Downgradable Identity-based Encryption and Applications

Olivier Blazy[†], Paul Germouty[‡], and Duong Hieu Phan[†]

† Université de Limoges, XLim, France
‡

Abstract. In Identity-based cryptography, in order to generalize one receiver encryption to multi-receiver encryption, wildcards were introduced: WIBE enables wildcard in receivers' pattern and Wicked-IBE allows one to generate a key for identities with wildcard. However, the use of wildcard makes the construction of WIBE, Wicked-IBE more complicated and significantly less efficient than the underlying IBE. The main reason is that the conventional identity's binary alphabet is extended to a ternary alphabet $\{0, 1, *\}$ and the wildcard $*$ is always treated in a convoluted way in encryption or in key generation. In this paper, we show that when dealing with multi-receiver setting, wildcard is not necessary. We introduce a new downgradable property for IBE scheme and show that any IBE with this property, called DIBE, can be efficiently transformed into WIBE or Wicked-IBE.
While WIBE and Wicked-IBE have been used to construct Broadcast encryption, we go a step further by employing DIBE to construct Attribute-based Encryption of which the access policy is expressed as a boolean formula in the disjunctive normal form.

Keywords. Identity-Based Encryption, Attribute-Based Encryption.

1 Introduction

Identity-based encryption (IBE) is a concept introduced by Shamir in [Sha84] allowing encrypting for a specific recipient using solely his identity (for example an email address or phone number) instead of public key. Decryption is done by using a user secret key for the said identity, obtained via a trusted authority. This concept avoids the use of Public Key Infrastructure in order to get a user's public key securely. This was the main argument to build such scheme, however a lot of works expose the fact that Identity-based Encryption schemes can be used to build other primitives like Adaptive Oblivious Transfer [GH07, BCG16].

The first instantiations of an IBE scheme arose in 2001 [Coc01, BF01, SOK00]. It was only in 2005 in [Wat05], that the first construction, with adaptive security in the standard model, was proposed. Adaptive security meaning that an adversary may select the challenge identity id^* after seeing the public key and arbitrarily many user secret keys for identities of his choice. The concept of IBE generalizes naturally to hierarchical IBE (HIBE). In an L-level HIBE, hierarchical identities are vectors of identities of maximal length L and user secret keys for a hierarchical identity can be delegated. An IBE is simply a L-level HIBE with $L = 1$.

From one receiver to multi-receiver setting: introduction of wildcard. As in the case of public-key encryption, passing from one receiver setting to multi-receiver setting is an important step. For this aim, wildcard IBE (WIBE) was introduced in [ACD+06] where the wildcard symbol (*) is added in identities to encrypt for a broad range of users at once. Along the same line, another generalization called WKD-IBE [AKN07] allows joker (*) symbol in users' secret keys to decrypt several targeted identities with a single key. Many others primitives, namely identity-based broadcast encryption [AKN07], identity-based traitor tracing [ADML+07], identity-based trace and revoke [PT11] schemes can be then constructed from WIBE and WKD-IBE.

Is wildcard really necessary for the multi-receiver setting? While the introduction of wildcard is very interesting, it makes the construction of WIBE, Wicked-IBE more complicated and thus less efficient than the underlying IBE. Basically the alphabet is extended from a conventional binary alphabet to a ternary alphabet $\{0, 1, *\}$ and the wildcard $*$ is treated in a special and different way than $\{0, 1\}$. Beside the efficiency, there is often a significant loss in reducing the security of the WIBE, Wicked-IBE to the underlying IBE.

We are thus interested in the following question: can we avoid wildcard in considering IBE in multi-receiver setting? This paper gives the positive answer. We propose a new property for IBE, called downgradable IBE (DIBE). While keeping the binary alphabet unchanged, we show that downgradable IBE is not less powerful than the other wildcard based IBE: efficient transformations from downgradable IBE to wildcard based IBE schemes will be given.

Interestingly, avoiding wildcard helps us to get very efficient constructions. We simply need to show that the downgradable property can be obtained from existing constructions. A recent paper [KLLO18] found instantiations for Wicked-IBE and wildcarded IBE with good improve of the previous schemes, showing the interest of the research for this subject. Our instantiation of DIBE, once transformed into WIBE or Wkd-IBE is even more efficient allowing a constant size ciphertext, a master public key linear in the size of the identity (instead of n^2) and is fully secure under the standard assumption DLin. Indirectly our instantiation also improve the identity-based broadcast encryption, identity-based traitor tracing, identity-based trace and revoke schemes which rely on the WIBE and Wicked-IBE.

Toward efficient transformations from DIBE to ABE. Attribute-Based Encryption (ABE), introduced by Sahai and Waters [SW05], is a generalization of both identity-based encryption and broadcast encryption. It gives a flexible way to define the target group of people who can receive the message: the target set can be defined in a more structural way via access policies on the user's attributes. While broadcast encryption can be obtained from WIBE, as far as we know, there is still no generic construction of ABE from any variant of IBE. We will show a transformation from DIBE to ABE where the access policies is in DNF.

In the papers [AKN07, FP12], they show how some variant of IBE, WKD-IBE for the first one and HIBE for the second one, can be used to create broadcast encryption. ABE encompass the notion of Broadcast Encryption, thus our work achieves the willing of constructing the complex primitive like ABE from the much more simple IBE.

1.1 This work

Downgradable IBE In this work we introduce the notion of *Downgradable Identity-based Encryption* (DIBE). A downgradable IBE is an identity-based encryption where a user possessing a key for an identity $\mathsf{usk}[id]$ can downgrade his key to any identity $\tilde{\mathsf{id}}$ with the restriction that he can only transform 1 into 0 in his identity string. More formally, the set $\tilde{\mathsf{ID}} = \{\tilde{\mathsf{id}} | \forall i, \tilde{\mathsf{id}}_i = 1 \Rightarrow \mathsf{id}_i = 1\}$.

From Downgradable IBE to HIBE, WIBE, WKD-IBE We later show that our new primitive encompasses other previous primitives, and that it can be tightly transformed into all of them. We then propose a generic framework, and an instantiation inspired by [BKP14], and show that thanks to our transform, we can obtain efficient WIBE, and WKD-IBE. This can be seen as a new method to design Wildcard-based IBE: one just need to prove the downgradable property of the IBE and then apply our direct transformation.

Moving to Attribute-Based Encryption. We also show how to generically transform a Downgradable IBE into an Attribute-based Encryption by using the properties of the DIBE and associating each attribute to a bit in the identity bit string. Our instantiation of DIBE lead to a secure ABE scheme with boolean formula in DNF.

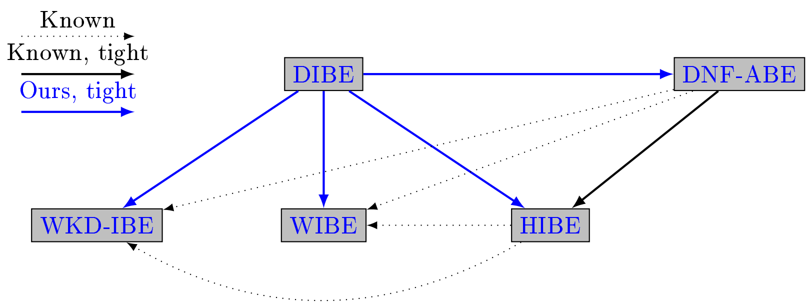

Fig. 1. Relations Between Primitives

1.2 Comparison to existing work

We propose a construction of DIBE inspired by the Hash-Proof based HIBE from [BKP14]. Interestingly, our construction combined with the WKD-DIBE, Wild-DIBE transformations are way more efficient than the existing WIBE and WKD-IBE. We compare them in figure 2, where we set the number of pattern and the size of the identity to the same value n, q_k correspond to the number adversary's key derivation queries. ℓ is the number of bits of identity that a user is allow to delegate a key to (e.g. his height in the hierarchical tree). A more detailed comparison can be found in section 7. The improvements both in term of security and efficiency make those schemes now more suitable for practical applications.

Name	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{usk}\rvert$	$\lvert\mathsf{C}\rvert$	assump.	Loss
WKD [AKN07]	$(n+1)n+3$	$n+2$	2	BDDH	$O(q_k^n)$
our WKD-DIBE	$4n+2$	$3n+5$	5	DLin (any $k-\mathsf{MDDH}$)	$O(q_k)$
WIBE [BDNS07]	$(n+1)n+3$	$n+1$	$(n+1)n+2$	BDDH	$O(n^2 q_k^n)$
our Wild-DIBE	$4n+2$	$3n+5$	5	DLin (any $k-\mathsf{MDDH}$)	$O(q_k)$

Fig. 2. Efficiency Comparison Between our Transformations and Previous Schemes

1.3 Open problems

We managed to create an efficient Ciphertext Policy Attribute-based Encryption for boolean formula in DNF. This improve our knowledge of the relation Between IBE and ABE. But finally how close IBE and ABE are? Is it possible to extend efficiently our idea to fit other/any kind of access structure.

2 Definitions

2.1 Notation

- If $\boldsymbol{x} \in \mathcal{BS}^n$, then $|\boldsymbol{x}|$ denotes the length n of the vector. Further, $x \xleftarrow{\$} \mathcal{BS}$ denotes the process of sampling an element x from set $\mathcal{BS}$ uniformly at random.
- If $\mathbf{A} \in \mathbb{Z}_p^{(k+1)\times n}$ is a matrix, then $\overline{\mathbf{A}} \in \mathbb{Z}_p^{k\times n}$ denotes the upper matrix of $\mathbf{A}$ and then $\underline{\mathbf{A}} \in \mathbb{Z}_p^{1\times k}$ denotes the last row of $\mathbf{A}$.
- We are going to define a relation $\preceq$ between two strings s, t of the same length ℓ, such that $s \preceq t$ if and only if $\forall i \in [\![1, \ell]\!], s[i] \leq t[i]$. As an extension, given a set S of strings of length ℓ and a similarly long string t, we are going to say that $t \preceq S$, if there exists $s \in S$ such that $t \preceq s$. One has to pay attention that $\preceq$ is not total, for example, 10 and 01 can not be compared.
  Similarly, we define a relation $\preceq_*$ between two strings s, t of the same length ℓ, such that $s \preceq_* t$ if and only if $\forall i \in [\![1, \ell]\!], s[i] \preceq t[i] \vee s[i] = *$.
- **Games.** We use games for our security reductions. A game G is defined by procedures $\mathsf{Initialize}$ and $\mathsf{Finalize}$, plus some optional procedures $\mathsf{P}_1, \ldots, \mathsf{P}_n$. All procedures are given using pseudo-code, where initially all variables are undefined. An adversary $\mathcal{A}$ is executed in game G if it first calls $\mathsf{Initialize}$, obtaining its output. Next, it may make arbitrary queries to P_i (according to their specification), again obtaining their output. Finally, it makes one single call to $\mathsf{Finalize}(\cdot)$ and stops. We define $\mathsf{G}^{\mathcal{A}}$ as the output of $\mathcal{A}$'s call to $\mathsf{Finalize}$.

2.2 Pairing groups and Matrix Diffie-Hellman Assumption

Let GGen be a probabilistic polynomial time (PPT) algorithm that on input $1^{\mathfrak{K}}$ returns a description $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, g_1, g_2, e)$ of asymmetric pairing groups where $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are cyclic groups of order q for a λ-bit prime q, g_1 and g_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and

$e : \mathbb{G}_1 \times \mathbb{G}_2$ is an efficiently computable (non-degenerated) bilinear map. Define $g_T := e(g_1, g_2)$, which is a generator in $\mathbb{G}_T$.

We use implicit representation of group elements as introduced in [EHK+13]. For $s \in \{1, 2, T\}$ and $a \in \mathbb{Z}_p$ define $[a]_s = g_s^a \in \mathbb{G}_s$ as the *implicit representation* of a in $\mathbb{G}_s$. More generally, for a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{Z}_p^{n \times m}$ we define $[\mathbf{A}]_s$ as the implicit representation of $\mathbf{A}$ in $\mathbb{G}_s$. Obviously, given $[a]_s \in \mathbb{G}_s$ and a scalar $x \in \mathbb{Z}_p$, one can efficiently compute $[ax]_s \in \mathbb{G}_s$. Further, given $[a]_1, [a]_2$ one can efficiently compute $[ab]_T$ using the pairing e. For $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}_p^k$ define $e([\boldsymbol{a}]_1, [\boldsymbol{b}]_2) := [\boldsymbol{a}^\top \boldsymbol{b}]_T \in \mathbb{G}_T$.

We recall the definition of the matrix Diffie-Hellman (MDDH) assumption [EHK+13].

Definition 1 (Matrix Distribution). *Let $k \in \mathbb{N}$. We call $\mathcal{D}_k$ a matrix distribution if it outputs matrices in $\mathbb{Z}_p^{(k+1) \times k}$ of full rank k in polynomial time.*

We assume the first k rows of $\mathbf{A} \xleftarrow{\$} \mathcal{D}_k$ form an invertible matrix. The $\mathcal{D}_k$-Matrix Diffie-Hellman problem is to distinguish the two distributions $([\mathbf{A}], [\mathbf{A}\boldsymbol{w}])$ and $([\mathbf{A}], [\boldsymbol{u}])$ where $\mathbf{A} \xleftarrow{\$} \mathcal{D}_k$, $\boldsymbol{w} \xleftarrow{\$} \mathbb{Z}_p^k$ and $\boldsymbol{u} \xleftarrow{\$} \mathbb{Z}_p^{k+1}$.

Definition 2 ($\mathcal{D}_k$-Matrix Diffie-Hellman Assumption $\mathcal{D}_k$-MDDH). *Let $\mathcal{D}_k$ be a matrix distribution and $s \in \{1, 2, T\}$. We say that the $\mathcal{D}_k$-Matrix Diffie-Hellman ($\mathcal{D}_k$-MDDH) Assumption holds relative to GGen in group $\mathbb{G}_s$ if for all PPT adversaries $\mathcal{D}$,*

$$\begin{aligned} &\mathbf{Adv}_{\mathcal{D}_k, \mathsf{GGen}}(\mathcal{D}) \\ := &\,|\Pr[\mathcal{D}(\mathcal{G}, [\mathbf{A}]_s, [\mathbf{A}\boldsymbol{w}]_s) = 1] - \Pr[\mathcal{D}(\mathcal{G}, [\mathbf{A}]_s, [\boldsymbol{u}]_s) = 1]| = \mathsf{negl}(\lambda), \end{aligned}$$

where the probability is taken over $\mathcal{G} \xleftarrow{\$} \mathsf{GGen}(1^\lambda)$, $\mathbf{A} \xleftarrow{\$} \mathcal{D}_k, \boldsymbol{w} \xleftarrow{\$} \mathbb{Z}_p^k, \boldsymbol{u} \xleftarrow{\$} \mathbb{Z}_p^{k+1}$. This assumption is Random Self Reducible.

2.3 Identity-based Key Encapsulation

We now recall syntax and security of IBE in terms of an ID-based key encapsulation mechanism IBKEM. Every IBKEM can be transformed into an ID-based encryption scheme IBE using a (one-time secure) symmetric cipher.

Definition 3 (Identity-based Key Encapsulation Scheme). *An identity-based key encapsulation (IBKEM) scheme IBKEM consists of four PPT algorithms $\mathsf{IBKEM} = (\mathsf{Gen}, \mathsf{USKGen}, \mathsf{Enc}, \mathsf{Dec})$ with the following properties.*

- *The probabilistic key generation algorithm* $\mathsf{Gen}(\mathfrak{K})$ *returns the (master) public/secret key* $(\mathsf{mpk}, \mathsf{msk})$. *We assume that* mpk *implicitly defines a message space* $\mathcal{M}$, *an identity space* ID, *a key space* $\mathcal{K}$, *and ciphertext space* CS.
- *The probabilistic user secret key generation algorithm* $\mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$ *returns the user secret-key* $\mathsf{usk}[\mathsf{id}]$ *for identity* $\mathsf{id} \in \mathsf{ID}$.
- *The probabilistic encapsulation algorithm* $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id})$ *returns the symmetric key* $\mathsf{sk} \in \mathcal{K}$ *together with a ciphertext* $\mathsf{C} \in \mathsf{CS}$ *with respect to identity* id.
- *The deterministic decapsulation algorithm* $\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{C})$ *returns the decapsulated key* $\mathsf{sk} \in \mathcal{K}$ *or the reject symbol* $\perp$.

For perfect correctness we require that for all $\mathfrak{K} \in \mathbb{N}$, *all pairs* $(\mathsf{mpk}, \mathsf{msk})$ *honestly generated by* $\mathsf{Gen}(\mathfrak{K})$, *all identities* $\mathsf{id} \in \mathsf{ID}$, *all* $\mathsf{usk}[\mathsf{id}]$ *generated by* $\mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$ *and all* $(\mathsf{sk}, \mathsf{C})$ *output by* $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id})$:

$$\Pr[\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{C}) = \mathsf{sk}] = 1.$$

The security requirements for an IBKEM we consider here are indistinguishability and anonymity against chosen plaintext and identity attacks (IND-ID-CPA and ANON-ID-CPA). Instead of defining both security notions separately, we define pseudorandom ciphertexts against chosen plaintext and identity attacks (PR-ID-CPA) which means that challenge key and ciphertext are both pseudorandom. Note that PR-ID-CPA trivially implies IND-ID-CPA and ANON-ID-CPA. We define PR-ID-CPA-security of IBKEM formally via the games given in Figure 3.

Procedure $\mathsf{Initialize}$:	**Procedure** $\mathsf{Enc}(\mathsf{id}^*)$: //one query
$(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{Gen}(\mathfrak{K})$	$(\mathsf{sk}^*, \mathsf{C}^*) \xleftarrow{\$} \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}^*)$
Return mpk	$\boxed{\mathsf{sk}^* \xleftarrow{\$} \mathcal{K}; \mathsf{C}^* \xleftarrow{\$} \mathsf{CS}}$
Procedure $\mathsf{USKGen}(\mathsf{id})$:	Return $(\mathsf{sk}^*, \mathsf{C}^*)$
$\mathcal{Q}_{\mathsf{ID}} = \mathcal{Q}_{\mathsf{ID}} \cup \{\mathsf{id}\}$	**Procedure** $\mathsf{Finalize}(\beta)$:
Return $\mathsf{usk}[\mathsf{id}] \xleftarrow{\$} \mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$	Return $(\mathsf{id}^* \notin \mathcal{Q}_{\mathsf{ID}}) \wedge \beta$

Fig. 3. Security Games $\mathsf{PR\text{-}ID\text{-}CPA}_{\mathsf{real}}$ and $\boxed{\mathsf{PR\text{-}ID\text{-}CPA}_{\mathsf{rand}}}$ for defining PR-ID-CPA-security.

Definition 4 ($\mathsf{PR\text{-}ID\text{-}CPA}$ **Security).** *An identity-based key encapsulation scheme* IBKEM *is* $\mathsf{PR\text{-}ID\text{-}CPA}$*-secure if for all PPT* $\mathcal{A}$*,* $\mathsf{Adv}^{\mathsf{pr\text{-}id\text{-}cpa}}_{\mathsf{IBKEM}}(\mathcal{A}) := |\Pr[\mathsf{PR\text{-}ID\text{-}CPA}^{\mathcal{A}}_{\mathsf{real}} \Rightarrow 1] - \Pr[\mathsf{PR\text{-}ID\text{-}CPA}^{\mathcal{A}}_{\mathsf{rand}} \Rightarrow 1]|$ *is negligible.*

3 Downgradable Identity-Based Encryption

In this section we introduce the notion of Downgradable Identity-Based Encryption. There is a lot of different variant of IBE in the nowadays, add another one seems to be not useful but we stress that our is not here to be used as a simple scheme but as a key pillar to create ABE from IBE. Also in section 4 we explain the relations between different variant of IBE and how DIBE can be transformed into them. For simplicity we are going to express in term of Key Encapsulation, as it can then be trivially transformed into an encryption.

Definition 5 (Downgradable Identity-based Key Encapsulation Scheme). *A Downgradable identity-based key encapsulation (DIBKEM) scheme* DIBKEM *consists of five PPT algorithms* $\mathsf{DIBKEM} = (\mathsf{Gen}, \mathsf{USKGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{USKDown})$ *with the following properties.*

- *The probabilistic key generation algorithm* $\mathsf{Gen}(\mathfrak{K})$ *returns the (master) public/secret key* $(\mathsf{mpk}, \mathsf{msk})$*. We assume that* mpk *implicitly defines a message space* $\mathcal{M}$*, an identity space* ID*, a key space* $\mathcal{K}$*, and ciphertext space* CS*.*
- *The probabilistic user secret key generation algorithm* $\mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$ *returns the user secret-key* $\mathsf{usk}[\mathsf{id}]$ *for identity* $\mathsf{id} \in \mathsf{ID}$*.*
- *The probabilistic encapsulation algorithm* $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id})$ *returns the symmetric key* $\mathsf{sk} \in \mathcal{K}$ *together with a ciphertext* $\mathsf{C} \in \mathsf{CS}$ *with respect to identity* id*.*
- *The deterministic decapsulation algorithm* $\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{C})$ *returns the decapsulated key* $\mathsf{sk} \in \mathcal{K}$ *or the reject symbol* $\perp$*.*
- *The probabilistic user secret key downgrade algorithm* $\mathsf{USKDown}(\mathsf{usk}[\mathsf{id}], \tilde{\mathsf{id}})$ *returns the user secret-key* $\mathsf{usk}[\tilde{\mathsf{id}}]$ *as long as* $\tilde{\mathsf{id}} \preceq \mathsf{id}$*.*

For perfect correctness we require that for all $\mathfrak{K} \in \mathbb{N}$*, all pairs* $(\mathsf{mpk}, \mathsf{msk})$ *honestly generated by* $\mathsf{Gen}(\mathfrak{K})$*, all identities* $\mathsf{id} \in \mathsf{ID}$*, all* $\mathsf{usk}[\mathsf{id}]$ *generated by* $\mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$ *and all* $(\mathsf{sk}, \mathsf{C})$ *output by* $\mathsf{Enc}(\mathsf{mpk}, \mathsf{id})$*:*

$$\Pr[\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{C}) = \mathsf{sk}] = 1.$$

We also require the distribution of $\mathsf{usk}[\tilde{\mathsf{id}}]$ *from* $\mathsf{USKDown}(\mathsf{usk}[\mathsf{id}], \tilde{\mathsf{id}})$ *to be identical to the one from* $\mathsf{USKGen}(\mathsf{msk}, \tilde{\mathsf{id}})$*.*

The security requirements we consider here are indistinguishability and anonymity against chosen plaintext and identity attacks (IND-ID-CPA and ANON-ID-CPA). Instead of defining both security notions separately, we define pseudorandom ciphertexts against chosen plaintext and identity attacks (PR-ID-CPA) which means that challenge key and ciphertext are both pseudorandom. We define PR-ID-CPA-security of DIBKEM formally via the games given in Figure 4.

Procedure Initialize: $(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{Gen}(\mathfrak{K})$ Return mpk **Procedure** USKGen(id): $\mathcal{Q}_{\mathsf{ID}} = \mathcal{Q}_{\mathsf{ID}} \cup \{\mathsf{id}\}$ Return $\mathsf{usk}[\mathsf{id}] \xleftarrow{\$} \mathsf{USKGen}(\mathsf{msk}, \mathsf{id})$	**Procedure** Enc(id*): //one query $(\mathsf{sk}^*, \mathsf{C}^*) \xleftarrow{\$} \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}^*)$ $\boxed{\mathsf{sk}^* \xleftarrow{\$} \mathcal{K}; \mathsf{C}^* \xleftarrow{\$} \mathsf{CS}}$ Return $(\mathsf{sk}^*, \mathsf{C}^*)$ **Procedure** Finalize(β): Return $(\neg(\mathsf{id}^* \preceq \mathcal{Q}_{\mathsf{ID}})) \wedge \beta$

Fig. 4. Security Games $\mathsf{PR\text{-}ID\text{-}CPA}_{\mathsf{real}}$ and $\boxed{\mathsf{PR\text{-}ID\text{-}CPA}_{\mathsf{rand}}}$ for defining PR-ID-CPA-security for DIBKEM.

Definition 6 (PR-ID-CPA **Security).** *A downgradable identity-based key encapsulation scheme* DIBKEM *is* PR-ID-CPA*-secure if for all PPT $\mathcal{A}$,* $\mathsf{Adv}^{\mathsf{pr\text{-}id\text{-}cpa}}_{\mathsf{DIBKEM}}(\mathcal{A}) := |\Pr[\mathsf{PR\text{-}ID\text{-}CPA}^{\mathcal{A}}_{\mathsf{real}} \Rightarrow 1] - \Pr[\mathsf{PR\text{-}ID\text{-}CPA}^{\mathcal{A}}_{\mathsf{rand}} \Rightarrow 1]|$ *is negligible.*

We stress the importance of the condition: $(\neg(\mathsf{id}^* \preceq \mathcal{Q}_{\mathsf{ID}}))$. This is here to guarantee that the adversary did not query an identity that can be downgraded to the challenge one, as this would allow for a trivial attack.

4 Transformation to classical primitives

Here, we are going to show how a Downgradable IBE relates to other primitives from the same family. Note that there is notions generalizing WIBE and WKD-IBE called WW-IBE described in [ACP12] and SWIBE described in [KLLO18] but their instantiation lead to not practical schemes. We can note that HIBE and WIBE have been linked in [AFL12]. In our work we are motivated in achieving a fully secure HIBE which would be inefficient using their construction.

4.1 From DIBE to WIBE

Wildcard Identity-Based Encryption is a concept introduced in [ACD$^+$06]. The idea is to be able to encrypt message for serveral identities by fixing some identity bits and letting others free (symbolized by the $*$). Thus only people with identity matching the one used to encrypt can decrypt. We say that id matches id' if $\forall i$ $\mathsf{id}_i = \mathsf{id}'_i$ or $\mathsf{id}'_i = *$. Detailed definitions are included in Appendix A

We are now given a $\mathsf{DIBKEM}(\mathsf{Gen}, \mathsf{USKGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{USKDown})$, let us show how to build the corresponding Wild-IBKEM.

As with all the following constructions, the heart of the transformation will be to use a DIBKEM for identity of size 2ℓ to handle identities of size ℓ.

Let's consider an identity wid of size ℓ, we define $\mathsf{id} = \phi(\mathsf{wid})$ as follows:

$$\mathsf{id}[2i, 2i+1] = \begin{cases} 01 & \text{if } \mathsf{wid}[i] = 0 \\ 10 & \text{if } \mathsf{wid}[i] = 1 \\ 00 & \text{otherwise.} \end{cases}$$

Now we can define :

- $\mathsf{WIBE.Gen}(\mathfrak{K}) : \mathsf{Gen}(\mathfrak{K})$, except that instead of defining ID as strings of size 2ℓ, we suppose the public key define WID of enriched identities of size ℓ.
- $\mathsf{WIBE.USKGen}(\mathsf{sk}, \mathsf{id}) = \mathsf{USKGen}(\mathsf{sk}, \phi(\mathsf{id}))$.
- $\mathsf{WIBE.Enc}(\mathsf{mpk}, \mathsf{id}) = \mathsf{Enc}(\mathsf{mpk}, \phi(\mathsf{id}))$.
- $\mathsf{WIBE.Dec}(\mathsf{usk}[\mathsf{id}], \hat{\mathsf{id}}, \mathsf{C})$ checks if $\hat{\mathsf{id}} \preceq \mathsf{id}$, then computes $\mathsf{usk}[\phi(\hat{\mathsf{id}})] = \mathsf{USKDown}(\mathsf{usk}[\phi(\mathsf{id})])$. Returns $\mathsf{Dec}(\mathsf{usk}[\phi(\hat{\mathsf{id}})], \hat{\mathsf{id}}, \mathsf{C})$ or rejects with $\perp$.

4.2 From DIBE to HIBE

Hierarchical Identity-Based Encryption is a concept introduced in [GS02]. The idea of this primitive is to introduce a hierarchy in the user secret key. A user can create a secret key from his one for any identity with prefix his own identity. Detailed definitions are included in Appendix A

This time, we are going to map the identity space to a bigger set, with joker identity that can be downgraded to both 0 or 1.

Let's consider an identity hid of size ℓ, we define $\mathsf{id} = \phi(\mathsf{hid})$ as follows:

$$\mathsf{id}[2i, 2i+1] = \begin{cases} 01 & \text{if } \mathsf{hid}[i] = 0 \\ 10 & \text{if } \mathsf{hid}[i] = 1 \\ 11 & \text{otherwise}(\mathsf{hid}[\mathsf{i}] = \perp). \end{cases}$$

Now we can define :

– $\mathsf{HIB.Gen}(\mathfrak{K})$: $\mathsf{Gen}(\mathfrak{K})$, except instead of defining ID as strings of size 2ℓ, we suppose the public key define HID of enriched identities of size ℓ.
– $\mathsf{HIB.USKGen}(\mathsf{sk}, \mathsf{id}) = \mathsf{USKGen}(\mathsf{sk}, \phi(\mathsf{id}))$. It should be noted that in case of an DIBKEM, some identities are never to be queried to the downgradable IBKEM: those with 00 is $2i, 2i+1$, or those with 11 at $2i, 2i+1$ and then a 0 (this would correspond to *punctured* identities).
– $\mathsf{HIB.USKDel}(\mathsf{usk}[\mathsf{id}], \mathsf{id} \in \mathcal{BS}^p, \mathsf{id}_{p+1}) = \mathsf{USKDown}(\mathsf{usk}[\phi(\mathsf{id})], \phi(\mathsf{id}||\mathsf{id}_{p+1}))$. By construction we have $\phi(\mathsf{id}||\mathsf{id}_{p+1}) \preceq \phi(\mathsf{id})$.
– $\mathsf{HIB.Enc}(\mathsf{mpk}, \mathsf{id}) = \mathsf{Enc}(\mathsf{mpk}, \phi(\mathsf{id}))$.
– $\mathsf{HIB.Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{C})$ returns $\mathsf{Dec}(\mathsf{usk}[\phi(\mathsf{id})], \phi(\mathsf{id}), \mathsf{C})$ or the reject symbol $\perp$.

4.3 From DIBE to Wicked IBE

The paper [AKN07] presents a variant of Identity-based Encryption called Wicked IBE (WKD-IBE). A wicked IBE or wildcard key derivation IBE is a generalization of the concept of limited delegation concept by Boneh-Boyen-Goh [BBG05].

This scheme allows secret key associated with a pattern $P = (P_1, ..., P_l) \in \{\{0,1\}^* \cup \{*\}\}^l$ to be delegated for a pattern $P' = (P'_1, ..., P'_{l'})$ that matches P. We say that P' match P if $\forall i \leq l'$ $P'_i = P_i$ or $P_i = *$ and $\forall l' + 1 \leq i \leq l$ $P_i = *$.

Here again, we are going to map the identity space to a bigger set.

Let's consider an identity id of size ℓ, we define $\mathsf{id} = \phi(\mathsf{wkdid})$ as follows:

$$\mathsf{id}[2i, 2i+1] = \begin{cases} 01 & \text{if } \mathsf{wkdid}[i] = 0 \\ 10 & \text{if } \mathsf{wkdid}[i] = 1 \\ 11 & \text{if } \mathsf{wkdid}[i] = * \end{cases}$$

Now we can define :
– $\mathsf{WKDIB.Gen}(\mathfrak{K})$: $\mathsf{Gen}(\mathfrak{K})$, except instead of defining ID as strings of size 2ℓ, we suppose the public key define WKDID of enriched identities of size ℓ.
– $\mathsf{WKDIB.USKGen}(\mathsf{msk}, \mathsf{id}) = \mathsf{USKGen}(\mathsf{msk}, \phi(\mathsf{id}))$. It should be noted that in case of an WKD-DIBE, some identities are never to be queried to the downgradable IBE: those with 00.
– $\mathsf{WKDIB.USKDel}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{id}') = \mathsf{USKDown}(\mathsf{usk}[\phi(\mathsf{id})], \phi(\mathsf{id}), \phi(\mathsf{id}'))$.
– $\mathsf{WKDIB.Enc}(\mathsf{mpk}, \mathsf{id}) = \mathsf{Enc}(\mathsf{mpk}, \phi(\mathsf{id}))$.
– $\mathsf{WKDIB.Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{C})$ returns $\mathsf{Dec}(\mathsf{usk}[\phi(\mathsf{id})], \phi(\mathsf{id}), \mathsf{C})$ or the reject symbol $\perp$.

Remark 7. It can be noted, that all those transformations end up using 4 bits instead to encode a ternary alphabet. So there is a bit wasted in every given transformation. This could easily be avoided by using a more convoluted encoding, however this is already enough to show the link between the construction;also, this allows to build a scheme both wicked and wildcarded.

4.4 From Wicked IBE to DIBE

We can easily transform a Wicked IBE scheme into DIBE by using only identity made of 0 and $*$. In fact the element 1 of the DIBE play the role of the $*$ of the Wicked IBE. Morally a DIBE can be seen as a Wicked IBE where the patterns are made of only 2 distinct elements instead of 3.

5 ABE

In this section, we consider Attribute Based Encryption (ABE) and present a transformation from DIBE to ABE. We recall the definition and the security requirement:

Definition 8 (Attribute-based Encryption).
An Attribute-based encryption (ABE) scheme ABE *consists of four PPT algorithms* $\mathsf{ABKEM} = (\mathsf{Gen}, \mathsf{USKGen}, \mathsf{Enc}, \mathsf{Dec})$ *with the following properties.*

- *The probabilistic key generation algorithm* $\mathsf{Gen}(\mathfrak{K})$ *returns the (master) public/secret key* $(\mathsf{pk}, \mathsf{sk})$*. We assume that* pk *implicitly defines a message space* $\mathcal{M}$*, an Attribute space* AS*, and ciphertext space* CS*.*
- *The probabilistic user secret key generation algorithm* $\mathsf{USKGen}(\mathsf{sk}, \mathbb{A})$ *that takes as input the master secret key* sk *and a set of attributes* $\mathbb{A} \subset \mathsf{AS}$ *and returns the user secret-key* $\mathsf{usk}[\mathbb{A}]$*.*
- *The probabilistic encryption algorithm* $\mathsf{Enc}(\mathsf{pk}, \mathbb{F}, M)$ *returns a ciphertext* $\mathsf{C} \in \mathsf{CS}$ *with respect to the access structure* $\mathbb{F}$*.*
- *The deterministic decryption algorithm* $\mathsf{Dec}(\mathsf{usk}[\mathbb{A}], \mathbb{F}, \mathbb{A}, \mathsf{C})$ *returns the decrypted message* $M \in \mathcal{M}$ *or the reject symbol* $\perp$*.*

For perfect correctness we require that for all $\mathfrak{K} \in \mathbb{N}$*, all pairs* $(\mathsf{pk}, \mathsf{sk})$ *generated by* $\mathsf{Gen}(\mathfrak{K})$*, all access structure* $\mathbb{F}$*, all set of attribute* $\mathbb{A} \subset \mathsf{AS}$ *satisfying* $\mathbb{F}$*, all* $\mathsf{usk}[\mathbb{A}]$ *generated by* $\mathsf{USKGen}(\mathsf{sk}, \mathbb{A})$ *and all* C *output by* $\mathsf{Enc}(\mathsf{pk}, \mathbb{F}, M)$*:*

$$\Pr[\mathsf{Dec}(\mathsf{usk}[\mathbb{A}], \mathbb{F}, \mathbb{A}, \mathsf{C}) = M] = 1.$$

Like before, we encompass the classical security hypotheses for an ABE, with a $\mathsf{PR\text{-}A\text{-}CPA}$ one as described in Figure 5.

Procedure $\mathsf{Initialize}$:
$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(\mathfrak{K})$
Return pk

Procedure $\mathsf{USKGen}(\mathbb{A})$:
$\mathcal{Q}_A \leftarrow \mathcal{Q}_A \cup \{\mathbb{A}\}$
Return $\mathsf{usk}[\mathbb{A}] \xleftarrow{\$} \mathsf{USKGen}(\mathsf{sk}, \mathbb{A})$

Procedure $\mathsf{Enc}(\mathbb{F}^*)$: //one query
$(\mathsf{sk}^*, \mathsf{C}^*) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, \mathbb{F}^*, M^*)$
$\boxed{\mathsf{C}^* \xleftarrow{\$} \mathsf{CS}}$
Return (C^*)

Procedure $\mathsf{Finalize}(\beta)$:
Return $(\forall \mathbb{A} \in \mathcal{Q}_A, \mathbb{A}$ doesn't verify $\mathbb{F}) \wedge \beta$

Fig. 5. Security Games $\mathsf{PR\text{-}A\text{-}CPA}_{\mathsf{real}}$ and $\boxed{\mathsf{PR\text{-}A\text{-}CPA}_{\mathsf{rand}}}$ for defining PR-A-CPA-security.

Definition 9 (PR-A-CPA **Security).** *An identity-based key encapsulation scheme* ABKEM *is* PR-A-CPA*-secure if for all PPT $\mathcal{A}$,* $\mathsf{Adv}^{\mathsf{PR\text{-}A\text{-}CPA}}_{\mathsf{ABKEM}}(\mathcal{A}) := |\Pr[\mathsf{PR\text{-}A\text{-}CPA}^{\mathcal{A}}_{\mathsf{real}} \Rightarrow 1] - \Pr[\mathsf{PR\text{-}A\text{-}CPA}^{\mathcal{A}}_{\mathsf{rand}} \Rightarrow 1]|$ *is negligible.*

In a usual notion of (ciphertext-policy) ABE, a key is associated with a set $\mathbb{A}$ of attributes in the attribute universe $\mathcal{U}$, while a ciphertext is associated with an access policy $\mathbb{F}$ (or called access structure) over attributes. The decryption can be done if $\mathbb{A}$ satisfies $\mathbb{F}$. We can see that IBE is a special case of ABE where both $\mathbb{A}$ and $\mathbb{F}$ are singletons, that is, each is an identity in the universe $\mathcal{U}$.

In this paper, we confine ABE in the two following aspects. First, we restrict the universe $\mathcal{U}$ to be of polynomial size in security parameter; this is often called small-universe ABE (as opposed to large-universe ABE where $\mathcal{U}$ can be of super polynomial size.). Second, we allow only DNF formulae in expressing policies (as opposed to any boolean formulae, or equivalently, any access structures).

Our idea for obtaining a (small-universe) ABE scheme for DNF formulae from any DIBE scheme is as follows. For simplicity and wlog, we set the universe as $\mathcal{U} = \{1, \ldots, n\}$. We will use DIBE with identity length n. For any set $S \subseteq \mathcal{U}$, we define $\mathsf{id}_S \in \{0,1\}^n$ where its i-th position is defined by

$$\mathsf{id}_S[i] := \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}.$$

To issue an ABE key for a set $\mathbb{A} \subseteq \mathcal{U}$, we use a DIBE key for $\mathsf{id}_{\mathbb{A}}$. On the other hand, to encrypt a message M in ABE with a DNF policy $\mathbb{F} = \bigvee_{j=1}^{k}(\bigwedge_{a \in S_j} a)$, where each attribute a is in $\mathcal{U}$, we encrypt the same message M in DIBE each with id_{S_j} for all $j \in [1, k]$; this will result in k ciphertexts of the DIBE scheme. Note that k is the number of OR, the disjunction, in the DNF formula.

Decryption can be done as follows. Suppose $\mathbb{A}$ satisfies $\mathbb{F}$. Hence, we have that there exists S_j (defined in the formula $\mathbb{F}$) such that $S_j \subseteq \mathbb{A}$. We then derive a DIBE key for id_{S_j} from our ABE key for $\mathbb{A}$ (which is then a DIBE key for $\mathsf{id}_{\mathbb{A}}$); this can be done since $S_j \subseteq \mathbb{A}$ implies that any positions of 1 in id_{S_j} will also contain 1 in $\mathsf{id}_{\mathbb{A}}$ (and thus the derivation is possible). We finally decrypt the ciphertext associated with id_{S_j} to obtain the message M. We summarize this transformation in Fig 6.

<u>$\mathsf{Setup}(\mathsf{param})$:</u> Run $\mathsf{Gen}_{\mathsf{DIBE}}(\mathfrak{K})$ Return $(\mathsf{mpk}, \mathsf{msk})$ <u>$\mathsf{KeyGen}(\mathsf{msk}, \mathbb{A})$:</u> Return $\mathsf{usk}[\mathbb{A}] \leftarrow \mathsf{USKGen}_{\mathsf{DIBE}}(\mathsf{msk}, \mathsf{id}_{\mathbb{A}})$	<u>$\mathsf{Encrypt}(\mathsf{mpk}, \mathbb{F}, M)$:</u> Parse $\mathbb{F} = \bigvee_{j=1}^{k}(\bigwedge_{a \in S_j} a)$ For all $j \in [1, k]$, compute: $(\mathsf{C}_j, K_j) \leftarrow \mathsf{Enc}_{\mathsf{DIBE}}(\mathsf{mpk}, \mathsf{id}_{S_j})$ and $C'_j \leftarrow M \oplus K_j$ Return $\mathsf{C} = (\mathsf{C}_1, \ldots, \mathsf{C}_k, C'_1, \ldots C'_k)$ <u>$\mathsf{Decrypt}(\mathsf{usk}[\mathbb{A}], \mathbb{F}, \mathbb{A}, \mathsf{C})$:</u> Parse $\mathbb{F} = \bigvee_{j=1}^{k}(\bigwedge_{a \in S_j} a)$ Find $j \in [1, k]$ s.t. $S_j \subseteq \mathbb{A}$ Compute $U \leftarrow \mathsf{USKDown}_{\mathsf{DIBE}}(\mathsf{usk}[\mathbb{A}], \mathsf{id}_{S_j})$ Compute $K_j \leftarrow \mathsf{Dec}_{\mathsf{DIBE}}(U, \mathsf{id}_{S_j}, \mathsf{C}_j)$ Return $M = C'_j \oplus K_j$

Fig. 6. ABE from DIBE

We have the following security theorem for the above ABE scheme. The proof is very simple and is done by a straightforward hybrid argument over k ciphertexts of DIBE. Note that the advantage definition for ABE is defined similarly to other primitives and is captured in Appendix **??**.

Theorem 10. *The above* ABE *from* DIBE *is* pr-a-cpa *secure under the* pr-id-cpa *security of the* DIBE *scheme used. In particular for all adversaries $\mathcal{A}$, we have that $\mathsf{Adv}^{\mathsf{PR\text{-}A\text{-}CPA}}_{\mathsf{ABE}}(\mathcal{A}) \leq k \cdot \mathsf{Adv}^{\mathsf{pr\text{-}id\text{-}cpa}}_{\mathsf{DIBE}}(\mathcal{A})$ where k is the number of OR in the DNF formula (associated to the challenge ciphertext).*

Proof. We prove our transformation via a sequence of games beginning with the real game for the pr-a-cpa security of the ABE and ending up with a game where the ciphertext of the ABE is uniformly chosen at random e.g. a game where adversary's advantage is reduce to 0.

Let $\mathcal{A}$ be an adversary against the $\mathsf{pr\text{-}a\text{-}cpa}$ security of our transformation. Let C be the simulator of the $\mathsf{pr\text{-}a\text{-}cpa}$ experience.

Game G_0: This is the real security game.

Game $\mathsf{G}_{1.1}$: In this game the simulator generates correctly every ciphertexts but the first one. The first ciphertext is replaced by a random element of the ciphertext space. $\mathsf{G}_{1.1}$ is indistinguishable from Game 0 if the $\mathsf{pr\text{-}id\text{-}cpa}$ security holds for the DIBE used.

$$\mathsf{Adv}^{\mathsf{G}_0,\mathsf{G}_{1.1}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{pr\text{-}id\text{-}cpa}}_{\mathsf{DIBE}}(\mathcal{A})$$

Game $\mathsf{G}_{1.i}$: This game is the same than the game $\mathsf{G}_{1.i-1}$ but the i-th ciphertext is replaced by a random element of the ciphertext space. $\mathsf{G}_{1.i}$ is indistinguishable from $\mathsf{G}_{1.i-1}$ if the $\mathsf{pr\text{-}id\text{-}cpa}$ security holds for the DIBE used.

$$\mathsf{Adv}^{\mathsf{G}_{1.i-1},\mathsf{G}_{1.i}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{pr\text{-}id\text{-}cpa}}_{\mathsf{DIBE}}(\mathcal{A})$$

Game $\mathsf{G}_{1.k}$: in this game all ciphertexts are random elements, $\mathsf{G}_{1.k}$ is indistinguishable from $\mathsf{G}_{1.k-1}$ if the $\mathsf{pr\text{-}id\text{-}cpa}$ security holds for the DIBE used.

$$\mathsf{Adv}^{\mathsf{G}_{1.k-1},\mathsf{G}_{1.k}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{pr\text{-}id\text{-}cpa}}_{\mathsf{DIBE}}(\mathcal{A})$$

At this point our current game $\mathsf{G}_{1.k}$ has for challenge encryption only random elements. This means that an adversary has no advantage in winning this game. We finally end up with the advantage of $\mathcal{A}$ in winning the original security game:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{PR\text{-}A\text{-}CPA}}_{\mathsf{ABE}}(\mathcal{A}) &\leq \mathsf{Adv}^{\mathsf{G}_0,\mathsf{G}_{1.k}}(\mathcal{A}) \\ &\leq \sum_{i=1}^{k} \mathsf{Adv}^{\mathsf{G}_{1.i-1},\mathsf{G}_{1.i}}(\mathcal{A}) \\ &\leq k \times \mathsf{Adv}^{\mathsf{pr\text{-}id\text{-}cpa}}_{\mathsf{DIBE}}(\mathcal{A})\end{aligned}$$

□

6 Instantiation

Theorem 11. *Under the* $\mathcal{D}_k$-MDDH *assumption, the scheme presented in figure 7 is* $\mathsf{PR\text{-}ID\text{-}CPA}$ *secure. For all adversaries* $\mathcal{A}$ *there exists an adversary* $\mathcal{B}$ *with* $\mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B})$ *and* $\mathbf{Adv}_{\mathsf{DIBKEM},\mathcal{D}_k}(\mathcal{B})^{\mathsf{PR\text{-}ID\text{-}CPA}}(\mathcal{A}) \leq (\mathbf{Adv}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{B}) + 2q_k(\mathbf{Adv}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{B}) + 1/q)$ [1].

[1] We recall that q_k is the maximal number of query to the Eval oracle

$\underline{\mathsf{Gen}(\mathsf{param}):}$

$\mathbf{A} \xleftarrow{\$} \mathcal{D}_k, \mathbf{B} = \bar{\mathbf{A}}$

For $i = 0, \ldots, \ell$:

$\quad \boldsymbol{z}_i \xleftarrow{\$} \mathbb{Z}_p^{k+1\times n}; \mathbf{Z}_i = \mathbf{z}_i^\top \cdot \mathbf{A} \in \mathbb{Z}_p^{n\times k}$

$\boldsymbol{z}' \xleftarrow{\$} \mathbb{Z}_p^{k+1}; \boldsymbol{Z}' = \boldsymbol{z}'^\top \cdot \mathbf{A} \in \mathbb{Z}_p^{1\times k}$

$\mathsf{mpk} := (\mathcal{G}, [\mathbf{A}]_1, ([\mathbf{Z}_i]_1)_{0\le i\le \ell}, [\boldsymbol{Z}']_1)$

$\mathsf{msk} := ((\boldsymbol{z}_i)_{0\le i\le\ell}, \boldsymbol{z}')$

Return $(\mathsf{mpk}, \mathsf{msk})$

$\underline{\mathsf{USKGen}(\mathsf{msk}, \mathsf{id} \in \mathsf{ID}):}$

$\boldsymbol{t} \xleftarrow{\$} \mathbb{Z}_p^n$;

$\boldsymbol{v} = \sum_{i=0}^{l(\mathsf{id})} \mathsf{id}_i \mathbf{z}_i \boldsymbol{t} + \boldsymbol{z}' \in \mathbb{Z}_p^{k+1}$

$\mathbf{S} \xleftarrow{\$} \mathbb{Z}_p^{n'\times\mu}$; $\mathbf{T} = \mathbf{B} \cdot \mathbf{S} \in \mathbb{Z}_p^{n\times\mu}$

$\mathbf{V} = \sum_{i=0}^{l(\mathsf{id})} \mathsf{id}_i \mathbf{Z}_i \mathbf{T} \in \mathbb{Z}_p^{(k+1)\times\mu}$

For $i, \mathsf{id}[i] = 1$:

$\quad \boldsymbol{e}_i = \mathbf{Z}_i \boldsymbol{t} \in \mathbb{Z}_p^{k+1}$; $\boldsymbol{E}_i = \mathbf{Z}_i \mathbf{T} \in \mathbb{Z}_p^{k+1\times\mu}$

$\mathsf{usk}[\mathsf{id}] := ([\boldsymbol{t}]_2, [\boldsymbol{v}]_2) \in \mathbb{G}_2^n \times \mathbb{G}_2^{k+1}$

$\mathsf{udk}[\mathsf{id}] := ([\mathbf{T}]_2, [\mathbf{V}]_2, ([\boldsymbol{e}_i]_2, [\mathbf{E}_i]_2)_{i,\mathsf{id}[i]=1})$

$\quad \in \mathbb{G}_2^{n\times\mu} \times \mathbb{G}_2^{(k+1)\times\mu} \times (\mathbb{G}_2^{k+1} \times \mathbb{G}_2^{(k+1)\times\mu})^{\mathsf{Ham}(\mathsf{id})}$

Return $(\mathsf{usk}[\mathsf{id}], \mathsf{udk}[\mathsf{id}])$

$\underline{\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}):}$

$\boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_p^k$

$\boldsymbol{c}_0 = \mathbf{A}\boldsymbol{r} \in \mathbb{Z}_p^{k+1}$

$\boldsymbol{c}_1 = (\sum_{i=0}^{l(\mathsf{id})} \mathsf{id}_i \mathbf{Z}_i) \cdot \boldsymbol{r} \in \mathbb{Z}_p^n$

$K = \boldsymbol{z}'_0 \cdot \boldsymbol{r} \in \mathbb{Z}_p$.

Return $\mathsf{sk} = [K]_T$ and $\mathsf{C} = ([\boldsymbol{c}_0]_1, [\boldsymbol{c}_1]_1)$

$\underline{\mathsf{USKDown}(\mathsf{usk}[\mathsf{id}], \tilde{\mathsf{id}}):}$

If $\neg(\tilde{\mathsf{id}} \preceq \mathsf{id})$, then return $\perp$

Set $\mathcal{I} = \{i | \tilde{\mathsf{id}}[i] = 0 \wedge \mathsf{id}[i] = 1\}$

// Downgrading the key:

$\quad \hat{\boldsymbol{v}} = \boldsymbol{v} + \sum_{i\in\mathcal{I}} \tilde{\mathsf{id}}_i \boldsymbol{e}_i \in \mathbb{Z}_p^k + 1$

$\quad \hat{\boldsymbol{V}} = \boldsymbol{V} + \sum_{i\in\mathcal{I}} \tilde{\mathsf{id}}_i \boldsymbol{E}_i \in \mathbb{Z}_p^{k\times\mu}$

// Rerandomization of $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{V}})$:

$\quad \boldsymbol{s}' \xleftarrow{\$} \mathbb{Z}_p^\mu$; $\boldsymbol{S}' \xleftarrow{\$} \mathbb{Z}_p^{\mu\times\mu}$

$\quad \boldsymbol{t}' = \boldsymbol{t} + \mathbf{T}\boldsymbol{s}' \in \mathbb{Z}_p^n$;

$\quad \boldsymbol{T}' = \hat{\boldsymbol{T}} \cdot \boldsymbol{S}' \in \mathbb{Z}_p^{n\times\mu}$

$\quad \hat{\boldsymbol{v}}' = \hat{\boldsymbol{v}} + \hat{\boldsymbol{V}} \cdot \boldsymbol{s}' \in \mathbb{Z}_p^k$;

$\quad \boldsymbol{V}' = \hat{\boldsymbol{V}} \cdot \boldsymbol{S}' \in \mathbb{Z}_p^{(k+1)\times\mu}$

// Rerandomization of $\boldsymbol{e}_i$:

$\quad$ For $i, \tilde{\mathsf{id}}[i] = 1$:

$\quad\quad \boldsymbol{e}'_i = \boldsymbol{e}_i + \mathbf{E}_i \boldsymbol{s}' \in \mathbb{Z}_p^{k+1}$;

$\quad\quad \boldsymbol{E}'_i = \boldsymbol{E}_i \cdot \boldsymbol{S}' \in \mathbb{Z}_p^{(k+1)\times\mu}$

$\mathsf{usk}[\tilde{\mathsf{id}}] := ([\boldsymbol{t}']_2, [\hat{\boldsymbol{v}}']_2)$

$\mathsf{udk}[\tilde{\mathsf{id}}] := ([\boldsymbol{T}']_2, [\boldsymbol{V}']_2, [\boldsymbol{e}_i']_2, [\boldsymbol{E}'_i]_2)$

Return $(\mathsf{usk}[\tilde{\mathsf{id}}], \mathsf{udk}[\tilde{\mathsf{id}}])$

$\underline{\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{C}):}$

Parse $\mathsf{usk}[\mathsf{id}] = ([\boldsymbol{t}]_2, [\boldsymbol{v}]_2)$

Parse $\mathsf{C} = ([\boldsymbol{c}_0]_1, [\boldsymbol{c}_1]_1)$

$\mathsf{sk} = e([\boldsymbol{c}_0]_1, [\boldsymbol{v}]_2) \cdot e([\boldsymbol{c}_1]_1, [\boldsymbol{t}]_2)^{-1}$

Return $\mathsf{sk} \in \mathbb{G}_T$

Fig. 7. A Downgradable IBE based on MDDH. For readability, the user secret key is split here between usk for the decapsulation, and udk used for the downgrade operation.

The proof is detailed in Appendix B.

Remark 12. This instantiation respect the formal definition of DIBKEM of 3. However for efficiency purpose one can remark that for realizing WIBE or ABE the user's secret keys does not need to be rerandomize during the delegation phase since it will not be used by another user. It introduce the concept of self-delegatable-only scheme. Thus we can avoid the heavy elements $\boldsymbol{T}, \boldsymbol{S}, \boldsymbol{E}$ of the user secret keys, the self-delegetable-only scheme is describe in figure 7 when removing the gray parts.

7 Efficiency Comparison

In this section we compare the schemes obtained by using our instantiation of DIBE (see sec. 6) and our transformations described in the section 4. We end up with the most efficient scheme for full security in the standard model and under classical hypothesis for WIBE, WKD-IBE and of similar efficiency for HIBE.

In the example of WIBE and WKD-IBE given below the parameters will grow exponentially in the number of query from the adversary, where our will be only linear. This is a parameter to take into account because the size of the keys for the same security will depend on this security loss.

To compare efficiency in a simple way, we choose to consider the case where the number of pattern is maximal e.g. the size of pattern is equal to 1, thus the number of pattern is n which is the length of the identity. The value q_k correspond to the number of derivation key oracle request made by the adversary. [2]

Name	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{usk}\rvert$	$\lvert\mathsf{C}\rvert$	assump.	Sec	Loss
WKD [AKN07]	$n+4$	$n+2$	2	BDDH	Sel. standard	$O(nq_k)$
WKD [AKN07]	$(n+1)n+3$	$n+2$	2	BDDH	Full standard	$O(q_k^n)$
WKD-DIBE	$4n+2$	$3n+5$	5	DLin (any $k-$MDDH)	Full standard	$O(q_k)$
SWIBE [KLLO18]	$n+4$	$2n+3$	4	ROM	Full	$O((n+1)(q_k+1)^n)$
WIBE [BDNS07]	$(n+1)n+3$	$n+1$	$(n+1)n+2$	BDDH	Full standard	$O(n^2q_k^n)$
Wild-DIBE	$4n+2$	$3n+5$	5	DLin (any $k-$MDDH)	Full standard	$O(q_k)$

Fig. 8. Efficiency Comparison Between our Transformations and Previous Schemes

Efficiency comparison for HIBE The figure 9 compares the HIBE built via our DIBE. Our instantiation of DIBE inherit its efficiency from the HIBE from [BKP14], except we need to artificially double the size of the identities. Here ℓ is the number of free bits in an identity (the ones to delegate). Note that for the case of root of the hierarchy e.g. the user with an empty bit string as identity, $\ell = n$.

It should be noted, that while we rely on the same underlying principle, our security reduction does not need handle $\perp$ symbol as [BKP14], which allows to circumvent the worrisome parts of their proofs.

[2] In the original version of [AKN07] they include an element in the ciphertext to turn their scheme into an encryption scheme. Since our scheme is a Key Encapsulation Mechanism we remove this element when comparing both schemes.

Name	|pk|	|usk|	|C|	assump.	Loss
HIBE [BBG05]	$n+4$	$2+\ell$	5	DLin	sel. $O(n \cdot q_k)$
HIBE [BKP14]	$2n+1$	$11\ell+5$	5	DLin (any $k-$MDDH)	$O(n)$
H-DIBE	$4n+2$	$11n+5$	5	DLin (any $k-$MDDH)	$O(q_k)$

Fig. 9. Efficiency Comparison Between our Transformations and HIBE schemes

Efficiency comparison for ABE Our instantiation leads to a very efficient ABE scheme. This scheme would be one of the most practical. However we achieve ABE where the access structure has to be a boolean formula in the DNF which is less general than allowing any kind of access structure (which is done in others practical schemes).

Name	|pk|	|sk|	|C|	pairing	exp $\mathbb{G}$	exp $\mathbb{G}_t$	Reduction Loss
[OT10]	$4U+2$	$3U+3$	$7m+5$	$7m+5$	0	m	$O(q_k)$
[LW12]	$24U+12$	$6U+6$	$6m+6$	$6m+9$	0	m	$O(q_k)$
[CGW15]	$6UR+12$	$3UR+3$	$3m+3$	6	$6m$	0	$O(q_k)$
[Att16] scheme 10	$6UR+12$	$3UR+6$	$3m+6$	9	$6m$	0	$O(q_k)$
[Att16] scheme 13	$96(M+TR)^2+log(UR)$	$3UR+6$	$3m+6$	9	$6m$	0	$O(q_k)$
Our DNF-ABE	$4U+2$	$3U+3$	$3k+2$	13	0	0	$O(q_k)$

Fig. 10. Efficiency Comparison of Practical CP-ABE Schemes

Fig. 10 presents a non exhaustive comparison of our ABE schemes with efficient ones. They are all full secure under the classical assumption DLin. U is the size of the universe of attributes. m is the number of attributes in a policy. t is the size of an attribute set, and T is the maximum size of t (if bounded). R is the maximum number of attributes multi used in one policy (if bounded). q_k is again the number of all the key queries made by the adversary during security game. For our scheme, k is the number of OR, the disjunction, in the associated DNF formula.

Acknowledgements.

This work was supported in part by the French ANR: IDFIX (ANR-16-CE39-0004) Project

References

[ACD+06] Michel Abdalla, Dario Catalano, Alex Dent, John Malone-Lee, Gregory Neven, and Nigel Smart. Identity-based encryption gone wild. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006, Part II*, volume 4052 of *LNCS*, pages 300–311. Springer, Heidelberg, July 2006.

[ACP12] Michel Abdalla, Angelo De Caro, and Duong Hieu Phan. Generalized key delegation for wildcarded identity-based and inner-product encryption. *IEEE Transactions on Information Forensics and security*, 7(6):1695–1706, 2012.

[ADML+07] Michel Abdalla, Alexander W. Dent, John Malone-Lee, Gregory Neven, Duong Hieu Phan, and Nigel P. Smart. Identity-based traitor tracing. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 361–376. Springer, Heidelberg, April 2007.

[AFL12] Michel Abdalla, Dario Fiore, and Vadim Lyubashevsky. From selective to full security: Semi-generic transformations in the standard model. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 316–333. Springer, Heidelberg, May 2012.

[AKN07] Michel Abdalla, Eike Kiltz, and Gregory Neven. Generalized key delegation for hierarchical identity-based encryption. In Joachim Biskup and Javier López, editors, *ESORICS 2007*, volume 4734 of *LNCS*, pages 139–154. Springer, Heidelberg, September 2007.

[Att16] Nuttapong Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 591–623. Springer, Heidelberg, December 2016.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, Heidelberg, May 2005.

[BCG16] Olivier Blazy, Céline Chevalier, and Paul Germouty. Adaptive oblivious transfer and generalization. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 217–247. Springer, Heidelberg, December 2016.

[BDNS07] James Birkett, Alexander W. Dent, Gregory Neven, and Jacob C. N. Schuldt. Efficient chosen-ciphertext secure identity-based encryption with wildcards. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *ACISP 07*, volume 4586 of *LNCS*, pages 274–292. Springer, Heidelberg, July 2007.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001.

[BKP14] Olivier Blazy, Eike Kiltz, and Jiaxin Pan. (Hierarchical) identity-based encryption from affine message authentication. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 408–425. Springer, Heidelberg, August 2014.

[CGW15] Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Heidelberg, April 2015.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *8th IMA International Conference on Cryptography and Coding*, volume 2260 of *LNCS*, pages 360–363. Springer, Heidelberg, December 2001.

[EHK+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013.

[FP12] Nelly Fazio and Irippuge Milinda Perera. Outsider-anonymous broadcast encryption with sublinear ciphertexts. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 225–242. Springer, Heidelberg, May 2012.

[GH07] Matthew Green and Susan Hohenberger. Blind identity-based encryption and simulatable oblivious transfer. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 265–282. Springer, Heidelberg, December 2007.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 548–566. Springer, Heidelberg, December 2002.

[KLLO18] Jihye Kim, Seunghwa Lee, Jiwon Lee, and Hyunok Oh. Scalable wildcarded identity-based encryption. In Javier López, Jianying Zhou, and Miguel Soriano, editors, *ESORICS 2018, Part II*, volume 11099 of *LNCS*, pages 269–287. Springer, Heidelberg, September 2018.

[LW12] Allison B. Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 180–198. Springer, Heidelberg, August 2012.

[OT10] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer, Heidelberg, August 2010.

[PT11] Duong Hieu Phan and Viet Cuong Trinh. Identity-based trace and revoke schemes. In Xavier Boyen and Xiaofeng Chen, editors, *ProvSec 2011*, volume 6980 of *LNCS*, pages 204–221. Springer, Heidelberg, October 2011.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Heidelberg, August 1984.

[SOK00] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing. In *SCIS 2000*, Okinawa, Japan, January 2000.

[SW05] Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.

[Wat05] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, Heidelberg, May 2005.

A Extra Definitions

A.1 Wildcard Identity-based Key Encapsulation Scheme

Definition 13 (Wildcard Identity-based Key Encapsulation Scheme). *A Wildcard identity-based key encapsulation scheme* WIBKEM *consists of five PPT algorithms* $\mathsf{WIBKEM} = (\mathsf{Gen}, \mathsf{USKGen}, \mathsf{Enc}, \mathsf{Dec})$ *with the following properties.*

- *The probabilistic key generation algorithm* $\mathsf{Gen}(\mathfrak{K})$ *returns the (master) public/secret key* $(\mathsf{pk}, \mathsf{sk})$*. We assume that* pk *implicitly defines a message space* $\mathcal{M}$*, an identity space* ID*, a key space* $\mathcal{K}$*, and ciphertext space* CS*.*
- *The probabilistic user secret key generation algorithm* $\mathsf{USKGen}(\mathsf{sk}, \mathsf{id})$ *returns the user secret-key* $\mathsf{usk}[\mathsf{id}]$ *for identity* $\mathsf{id} \in \mathsf{ID}$*.*
- *The probabilistic encapsulation algorithm* $\mathsf{Enc}(\mathsf{pk}, \mathsf{id})$ *returns the symmetric key* $\mathsf{sk} \in \mathcal{K}$ *together with a ciphertext* $\mathsf{C} \in \mathsf{CS}$ *with respect to an identity* $\mathsf{id} \in \hat{\mathsf{ID}}$*, this means that* $\forall i, \mathsf{id}_i \in \{0, 1, *\}$*.*
- *The deterministic decapsulation algorithm* $\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \hat{\mathsf{id}}, \mathsf{C})$ *returns the decapsulated key* $\mathsf{sk} \in \mathcal{K}$ *or the reject symbol* $\perp$*.*

For perfect correctness we require that for all $\mathfrak{K} \in \mathbb{N}$*, all pairs* $(\mathsf{pk}, \mathsf{sk})$ *generated by* $\mathsf{Gen}(\mathfrak{K})$*, all identities* $\mathsf{id} \in \mathsf{ID}$*, all* $\mathsf{usk}[\mathsf{id}]$ *generated by* $\mathsf{USKGen}(\mathsf{sk}, \mathsf{id})$ *and all* $(\mathsf{sk}, \mathsf{C})$ *output by* $\mathsf{Enc}(\mathsf{pk}, \hat{\mathsf{id}})$ *for* $\hat{\mathsf{id}} \in \hat{\mathsf{ID}}$ *such that* $\hat{\mathsf{id}} \preceq_* \mathsf{id}$*:*

$$\Pr[\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{C}) = \mathsf{sk}] = 1.$$

A.2 Hierarchical Identity-Based Key Encapsulation Mechanism

We recall syntax and security of a hierarchical identity-based key encapsulation mechanism (HIBKEM).

Definition 14 (Hierarchical Identity-Based Key Encapsulation Mechanism). *A hierarchical identity-based key encapsulation mechanism* DIBKEM *consists of five PPT algorithms* $\mathsf{DIBKEM} = (\mathsf{Gen}, \mathsf{USKDel}, \mathsf{USKGen}, \mathsf{Enc}, \mathsf{Dec})$ *with the following properties.*

- *The probabilistic key generation algorithm* $\mathsf{Gen}(\mathfrak{K})$ *returns the (master) public/secret key and delegation key* $(\mathsf{pk}, \mathsf{sk})$*. We assume that* pk *implicitly defines a message space* $\mathcal{M}$ *and hierarchical identity space* $\mathsf{ID} = \mathcal{BS}^{\leq m}$*, for some base identity set* $\mathcal{BS}$*.*
- *The probabilistic user secret key generation algorithm* $\mathsf{USKGen}(\mathsf{sk}, \mathsf{id})$ *returns a secret key* $\mathsf{usk}[\mathsf{id}]$ *for hierarchical identity* $\mathsf{id} \in \mathsf{ID}$*.*

– *The probabilistic key delegation algorithm* $\mathsf{USKDel}(\mathsf{usk}[\mathsf{id}], \mathsf{id} \in \mathcal{BS}^p, \mathsf{id}_{p+1} \in \mathcal{BS})$ *returns a user secret key* $\mathsf{usk}[\mathsf{id}|\mathsf{id}_{p+1}]$ *for the hierarchical identity* $\mathsf{id}' = \mathsf{id} \mid \mathsf{id}_{p+1} \in \mathcal{BS}^{p+1}$. *We require* $1 \leq |\mathsf{id}| \leq m - 1$.
– *The probabilistic encapsulation algorithm* $\mathsf{Enc}(\mathsf{pk}, \mathsf{id})$ *returns a symmetric key* $\mathsf{sk} \in \mathcal{K}$ *together with a ciphertext* C *with respect to the hierarchical identity* $\mathsf{id} \in \mathsf{ID}$.
– *The deterministic decapsulation algorithm* $\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{C})$ *returns a decapsulated key* $\mathsf{sk} \in \mathcal{K}$ *or* $\perp$.

For correctness we require that for all $\mathfrak{K} \in \mathbb{N}$, *all pairs* $(\mathsf{pk}, \mathsf{sk})$ *generated by* $\mathsf{Gen}(\mathfrak{K})$, *all* $\mathsf{id} \in \mathsf{ID}$, *all* $\mathsf{usk}[\mathsf{id}]$ *generated by* $\mathsf{USKGen}(\mathsf{sk}, \mathsf{id})$ *and all* (sk, c) *generated by* $\mathsf{Enc}(\mathsf{pk}, \mathsf{id})$:

$$\Pr[\mathsf{Dec}(\mathsf{usk}[\mathsf{id}], \mathsf{id}, \mathsf{C}) = \mathsf{sk}] = 1.$$

Moreover, we also require the distribution of $\mathsf{usk}[\mathsf{id}|\mathsf{id}_{p+1}]$ *generated with* $\mathsf{USKDel}(\mathsf{usk}[\mathsf{id}], \mathsf{udk}[\mathsf{id}], \mathsf{id}, \mathsf{id}_{p+1})$ *to be identical to the one from* $\mathsf{USKGen}(\mathsf{sk}, \mathsf{id}|\mathsf{id}_{p+1})$.

B Downgradable IBE Proof

Theorem 15. *Under the* $\mathcal{D}_k$-MDDH *assumption, the scheme presented in figure 7 is* $\mathsf{PR\text{-}ID\text{-}CPA}$ *secure. For all adversaries* $\mathcal{A}$ *there exists an adversary* $\mathcal{B}$ *with* $\mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B})$ *and* $\mathbf{Adv}_{\mathsf{DIBKEM},\mathcal{D}_k}(\mathcal{B})^{\mathsf{PR\text{-}ID\text{-}CPA}}(\mathcal{A}) \leq (\mathbf{Adv}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{B}) + 2q_k(\mathbf{Adv}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{B}) + 1/q)$ [3].

The inner block is a downgradable MAC

Definition 16. *An affine* MAC *over* $\mathbb{Z}_p^n$ *is* downgradable, *if the message space is* $\mathcal{M} = \{0,1\}^m$ *for some finite base set* $\{0,1\}$, $f_0'(\mathsf{m}) = 1$, *and there exists a public function* $f : \mathcal{M} \to \{0, \ldots, \ell\}$ *such that for all* $\mathsf{m}' \preceq \mathsf{m}$,

$$f_i(\mathsf{m}_i') = \begin{cases} f_i(\mathsf{m}_i) & \text{if } \mathsf{m}_i = \mathsf{m}_i' \\ f_i(0) & \text{otherwise} \end{cases}.$$

Let MAC be a delegatable affine MAC over $\mathbb{Z}_p^n$ with message space $\mathcal{M} = \{0,1\}^m$. To build a DIBE, we require a new notion denoted as $\mathsf{DPR_0\text{-}CMA}$ security. It differs from the classical security in two ways. Firstly, additional values needed for DIBE downgrade process are provided to the adversary through the call to $\mathsf{Initialize}$ and Eval. Secondly, Chal always returns a real $\boldsymbol{h}_0$. (In fact, the additional values actually allow the adversary to distinguish real from random $\boldsymbol{h}_0$.)

Let $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, g_1, g_2, e)$ be an asymmetric pairing group in par. Consider the games from Figure 11.

[3] We recall that q_k is the maximal number of query to the Eval oracle

<u>Initialize:</u> $\mathsf{sk_{MAC}} = (\mathbf{B}, (x_i)_{0\leq i\leq \ell}, x'_0) \xleftarrow{\$} \mathsf{Gen_{MAC}}(\mathsf{par})$ Return $([\mathbf{B}]_2, ([x_i^\top \mathbf{B}]_2)_{0\leq i\leq \ell})$ <u>Eval(m):</u> $\mathcal{Q}_\mathcal{M} = \mathcal{Q}_\mathcal{M} \cup \{\mathsf{m}\}$ $([t]_2, [u]_2) \xleftarrow{\$} \mathsf{Tag}(\mathsf{sk_{MAC}}, \mathsf{m})$ For $i, \mathsf{m}_i = 1$: $d_i = x_i^\top t \in \mathbb{Z}_p$ Return $([t]_2, [u]_2, ([d_i]_2))$	<u>Chal(m*):</u> // one query $h \xleftarrow{\$} \mathbb{Z}_p$ $h_0 = \sum f_i(\mathsf{m}_i^*) x_i \cdot h \in \mathbb{Z}_p^n$ $h_1 = x'_0 \cdot h \in \mathbb{Z}_p$ $\boxed{h_1 \xleftarrow{\$} \mathbb{Z}_p}$ Return $([h]_1, [h_0]_1, [h_1]_T)$ <u>Finalize($\beta \in \{0,1\}$):</u> Return $\beta \wedge (\mathsf{m}^* \not\preceq \mathcal{Q}_\mathcal{M})$

Fig. 11. Games $\mathsf{DPR\text{-}CMA_{real}}$, and $\mathsf{DPR_0\text{-}CMA_{rand}}$ (boxed) for defining $\mathsf{DPR_0\text{-}CMA}$ security.

Definition 17. *A delegatable affine* MAC *over* $\mathbb{Z}_p^n$ *is* $\mathsf{DPR_0\text{-}CMA}$*-secure if for all PPT* $\mathcal{A}$*,* $\mathsf{Adv}_{\mathsf{MAC}}^{\mathsf{dpr_0\text{-}cma}}(\mathcal{A}) := \Pr[\mathsf{DPR\text{-}CMA}_{\mathsf{real}}^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{DPR_0\text{-}CMA}_{\mathsf{rand}}^{\mathcal{A}} \Rightarrow 1]$ *is negligible.*

We explicit in Figure 12 the inner downgradable MAC we consider in our scheme. And then prove its security.

<u>$\mathsf{Gen_{MAC}}(\mathsf{par})$:</u> $\mathbf{B} \xleftarrow{\$} \mathcal{D}_k$ $x_0, \dots, x_l \xleftarrow{\$} \mathbb{Z}_p^{k+1}; x'_0 \xleftarrow{\$} \mathbb{Z}_p$ $\mathsf{sk_{MAC}} = (\mathbf{B}, x_0, \dots, x_l, x'_0)$ Return $\mathsf{sk_{MAC}}$ <u>$\mathsf{Tag}(\mathsf{sk_{MAC}}, \mathsf{m})$:</u> $s \xleftarrow{\$} \mathbb{Z}_p^k, t = \mathbf{B}s$ $u = (x_0^\top + \sum_{i=1}^{\lvert\mathsf{m}\rvert} \mathsf{m}_i \cdot x_i^\top)t + x'_0 \in \mathbb{Z}_p$ For $i, \mathsf{m}_i = 1$, $d'_i = (-x_i)t$ Return $\tau = ([t]_2, [u]_2, [d]_2) \in \mathbb{G}_2^{k+1} \times \mathbb{G}_2 \times \mathbb{G}_2^{\mathsf{Ham(m)}}$	<u>$\mathsf{Down}(\tau, \mathsf{m}, \mathsf{m}')$:</u> If $\mathsf{m}' \preceq \mathsf{m}$, $[u']_2 = [u + \sum_{i, \mathsf{m}'_i \neq \mathsf{m}_i} d_i]_2$ $\forall i, \mathsf{m}'_i = 1, [d_i]_2 = [d_i]_2$ Return $\tau' = ([t]_2, [u']_2, [d']_2) \in \mathbb{G}_2^{k+1} \times \mathbb{G}_2 \times \mathbb{G}_2^{\mathsf{Ham(m')}}$ <u>$\mathsf{Ver}(\mathsf{sk_{MAC}}, \tau, \mathsf{m})$:</u> If $u = (x_0^\top + \sum_{i=1}^{\lvert\mathsf{m}\rvert} \mathsf{m}_i \cdot x_i^\top)t + x'_0$ then return 1; Else return 0.

Fig. 12. Downgradable MAC from HPS [BKP14]

In this proof we will show that an adversary will be at some point against a standard affine MAC thus the security of the MAC we based our instantiation on, ensure the security of our Downgradable MAC. Intuitively, we will replace query by query the answer of the Eval oracle by pure randomness in $\mathbb{G}_2^{k+1} \times \mathbb{G}_2 \times \mathbb{G}_2^{\mathsf{Ham(m)}}$. This proof is close from the proof of security of the affine MAC from HPS in [BKP14]. G_0 is the real security game defined in 11. $\mathsf{G}_{1,i}$ the first $i-1$ answer to the Eval oracle are random and the rest is answered as in the real game. We also need a game to switch from $game g_{1,i}$ to the game $\mathsf{G}_{1,i+1}$. This new game will be called $\mathsf{G}'_{1,i}$. Here we will only describe how to come from $\mathsf{G}'_{i,1}$ to $\mathsf{G}_{i+1,1}$ since it is the only part that will differ from the proof in [BKP14].

Let m be the i-th query of the adversary, since $\mathsf{m}^* \not\leq \mathsf{m}$ there exists a j such that $\mathsf{m}^*_j \neq \mathsf{m}_j$ and $f_j(\mathsf{m}^*_j) \neq f_j(0)$. In this configuration the adversary not more information about x_j than in a standard affine MAC. We can thus reuse the argument of the original proof: there is an information-theoretic argument to show that $u - x'_0$ is uniformly random. To simplify our proof we assume that the adversary $\mathcal{A}$ knows x'_0 and all x_l with $l \notin \{0, j\}$. He may also know $\mathbf{B}^\top x_0$ and $\mathbf{B}^\top x_j$. We will show that $\mathcal{A}$ is unable to guess x_j and x_0, $\mathcal{A}$ has to solve the following matrix equation:

$$\begin{pmatrix} \mathbf{B}^\top x_0 \\ \mathbf{B}^\top \\ h_0 \\ u - x'_0 \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{B}^\top & 0 \\ 0 & \mathbf{B}^\top \\ h \cdot \mathbf{I}_{k+1} & \mathsf{m}^*_j h \cdot \mathbf{I}_{k+1} \\ t^\top & m_j t^\top \end{pmatrix}}_{\mathbf{M}} \cdot \begin{pmatrix} x_0 \\ x_j \end{pmatrix} \tag{1}$$

The $u - x'_0$ is linearly independent from the other rows: $t^\top$ is independent from $\mathbf{B}^\top$ because $t \notin span(\mathbf{B})$ with probability $(q-1)/q$, also $\mathsf{m}_j \neq \mathsf{m}^*_j$ which means that this last row is independent from the rows $\big(h \cdot \mathbf{I}_{k+1}\ \mathsf{m}^*_j h \cdot \mathbf{I}_{k+1}\big)$. Thus this system of equations has not enough equations to be solved e.g. $\mathcal{A}$ can not distinguish between a random and u (except for a probability 1/q).

Finally, we do all the other steps of the proof like in the original proof, and then we end up with the following lemma.

Lemma 18. *For all adversaries $\mathcal{A}$ there exists an adversary $\mathcal{B}$ with $\mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B})$ and $\mathbf{Adv}_{\mathsf{MAC}_{\mathsf{HPS}},\mathcal{D}_k}(\mathcal{B})^{\mathsf{DPR_0\text{-}CMA}}(\mathcal{A}) \leq 2q_k(\mathbf{Adv}_{\mathcal{D}_k,\mathsf{GGen}}(\mathcal{B}) + 1/q)$.*

Which leads to the security of the downgradable MAC.

Achieving Secure DIBE

We define the sequence of games G_0-G_4 as in Figure 13. Let $\mathcal{A}$ be an adversary against the $\mathsf{PR\text{-}ID\text{-}CPA}$ security of DIBKEM. G_0 is the real attack game.

We can see that G_1 is simply a rewriting of G_0.

Lemma 19. $\Pr[\mathsf{G}_1^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathsf{G}_0^{\mathcal{A}} \Rightarrow 1]$.

Initialize: // Games G_0-G_2, $\boxed{\mathsf{G}_3\text{-}\mathsf{G}_4}$

$\mathcal{G} \xleftarrow{\$} \mathsf{GGen}(\mathfrak{K})$; $\mathbf{A} \xleftarrow{\$} \mathcal{D}_k$

$\mathsf{sk}_{\mathsf{MAC}} = (\mathbf{B}, x_0, \dots, x_\ell, x_0') \xleftarrow{\$} \mathsf{Gen}_{\mathsf{MAC}}(\mathcal{G})$

$\forall i \in [\![0, \ell]\!]$:

$\mathbf{Y}_i \xleftarrow{\$} \mathbb{Z}_p^{k \times n}; \mathbf{Z}_i = (\mathbf{Y}_i^\top \mid x_i) \cdot \mathbf{A} \in \mathbb{Z}_p^{n \times k}$

$d_{i,1} = z_i^\top \cdot \mathbf{B} \in \mathbb{Z}_p^k$

$d_{i,2\text{-}n} = z_i^\top \cdot \mathbf{B} \in \mathbb{Z}_p^{k \times n-1}$

$\boxed{d_{i,2\text{-}n'} = (\overline{\mathbf{A}}^{-1})^\top (\mathbf{Z}_i^\top \mathbf{B} - \underline{\mathbf{A}}^\top x_i \mathbf{B})}$

$y_0' \xleftarrow{\$} \mathbb{Z}_p^k; z_0' = (y_0'^\top \mid x_0') \cdot \mathbf{A} \in \mathbb{Z}_p^{1 \times k}$

$\mathsf{pk} := ([\mathbf{A}]_1, ([\mathbf{Z}_i]_1)_{0 \le i \le \ell}, [z_0']_1)$

$\mathsf{dk} := ([\mathbf{B}]_2, ([\mathbf{d}_i]_2)_{0 \le i \le \ell})$

$\mathsf{sk} := ((\mathbf{Z}_i)_{0 \le i \le \ell}, z_0')$

Return $(\mathsf{pk}, \mathsf{dk})$

USKGen(id): //Games G_0-G_2, $\boxed{\mathsf{G}_3\text{-}\mathsf{G}_4}$

$\mathcal{Q}_{\mathsf{ID}} = \mathcal{Q}_{\mathsf{ID}} \cup \{\mathsf{id}\}$

$([t]_2, [u]_2) \xleftarrow{\$} \mathsf{Tag}(\mathsf{sk}_{\mathsf{MAC}}, \mathsf{id})$

$v = \sum_i f_i(\mathsf{id}) \mathbf{Y}_i t + y_0' \in \mathbb{Z}_p^k$

$\boxed{v^\top = (t^\top \sum f_i(\mathsf{id}) \mathbf{Z}_i + z_0' - u \cdot \underline{\mathbf{A}}) \cdot \overline{\mathbf{A}}^{-1}}$

For $i, \mathsf{id}[i] = 1$:

$d_{i,1} = \mathbf{x}_i^\top t \in \mathbb{Z}_p$

$d_{i,2\text{-}n} = \mathbf{Y}_i t \in \mathbb{Z}_p^k$;

$\boxed{d_{i,2\text{-}n}^\top = (t^\top \mathbf{Z}_i - d_{i,1} \underline{\mathbf{A}}) \overline{\mathbf{A}}^{-1} \in \mathbb{Z}_p^{1 \times k}}$

$\mathsf{usk}[\mathsf{id}] := ([t]_2, [u]_2, [v]_2) \in \mathbb{G}_2^n \times \mathbb{G}_2^1 \times \mathbb{G}_2^k$

$\mathsf{udk}[\mathsf{id}] := ([d_i]_2)_{\mathsf{id}[i]=1} \in (\mathbb{G}_2^{1+k})^{(\mathsf{Ham}(\mathsf{id}))}$

Return $(\mathsf{usk}[\mathsf{id}], \mathsf{udk}[\mathsf{id}])$

Enc(id*): //Games G_0, $\boxed{\mathsf{G}_1\text{-}\mathsf{G}_2}$, G_2, G_3

$r \xleftarrow{\$} \mathbb{Z}_p^k$

$c_0^* = \mathbf{A} r \in \mathbb{Z}_p^{k+1}$

$c_0^* \xleftarrow{\$} \mathbb{Z}_p^{k+1}$

$h \xleftarrow{\$} \mathbb{Z}_p$; $\overline{c_0^*} \xleftarrow{\$} \mathbb{Z}_p^k$;

$\underline{c_0^*} := h + \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \overline{c_0^*} \in \mathbb{Z}_p$

$c_1^* = (\sum_i f_i(\mathsf{id}^*) \mathbf{Z}_i) r \in \mathbb{Z}_p^n$

$\boxed{c_1^* = \sum_i f_i(\mathsf{id}^*)(\mathbf{Y}_i^\top \mid x_i) c_0^* \in \mathbb{Z}_p^n}$

$c_1^* = \sum_i f_i(\mathsf{id}^*)(\mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1} \overline{c_0^*} + x_i \cdot h)$

$K^* = z_0' \cdot r \in \mathbb{Z}_p$.

$\boxed{K^* = (y_0'^\top \mid x_0') c_0^* \in \mathbb{Z}_p}$

$K^* = z_0' \cdot \overline{\mathbf{A}}^{-1} \overline{c_0^*} + x_0' \cdot h$

Return $\mathsf{K}^* = [K^*]_T$ and $\mathsf{C}^* = ([c_0^*]_1, [c_1^*]_1)$

Enc(id*): //Game G_3, $\boxed{\mathsf{G}_4}$

$h \xleftarrow{\$} \mathbb{Z}_p$; $\overline{c_0^*} \xleftarrow{\$} \mathbb{Z}_p^k$; $\underline{c_0^*} := h + \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1} \overline{c_0^*} \in \mathbb{Z}_p$

$c_1^* = \sum_i f_i(\mathsf{id}_i^*)(\mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1} \overline{c_0^*} + x_i \cdot h)$

$K^* = z_0' \cdot \overline{\mathbf{A}}^{-1} \overline{c_0^*} + x_0' \cdot h$

$\boxed{K^* \xleftarrow{\$} \mathbb{Z}_p}$

Return $\mathsf{K}^* = [K^*]_T$ and $\mathsf{C}^* = ([c_0^*]_1, [c_1^*]_1)$

Finalize(β): //Games G_0-G_4

Return $(\mathsf{id}^* \not\subseteq \mathcal{Q}_{\mathsf{ID}}) \wedge \beta$

Fig. 13. Games G_0-G_4 for the proof

Lemma 20. *There exists an adversary $\mathcal{B}_1$ with $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{A})$ and*

$$\mathbf{Adv}_{\mathcal{D}_k, \mathsf{GGen}}(\mathcal{B}_1) \ge |\Pr[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{G}_1^{\mathcal{A}} \Rightarrow 1]|.$$

Lemma 21. $\Pr[\mathsf{G}_3^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathsf{G}_2^{\mathcal{A}} \Rightarrow 1]$.

Proof. G_3 is simulated without using y_0' and $(Y_i)_{0 \le i \le \ell}$. By $\mathbf{Y}_i^\top = (\mathbf{Z}_i - x_i \underline{\mathbf{A}}) \overline{\mathbf{A}}^{-1}$, we have

$$\mathbf{D}_i = (\overline{\mathbf{A}}^{-1})^\top(\mathbf{Z}_i^\top \mathbf{B} - \underline{\mathbf{A}}^\top d_i) = \underbrace{(\overline{\mathbf{A}}^{-1})^\top(\mathbf{Z}_i^\top - \underline{\mathbf{A}}^\top x_i^\top)}_{\mathbf{Y}_i}\mathbf{B}$$

$$d_i = (\overline{\mathbf{A}}^{-1})^\top \cdot (\mathbf{Z}_i^\top t - \underline{\mathbf{A}}^\top \underbrace{x_i^\top t}_{d_i}) = \mathbf{Y}_i t.$$

as in Game G_2. And so, we have $[v]_2$, K^* and C^* are identical to G_2.

$\square$

Lemma 22. *There exists an adversary $\mathcal{B}_2$ with $\mathbf{T}(\mathcal{B}_2) \approx \mathbf{T}(\mathcal{A})$ and* $\mathsf{Adv}_{\mathsf{MAC}}^{\mathsf{dpr}_0\text{-}\mathsf{cma}}(\mathcal{B}_2) \geq |\Pr[\mathsf{G}_4^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{G}_3^{\mathcal{A}} \Rightarrow 1]|$

Proof. In G_4, we answer the $\mathsf{Enc}(\mathsf{id}^*)$ query by choosing random K^*. We construct algorithm $\mathcal{B}_2$ in Figure 14 to show the differences between G_4 and G_3 is bounded by the advantage of breaking dpr_0-cma security of MAC.

<u>Initialize:</u> $\mathbf{A} \xleftarrow{\$} \mathcal{D}_k$ $([\mathbf{B}]_2, ([x_i^\top \mathbf{B}]_2)_{0\leq i\leq \ell}) \xleftarrow{\$} \mathsf{Initialize}_{\mathsf{MAC}}$ $\forall i \in [\![0, \ell]\!]$: $\quad \mathbf{Z}_i \xleftarrow{\$} \mathbb{Z}_p^{n\times k}$; $z_0' \xleftarrow{\$} \mathbb{Z}_p^{1\times k}$ $\mathsf{pk} := (\mathcal{G}, [\mathbf{A}]_1, ([\mathbf{Z}_i]_1)_{0\leq i\leq \ell}, [z_0']_1)$ Return $(\mathsf{pk}, \mathsf{dk})$ <u>Enc(id^*):</u> //only one query $([h]_1, [h_0]_1, [h_1]_T) \xleftarrow{\$} \mathsf{Chal}(\mathsf{id}^*)$ $\overline{c_0^*} \xleftarrow{\$} \mathbb{Z}_p^k$; $\underline{c_0^*} := h + \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1}\overline{c_0^*} \in \mathbb{Z}_p$ $c_1^* = \sum_i f_i(\mathsf{id}^*)\mathbf{Z}_i \cdot \overline{\mathbf{A}}^{-1}\overline{c_0^*} + h_0$ $K^* = z_0' \cdot \overline{\mathbf{A}}^{-1}\overline{c_0^*} + h_1$ Return $\mathsf{K}^* = [K^*]_T$ and $\mathsf{C}^* = ([c_0^*]_1, [c_1^*]_1)$	<u>USKGen(id):</u> $\mathcal{Q}_{\mathsf{ID}} = \mathcal{Q}_{\mathsf{ID}} \cup \{\mathsf{id}\}$ $([t]_2, [u]_2, [T]_2, [u]_2, ([d_i]_2, [D_i]_2)) \xleftarrow{\$} \mathsf{Eval}(\mathsf{id})$ $v^\top = (t^\top \sum f_i(\mathsf{id})\mathbf{Z}_i + z_0' - u \cdot \underline{\mathbf{A}}) \cdot (\overline{\mathbf{A}})^{-1}$ $V = (\overline{\mathbf{A}}^{-1})^\top(\sum f_i(\mathsf{id})Z_i^\top \cdot \mathbf{T} - \underline{\mathbf{A}}^\top \cdot u)$ For $i, \mathsf{id}_i = 1$: $\quad e_i^\top = (t^\top \mathbf{Z}_i - d_i\underline{\mathbf{A}})\overline{\mathbf{A}}^{-1} \in \mathbb{Z}_p^{1\times k}$ $\quad E_i = (\overline{\mathbf{A}}^{-1})^\top(\mathbf{Z}_i^\top \mathbf{T} - \underline{\mathbf{A}}^\top \cdot D_i) \in \mathbb{Z}_p^{k\times \mu}$ $\mathsf{usk}[\mathsf{id}] := ([t]_2, [u]_2, [v]_2) \in \mathbb{G}_2^n \times \mathbb{G}_2^1 \times \mathbb{G}_2^k$ $\mathsf{udk}[\mathsf{id}] := ([T]_2, [u]_2, [\mathbf{V}]_2, [e_i]_2, [E_i]_2))$ Return $(\mathsf{usk}[\mathsf{id}], \mathsf{udk}[\mathsf{id}])$ <u>Finalize(β):</u> Return $(\mathsf{id}^* \not\preceq \mathcal{Q}_{\mathsf{ID}}) \wedge \mathsf{Finalize}_{\mathsf{MAC}}(\beta)$

Fig. 14. Description of $\mathcal{B}_2$ (having access to the oracles $\mathsf{Initialize}_{\mathsf{MAC}}, \mathsf{Eval}, \mathsf{Chal}, \mathsf{Finalize}_{\mathsf{MAC}}$ for the proof of Lemma 22.

We note that, in games G_3 and G_4, the values x_i and x_i' are hidden until the call to $\mathsf{Enc}(\mathsf{id}^*)$ (because the adversary is not allowed to query an id such that $\mathsf{id}^* \preceq \mathsf{id}$). In both games $\mathsf{DPR}\text{-}\mathsf{CMA}_{\mathsf{real}}$ and $\mathsf{DPR}_0\text{-}\mathsf{CMA}_{\mathsf{rand}}$, we have $h = \underline{c_0^*} - \underline{\mathbf{A}}\overline{\mathbf{A}}^{-1}\overline{c_0^*}$. Hence $h_0 = \sum f_i(\mathsf{m}_i)x_i \cdot (\underline{c_0^*} - \underline{\mathbf{A}} \cdot \overline{\mathbf{A}}^{-1}\overline{c_0^*})$ which implies c_1^* is distributed identically in games G_3 and G_4. If h_1 is uniform (i.e., $\mathcal{B}_2$ is in Game $\mathsf{DPR_0\text{-}CMA_{rand}}$) then the view of $\mathcal{A}$ is the same as in G_4. If h_1 is real (i.e., $\mathcal{B}_2$ is in Game $\mathsf{DPR\text{-}CMA_{real}}$) then $K^* = z_0' \cdot \overline{\mathbf{A}}^{-1} \overline{c_0^*} + x_0' \cdot h$, which means the view of $\mathcal{A}$ is the same as in G_3.

$\square$

The proof follows by combining Lemmas 19-22.