Identity-Based Encryption Gone Wild

Michel Abdalla[1], Dario Catalano[1], Alexander W. Dent[2],
John Malone-Lee[3], Gregory Neven[1,4], and Nigel P. Smart[3]

[1] Département d'Informatique, Ecole Normale Supérieure,
45 rue d'Ulm, 75230 Paris Cedex 05, France.
Email: {Michel.Abdalla,Dario.Catalano,Gregory.Neven}@ens.fr
[2] Information Security Group,
Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, United Kingdom.
Email: a.dent@rhul.ac.uk
[3] Department of Computer Science, University of Bristol,
Woodland Road, Bristol, BS8 1UB, United Kingdom.
Email: {malone,nigel}@cs.bris.ac.uk
[4] Department of Electrical Engineering, Katholieke Universiteit Leuven,
Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium.
Email: Gregory.Neven@esat.kuleuven.be

Abstract. In this paper we introduce the notion of identity based encryption with wildcards, or WIBE for short. This allows the encryption of messages to multiple parties with common fields in their identity strings, for example email groups in a corporate hierarchy. We propose a full security notion and give efficient implementations meeting this notion in the standard model and in the random oracle model.

1 Introduction

The concept of identity based cryptography was introduced by Shamir as early as in 1984 [12]. However, it took nearly twenty years for an efficient identity based encryption (IBE) scheme to be proposed. In 2000 and 2001 respectively Sakai, Ohgishi and Kasahara [11] and Boneh and Franklin [5] proposed IBE schemes based on elliptic curve pairings. Also, in 2001 Cocks proposed a system based on the quadratic residuosity problem [7].

One of the main application areas proposed for IBE is that of email encryption. In this scenario, given an email address, one can encrypt a message to the owner of the email address without needing to obtain an authentic copy of the owner's public key first. In order to decrypt the email the recipient must authenticate itself to a trusted authority who generates a private key corresponding to the email address used to encrypt the message.

Our work is motivated by the fact that many email addresses correspond to groups of users rather than single individuals. Consider the scenario where there is some kind of organisational hierarchy. Take as an example an organisation called ECRYPT which is divided into virtual labs, AZTEC and STVL

for example. In addition, these virtual labs are further subdivided into working groups WG1, WG2 and WG3, and an administrative group ADMIN. Finally, each working group may consist of many individual members. There are several extensions of the IBE primitive to such a hierarchical setting (HIBE) [9, 8]. The idea is that each level can issue keys to users on the level below. For example the owner of the ECRYPT key can issue decryption keys for ECRYPT.AZTEC and ECRYPT.STVL.

Suppose that we wish to send an email to all the members of the AZTEC.WG1 working group, which includes personal addresses ECRYPT.AZTEC.WG1.Nigel, ECRYPT.AZTEC.WG1.Dario and ECRYPT.AZTEC.WG1.John. Given a standard HIBE one would have to encrypt the message to each user individually. To address this limitation we introduce the concept of *identity based encryption with wildcards* (WIBE). The way in which decryption keys are issued is exactly as in a standard HIBE scheme; what differs is encryption. Our primitive allows the encrypter to replace any component of the recipient identity with a *wildcard* so that any identity matching the *pattern* can decrypt. Denoting wildcards by $*$, in the example above the encrypter would use the identity ECRYPT.AZTEC.WG1.$*$ to encrypt to all members of the AZTEC.WG1 group. To send a message to the administrative members of all virtual labs, one can simply encrypt to identity ECRYPT.$*$.ADMIN.$*$.

It is often suggested that identity strings should be appended with the date so as to add timeliness to the message, and so try to mitigate the problems associated with key revocation. Using our technique we can now encrypt to a group of users, with a particular date, by encrypting to an identity of the form ECRYPT.AZTEC.WG1.$*$.22Oct2006 for example. Thus any individual in ECRYPT.AZTEC.WG1 in possession of a decryption key for 22nd October 2006 will be able to decrypt.

Our paper proceeds as follows. In the next section we give an overview of existing material that we will build upon. We formally introduce our new primitive and describe an appropriate security model in Section 3. In Section 4 we describe a generic construction that realises a WIBE from any HIBE. The construction is very simple, yet unsatisfactory as it requires secret keys whose size is exponential in the number of levels of the underlying HIBE.

In Section 5 we turn to the problem of constructing a WIBE scheme with polynomial-size (with respect to all relevant parameters) ciphertexts and keys. We present an efficient WIBE scheme based on Waters' HIBE scheme [13], and prove its security by reducing to the security of Waters' HIBE scheme. The proof, just like that of Waters [13], is in the standard model. In the full version of this paper [1] we give two more efficient constructions, based on the Boneh-Boyen [3] and the Boneh-Boyen-Goh [4] HIBE schemes, and provide security proofs in the random oracle model [2]. We compare the efficiency and security of all our schemes in Section 6, and we also sketch how chosen-ciphertext security can be achieved by adapting the technique of Canetti *et al.* [6].

2 Basic Definitions

In this section we introduce some notation, computational problems and basic primitives that we will use throughout the rest of the paper. Let $\mathbb{N} = \{0, 1, \ldots\}$ be the set of natural numbers. Let ε be the empty string. If $n \in \mathbb{N}$, then $\{0,1\}^n$ denotes the set of n-bit strings, and $\{0,1\}^*$ is the set of all bit strings. More generally, if S is a set, then S^n is the set of n-tuples of elements of S. If S is finite, then $x \xleftarrow{\$} S$ denotes the assignment to x of an element chosen uniformly at random from S. If A is an algorithm, then $y \leftarrow \mathsf{A}(x)$ denotes the assignment to y of the output of A on input x, and if A is randomised, then $y \xleftarrow{\$} \mathsf{A}(x)$ denotes that the output of an execution of $\mathsf{A}(x)$ with fresh coins is assigned to y.

The decisional bilinear Diffie-Hellman assumption. Let $\mathbb{G}, \mathbb{G}_{\mathrm{T}}$ be multiplicative groups of prime order p with an admissible map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_{\mathrm{T}}$. By admissible we mean that the map is bilinear, non-degenerate and efficiently computable. Bilinearity means that for all $a, b \in \mathbb{Z}_p$ and all $g \in \mathbb{G}$ we have $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$. By non-degenerate we mean that $\hat{e}(g, g) = 1$ if and only if $g = 1$.

In such a setting we can define a number of computational problems. We shall be interested in the following problem, called the bilinear decisional Diffie-Hellman (BDDH) problem: For a generator $g \in \mathbb{G}$, given

$$g \ , \ A = g^a \ , \ B = g^b \ , \ C = g^c \text{ and } Z = \hat{e}(g, g)^z,$$

the problem is to determine whether $Z = \hat{e}(g, g)^{abc}$ for hidden values of a, b, c and z. Formally, we define this via a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The challenger first generates random values $a, b, c, z \xleftarrow{\$} \mathbb{Z}_p$ and then it flips a bit β. If $\beta = 1$ it passes $\mathcal{A}$ the tuple $(g, A, B, C, \hat{e}(g, g)^{abc})$, if $\beta = 0$ it passes the tuple $(g, A, B, C, \hat{e}(g, g)^z)$. The adversary $\mathcal{A}$ then must output its guess β' for β. The adversary has advantage ϵ in solving the BDDH problem if

$$\left|\Pr[\mathcal{A}(g, A, B, C, \hat{e}(g, g)^{abc}) = 1] - \Pr[\mathcal{A}(g, A, B, C, \hat{e}(g, g)^z) = 1]\right| \geq 2\epsilon,$$

where the probabilities are over the choice of a, b, c, z and over the random coins consumed by $\mathcal{A}$.

Definition 1. *The (t, ϵ) BDDH assumption holds if no t-time adversary has at least ϵ advantage in the above game.*

We note that throughout this paper we will assume that the time t of an adversary includes its code size, in order to exclude trivial "lookup" adversaries.

Identity-based encryption schemes. An identity-based encryption (IBE) scheme is a tuple of algorithms $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ providing the following functionality. The trusted authority runs Setup to generate a master key pair (mpk, msk). It publishes the master public key mpk and keeps the master secret key msk private. When a user with identity ID wishes to become part of the system, the trusted authority generates a user decryption key $d_{ID} \xleftarrow{\$}$

$\mathsf{KeyDer}(msk, ID)$, and sends this key over a secure and authenticated channel to the user. To send an encrypted message $\mathtt{m}$ to the user with identity ID, the sender computes the ciphertext $C \xleftarrow{\$} \mathsf{Enc}(mpk, ID, \mathtt{m})$, which can be decrypted by the user as $\mathtt{m} \leftarrow \mathsf{Dec}(d_{ID}, C)$. We refer to [5] for details on the security definitions for IBE schemes.

HIERARCHICAL IBE SCHEMES. In a hierarchical IBE (HIBE) scheme, users are organised in a tree of depth L, with the root being the master trusted authority. The identity of a user at level $0 \leq \ell \leq L$ in the tree is given by a vector $ID = (ID_1, \ldots, ID_\ell) \in (\{0,1\}^*)^\ell$. A HIBE scheme is a tuple of algorithms $\mathcal{HIBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ providing the same functionality as in an IBE scheme, except that a user $ID = (ID_1, \ldots, ID_\ell)$ at level ℓ can use its own secret key d_{ID} to generate a secret key for any of its children $ID' = (ID_1, \ldots, ID_\ell, ID_{\ell+1})$ via $d_{ID'} \xleftarrow{\$} \mathsf{KeyDer}(d_{ID}, ID_{\ell+1})$. Note that by iteratively applying the KeyDer algorithm, user ID can derive secret keys for any of its descendants $ID' = (ID_1, \ldots, ID_{\ell+\delta})$, $\delta \geq 0$. We will occasionally use the overloaded notation $d_{ID'} \xleftarrow{\$} \mathsf{KeyDer}(d_{ID}, (ID_{\ell+1}, \ldots, ID_{\ell+\delta}))$ to denote this process. The secret key of the root identity at level 0 is $d_\varepsilon = msk$. Encryption and decryption are the same as for IBE, but with vectors as identities instead of ordinary bit strings. For $1 \leq i \leq \ell$ and $I \subseteq \{1, \ldots, \ell\}$, we will occasionally use the notations $ID|_{\leq i}$ to denote the vector $(ID_1, \ldots, ID_i)$, $ID|_{>i}$ to denote $(ID_{i+1}, \ldots, \ell)$, and $ID|_I$ to denote $(ID_{i_1}, \ldots, ID_{i_{|I|}})$ where $i_1, \ldots, i_{|I|}$ are the elements of I in increasing order. Also, if $S \subset \mathbb{N}$, then we define $S|_{\leq i} = \{j \in S : j \leq i\}$ and $S|_{>i} = \{j \in S : j > i\}$.

The security of a HIBE scheme is defined through the following game. In a first phase, the adversary is given as input the master public key mpk of a freshly generated key pair $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}$ as input. In a chosen-plaintext attack (IND-ID-CPA), the adversary is given access to a key derivation oracle that on input of an identity $ID = (ID_1, \ldots, ID_\ell)$, returns the secret key $d_{ID} \xleftarrow{\$} \mathsf{KeyDer}(msk, ID)$ corresponding to identity ID. In a chosen-ciphertext attack (IND-ID-CCA), the adversary is additionally given access to a decryption oracle that for a given identity $ID = (ID_1, \ldots, ID_\ell)$ and a given ciphertext C returns the decryption $\mathtt{m} \leftarrow \mathsf{Dec}(\mathsf{KeyDer}(msk, ID), C)$.

At the end of the first phase, the adversary outputs two equal-length challenge messages $\mathtt{m}_0^*, \mathtt{m}_1^* \in \{0,1\}^*$ and a challenge identity $ID^* = (ID_1^*, \ldots, ID_{\ell^*}^*)$, where $0 \leq \ell^* \leq L$. The game chooses a random bit $b \xleftarrow{\$} \{0,1\}^*$, generates a challenge ciphertext $C^* \xleftarrow{\$} \mathsf{Enc}(mpk, ID^*, \mathtt{m}_b^*)$ and gives C^* as input to the adversary for the second phase, during which it gets access to the same oracles as during the first phase. The adversary wins the game if it outputs a bit $b' = b$ without ever having queried the key derivation oracle on any ancestor identity $ID = (ID_1^*, \ldots, ID_\ell^*)$ of ID^*, $\ell \leq \ell^*$, and, additionally, in the IND-ID-CCA case, without ever having queried (ID^*, C^*) to the decryption oracle.

Definition 2. *A HIBE scheme is $(t, q_{\mathrm{K}}, \epsilon)$ IND-ID-CPA-secure if all t-time adversaries making at most q_{K} queries to the key derivation oracle have at most advantage ϵ in winning the IND-ID-CPA game described above.*

Definition 3. *A HIBE scheme is* $(t, q_{\mathrm{K}}, q_{\mathrm{D}}, \epsilon)$ *IND-ID-CCA-secure if all* t*-time adversaries making at most* q_{K} *queries to the key derivation oracle and at most* q_{D} *queries to the decryption oracle have at most advantage* ϵ *in winning the IND-ID-CCA game described above.*

3 Identity-Based Encryption with Wildcards

Syntax. Identity-based encryption with wildcards (WIBE) schemes are essentially a generalisation of HIBE schemes where at the time of encryption, the sender can decide to make the ciphertext decryptable not just by a single target identity ID, but by a whole group of users whose identities match a certain pattern. Such a pattern is described by a vector $P = (P_1, \ldots, P_\ell) \in (\{0,1\}^* \cup \{*\})^\ell$, where $*$ is a special wildcard symbol. We say that identity $ID = (ID_1, \ldots, ID_{\ell'})$ *matches* P, denoted $ID \in_* P$, if and only if $\ell' \leq \ell$ and $\forall\, i = 1 \ldots \ell'$: $ID_i = P_i$ or $P_i = *$. Note that under this definition, any ancestor of a matching identity is also a matching identity. This is reasonable for our purposes because any ancestor can derive the secret key of a matching descendant identity anyway.

More formally, a WIBE scheme is a tuple of algorithms $\mathcal{WIBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ providing the following functionality. The root authority first generates a master key pair $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}$. A user with identity $ID = (ID_1, \ldots, ID_\ell)$ can use its own decryption key d_{ID} to derive a decryption key for any user $ID' = (ID_1, \ldots, ID_\ell, ID_{\ell+1})$ on the level below by calling $d_{ID'} \xleftarrow{\$} \mathsf{KeyDer}(d_{ID}, ID_{\ell+1})$. We will again use the overloaded notation $\mathsf{KeyDer}(d_{ID}, (ID_{\ell+1}, \ldots, ID_{\ell+\delta}))$ to denote iterative key derivation for descendants. The secret key of the root identity is $d_\varepsilon = msk$.

To create a ciphertext of message $\mathtt{m} \in \{0,1\}^*$ intended for all identities matching pattern $P = (P_1, \ldots, P_\ell)$, the sender computes $C \xleftarrow{\$} \mathsf{Enc}(mpk, P, \mathtt{m})$. Any of the intended recipients $ID \in_* P$ can decrypt the ciphertext using its own decryption key as $\mathtt{m} \leftarrow \mathsf{Dec}(d_{ID}, C)$. Correctness requires that for all key pairs (mpk, msk) output by Setup, all messages $\mathtt{m} \in \{0,1\}^*$, all $0 \leq \ell \leq L$, all patterns $P \in (\{0,1\}^* \cup \{*\})^\ell$, and all identities $ID \in (\{0,1\}^*)^{\ell'}$ such that $ID \in_* P$, $\mathsf{Dec}(\, \mathsf{KeyDer}(msk, ID)\, ,\, \mathsf{Enc}(mpk, P, \mathtt{m})\,) = \mathtt{m}$ with probability one.

Security. We define the security of WIBE schemes in a way that is very similar to the case of HIBE schemes, but where the adversary chooses a challenge pattern instead of an identity to which the challenge ciphertext will be encrypted. Of course, the adversary is not able to query the key derivation oracle for any identity that matches the challenge pattern, nor is it able to query the decryption oracle with the challenge ciphertext and any identity that matches the challenge pattern.

More specifically, security is defined through the following game with an adversary. In the first phase, the adversary is run on input of the master public key of a freshly generated key pair $(mpk, msk) \xleftarrow{\$} \mathsf{Setup}$. In a chosen-plaintext attack (IND-WID-CPA), the adversary is given access to a key derivation oracle that on input $ID = (ID_1, \ldots, ID_\ell)$ returns $d_{ID} \xleftarrow{\$} \mathsf{KeyDer}(msk, ID)$. In

a chosen-ciphertext attack (IND-WID-CCA), the adversary additionally has access to a decryption oracle that on input a ciphertext C and an identity $ID = (ID_1, \ldots, ID_\ell)$ returns $\mathfrak{m} \leftarrow \mathsf{Dec}(\mathsf{KeyDer}(msk, ID), C)$.

At the end of the first phase, the adversary outputs two equal-length challenge messages $\mathfrak{m}_0^*, \mathfrak{m}_1^*$ and a challenge pattern $P^* = (P_1^*, \ldots, P_{\ell^*}^*)$ where $0 \leq \ell^* \leq L$. The adversary is given a challenge ciphertext $C^* \stackrel{\$}{\leftarrow} \mathsf{Enc}(mpk, P^*, \mathfrak{m}_b^*)$ for a randomly chosen bit b, and is given access to the same oracles as during the first phase of the attack. The second phase ends when the adversary outputs a bit b'. The adversary is said to win the IND-WID-CPA game if $b' = b$ and if it never queried the key derivation oracle for the keys of any identity that matches the target pattern (i.e., any ID such that $ID \in_* P^*$). Also, in a chosen-ciphertext attack (IND-WID-CCA), the adversary cannot query the decryption oracle on C^* with any matching identity $ID \in_* P^*$.

Definition 4. *A WIBE scheme is $(t, q_\mathrm{K}, \epsilon)$ IND-WID-CPA-secure if all t-time adversaries making at most q_K queries to the key derivation oracle have at most advantage ϵ in winning the IND-WID-CPA game described above.*

Definition 5. *A WIBE scheme is $(t, q_\mathrm{K}, q_\mathrm{D}, \epsilon)$ IND-WID-CCA-secure if all t-time adversaries making at most q_K queries to the key derivation oracle and at most q_D queries to the decryption oracle have at most advantage ϵ in winning the IND-WID-CCA game described above.*

4 A Generic Construction

We first point out that a WIBE scheme can be constructed from any HIBE scheme, albeit with a secret key size that is exponential in the depth of the hierarchy tree. Let "$*$" be a dedicated bitstring that is not allowed to occur as a user identity. Then the secret key of a user with identity $(ID_1, \ldots, ID_\ell)$ in the WIBE scheme contains the HIBE secret keys of all patterns matching this identity, i.e. the secret keys of all 2^ℓ identities $(ID'_1, \ldots, ID'_\ell)$ such that $ID'_i = ID_i$ or $ID'_i =$ "$*$" for all $i = 1, \ldots, \ell$. To encrypt to a pattern $(P_1, \ldots, P_\ell)$, one uses the HIBE scheme to encrypt to the identity obtained by replacing each wildcard in the pattern with the "$*$" string, i.e. the identity $(ID_1, \ldots, ID_\ell)$ where $ID_i =$ "$*$" if $P_i = *$ and $ID_i = P_i$ otherwise. Decryption is done by selecting the appropriate secret key from the list and using the decryption algorithm of the HIBE scheme.

The efficiency of the WIBE scheme thus obtained is roughly the same as that of the underlying HIBE scheme, except that the size of the secret key is 2^ℓ times that of a secret key in the underlying HIBE scheme. This may be acceptable for some applications, but may not be for others. Moreover, from a theoretical point of view, it is interesting to investigate whether WIBE schemes exist with overhead polynomial in all parameters. We answer this question in the affirmative here by presenting direct schemes with secret key size (and, unfortunately, also ciphertext size) linear in ℓ.

5 A Construction from Waters' HIBE Scheme

5.1 Waters' HIBE Scheme

In [13], Waters argued that his IBE scheme can easily be modified into a L-level HIBE scheme as per [3]. Here we explicitly present this construction as it will be useful in the understanding of our construction of a WIBE scheme.

Setup. The trusted authority chooses random generators g_1 and g_2 from $\mathbb{G}$ and a random value $\alpha \xleftarrow{\$} \mathbb{Z}_p$. For $i = 1, \ldots, L$ and $j = 0, \ldots, n$, it chooses group elements $u_{i,j} \xleftarrow{\$} \mathbb{G}$ where L is the maximum hierarchy depth and n is the length of an identity string. Next, it computes $h_1 \leftarrow g_1^\alpha$ and $h_2 \leftarrow g_2^\alpha$. The master public key is $mpk = (g_1, g_2, h_1, u_{1,0}, \ldots, u_{L,n})$, the corresponding master secret key is $msk = h_2$.

Key Derivation. A user's identity is given by a vector $ID = (ID_1, \ldots, ID_\ell)$ where each ID_i is a n-bit string, applying a collision-resistant hash function if necessary. Let "$j \in ID_i$" denote a variable j iterating over all bit positions $1 \leq j \leq n$ such that the j-th bit of ID_i is one. Using this notation, for $i = 1, \ldots, L$, we define the function

$$F_i(ID_i) = u_{i,0} \prod_{j \in ID} u_{i,j}$$

where the $u_{i,j}$ are the elements in the master public key. To compute the decryption key for identity ID from the master secret key, first random values $r_1, \ldots, r_\ell \xleftarrow{\$} \mathbb{Z}_p$ are chosen, then the private key d_{ID} is constructed as

$$(a_0, a_1, \ldots, a_\ell) = \left(h_2 \prod_{i=1}^{\ell} F_i(ID_i)^{r_i} \ , \ g_1^{r_1}, \ldots, g_1^{r_\ell} \right).$$

A secret key for identity $ID = (ID_1, \ldots, ID_\ell)$ can be computed by its parent with identity $ID|_{\leq \ell-1}$ as follows. Let $d_{ID|_{\leq \ell-1}} = (a_0, a_1, \ldots, a_{\ell-1})$. The parent chooses $r_\ell \xleftarrow{\$} \mathbb{Z}_p$ and outputs

$$d_{ID} = (a_0 \cdot F_i(ID_i)^{r_\ell} \ , \ a_1, \ldots, a_{\ell-1} \ , \ g_1^{r_\ell}) .$$

Encryption. To encrypt a message $\mathfrak{m} \in \mathbb{G}_\mathrm{T}$ for identity $ID = (ID_1, \ldots, ID_\ell)$, the sender chooses $t \xleftarrow{\$} \mathbb{Z}_p$; the ciphertext $C = (C_1, C_2, C_3)$ is computed as

$$C_1 \leftarrow g_1^t \ , \ C_2 \leftarrow \left(C_{2,i} = F_i(ID_i)^t \right)_{i=1,\ldots,\ell} \ , \ C_3 \leftarrow \mathfrak{m} \cdot \hat{e}(h_1, g_2)^t \ .$$

Decryption. If the receiver is the root authority (i.e., the empty identity $ID = \varepsilon$) holding the master key $msk = h_2$, then he can recover the message by computing $C_3 / \hat{e}(C_1, h_2)$. Any other receiver with identity $ID = (ID_1, \ldots, ID_\ell)$ and decryption key $d_{ID} = (a_0, a_1, \ldots, a_\ell)$ decrypts a ciphertext $C = (C_1, C_2, C_3)$ as $C_3 \cdot \prod_{i=1}^{\ell} \hat{e}\,(a_i, C_{2,i}) \,/ \hat{e}\,(C_1, a_0)$.

Waters [13] informally states that the above HIBE scheme is IND-ID-CPA secure in the sense that if there is an adversary with advantage ϵ against the HIBE making q_K private key extraction queries, then there is an algorithm solving the BDDH problem with advantage $\epsilon' = O((nq_\mathrm{K})^L \epsilon)$.

5.2 A Waters-based WIBE Scheme

We first introduce some additional notation. If $P = (P_1, \ldots, P_\ell)$ is a pattern, then let $|P| = \ell$ be the length of P, let $\mathrm{W}(P)$ be the set containing all wildcard indices in P, i.e. the indices $1 \leq i \leq \ell$ such that $P_i = *$, and let $\overline{\mathrm{W}}(P)$ be the complementary set containing all non-wildcard indices. Clearly $\mathrm{W}(P) \cap \overline{\mathrm{W}}(P) = \emptyset$ and $\mathrm{W}(P) \cup \overline{\mathrm{W}}(P) = \{1, \ldots, \ell\}$. We also extend the notations $P|_{\leq i}$, $P|_{>i}$ and $P|_I$ that we introduced for identity vectors to patterns in the natural way.

Let $\mathcal{W}a\text{-}\mathcal{HIBE} = (\mathsf{Setup}, \mathsf{KeyDer}, \mathsf{Enc}, \mathsf{Dec})$ be the HIBE scheme described in Section 5.1. From $\mathcal{W}a\text{-}\mathcal{HIBE}$, we can build a WIBE scheme $\mathcal{W}a\text{-}\mathcal{WIBE} = (\mathsf{Setup}', \mathsf{KeyDer}', \mathsf{Enc}', \mathsf{Dec}')$, where Setup' and KeyDer' are equal to those of the $\mathcal{W}a\text{-}\mathcal{HIBE}$ scheme (i.e., $\mathsf{Setup}' = \mathsf{Setup}$ and $\mathsf{KeyDer}' = \mathsf{KeyDer}$), and Enc' and Dec' are as follows.

Encryption. To create a ciphertext of message $\mathtt{m} \in \mathbb{G}_\mathrm{T}$ intended for all identities matching pattern $P = (P_1, \ldots, P_\ell)$, the sender chooses $t \xleftarrow{\$} \mathbb{Z}_p$ and outputs the ciphertext $C = (P, C_1, C_2, C_3, C_4)$, where

$$C_1 \leftarrow g_1^t \qquad C_2 \leftarrow \left(C_{2,i} = F_i(P_i)^t\right)_{i \in \overline{\mathrm{W}}(P)}$$
$$C_3 \leftarrow \mathtt{m} \cdot \hat{e}(h_1, g_2)^t \qquad C_4 \leftarrow \left(C_{4,i,j} = u_{i,j}^t\right)_{i \in \mathrm{W}(P),\ j=0,\ldots,n}$$

Decryption. If the receiver is the root authority (i.e., the empty identity $ID = \varepsilon$) holding the master key $msk = h_2$, then it can recover the message by computing $C_3/\hat{e}(C_1, h_2)$. Any other receiver with identity $ID = (ID_1, \ldots, ID_\ell)$ matching the pattern P to which the ciphertext was created (i.e., $ID \in_* P$) can decrypt the ciphertext $C = (P, C_1, C_2, C_3, C_4)$ by computing $C_2' = \left(C_{2,i}'\right)_{i=1,\ldots,\ell}$ as

$$C_{2,i}' = F_i(ID_i)^t \leftarrow \begin{cases} C_{2,i} & \text{if } i \in \overline{\mathrm{W}}(P) \\ C_{4,i,0} \cdot \prod_{j \in ID_i} C_{4,i,j} & \text{if } i \in \mathrm{W}(P) \end{cases}$$

and by using his secret key to decrypt the ciphertext $C' = (C_1, C_2', C_3)$ via the Dec algorithm of the $\mathcal{W}a\text{-}\mathcal{HIBE}$ scheme.

Theorem 6. *Let $\mathcal{W}a\text{-}\mathcal{HIBE}$ be the HIBE scheme in Section 5.1 and let L be the maximum hierarchy depth. Let $\mathcal{W}a\text{-}\mathcal{WIBE}$ be the WIBE scheme derived from $\mathcal{W}a\text{-}\mathcal{HIBE}$ as described in Section 5.2. If $\mathcal{W}a\text{-}\mathcal{HIBE}$ is $(t, q_\mathrm{K}, \epsilon)$ IND-ID-CPA-secure then $\mathcal{W}a\text{-}\mathcal{WIBE}$ is $(t', q_\mathrm{K}', \epsilon')$ IND-WID-CPA-secure where*

$$t' = t + t_{\exp} Ln(1 + q_\mathrm{K}) \ , \quad q_\mathrm{K}' = q_\mathrm{K} \ , \quad \epsilon' \geq \epsilon / 2^L$$

and $t_{\exp}$ is the time it takes to perform an exponentiation in $\mathbb{G}$.

Proof. The proof of Theorem 6 is by contradiction. That is, we first assume that there exists an adversary $\mathcal{A}$ that breaks the IND-WID-CPA-security of the $\mathcal{W}a\text{-}\mathcal{WIBE}$ scheme and then we show how to efficiently build another adversary $\mathcal{B}$ which uses $\mathcal{A}$ to break the security of the $\mathcal{W}a\text{-}\mathcal{HIBE}$ scheme.

Let $mpk_{\mathrm{H}} = (g_1, g_2, h_1, u_{1,0}, \ldots, u_{L,n})$ be the master public key of the $\mathcal{Wa}$-$\mathcal{HIBE}$ scheme that adversary $\mathcal{B}$ receives as input for its first phase. The idea of the proof is that $\mathcal{B}$ will guess upfront where in the challenge pattern P^* the wildcards are going to be, and "project" the non-wildcard levels of the identity tree of the WIBE scheme onto the first levels of the HIBE scheme. In particular, $\mathcal{B}$ will reuse values $u_{i,j}$ from mpk_{H} for the non-wildcard levels, and will embed new values $u'_{i,j}$ values of which $\mathcal{B}$ knows the discrete logarithms for wildcard levels.

First, $\mathcal{B}$ guesses a random vector $\hat{P} = (\hat{P}_1, \ldots, \hat{P}_L) \xleftarrow{\$} \{\varepsilon, *\}^L$. Define the projection function $\pi : \{1, \ldots, L\} \to \{0, \ldots, L\}$ such that

$$\pi(i) = 0 \text{ if } i \in \mathrm{W}(\hat{P}) \quad \text{and} \quad \pi(i) = i - \left|\mathrm{W}(\hat{P})|_{\leq i}\right| \text{ otherwise.}$$

Intuitively, $\mathcal{B}$ will "project" identities at level i of the WIBE scheme onto level $\pi(i)$ of the HIBE scheme whenever $\pi(i) \neq 0$. Next, the adversary $\mathcal{B}$ runs adversary $\mathcal{A}$ providing it as input for its first phase a public-key $mpk_{\mathrm{W}} = (g_1, g_2, h_1, u'_{1,0}, \ldots, u'_{L,n})$, where for all $1 \leq i \leq L$ and $0 \leq j \leq n$, the elements $u'_{i,j}$ are generated as $u'_{i,j} \leftarrow g_1^{\alpha_{i,j}}$ where $\alpha_{i,j} \xleftarrow{\$} \mathbb{Z}_p$ if $i \in \mathrm{W}(\hat{P})$, and $u'_{i,j} \leftarrow u_{\pi(i),j}$ otherwise. Define functions $F'_i(ID'_i) = u'_{i,0} \prod_{j \in ID'_i} u'_{i,j}$. Notice that $mpk_{\mathcal{A}}$ is distributed exactly as it would be if produced by the setup algorithm described in Section 5.2.

During the first phase, $\mathcal{B}$ has to answer all the key derivation queries $ID' = (ID'_1, \ldots, ID'_\ell)$ that $\mathcal{A}$ is allowed to ask. For that, $\mathcal{B}$ first computes the corresponding identity on the HIBE tree $ID = ID'|_{\overline{\mathrm{W}}(\hat{P})}$, which is the identity obtained by removing from ID' all components at levels where $\hat{P}$ contains a wildcard. That is, the identity ID is obtained from ID' by projecting the component at level i of the WIBE onto level $\pi(i)$ of the HIBE if $\pi(i) \neq 0$. $\mathcal{B}$ then queries its own key derivation oracle for the $\mathcal{Wa}$-$\mathcal{HIBE}$ scheme on input ID to get the key $d = (a_0, \ldots, a_{\pi(\ell)})$. From this, it computes the key $d' = (a'_0, \ldots, a'_\ell)$ as

$$a'_0 \leftarrow a_0 \cdot \textstyle\prod_{i \in \mathrm{W}(\hat{P})} F'_i(ID'_i)^{r_i} \;,\; a'_i \leftarrow \begin{cases} g_1^{r_i} & \text{if } i \in \mathrm{W}(\hat{P}) \\ a_{\pi(i)} & \text{if } i \in \overline{\mathrm{W}}(\hat{P}) \end{cases}$$

where $r_i \xleftarrow{\$} \mathbb{Z}_p$ for all $i \in \mathrm{W}(\hat{P})$. At the end of its first phase, $\mathcal{A}$ outputs the challenge pattern $P^* = (P^*_1, \ldots, P^*_{\ell^*})$ and challenge messages $\mathfrak{m}^*_0, \mathfrak{m}^*_1$. If $\mathrm{W}(P^*) \neq \mathrm{W}(\hat{P})$ then $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ outputs the corresponding HIBE identity $ID^* = P^*|_{\overline{\mathrm{W}}(P^*)}$ together with challenge messages $\mathfrak{m}^*_0, \mathfrak{m}^*_1$. Let $C^* = (C^*_1, C^*_2, C^*_3)$ be the challenge ciphertext that $\mathcal{B}$ receives in return from its challenger, meaning that C^* is an encryption of $\mathfrak{m}^*_b$ with respect to the identity ID^*, where b is the secret bit chosen at random by the challenger. $\mathcal{B}$ sets $C'^*_1 \leftarrow C^*_1$, $C'^*_2 \leftarrow C^*_2$, $C'^*_3 \leftarrow C^*_3$ and $C'^*_4 \leftarrow (C^{*\alpha_{i,j}}_1)_{i \in \mathrm{W}(P^*),\, j=0,\ldots,n}$ and sends to $\mathcal{A}$ the ciphertext $C'^* = (P^*, C'^*_1, C'^*_2, C'^*_3, C'^*_4)$ as the input for its second phase. During the second phase, $\mathcal{A}$ is then allowed to issue more key derivation queries, which are answered by $\mathcal{B}$ exactly as in the first phase. When $\mathcal{A}$ outputs a bit b', $\mathcal{B}$ outputs b' and stops.

In order to analyse the success probability of $\mathcal{B}$, we first need to show that the simulation it provides to $\mathcal{A}$ is correct. The secret key $d' = (a'_0, \ldots, a'_\ell)$ returned for identity $(ID'_1, \ldots, ID'_\ell)$ can be seen to be correctly distributed since if $a'_i = g_1^{r_i}$ for $1 \leq i \leq \ell$ then

$$\begin{aligned} a'_0 &= h_2 \cdot \textstyle\prod_{i \in \overline{\mathrm{W}}(\hat{P})} F_{\pi(i)}(ID'_i)^{r_i} \cdot \prod_{i \in \mathrm{W}(\hat{P})} F'_i(ID'_i)^{r_i} \\ &= h_2 \cdot \prod_{i \in \overline{\mathrm{W}}(\hat{P})} \left(u_{\pi(i),0} \textstyle\prod_{j \in ID'_i} u_{\pi(i),j} \right)^{r_i} \cdot \prod_{i \in \mathrm{W}(\hat{P})} F'_i(ID'_i)^{r_i} \\ &= h_2 \cdot \prod_{i \in \overline{\mathrm{W}}(\hat{P})} \left(u'_{i,0} \textstyle\prod_{j \in ID'_i} u'_{i,j} \right)^{r_i} \cdot \prod_{i \in \mathrm{W}(\hat{P})} F'_i(ID'_i)^{r_i} \\ &= h_2 \cdot \textstyle\prod_{i=1}^{\ell} F'_i(ID'_i)^{r_i} \end{aligned}$$

Moreover, the challenge ciphertext $C'^* = (P^*, C'^*_1, C'^*_2, C'^*_3, C'^*_4)$ sent to $\mathcal{A}$ can be seen to be correctly formed when $\mathrm{W}(P^*) = \mathrm{W}(\hat{P})$ as follows. Consider the ciphertext $C^* = (C^*_1, C^*_2, C^*_3)$ that $\mathcal{B}$ receives back from the challenger after outputting $(ID^*, \mathfrak{m}^*_0, \mathfrak{m}^*_1)$ where $ID^* = P^*|_{\overline{\mathrm{W}}(P^*)}$. We know that, for unknown values $t \in \mathbb{Z}_p$ and $b \in \{0, 1\}$, $C^*_1 = g^t$, $C^*_3 = \mathfrak{m}^*_b \cdot \hat{e}(h_1, g_2)^t$ and

$$C^*_2 \;=\; \left(C^*_{2,i} = F_i(ID^*_i)^t\right)_{i=1,\ldots,\pi(\ell^*)} \;=\; \left(C'^*_{2,i} = F'_i(P^*_i)^t\right)_{i \in \overline{\mathrm{W}}(P^*)} \;.$$

Since $\mathcal{B}$ sets $C'^*_1 = C^*_1$, $C'^*_2 = C^*_2$ and $C'^*_3 = C^*_3$, it follows that C'^*_1, C'^*_2 and C'^*_3 are of the correct form. To show that C^*_4 is correctly formed, notice that $u'_{i,j} = g_1^{\alpha_{i,j}}$ for indices $i \in \mathrm{W}(P^*)$ and $j = 0, \ldots, n$. Thus, $C'^*_{4,i,j} = (C^*_1)^{\alpha_{i,j}} = {g_1}^{t\,\alpha_{i,j}} = (g_1^{\alpha_{i,j}})^t = {u'_{i,j}}^t$ as required.

We also need to argue that $\mathcal{B}$ does not query its key derivation oracle on any identities that are considered illegal in the IND-ID-CPA game when its guess for $\mathrm{W}(P^*)$ is correct. Illegal identities are the challenge identity $ID^* = P^*|_{\overline{\mathrm{W}}(P^*)}$ or any ancestors of it, i.e. any $ID^*|_{\leq \ell}$ for $\ell \leq \ell^*$. Adversary $\mathcal{B}$ only makes such queries when $\mathcal{A}$ queries its key derivation oracle on an identity $ID' = (ID'_1, \ldots, ID'_{\ell'})$ such that $\ell' \leq \ell^*$ and $ID'_i = P^*_i$ for all $i \in \overline{\mathrm{W}}(P^*)|_{\leq \ell'}$. By our matching definition, this would mean that $ID' \in_* P^*$, which is illegal in the IND-WID-CPA game as well. Note that, whenever $\ell' > \ell^*$, we always have that $|ID| > |ID^*|$ since $\mathrm{W}(\hat{P})|_{> \ell^*} = \emptyset$.

To conclude the proof, we notice that the success probability of $\mathcal{B}$ is at least that of $\mathcal{A}$ when its guess for $\mathrm{W}(P^*)$ is correct. Let ϵ be the probability that $\mathcal{A}$ wins the IND-WID-CPA game. Thus, it follows that the overall success probability of $\mathcal{B}$ winning the IND-ID-CPA game is at least $\epsilon' \geq \epsilon / 2^L$.

Remark 7. The factor of 2^L in the security reduction is not a major drawback given the state of the art in HIBE constructions, which also lose this factor. In addition, we only lose a factor of L^2 when encrypting to patterns with a single sequence of consecutive wildcards, for example $(ID_1, *, *, *, ID_5)$ or $(ID_1, *, *)$.

Scheme	$\|mpk\|$	$\|d\|$	$\|C\|$	Dec	Assumption	RO
Generic	$\|mpk_{\mathcal{HIBE}}\|$	$2^L \cdot \|d_{\mathcal{HIBE}}\|$	$\|C_{\mathcal{HIBE}}\|$	$\mathsf{Dec}_{\mathcal{HIBE}}$	$\mathcal{HIBE}$ is IND-ID-CPA	No
$\mathcal{Wa}$-$\mathcal{WIBE}$	$(n+1)L+3$	$L+1$	$(n+1)L+2$	$L+1$	BDDH	No
$\mathcal{BB}$-$\mathcal{WIBE}$	$2L+3$	$L+1$	$2L+2$	$L+1$	BDDH	Yes
$\mathcal{BBG}$-$\mathcal{WIBE}$	$L+4$	$L+2$	$L+3$	2	L-BDHI	Yes

Fig. 1. Efficiency and security comparison between the generic scheme of Section 4, the $\mathcal{Wa}$-$\mathcal{WIBE}$ scheme of Section 5.2, and the $\mathcal{BB}$-$\mathcal{WIBE}$ and $\mathcal{BBG}$-$\mathcal{WIBE}$ schemes presented in the full version [1]. The schemes are compared in terms of master public key size ($|mpk|$), user secret key size ($|d|$), ciphertext size ($|C|$), decryption time (Dec), the security assumption under which the scheme is proved secure, and whether this proof is in the random oracle model or not. (The generic construction does not introduce any random oracles, but if the security proof of the HIBE scheme is in the random oracle model, then the WIBE obviously inherits this property.) Values refer to the underlying HIBE scheme for the generic scheme, and to the number of group elements ($|mpk|$, $|d|$, $|C|$) or pairing computations (Dec) for the other schemes. L is the maximal hierarchy depth and n is the bit length of an identity string. Figures are worst-case values, usually occurring for identities at level L with all-wildcard ciphertexts. L-BDHI refers to the decisional bilinear Diffie-Hellman inversion assumption [10, 3].

6 Alternative Constructions and Extensions

In the full version of this paper [1], we present two alternative WIBE implementations, namely the $\mathcal{BB}$-$\mathcal{WIBE}$ scheme based on the Boneh-Boyen HIBE scheme [3] and the $\mathcal{BBG}$-$\mathcal{WIBE}$ scheme based on the Boneh-Boyen-Goh HIBE scheme [4], respectively. We omit them here due to space restrictions. Both of these schemes have security proofs in the standard model under a weaker security notion that can be seen as a variant of selective-ID security with wildcards. Security under the full notion presented in Section 3 can be achieved in the random oracle model [2] at the cost of losing a factor q_{H}^L in the reduction, where q_{H} is the number of an adversary's random oracle queries and L is the maximum depth of the hierarchy. Both schemes have efficiency polynomial in all parameters, unlike the generic construction of Section 4, and offer advantages over the $\mathcal{Wa}$-$\mathcal{WIBE}$ scheme in master public key length, ciphertext size and encryption/decryption time. A comparison between all our schemes is provided in Fig. 1.

While the efficiency of our direct schemes is polynomial in all parameters, we stress that their security degrades exponentially with the hierarchy depth L. So just as is the case for the current state of the art in HIBE schemes, we have to leave the construction of a WIBE scheme with polynomial efficiency *and* security in all parameters as an open problem.

Also in the full version of the paper [1], we achieve chosen ciphertext security by adapting the technique of Canetti, Halevi and Katz [6]. In particular, we show that we may use a $(2L+2)$-level CPA-secure WIBE and a strongly unforgeable signature scheme (SigGen, Sign, Verify) to construct an L-level CCA-secure WIBE.

Acknowledgments

We would like to thank James Birkett, Jacob Schuldt, Brent Waters and the anonymous referees of ICALP 2006 for their valuable input. This work was supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The first two authors were supported in part by France Telecom R&D as part of the contract CIDRE, between France Telecom R&D and École normale supérieure. The fifth author is a Postdoctoral Fellow of the Research Foundation – Flanders (FWO-Vlaanderen), and was supported in part by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government.

References

1. M. Abdalla, D. Catalano, A. W. Dent, J. Malone-Lee, G. Neven, and N. P. Smart. Identity-based encryption gone wild. Cryptology ePrint Archive, 2006.
2. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 1993*, pages 62–73, 1993.
3. D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer-Verlag, 2004.
4. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer-Verlag, 2005.
5. D. Boneh and M. K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.
6. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer-Verlag, 2004.
7. C. Cocks. An identity based encryption scheme based on quadratic residues. In *Cryptography and Coding, 8th IMA International Conference*, volume 2260 of *LNCS*, pages 360–363. Springer-Verlag, 2001.
8. C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 548–566. Springer-Verlag, 2002.
9. J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 466–481. Springer-Verlag, 2002.
10. S. Mitsunari, R. Saka, and M. Kasahara. A new traitor tracing. *IEICE Transactions*, E85-A(2):481–484, 2002.
11. R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *SCIS 2000*, Okinawa, Japan, 2000.
12. A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer-Verlag, 1985.
13. B. R. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer-Verlag, 2005.