# Variants of Waters' Dual System Primitives Using Asymmetric Pairings

## (Extended Abstract)

Somindu C. Ramanna[1], Sanjit Chatterjee[2], and Palash Sarkar[1]

[1] Applied Statistics Unit
Indian Statistical Institute
203, B.T. Road, Kolkata
India 700108.
e-mail: {somindu_r,palash}@isical.ac.in
[2] Department of Computer Science and Automation
Indian Institute of Science
Bangalore, India 560012.
e-mail: sanjit@csa.iisc.ernet.in

**Abstract.** Waters, in 2009, introduced an important technique, called dual system encryption, to construct identity-based encryption (IBE) and related schemes. The resulting IBE scheme was described in the setting of symmetric pairing. A key feature of the construction is the presence of random tags in the ciphertext and decryption key. Later work by Lewko and Waters removed the tags and proceeding through composite-order pairings led to a more efficient dual system IBE scheme using asymmetric pairings whose security is based on non-standard but static assumptions. In this work, we have systematically simplified Waters 2009 IBE scheme in the setting of asymmetric pairing. The simplifications retain tags used in the original description. This leads to several variants, the first one of which is based on standard assumptions and in comparison to Waters' original scheme reduces ciphertexts and keys by two elements each. Going through several stages of simplifications, we finally obtain a simple scheme whose security can be based on two standard assumptions and a natural and minimal extension of the decision Diffie-Hellman problem for asymmetric pairing groups. The scheme itself is also minimal in the sense that apart from the tags, both encryption and key generation use exactly one randomiser each. This final scheme is more efficient than both the previous dual system IBE scheme in the asymmetric setting due to Lewko and Waters and the more recent dual system IBE scheme due to Lewko. We extend the IBE scheme to hierarchical IBE (HIBE) and broadcast encryption (BE) schemes. Both primitives are secure in their respective full models and have better efficiencies compared to previously known schemes offering the same level and type of security.

**Keywords:** identity-based encryption, dual system encryption, asymmetric pairing.

## 1 Introduction

Constructions of identity-based encryption schemes constitute one of the most challenging problems of public-key cryptography. The notion of IBE was proposed in [14] and solved in [3, 7]. This lead to a great deal of research on the topic. The solution in [3], though simple and elegant, had several features which were not satisfactory from a theoretical point of view.

In this work, we will be interested in IBE schemes built from bilinear pairings. Till date, most pairing based cryptographic schemes have been based on a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, where $\mathbb{G}$ is a prime-order group of elliptic curve points over a finite field and $\mathbb{G}_T$ is a subgroup of a finite field. Such maps arise from Weil and Tate pairings and there is an extensive literature on efficient implementation of such maps. Since the two components of the domain of $e$ are same, such an $e$ is called a symmetric pairing. Another kind of pairings, where the order of $\mathbb{G}$ is composite has been proposed [4]. Such pairings are called composite-order pairings and provide additional flexibility in designing schemes. The trade-off, however, is that computing the pairing itself becomes significantly slower and also the representation of the group elements becomes substantially longer.

Symmetric pairings (over prime order groups), are neither the most general nor the most efficient of possible pairings over elliptic curves. A general bilinear map is of the form $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$ is a prime-order group of points of an elliptic curve over a finite field $\mathbb{F}$ and $\mathbb{G}_2$ is a group (of the same prime-order) of points of the same curve over an extension of $\mathbb{F}$. Such maps are called asymmetric pairings. Studies [16, 8, 5] have indicated that compared to symmetric pairings, asymmetric pairings are much faster and more compact to implement.

An important work on pairing based IBE is [17] which builds upon earlier work in [1, 2] to provide an efficient IBE scheme with full security without random oracles. Variants have been reported [6, 12] which result in IBE schemes which are efficient and have practical sized parameters. Though important, a drawback of the scheme in [17] is that the size of the public parameters grows linearly with the security parameter.

In a major innovation, Waters [18] introduced a new technique – called dual system encryption – for construction of IBE schemes and related primitives. The scheme presented in [18] has the feature that the size of the public parameters is constant while retaining full security. Dual system encryption is by itself an interesting notion and worthy of further investigation. The goal of a better understanding of dual system encryption would be to obtain IBE schemes with improved efficiency compared to the one proposed in [18].

An immediate follow-up work [11] took the route of composite-order pairings. Such pairing groups have 'more structure' which can possibly help in getting a clearer understanding of the technique. (Waters remarks in [18] that his scheme was first obtained for composite order groups.) The approach taken by [11] is to look at a realization of the IBE scheme of [1] in the setting of composite order groups so as to obtain adaptive-id security. They also gave a conversion of their

composite-order IBE scheme to an IBE scheme using prime-order asymmetric pairing. In a very recent work [10], the framework of dual system encryption has been thoroughly investigated and an IBE scheme using prime-order pairing has been presented. We note that the conversion from composite-order to prime-order pairings in [11] and considering prime-order groups in [10] are motivated by efficiency considerations.

Waters IBE scheme in [18] is based on symmetric pairings. The security of the scheme is based on the hardness of the decision linear (DLin) and the decision bilinear Diffie-Hellman (DBDH) assumptions. It is of interest to convert this to asymmetric pairings. For one thing, this will enable faster and smaller implementations which will arise from the advantages of asymmetric pairings over their symmetric variants. There is, however, another reason. Use of asymmetric pairings brings forward the possibility of reducing the number of group elements in ciphertexts and keys. In fact, Waters [18] himself mentions: "using the SXDH assumption we might hope to shave off three group elements from both ciphertexts and private keys". The rationale for this comment is that for asymmetric pairings with no known efficiently computable isomorphisms between the groups $\mathbb{G}_1$ and $\mathbb{G}_2$, the decision Diffie-Hellman (DDH) assumption holds for both $\mathbb{G}_1$ and $\mathbb{G}_2$. This is the symmetric external Diffie-Hellman (SXDH) assumption. For symmetric pairings the DDH assumption does not hold in $\mathbb{G}$. Using the SXDH assumption will potentially lead to a simpler scheme requiring a lesser number of group elements.

Following up on the above mentioned remark by Waters, we have systematically investigated the various possibilities for using asymmetric pairings. To start the study, we performed a straightforward conversion to the setting of asymmetric pairings. The scheme in [18] is quite complex. Several scalars are used in the public parameters, encryption and key generation. These have definite and inter-connected roles in the security proof. Our first task was to pin down the relationships between these scalars and separate them out. This enabled us to work with one group of scalars with minimal changes to other groups.

With a good understanding of the roles of the scalars, we are able to apply simplifications in a stage-wise manner. The first simplification gives an IBE scheme (Scheme 1) which shrinks ciphertexts and keys by two elements each and whose security can be based on DDH1 (DDH assumption in $\mathbb{G}_1$), DLin and DBDH assumptions. We argue that the DDH2 assumption cannot be directly used. So, the afore-mentioned suggestion by Waters cannot be fulfilled. On the other hand, we show that using a natural and minimal extension of the DDH2 assumption, a significantly more efficient scheme (Scheme 6) can be obtained.

Waters original scheme [18] used random tags in the ciphertext and the decryption key. Simplification of this scheme by both Lewko-Waters [11] and Lewko [10] yielded IBE schemes which did not use such tags. In contrast, all our simplifications retain the tags used in the original description [18]. Even then, we are able to obtain significant simplifications and efficiency improvements. This suggests that for the purpose of simplification as an IBE it is not important to

do away with the tags. Removing them has other positive consequences such as obtaining a constant size ciphertext hierarchical IBE [11].

Scheme 6 has the interesting feature that, apart from the tags, exactly one randomiser each is used for encryption and key generation which is minimal in case of ciphertext. However, it is not known whether the key generation could be made deterministic within the dual system framework. To show that our simplification retains the flexibility of the original technique by Waters, we obtain an analogue of the HIBE scheme and prove it secure in the full security model. This HIBE scheme inherits all the security properties from [18], but, provides improved efficiency. From this HIBE scheme we construct an adaptively secure BE scheme which is more efficient than all the previously known BE schemes with adaptive security. We provide only the construction of the BE scheme here; the full version of this paper [13] contains the security proof. The construction and proof for the HIBE scheme will appear in the full version [13].

A comparison of the features of various IBE schemes based on the dual system technique is shown in Tables 1 and 2. The columns $\#\mathcal{PP}$, $\#\mathcal{MSK}$, #cpr, #key provide the number of group elements in the public parameters, the master secret key, ciphertexts and decryption keys. The public parameters and ciphertexts consist of elements of $\mathbb{G}_1$ while the master secret key and decryption keys consist of elements of $\mathbb{G}_2$. Encryption efficiency counts the number of scalar multiplications in $\mathbb{G}_1$ while decryption efficiency counts the number of pairings that are required. Key generation (a less frequent activity) efficiency is given by the number of scalar multiplications in $\mathbb{G}_2$. Currently, Scheme 6 is the most efficient among all the known dual system IBE schemes.

| scheme | $\#\mathcal{PP}$ | $\#\mathcal{MSK}$ | #cpr | #key | enc eff | dec eff | key gen | assump |
|---|---|---|---|---|---|---|---|---|
| Waters-09 [18] | 13 | 5 | 9 | 8 | 14 | 9 | 12 | DLin, DBDH |
| Lewko-11 [10] | 24 | 30 | 6 | 6 | 24 | 6 | 6 | DLin |
| Scheme 1 | 9 | 8 | 7 | 6 | 10 | 6 | 9 | DDH1, DLin, DBDH |

**Table 1.** Comparison of dual system IBE schemes secure under standard assumptions. Waters-09 and Lewko-11 use symmetric pairings while Scheme 6 uses asymmetric pairings.

| scheme | $\#\mathcal{PP}$ | $\#\mathcal{MSK}$ | #cpr | #key | enc eff | dec eff | key gen | assump |
|---|---|---|---|---|---|---|---|---|
| LW [11] | 9 | 6 | 6 | 6 | 9 | 6 | 10 | LW1, LW2, DBDH |
| Scheme 6 | 6 | 7 | 4 | 4 | 7 | 3 | 6 | DDH1, DDH2v, DBDH |

**Table 2.** Comparison of dual system IBE schemes secure under non-standard but static assumptions. Both the schemes use asymmetric pairings. DDH1 is a weaker assumption than LW1 and DDH2v is a weaker assumption than LW2.

The figures in the table indicate that Scheme 6 is more efficient than Lewko-Waters scheme. In particular, decryption in Scheme 6 is about twice as fast as that of LW scheme (note that both constructions are based on Type-3 pairings). Since Scheme 1 and Scheme 6 use Type-3 pairings which offer much better

performance compared to symmetric pairings, the gain in speed over Waters' scheme and Lewko's scheme cannot be quantified just in terms of the number of operations performed. It also depends on the performance gain of asymmetric pairings over their symmetric variants for the chosen security level.

## 2 Prerequisites

We follow standard definitions and corresponding full security models of IBE, HIBE and BE schemes. Here we briefly describe asymmetric pairings and related assumptions. For more details on these the reader is referred to [16, 8, 5].

### 2.1 Bilinear maps

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be cyclic groups of prime order $p$. $\mathbb{G}_1$ and $\mathbb{G}_2$ are written additively while $\mathbb{G}_T$ is written multiplicatively. A cryptographic bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ has the following properties.

1. **Bilinearity:** For elements $A_1, B_1 \in \mathbb{G}_1$ and $A_2, B_2 \in \mathbb{G}_2$, $e(A_1 + B_1, A_2) = e(A_1, A_2)e(B_1, A_2)$ and $e(A_1, A_2 + B_2) = e(A_1, A_2)e(A_1, B_2)$.
2. **Non-degeneracy:** If $e(P_1, P_2) = 1_T$, the identity of $\mathbb{G}_T$, then either $P_1$ is the identity of $\mathbb{G}_1$ or $P_2$ is the identity of $\mathbb{G}_2$.
3. **Efficiency:** The map $e$ is efficiently computable.

A bilinear map is called symmetric or a Type-1 bilinear map if $\mathbb{G}_1 = \mathbb{G}_2$; otherwise it is asymmetric. Asymmetric bilinear maps are further classified into Type-2 and Type-3 bilinear maps. In the Type-2 setting, there is an efficiently computable isomorphism either from $\mathbb{G}_1$ to $\mathbb{G}_2$ or from $\mathbb{G}_2$ to $\mathbb{G}_1$ whereas in the Type-3 setting no such isomorphisms are known. Previous works [16, 8, 5] have established that the Type-3 setting is the most efficient from an implementation point of view.

We introduce some notation: Given generators $P_1$ of $\mathbb{G}_1$ and $P_2$ of $\mathbb{G}_2$ and elements $R_1 \in \mathbb{G}_1$ and $R_2 \in \mathbb{G}_2$, the notation $R_1 \sim R_2$ indicates that $R_1$ has the same discrete logarithm to base $P_1$ as that of $R_2$ to base $P_2$. For a set $\mathbb{X}$, let $x \in_{\mathrm{R}} \mathbb{X}$ denote that $x$ is a uniform random element of $\mathbb{X}$.

In the following, we will assume the availability of a Type-3 bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ where $\mathbb{G}_1 = \langle P_1 \rangle$, $\mathbb{G}_2 = \langle P_2 \rangle$ and both $\mathbb{G}_1$ and $\mathbb{G}_2$ are groups of the same prime order $p$. Being of prime order, any non-identity element of $\mathbb{G}_1$ is a generator of the group and the same holds for $\mathbb{G}_2$.

### 2.2 Hardness Assumption

We introduce a new hardness assumption for Type-3 pairings. Here we provide a discussion of this. The other standard hardness assumptions required in this work are DDH in $\mathbb{G}_1$, DLin and DBDH assumptions.

Let $P_1$ and $P_2$ be random generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. The DDH problem in $\mathbb{G}_1$ is to decide, given $(P_1, x_1P_1, x_2P_1, P_2, Z_1)$, whether $Z_1 = x_1x_2P_1$

or $Z_1$ is a random element of $\mathbb{G}_1$. Here $x_1, x_2 \in_R \mathbb{Z}_p$. Similarly one can define the DDH assumption in $\mathbb{G}_2$. In this case, an instance will have the form $(P_1, P_2, x_1P_2, x_2P_2, Z_2)$ and the task is to determine whether $Z_2 = x_1x_2P_2$ or whether $Z_2$ is a random element of $\mathbb{G}_2$. For convenience we will denote the DDH problem in $\mathbb{G}_1$ as DDH1 and that in $\mathbb{G}_2$ as DDH2. The symmetric external Diffie-Hellman (SXDH) assumption is that both DDH1 and DDH2 problems are hard. Note that for a symmetric pairing (i.e., for $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G} = \langle P \rangle$), DDH is easy to solve by comparing $e(P, Z_2)$ with $e(x_1P, x_2P)$.

We will use DDH1 in our proofs. But DDH2 is not directly applicable to our proofs. An instance of DDH2 has a single element $P_1$ of $\mathbb{G}_1$. For our proofs, we will require some information about $x_1P_1$ to be carried as part of the instance. If the instance is directly augmented by $x_1P_1$, then the problem becomes easy, since one can compute the pairing $e(x_1P_1, x_2P_2)$ and compare to $e(P_1, Z_2)$. Suppose that instead of $x_1P_1$ we include the elements $zP_1$ and $zx_1P_1$ where $z$ is chosen randomly from $\mathbb{Z}_p$. This pair of elements carries some information about $x_1P_1$, but, not the element itself. An instance will now be $(P_1, zP_1, zx_1P_1, P_2, x_1P_2, x_2P_2, Z_2)$. It, however, is easy to check whether $Z_2$ equals $x_1x_2P_2$ by checking whether $e(zx_1P_1, x_2P_2)$ equals $e(zP_1, Z_2)$. This suggests that the information about $zP_1$ itself needs to be blinded by another randomiser. So, instead of having $zP_1$ directly, the elements $dP_1, dzP_1$ and $dP_2$ are included where $d$ is a random element of $\mathbb{Z}_p$. The information about $x_1P_1$ is carried by the elements $dP_1, dzP_1, zx_1P_1$ and $dP_2$. Augmenting an instance of DDH2 with these elements embeds information about $x_1P_1$ but, does not seem to provide any way to use this information to determine whether $Z_2$ is real or random. The entire thing can now be formulated as an assumption in the following manner.

**Assumption DDH2v.** Let $P_1$, $P_2$ be random generators of $\mathbb{G}_1$, $\mathbb{G}_2$ respectively and let $x_1, x_2, d, z$ be random elements of $\mathbb{Z}_p$. The DDH2v problem is to decide, given $(P_1, dP_1, dzP_1, zx_1P_1, P_2, dP_2, x_1P_2, x_2P_2, Z_2)$, whether $Z_2 = x_1x_2P_2$ or $Z_2$ is a random element of $\mathbb{G}_2$.

This corresponds to a two-level blinding of $x_1P_1$. We have seen that providing $x_1P_1$ directly or using a single-level blinding makes the problem easy. So, a two-level blinding is the minimum that one has to use to get to an assumption about hardness.

The assumption DDH2v (the "v" stands for variant) is no harder than DDH2. This is because an instance of DDH2v contains an embedded instance of DDH2 and an algorithm to solve DDH2 can be invoked on this embedded instance to solve the instance of DDH2v. On the other hand, there is no clear way of using an algorithm to solve DDH2v to solve DDH2. Intuitively, this is due to the fact that an instance of DDH2v contains some information about $x_1P_1$ whereas an instance of DDH2 does not contain any such information.

In our reduction, we will use the assumption DDH2v. Since assumption DDH2v does not appear earlier in the literature, it is a non-standard assumption. Having said this, we would also like to remark that DDH2v arises naturally as a minimal assumption when one tries to augment an instance of DDH2 with

some information about $x_1 P_1$ while maintaining the hardness of the problem. A proof of security of this assumption in the generic group model is provided in the full version [13]. We feel that assumption DDH2v will have applications elsewhere for schemes based on asymmetric pairings.

## 3 Framework for Conversion

Our goal is to transform Waters-2009 IBE scheme to the asymmetric setting so that we can reduce the number of components both in the ciphertext and the key. To that end, we first perform a straightforward conversion of Waters IBE from the setting of symmetric pairing to the setting of asymmetric pairing. (See [18] for the original description of Waters 2009 scheme.)

Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a Type 3 bilinear map and let $P_1$ and $P_2$ be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. After the conversion, either the ciphertext or the key will consist of elements of $\mathbb{G}_1$; the other will consist of elements from $\mathbb{G}_2$. Elements of $\mathbb{G}_1$ have shorter representation compared to those of $\mathbb{G}_2$. For encryption, we want the ciphertext to be short and hence we choose its elements to be from $\mathbb{G}_1$. The public parameters will consist of elements of $\mathbb{G}_1$ whereas the master secret key will consist of elements of $\mathbb{G}_2$. We note that if the final goal were to construct a signature scheme, then one would perform a conversion where the secret key consists of elements of $\mathbb{G}_1$.

A straightforward conversion will have the same structure as the one described in [18]. We use the convention in this and later schemes that the subscript 1 will denote elements of $\mathbb{G}_1$ while the subscript 2 will denote elements of $\mathbb{G}_2$. Further, messages are elements of $\mathbb{G}_T$ and identities are elements of $\mathbb{Z}_p$.

To generate the public parameters $\mathcal{PP}$, first choose $\alpha, b, a_1, a_2$ at random from $\mathbb{Z}_p$ and consider the following. Let $v, v'$ and $v''$ be random elements of $\mathbb{Z}_p$ and define $V_2 = vP_2$, $V_2' = v'P_2$ and $V_2'' = v''P_2$. Let $\tau = v + a_1 v'$ and $\tau' = v + a_2 v''$. Set $T_1 = \tau P_1$ and $T_1' = \tau' P_1$. The $\mathcal{PP}$ will have elements $Q_1, U_1, W_1 \in \mathbb{G}_1$ and correspondingly the master secret key will have elements $Q_2, U_2, W_2 \in \mathbb{G}_2$ with $Q_2 \sim Q_1$, $U_2 \sim U_1$ and $W_2 \sim W_1$. The structure of the $\mathcal{PP}$ and the $\mathcal{MSK}$ are as follows.

$$\begin{aligned}\mathcal{PP} \quad &: (P_1, bP_1, a_1P_1, a_2P_1, ba_1P_1, ba_2P_1, T_1, T_1', bT_1, bT_1',\\ &\quad Q_1, W_1, U_1, e(P_1,P_2)^{ba_1\alpha}).\\ \mathcal{MSK}&: (P_2, \alpha P_2, a_1\alpha P_2, V_2, V_2', V_2'', Q_2, W_2, U_2).\end{aligned}$$

**Encrypt**$(M, \mathsf{id}, \mathcal{PP})$: Randomisers $s_1, s_2, t, \mathsf{ctag}$ are chosen from $\mathbb{Z}_p$ and define $s = s_1 + s_2$. The ciphertext is $(C_0, C_1, \ldots, C_7, E_1, E_2, \mathsf{ctag})$ where the various elements are defined as follows.

$$\begin{aligned}&C_0 = M \cdot e(P_1,P_2)^{ba_1\alpha s_2}\\ &C_1 = bsP_1,\ C_2 = ba_1s_1P_1,\ C_3 = a_1s_1P_1,\ C_4 = ba_2s_2P_1,\\ &C_5 = a_2s_2P_1, C_6 = s_1T_1 + s_2T_1',\ C_7 = s_1bT_1 + s_2bT_1' - tW_1\\ &E_1 = t(\mathsf{id}Q_1 + \mathsf{ctag}W_1 + U_1),\ E_2 = tP_1\end{aligned}$$

**KeyGen**($\mathsf{id}, \mathcal{MSK}, \mathcal{PP}$): Randomisers $r_1, r_2, z_1, z_2, \mathsf{ktag}$ are chosen from $\mathbb{Z}_p$ and define $r = r_1 + r_2$. The key $\mathcal{SK}_{\mathsf{id}}$ is $(K_1, \ldots, K_7, \mathsf{ktag})$ where the various elements are defined as follows.

$$K_1 = a_1\alpha P_2 + rV_2,\ K_2 = -\alpha P_2 + rV_2' + z_1 P_2,\ K_3 = -z_1 b P_2$$
$$K_4 = rV_2'' + z_2 P_2,\ K_5 = -z_2 b P_2,\ K_6 = r_2 b P_2,\ K_7 = r_1 P_2$$
$$D = r_1(\mathsf{id}Q_2 + \mathsf{ktag}W_2 + U_2).$$

The decryption algorithm (as described by Waters) requires 9 pairings and succeeds only if $\mathsf{ctag}$ in the ciphertext is not equal to $\mathsf{ktag}$ of the decryption key, an event which occurs with overwhelming probability (see [18] for the details).

Waters defines algorithms to generate semi-functional ciphertexts and keys. These cannot be computed without knowledge of the secret components and are only used in the security reduction. They are defined such that one should be able to decrypt a semi-functional ciphertext with a normal key and a normal ciphertext with a semi-functional key; but decryption of a semi-functional ciphertext with a semi-functional key should fail.

*Semi-functional Ciphertext:* Let $C_0', \ldots, C_7', E_1', E_2', \mathsf{ctag}$ be the ciphertext elements normally generated by the **Encrypt** algorithm for message $M$ and identity $\mathsf{id}$. Choose $\mu \in \mathbb{Z}_p$ at random. Let $V_1' = v'P_1$ and $V_1'' = v''P_1$ so that $V_1' \sim V_2'$ and $V_1'' \sim V_2''$. The semi-functional ciphertext generation algorithm will modify the normal ciphertext as: $C_0 = C_0'$, $C_1 = C_1'$, $C_2 = C_2'$, $C_3 = C_3'$, $E_1 = E_1'$, $E_2 = E_2'$ and

$$C_4 = C_4' + ba_2\mu P_1,\ C_5 = C_5' + a_2\mu P_1,\ C_6 = C_6' - a_2\mu V_1'',\ C_7 = C_7' - ba_2\mu V_1''.$$

*Semi-functional Key:* Let $K_1', \ldots, K_7', D', \mathsf{ktag}$ be secret key components normally generated by the **KeyGen** algorithm for the identity $\mathsf{id}$. Choose at random $\gamma \in \mathbb{Z}_p$. The semi-functional key generation algorithm will modify the normal key as: $K_3 = K_3'$, $K_5 = K_5'$, $K_6 = K_6'$, $K_7 = K_7'$, $D = D'$ and

$$K_1 = K_1' - a_1 a_2 \gamma P_2,\ K_2 = K_2' + a_2\gamma P_2,\ K_4 = K_4' + a_1\gamma P_2.$$

It is easy to see that one can decrypt a semi-functional ciphertext with a normal key and a normal ciphertext with a semi-functional key. However, decryption of a semi-functional ciphertext with a semi-functional key will fail because the masking factor $e(P_1, P_2)^{ba_1\alpha s_2}$ will be blinded by the factor $e(P_1, P_2)^{ba_1a_2\mu\gamma}$.

**Security proof.** The security argument for the scheme proceeds through $q + 3$ games where $q$ is the number of key extraction queries made by the adversary. These games are

$$\mathrm{Game}_{real}, \mathrm{Game}_0, \ldots, \mathrm{Game}_q, \mathrm{Game}_{final}.$$

The transition between these games can be seen as three different reductions.

**First reduction:** The transition from $\text{Game}_{real}$ to $\text{Game}_0$ is made by replacing the challenge ciphertext by a semi-functional ciphertext. It is argued that detecting this change should be hard.

**Second reduction:** There is a sequence of $q$ changes from $\text{Game}_{k-1}$ to $\text{Game}_k$ (for $k = 1, \ldots, q$). The $k$-th change is as follows. For the queries numbered 1 to $k-1$, the adversary is given a semi-functional key; for queries numbered $k+1$ to $q$, the adversary is given a normal key. For the $k$-th query, the adversary is given a response such that deciding whether the response is normal or semi-functional is hard. The challenge ciphertext is semi-functional as in the first reduction.

**Third reduction:** This tackles the transition from $\text{Game}_q$ to $\text{Game}_{final}$. At this point, all responses to key extraction queries are semi-functional and so is the challenge ciphertext. In the last transition, the challenge ciphertext is changed such that deciding whether it is the encryption of a message or whether it is statistically independent of the challenge messages is hard.

The first and second reductions are based on the hardness of the DLin problem whereas the third reduction is based on the hardness of the DBDH problem. In the proof, the second reduction is the most complex step. The subtle point is that the simulator should not be able to generate a semi-functional ciphertext for the $k$-th identity which will allow it to easily determine whether the key for this identity is semi-functional or not. This is ensured by using algebraic techniques from [1] to create $\mathsf{ktag}$ using a pair-wise independent function so that the simulator is able to create a semi-functional ciphertext for $\mathsf{id}_k$ only with $\mathsf{ctag} = \mathsf{ktag}$, in which case decryption fails unconditionally and hence the simulator gains no information.

### 3.1 An Analysis

Our conversion to asymmetric pairing and subsequent simplifications are based on an analysis of the various scalars used in the scheme and their respective roles in the proof. Based on the scheme itself and a study of the three reductions used by Waters, we make the following observations.

1. $\mathcal{PP}$ uses the scalars $a_1, a_2$ and $b$, while $\mathcal{MSK}$ uses the scalars $\alpha$ and $a_1$.
2. Key generation uses scalar randomisers $r_1, r_2$ and $z_1, z_2$. The scalar $r$ is set to $r_1 + r_2$. We will call this the *split of* $r$.
3. Ciphertext generation uses the scalar randomisers $s_1, s_2$ and $t$. The scalar $s$ is set to $s_1 + s_2$. We will call this the *split of* $s$.
4. The first two reductions in Waters proof are based on the DLin assumption. The first reduction uses the split of $s$ whereas the second reduction uses the split of $r$.

For conversion to asymmetric pairing, the following points are to be noted. These have been inferred from a careful analysis of the security proof in [18].

1. The scalar $\alpha$ needs to be retained.

2. The tags are chosen randomly and they play a crucial role in the security argument. We do not consider the question of removing tags in this paper. If the tags are removed, then it will be necessary to introduce copies of the identity-hash (as done in [11]) to obtain the functionality of tags in the semi-functional components. This leads to an increase in the number of elements in the ciphertext and key.
3. There are three basic possibilities for simplification: remove the split of $s$; remove the split of $r$; remove $z_1, z_2$.
4. Getting rid of $a_1$ and $a_2$ and using a single $a$ will eliminate the requirement of the split of $s$. This also means that the separate $z_1$ and $z_2$ are not required and instead a single $z$ can be used.
5. Removing the split of $r$ does not have a direct influence on the other scalars.
6. Removing the split of $r$ and also $z_1, z_2$ means that the scalar $b$ is no longer required.
7. In all but one of our schemes, the scalar $t$ is kept either as part of the ciphertext or as part of the key. In the final scheme, we show that the scalar $t$ can also be removed. For this scheme, there is a single randomiser $s$ for the ciphertext and a single randomiser $r$ for the key, excluding the tags.
8. If the first reduction is to be based on DLin, then the split of $s$ and $a_1$,$a_2$ must be retained. If the split is removed, then we can base the first reduction on DDH1.
9. If the split of $r$ is retained, then the second reduction has to be based on DLin. If it is removed, we can no longer base the second reduction on DLin. However, it can neither be based on DDH2 for the following reason. An instance of DDH2 will provide $P_1$ and some elements of $\mathbb{G}_2$. Apart from $P_1$ no other element of $\mathbb{G}_1$ is provided. The $\mathcal{PP}$ consists of elements of $\mathbb{G}_1$ which have to be related to the instance in some way. Just having $P_1$ does not provide any way to construct the $\mathcal{PP}$ in the second reduction. So, removing the split of $r$ implies that the second reduction can be based on neither DLin nor DDH2. The assumption DDH2v introduced in Section 2 provides the necessary mechanism for carrying the proof through.

Based on the above points, we explore the different natural ways in which Waters 2009 IBE scheme can be converted to asymmetric pairing. These are discussed below.

**Scheme 1:** Remove the split of $s$. This eliminates the requirement of having separate $a_1, a_2$ and $z_1, z_2$. Reductions of ciphertext and key are by two elements each. Removing the split of $s$ allows the first reduction to be based on DDH1. Since the split of $r$ is retained, the second reduction is still based on DLin.

**Scheme 2.** Retain the split of $s$; this means that separate $a_1$ and $a_2$ are required. Remove the split of $r$ and also remove $z_1$ and $z_2$; this means that $b$ can be removed. Leads to reductions of ciphertext and key by 3 elements each. The first reduction of the proof can be based on DLin, but, the second reduction cannot be based on either DLin or DDH2.

**Scheme 3:** Remove the split of $s$; retain the split of $r$ but, remove $z$. Reductions of ciphertext and key are by 3 elements each. In the proof, the first reduction can be based on DLin. The second reduction cannot be based on DDH2. Neither can it be based on DLin. This requires a more involved reasoning which we provide in the full version [13].

**Scheme 4:** Remove the splits of both $r$ and $s$, but, retain $z$. Ciphertext and key are reduced by 3 elements each. In the proof, the first reduction can be based on DDH1, but, the second reduction cannot be based on either DLin or DDH2.

**Scheme 5:** Remove the splits of both $r$ and $s$ and also remove $z$. Ciphertext and keys are reduced by 4 elements each. As in the previous case, the first reduction of the proof can be based on DDH1, but, the second reduction cannot be based on either DLin or DDH2.

**Scheme 6:** In Schemes 1 to 5, the randomiser $t$ is present in the ciphertext. In Scheme 6, the splits of both $r$ and $s$ are removed; $z$ is removed and the role of $t$ is played by $s$. This leads to a scheme where there is exactly one randomiser for encryption and exactly one randomiser for key generation. Compared to Waters' IBE [18], ciphertext size is reduced by 5 elements and the key size by 4 elements. The first reduction of the proof can be based on DDH1, while the second reduction is based on assumption DDH2v.

In Table 3, we provide the use of scalars in the various schemes. This illustrates the manner in which the simplification has been obtained.

| scheme | $\mathcal{PP}$ | $\mathcal{MSK}$ | key gen | enc |
|---|---|---|---|---|
| Waters-09 [18] | $\alpha, a_1, a_2, b$ | $\alpha, a_1$ | $r_1, r_2, (r = r_1 + r_2), z_1, z_2$ | $s_1, s_2, (s = s_1 + s_2), t$ |
| Scheme 1 | $\alpha, a, b$ | $\alpha, b$ | $r_1, r_2, (r = r_1 + r_2), z$ | $s, t$ |
| Scheme 2 | $\alpha, a_1, a_2$ | $\alpha$ | $r$ | $s_1, s_2, (s = s_1 + s_2), t$ |
| Scheme 3 | $\alpha, a, b$ | $\alpha, b$ | $r_1, r_2, (r = r_1 + r_2)$ | $s, t$ |
| Scheme 4 | $\alpha, a, b$ | $\alpha, b$ | $r, z$ | $s, t$ |
| Scheme 5 | $\alpha, a$ | $\alpha$ | $r$ | $s, t$ |
| Scheme 6 | $\alpha, a$ | $\alpha$ | $r$ | $s$ |

**Table 3.** Usage of scalars in various schemes. Note that all the schemes use ktag for key generation and ctag for encryption.

## 4 Constructions

In this section, we provide the description of Scheme 6. In the full version [13], the description of Scheme 1 along with its security proof are provided. For Schemes 2 to 5, only the descriptions are provided in the full version. These schemes primarily serve the purpose of showing the stepping stones in moving from Scheme 1 to Scheme 6. In Section 5, we present a security proof for Scheme 6.

### 4.1 Scheme 6

Descriptions of $\mathcal{PP}$, $\mathcal{MSK}$, ciphertext generation, key generation and decryption are provided.

Parameters $P_1, P_2, Q_1, W_1, U_1, Q_2, W_2, U_2, \alpha$ are chosen as described in Section 3. Let $a, v, v'$ be random elements of $\mathbb{Z}_p$. Set $V_2 = vP_2$, $V_2' = v'P_2$ and $\tau = v + av'$ so that $\tau P_2 = V_2 + aV_2'$.

$\mathcal{PP}$ : $(P_1, aP_1, \tau P_1, Q_1, W_1, U_1, e(P_1, P_2)^\alpha)$.
$\mathcal{MSK}$: $(P_2, \alpha P_2, V_2, V_2', Q_2, W_2, U_2)$.

**Encrypt**$(M, \mathsf{id}, \mathcal{PP})$: Choose random $s, \mathsf{ctag}$ from $\mathbb{Z}_p$; ciphertext $\mathcal{C}$ is given by $(C_0, C_1, C_2, C_3, E, \mathsf{ctag})$ where the elements are defined as follows.

$C_0 = M \cdot e(P_1, P_2)^{\alpha s}$,
$C_1 = sP_1,\ C_2 = asP_1,\ C_3 = -\tau sP_1 + sW_1,\ E = s(\mathsf{id}Q_1 + \mathsf{ctag}W_1 + U_1)$.

**KeyGen**$(\mathsf{id}, \mathcal{MSK}, \mathcal{PP})$: Choose random $r, \mathsf{ktag}$ from $\mathbb{Z}_p$; the secret key $\mathcal{SK}_{\mathsf{id}}$ is $(K_1, K_2, K_3, D, \mathsf{ktag})$ where the elements are defined as follows.

$K_1 = \alpha P_2 + rV_2,\ K_2 = rV_2',\ K_3 = rP_2,\ D = r(\mathsf{id}Q_2 + \mathsf{ktag}W_2 + U_2)$.

**Decrypt** $(\mathcal{C}, \mathsf{id}, \mathcal{SK}_{\mathsf{id}}, \mathcal{PP})$: As before, decryption succeeds only when $\mathsf{ctag} \neq \mathsf{ktag}$. Define $\vartheta = (\mathsf{ctag} - \mathsf{ktag})^{-1}$. Decryption is done by unmasking the message as follows.

$$M = \frac{C_0}{e(C_1, K_1 + \vartheta D)e(C_2, K_2)e(C_3 - \vartheta E, K_3)}$$

The correctness of decryption is shown by the following calculations. We break up the denominator into two parts - $A_1$ and $A_2$ such that the product $A_1A_2$ gives the masking factor.

$$\begin{aligned}
A_1 &= e(C_1, \vartheta D)e(-\delta E, K_3) \\
&= e(C_1, D)^\vartheta e(-E, K_3)^\vartheta \\
&= e(sP_1, r(\mathsf{id}Q_2 + \mathsf{ktag}W_2 + U_2))^\vartheta e(-s(\mathsf{id}Q_1 + \mathsf{ctag}W_1 + U_1), rP_2)^\vartheta \\
&= e(-(\mathsf{id}Q_1 + \mathsf{ktag}W_1 + U_1), P_2)^{-rs\vartheta} e(\mathsf{id}Q_1 + \mathsf{ctag}W_1 + U_1, P_2)^{-rs\vartheta} \\
&= e(\vartheta(\mathsf{ctag} - \mathsf{ktag})W_1, P_2)^{rs} \\
&= e(W_1, P_2)^{-rs} \\
A_2 &= e(C_1, K_1)e(C_2, K_2)e(C_3, K_3) \\
&= e(sP_1, \alpha P_2 + rV_2)e(asP_1, rV_2')e(\tau sP_1 + sW_1, rP_2) \\
&= e(P_1, P_2)^{\alpha s} e(P_1, V_2 + aV_2' - \tau P_2)^{rs} e(W_1, P_2)^{rs} \\
&= e(P_1, P_2)^{\alpha s} e(W_1, P_2)^{rs}
\end{aligned}$$

**Extension to HIBE:** Waters extends the IBE scheme in [18] in a natural way to a HIBE scheme. In the full version of this paper [13], we show that our simplification of Waters scheme retains the original flexibility and describe a HIBE which extends Scheme 6. This HIBE scheme is secure under the DDH1, DDH2v and the DBDH assumptions and provides lesser and smaller parameters and better efficiencies of key generation, delegation, encryption and decryption compared to the HIBE in [18]. The security proof for the HIBE follows the Shi-Waters model [15] and is given in [13].

**Conversion to Signature Scheme:** There is a "dual" of Scheme 6 where the ciphertext elements are in $\mathbb{G}_2$ and decryption keys consist of elements of $\mathbb{G}_1$. Using Naor's observation, this dual of Scheme 6 can be converted to a secure signature scheme. The signatures will be composed of elements of $\mathbb{G}_1$ and will be smaller than the signatures obtained by the conversion of Waters' 2009 scheme to a signature scheme. In a similar manner, one can convert the dual of our HIBE to obtain a HIBS scheme where signatures consist of elements of $\mathbb{G}_1$.

### 4.2 Broadcast Encryption

The full version of Waters paper described a public key broadcast encryption (BE) scheme based on the dual system IBE in [18]. In this section, we describe a BE scheme based on Scheme 6 in the Type-3 pairing setting. The security proof is given in the full version [13] and is based on the hardness of the DDH1, DDH2v and the DBDH problems. The new BE scheme provides adaptive security and is more efficient than previously known BE schemes providing adaptive security [9, 18]. In what follows, $n$ denotes the total number of users and $\{1, \ldots, n\}$, the set of users.

**Setup**$(n)$: Generators $P_1 \in_R \mathbb{G}_1$ and $P_2 \in_R \mathbb{G}_2$ are chosen. Also choose random elements $Q_{1,1}, \ldots Q_{1,n}, W_1 \in \mathbb{G}_1$ and $Q_{2,1}, \ldots, Q_{2,n}, W_2 \in \mathbb{G}_2$ such that $Q_{2,i} \sim Q_{1,i}$ for $1 \leq i \leq n$, $W_2 \sim W_1$. Let $\alpha, a, v, v'$ be random elements of $\mathbb{Z}_p$. Set $V_2 = vP_2$, $V_2' = v'P_2$ and $\tau = v + av'$ so that $\tau P_2 = V_2 + aV_2'$. The public key $\mathcal{PK}$ and secret key $\mathcal{SK}$ are given by

$$\begin{aligned} \mathcal{PK} &: (P_1, aP_1, \tau P_1, Q_{1,1}, \ldots, Q_{1,n}, W_1, e(P_1, P_2)^\alpha). \\ \mathcal{SK} &: (P_2, \alpha P_2, V_2, V_2', Q_{2,1}, \ldots, Q_{2,n}, W_2). \end{aligned}$$

**Encrypt**$(\mathcal{PK}, S \subseteq \{1, \ldots, n\}, M)$: Choose random $s$ from $\mathbb{Z}_p$; ciphertext $\mathcal{C}$ for the subset $S$ of users is $(C_0, C_1, C_2, C_3, E)$ where the elements are defined as follows.

$$\begin{aligned} &C_0 = M \cdot e(P_1, P_2)^{\alpha s}, \\ &C_1 = sP_1,\ C_2 = asP_1,\ C_3 = -\tau s P_1 + sW_1,\ E = s(\textstyle\sum_{i \in S} Q_{1,i}). \end{aligned}$$

**KeyGen**$(\mathcal{SK}, j \in \{1, \ldots, n\})$: Let $r$ be chosen at random from $\mathbb{Z}_p$; secret key for user $j$ is $\mathcal{SK}_j = (K_1, K_2, K_3, D, \forall_{i \neq j} D_i)$ where the elements are defined as follows.

$K_1 = \alpha P_2 + rV_2,\ K_2 = rV_2',\ K_3 = rP_2$
$D = r(Q_{2,j} + W_2),\ D_i = rQ_{2,i}$ for $i \neq j$.

**Decrypt** $(\mathcal{C}, S, \mathcal{SK}_j)$: Decryption works only if $j \in S$. Unmask the message as

$$M = \frac{C_0}{e(C_1, K_1 - D - \sum_{\substack{i \in S \\ i \neq j}} D_i) e(C_2, K_2) e(C_3 + E, K_3)}.$$

## 5 Security Proof for Scheme 6

First we define the semi-functional key and ciphertext for Scheme 6. As mentioned earlier, these are used only in the security reductions and are not part of the scheme itself.

*Semi-functional ciphertext:* Let $(C_0', C_1', C_2', C_3', E', \mathsf{ctag})$ be a normal ciphertext. Choose a random $\mu$ from $\mathbb{Z}_p$. The semi-functional ciphertext is given by $(C_0, C_1, C_2, C_3, E, \mathsf{ctag})$ where $C_0 = C_0'$, $C_1 = C_1'$, $C_2 = C_2' + \mu P_1$, $C_3 = C_3' - \mu V_1'$ and $E = E'$.

*Semi-functional key:* Let $(K_1', K_2', K_3', D, \mathsf{ktag})$ be a normal key. Choose a random $\gamma$ from $\mathbb{Z}_p$. The semi-functional key is $(K_1, K_2, K_3, D, \mathsf{ktag})$ where $K_1 = K_1' - a\gamma P_2$, $K_2 = K_2' + \gamma P_2$, $K_3 = K_3'$ and $D = D'$.

Let $\text{Game}_{real}$, $\text{Game}_k$ (for $0 \leq k \leq q$) and $\text{Game}_{final}$ be defined as in Section 3. Let Let $X_{real}$, $X_k$ and $X_{final}$ denote the events that the adversary wins in $\text{Game}_{real}$, $\text{Game}_k$ and $\text{Game}_{final}$ for $0 \leq k \leq q$ respectively.

**Lemma 1.** *If there exists an adversary $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\text{Game}_{real}} - \mathsf{Adv}^{\mathcal{A}}_{\text{Game}_0} = \varepsilon$, then we can build an algorithm $\mathcal{B}$ having advantage $\varepsilon$ in solving the* DDH1 *problem.*

*Proof.* The algorithm $\mathcal{B}$ receives $(P_1, sP_1, aP_1, P_2, Z_1)$ as an instance of DDH1. We describe how it will simulate each phase in the security game.
**Setup:** $\mathcal{B}$ chooses random elements $\alpha, y_v, y_v', y_q, y_w, y_u$ from $\mathbb{Z}_p$ and sets the parameters as follows: $P_1 = P_1$, $aP_1 = aP_1$, $Q_1 = y_qP_1$, $W_1 = y_wP_1$, $U_1 = y_uP_1$, $P_2 = P_2$, $V_2 = y_vP_2$, $V_2' = y_v'P_2$, $Q_2 = y_qP_2$, $W_2 = y_wP_2$, $U_2 = y_uP_2$. The element $\tau P_1$ is computed as $y_vP_1 + y_v'(aP_1)$ implicitly setting $\tau = y_v + ay_v'$. The simulator computes the remaining parameters using $\alpha$ and gives the following public parameters to $\mathcal{A}$: $\mathcal{PP} = (P_1, P_2, aP_1, \tau P_1, Q_1, W_1, U_1, e(P_1, P_2)^\alpha)$.
**Phase 1:** $\mathcal{A}$ makes a number of key extract queries. $\mathcal{B}$ knows the master secret and using that it returns a normal key for every key extract query made by $\mathcal{A}$.
**Challenge:** $\mathcal{B}$ receives the target identity $\mathsf{id}^*$ and two messages $M_0$ and $M_1$ from $\mathcal{A}$. It chooses $\beta \in \{0, 1\}$ at random. To encrypt $M_\beta$, $\mathcal{B}$ chooses $\mathsf{ctag}^*$ at random from $\mathbb{Z}_p$ and computes the ciphertext elements as follows: $C_0 = M_\beta \cdot e(sP_1, P_2)^\alpha$, $C_1 = sP_1$, $C_2 = Z_1$, $C_3 = -y_v(sP_1) - y_v'Z_1 + y_w(sP_1)$ and $E = (\mathsf{id}^* y_q + \mathsf{ctag}^* y_w + y_u)(sP_1)$. $\mathcal{B}$ returns $\mathcal{C}^* = (C_0, C_1, C_2, C_3, E, \mathsf{ctag}^*)$ to $\mathcal{A}$.

If $Z_1 = asP_1$ then the challenge ciphertext is normal; otherwise if $Z_1$ is a random element of $\mathbb{G}_1$ i.e., $Z_1 = (as+c)P_1$ then the ciphertext is semi-functional with $\mu = c$. Note that, to check whether $\mathcal{C}^*$ is semi-functional or not, $\mathcal{B}$ itself could try to decrypt it with a semi-functional key for $\mathsf{id}^*$. However since $aP_2$ is not known to $\mathcal{B}$, it cannot create such a key.

**Phase 2:** As in first phase, $\mathcal{B}$ returns a normal key for every query.

**Guess:** The adversary returns its guess $\beta'$ to $\mathcal{B}$.

If $\mathcal{C}^*$ is normal then $\mathcal{B}$ simulates $\text{Game}_{real}$ and if it is semi-functional $\mathcal{B}$ simulates $\text{Game}_0$. Therefore if $\mathcal{A}$ is able to distinguish between $\text{Game}_{real}$ and $\text{Game}_0$, then the $\mathcal{B}$ can solve the DDH1 problem with advantage

$$\mathsf{Adv}^{\mathcal{B}}_{\text{DDH1}} = |\Pr[X_{real}] - \Pr[X_0]| = \mathsf{Adv}^{\mathcal{A}}_{\text{Game}_{real}} - \mathsf{Adv}^{\mathcal{A}}_{\text{Game}_0} = \varepsilon.$$

□

**Lemma 2.** *If there exists an adversary $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\text{Game}_{k-1}} - \mathsf{Adv}^{\mathcal{A}}_{\text{Game}_k} = \varepsilon$, then we can build an algorithm $\mathcal{B}$ having advantage $\varepsilon$ in breaking the assumption* DDH2v.

*Proof.* Let $(P_1, dP_1, dzP_1, zx_1P_1, P_2, dP_2, x_1P_2, x_2P_2, Z_2)$ denote the instance of DDH2v that $\mathcal{B}$ receives.

**Setup:** $\mathcal{B}$ chooses random elements $a, \alpha, \lambda, \nu, y'_v, y_q, y_u, y_w \in_R \mathbb{Z}_p$ and sets the parameters as follows. $P_1 = P_1, P_2 = P_2, Q_2 = -\lambda(dP_2) + y_qP_2, U_2 = -\nu(dP_2) + y_uP_2, W_2 = dP_2 + y_wP_2, V_2 = -a(x_1P_2)$ and $V'_2 = x_1P_2 + y'_vP_2$ setting $\tau = ay'_v$ using which one can compute $\tau P_1 = ay'_vP_1$. The public parameters $Q_1, W_1, U_1$ can be computed since $\mathcal{B}$ has $dP_1$. The remaining parameters required to provide $\mathcal{PP}$ to $\mathcal{A}$ are computed using $a$, $\alpha$ and other elements of the problem instance.

**Phases 1 and 2:** The key extraction queries for identities $\mathsf{id}_1, \ldots, \mathsf{id}_q$ are answered in the following way. For $i < k$, a semi-functional key is returned and for $i > k$ a normal key is returned. Note that normal and semi-functional keys can be generated since $\mathcal{B}$ has the $\mathcal{MSK}$ and knows $a$. For $i = k$, a normal key $K'_1, K'_2, K'_3, D'$ is generated using randomiser $r' \in_R \mathbb{Z}_p$, $\mathsf{ktag} = \lambda\mathsf{id}_k + \nu$ and then modified as: $K_1 = K'_1 - aZ_2$, $K_2 = K'_2 + Z_2 + y'_v(x_2P_2)$, $K_3 = K'_3 + x_2P_2$ and $D = D + (y_q\mathsf{id} + y_w\mathsf{ktag} + y_u)(x_2P_2)$, thus implicitly setting $r = r' + x_2$. Since $dx_2P_2$ is not known to $\mathcal{B}$ it can create $D$ only when $\mathsf{ktag} = \lambda\mathsf{id}_k + \nu$. If $Z_2 = x_1x_2P_2$ then the key for $\mathsf{id}_k$ will be normal and otherwise it will be semi-functional with $\gamma = c$ where $Z_2 = (x_1x_2 + c)P_2$. Note that a semi-functional ciphertext for $\mathsf{id}_k$ with any value of $\mathsf{ctag}$ except for $\neq \lambda\mathsf{id}_k + \nu$ cannot be generated without the knowledge of $dx_1zP_1$ which is neither available from the assumption nor can be computed by $\mathcal{B}$. This rules out the obvious way of checking whether the key for $\mathsf{id}_k$ is semi-functional or not.

**Challenge:** $\mathcal{B}$ receives two messages $M_0, M_1$ and a challenge identity $\mathsf{id}^*$ during the challenge phase. It chooses $\beta \in_R \{0, 1\}$ and sets $\mathsf{ctag}^* = \lambda\mathsf{id}^* + \nu$. Since $\lambda$ and $\nu$ are chosen independently and uniformly at random, the function $\lambda X + \nu$ is a pairwise independent function for a variable $X$ over $\mathbb{Z}_p$. This causes the tag values of the challenge ciphertext and the $k$-th key to appear

properly distributed from the adversary's view. $\mathcal{B}$ computes the ciphertext elements as: $C_0 = e(zx_1P_1, P_2)^{\alpha}$, $C_1 = zx_1P_1$, $C_2 = a(zx_1P_1) + dzP_1$, $C_3 = (y_w - ay'_v)(zx_1P_1) - y'_v(dzP_1)$ and $E = (y_q\mathsf{id} + \mathsf{ctag}^* y_w + y_u)(zx_1P_1)$, setting $s = zx_1$ and $\mu = dz$. It is easy to check that $C_3$ is well-formed.

Now $\mathcal{A}$ will be able to distinguish between $\text{Game}_{k-1}$ and $\text{Game}_k$ if it can decide whether $\mathcal{SK}_{\mathsf{id}_k}$ is normal or semi-functional. In this case $\mathcal{B}$ can break the assumption DDH2v with advantage

$$\mathsf{Adv}^{\mathcal{B}}_{\text{DDH2v}} = |\Pr[X_{k-1}] - \Pr[X_k]| = \mathsf{Adv}^{\mathcal{A}}_{\text{Game}_{k-1}} - \mathsf{Adv}^{\mathcal{A}}_{\text{Game}_k} = \varepsilon.$$

□

**Lemma 3.** *If there exists an adversary $\mathcal{A}$ such that $\mathsf{Adv}^{\mathcal{A}}_{\text{Game}_q} - \mathsf{Adv}^{\mathcal{A}}_{final} = \varepsilon$, then we can build an algorithm $\mathcal{B}$ having advantage $\varepsilon$ in breaking the* DBDH *assumption.*

*Proof.* $\mathcal{B}$ receives $(P_1, xP_1, aP_1, sP_1, P_2, xP_2, aP_2, sP_2, Z)$ as an instance of the DBDH problem.

**Setup:** With $y_v, y'_v, y_q, y_w, y_u$ chosen at random from $\mathbb{Z}_p$, $\mathcal{B}$ sets the parameters as: $P_1 = P_1$, $P_2 = P_2$, $aP_1 = aP_1$, $V_2 = y_vP_2$, $V'_2 = y'_vP_2$, $\tau P_1 = y_vP_1 + y'_v(aP_1)$, $Q_1 = y_qP_1$, $W_1 = y_wP_1$, $U_1 = y_uP_1$, $e(P_1, P_2)^{\alpha} = e(xP_1, aP_2)$, thus implicitly setting $a = a$, $\alpha = xa$ and $\tau = y_v + ay'_v$. The remaining parameters can be computed easily. $\mathcal{B}$ returns $\mathcal{PP}$ to $\mathcal{A}$.

**Phases 1 and 2:** When $\mathcal{A}$ asks for the secret key for the $i$'th identity $\mathsf{id}_i$, $\mathcal{B}$ chooses at random $r, \mathsf{ktag}, \gamma' \in \mathbb{Z}_p$ implicitly setting $\gamma' = x - \gamma$. It then computes a semi-functional key for $\mathsf{id}_i$ as follows.

$$K_1 = \gamma'(aP_2) + rV_2 = xaP_2 - a\gamma P_2 + rV_2 = \alpha P_2 + rV_2 - a\gamma P_2$$

$$K_2 = rV'_2 - \gamma' P_2 + xP_2 = rV'_2 - xP_2 + \gamma P_2 + xP_2 = rV'_2 + \gamma P_2$$

$$K_3 = rP_2, D = r(\mathsf{id}_i Q_2 + \mathsf{ktag} W_2 + U_2).$$

Here $\mathcal{B}$ knows $\gamma'$ but not $\gamma$. Also, observe that $\mathcal{B}$ does not know $\alpha$ and hence cannot create a normal key.

**Challenge:** $\mathcal{B}$ receives the challenge identity $\mathsf{id}^*$ and two messages $M_0$ and $M_1$ from $\mathcal{A}$. It chooses $\beta \in \{0, 1\}$ and $\mathsf{ctag}^*, \mu' \in \mathbb{Z}_p$ at random and generates a semi-functional challenge ciphertext as follows. Here $\mathcal{B}$ implicitly sets $\mu' = \mu + as$ and it does not know $\mu$.

$$C_0 = M_\beta \cdot Z$$

$$C_1 = sP_1, C_2 = \mu' P_1 = asP_1 + \mu P_1$$

$$C_3 = -y_v(sP_1) - \mu' y'_v P_1 + y_w(sP_1) = -\tau sP_1 - \mu V'_1 + sW_1$$

$$E = (y_q\mathsf{id}^* + y_w\mathsf{ctag}^* + y_u)(sP_1)$$

The challenge ciphertext $\mathcal{C}^* = (C_0, C_1, C_2, C_3, E, \mathsf{ctag}^*)$ is returned to $\mathcal{A}$. If $Z = e(P_1, P_2)^{xas}$ then $\mathcal{C}^*$ will be a semi-functional encryption of $M_\beta$; if $Z$ is a random element of $\mathbb{G}_T$ then $\mathcal{C}^*$ will be a semi-functional encryption of a random

message. If $\mathcal{A}$ can identify whether the game simulated was $\text{Game}_q$ or $\text{Game}_{final}$, then $\mathcal{B}$ will be able to decide whether $Z = e(P_1, P_2)^{xas}$ or not and hence break the DBDH assumption with advantage

$$\mathsf{Adv}^{\mathcal{B}}_{\text{DBDH}} = |\Pr[X_q] - \Pr[X_{final}]| = \mathsf{Adv}^{\mathcal{A}}_{\text{Game}_q} - \mathsf{Adv}^{\mathcal{A}}_{\text{Game}_{final}} = \varepsilon.$$

□

**Theorem 1.** *If the* DDH1*,* DDH2v *and* DBDH *assumptions hold, then no polynomial time adversary $\mathcal{A}$ making at most $q$ key extraction queries can break the security of Scheme 6.*

*Proof.* Using lemmas 1, 2 and 3, we have for any polynomial time attacker $\mathcal{A}$,

$$\begin{aligned}\mathsf{Adv}^{\mathcal{A}}_{\text{Scheme 6}} &\leq |\Pr[X_{real}] - \Pr[X_0]| + \sum_{k=1}^{q} (|\Pr[X_{k-1}] - \Pr[X_k]|) \\ &\quad + |\Pr[X_q] - \Pr[X_{final}]| \\ &= \varepsilon_{\text{DDH1}} + q\varepsilon_{\text{DDH2v}} + \varepsilon_{\text{DBDH}}\end{aligned}$$

which is negligible in the security parameter $\kappa$. □

## 6 Conclusion

We have converted Waters dual system IBE scheme from the setting of symmetric pairings to that of asymmetric pairings. This has been done in a systematic manner going through several stages of simplifications. We have described in detail an IBE scheme, Scheme 6, which is quite simple and minimal in the sense that both encryption and key generation use one randomiser each. The security of Scheme 6 is based on two standard assumptions and a natural and minimal extension of the DDH assumption for $\mathbb{G}_2$. On the other hand, security of Scheme 1 is based on standard assumptions and reduces the sizes of ciphertexts and keys by 2 elements each from the original scheme of Waters.

## Acknowledgement

We would like to thank the anonymous reviewers of PKC 2012 for helpful comments and suggestions.

## References

1. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.

2. Dan Boneh and Xavier Boyen. Secure identity-based encryption without random oracles. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer, 2004.
3. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003. Earlier version appeared in the proceedings of CRYPTO 2001.
4. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In Joe Kilian, editor, *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2005.
5. Sanjit Chatterjee and Alfred Menezes. On cryptographic protocols employing asymmetric pairings – the role of $\psi$ revisited. *Discrete Applied Mathematics*, 159(13):1311–1322, 2011.
6. Sanjit Chatterjee and Palash Sarkar. Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model. In Dong Ho Won and Seungjoo Kim, editors, *ICISC*, volume 3935 of *Lecture Notes in Computer Science*, pages 424–440. Springer, 2005.
7. Clifford Cocks. An identity-based encryption scheme based on quadratic residues. In Bahram Honary, editor, *IMA Int. Conf.*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer, 2001.
8. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.
9. Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 171–188. Springer, 2009.
10. Allison Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. Cryptology ePrint Archive, Report 2011/490, 2011. `http://eprint.iacr.org/`.
11. Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 455–479. Springer, 2010.
12. David Naccache. Secure and practical identity-based encryption. *IET Information Security*, 1(2):59–64, 2007.
13. Somindu C. Ramanna, Sanjit Chatterjee, and Palash Sarkar. Variants of Waters' dual system primitives using asymmetric pairings. Cryptology ePrint Archive, Report 2012/024, 2012. `http://eprint.iacr.org/`.
14. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.
15. Elaine Shi and Brent Waters. Delegating capabilities in predicate encryption systems. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 560–578. Springer, 2008.
16. Nigel P. Smart and Frederik Vercauteren. On computable isomorphisms in efficient asymmetric pairing-based systems. *Discrete Applied Mathematics*, 155(4):538–547, 2007.
17. Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.
18. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, 2009.