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) \neq ([\mathbf{c}_0^*]_1, R^*_{\mathsf{CH}})$,
Return $\perp$.
If $\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^* \wedge ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) = ([\mathbf{c}_0^*]_1, R^*_{\mathsf{CH}})$
$\wedge [\mathbf{c}_1]_1 = [\mathbf{c}_1^*]_1 \wedge \chi \neq \chi^*$
$\mathsf{K} := \mathsf{K}^*$.
If $\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^* \wedge ([\mathbf{c}_0]_1, R_{\mathsf{CH}}) = ([\mathbf{c}_0^*]_1, R^*_{\mathsf{CH}})$
$\wedge [\mathbf{c}_1]_1 \neq [\mathbf{c}_1^*]_1$
Return $\perp$.
If $\mathsf{id} = \mathsf{id}^* \wedge \mathsf{id}' \neq \mathsf{id}'^*$, do the following:
$\big([\mathbf{t}]_2, [u']_2, \{[d'_i]_2\}_{i\in[l(\mathsf{id}^*|\mathsf{id}')+1,\ell]}\big)$
$\leftarrow \text{EVAL}_{\mathsf{MAC}}(\mathsf{id}^*|\mathsf{id}')$.
$[\mathbf{v}'^\top]_2 := \Big[\Big(\sum_{i=0}^{l(\mathsf{id}^*|\mathsf{id}')} f_i(\mathsf{id}^*|\mathsf{id}') \cdot \mathbf{t}^\top \cdot \mathbf{Z}_i + \mathbf{z}'_0 - u' \cdot \underline{\mathbf{A}}\Big) \cdot \overline{\mathbf{A}}^{-1}\Big]_2$.
$\mathsf{K} := \big[(\mathbf{v}'^\top \mid u') \cdot \mathbf{c}_0\big]_T / \big[\mathbf{t}^\top \cdot \mathbf{c}_1\big]_T \in \mathbb{G}_T$.
Else $\mathsf{id} \neq \mathsf{id}^*$, do the following:
$\big([\mathbf{t}]_2, [u]_2, \{[d_i]_2\}_{i\in[l(\mathsf{id})+1,\ell]}\big) \leftarrow \text{EVAL}_{\mathsf{MAC}}(\mathsf{id})$.
$[\mathbf{v}^\top]_2 := \Big[\Big(\sum_{i=0}^{l(\mathsf{id})} f_i(\mathsf{id}) \cdot \mathbf{t}^\top \cdot \mathbf{Z}_i + \mathbf{z}'_0 - u \cdot \underline{\mathbf{A}}\Big) \cdot \overline{\mathbf{A}}^{-1}\Big]_2$.
For $i \in [l(\mathsf{id})+1, \ell]$,
$[\mathbf{e}_i]_2 := \Big[(\overline{\mathbf{A}}^{-1})^\top \cdot (\mathbf{Z}_i^\top \cdot \mathbf{t} - \underline{\mathbf{A}}^\top \cdot d_i)\Big]_2$.
$[u']_2 := \Big[u + \sum_{i=l(\mathsf{id})+1}^{l(\mathsf{id}|\mathsf{id}')} f_i(\mathsf{id}|\mathsf{id}') \cdot d_i\Big]_2 \in \mathbb{G}_2$.
$[\mathbf{v}']_2 := \Big[\mathbf{v} + \sum_{i=l(\mathsf{id})+1}^{l(\mathsf{id}|\mathsf{id}')} f_i(\mathsf{id}|\mathsf{id}') \cdot \mathbf{e}_i\Big]_2 \in \mathbb{G}_2^k$.
$\mathsf{K} := \big[(\mathbf{v}'^\top \mid u') \cdot \mathbf{c}_0\big]_T / \big[\mathbf{t}^\top \cdot \mathbf{c}_1\big]_T \in \mathbb{G}_T$.
$(k_1, k_2, k_3) := \mathsf{H}(\mathsf{K}) \in \{0,1\}^{3\lambda}$.
If $\chi_2 = k_2 \cdot \chi_1 + k_3$, Return $m := \chi_1 - k_1$.
Else $\chi_2 \neq k_2 \cdot \chi_1 + k_3$, Return $\perp$.

FINALIZE(β'):
$\mathcal{B}_4$ outputs β' to its challenger.

Fig. 9 Description of $\mathcal{B}_3$ (resp. $\mathcal{B}_4$) for the proof of Lemma 5 (resp. Lemma 6), which has access to the oracles $\text{EVAL}_{\mathsf{MAC}}$ and $\text{CHAL}_{\mathsf{MAC}}$ of the weak-APR-CMA game (cf. Fig. 2). The framed parts appear only in the description of $\mathcal{B}_3$, while the shadowed parts appear only in the description of $\mathcal{B}_4$

If $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$ is real, then $\mathbf{h}_0 = \sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \mathbf{x}_i \cdot h$. Thus (16) is equivalent to

$$\mathbf{c}_1 = \sum f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot (\mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}_0^*} + \mathbf{x}_i \cdot h).$$

So in this case, $\mathcal{B}_3$ outputs 1 if and only if the event Hit happens in the above simulated game, i.e., Hit happens in G_8 (since the simulation is perfect).

Whereas if $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$ is random, then $\mathbf{h}_0$ is uniformly chosen from $\mathbb{Z}_q^n$ and independent of other parts of the above game, thus (16) can happen with probability $1/q^n$. So in this case, $\mathcal{B}_3$ outputs 1 with probability at most Q_d/q^n.

Accordingly, it holds that $\big|\Pr_8[\mathsf{Hit}] - Q_d/q^n\big| \leq \mathsf{Adv}^{weak\text{-}apr\text{-}cma}_{\mathsf{MAC},\mathcal{B}_3}(\lambda)$.

Overall we have that

$$\begin{aligned}\left|\Pr_7[\mathsf{Win}] - \Pr_8[\mathsf{Win}]\right| &\leq Q_d/2^{\lambda} + \Pr_8[\mathsf{Hit}] \\ &\leq Q_d/2^{\lambda} + \mathsf{Adv}^{\textit{weak-apr-cma}}_{\mathsf{MAC},\mathcal{B}_3}(\lambda) + Q_d/q^n,\end{aligned}$$

and the lemma follows. □

Now in G_8, we are in the position to make the reduction to the weak APR-CMA security of the de-randomized delegatable affine MAC.

Lemma 6 *There exists a PPT adversary $\mathcal{B}_4$ against the weak APR-CMA security of the de-randomized delegatable affine* MAC*, such that*

$$\left|\Pr_8[\mathsf{Win}] - 1/2\right| \leq \mathsf{Adv}^{\textit{weak-apr-cma}}_{\mathsf{MAC},\mathcal{B}_4}(\lambda).$$

Proof of Lemma 6 We construct a PPT adversary $\mathcal{B}_4$ in Fig. 9 against the weak APR-CMA security of MAC. $\mathcal{B}_4$ has access to $\textsc{Eval}_{\mathsf{MAC}}$ oracle and one access to $\textsc{Chal}_{\mathsf{MAC}}$ oracle, and aims to tell the output $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$ of $\textsc{Chal}_{\mathsf{MAC}}$ is properly computed or randomly chosen. In Initialize, $\mathcal{B}_4$ does not choose $\mathsf{sk}_{\mathsf{MAC}} = (\mathsf{k}_{\mathsf{PRF}}, \mathbf{B}, \mathbf{x}_0, \ldots, \mathbf{x}_\ell, x'_0)$, and implicitly sets $\mathsf{sk}_{\mathsf{MAC}}$ to be the secret key used by its weak APR-CMA challenger. It invokes $(ek_{\mathsf{CH}}, td_{\mathsf{CH}}) \leftarrow\!\$\ \mathsf{CH.Gen}(1^\lambda)$, picks $\mathbf{A}, \mathbf{Z}_i, \mathbf{z}'_0$ randomly, and sets $\mathsf{td} := \big(td_{\mathsf{CH}}, \mathbf{A}, \{\mathbf{Z}_i\}_{i\in[0,\ell]}, \mathbf{z}'_0\big)$ as its trapdoor. $\mathcal{B}_4$ simulates the scenario of $\beta = 0$ or $\beta = 1$ in G_8 for $\mathcal{A}$ as follows.

- For $\textsc{Enc}(\mathsf{id}^*, m^*)$, it picks $[\tilde{\mathbf{c}}^*_0]_1 \leftarrow\!\$\ \mathbb{G}_1^{k+1}$, $\tilde{R}^*_{\mathsf{CH}} \leftarrow\!\$\ \mathcal{R}_{\mathsf{CH}}$, and computes $\mathsf{id}'^* := \mathsf{Eval}(ek_{\mathsf{CH}}, [\tilde{\mathbf{c}}^*_0]_1; \tilde{R}^*_{\mathsf{CH}})$. Then $\mathcal{B}_4$ submits $\mathsf{id}^*|\mathsf{id}'^*$ to its own $\textsc{Chal}_{\mathsf{MAC}}$ oracle, and obtains $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$. Then it picks $[\overline{\mathbf{c}^*_0}]_1 \leftarrow\!\$\ \mathbb{G}_1^k$, computes $[\underline{\mathbf{c}}^*_0]_1 := \big[h + \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{c}^*_0}\big]_1$, and reopens $R^*_{\mathsf{CH}} \leftarrow \mathsf{Equiv}\big(td_{\mathsf{CH}}, [\tilde{\mathbf{c}}^*_0]_1, \tilde{R}^*_{\mathsf{CH}}, [\mathbf{c}^*_0]_1\big)$ using the trapdoor td_{CH}. Finally, it computes

$$[\mathbf{c}^*_1]_1 := \left[\sum\nolimits_{i=0}^{l(\mathsf{id}^*|\mathsf{id}'^*)} f_i(\mathsf{id}^*|\mathsf{id}'^*)\mathbf{Z}_i\overline{\mathbf{A}}^{-1}\overline{\mathbf{c}^*_0} + \mathbf{h}_0\right]_1, \mathsf{K}^* := \left[\mathbf{z}'_0\overline{\mathbf{A}}^{-1}\overline{\mathbf{c}^*_0} + h_1\right]_T.$$

  Similar to the proof of the previous lemma, R^*_{CH} is uniformly distributed over $\mathcal{R}_{\mathsf{CH}}$ and independent of $[\mathbf{c}^*_0]_1$, same as G_8.
  If $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$ is real, then

$$\mathbf{h}_0 = \sum\nolimits_{i=0}^{l(\mathsf{id}^*|\mathsf{id}'^*)} f_i(\mathsf{id}^*|\mathsf{id}'^*) \cdot \mathbf{x}_i \cdot h, \quad h_1 = x'_0 \cdot h.$$

  Thus $[\mathbf{c}^*_1]_1$ and K^* are computed as in the scenario of $\beta = 0$ in G_8.
  If $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$ is random, then $\mathbf{h}_0$ and h_1 are randomly chosen. Thus $[\mathbf{c}^*_1]_1$ and K^* are uniformly distributed, as in the scenario of $\beta = 1$ in G_8.
- For $\textsc{UskGen}(\mathsf{id})$, $\mathsf{id} \neq \mathsf{id}^*$, $\mathcal{B}_4$ submits id to its own $\textsc{Eval}_{\mathsf{MAC}}$ oracle, and obtains $\big([\mathbf{t}]_2, [u]_2, \{[d_i]_2\}_{i\in[l(\mathsf{id})+1,\ell]}\big)$. It then computes $[\mathbf{v}]_2$ from $[u]_2$ and $[\mathbf{e}_i]_2$ from $[d_i]_2$ with the trapdoor $\mathbf{A}, \mathbf{Z}_i, \mathbf{z}'_0$, as in G_8.
- For $\textsc{Dec}(\mathsf{id}, \langle \mathsf{C}, \chi \rangle)$, in the case of $\mathsf{id}|\mathsf{id}' = \mathsf{id}^*|\mathsf{id}'^*$, $\mathcal{B}_4$ responds without using the secret key, as in G_8. In the case of $\mathsf{id} = \mathsf{id}^* \wedge \mathsf{id}' \neq \mathsf{id}'^*$, $\mathcal{B}_4$ submits $\mathsf{id}^*|\mathsf{id}'$ to $\textsc{Eval}_{\mathsf{MAC}}$, and obtains $\big([\mathbf{t}]_2, [u']_2, \{[d'_i]_2\}_{i\in[l(\mathsf{id}^*|\mathsf{id}')+1,\ell]}\big)$. It then computes $[\mathbf{v}']_2$ from $[u']_2$ with the trapdoor $\mathbf{A}, \mathbf{Z}_i, \mathbf{z}'_0$, as in G_8. In the case of $\mathsf{id} \neq \mathsf{id}^*$, $\mathcal{B}_4$ submits id to its own $\textsc{Eval}_{\mathsf{MAC}}$ oracle, and obtains $\big([\mathbf{t}]_2, [u]_2, \{[d_i]_2\}_{i\in[l(\mathsf{id})+1,\ell]}\big)$. It then computes $[\mathbf{v}]_2$ from $[u]_2$ and $[\mathbf{e}_i]_2$ from $[d_i]_2$ with the trapdoor $\mathbf{A}, \mathbf{Z}_i, \mathbf{z}'_0$, as in G_8.

- For FINALIZE, $\mathcal{B}_4$ outputs whatever $\mathcal{A}$ outputs.

Hence, if $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$ is real, $\mathcal{B}_4$ perfectly simulates the scenario of $\beta = 0$ in G_8 with $\mathcal{A}$; if $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$ is random, $\mathcal{B}_4$ perfectly simulates the scenario of $\beta = 1$ in G_8 with $\mathcal{A}$. Any difference between $\beta = 0$ and $\beta = 1$ in G_8 results in $\mathcal{B}_4$'s advantage over the weak APR-CMA security game. Then the lemma follows.

Taking all things together, Theorem 2 follows. □

5 Application to simulation-based selective opening secure IBE

In a selective opening attack, an adversary sees a vector of ciphertexts, adaptively chooses to open some of them, and obtains the corresponding plaintexts and random coins used in the creation of the ciphertexts. When considering selective opening, chosen-ciphertext attack (SO-CCA2), the adversary also has access to a decryption oracle. The simulation-based SO-CCA2 (SIM-SO-CCA2) security requires: what a PPT SO-CCA2 adversary can compute can also be simulated by a PPT simulator with access only to the opened messages.

Our IBE in Sect. 4, enjoying tight PR-ID-CCA2 security, i.e., IND-ID-CCA2 security and ciphertext pseudorandomness, can be used to construct SIM-SO-CCA2 secure IBE scheme following the work of [27]. Recall that Lai et al. [27] gave a paradigm for constructing SIM-SO-CCA2 secure IBE from the so-called extractable IBE with IND-ID-CCA2 security and an information-theoretic primitive called strengthened cross authentication code. They also proposed two instantiations of the extractable IBE with IND-ID-CCA2 security based on the subgroup indistinguishability assumption over bilinear groups of composite order and the DLIN assumption over bilinear groups of prime order, respectively. However, both of the two extractable IBEs have loose reductions. In Appendix 3, we showed how to construct an extractable IBE from our IBE with pseudorandom ciphertexts. We proved that the PR-ID-CCA2 security of IBE implies IND-ID-CCA2 security of the extractable IBE. Therefore, our IBE in Sect. 4 can be employed to construct the first extractable IBE with tight IND-ID-CCA2 security, which in turn results in a SIM-SO-CCA2 secure IBE which enjoys a tighter security reduction than those in [27] and is also the first scheme based on the Matrix DDH assumption.

Acknowledgements We would like to thank the referees for their helpful comments and suggestions. Shuai Han and Shengli Liu were supported by the National Natural Science Foundation of China (Grant Nos. 61672346, 61373153). Baodong Qin was supported by the National Natural Science Foundation of China (Grant No. 61502400), by the Natural Science Foundation of Southwest University of Science and Technology (Grant No. 16zx7107) and by the Foundation of Sichuan Educational Committee (Grant No. 16ZB0140). Dawu Gu was sponsored by Program of Shanghai Subject Chief Scientist (No. 16XD1401300).

Appendix 1: Efficiency comparison table of tightly secure IBEs

See Table 3

Table 3 Comparison between the known tightly-secure IBEs with identity space $\mathcal{ID} = \{0,1\}^\lambda$ in prime order groups based on standard assumptions

Scheme	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{usk}\rvert$	$\lvert\mathsf{C}\rvert$
CW13 [12]	$2\lambda k^2 + 2k^2 + k$	$4k$	$4k$
BKP14's $\mathsf{IBE}_{\mathsf{NR}}$ [6]	$2\lambda k^2 + k^2 + 2k$	$2k+1$	$2k+1$
BKP14's $\mathsf{HIBE}_{\mathsf{NR}}$ [6] + CHK	$8\lambda(k^2 + k/2) + 2k^2 + 2k$	$2\lambda(k+1) + 2k + 1$	$2k+1$
AHY15[a] [3]	$16\lambda + 11$	8	8
GCDCT16 [19]	$6\lambda k^2 + 3k^2 + k$	$6k$	$6k$
Ours: $\mathsf{IBE}_{\mathsf{NR}}$ (§4)	$4\lambda k^2 + k^2 + 2k$	$2\lambda(k+1) + 2k + 1$	$2k+1$

"CHK" stands for the general paradigm proposed by [11]. k is the parameter for the k-Linear and $\mathcal{D}_k$-MDDH assumptions. AHY15's [3] security is based on the 2-Linear assumption, i.e., $k = 2$. $\lvert\mathsf{pk}\rvert$, $\lvert\mathsf{usk}\rvert$ and $\lvert\mathsf{C}\rvert$ denote the size per public key, user secret key and ciphertext, respectively. Here we count the number of group elements in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$

[a] AHY15 [3] also proposed two non-anonymous schemes, here we only show the anonymous one

Appendix 2: Proof of Lemma 3

In $\mathsf{G}_{\zeta,\eta-1}$ (resp., $\mathsf{G}_{\zeta,\eta}$), the challenger uses $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1})$ (resp., $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta})$) as the randomness to compute $[u]_2$ in $\textsc{Eval}(\mathsf{m})$ and uses $\mathsf{RT}(\mathsf{m}^*_{|\zeta,\eta-1})$ (resp., $\mathsf{RT}(\mathsf{m}^*_{|\zeta,\eta})$) as the randomness to compute h_1 in $\textsc{Chal}(\mathsf{m}^*)$. By the special property of RT, $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1})$ and $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta})$ are the same when the message m satisfies $|\mathsf{m}| < (\zeta-1)\lambda + \eta$ or $\mathsf{m}_{\zeta,\eta} = b_{\zeta,\eta}$, and they are independent of each other when the message m satisfies $|\mathsf{m}| \geq (\zeta-1)\lambda + \eta$ and $\mathsf{m}_{\zeta,\eta} = 1 - b_{\zeta,\eta}$. The difference between $\mathsf{G}_{\zeta,\eta-1}$ and $\mathsf{G}_{\zeta,\eta}$ can be reduced to the Q-fold $\mathcal{D}_k$-MDDH assumption for the group $\mathbb{G}_2$.

More precisely, we construct PPT adversaries $\mathcal{B}_1, \mathcal{B}_2\big(\mathcal{PG}, [\mathbf{A}]_2, [\mathbf{H}]_2\big)$, where $\mathcal{PG} \leftarrow_\$ \mathsf{PGGen}(1^\lambda)$, $\mathbf{A} \leftarrow_\$ \mathcal{D}_k$, and $\mathbf{H} = \mathbf{A}\cdot\mathbf{W} + \mathbf{R}$ with $\mathbf{W} \leftarrow_\$ \mathbb{Z}_q^{k\times Q}$, to distinguish whether $\mathbf{R} = \mathbf{0}$ (i.e., $\big(\mathcal{PG}, [\mathbf{A}]_2, [\mathbf{H}]_2\big)$ is identical to the real $\mathcal{D}_k$-MDDH distribution) or $\mathbf{R} \leftarrow_\$ \mathbb{Z}_q^{(k+1)\times Q}$ (i.e., $\big(\mathcal{PG}, [\mathbf{A}]_2, [\mathbf{H}]_2\big)$ is identical to the random $\mathcal{D}_k$-MDDH distribution) in Fig. 10. $\mathcal{B}_1$ and $\mathcal{B}_2$ are the same except their strategies in $\textsc{Finalize}$.

In $\textsc{Initialize}$, $\mathcal{B}_1, \mathcal{B}_2$ choose $\mathbf{r} \leftarrow_\$ \mathbb{Z}_q^{k+1}$ and set $\mathbf{x}_{\zeta,\eta}^{(1-b_{\zeta,\eta})^\top} := \mathbf{r}^\top\mathbf{A}\overline{\mathbf{A}}^{-1}$ implicitly. Observe that $\mathbf{x}_{\zeta,\eta}^{(1-b_{\zeta,\eta})^\top} = \mathbf{r}^\top\mathbf{A}\overline{\mathbf{A}}^{-1} = (\overline{\mathbf{r}}^\top\overline{\mathbf{A}} + \underline{\mathbf{r}}^\top\underline{\mathbf{A}})\overline{\mathbf{A}}^{-1} = \overline{\mathbf{r}}^\top + \underline{\mathbf{r}}^\top\underline{\mathbf{A}}\overline{\mathbf{A}}^{-1}$, thus it is uniformly distributed over $\mathbb{Z}_q^{1\times k}$ because of the randomness of $\overline{\mathbf{r}}$, as in $\mathsf{G}_{\zeta,\eta-1}$ and $\mathsf{G}_{\zeta,\eta}$.

In $\textsc{Eval}(\mathsf{m})$, if $|\mathsf{m}| < (\zeta-1)\lambda + \eta$, then $\mathsf{m}_{|\zeta,\eta-1} = \mathsf{m}_{|\zeta,\eta} = \mathsf{m}$, and $\mathcal{B}_1, \mathcal{B}_2$ use the randomness $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1})$, which equals $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta})$, to compute $[u]_2$, as in $\mathsf{G}_{\zeta,\eta-1}$ and $\mathsf{G}_{\zeta,\eta}$. And note that $\mathcal{B}_1, \mathcal{B}_2$ can compute $[d_{\zeta,\eta}^{(1-b_{\zeta,\eta})}]_2 := [\mathbf{r}^\top\cdot\mathbf{A}\cdot\mathbf{s}]_2$, which is the same as $\mathsf{G}_{\zeta,\eta-1}$ and $\mathsf{G}_{\zeta,\eta}$, since

$$[\mathbf{r}^\top\cdot\mathbf{A}\cdot\mathbf{s}]_2 = \Big[(\mathbf{r}^\top\mathbf{A}\overline{\mathbf{A}}^{-1})\cdot(\overline{\mathbf{A}}\mathbf{s})\Big]_2 = [\mathbf{x}_{\zeta,\eta}^{(1-b_{\zeta,\eta})^\top}\cdot\mathbf{t}]_2.$$

In $\textsc{Eval}(\mathsf{m})$, if $|\mathsf{m}| \geq (\zeta-1)\lambda+\eta$, $\mathcal{B}_1, \mathcal{B}_2$ implement an injective function $\alpha : \{0,1\}^* \longrightarrow [1, Q]$ on the fly, and compute $\mathbf{s}' := \mathsf{TRF}(\mathsf{m}) \in \mathbb{Z}_q^k$ and $[\mathbf{t}]_2 := \big[\overline{\mathbf{A}}\cdot\mathbf{s}' + \overline{\mathbf{H}}_c\big]_2$, where $c := \alpha(\mathsf{m}_{|\zeta,\eta-1}) \in [1, Q]$ and $\overline{\mathbf{H}}_c$ is the cth column of the matrix $\overline{\mathbf{H}}$. Then for message m with $\mathsf{m}_{\zeta,\eta} = b_{\zeta,\eta}$, $\mathcal{B}_1, \mathcal{B}_2$ use $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1})$, which equals $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta})$, to compute $[u]_2$ the same way as in $\mathsf{G}_{\zeta,\eta-1}$ and $\mathsf{G}_{\zeta,\eta}$. As for message m with $\mathsf{m}_{\zeta,\eta} = 1 - b_{\zeta,\eta}$, $\mathcal{B}_1, \mathcal{B}_2$ compute $[u]_2$ in a different way with

$\underline{\text{INITIALIZE}_{\mathsf{MAC}}}$:
$\mathsf{RT}(\varepsilon) = x'_0 \leftarrow\$ \mathbb{Z}_q$.
$(b_{1,1}, \cdots, b_{L,\lambda}) \leftarrow\$ \{0,1\}^{L\lambda}$.
For $(i,j) \in [L] \times [\lambda] \setminus \{(\zeta,\eta)\}$
  $\mathbf{x}_{i,j}^{(0)}, \mathbf{x}_{i,j}^{(1)} \leftarrow\$ \mathbb{Z}_q^k$.
For $(i,j) = (\zeta,\eta)$
  $\mathbf{x}_{\zeta,\eta}^{(b_{\zeta,\eta})} \leftarrow\$ \mathbb{Z}_q^k$. $\mathbf{r} \leftarrow\$ \mathbb{Z}_q^{k+1}$
  $(\mathbf{x}_{\zeta,\eta}^{(1-b_{\zeta,\eta})})^\top := \mathbf{r}^\top \mathbf{A}\, \overline{\mathbf{A}}^{-1}$ implicitly).
$\beta \leftarrow\$ \{0,1\}$. // challenge bit
Return ε.

$\underline{\text{CHAL}(\mathsf{m}^*)}$: // one query
If $\mathsf{prefix}(\mathsf{m}^*) \cap \mathcal{Q}_{\mathcal{EVA}} \neq \emptyset$, Return $\perp$.
Parse $\mathsf{m}^* = (\mathsf{m}^*_{i,j})_{i\in[p(\mathsf{m}^*)],j\in[\lambda]} \in \mathcal{M}$.
$h \leftarrow\$ \mathbb{Z}_q$.
If $|\mathsf{m}^*| < (\zeta-1)\lambda + \eta \vee \mathsf{m}^*_{\zeta,\eta} = b_{\zeta,\eta}$
  $\mathbf{h}_0 := \sum_{i=1}^{p(\mathsf{m}^*)} \sum_{j=1}^{\lambda} \mathbf{x}_{i,j}^{(\mathsf{m}^*_{i,j})} \cdot h \in \mathbb{Z}_q^k$.
  $h_1 := \mathrm{RT}(\mathsf{m}^*_{|\zeta,\eta-1}) \cdot h \in \mathbb{Z}_q$.
Else $|\mathsf{m}^*| \geq (\zeta-1)\lambda + \eta \wedge \mathsf{m}^*_{\zeta,\eta} = 1 - b_{\zeta,\eta}$
  Abort, and set $\mathsf{abort} := \mathsf{true}$.
If $\beta = 1$
  $\mathbf{h}_0 \leftarrow\$ \mathbb{Z}_q^k$. $h_1 \leftarrow\$ \mathbb{Z}_q$.
Return $([h]_1, [\mathbf{h}_0]_1, [h_1]_T)$.

$\underline{\text{FINALIZE}(\beta')}$:
$\mathcal{B}_1$ returns $\overline{\mathsf{abort}}$.
$\mathcal{B}_2$ returns $\overline{\mathsf{abort}} \wedge (\beta' = \beta)$.

$\underline{\text{EVAL}(\mathsf{m})}$:
If $\mathsf{m} \in \mathsf{prefix}(\mathsf{m}^*)$, Return $\perp$.
$\mathcal{Q}_{\mathcal{EVA}} := \mathcal{Q}_{\mathcal{EVA}} \cup \{\mathsf{m}\}$.
Parse $\mathsf{m} = (\mathsf{m}_{i,j})_{i\in[p(\mathsf{m})],j\in[\lambda]} \in \mathcal{M}$.
If $|\mathsf{m}| < (\zeta-1)\lambda + \eta$,
  $\mathbf{s} := \mathsf{TRF}(\mathsf{m}) \in \mathbb{Z}_q^k$. $[\mathbf{t}]_2 := \left[\overline{\mathbf{A}} \cdot \mathbf{s}\right]_2 \in \mathbb{G}_2^k$.
  $[u]_2 := \left[\sum_{i=1}^{p(\mathsf{m})} \sum_{j=1}^{\lambda} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \mathbf{t} + \mathrm{RT}(\mathsf{m}_{|\zeta,\eta-1})\right]_2$
  For $(i,j) \in [p(\mathsf{m})+1, L] \times [\lambda] \setminus \{(\zeta,\eta)\}$
    $[d_{i,j}^{(0)}]_2 := [\mathbf{x}_{i,j}^{(0)\top} \cdot \mathbf{t}]_2$, $[d_{i,j}^{(1)}]_2 := [\mathbf{x}_{i,j}^{(1)\top} \cdot \mathbf{t}]_2$.
  For $(i,j) = (\zeta,\eta)$
    $[d_{\zeta,\eta}^{(b_{\zeta,\eta})}]_2 := [\mathbf{x}_{\zeta,\eta}^{(b_{\zeta,\eta})\top} \cdot \mathbf{t}]_2$, $[d_{\zeta,\eta}^{(1-b_{\zeta,\eta})}]_2 := [\mathbf{r}^\top \mathbf{A}\, \mathbf{s}]_2$.
If $|\mathsf{m}| \geq (\zeta-1)\lambda + \eta$,
  $c := \alpha(\mathsf{m}_{|\zeta,\eta-1}) \in [1, Q]$.
  $\mathbf{s}' := \mathsf{TRF}(\mathsf{m}) \in \mathbb{Z}_q^k$. $[\mathbf{t}]_2 := \left[\overline{\mathbf{A}} \cdot \mathbf{s}' + \overline{\mathbf{H}}_c\right]_2 \in \mathbb{G}_2^k$.
  If $\mathsf{m}_{\zeta,\eta} = b_{\zeta,\eta}$
    $[u]_2 := \left[\sum_{i=1}^{p(\mathsf{m})} \sum_{j=1}^{\lambda} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \mathbf{t} + \mathrm{RT}(\mathsf{m}_{|\zeta,\eta-1})\right]_2$.
  Else $\mathsf{m}_{\zeta,\eta} = 1 - b_{\zeta,\eta}$
    $[u]_2 := \Big[\sum_{\substack{(i,j)\in[p(\mathsf{m})]\times[\lambda] \\ (i,j)\neq(\zeta,\eta)}} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \mathbf{t} + \mathrm{RT}(\mathsf{m}_{|\zeta,\eta-1}) + \mathbf{r}^\top \cdot (\mathbf{A} \cdot \mathbf{s}' + \mathbf{H}_c)\Big]_2$.
  For $i \in [p(\mathsf{m})+1, L]$, $j \in [\lambda]$
    $[d_{i,j}^{(0)}]_2 := [\mathbf{x}_{i,j}^{(0)\top} \cdot \mathbf{t}]_2$, $[d_{i,j}^{(1)}]_2 := [\mathbf{x}_{i,j}^{(1)\top} \cdot \mathbf{t}]_2$.
Return
  $\left([\mathbf{t}]_2, [u]_2, \{[d_{i,j}^{(0)}]_2, [d_{i,j}^{(1)}]_2\}_{i\in[p(\mathsf{m})+1,L],j\in[\lambda]}\right)$.

Private $\underline{\mathrm{RT}(\mathsf{x})}$. // the same as Fig. 4

Fig. 10 Description of $\mathcal{B}_1, \mathcal{B}_2(\mathcal{PG}, [\mathbf{A}]_2, [\mathbf{H}]_2)$ for the proof of Lemma 3. Here $\mathbf{H}_c$ denotes the cth column of the matrix $\mathbf{H}$, and $\alpha : \{0,1\}^* \longrightarrow [1, Q]$ is an injective function implemented by $\mathcal{B}_1, \mathcal{B}_2$ on the fly

$$[u]_2 = \left[\sum_{\substack{(i,j)\in[p(\mathsf{m})]\times[\lambda] \\ (i,j)\neq(\zeta,\eta)}} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \mathbf{t} + \mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1}) + \mathbf{r}^\top \cdot (\mathbf{A} \cdot \mathbf{s}' + \mathbf{H}_c)\right]_2,$$

where $\mathbf{H}_c$ is the cth column of $\mathbf{H}$. We analyze the simulation as follows. Since $\mathbf{H} = \mathbf{A} \cdot \mathbf{W} + \mathbf{R}$, then $\mathbf{H}_c = \mathbf{A} \cdot \mathbf{W}_c + \mathbf{R}_c$ and $\overline{\mathbf{H}}_c = \overline{\mathbf{A}} \cdot \mathbf{W}_c + \overline{\mathbf{R}}_c$. Thus $[\mathbf{t}]_2 = \left[\overline{\mathbf{A}} \cdot \mathbf{s}' + \overline{\mathbf{H}}_c\right]_2 = \left[\overline{\mathbf{A}} \cdot (\mathbf{s}' + \mathbf{W}_c + \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{R}}_c)\right]_2 = \left[\overline{\mathbf{A}} \cdot \mathbf{s}\right]_2$, where $\mathbf{s} := \mathbf{s}' + \mathbf{W}_c + \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{R}}_c = \mathsf{TRF}(\mathsf{m}) + \mathbf{W}_{\alpha(\mathsf{m}_{|\zeta,\eta-1})} + \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{R}}_{\alpha(\mathsf{m}_{|\zeta,\eta-1})}$ is also a truly random function of m. For message m with $\mathsf{m}_{\zeta,\eta} = 1 - b_{\zeta,\eta}$, we have that

$$\begin{aligned}
[u]_2 &= \left[\sum_{(i,j)\neq(\zeta,\eta)} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \mathbf{t} + \mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1}) + \mathbf{r}^\top \cdot (\mathbf{A} \cdot \mathbf{s}' + \mathbf{H}_c)\right]_2 \\
&= \left[\sum_{(i,j)\neq(\zeta,\eta)} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \mathbf{t} + \mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1}) + \mathbf{r}^\top \cdot \mathbf{A} \cdot (\mathbf{s}' + \mathbf{W}_c) + \mathbf{r}^\top \cdot \mathbf{R}_c\right]_2 \\
&= \left[\sum_{(i,j)\neq(\zeta,\eta)} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \mathbf{t} + \mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1}) + \underbrace{\mathbf{r}^\top \mathbf{A} \overline{\mathbf{A}}^{-1}}_{\mathbf{x}_{\zeta,\eta}^{(1-b_{\zeta,\eta})\top}} \cdot \underbrace{\overline{\mathbf{A}}(\mathbf{s}' + \mathbf{W}_c)}_{\mathbf{t}-\overline{\mathbf{R}}_c} + \mathbf{r}^\top \mathbf{R}_c\right]_2 \\
&= \left[\sum_{i=1}^{p(\mathsf{m})} \sum_{j=1}^{\lambda} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \mathbf{t} + \mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1}) + \mathbf{r}^\top \cdot (\mathbf{R}_c - \mathbf{A} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{R}}_c)\right]_2 \\
&= \left[\sum_{i=1}^{p(\mathsf{m})} \sum_{j=1}^{\lambda} \mathbf{x}_{i,j}^{(\mathsf{m}_{i,j})^\top} \cdot \mathbf{t} + \mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1}) + \underline{\mathbf{r}} \cdot (\underline{\mathbf{R}}_c - \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{R}}_c)\right]_2.
\end{aligned}$$

- Case 1: $\mathbf{H} = \mathbf{A} \cdot \mathbf{W} + \mathbf{R}$ for $\mathbf{R} = \mathbf{0}$. Then $\underline{\mathbf{r}} \cdot (\underline{\mathbf{R}}_c - \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{R}}_c) = 0$, and it also uses the randomness $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1})$ to compute $[u]_2$ for message m with $\mathsf{m}_{\zeta,\eta} = 1 - b_{\zeta,\eta}$, as in $\mathsf{G}_{\zeta,\eta-1}$. Furthermore in this case, $b_{\zeta,\eta}$ is completely hidden from the point of view of $\mathcal{A}$, thus $\mathcal{A}$ can submit a message m^* in $\textsc{Chal}(\mathsf{m}^*)$ such that $|\mathsf{m}^*| < (\zeta-1)\lambda + \eta \vee \mathsf{m}^*_{\zeta,\eta} = b_{\zeta,\eta}$, i.e., $\overline{\mathsf{abort}}$ occurs, with probability at least $1/2$. Therefore,

$$\Pr\left[\overline{\mathsf{abort}} \mid \text{Case 1}\right] \geq 1/2. \tag{17}$$

- Case 2: $\mathbf{H} = \mathbf{A} \cdot \mathbf{W} + \mathbf{R}$ for $\mathbf{R} \leftarrow_\$ \mathbb{Z}_q^{(k+1)\times Q}$. Then with probability $(1 - 1/q)$, $\underline{\mathbf{r}}$ is non-zero, thus $\underline{\mathbf{r}} \cdot (\underline{\mathbf{R}}_c - \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{R}}_c)$ is uniformly random due to the randomness of $\underline{\mathbf{R}}_c$. In this case, an independent randomness $\mathsf{RT}(\mathsf{m}_{|\zeta,\eta}) := \mathsf{RT}(\mathsf{m}_{|\zeta,\eta-1}) + \underline{\mathbf{r}} \cdot (\underline{\mathbf{R}}_c - \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \cdot \overline{\mathbf{R}}_c)$ is employed for m with $\mathsf{m}_{\zeta,\eta} = 1 - b_{\zeta,\eta}$, as in $\mathsf{G}_{\zeta,\eta}$.

In $\textsc{Chal}(\mathsf{m}^*)$, if $|\mathsf{m}^*| < (\zeta-1)\lambda + \eta \vee \mathsf{m}^*_{\zeta,\eta} = b_{\zeta,\eta}$, note that $\mathsf{RT}(\mathsf{m}^*_{|\zeta,\eta-1}) = \mathsf{RT}(\mathsf{m}^*_{|\zeta,\eta})$, $\mathcal{B}_1, \mathcal{B}_2$ use the randomness $\mathsf{RT}(\mathsf{m}^*_{|\zeta,\eta-1})$, which equals $\mathsf{RT}(\mathsf{m}^*_{|\zeta,\eta})$, to compute h_1 perfectly as in $\mathsf{G}_{\zeta,\eta-1}$ and $\mathsf{G}_{\zeta,\eta}$. Meanwhile, if $|\mathsf{m}^*| \geq (\zeta-1)\lambda + \eta \wedge \mathsf{m}^*_{\zeta,\eta} = 1 - b_{\zeta,\eta}$, $\mathcal{B}_1, \mathcal{B}_2$ abort the game played with $\mathcal{A}$ immediately and set $\mathsf{abort} = \mathsf{true}$.

In summary, if $\overline{\mathsf{abort}}$ occurs, then with probability $(1 - 1/q)$, if it is the Case 1: $\mathbf{H} = \mathbf{A} \cdot \mathbf{W} + \mathbf{R}$ for $\mathbf{R} = \mathbf{0}$ (resp., Case 2: $\mathbf{H} = \mathbf{A} \cdot \mathbf{W} + \mathbf{R}$ for $\mathbf{R} \leftarrow_\$ \mathbb{Z}_q^{(k+1)\times Q}$), $\mathcal{B}_1, \mathcal{B}_2$ perfectly simulate game $\mathsf{G}_{\zeta,\eta-1}$ (resp., game $\mathsf{G}_{\zeta,\eta}$) with $\mathcal{A}$. In $\textsc{Finalize}$, $\mathcal{B}_1$'s strategy is to return 1 to its $\mathcal{D}_k$-MDDH challenger if and only if $\overline{\mathsf{abort}}$ occurs, and $\mathcal{B}_2$'s strategy is to return 1 to its $\mathcal{D}_k$-MDDH challenger if and only if $\overline{\mathsf{abort}}$ occurs and $\beta' = \beta$ holds (i.e., Win occurs). Thus we have the following equations

$$\begin{aligned}
&\mathsf{Adv}^{Q,\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_2,\mathcal{B}_1}(\lambda) = \left|\Pr\left[\overline{\mathsf{abort}} \mid \text{Case 1}\right] - \Pr\left[\overline{\mathsf{abort}} \mid \text{Case 2}\right]\right|,\\
&\mathsf{Adv}^{Q,\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_2,\mathcal{B}_2}(\lambda) = \left|\Pr\left[\overline{\mathsf{abort}} \wedge \mathsf{Win} \mid \text{Case 1}\right] - \Pr\left[\overline{\mathsf{abort}} \wedge \mathsf{Win} \mid \text{Case 2}\right]\right|\\
&= \Big|\Pr\left[\overline{\mathsf{abort}} \mid \text{Case 1}\right] \cdot \Pr\left[\mathsf{Win} \mid \text{Case 1} \wedge \overline{\mathsf{abort}}\right]\\
&\quad - \Pr\left[\overline{\mathsf{abort}} \mid \text{Case 2}\right] \cdot \Pr\left[\mathsf{Win} \mid \text{Case 2} \wedge \overline{\mathsf{abort}}\right]\Big|\\
&\geq (1 - 1/q) \cdot \left|\Pr\left[\overline{\mathsf{abort}} \mid \text{Case 1}\right] \cdot \Pr_{\zeta,\eta-1}[\mathsf{Win}] - \Pr\left[\overline{\mathsf{abort}} \mid \text{Case 2}\right] \cdot \Pr_{\zeta,\eta}[\mathsf{Win}]\right|\\
&\geq (1 - 1/q) \cdot \Pr\left[\overline{\mathsf{abort}} \mid \text{Case 1}\right] \cdot \left|\Pr_{\zeta,\eta-1}[\mathsf{Win}] - \Pr_{\zeta,\eta}[\mathsf{Win}]\right|\\
&\quad -(1 - 1/q) \cdot \left|\Pr\left[\overline{\mathsf{abort}} \mid \text{Case 1}\right] - \Pr\left[\overline{\mathsf{abort}} \mid \text{Case 2}\right]\right| \cdot \Pr_{1,\eta}[\mathsf{Win}]\\
&\overset{(17)}{\geq} \frac{1}{2} \cdot \frac{1}{2} \cdot \left|\Pr_{\zeta,\eta-1}[\mathsf{Win}] - \Pr_{\zeta,\eta}[\mathsf{Win}]\right| - \left|\Pr\left[\overline{\mathsf{abort}} \mid \text{Case 1}\right] - \Pr\left[\overline{\mathsf{abort}} \mid \text{Case 2}\right]\right|.
\end{aligned}$$

By combining the above two equations, we get that

$$\left|\Pr_{\zeta,\eta-1}[\mathsf{Win}] - \Pr_{\zeta,\eta}[\mathsf{Win}]\right| \leq 4 \cdot \left(\mathsf{Adv}^{Q,\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_2,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{Q,\mathcal{D}_k-mddh}_{\mathsf{PGGen},\mathbb{G}_2,\mathcal{B}_2}(\lambda)\right).$$

□

Appendix 3: Extractable IBE from our IBE

Extractable IBE and its IND-ID-CCA2 security

We review the definition of extractable IBE from [27].

Definition 12 (*Extractable identity-based encryption*) An extractable identity-based encryption (extractable IBE) scheme $\mathsf{IBE}_{ex} = (\mathsf{Gen}_{ex}, \mathsf{USKGen}_{ex}, \mathsf{Enc}_{ex}, \mathsf{Dec}_{ex})$ consists of the following four PPT algorithms:

- $\mathsf{Gen}_{ex}(1^\lambda)$ takes as input a security parameter λ. It generates a public key pk and a master secret key msk. The public key pk defines an identity space $\mathcal{ID}$, a ciphertext space $\mathcal{C}$ and a session key space $\mathcal{K}$.
- $\mathsf{USKGen}_{ex}(\mathsf{msk}, \mathsf{id})$ takes as input the master secret key msk and an identity $\mathsf{id} \in \mathcal{ID}$. It produces a user secret key $\mathsf{usk}[\mathsf{id}]$ for id.
- $\mathsf{Enc}_{ex}(\mathsf{pk}, \mathsf{id}, \theta)$ takes as input the public key pk, an identity $\mathsf{id} \in \mathcal{ID}$ and a bit $\theta \in \{0, 1\}$. It outputs a ciphertext $\mathsf{CT} \in \mathcal{C}$ if $\theta = 0$, and outputs a ciphertext and a session key $(\mathsf{CT}, K) \in \mathcal{C} \times \mathcal{K}$ if $\theta = 1$.
- $\mathsf{Dec}_{ex}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{CT})$ takes as input a user secret key $\mathsf{usk}[\mathsf{id}]$, an identity $\mathsf{id} \in \mathcal{ID}$ and a ciphertext $\mathsf{CT} \in \mathcal{C}$. It outputs a bit $\theta' \in \{0, 1\}$ and a session key $K' \in \mathcal{K}$.

Correctness

An extractable IBE scheme has completeness error ϵ, if for all $\lambda \in \mathbb{N}$, $(\mathsf{pk}, \mathsf{msk}) \leftarrow_{\$} \mathsf{Gen}_{ex}(1^\lambda)$, $\mathsf{id} \in \mathcal{ID}$, $\mathsf{usk}[\mathsf{id}] \leftarrow_{\$} \mathsf{USKGen}_{ex}(\mathsf{msk}, \mathsf{id})$, $\theta \in \{0, 1\}$, $\mathsf{CT}/(\mathsf{CT}, K) \leftarrow_{\$} \mathsf{Enc}_{ex}(\mathsf{pk}, \mathsf{id}, \theta)$ and $(\theta', K') \leftarrow \mathsf{Dec}_{ex}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{CT})$:

- The probability that $\theta' = \theta$ is at least $1 - \epsilon$, where the probability is taken over the coins used in Enc_{ex}.
- If $\theta = 1$ then $\theta' = \theta$ and $K' = K$. If $\theta' = 0$, K' is uniformly distributed in $\mathcal{K}$.

Security

The IND-ID-CCA2 security of extractable IBE is a combination of IND-ID-CCA2 security of one-bit IBE and IND-ID-CCA2 security of identity-based key encapsulation mechanism (IBKEM). The security notion is defined by game in Fig. 11.

Definition 13 (*IND-ID-CCA2 security for extractable IBE*) An extractable identity-based encryption scheme IBE_{ex} is IND-ID-CCA2 secure, if for any PPT adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{ind\text{-}id\text{-}cca2}_{\mathsf{IBE}_{ex},\mathcal{A}}(\lambda) := |\Pr[\mathsf{IND\text{-}ID\text{-}CCA2}^{\mathcal{A}} \Rightarrow 1] - 1/2|$ is negligible in λ, where game $\mathsf{IND\text{-}ID\text{-}CCA2}$ is specified in Fig. 11.

Construction of extractable IBE from our IBE

Our IBE $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k] = (\mathsf{Gen}, \mathsf{USKGen}, \mathsf{Enc}, \mathsf{Dec})$ in Fig. 6 which is based on the $\mathcal{D}_k$-MDDH assumption can be converted into an extractable IBE $\mathsf{IBE}_{ex} = (\mathsf{Gen}_{ex}, \mathsf{USKGen}_{ex}, \mathsf{Enc}_{ex}, \mathsf{Dec}_{ex})$, as shown in Fig. 12.

The resulting extractable IBE IBE_{ex} has completeness error $2^{-\lambda}$. If $\theta = 1$, the decryption algorithm always undoes the encryptions, due to the perfect correctness of $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k]$. If $\theta = 0$, $\mathsf{CT} = \langle \mathsf{C}, \chi \rangle = \langle [\mathbf{c}_0]_1, [\mathbf{c}_1]_1, R_{\mathsf{CH}}, \chi_1, \chi_2 \rangle$ is random. Hence

Procedure INITIALIZE:
$(\mathsf{pk}, \mathsf{msk}) \leftarrow\$ \mathsf{Gen}_{ex}(1^\lambda)$.
$\beta \leftarrow\$ \{0,1\}$. // challenge bit
Return pk.

Procedure ENC(id^*): // one query
If $\mathsf{id}^* \in \mathcal{Q}_{\mathcal{USK}}$, Return $\perp$.
If $\beta = 1$
  $(\mathsf{CT}^*, K^*) \leftarrow\$ \mathsf{Enc}_{ex}(\mathsf{pk}, \mathsf{id}^*, 1)$.
If $\beta = 0$
  $\mathsf{CT}^* \leftarrow\$ \mathsf{Enc}_{ex}(\mathsf{pk}, \mathsf{id}^*, 0)$, $K^* \leftarrow\$ \mathcal{K}$.
Return (CT^*, K^*).

Procedure USKGEN(id):
If $\mathsf{id} = \mathsf{id}^*$, Return $\perp$.
$\mathcal{Q}_{\mathcal{USK}} := \mathcal{Q}_{\mathcal{USK}} \cup \{\mathsf{id}\}$.
$\mathsf{usk}[\mathsf{id}] \leftarrow\$ \mathsf{USKGen}_{ex}(\mathsf{msk}, \mathsf{id})$.
Return $\mathsf{usk}[\mathsf{id}]$.

Procedure DEC(id, CT):
If $(\mathsf{id}, \mathsf{CT}) = (\mathsf{id}^*, \mathsf{CT}^*)$, Return $\perp$.
$\mathsf{usk}[\mathsf{id}] \leftarrow\$ \mathsf{USKGen}_{ex}(\mathsf{msk}, \mathsf{id})$.
$(\theta, K) \leftarrow\$ \mathsf{Dec}_{ex}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{CT})$.
Return $(\theta \in \{0,1\}, K)$.

Procedure FINALIZE(β'):
Return $(\beta' = \beta)$.

Fig. 11 IND-ID-CCA2 security game for IBE_{ex}

$(\mathsf{pk}, \mathsf{msk}) \leftarrow\$ \mathsf{Gen}_{ex}(1^\lambda)$:
  $(\mathsf{pk}, \mathsf{msk}) \leftarrow\$ \mathsf{Gen}(1^\lambda)$.
Return $(\mathsf{pk}, \mathsf{msk})$.

$(\mathsf{CT}, K)/\mathsf{CT} \leftarrow\$ \mathsf{Enc}_{ex}(\mathsf{pk}, \mathsf{id}, \theta \in \{0,1\})$:
  If $\theta = 1$, $m \leftarrow\$ \{0,1\}^\lambda$,
    $\langle \mathsf{C}, \chi \rangle \leftarrow\$ \mathsf{Enc}(\mathsf{pk}, \mathsf{id}, m)$,
    $\mathsf{CT} := \langle \mathsf{C}, \chi \rangle$, $K := m$,
    Return (CT, K).
  If $\theta = 0$, choose a random $\mathsf{CT} = \langle \mathsf{C}, \chi \rangle$,
    Return CT.

$\mathsf{usk}[\mathsf{id}] \leftarrow \mathsf{USKGen}_{ex}(\mathsf{msk}, \mathsf{id})$:
  $\mathsf{usk}[\mathsf{id}] \leftarrow \mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$.
Return $\mathsf{usk}[\mathsf{id}]$.

$(\theta', K) \leftarrow\$ \mathsf{Dec}_{ex}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{CT})$:
  Parse $\mathsf{CT} = \langle \mathsf{C}, \chi \rangle$.
  $m' \leftarrow \mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \langle \mathsf{C}, \chi \rangle)$.
  If $m' = \perp$,
    choose $K \leftarrow\$ \{0,1\}^\lambda$.
    Return $(0, K)$.
  Else $K := m'$, Return $(1, K)$.

Fig. 12 Construction of Extractable IBE IBE_{ex} from $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k]$

in the Dec algorithm of $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k]$, $\chi_2 = k_2 \cdot \chi_1 + k_3$ will hold with probability at most $2^{-\lambda}$. So Dec_{ex} will output $\theta' = 0$, when CT is an encryption of $\theta = 0$, except with probability at most $2^{-\lambda}$.

Theorem 3 *If* $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k]$ *is PR-ID-CCA2 secure, then the extractable IBE scheme* IBE_{ex} *in Fig. 12 is IND-ID-CCA2 secure.*

More precisely, suppose that $\mathcal{A}$ *is a PPT adversary against the IND-ID-CCA2 security of* IBE_{ex}*, then there exists a PPT adversary* $\mathcal{B}$ *against the PR-ID-CCA2 security of* $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k]$*, such that*

$$\mathsf{Adv}^{pr\text{-}id\text{-}cca2}_{\mathsf{IBE}[\mathsf{MAC},\mathsf{CH},\mathcal{D}_k],\mathcal{B}}(\lambda) = \mathsf{Adv}^{ind\text{-}id\text{-}cca2}_{\mathsf{IBE}_{ex},\mathcal{A}}(\lambda).$$

Proof of Theorem 3 The PR-ID-CCA2 adversary $\mathcal{B}$ of $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k]$ will invoke $\mathcal{A}$ to guess bit β. To do so, $\mathcal{B}$ simulates the IND-ID-CCA2 game for $\mathcal{A}$ as follows.

When the PR-ID-CCA2 challenger gives pk to $\mathcal{B}$, $\mathcal{B}$ forwards pk to $\mathcal{A}$. For all $\mathcal{A}$'s user secret key generation queries, $\mathcal{B}$ will query its own user secret key generation oracles for answers. Since IBE_{ex} and $\mathsf{IBE}[\mathsf{MAC}, \mathsf{CH}, \mathcal{D}_k]$ share the same user secret key generation algorithm, the simulation is perfect for $\mathcal{A}$.

For each decryption query $\mathsf{CT} = \langle \mathsf{C}, \chi \rangle$ from $\mathcal{A}$, $\mathcal{B}$ will query its own decryption oracle and get $m' \leftarrow \mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \langle \mathsf{C}, \chi \rangle)$. If $m' = \perp$, $\mathcal{B}$ chooses $K \leftarrow_{\$} \{0,1\}^{\lambda}$ and returns $(0, K)$ to $\mathcal{A}$. Otherwise $\mathcal{B}$ sets $K := m'$, and returns $(1, K)$. Clearly, $\mathcal{B}$ gives a perfect simulation of decryption oracle for $\mathcal{A}$.

When $\mathcal{A}$ submits a challenge identity id^*, $\mathcal{B}$ will choose a random message $m^* \leftarrow_{\$} \{0,1\}^{\lambda}$ and forward (id^*, m^*) to its own challenger. Then $\mathcal{B}$ will obtain a challenge $\langle \mathsf{C}^*, \chi^* \rangle$, which is either the output of $\mathsf{Enc}(\mathsf{pk}, \mathsf{id}, m^*)$ (when $\beta = 1$) or randomly chosen (when $\beta = 0$). $\mathcal{B}$ sends $(\mathsf{CT}^* := \langle \mathsf{C}^*, \chi^* \rangle, K^* := m^*)$ to $\mathcal{A}$.

- If $\langle \mathsf{C}^*, \chi^* \rangle = \mathsf{Enc}(\mathsf{pk}, \mathsf{id}, m^*)$, $(\mathsf{CT}^* = \langle \mathsf{C}^*, \chi^* \rangle, K^* = m^*)$ corresponds to an encryption of $\beta = 1$ for IBE_{ex}.
- If $\mathsf{CT}^* = \langle \mathsf{C}^*, \chi^* \rangle$ is randomly chosen, CT^* corresponds to an encryption of $\beta = 0$ for IBE_{ex}. In this case CT^* and $K^* := m^*$ are both independently and randomly chosen.

Hence $\mathcal{B}$ perfectly simulates the challenge for $\mathcal{A}$. Finally, $\mathcal{B}$ returns the guessing bit β' of $\mathcal{A}$ to its own challenger. Then $\mathcal{B}$ has the same advantage as $\mathcal{A}$. □

References

1. Abdalla M., Bellare M., Catalano D., Kiltz E., Kohno T., Lange T., Malone-Lee J., Neven G., Paillier P., Shi H.: Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions. In: Shoup, V. (ed.) CRYPTO 2005, pp. 205–222 (2005).
2. Agrawal S., Boneh D., Boyen X.: Efficient lattice (H)IBE in the standard model. In: Gilbert H. (ed.) EUROCRYPT 2010, pp. 553–572 (2010).
3. Attrapadung N., Hanaoka G., Yamada S.: A framework for identity-based encryption with almost tight security. In: Iwata T., Cheon J.H. (eds.) ASIACRYPT 2015, Part I, pp. 521–549. Springer, Heidelberg (2015).
4. Bellare M., Goldwasser S.: New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In: Brassard, G. (ed.) CRYPTO 1989, pp. 194–211 (1989).
5. Bellare M., Waters B., Yilek S.: Identity-based encryption secure against selective opening attack. In: Ishai Y. (ed.) TCC 2011, pp. 235–252 (2011).
6. Blazy O., Kiltz E., Pan J.: (Hierarchical) identity-based encryption from affine message authentication. In: Garay J.A., Gennaro R. (eds.) CRYPTO 2014, Part I, pp. 408–425 (2014).
7. Boneh D., Boyen X.: Secure identity based encryption without random oracles. In: Franklin M.K. (ed.) CRYPTO 2004, pp. 443–459 (2004).
8. Boneh D., Franklin M.K.: Identity-based encryption from the weil pairing. In: Kilian J. (ed.) CRYPTO 2001, pp. 213–229 (2001).
9. Boneh D., Crescenzo G.D., Ostrovsky R., Persiano G.: Public key encryption with keyword search. In: Cachin C., Camenisch J. (eds.) EUROCRYPT 2004, pp. 506–522 (2004).
10. Boyen X., Waters B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork C. (ed.) CRYPTO 2006, pp. 290–307 (2006).
11. Canetti R., Halevi S., Katz J.: Chosen-ciphertext security from identity-based encryption. In: Cachin C., Camenisch J. (eds.) EUROCRYPT 2004, pp. 207–222 (2004).
12. Chen J., Wee H.: Fully, (almost) tightly secure IBE and dual system groups. In: Canetti R., Garay J.A. (eds.) CRYPTO 2013, Part II, pp. 435–460 (2013).
13. Cocks C.: An identity based encryption scheme based on quadratic residues. In: Honary B. (ed.) Cryptography and Coding, pp. 360–363 (2001).
14. Döttling N., Schröder D.: Efficient pseudorandom functions via on-the-fly adaptation. In: Gennaro R., Robshaw M. (eds.) CRYPTO 2015, Part I, pp. 329–350 (2015).
15. Escala A., Herold G., Kiltz E., Ràfols C., Villar J.L.: An algebraic framework for diffie-hellman assumptions. In: Canetti R., Garay J.A. (eds.) CRYPTO 2013, Part II, pp. 129–147 (2013).
16. Fehr S., Hofheinz D., Kiltz E., Wee H.: Encryption schemes secure against chosen-ciphertext selective opening attacks. In: Gilbert H. (ed.) EUROCRYPT 2010, pp. 381–402 (2010).
17. Gentry C.: Practical identity-based encryption without random oracles. In: Vaudenay S. (ed.) EUROCRYPT 2006, pp. 445–464 (2006).
18. Goldreich O., Goldwasser S., Micali S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986).

19. Gong J., Chen J., Dong X., Cao Z., Tang S.: Extended nested dual system groups, revisited. In: Cheng C., Chung K., Persiano G., Yang B. (eds.) PKC 2016, Part I, pp. 133–163 (2016).
20. Groth J., Sahai A.: Efficient non-interactive proof systems for bilinear groups. In: Smart N.P. (ed.) EUROCRYPT 2008, pp. 415–432. Springer, Heidelberg (2008).
21. Håstad J., Impagliazzo R., Levin L.A., Luby M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999).
22. Hofheinz D.: All-but-many lossy trapdoor functions. In: Pointcheval D., Johansson T. (eds.) EUROCRYPT 2012, pp. 209–227 (2012).
23. Hofheinz D., Kiltz E.: Secure hybrid encryption from weakened key encapsulation. In: Menezes A. (ed.) CRYPTO 2007, pp. 553–571 (2007).
24. Hofheinz D., Koch J., Striecks C.: Identity-based encryption with (almost) tight security in the multi-instance, multi-ciphertext setting. In: Katz J. (ed.) PKC 2015, pp. 799–822 (2015).
25. Hohenberger S., Waters B.: Realizing hash-and-sign signatures under standard assumptions. In: Joux A. (ed.) EUROCRYPT 2009, pp. 333–350 (2009).
26. Krawczyk H., Rabin T.: Chameleon signatures. In: NDSS 2000 (2000).
27. Lai J., Deng R.H., Liu S., Weng J., Zhao Y.: Identity-based encryption secure against selective opening chosen-ciphertext attack. In: Nguyen P.Q., Oswald E. (eds.) EUROCRYPT 2014, pp. 77–92 (2014).
28. Lewko A.B.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: Pointcheval D., Johansson T. (eds.) EUROCRYPT 2012, pp. 318–335 (2012).
29. Naor M., Reingold O.: Number-theoretic constructions of efficient pseudo-random functions. J. ACM **51**(2), 231–262 (2004).
30. Shamir A.: Identity-based cryptosystems and signature schemes. In: Blakley G.R., Chaum D. (eds.) CRYPTO 1984, pp. 47–53 (1984).
31. Waters B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005, pp. 114–127 (2005).
32. Waters B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In: Halevi S. (ed.) CRYPTO 2009, pp. 619–636 (2009).
33. Wegman M.N., Carter L.: New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci. **22**(3), 265–279 (1981).