(Augmented) Broadcast Encryption from Identity Based Encryption with Wildcard

Anaïs Barthoulot[1,2], Olivier Blazy[3], and Sébastien Canard[1]

[1] Orange Innovation, Caen, France
{anais.barthoulot,sebastien.canard}@orange.com
[2] Université de Limoges, XLim, France
[3] École Polytechnique, Palaiseau, France
olivier.blazy@polytechnique.edu

Abstract. Several broadcast encryption (BE) constructions have been proposed since Fiat and Naor introduced the concept, some achieving short parameters size while others achieve better security. Since 1994, a lot of alternatives to BE have moreover been additionally proposed, such as the broadcast and trace (BT) primitive which is a combination of broadcast encryption and traitor tracing. Among the other variants of BE, the notion of augmented BE (AugBE), introduced by Boneh and Waters in 2006, corresponds to a BE scheme with the particularity that the encryption algorithm takes an index as an additional parameter. If an AugBE scheme is both message and index hiding, it has been proved that it can generically be used to construct a secure BT scheme. Hence, any new result related to the former gives an improvement to the latter. In this paper, we first show that both BE and AugBE can be obtained by using an identity-based encryption scheme with wildcard (WIBE). We also introduce the new notion of anonymous AugBE, where the used users set is hidden, and prove that it implies index hiding. We then provide two different WIBE constructions. The first one has constant size ciphertext and used to construct a new constant size ciphertext BE scheme with adaptive CPA security, in the standard model (under the SXDH assumption). The second WIBE provides pattern-hiding, a new definition we introduced, and serves as a basis for the first anonymous AugBE scheme (and subsequently a BT scheme since our scheme is also index hiding by nature) in the literature, with adaptive security in the standard model (under the XDLin assumption).

Keywords: Broadcast Encryption · Augmented Broadcast Encryption · Identity Based Encryption with Wildcard · Broadcast and Trace · Pairings.

1 Introduction

Broadcast Encryption. Broadcast Encryption (BE), defined by Fiat and Naor [15], is a public key encryption scheme in which the encryption algorithm takes as input the public key pk, a message m, a subset $S \subseteq [N]$ of users (N being the number of users in the system), and such that the output ciphertext can be decrypted by any user in the subset S. Regarding related work, Boneh *et al.* ([8]) were the first to achieve constant size ciphertext (i.e., independent of the number of users in the set), but the security was only selective and proven in the generic group model. Recently, Agrawal *et al.* ([3]) achieves constant size parameters with a security proven in the standard model. But it is only selective secure and their scheme combines both pairings and lattices. Lastly, Gay *et al.* ([17]) proposes a scheme based on pairings with constant size ciphertext. As far as we know, this is the only BE scheme with a constant-size ciphertext and providing adaptive security in the standard model.

Augmented Broadcast Encryption. In 2006, Boneh and Waters [10] introduced *Augmented* Broadcast Encryption (AugBE), in which the encryption algorithm takes as additional input an index $\mathsf{ind} \in [N+1]$. As for any BE scheme, the output ciphertext can be decrypted by any user in the subset S, but it is additionally required that the user's index is greater or equal to ind. In particular, if $\mathsf{ind} = N+1$, no one can decrypt. Regarding security, an AugBE scheme should verify both some indistinguishability security (usually called message-hiding in this context), and some index-hiding security to protect the index. Both properties can be defined in a selective or in an adaptive way. The first AugBE constructions [10, 16] give a ciphertext's size in $O(\sqrt{N})$. In [18], using both pairings and lattices, Goyal *et al.* propose a selectively secure construction with ciphertext size in $O(N^\epsilon)$ $(0 < \epsilon \leq 1/2)$. Goyal *et al.* also propose in [19] a generic construction of an AugBE based on Positional Witness Encryption (PWE). Their scheme is the first one providing constant parameters. However, currently only few instantiations of PWE exist and all rely on multilinear maps.

Broadcast and Trace. Another variant of BE is the Broadcast and Trace (BT) primitive, i.e., the combination of BE and Traitor Tracing (a message is encrypted for the whole subset $[N]$ but if some subset of traitors uses their secret keys to produce a pirate decoder, then the tracing procedure can identify at least one of the traitors). In [10, 19], it was demonstrated that a BT scheme can be constructed from any message and index-hiding AugBE. As for traitor tracing, BT schemes can achieve either public (anyone can find the traitors) or private (traitors can only be retrieved by the owner of a specific master key) traceability, and both cases are indeed useful for different kinds of use cases. Theoretically speaking, public traceability is however known to be harder to achieve [9]. By construction, the Boneh-Waters AugBE definition [10] gives a publicly traceable BT scheme. Goyal *et al.*[18] have recently given another definition of AugBE that is suitable for the private case, where two encryption algorithms called Encrypt and Index-Encrypt need to be defined. Their resulting BT is based on pairings and lattices, has ciphertext in $O(N^\epsilon)$, for $0 < \epsilon \leq 1/2$, and is secretly traceable. In 2020, Zhandry [31] proposed a secretly traceable BT scheme based only on pairings, that has constant size ciphertext, but is only secure in the generic group model. To the best of our knowledge, it does not exist yet an efficient

AugBE/BT scheme which is adaptively secure in the standard model. In this paper, we only focus on the Boneh-Waters' AugBE definition, and thus public traceability.

Our contributions. In this paper, our idea is to use identity based encryption with wildcard (WIBE) [2] to construct BE schemes. In a WIBE scheme, private keys and ciphertexts are created for vectors of size L (called patterns) over a set $\mathcal{U}$, which contains a wildcard symbol "$\star$". A message encrypted for a pattern $\boldsymbol{P}$ can be decrypted by a secret key made for a pattern $\boldsymbol{P}^{'}$ such that $\boldsymbol{P}^{'}$ belongs to $\boldsymbol{P}$, i.e. if for all $i \in [L]$, $P_i = \star$ or $P_i = P^{'}_i$. More precisely, we provide two main contributions to broadcast encryption:

- we prove (Section 3.1) that WIBE can be used to construct BE schemes. Then, any new result on WIBE directly gives an improvement in the BE setting;
- we also prove (Section 3.2) that if WIBE satisfies some additional specific security property it can be used to construct AugBE schemes.

As a complement to those two results, we additionally the following minor contributions:

- we propose (Section 4) two new WIBE schemes, in the pairing setting and proven adaptively secure in the standard model. The first one has constant size ciphertext while the other achieves pattern hiding, the new property we introduced;
- our first WIBE construction gives a constant-size ciphertext BE scheme with adaptive security, proven in the standard model and using only pairings. Compared to the only existing equivalent construction [17], ours does not have constant size secret keys but has shorter public key (Table 1).
- our second WIBE construction gives us the first AugBE scheme, based on pairings, which is adaptively secure in the standard model. Using the generic transformation [10, 19], this gives us a BT scheme with similar characteristics. Compare to the state-of-the-art (Table 2), and especially Zhandry's BT scheme [31], we do not reach constant-size ciphertext, but we provide the harder publicly traceable property (while Zhandry's scheme is secretly traceable), and we prove the security of our construction in the standard model while Zhandry's is only proven secure in the generic group model.

Details on our generic constructions. We first remark that any subset $S^* \subseteq [N]$ can be represented as a pattern in $\boldsymbol{P} \in \{0, \star\}^N$, where for $j \in [\![1, N]\!]$, $P_j = \star$ if $j \in S^*$ and $P_j = 0$ otherwise. This fact can then be used to associate such pattern to the BE encryption set S. Additionally, any user identity $i \in [N]$ can be represented as a pattern $\boldsymbol{P}^i \in \{0, 1\}^N$ such that for $j \in [\![1, N]\!]$, $P^i_j = 1$ if $i = j$ and $P^i_j = 0$ otherwise. This finally gives us that $i \in S$ iff $\boldsymbol{P}^i$ belongs to $\boldsymbol{P}$. Regarding AugBE, we have noticed that the decrypting condition $i \geq \mathsf{ind}$ for any $i \in [N]$, $\mathsf{ind} \in [N+1]$ can be rewritten as $i \in \{\mathsf{ind}, \mathsf{ind}+1, \cdots, N+1\}$. It follows that the AugBE decrypting condition becomes $i \in S \cap \{\mathsf{ind}, \mathsf{ind}+1, \cdots, N+1\}$.

Then, we can associate encryption and key patterns as for the BE scheme to build our AugBE. From that, our generic constructions are then quite straightforward, and the security also comes directly, assuming that the used WIBE is indistinguishable. But in order to obtain AugBE security, we need a WIBE scheme that does not give information about the pattern used in encryption. For this purpose, we introduce such definition that we call *pattern-hiding*, and which may be of independent interest. We finally remark that using a pattern-hiding WIBE, we additionally freely obtain for the AugBE that the used user set is hidden into the ciphertext: this is the anonymity property, which has never been considered until now for AugBE. Saying that, it remains us to build such (pattern-hiding) WIBE.

Details on our WIBE constructions. We started from the paper of Kim *et al.* [21], who proposes a selectively secure WIBE scheme with constant size ciphertext. We have first adapted it to our keys and ciphertexts patterns, and using [20]'s idea to use composite order bilinear groups we obtained adaptive security. Afterwards, we have moved it from a symmetric bilinear group setting to an asymmetric prime order one, thanks to the combination of the work on Dual Pairing Vector Spaces by Lewko [22] with the one of Chen *et al.* [14] [4]. Our first scheme is then adaptively secure under the Symmetric External Diffie-Hellman assumption. We have then modified this first scheme to achieve the pattern-hiding property. Inspired by the work of [28] on attribute-hiding inner product encryption scheme, we obtain a new WIBE scheme that is adaptively pattern-hiding in the standard model, based on the External Decisional Linear Assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$. The idea is to use the orthogonality of dual pairing vector spaces as follow: let $\mathbb{B}, \mathbb{B}^*$ be two dual orthonormal bases with L elements in each. The secret key is computed using the elements of $\mathbb{B}^*$ corresponding to the positions where the associated pattern is equal to 1; the ciphertext is computed using the elements of $\mathbb{B}$ corresponding to the positions where the associated pattern is equal to 0. If the secret key pattern belongs to the ciphertext pattern, then the intersection of the two above sets is empty. Thus, thanks to the orthogonality property the elements in the key and in the ciphertext will cancel with each other. However, as we are using dual orthonormal bases of size L, each element of the bases has size L which results in a scheme with linear (in the number of user in the scheme) ciphertext and secret keys, and with quadratic public key (as we need to give the L elements of size L for encryption). Also notice that now the target set is no longer given as an additional parameter.

Broadcast encryption efficiency comparison. In Table 1, we give a comparison between our BE scheme (taking the case $L = N$ in the WIBE scheme of Section 4.1) and existing BE schemes.

[4] [5] proposed generic methods to transfer a composite order group scheme into a prime order group scheme via computational pair encodings. We do not used this method as the less general method of [22] and [14] is enough as we are considering simple predicates and encodings.

Table 1. Broadcast Encryption comparison; "GGM", "Sym" and "Asym" stand for "Generic Group Model", "Symmetric" and "Asymmetric" respectively. Here $t \in \mathbb{N}$, such that t divides N.

Scheme	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{sk}_i\rvert$	$\lvert\mathsf{ct}\rvert$	Security	Assumption	Model	Settings
Section 4.1	$O(N)$	$O(N)$	$O(1)$	Adaptive	SXDH	Standard	Asym pairings
[3]	$O(\lambda)$	$O(\lambda)$	$O(\lambda)$	Selective	LWE, KOALA	Standard	Lattices
[17]	$O(N^2)$	$O(1)$	$O(1)$	Adaptive	k−Lin	Standard	Asym pairings
[13]	$O(t+N/t)$	$O(N/t)$	$O(t)$	Adaptive	k−Lin	Standard	Asym pairings
[8]	$O(N)$	$O(1)$	$O(1)$	Selective	N-BDHE	GGM	Sym pairings

Our scheme is not as efficient as [3]'s scheme, which is currently the most efficient BE scheme in the literature. However, our scheme satisfies the stronger adaptive security notion, and is proven secure under standard assumption. Compare to the adaptively secure scheme given in [17], we have a bigger user secret key (sk_i) size, but a shorter public key (pk) size. To be exhaustive, [11] proposed a scheme with all parameters in $\mathsf{poly}(\log(N))$, with adaptive security. However, this scheme is using multilinear maps and its security is proven in the GGM. [12] proposed a scheme with all parameters in $\mathsf{poly}(n, \log(N))$ using lattices, but no security proof is given.

Augmented broadcast encryption and broadcast and trace efficiency comparison. Using our generic construction and our WIBE instantiation from section 4.2 with $L = N$ we obtain an instantiation of AugBE. The resulting scheme is the first proven adaptively secure in the standard model. Our scheme can itself be turned into a BT scheme, using the generic construction given in [10, 19]. Table 2 gives a comparison between our resulting BT and existing ones.

Table 2. Broadcast and Trace schemes comparison; tk, "p.o", "c.o", "PWE", "std" "Multi", "P" and "S" mean tracing key, "prime order" "composite order", "Positional Witness Encryption", "standard", "multilinear", "public" and "secret respectively, $0 < \epsilon \leq 1/2$.

Scheme	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{sk}_i\rvert$	$\lvert\mathsf{ct}\rvert$	Users set	Security	Model	tk	Object
Section 4.2	$O(N^2)$	$O(N)$	$O(N)$	×	Adaptive	Std	P	Pairings p.o.
[31]	$O(N)$	$O(N)$	$O(1)$	Given	Adaptive	GGM	S	Pairings p.o.
[18]	$\Omega(N)$	$\Omega(N^2)$	$O(N^\epsilon)$	Given	Selective	GGM	S	Pairing, lattices
[19]	$\mathsf{poly}(1^\lambda)$	$\mathsf{poly}(1^\lambda)$	$\mathsf{poly}(1^\lambda)$	Given	Adaptive	Multi	P	PWE
[10]	$O(\sqrt{N})$	$O(\sqrt{N})$	$O(\sqrt{N})$	Given	Adaptive	GGM	P	Pairing c.o.

As we can see our scheme is the first BT scheme (as far as we know) that does not need the description of the user sets to be able to decrypt, and that has security proven in the standard model. Moreover, our scheme is publicly traceable (known to be harder to achieve than private traceability), and uses pairings in prime order group while other existing publicly traceable schemes are using either pairings in composite order group (less secure), or positional witness encryption. Regarding efficiency, our resulting BT scheme has a complexity similar to a "trivial" scheme [29] (with all parameters sizes linear in the number of users). However, the claimed of our work is not to provide a new efficient Broadcast and Trace scheme, but a new generic way to build AugBE schemes, and our generic construction could be more efficient than a "trivial" BT scheme, even if our current instantiation is not. Moreover, we also consider that our proposal has the additional feature of anonymity, that the trivial construction could not have without being less efficient than ours. With such property, the users set is included in the ciphertext and no longer given in the clear which leads to a linear additional computational cost during decryption. Anonymity in the context of BT seems to be an overkill, but we think that for applications in which being in the used users set reveals some private information about users, it might be a real interest to use an anonymous scheme. Please refer to appendix D for more details about anonymity in the context of Broadcast (and Trace) Encryption. Hence, if our new instantiation of a BT scheme is not more efficient than the "trivial" scheme, it has some specific features that could not be obtained so easily "trivially".

Potential applications. In this work, we introduced an extension of WIBE security, pattern hiding security, in order to obtain the required security for the built AugBE. However, this new security may be of independent interest, in constructing fuzzy extractors for example or for access control encryption, as anonymity is an important required property in such schemes [30].

2 Preliminaries

Notations. Let "PPT" denotes "probabilistic polynomial-time" and unless specified, we consider that any PPT adversary $\mathcal{A}$ has output in $\{0,1\}$. For $a, b \in \mathbb{N}$ we denote $\{1, 2, \cdots, a\}$ as $[a]$, and $\{a, a+1, \cdots, b\}$ as $[\![a, b]\!]$. For every finite set S, $x \leftarrow S$ denotes a uniformly random element x from the set S. The security parameter of our schemes is denoted by 1^λ, where $\lambda \in \mathbb{N}$. Vectors are written with **bold face** lower case letters, patterns and matrices with **bold face** upper case letters. Regarding security definitions, we always present them in the adaptive way; the selective version can easily be derived. We also consider in this work only security against Chosen Plaintext Attacks (CPA). We also do not consider the *multi-challenge* setting ([17]) for BE. In each security definition, adversary is allowed to query at most $Q \in \mathbb{N}$ secret keys. For Broadcast Encryption and its variants, we have chosen to put the description of the target S as an input of the decryption algorithm. A consequence is that for any scheme (ours and the state-of-the-art ones), the size of S is not taken into account for the computation (and comparison) of the ciphertext's size, unless specified.

2.1 Broadcast Encryption

Definition 1. ***Broadcast Encryption*** *[15] [17]. A* broadcast encryption *scheme consists of algorithms (*$\mathsf{Setup}, \mathsf{KeyGen}$, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$*):*

- $\mathsf{Setup}(1^\lambda, 1^N) \to (\mathsf{pk}, \mathsf{msk})$. *This algorithm takes as input* 1^λ *and the number of users* 1^N*. It outputs the public parameters* pk *and the master secret key* msk*.*

- $\mathsf{KeyGen}(\mathsf{msk}, i) \to \mathsf{sk}_i$. *This algorithm gets as input the master secret key* msk *and an index* $i \in [N]$. *It outputs the secret key* sk_i *for user* i.
- $\mathsf{Encrypt}(\mathsf{pk}, S, \mathsf{m}) \to \mathsf{ct}_S$. *This algorithm gets as input* pk, *a message* m *and a subset* $S \subseteq [N]$. *It outputs a ciphertext* ct_S.
- $\mathsf{Decrypt}(\mathsf{pk}, S, i, \mathsf{sk}_i, \mathsf{ct}_S) \to \mathsf{m}$ *or* $\perp$. *This algorithm gets as input* $\mathsf{pk}, S, i, \mathsf{sk}_i$ *and* ct_S. *It outputs message* m *or reject symbol* $\perp$.

Definition 2. ***BE Correctness*** *[17]. Let (*$\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}$*) be a BE scheme. We require that for all* $S \subseteq [N]$, *messages* m, *and* $i \in [N]$ *for which* $i \in S$,

$$\Pr\left[ct_S \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S, \mathsf{m}), \mathsf{sk}_i \leftarrow \mathsf{KeyGen}(\mathsf{msk}, i) | \mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_i, ct_S) = \mathsf{m}\right] = 1$$

where the probability is taken over $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$ *and the coins of* $\mathsf{Encrypt}$.

Definition 3. ***Adaptive security*** *(IND-CPA-BE)[17]. A BE scheme is said* adaptively secure *(or satisfying* IND-CPA-BE *security) if all PPT adversaries* $\mathcal{A}$ *have at most negligible advantage in the game presented in Figure 1, where* $\mathcal{A}$*'s advantage is defined as* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IND\text{-}CPA\text{-}BE}}(\lambda) := \left|\Pr\left[b^{'} = b\right] - 1/2\right|$.

`SETUP`: challenger $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, 1^N)$ to generate pk and msk, and gives pk to $\mathcal{A}$.
`KEY QUERY`: $\mathcal{A}$ issues queries to $\mathcal{C}$ for index $i \in [N]$. $\mathcal{C}$ returns $\mathsf{sk}_i \leftarrow \mathsf{KeyGen}(\mathsf{msk}, i)$.
`CHALLENGE`: $\mathcal{A}$ selects messages $\mathsf{m}_0, \mathsf{m}_1$ and set $S^* \subseteq [N]$ of users. We require that $\mathcal{A}$ has not issued key queries for any $i \in S^*$. $\mathcal{A}$ passes $\mathsf{m}_0, \mathsf{m}_1$ and S^* to $\mathcal{C}$. The latter picks $b \in \{0, 1\}$ random and computes $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S^*, \mathsf{m}_b)$ which is returned to $\mathcal{A}$.
`KEY QUERY`: $\mathcal{A}$ makes queries for index $i \in [N]$ with the restriction that $i \notin S^*$.
`GUESS`: $\mathcal{A}$ outputs its guess $b^{'} \in \{0, 1\}$ for b, and wins the game if $b^{'} = b$.

Fig. 1. IND-CPA-BE security game.

2.2 Augmented Broadcast Encryption

Definition 4. ***Augmented Broadcast Encryption scheme*** *(AugBE) [10, 19] . An AugBE scheme is a tuple of algorithms (*Setup, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$*):*

- Setup $(1^\lambda, 1^N) \to (msk, pk, \{sk_1, \cdots, sk_N\})$. *This algorithm takes as input* 1^λ *and the number of users* N. *It outputs a master secret key* msk, *a public key* pk *and secret keys* $\{sk_1, \cdots, sk_N\}$, *where* sk_i *is the secret key for user* i.
- $\mathsf{Encrypt}$ $(pk,\ S, m, \mathsf{ind}) \to \mathsf{ct}$. *It takes as input the public key* pk, *a set of users* $S \subseteq [N]$, *a message* m, *an index* $\mathsf{ind} \in [N{+}1]$, *and outputs a ciphertext* ct.

- $\mathsf{Decrypt}$ *(pk, sk_i, S, ct) → m or ⊥. This algorithm takes as input the public key pk, the secret key for i^{th} user sk_i, a set of users $S \subseteq [N]$, a ciphertext ct and outputs a message m or reject symbol ⊥.*

Definition 5. ***AugBE Correctness*** *[19] . An AugBE scheme is said to be* correct *if for every security parameter $\lambda \in \mathbb{N}$, any number of users $N \in \mathbb{N}$, any message m, any subset of users $S \subseteq [N]$, any index* $\mathsf{ind} \in [N]$*, any $i \in S \cap \{ind, \cdots, N\}$, $(msk, pk, \{sk_1, \cdots, sk_N\}) \leftarrow \mathsf{Setup}(1^\lambda, 1^N)$ and $ct \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S, m, \mathsf{ind})$, we have:* $\mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_i, S, \mathsf{ct}) = \mathsf{m}$.

Definition 6. ***Message Hiding Security*** *[19]. An AugBE scheme satisfies* adaptive message hiding security *if for every stateful PPT adversary $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(.)$ *such that for every $\lambda \in \mathbb{N}$, the advantage of $\mathcal{A}$ to win the game presented in Figure 2 is lower or equal to* $1/2 + \mathsf{negl}(\lambda)$.

`SETUP`: challenger $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, 1^N)$ to obtain $\mathsf{msk}, \mathsf{pk}, \{\mathsf{sk}_i\}_{i \in [N]}$ and gives pk to $\mathcal{A}$.
`KEY QUERY`: $\mathcal{A}$ chooses an index $i \in [N]$ and sends it to $\mathcal{C}$, who responds with sk_i.
`CHALLENGE`: $\mathcal{A}$ chooses two messages $\mathsf{m}_0, \mathsf{m}_1$ and a challenge set S^* and sends it to $\mathcal{C}$. $\mathcal{C}$ chooses $b \in \{0, 1\}$, runs $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S^*, \mathsf{m}_b, N + 1)$ and gives ct^* to $\mathcal{A}$.
`KEY QUERY`: $\mathcal{A}$ chooses an index $i \in [N]$ and sends it to $\mathcal{C}$, who responds with sk_i.
`GUESS`: $\mathcal{A}$ outputs its guess $b^{'} \in \{0, 1\}$ for b, and wins the game if $b^{'} = b$.

Fig. 2. Adaptive message hiding security game.

Definition 7. ***Index Hiding Security*** *[19]. An AugBE scheme satisfies* adaptive index hiding security *if for every stateful PPT adversary $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(.)$ *such that for every $\lambda \in \mathbb{N}$, the advantage of $\mathcal{A}$ to win the game presented in Figure 3 is lower or equal to* $1/2 + \mathsf{negl}(\lambda)$.

`SETUP`: challenger $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, 1^N)$ to obtain $\mathsf{pk}, \mathsf{msk}, \{\mathsf{sk}_i\}_{i \in [N]}$ and gives pk to $\mathcal{A}$.
`KEY QUERY`: at each query, $\mathcal{A}$ chooses an index $i \in [N]$ and sends it to $\mathcal{C}$. $\mathcal{C}$ responds with sk_i. Let S be the set of indices for which a key is queried by $\mathcal{A}$.
`CHALLENGE`: $\mathcal{A}$ chooses a message m, a challenge set S^* and an index $\mathsf{ind} \in [N]$ and sends them to $\mathcal{C}$. If $\mathsf{ind} \in S \cap S^*$, $\mathcal{C}$ aborts. Otherwise, $\mathcal{C}$ chooses $b \in \{0, 1\}$, runs $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S^*, \mathsf{m}, \mathsf{ind} + b)$ and gives ct^* to $\mathcal{A}$.
`KEY QUERY`: at each query, $\mathcal{A}$ chooses an index $i \in [N]$ and sends it to $\mathcal{C}$ who adds i to S. If $\mathsf{ind} \in S \cap S^*$, $\mathcal{C}$ aborts. Otherwise $\mathcal{C}$ responds with sk_i.
`GUESS`: $\mathcal{A}$ outputs its guess $b^{'} \in \{0, 1\}$ for b, and wins the game if $b^{'} = b$.

Fig. 3. Adaptive index hiding security game.

Finally, we introduce a new security property for AugBE: **anonymity**. The below definition, close to the one for BE schemes ([26]), provides the adaptive version.

Definition 8. ***Anonymous*** *AugBE (*ANO-AUGE-BE*). We say that an AugBE scheme is adaptively* anonymous *if all adaptive PPT adversaries $\mathcal{A}$ have at most negligible advantage in the game presented in Figure 4, where $\mathcal{A}$'s advantage is defined as* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ano\text{-}augbe}}(\lambda) = \left|\Pr\left[b^{'} = b\right] - 1/2\right|$.

SETUP: challenger $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, 1^N)$ to obtain pk, msk, $\{\mathsf{sk}_i\}_{i\in[N]}$, and gives pk to $\mathcal{A}$.

KEY QUERY: $\mathcal{A}$ can issue queries to the challenger for index $i \in [N]$. $\mathcal{C}$ responds with sk_i.

CHALLENGE: $\mathcal{A}$ selects a message m, two distinct sets $S^0, S^1 \subseteq [N]$ of users and an index $\mathsf{ind} \in [N+1]$. We impose that $\mathcal{A}$ has not issued key queries for any $i \geq \mathsf{ind}$ such that $i \notin S^0 \cap S^1$. The adversary $\mathcal{A}$ passes $\mathsf{m}, S^0, S^1, \mathsf{ind}$ to $\mathcal{C}$. The latter picks a random bit $b \in \{0,1\}$ and computes $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S^b, \mathsf{m}, \mathsf{ind})$ which is returned to $\mathcal{A}$.

KEY QUERY: $\mathcal{A}$ makes queries for index $i \in [N]$ such that if $i \geq \mathsf{ind}$ then $i \in S^0 \cap S^1$.

GUESS: $\mathcal{A}$ outputs its guess $b^{'} \in \{0,1\}$ for b, and wins the game if $b^{'} = b$.

Fig. 4. ANO-AUGE-BE security game.

In the following theorem, we then prove that this new anonymity property is enough to obtain an index-hiding AugBE.

Theorem 1. *If an AugBE scheme is anonymous, then it is also index hiding.*

Proof. Let $\mathcal{C}$ be a challenger and $\mathcal{B}$ be an adversary that wins the index hiding security game with non negligible advantage. Informally, index hiding means that an adversary cannot distinguish between an encryption to index ind and one to index $\mathsf{ind}+1$ without the key $\mathsf{sk}_{\mathsf{ind}}$ and that an adversary cannot distinguish an encryption to index ind and one to index $\mathsf{ind}+1$ when ind is not in the target set S^* ([10]). Thus $\mathcal{B}$ can either distinguish which index was used in encryption when $\mathsf{ind} \in S^*$ and without knowing $\mathsf{sk}_{\mathsf{ind}}$, or he can distinguish the encryption index when $\mathsf{ind} \notin S^*$, knowing $\mathsf{sk}_{\mathsf{ind}}$. Therefore he either chooses $\mathsf{ind} \in S^*$ or $\mathsf{ind} \notin S^*$ but in this case he asks $\mathsf{sk}_{\mathsf{ind}}$ otherwise he would have advantage equal to 1/2. We construct, in Figure 5, an adversary $\mathcal{A}$ that wins the anonymous security game with non negligible advantage.

We have that if all $\mathcal{B}$'s queries satisfy the game constraints, then all $\mathcal{A}$'s queries have the same property. Thus $\mathcal{A}$'s simulation is perfect and the advantage of $\mathcal{A}$ is the same as $\mathcal{B}$'s. This concludes the proof.

Note 1. If $\mathsf{ind} \in S^*$, then $\mathsf{ind} \in S^0 \wedge \mathsf{ind} \notin S^1$ thus adversary cannot query $\mathsf{sk}_{\mathsf{ind}}$. If $\mathsf{ind} \notin S^*$, then $\mathsf{ind} \notin S^0 \wedge \mathsf{ind} \notin S^1$ thus adversary can query $\mathsf{sk}_{\mathsf{ind}}$.

SETUP: $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, 1^N) \to (\mathsf{msk}, \mathsf{pk}, \mathsf{sk}_1, \cdots, \mathsf{sk}_N)$, and sends pk to $\mathcal{A}$, and $\mathcal{B}$.

KEY QUERY: $\mathcal{B}$ chooses $i \in [N]$, sends it to $\mathcal{A}$ who sends it to $\mathcal{C}$. The later sends sk_i to $\mathcal{A}$ who sends it to $\mathcal{B}$.

CHALLENGE: $\mathcal{B}$ chooses a message m, a set S^* of users and an index $\mathsf{ind} \in [N]$ and sends $\mathsf{m}, S^*, \mathsf{ind}$ to $\mathcal{A}$. The latter creates the sets $S^0 = S^* \cap \{\mathsf{ind}, \cdots, N\}$ and $S^1 = S^* \cap \{\mathsf{ind} + 1, \cdots, N\}$. $\mathcal{A}$ sends m, S^0, S^1 to $\mathcal{C}$. If for any queried i, $i \in S^0 \wedge i \notin S^1$ then $\mathcal{C}$ aborts. Otherwise, it chooses $b \leftarrow \{0, 1\}$ and sets $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S^b, \mathsf{m}, 1)$. It sends ct^* to $\mathcal{A}$ who sends it to $\mathcal{B}$.

KEY QUERY: $\mathcal{A}$ and $\mathcal{B}$ act like in the previous KEY QUERY step. If $i \in S^0 \wedge i \notin S^1$, $\mathcal{C}$ aborts. Otherwise, it sends sk_i to $\mathcal{A}$ who sends it to $\mathcal{B}$.

GUESS: $\mathcal{B}$ outputs its guess $b^{'}$ to $\mathcal{A}$, who outputs it as its guess.

Fig. 5. Construction of ANO-AUGE-BE adversary from index hiding adversary.

Note 2. Index hiding does not imply anonymous. Indeed, in the index hiding security game, in the case where ind is not in the challenge, knowing the challenge set does not help determining if ind or $\mathsf{ind} + 1$ was used for encryption.

2.3 Identity-Based Encryption with Wildcard

Definition 9. *A pattern $\boldsymbol{P}$ is a vector $(P_1, \cdots, P_L) \in \mathcal{U}^L$, where $\mathcal{U}$ is a set with a special wildcard symbol "$\star$", and $L \in \mathbb{N}$. A pattern $\boldsymbol{P}^{'} = (P_1^{'}, \cdots, P_L^{'})$* belongs *to $\boldsymbol{P}$, denoted $\boldsymbol{P}^{'} \in_\star \boldsymbol{P}$, if and only if $\forall i \in \{1, \cdots, L\}$, $(P_i^{'} = P_i) \vee (P_i = \star)$. For a pattern $\boldsymbol{P} \in \mathcal{U}^L$, $W(\boldsymbol{P})$ denoted the set of all indices $i \in \{1, \cdots, L\}$ such that $P_i = \star$, and $\overline{W}(\boldsymbol{P})$ is the complementary set.*

Definition 10. ***Identity-based Encryption with Wildcard*** *(WIBE) [2, 21]. A WIBE scheme consists of four algorithms:*

- $\mathsf{Setup}(1^\lambda, 1^L)$*: the setup algorithm takes as input 1^λ and the pattern length $L \in \mathbb{N}$. It outputs a public key pk and a master secret key msk.*
- $\mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{P})$*: the key derivation algorithm takes as input msk and a pattern $\boldsymbol{P}$ and create a secret key $\mathsf{sk}_{\boldsymbol{P}}$ for $\boldsymbol{P}$. It can also take as input a secret key $\mathsf{sk}_{\boldsymbol{P}'}$ for a pattern $\boldsymbol{P}^{'}$ instead of msk and derive a secret key for any pattern $\boldsymbol{P} \in_\star \boldsymbol{P}^{'}$.*
- $\mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{P}, \mathsf{m})$*: this algorithm takes as input the public key pk, a pattern $\boldsymbol{P}$ and a message m. It outputs ciphertext ct for pattern $\boldsymbol{P}$.*
- $\mathsf{Decrypt}(\mathsf{sk}_{\boldsymbol{P}}, \mathsf{ct}, \boldsymbol{P}^{'})$*: the decryption algorithm takes as input a user secret key $\mathsf{sk}_{\boldsymbol{P}}$ for a pattern $\boldsymbol{P}$ and a ciphertext ct for a pattern $\boldsymbol{P}^{'}$. Any user in possession of the secret key for a pattern $\boldsymbol{P}$ that belongs to $\boldsymbol{P}^{'}$ can decrypt the ciphertext using $\mathsf{sk}_{\boldsymbol{P}}$, and the algorithm outputs message m.*

Definition 11. ***WIBE Correctness*** *[21]. Correctness requires that for all key pairs $(\mathsf{pk}, \mathsf{msk})$ output by* Setup*, all messages m, and all patterns $\boldsymbol{P}, \boldsymbol{P}^{'} \in \mathcal{U}^L$, such that $\boldsymbol{P}^{'} \in_\star \boldsymbol{P}$ then* $\mathsf{Decrypt}(\mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{P}^{'}), \mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{P}, \mathsf{m})) = \mathsf{m}$.

In the sequel we will only consider adaptive indistinguishability CPA security of WIBE (IND-WID-CPA). We introduce another security definition for WIBE: adaptive (resp. selective) pattern-hiding security. For lack of space we present both IND-WID-CPA and pattern-hiding security games in one.

Definition 12. ***Adaptive security****. The advantage of an adversary $\mathcal{A}$ in the game presented in Figure 6 is defined as $\mathsf{Adv}_{\mathcal{A}}^{WIBE}(\lambda) = \Pr[\mathcal{A}\ wins\ ] - 1/2$ for any $\lambda \in \mathbb{N}$. A WIBE scheme is* adaptively secure *if for all PPT adversaries $\mathcal{A}$, all $\lambda \in \mathbb{N}$, $\mathsf{Adv}_{\mathcal{A}}^{WIBE}(\lambda)$ is negligible. For each run of the game, we define a variable s as $s = 0$ if $m^0 \neq m^1$ and $\boldsymbol{P}^0 = \boldsymbol{P}^1 = \boldsymbol{P}^*$, and $s = 1$ if $m^0 = m^1 = m$ and $P^0 \neq P^1$. The case $s = 0$ corresponds to* IND-WID-CPA *security, and the case $s = 1$ corresponds to pattern-hiding security. Let $\mathcal{C}$ be a challenger.*

SETUP: challenger $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, 1^L)$ to get keys pk and msk, and pk is given to $\mathcal{A}$.

KEY QUERY: $\mathcal{A}$ may adaptively query a key for pattern $\boldsymbol{P}$. In response, $\mathcal{A}$ is given the corresponding secret key $\mathsf{sk}_{\boldsymbol{P}} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{P})$.

CHALLENGE: $\mathcal{A}$ outputs challenge patterns $\boldsymbol{P}^0, \boldsymbol{P}^1$ and challenge messages $\mathsf{m}_0, \mathsf{m}_1$, subject to the following restrictions:

- if $s = 0$, $\boldsymbol{P} \notin_\star \boldsymbol{P}^*$ for all the key queried pattern $\boldsymbol{P}$.
- if $s = 1$, any key query $\boldsymbol{P}$ verifies one of the following conditions:
  - $\boldsymbol{P} \in_\star \boldsymbol{P}^0 \wedge \boldsymbol{P} \in_\star \boldsymbol{P}^1$
  - $\boldsymbol{P} \notin_\star \boldsymbol{P}^0 \wedge \boldsymbol{P} \notin_\star \boldsymbol{P}^1$

A random bit b is chosen. $\mathcal{A}$ is given $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{P}^b, \mathsf{m}^b)$.

KEY QUERY: The adversary may continue to issue key queries for additional pattern $\boldsymbol{P}$, subject to the restrictions given above. $\mathcal{A}$ is given the corresponding key $\mathsf{sk}_{\boldsymbol{P}} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{P})$.

GUESS: $\mathcal{A}$ outputs a bit $b^{'}$, and wins if $b^{'} = b$.

Fig. 6. Adaptive security game.

Note 3. [1] introduced ***anonymous*** *WIBE, but the difference with our notion of* ***pattern-hiding*** *is that in anonymous security game the adversary can only query keys that do not decrypt the challenge ciphertext. In our definition, adversary can query keys that decrypt the challenge ciphertext for both challenge patterns.*

Note 4. Also notice that if a WIBE is pattern-hiding, then the decryption algorithm does no longer take as input the pattern associated to the ciphertext.

2.4 Other Definitions

Definition 13. ***Asymmetric bilinear pairing groups*** *[14].* Asymmetric bilinear groups *$\Gamma = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ are tuple of prime p, cyclic (multiplicative) groups $\mathbb{G}_1, \mathbb{G}_2$, $\mathbb{G}_T$ of order p, $g_1 \neq 1 \in \mathbb{G}_1$, $g_2 \neq 1 \in \mathbb{G}_2$, and a* *polynomial-time computable non-degenerate bilinear pairing* $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, *i.e.* $e(g_1^s, g_2^t) = e(g_1, g_2)^{st}$ *and* $e(g_1, g_2) \neq 1$.

Note 5. For any group element $g \in \mathbb{G}$, and any vector $\boldsymbol{v}$ of size $l \in \mathbb{N}$, we denote by $g^{\boldsymbol{v}}$ the vector $(g^{v_1}, \cdots, g^{v_l})$. Let $\boldsymbol{u}, \boldsymbol{v}$ be two vectors of length L. Then by $g^{\boldsymbol{u} \cdot \boldsymbol{v}}$, we denote the element g^α, where $\alpha = \boldsymbol{u} \cdot \boldsymbol{v} = u_1 \cdot v_1 + u_2 \cdot v_2 + \cdots + u_L \cdot v_L$.

Definition 14. ***Dual pairing vector spaces*** *(DPVS) [14]. For a prime p and a fixed (constant) dimension n, we choose two random bases* $\mathbb{B} = (\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n)$ *and* $\mathbb{B}^* = (\boldsymbol{b}_1^*, \cdots, \boldsymbol{b}_n^*)$ *of* $\mathbb{Z}_p^n$*, subject to the constraint that they are* ***dual orthonormal****, meaning that* $\boldsymbol{b}_i \cdot \boldsymbol{b}_j^* = 0 \pmod{p}$ *whenever* $i \neq j$*, and* $\boldsymbol{b}_i \cdot \boldsymbol{b}_i^* = \psi \pmod{p}$ *for all i, where* ψ *is a uniformly random element of* $\mathbb{Z}_p$*. Here the elements of* $\mathbb{B}, \mathbb{B}^*$ *are vectors and* $\cdot$ *corresponds to the scalar product. We denote such algorithm as* $\mathsf{Dual}(\mathbb{Z}_p^n)$*. For generators* $g_1 \in \mathbb{G}_1$ *and* $g_2 \in \mathbb{G}_2$*, we note that* $e(g_1^{\boldsymbol{b}_i}, g_2^{\boldsymbol{b}_j^*}) = 1$ *whenever* $i \neq j$*.*

Definition 15. ***Symmetric External Diffie-Hellman*** *(SXDH) [14].The SXDH assumption holds if DDH problems are intractable in both* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*.*

Definition 16. ***eXternal Decision Linear 1 Assumption*** *(XDLin1) [7]. Let* $\mathbb{G}_1, \mathbb{G}_2$ *be cyclic groups of prime order, with generators* (g_1, g_2)*, and* $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ *be a bilinear map. The XDLin1 assumption states that given a tuple* $(g_1, g_1^x, g_1^y, g_1^{ax}, g_1^{by}, g_2, g_2^x, g_2^y, g_2^{ax}, g_2^{by}, g_1^c)$ *it is hard to decide if* $c = a + b$ *or not, for random* $a, b, x, y \in \mathbb{Z}_p$*.*

The **eXternal Decision Linear 2 Assumption** (XDLin2) is defined similarly, except that the last element of the tuple is equal to g_2^c, where c either equals $a + b$, or is random.

3 Generic Construction of AugBE from WIBE

This section presents two generic broadcast encryption constructions from identity based encryption with wildcard: one for a basic BE scheme and the other for an AugBE scheme. It also formalizes which properties of WIBE are needed in order to obtain a secure BE (resp. AugBE). For sake of simplicity, we admit in proofs that the number of keys queried is always lower or equal to the maximal number Q of keys that an adversary is allowed to query. All proofs are done for adaptive security definitions and can be adapted to the selective case. The length of patterns is $L \in \mathbb{N}$.

3.1 Broadcast Encryption from WIBE

Let $\mathcal{WIBE} = (w.\mathsf{Setup}, w.\mathsf{KeyDer}, w.\mathsf{Encrypt}, w.\mathsf{Decrypt})$ be an identity based encryption with wildcard scheme for key pattern space $\{0,1\}^L \backslash \{\mathbf{0}^L\}$ and ciphertext pattern space $\{0,\star\}^L \backslash \{\mathbf{0}^L\}$. Let $N \in \mathbb{N}$ be the number of users in the scheme. We construct a BE scheme BE $= (\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{Decrypt})$ in Figure 7.

- $\mathsf{Setup}(1^\lambda, 1^N)$: set $L = N$, run $w.\mathsf{Setup}(1^\lambda, 1^N)$ and set $\mathsf{pk} = w.\mathsf{pk}$ and $\mathsf{msk} = w.\mathsf{msk}$.
- $\mathsf{KeyGen}(\mathsf{msk}, i \in [N])$: define $\boldsymbol{P}^{'} \in \{0,1\}^N$ such that for $j \in [\![1, N]\!]$, $P^{'}_j = 0$ if $j \neq i$ and $P^{'}_j = 1$ if $i = j$. Then set $\mathsf{sk}_i = w.\mathsf{KeyDer}(w.\mathsf{msk}, \boldsymbol{P}^{'})$. It outputs sk_i.
- $\mathsf{Encrypt}(\mathsf{pk}, S, \mathsf{m})$: first, associate S with a pattern $\boldsymbol{P}$ in $\{0, \star\}^N$ such that for $j \in [\![1, N]\!]$, $P_j = \star$ if $j \in S$ and $P_j = 0$ otherwise. Finally compute $\mathsf{ct} = w.\mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{P}, \mathsf{m})$ and outputs ct.
- $\mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_i, \mathsf{ct}, S)$: gets $\mathsf{m} \leftarrow w.\mathsf{Decrypt}(\mathsf{sk}_i, \boldsymbol{P}, \mathsf{ct})$ if $i \in S$, $\perp$ otherwise.

Fig. 7. Generic construction of BE from WIBE.

Note 6. Encryption for pattern $\mathbf{0}^L$ is not relevant here as it means that no one can decrypt, that is why we excluded this pattern of encryption pattern space. Secret key for pattern $\mathbf{0}^L$ is not relevant either as it corresponds to none of the users.

Theorem 2. *The BE scheme obtained is correct if the underlying WIBE is correct.*

Proof. $\boldsymbol{P}^i \in_\star \boldsymbol{P}$ implies that $P^i_i = P_i$ or $P^i = \star$. As $P^i_i = 1$, we have that $P^i = \star$ and thus $i \notin \overline{W}(\boldsymbol{P})$, i.e. $i \in S$. Suppose that $i \in S$. By construction for all $j \in [N], j \neq i$, $P^i_j = 0$ and either $P^i_j = 0 = P_j$ or $P_j = \star$, and $P_i = \star$, i.e. $\boldsymbol{P}^i \in_\star \boldsymbol{P}$. Then correctness follows from WIBE's correctness.

Theorem 3. *If $\mathcal{WIBE}$ satisfies adaptive (resp selective)* $\mathsf{IND\text{-}WID\text{-}CPA}$ *security, then the obtained BE scheme satisfies adaptive (resp selective)* $\mathsf{IND\text{-}CPA\text{-}BE}$ *security.*

Proof. Let $\mathcal{B}$ be an adversary against $\mathsf{IND\text{-}CPA\text{-}BE}$ security, that wins with non negligible advantage. In Figure 8 we construct $\mathcal{A}$, an adversary against $\mathsf{IND\text{-}WID\text{-}CPA}$ that uses $\mathcal{B}$ and wins with non negligible advantage. Let $\mathcal{C}$ be a challenger.

If all $\mathcal{B}$'s queries satisfy the game constraints, then all $\mathcal{A}$'s queries have the same property. Thus, $\mathcal{A}$'s simulation is perfect and the advantage of $\mathcal{A}$ is the same as $\mathcal{B}$'s.

3.2 Augmented Broadcast Encryption from WIBE

Let $\mathcal{WIBE} = (w.\mathsf{Setup}, w.\mathsf{KeyDer}, w.\mathsf{Encrypt},\ w.\mathsf{Decrypt})$ be an identity based encryption with wildcard scheme for key pattern space $\{0,1\}^L \backslash \{\mathbf{0}^L\}$ and ciphertext pattern space $\{0,\star\}^L$. Let $N \in \mathbb{N}$ be the number of users in the scheme. We now construct an AugBE scheme AugBE $= (\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{Decrypt})$ in Figure 9.

Note 7. Here encryption for pattern $\mathbf{0}^L$ corresponds to encryption for index $N + 1$.

SETUP: $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, 1^N) \to (\mathsf{msk}, \mathsf{pk})$ and gives pk to $\mathcal{A}$, who gives it to $\mathcal{B}$.

KEY QUERY: $\mathcal{B}$ chooses an index $i \in [N]$ and sends it to $\mathcal{A}$, who creates $\boldsymbol{P}^i$, for $j \in [\![1, N]\!]$, such that $P^i_j = 1$ if $i = j$ and $P^i_j = 0$ otherwise. $\mathcal{A}$ sends $\boldsymbol{P}^i$ to $\mathcal{C}$. The latter runs $\mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{P}^i) \to \mathsf{sk}_{\boldsymbol{P}^i}$ and sends $sk_{\boldsymbol{P}^i}$ to $\mathcal{A}$, who sends it as sk_i to $\mathcal{B}$.

CHALLENGE: $\mathcal{B}$ chooses $\mathsf{m}_0, \mathsf{m}_1$ and a set S^*; it sends it to $\mathcal{A}$ who creates the pattern $\boldsymbol{P}^*$, for $j \in [\![1, N]\!]$ s.t. $\boldsymbol{P}^*_j = 0$ if $j \notin S^*$, $P^*_j = \star$ otherwise, and sends $\boldsymbol{P}^*, \mathsf{m}_0, \mathsf{m}_1$ to $\mathcal{C}$. If for any queried $\boldsymbol{P}^i$, $\boldsymbol{P}^i \in_\star \boldsymbol{P}^*$ then $\mathcal{C}$ aborts. Otherwise it chooses $b \in \{0, 1\}$ and runs $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{P}^*, \mathsf{m}_b)$. It sends ct^* to $\mathcal{A}$ who sends it to $\mathcal{B}$.

KEY QUERY: $\mathcal{B}$ chooses index $i \in [N]$, sends it to $\mathcal{A}$, who creates $\boldsymbol{P}^i$, for $j \in [\![1, N]\!]$, s.t. $P^i_j = 1$ if $i = j$ and $P^i_j = 0$ otherwise. $\mathcal{A}$ sends $\boldsymbol{P}^i$ to $\mathcal{C}$. If $\boldsymbol{P}^i \in_\star \boldsymbol{P}^*$, aborts. Otherwise $\mathcal{C}$ runs $\mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{P}^i) \to \mathsf{sk}_{\boldsymbol{P}^i}$ and sends $sk_{\boldsymbol{P}^i}$ to $\mathcal{A}$, who sends it as sk_i to $\mathcal{B}$.

GUESS: $\mathcal{B}$ outputs a bit $b^{'}$ to $\mathcal{A}$ who outputs it as its guess.

Fig. 8. Construction of IND-WID-CPA adversary from IND-CPA-BE adversary.

- $\mathsf{Setup}(1^\lambda, 1^N)$: set $L = N$, and run $w.\mathsf{Setup}(1^\lambda, 1^N)$ to obtain $w.\mathsf{pk}, w.\mathsf{msk}$. Then for each $i \in [N]$, define $\boldsymbol{P}^{'} \in \{0, 1\}^N$ such that for $j \in [\![1, N]\!]$, $P^{'}_j = 0$ if $j \neq i$ and $P^{'}_j = 1$ if $i = j$. Then set $\mathsf{sk}_i = w.\mathsf{KeyDer}(w.\mathsf{msk}, \boldsymbol{P}^{'})$, $(\mathsf{pk}, \mathsf{msk}) = (w.\mathsf{pk}, w.\mathsf{msk})$. It outputs msk, pk and $\{\mathsf{sk}_i\}_{i \in [N]}$.
- $\mathsf{Encrypt}(\mathsf{pk}, S, \mathsf{ind}, \mathsf{m})$: here $\mathsf{ind} \in [N + 1]$. Associate S with a pattern $\boldsymbol{P}^*$ in $\{0, \star\}^N$ such that for $j \in [\![1, N]\!]$, $P^*_j = \star$ if $j \in S$ and $P^*_j = 0$ otherwise. Then define the pattern $\boldsymbol{P}^{\mathsf{ind}} \in \{0, \star\}^N$ such that for $j \in [\![1, N]\!]$, $P^{\mathsf{ind}}_j = 0$ if $j < \mathsf{ind}$ and $P^{\mathsf{ind}}_j = \star$ otherwise. Finally, define $\boldsymbol{P} \in \{0, \star\}^N$ such that for $j \in [\![1, N]\!]$, $P_j = P^*_j \wedge P^{\mathsf{ind}}_j$ with the following rule : $\star \wedge 0 = 0$. Finally compute $\mathsf{ct} = w.\mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{P}, \mathsf{m})$ and outputs ct.
- $\mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_i, \mathsf{ct})$: compute $\mathsf{m} \leftarrow w.\mathsf{Decrypt}(\mathsf{sk}_i, \mathsf{ct})$ if $i \in S \wedge i \geq ind$, $\perp$ otherwise.

Fig. 9. Generic construction of AugBE from WIBE.

Note 8. As the underlying WIBE is pattern-hiding, the AugBE decryption algorithm does not take as input the set for which the message was encrypted.

Theorem 4. *The AugBE scheme obtained is correct if the underlying WIBE is correct.*

Proof. $\boldsymbol{P}^i \in_\star \boldsymbol{P}$ implies that $P^i_i = P_i$ or $P_i = \star$. As $P^i_i = 1$, we have that $P^i = \star$ and thus $i \notin \overline{W}(\boldsymbol{P})$, i.e. $i \in S \wedge i \geq \mathsf{ind}$. Suppose that $i \in S \wedge i \geq \mathsf{ind}$. By construction for all $j \in [N], j \neq i$, $P^i_j = 0$ and either $P^i_j = 0 = P_j$ or $P_j = \star$, and $P_i = \star$, i.e. $\boldsymbol{P}^i \in_\star \boldsymbol{P}$. Then correctness follows from WIBE's correctness.

Theorem 5. *If WIBE satisfies adaptive (resp. selective)* IND-WID-CPA *security, then the obtained AugBE scheme satisfies adaptive (resp. selective) message hiding security.*

Proof. Let $\mathcal{B}$ be an adversary against message hiding security, that wins with non negligible advantage. In Figure 10 we construct $\mathcal{A}$ an adversary against $\mathsf{IND\text{-}WID\text{-}CPA}$ that uses $\mathcal{B}$ and wins with non negligible advantage. Let $\mathcal{C}$ be a challenger.

SETUP: $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, 1^N) \rightarrow (\mathsf{msk}, \mathsf{pk})$ and sends pk to $\mathcal{A}$, who sends it to $\mathcal{B}$.

KEY QUERY: $\mathcal{B}$ chooses $i \in [N]$, sends it to $\mathcal{A}$ who creates the pattern $\boldsymbol{P}^i$ such that for $j \in [\![1, N]\!]$, $P_j^i = 1$ if $i = j$, $P_j^i = 0$ otherwise. $\mathcal{A}$ sends $\boldsymbol{P}^i$ to $\mathcal{C}$, who responds with $\mathsf{sk}_{\boldsymbol{P}^i} \leftarrow \mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{P}^i)$. $\mathcal{A}$ sends $\mathsf{sk}_{\boldsymbol{P}^i}$ to $\mathcal{B}$ as sk_i.

CHALLENGE: $\mathcal{B}$ chooses messages $\mathsf{m}_0, \mathsf{m}_1$ and a set S^*. It sends $\mathsf{m}_0, \mathsf{m}_1, S^*$ to $\mathcal{A}$, who creates pattern $\boldsymbol{P}^*$, such that for $j \in [\![1, N]\!]$, $P_j^* = 0$. $\mathcal{A}$ sends $\mathsf{m}_0, \mathsf{m}_1, P^*$ to $\mathcal{C}$, who chooses $b \leftarrow \{0, 1\}$ and runs $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{P}^*, \mathsf{m}_b)$. $\mathcal{C}$ gives ct^* to $\mathcal{A}$, who sends it to $\mathcal{B}$.

KEY QUERY: $\mathcal{A}, \mathcal{B}, \mathcal{C}$ act like in the previous KEY QUERY step.

GUESS: $\mathcal{B}$ outputs its guess $b^{'}$ to $\mathcal{A}$, who outputs it as its guess.

Fig. 10. Construction of $\mathsf{IND\text{-}WID\text{-}CPA}$ adversary from message hiding adversary.

If all $\mathcal{B}$'s queries satisfy the game constraints, then all $\mathcal{A}$'s queries have the same property. Thus $\mathcal{A}$'s simulation is perfect and the advantage of $\mathcal{A}$ is the same as $\mathcal{B}$'s. This concludes the proof.

Note 9. Pattern $\boldsymbol{P}^*$ is equal to $\mathbf{0}^N$. Then, for all $i \in [N]$, $\boldsymbol{P}^i \notin_\star \boldsymbol{P}^*$: the WIBE adversary's constraint is always verified and we do not specify it in the proof.

Theorem 6. *If WIBE satisfies adaptive (resp. selective) pattern-hiding security, then the obtained AugBE scheme satisfies adaptive (resp. selective) anonymous security.*

Proof. Let $\mathcal{C}$ be a challenger and $\mathcal{B}$ be an adversary that wins the anonymous security game with non negligible advantage. We construct, in Figure 11, an adversary $\mathcal{A}$ that uses $\mathcal{B}$ and wins the pattern-hiding security game with non negligible advantage.

If all $\mathcal{B}$'s queries satisfy the game constraint, then all $\mathcal{A}$'s queries have the same property. Thus $\mathcal{A}$'s simulation is perfect, and the advantage of $\mathcal{A}$ is the same as $\mathcal{B}$'s. This concludes the proof.

Combining theorem 1 and 6 we obtain that if WIBE satisfies adaptive (resp. selective) pattern-hiding security then the AugBE scheme obtained from the WIBE satisfies adaptive (resp. selective) index hiding security.

4 Instantiations of WIBE

In this section, we first present a WIBE that has constant-size ciphertext but does not provide the pattern-hiding property, then a second scheme which does not have constant-size ciphertext but is proved to be pattern-hiding. Both do

`SETUP`: $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, 1^N) \to (\mathsf{msk}, \mathsf{pk})$ and sends pk to $\mathcal{A}$, who sends it to $\mathcal{B}$.

`KEY QUERY`: $\mathcal{B}$ chooses $i \in [N]$, sends it to $\mathcal{A}$ who creates the pattern $\boldsymbol{P}^i$ such that for $j \in [\![1, N]\!]$, $P^i_j = 1$ if $i = j$, $P^i_j = 0$ otherwise. $\mathcal{A}$ sends $\boldsymbol{P}^i$ to $\mathcal{C}$, who responds with $\mathsf{sk}_{\boldsymbol{P}^i} \leftarrow \mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{P}^i)$. $\mathcal{A}$ sends $\mathsf{sk}_{\boldsymbol{P}^i}$ to $\mathcal{B}$ as sk_i.

`CHALLENGE`: $\mathcal{B}$ chooses a message m, two sets S^0, S^1 and sends m, S^0, S^1 to $\mathcal{A}$. The latter creates the patterns $\boldsymbol{P}^0$, $\boldsymbol{P}^1$ such that for $j \in [\![1, N]\!]$, $P^0_j = \star$ if $j \in S^0$, $P^0_j = 0$ otherwise, and $P^1_j = \star$ if $j \in S^1$, $P^1_j = 0$ otherwise. $\mathcal{A}$ sends $\mathsf{m}, \boldsymbol{P}^0, \boldsymbol{P}^1$ to $\mathcal{C}$. If for any queried $\boldsymbol{P}^i$, $\boldsymbol{P}^i \in_\star \boldsymbol{P}^0 \wedge \boldsymbol{P}^i \notin_\star \boldsymbol{P}^1$ or $\boldsymbol{P}^i \notin_\star \boldsymbol{P}^0 \wedge \boldsymbol{P}^i \in_\star \boldsymbol{P}^1$, $\mathcal{C}$ aborts. Otherwise, it chooses $b \leftarrow \{0, 1\}$ and sets $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{P}^b, \mathsf{m})$. It sends ct^* to $\mathcal{A}$ who sends it to $\mathcal{B}$.

`KEY QUERY`: $\mathcal{A}$ and $\mathcal{B}$ act like in the previous `KEY QUERY` step. If $\boldsymbol{P}^i \in_\star \boldsymbol{P}^0 \wedge \boldsymbol{P}^i \notin \boldsymbol{P}^1$ or $\boldsymbol{P}^i \notin_\star \boldsymbol{P}^0 \wedge \boldsymbol{P}^i \in \boldsymbol{P}^1$, $\mathcal{C}$ aborts. Otherwise, it runs $\mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{P}^i) \to \mathsf{sk}_{\boldsymbol{P}^i}$ and sends $\mathsf{sk}_{\boldsymbol{P}^i}$ to $\mathcal{A}$ who sends it as sk_i to $\mathcal{B}$.

`GUESS`: $\mathcal{B}$ outputs its guess $b^{'}$ to $\mathcal{A}$, who outputs it as its guess.

Fig. 11. Construction of pattern-hiding adversary from AugBE anonymous adversary.

not allow key derivation for a pattern from another pattern's key (thus KeyDer algorithm will be written KeyGen). As in the previous section, both schemes have key pattern space equal to $\{0,1\}^L \backslash \{\mathbf{0}^L\}$, and ciphertext pattern space is equal to $\{0,\star\}^L \backslash \{\mathbf{0}^L\}$ for the first scheme and to $\{0,\star\}^L \backslash \{\star^L\}$ for our second scheme. Let $\boldsymbol{P}^{'} \in \{0,1\}^L \backslash \{\mathbf{0}^L\}$ and $\boldsymbol{P} \in \{0,\star\}^L$ be patterns. We define $\mathcal{I} = \left\{ i \in [L] | P^{'}_i = 1 \right\}$ and $\mathcal{O} = \left\{ i \in [L] | P^{'}_i = 0 \right\}$; notice that $[L] = \mathcal{I} \cup \mathcal{O}$. Also notice that $\boldsymbol{P}^{'} \in_\star \boldsymbol{P} \Longrightarrow \forall i \in [L]$, if $P^{'} = 1$ then $P_i = \star$ and thus $\mathcal{I} \subseteq W(\boldsymbol{P})$.

4.1 WIBE with Constant Size Ciphertext

We start by our first WIBE scheme (Figure 12), which has a constant-size ciphertext, and can be used to instantiate our BE scheme given in the previous section.

- $\mathsf{Setup}(1^\lambda, 1^L)$: generate an asymmetric bilinear pairing group $\varGamma = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ for sufficiently large prime order p. Sample random dual orthonormal bases $(\mathbb{D}, \mathbb{D}^*) \leftarrow \mathsf{Dual}(\mathbb{Z}_p^4)$. Let $\boldsymbol{d}_1, \cdots, \boldsymbol{d}_4$ denote the elements of $\mathbb{D}$ and $\boldsymbol{d}^*_1, \cdots, \boldsymbol{d}^*_4$ denote the elements of $\mathbb{D}^*$. Pick $\alpha, a_1, \cdots, a_L \leftarrow \mathbb{Z}_p$. The public key is computed as: $\mathsf{pk} = (\varGamma, p, e(g_1, g_2)^{\alpha \boldsymbol{d}_1 \cdot \boldsymbol{d}^*_1}, g_1^{\boldsymbol{d}_1}, \boldsymbol{h}_1 = g_1^{a_1 \cdot \boldsymbol{d}_2}, \cdots, \boldsymbol{h}_L = g_1^{a_L \cdot \boldsymbol{d}_2})$ and the master secret key is $\mathsf{msk} = (\alpha, g_2^{\boldsymbol{d}^*_1}, g_2^{\boldsymbol{d}^*_2}, a_1, \cdots, a_L)$.
- $\mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{P}^{'})$: pick $r \leftarrow \mathbb{Z}_p$. Compute $\boldsymbol{a} = g_2^{\alpha \boldsymbol{d}^*_1 + r \cdot \sum_{i \in \mathcal{I}} a_i \cdot \boldsymbol{d}^*_1 - r \cdot \boldsymbol{d}^*_2}$ and $\boldsymbol{b}_i = g_2^{r \cdot a_i \cdot \boldsymbol{d}^*_1}$ for $i \in \mathcal{O}$. The secret key is $\mathsf{sk}_{\boldsymbol{P}'} = (\boldsymbol{a}, \{\boldsymbol{b}_i\}_{i \in \mathcal{O}})$.
- $\mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{P}, \mathsf{m} \in \mathbb{G}_T)$: choose $s \leftarrow \mathbb{Z}_p$ and compute $\mathsf{ct} = (c_1, \boldsymbol{c}_2)$ where $c_1 = \mathsf{m} \cdot (e(g_1, g_2)^{\alpha \boldsymbol{d}^*_1 \cdot \boldsymbol{d}_1})^s$, $\boldsymbol{c}_2 = g_1^{s \boldsymbol{d}_1} \cdot \prod_{i \in W(\boldsymbol{P})} \boldsymbol{h}_i^s$.
- $\mathsf{Decrypt}(\mathsf{sk}_{\boldsymbol{P}'}, \boldsymbol{P}, \mathsf{ct})$: compute $\boldsymbol{a}^{'} = \boldsymbol{a} \prod_{i \in W(\boldsymbol{P}) \cap \mathcal{O}} \boldsymbol{b}_i$ and finally $\boldsymbol{C}_1 \cdot \frac{1}{e(\boldsymbol{c}_2, \boldsymbol{a}^{'})}$.

Fig. 12. An adaptive WIBE in prime order group, with constant size ciphertext.

Theorem 7. *Our first WIBE scheme is correct.*

Proof.

$$e(\boldsymbol{c}_2, \boldsymbol{a}^{'}) = e\left(g_1^{s\boldsymbol{d}_1} \cdot \prod_{i\in W(\boldsymbol{P})} \boldsymbol{h}_i^s, g_2^{\alpha\boldsymbol{d}_1^*+r.\sum_{i\in\mathcal{I}} a_i.\boldsymbol{d}_1^*-r.\boldsymbol{d}_2^*} \cdot \prod_{i\in W(\boldsymbol{P})\cap\mathcal{O}} g_2^{r\boldsymbol{d}_1^* a_i}\right)$$
$$= e\left(g_1^{s\boldsymbol{d}_1}, g_2^{\alpha\boldsymbol{d}_1^*}\right).e\left(g_1^{s\boldsymbol{d}_1}, g_2^{r.\boldsymbol{d}_1^*(\sum_{i\in\mathcal{I}} a_i+\sum_{i\in W(\boldsymbol{P})\cap\mathcal{O}} a_i)}\right).e\left(g_1^{s\boldsymbol{d}_2\sum_{i\in W(\boldsymbol{P})} a_i}, g_2^{-r.\boldsymbol{d}_2^*}\right)$$

As thanks to dual vector spaces properties: $e\left(g_1^{s\boldsymbol{d}_1}, g_2^{-r\boldsymbol{d}_2^*}\right) = e(g_1, g_2)^0$ and $e\left(g_1^{s\boldsymbol{d}_2 \sum\limits_{i\in W(\boldsymbol{P})} a_i}, g_2^{\alpha\boldsymbol{d}_1^*+r.\sum\limits_{i\in\mathcal{I}} a_i.\boldsymbol{d}_1^*+\sum_{i\in W(P)\cap\mathcal{O}} r.a_i.\boldsymbol{d}_1^*}\right) = e(g_1, g_2)^0 = 1$. The first pairing is equal to $(e(g_1, g_2)^{\alpha\boldsymbol{d}_1.\boldsymbol{d}_1^*})^s$ which will canceled with the element of c_1. The second pairing is equal to $e(g_1, g_2)^{sr\psi(\sum_{i\in\mathcal{I}} a_i+\sum_{i\in W(\boldsymbol{P})\cap\mathcal{O}} a_i)}$ and the third pairing is equal to $e(g_1, g_2)^{-rs\psi\sum_{i\in W(\boldsymbol{P})} a_i}$.
As user is allowed to decrypt then $\mathcal{I} \subseteq W(\boldsymbol{P})$, thus we can rewrite $\mathcal{I}$ as $\mathcal{I}\cap W(\boldsymbol{P})$ and we have that $\sum_{i\in\mathcal{I}} a_i + \sum_{i\in W(\boldsymbol{P})\cap\mathcal{O}} a_i = \sum_{i\in W(\boldsymbol{P})\cap(\mathcal{I}\cup\mathcal{O})} a_i = \sum_{i\in W(\boldsymbol{P})} a_i$. Therefore multiplying the two last pairings gives 1, and user can decrypt.

Theorem 8. *If SXDH holds then our scheme satisfies adaptive* IND-WID-CPA*.*

Our proof is based on the ones of [22] (Section 4.6) and [14] (Section 4) and is using **dual system encryption** ([24]). We introduce a second form of keys and ciphertexts: semi functional keys and semi functional ciphertexts. Let $\mathsf{sk} = (\boldsymbol{a}, \{\boldsymbol{b}_i\}_{i\in\mathcal{O}})$ be a normal key, and $t_3, t_4, \{t_{b,i}\}_{i\in\mathcal{O}}$ be random elements of $\mathbb{Z}_p$. We define a semi functional key as $\mathsf{sk}^{'} = (\boldsymbol{a}^{'}, \left\{\boldsymbol{b}_i^{'}\right\}_{i\in\mathcal{O}})$ where $\boldsymbol{a}^{'} = \boldsymbol{a} \cdot g_2^{t_3\cdot\boldsymbol{d}_3^*+t_4\cdot\boldsymbol{d}_4^*}$ and $\boldsymbol{b}_i^{'} = \boldsymbol{b}_i \cdot g_2^{t_{b,i}\cdot\boldsymbol{d}_3^*}$ for $i \in \mathcal{O}$.

Let $\mathsf{ct} = (c_1, \boldsymbol{c}_2)$ be a normal ciphertext, and $z_3, z_4 \leftarrow \mathbb{Z}_p$. We define a semi functional ciphertext as $\mathsf{ct}^{'} = (c_1^{'}, \boldsymbol{c}_2^{'})$ where $c_1^{'} = c_1$ and $\boldsymbol{c}_2^{'} = \boldsymbol{c}_2 \cdot g_1^{z_3\cdot\boldsymbol{d}_3+z_4\cdot\boldsymbol{d}_4}$.

We are going to prove Theorem 8 with a sequence of $Q + 3$ hybrids games.

- Game_0: is the real IND-WID-CPA security game (Definition 12 for $s = 0$).
- Game_1: is as Game_0 except that the challenge ciphertext is semi-functional.
- Game_{2-j}: for j from 1 to Q, Game_{2-j} is the same as Game_1 except that the first j keys are semi-functional and the remaining keys are normal.
- Game_3: is the same as Game_{2-Q}, except that the challenge ciphertext is a semi-function encryption of a random message in $\mathbb{G}_T$.

For lack of space, we only give an overview of the proofs of indistinguishability between these games (refer to Annex B for the full proofs). Moving from symmetric pairings to asymmetric pairings is not an issue if elements are taken in the correct group ($\mathbb{G}_1$ for ciphertext and public key elements, and $\mathbb{G}_2$ for secret keys elements). The proofs are using assumptions called DS1 and DS2, presented in Annex A. Here is the idea of how to prove indistinguishability between theses games.

- If an adversary can distinguish Game_0 from Game_1 then we can build an adversary with non-negligible advantage against DS1 with $k = 2$ and $n = 4$.
- If an adversary can distinguish $\mathsf{Game}_{2-(j-1)}$ from Game_{2-j} then we can build an adversary with non-negligible advantage against DS2 with $k = 2$ and $n = 4$.
- If an adversary can distinguish Game_{2-Q} from Game_3 then we can build an adversary with non-negligible advantage against DS1 with $k = 1$ and $n = 4$. We prove this in two steps, by randomizing each appearance of s in the $\boldsymbol{c}_2$ term of the ciphertext, thereby severing its link with the blinding factor. The end result is a semi-functional encryption of a random message. As a first step, we consider an intermediary game, called $\mathsf{Game}_{2-Q'}$, that is exactly like Game_{2-Q}, except that in the $\boldsymbol{c}_2$ term of the challenge ciphertext the coefficient of $\boldsymbol{d}_2$ is changed from being $s\sum_{i \in W(P)} a_i$ to a fresh random value in $\mathbb{Z}_p$. Then we prove that
  - If an adversary can distinguish Game_{2-Q} from $\mathsf{Game}_{2-Q'}$ then we can build an adversary with non-negligible advantage against DS1 with $k = 1$ and $n = 4$.
  - If an adversary can distinguish $\mathsf{Game}_{2-Q'}$ from Game_3 then we can build an adversary with non-negligible advantage against DS2 with $k = 1$ and $n = 4$

4.2 Pattern Hiding WIBE

We describe our scheme (Figure 13), which can be used to obtain an instantiation of our AugBE scheme given in Section 3.

- $\mathsf{Setup}(1^\lambda, 1^L)$: generate an asymmetric bilinear pairing group $\mathit{\Gamma} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ for sufficiently large prime order p. Sample random dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*) \leftarrow \mathsf{Dual}(\mathbb{Z}_p^{4L+2})$. Let $\boldsymbol{b}_0, \cdots, \boldsymbol{b}_{4L+1}$ denote the elements of $\mathbb{B}$. Pick $\alpha \leftarrow \mathbb{Z}_p$. The public key is computed as: $\mathsf{pk} = (\mathit{\Gamma}, p, e(g_1, g_2)^{\alpha \boldsymbol{b}_0 \cdot \boldsymbol{b}_0^*}, g_1^{\boldsymbol{b}_0}, g_1^{\boldsymbol{b}_{4L+1}}, \boldsymbol{h}_1 = g_1^{\boldsymbol{b}_1}, \cdots, \boldsymbol{h}_L = g_1^{\boldsymbol{b}_L})$ and the master secret key is $\mathsf{msk} = (\alpha, g_2^{\boldsymbol{b}_0^*}, g_2^{\boldsymbol{b}_1^*}, \cdots g_2^{\boldsymbol{b}_L^*}, g_2^{\boldsymbol{b}_{3L+1}^*} \cdots, g_2^{\boldsymbol{b}_{4L}^*})$.
- $\mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{P}^{'})$: pick $\boldsymbol{r}, \boldsymbol{\eta} \in \mathbb{Z}_p^L$. The secret key is $\mathsf{sk}_{\boldsymbol{P}'} = g_2^{\alpha \boldsymbol{b}_0^* + \sum_{j \in \mathcal{I}} r_j \boldsymbol{b}_j^* + \sum_{l=1}^{L} \eta_l \cdot \boldsymbol{b}_{3L+l}^*}$.
- $\mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{P}, \mathsf{m} \in \mathbb{G}_T)$: choose $s_1, s_2, s_3 \leftarrow \mathbb{Z}_p$ and compute $\mathsf{ct} = (c_1, \boldsymbol{c}_2)$ where $c_1 = \mathsf{m} \cdot (e(g_1, g_2)^{\alpha b_0^* \cdot b_0})^{s_1}$, $\boldsymbol{c}_2 = g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1}} \cdot \prod_{i \in \bar{W}(\boldsymbol{P})} \boldsymbol{h}_i^{s_3}$.
- $\mathsf{Decrypt}(\mathsf{sk}_{\boldsymbol{P}'}, \mathsf{ct})$: compute $c_1 \cdot \frac{1}{e(\boldsymbol{c}_2, \mathsf{sk}_{\boldsymbol{P}'})}$.

Fig. 13. An adaptive WIBE in prime order group, satisfying pattern-hiding.

Theorem 9. *Our WIBE scheme is correct.*

Proof.

$$\begin{aligned} e(\boldsymbol{c}_2, \mathsf{sk}_{\boldsymbol{P}'}) &= e\left(g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1}} \cdot \prod_{i \in \bar{W}(\boldsymbol{P})} \boldsymbol{h}_i^{s_3}, g_2^{\alpha \boldsymbol{b}_0^* + \sum_{j\in\mathcal{I}} r_j \boldsymbol{b}_j^* + \sum_{l=1}^{L} \eta_l \cdot \boldsymbol{b}^* 3L+l} \right) \\ &= e\left(g_1^{s_1 \boldsymbol{b}_0}, g_2^{\alpha \boldsymbol{b}_0^*} \right) \cdot e\left(g_1^{\sum_{i\in\bar{W}(\boldsymbol{P})} \boldsymbol{b}_i \cdot s_3}, g_2^{\sum_{j\in\mathcal{I}} r_j \boldsymbol{b}_j^*} \right) \end{aligned}$$

The last row is obtained thanks to dual vector spaces properties. The first pairing cancels itself with the pairing in c_1. Now, let's see the value of $\sum_{i\in\bar{W}(\boldsymbol{P})} \boldsymbol{b}_i \cdot \sum_{j\in\mathcal{I}} \boldsymbol{b}_j^*$. Suppose that user with pattern $\boldsymbol{P}^{'}$ is allowed to decrypt. Then $\boldsymbol{P}^{'} \in_\star \boldsymbol{P}$, that means that $\mathcal{I} \subseteq W(\boldsymbol{P})$. Thus $\mathcal{I} \cap \bar{W}(\boldsymbol{P}) = \varnothing$, and thanks to dual vector spaces properties, the above product is equal to 0 and decryptor obtains m.

Theorem 10. *If XDLin*1*, XDLin*2 *hold, then our scheme is adaptively pattern-hiding secure, in the standard model.*

Our proof is inspired by the one of [28] (Section 4.3) for their IPE scheme: the security is proven throughout a series of games. We start with the two following games.

- Game_0 is the original game given in the WIBE security definition (Definition 12).
- $\mathsf{Game}_{0'}$ is the same as Game_0 except that a coin $t \in \{0,1\}$ is chosen before setup, and the game is aborted in challenge step if $t \neq s$.

First, we execute a preliminary game transformation from Game_0 to $\mathsf{Game}_{0'}$. We define that adversary $\mathcal{A}$ wins with probability $1/2$ when the game is aborted (and the advantage in $\mathsf{Game}_{0'}$ is $\Pr[\mathcal{A} \text{ wins}] - 1/2$ as well). Since t is independent from s, the game is aborted with probability $1/2$. Hence, the advantage in $\mathsf{Game}_{0'}$ is a half of that in Game_0, i.e., $\mathsf{Adv}_{\mathcal{A}}^{0'}(\lambda) = 1/2 \cdot \mathsf{Adv}_{\mathcal{A}}^{0}(\lambda)$. Moreover, $\Pr[\mathcal{A} \text{ wins}] = 1/2 \cdot (\Pr[\mathcal{A} \text{ wins} | t = 0] + \Pr[\mathcal{A} \text{ wins} | t = 1])$ in $\mathsf{Game}_{0'}$ since t is uniformly and independently generated.

For lack of space, we only present the idea of the security proofs when $t = 0$ and $t = 1$. For the full proofs, refer to Annex B.

IND-WID-CPA security ($\boldsymbol{t = 0}$). This proof is similar to the one of [23] (Section 3.5.2); it uses a series of $Q+2$ games:

- Game_1: is the same as $\mathsf{Game}_{0'}$ except that the challenge ciphertext $(c_1, \boldsymbol{c}_2)$ for challenge plaintexts $(\mathsf{m}_0, \mathsf{m}_1)$ and challenge pattern $\boldsymbol{P}^*$ is changed into **temporal 1** form: $s_1, s_2, s_3, t_1, \cdots, t_L \leftarrow \mathbb{Z}_p, b \leftarrow \{0,1\}$, and requires that $P_1^* \neq \star$,

$$c_1 = \mathsf{m}_b \cdot e(g_1, g_2)^{\alpha \boldsymbol{b}_0 \cdot \boldsymbol{b}_0^* s_1}, \;\; \boldsymbol{c}_2 = g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + s_3 \sum_{i\in\bar{W}(\boldsymbol{P}^*)} \boldsymbol{b}_i + \sum_{l=1}^{L} t_l \boldsymbol{b}_{L+l}} \tag{1}$$

- Game_{2-k} ($k \in [\![1, Q]\!]$): is the same as $\mathsf{Game}_{2-(k-1)}$ (for $k = 1$, $\mathsf{Game}_{2-(k-1)}$ is Game_1) except that the reply to the k-th key queried for $\boldsymbol{P}$ is changed into **temporal 1** form: $\alpha, \{r_j\}_{j\in\mathcal{I}}, \{\eta_i, x_i\}_{i\in[\![1,L]\!]} \leftarrow \mathbb{Z}_p$,

$$\boldsymbol{sk_P} = g_2^{\alpha \boldsymbol{b}_0^* + \sum_{j\in\mathcal{I}} r_j \boldsymbol{b}_j^* + \sum_{l=1}^{L} x_l \boldsymbol{b}_{L+l}^* + \sum_{l=1}^{L} \eta_l \boldsymbol{b}_{3L+l}^*} \tag{2}$$

- Game_3: is the same as Game_{2-Q} except that the challenge ciphertext $(c_1, \boldsymbol{c}_2)$ for challenge plaintexts $(\mathsf{m}_0, \mathsf{m}_1)$ and challenge pattern $\boldsymbol{P}^*$ is changed into **unbiased** form: $s_1^{'}, \{\tilde{s}_i\}_{i\in[\![1,L]\!]} \leftarrow \mathbb{Z}_p$

$$c_1 = \mathsf{m}_b \cdot e(g_1, g_2)^{\alpha \boldsymbol{b}_0 \cdot \boldsymbol{b}_0^* s_1}, \ \ \boldsymbol{c}_2 = g_1^{s_1^{'} \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + \sum_{i=1}^{L} \tilde{s}_i \boldsymbol{b}_i + \sum_{l=1}^{L} t_l \boldsymbol{b}_{L+l}} \tag{3}$$

  and all the other variables are generated as in Game_{2-Q}.

Indistinguishability is proven using intermediate problems (defined in Annex A) that hold if $\mathsf{XDLin1}, \mathsf{XDLin2}$ hold. If an adversary can distinguish $\mathsf{Game}_{0'}$ from Game_1 then an adversary against Problem 1 bis (Definition 20) can be created. Then, they build an adversary against Problem 2 bis (Definition 22) using an adversary that distinguishes $\mathsf{Game}_{2-(k-1)}$ from Game_{2-k}. Finally they proved that the advantage of an adversary in winning Game_{2-Q} is the same than the one of an adversary winning Game_3; and the latter is equal to 0. The original proofs are made in the symmetric pairing settings but they can easily be made in the asymmetric setting by taking elements in the correct group.

Pattern hiding security ($\boldsymbol{t = 1}$). The proof is done as in [28] (Section 4.3.3), except that it is turned into the asymmetric setting (easily when considering elements in the correct group). It uses a sequence of $4Q+2$ games using different forms of ciphertexts and keys that we introduce. The different forms of ciphertext are defined according to challenge patterns $\boldsymbol{P}^0, \boldsymbol{P}^1$. c_1 is the same in all forms, just $\boldsymbol{c}_2$ is different:

- Game_1: is as $\mathsf{Game}_{0'}$ except that the ciphertext is changed to **temporal 0** form: let $b \in \{0,1\}, t \in \mathbb{Z}_p$ and suppose that $P_1^b = 0$. Define $\boldsymbol{c}_2$ as

$$g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + s_3 \sum\limits_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i + t \boldsymbol{b}_{L+1}} \tag{4}$$

- For $1 \leq h \leq Q$ (the number of keys queried), we define the following 4 games:
  - Game_{2-h-1}: in this game, the challenge ciphertext is changed to **temporal 1** form: let $b \in \{0,1\}, t, u, \tilde{u} \in \mathbb{Z}_p$. Define $\boldsymbol{c}_2$ as g_1 with exponent

$$\begin{gathered} s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + s_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i + t \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_{L+i} + u \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} \\ + \tilde{u} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i} \end{gathered} \tag{5}$$

and the first $h-1$ keys are **temporal 2** forms: let $\boldsymbol{x} \in \mathbb{Z}_p^L$ be a random vector. Define the key as

$$g_2^{\alpha \boldsymbol{b}_0^* + \sum_{j\in\mathcal{I}} r_j \boldsymbol{b}_j^* + \sum_{j\in\mathcal{I}} x_j \boldsymbol{b}_{2L+j}^* + \sum_{l=1}^{L} \eta_l \cdot \boldsymbol{b}_{3L+l}^*} \tag{6}$$

while the remaining keys are normal.

  - Game_{2-h-2}: in this game the h-th key is changed to **temporal 1** form: let $\boldsymbol{z} \in \mathbb{Z}_p^L$ be a random vector. Define the key as

    $$g_2^{\alpha \boldsymbol{b}_0^* + \sum_{j\in\mathcal{I}} r_j \boldsymbol{b}_j^* + \sum_{j\in\mathcal{I}} z_j \boldsymbol{b}_{L+j}^* + \sum_{l=1}^{L} \eta_l \cdot \boldsymbol{b}_{3L+l}^*} \tag{7}$$

    while the remaining keys and the challenge ciphertext are the same as in Game_{2-h-1}.
  - Game_{2-h-3}: in this game, challenge ciphertext is changed to **temporal 2** form: let $b \in \{0,1\}, t, \tilde{t}, u, \tilde{u} \in \mathbb{Z}_p$. Define $\boldsymbol{c}_2$ as g_1 with exponent

    $$\begin{aligned} s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + s_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i + t \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{L+i} \\ + \tilde{t} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{L+i} + u \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i} \end{aligned} \tag{8}$$

    while all the queried keys are the same as in Game_{2-h-2}.
  - Game_{2-h-4}: in this game, the h-th key is changed to **temporal 2** form (eq. 6) while the remaining keys and the challenge ciphertext are as in Game_{2-h-3}.

- Game_3: the challenge ciphertext is changed to **unbiased form**: let $b \in \{0,1\}, w, \tilde{w}, t, \tilde{t}, u, \tilde{u} \in \mathbb{Z}_p$. Define $\boldsymbol{c}_2$ as g_1 with exponent

  $$\begin{aligned} s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + w \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_i + \tilde{w} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_i + t \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{L+i} \\ + \tilde{t} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{L+i} + u \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i} \end{aligned} \tag{9}$$

  while all the queried keys are **temporal 2** form (eq. 6). In this game, the advantage of adversary is 0.

Indistinguishability between games is proven as in the original proof, using intermediate problems (defined in Annex A) that hold if $\mathsf{XDLin1}, \mathsf{XDLin2}$ hold:

- If there exists an adversary that can distinguish $\mathsf{Game}_{0'}$ from Game_1 then there exists an adversary that breaks Problem 1 (Definition 19).
- $\mathsf{Game}_{2-(h-1)-4}$ can conceptually be changed into Game_{2-h-1}. The advantage of an adversary in distinguishing theses games is equal to $4/p$ when $h = 1$, otherwise it is equal to $3/p$.
- If there exists an adversary that can distinguish Game_{2-h-1} from Game_{2-h-2} then there exists an adversary that breaks Problem 2 (Definition 21).
- If there exists an adversary that can distinguish Game_{2-h-3} from Game_{2-h-4} then there exists an adversary that breaks Problem 3 (Definition 23).

- Game_{2-Q-4} can conceptually be changed into Game_3. The advantage of an adversary in distinguishing theses games is equal to $3/p$.

The only part of the original proof that cannot be done for our scheme is the one that proves the indistinguishability of Game_{2-h-2} and Game_{2-h-3}. Indeed, [28] proved that Game_{2-h-2} can be conceptually changed to Game_{2-h-3} with a change of bases and an intermediate game. However, with their change of bases $\mathbb{B}, \mathbb{B}^*$ to $\mathbb{D}, \mathbb{D}^*$, the h-th key of our scheme can no longer decrypt the ciphertext. Thus, the adversary can distinguish the different games as in one case the h-th key decrypts the challenge ciphertext but not in the other case. That is because, with the definition of $\mathbb{D}, \mathbb{D}^*$, some elements of $\mathbb{B}$ (resp. $\mathbb{B}^*$) are now linear combination of elements of $\mathbb{D}$ (resp. $\mathbb{D}^*$). Thus, the set $\bar{W}(\boldsymbol{P}^b) \cap \mathcal{I}$ is no longer equal to $\varnothing$ (the decryption condition) but is equal to $\bar{W}(\boldsymbol{P}^b)$. In our proof, we change the way the new dual orthonormal bases are computed. We define new dual orthonormal bases $(\mathbb{D}, \mathbb{D}^*)$, following the idea of the last lemma in the original proof. Let $\theta_i, \tau_i \leftarrow \mathbb{Z}_p$ and for $i \in [\![1, L]\!]$ set $\boldsymbol{d}_i = \tau_i^{-1}\boldsymbol{b}_i + \theta_i\boldsymbol{b}_{L+i}$, $\boldsymbol{d}_{L+i} = \tau_i\boldsymbol{b}_{L+i}$, $\boldsymbol{d}_i^* = \tau_i\boldsymbol{b}_i^*$, $\boldsymbol{d}_{L+i}^* = -\theta_i\boldsymbol{b}_i^* + \tau_i^{-1}\boldsymbol{b}_{L+i}^*$ and $\mathbb{D} = (\boldsymbol{b}_0, \boldsymbol{d}_1 \cdots, \boldsymbol{d}_L, \boldsymbol{d}_{L+1}, \cdots, \boldsymbol{d}_{2L}, \boldsymbol{b}_{2L+1} \cdots \boldsymbol{b}_{4L+1})$, and $\mathbb{D}^* = (\boldsymbol{b}_0^*, \boldsymbol{d}_1^*, \cdots, \boldsymbol{d}_L^*, \boldsymbol{b}_L^*, \boldsymbol{d}_{L+1}^*, \cdots, \boldsymbol{d}_{2L}^*, \boldsymbol{b}_{2L+1}^*, \cdots, \boldsymbol{b}_{4L+1}^*)$. This solves the issue raised by our scheme's construction and allows us to prove the indistinguishability between the two games.

Acknowledgements We would like to thank Sherman S. M. Chow and anonymous reviewers for their helpful discussions and valuable comments. Part of this work has received funding from the French National Research Agency (ANR), PRESTO project number ANR-19-CE39-0011-01.

References

1. Abdalla, M., Caro, A.D., Phan, D.H.: Generalized key delegation for wildcarded identity-based and inner-product encryption. IEEE Trans. Inf. Forensics Secur. **7**(6), 1695–1706 (2012). https://doi.org/10.1109/TIFS.2012.2213594, https://doi.org/10.1109/TIFS.2012.2213594
2. Abdalla, M., Catalano, D., Dent, A., Malone-Lee, J., Neven, G., Smart, N.: Identity-based encryption gone wild. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006, Part II. LNCS, vol. 4052, pp. 300–311. Springer, Heidelberg (Jul 2006). https://doi.org/10.1007/11787006_26
3. Agrawal, S., Wichs, D., Yamada, S.: Optimal broadcast encryption from LWE and pairings in the standard model. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part I. LNCS, vol. 12550, pp. 149–178. Springer, Heidelberg (Nov 2020). https://doi.org/10.1007/978-3-030-64375-1_6
4. Ak, M., Pehlivanoglu, S., Selcuk, A.: Anonymous trace and revoke. In: Journal of Computational and Applied Mathematics. pp. 586–591 (2014)
5. Attrapadung, N.: Dual system encryption framework in prime-order groups via computational pair encodings. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 591–623. Springer, Heidelberg (Dec 2016). https://doi.org/10.1007/978-3-662-53890-6_20

6. Barth, A., Boneh, D., Waters, B.: Privacy in encrypted content distribution using private broadcast encryption. In: Di Crescenzo, G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 52–64. Springer, Heidelberg (Feb / Mar 2006)
7. Blazy, O., Kakvi, S.A.: Skipping the q in group signatures. In: Bhargavan, K., Oswald, E., Prabhakaran, M. (eds.) INDOCRYPT 2020. LNCS, vol. 12578, pp. 553–575. Springer, Heidelberg (Dec 2020). https://doi.org/10.1007/978-3-030-65277-7_25
8. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (Aug 2005). https://doi.org/10.1007/11535218_16
9. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (May / Jun 2006). https://doi.org/10.1007/11761679_34
10. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) ACM CCS 2006. pp. 211–220. ACM Press (Oct / Nov 2006). https://doi.org/10.1145/1180405.1180432
11. Boneh, D., Waters, B., Zhandry, M.: Low overhead broadcast encryption from multilinear maps. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 206–223. Springer, Heidelberg (Aug 2014). https://doi.org/10.1007/978-3-662-44371-2_12
12. Brakerski, Z., Vaikuntanathan, V.: Lattice-inspired broadcast encryption and succinct ciphertext-policy abe. Cryptology ePrint Archive, Report 2020/191 (2020), https://ia.cr/2020/191
13. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 595–624. Springer, Heidelberg (Apr 2015). https://doi.org/10.1007/978-3-662-46803-6_20
14. Chen, J., Lim, H.W., Ling, S., Wang, H., Wee, H.: Shorter IBE and signatures via asymmetric pairings. In: Abdalla, M., Lange, T. (eds.) PAIRING 2012. LNCS, vol. 7708, pp. 122–140. Springer, Heidelberg (May 2013). https://doi.org/10.1007/978-3-642-36334-4_8
15. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO'93. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48329-2_40
16. Garg, S., Kumarasubramanian, A., Sahai, A., Waters, B.: Building efficient fully collusion-resilient traitor tracing and revocation schemes. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 2010. pp. 121–130. ACM Press (Oct 2010). https://doi.org/10.1145/1866307.1866322
17. Gay, R., Kowalczyk, L., Wee, H.: Tight adaptively secure broadcast encryption with short ciphertexts and keys. In: Catalano, D., De Prisco, R. (eds.) SCN 18. LNCS, vol. 11035, pp. 123–139. Springer, Heidelberg (Sep 2018). https://doi.org/10.1007/978-3-319-98113-0_7
18. Goyal, R., Quach, W., Waters, B., Wichs, D.: Broadcast and trace with N^{ϵ} ciphertext size from standard assumptions. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 826–855. Springer, Heidelberg (Aug 2019). https://doi.org/10.1007/978-3-030-26954-8_27

19. Goyal, R., Vusirikala, S., Waters, B.: Collusion resistant broadcast and trace from positional witness encryption. In: Lin, D., Sako, K. (eds.) PKC 2019, Part II. LNCS, vol. 11443, pp. 3–33. Springer, Heidelberg (Apr 2019). https://doi.org/10.1007/978-3-030-17259-6_1
20. Kim, J., Lee, J., Lee, S., Oh, H.: Scalable wildcarded identity-based encryption with full security. Electronics (2020)
21. Kim, J., Lee, S., Lee, J., Oh, H.: Scalable wildcarded identity-based encryption. In: López, J., Zhou, J., Soriano, M. (eds.) ESORICS 2018, Part II. LNCS, vol. 11099, pp. 269–287. Springer, Heidelberg (Sep 2018). https://doi.org/10.1007/978-3-319-98989-1_14
22. Lewko, A.B.: Tools for simulating features of composite order bilinear groups in the prime order setting. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 318–335. Springer, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_20
23. Lewko, A.B., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (May / Jun 2010). https://doi.org/10.1007/978-3-642-13190-5_4
24. Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (Feb 2010). https://doi.org/10.1007/978-3-642-11799-2_27
25. Li, J., Gong, J.: Improved anonymous broadcast encryptions - tight security and shorter ciphertext. In: Preneel, B., Vercauteren, F. (eds.) ACNS 18. LNCS, vol. 10892, pp. 497–515. Springer, Heidelberg (Jul 2018). https://doi.org/10.1007/978-3-319-93387-0_26
26. Libert, B., Paterson, K.G., Quaglia, E.A.: Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 206–224. Springer, Heidelberg (May 2012). https://doi.org/10.1007/978-3-642-30057-8_13
27. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (Aug 2010). https://doi.org/10.1007/978-3-642-14623-7_11
28. Okamoto, T., Takashima, K.: Adaptively attribute-hiding (hierarchical) inner product encryption. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 591–608. Springer, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_35
29. Phan, D.H.: Some Advances in Broadcast Encryption and Traitor Tracing. Habilitation à diriger des recherches, Ecole normale supérieure - ENS PARIS (Nov 2014), https://tel.archives-ouvertes.fr/tel-02384086
30. Wang, X., Chow, S.S.M.: Cross-domain access control encryption: Arbitrary-policy, constant-size, efficient. In: 2021 IEEE Symposium on Security and Privacy. pp. 748–761. IEEE Computer Society Press (May 2021). https://doi.org/10.1109/SP40001.2021.00023
31. Zhandry, M.: New techniques for traitor tracing: Size $N^{1/3}$ and more from pairings. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 652–682. Springer, Heidelberg (Aug 2020). https://doi.org/10.1007/978-3-030-56784-2_22

A (Intermediate) assumptions and problems

In this appendix we present the intermediate assumptions and problems used to prove the security of our schemes in section 4. Their reduction to well known assumptions is presented in appendix C.3. We start by recalling the DDH assumption on which SXDH relies.

Definition 17. ***Decisional Diffie-Hellman assumption in*** $\mathbb{G}_1$ *(DDH$_1$) [14]. Given an asymmetric bilinear pairing group $\mathit{\Gamma} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$, we define the following distribution: $a, b, c \leftarrow \mathbb{Z}_p$, $D = (\mathit{\Gamma}, g_1, g_2, g_1^a, g_2^b)$. We assume that for any PPT algorithm $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{DDH}_1}(\lambda) = \left|\Pr\left[\mathcal{A}(D, g_1^{ab})\right] - \Pr\left[\mathcal{A}(D, g_1^{ab+c})\right]\right|$ is negligible in the security parameter λ.*

The dual of above assumption is Decisional Diffie-Hellman assumption in $\mathbb{G}_2$ (denoted as DDH_2),which is identical to DDH_1 with the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ reversed.

Now we present the DS1 and DS2 assumptions used for our first scheme (section 4.1).

Definition 18. ***Decisional subspace assumption in*** $\mathbb{G}_1$ *(DS1) [14]. Given an asymmetric bilinear group generator $\mathcal{G}(.)$, define the following distribution*

$$\begin{aligned}&\mathit{\Gamma} = (p, \mathbb{G}_1,\ \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}(1^\lambda), (\mathbb{B}, \mathbb{B}^*) \leftarrow \mathsf{Dual}(\mathbb{Z}_p^n), \tau_1, \tau_2, \mu_1, \mu_2 \leftarrow \mathbb{Z}_p,\\ &\boldsymbol{u}_1 = g_2^{\mu_1.\boldsymbol{b}_1^* + \mu_2.\boldsymbol{b}_{k+1}^*}, \cdots, \boldsymbol{u}_k = g_2^{\mu_1.\boldsymbol{b}_k^* + \mu_2\boldsymbol{b}_{2k}^*}, \boldsymbol{v}_1 = g_1^{\tau_1.\boldsymbol{b}_1}, \cdots, \boldsymbol{v}_k = g_1^{\tau_1.\boldsymbol{b}_k},\\ &\boldsymbol{w}_1 = g_1^{\tau_1.\boldsymbol{b}_1 + \tau_2\boldsymbol{b}_{k+1}}, \cdots, \boldsymbol{w}_k = g_1^{\tau_1.\boldsymbol{b}_k + \mu_2\boldsymbol{b}_{2k}},\\ &D = (\mathit{\Gamma}, g_2^{\boldsymbol{b}_1^*}, \cdots, g_2^{\boldsymbol{b}_k^*}, g_2^{\boldsymbol{b}_{2k+1}^*}, \cdots, g_2^{\boldsymbol{b}_n^*}, g_1^{\boldsymbol{b}_1}, \cdots, g_1^{\boldsymbol{b}_n}, \boldsymbol{u}_1, \cdots, \boldsymbol{u}_k, \mu_2),\end{aligned}$$

where k, n are fixed positive integers that satisfy $2k \leq n$. We assume that for any PPT algorithm $\mathcal{A}$, the following is negligible in 1^λ.

$$\mathsf{Adv}_{\mathcal{A}}^{DS1}(\lambda) = \left|\Pr\left[\mathcal{A}(D, \boldsymbol{v}_1, \cdots, \boldsymbol{w}_k) = 1\right] - \Pr\left[\mathcal{A}(D, \boldsymbol{w}_1, \cdots, \boldsymbol{v}_k) = 1\right]\right|$$

Lemma 1. *If the decisional Diffie Hellman assumption (DDH) in $\mathbb{G}_1$ holds, then the decisional subspace assumption in $\mathbb{G}_1$ (DS1) also holds.*

For the proof, refer to [14]. The **decisional subspace assumption** in $\mathbb{G}_2$ is defined as identical to DS1 with the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ reversed. DS2 holds if DDH in $\mathbb{G}_2$ holds. The proof is done as for $\mathbb{G}_1$.

In section 4.2, to prove the security of our second WIBE, we used intermediate problems based on the ones of [28]. Ours are however in the asymmetric pairing setting and can be reduced to $\mathsf{XDLin}_1, \mathsf{XDLin2}$ assumptions.

Definition 19. ***Problem 1*** *is to guess β, given $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{B}, \hat{\mathbb{B}}^*, \boldsymbol{e}_{\beta,1}, \{\boldsymbol{e}_i\}_{i=2,\cdots,n}) \leftarrow \mathcal{G}_\beta^{P1}(1^\lambda, n)$, where $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p,\ g_1, g_2, e)$ is an asymmetric bilinear pairing group, $(\mathbb{B}, \mathbb{B}^*) \leftarrow \mathsf{Dual}(\mathbb{Z}_p^{4n+2})$, $\hat{\mathbb{B}}^* = (\boldsymbol{b}_0^*, \cdots, \boldsymbol{b}_n^*, \boldsymbol{b}_{2n+1}^*, \cdots, \boldsymbol{b}_{4n+1}^*)$ and*

$$\begin{aligned}&\omega, \gamma, z \leftarrow \mathbb{Z}_p, && \boldsymbol{e}_{0,1} = g_1^{\omega\boldsymbol{b}_1 + \gamma\boldsymbol{b}_{4n+1}}\\ &\boldsymbol{e}_{1,1} = g_1^{\omega\boldsymbol{b}_1 + z\boldsymbol{b}_{n+1} + \gamma\boldsymbol{b}_{4n+1}} && \boldsymbol{e}_i = g_1^{\omega\boldsymbol{b}_i} \textit{ for } i = 2, \cdots, n.\end{aligned}$$

For a probabilistic adversary $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 1 is defined as

$$\mathsf{Adv}_{\mathcal{B}}^{P1}(\lambda) = \left| \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_0^{P1}(1^\lambda, n)\right] - \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_1^{P1}(1^\lambda, n)\right] \right|$$

Lemma 2. *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{E}$, such that for any security parameter λ, $\mathsf{Adv}_{\mathcal{B}}^{P1}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLin1}}(\lambda) + 5/p$.*

Definition 20. ***Problem 1 bis*** *is to guess β, given $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{B}, \hat{\mathbb{B}^*}, \boldsymbol{e}_{\beta,1}, \{\boldsymbol{e}_i\}_{i \in [\![1,n]\!]}) \leftarrow \mathcal{G}_\beta^{P1b}(1^\lambda, n)$, where $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ is an asymmetric bilinear pairing group, $(\mathbb{B}, \mathbb{B}^*) \leftarrow \mathsf{Dual}(\mathbb{Z}_p^{4n+2})$, $\hat{\mathbb{B}}^* = (\boldsymbol{b}_0^*, \cdots, \boldsymbol{b}_n^*, \boldsymbol{b}_{2n+1}^*, \cdots, \boldsymbol{b}_{4n+1}^*)$ and*

$$\omega, \gamma, \{\boldsymbol{z}_i\}_{i=1}^n \leftarrow \mathbb{Z}_p^n \qquad \boldsymbol{e}_{0,1} = g_1^{\omega \boldsymbol{b}_1 + \gamma \boldsymbol{b}_{4n+1}}$$

$$\boldsymbol{e}_{1,1} = g_1^{\omega \boldsymbol{b}_1 + \sum_{i=1}^n \sum_{j=1}^n z_{i,j} \boldsymbol{b}_{n+i} + \gamma \boldsymbol{b}_{4n+1}} \quad \boldsymbol{e}_i = g_1^{\omega \boldsymbol{b}_i},$$

for $i = 2, \cdots, n$. For a probabilistic adversary $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 1 bis is defined as

$$\mathsf{Adv}_{\mathcal{B}}^{P1b}(\lambda) = \left| \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_0^{P1b}(1^\lambda, n)\right] - \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_1^{P1b}(1^\lambda, n)\right] \right|$$

Lemma 3. *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{E}$, such that for any security parameter λ, $\mathsf{Adv}_{\mathcal{B}}^{P1b}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLin1}}(\lambda) + 5/p$.*

Definition 21. ***Problem 2*** *is to guess β, given $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \mathbb{B}^*, \left\{\boldsymbol{h}_{\beta,i}^*, \boldsymbol{e}_i\right\}_{i \in [\![1,n]\!]}) \leftarrow \mathcal{G}_\beta^{P2}(1^\lambda, n)$, where $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ is an asymmetric bilinear pairing group, $(\mathbb{B}, \mathbb{B}^*) \leftarrow \mathsf{Dual}(\mathbb{Z}_p^{4n+2})$, $\hat{\mathbb{B}} = (\boldsymbol{b}_0, \cdots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \cdots, \boldsymbol{b}_{4n+1})$, $\delta, \tau, \delta_0, \omega, \sigma \leftarrow \mathbb{Z}_p$ and for $i \in [\![1,n]\!]$:*

$$\boldsymbol{h}_{0,i}^* = g_2^{\delta \boldsymbol{b}_i^* + \delta_0 \boldsymbol{b}_{3n+i}^*}, \ \boldsymbol{h}_{1,i}^* = g_2^{\delta \boldsymbol{b}_i^* + \tau \boldsymbol{b}_{n+i}^* + \delta_0 \boldsymbol{b}_{3n+i}^*}, \ \boldsymbol{e}_i = g_1^{\omega \boldsymbol{b}_i + \sigma \boldsymbol{b}_{n+i}}.$$

For a probabilistic adversary $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 2 is defined as

$$\mathsf{Adv}_{\mathcal{B}}^{P2}(\lambda) = \left| \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_0^{P2}(1^\lambda, n)\right] - \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_1^{P2}(1^\lambda, n)\right] \right|$$

Lemma 4. *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{E}$, such that for any security parameter λ, $\mathsf{Adv}_{\mathcal{B}}^{P2}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLin2}}(\lambda) + 5/p$.*

Definition 22. ***Problem 2 bis*** *is to guess β, given $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \mathbb{B}^*, \left\{\boldsymbol{h}_{\beta,i}^*, \boldsymbol{e}_i\right\}_{i \in [\![1,n]\!]}) \leftarrow \mathcal{G}_\beta^{P2b}(1^\lambda, n)$, where $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ is an asymmetric bilinear pairing group, $(\mathbb{B}, \mathbb{B}^*) \leftarrow \mathsf{Dual}(\mathbb{Z}_p^{4n+2})$, $\hat{\mathbb{B}} = (\boldsymbol{b}_0, \cdots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \cdots, \boldsymbol{b}_{4n+1})$ and*

$$\delta, \tau, \delta_0, \omega, \sigma \leftarrow \mathbb{Z}_p, \left\{\delta_i \leftarrow \mathbb{Z}_p^n\right\}_{i=1}^n, \qquad \boldsymbol{Z} \leftarrow \mathsf{GL}(n, \mathbb{Z}_p), \boldsymbol{U} = (\boldsymbol{Z}^{-1})^\top$$

$$\text{for } i \in [\![1,n]\!] : \qquad \boldsymbol{h}_{0,i}^* = g_2^{\delta \boldsymbol{b}_i^* + \sum_{j=1}^n \delta_{i,j} \boldsymbol{b}_{3n+i}^*}$$

$$\boldsymbol{h}_{1,i}^* = g_2^{\delta \boldsymbol{b}_i^* + \sum_{j=1}^n u_{i,j} \boldsymbol{b}_{n+i}^* + \sum_{j=1}^n \delta_{i,j} \boldsymbol{b}_{3n+i}^*} \quad \boldsymbol{e}_i = g_1^{\omega \boldsymbol{b}_i + \tau \sum_{j=1}^n z_{i,j} \boldsymbol{b}_{n+i}}.$$

For a probabilistic adversary $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 2bis is defined as

$$\mathsf{Adv}_{\mathcal{B}}^{P2b}(\lambda) = \left| \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_0^{P2b}(1^\lambda, n)\right] - \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_1^{P2b}(1^\lambda, n)\right] \right|$$

Lemma 5. *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{E}$, such that for any security parameter λ, $\mathsf{Adv}_{\mathcal{B}}^{P2b}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLin2}}(\lambda) + 5/p$.*

Definition 23. ***Problem 3** is to guess β, given $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \hat{\mathbb{B}}^*, \left\{\boldsymbol{h}_{\beta,i}^*, \boldsymbol{e}_i, \boldsymbol{f}_i\right\}_{i \in [\![1,n]\!]}) \leftarrow \mathcal{G}_\beta^{P3}(1^\lambda, n)$, where $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ is an asymmetric bilinear pairing group, $(\mathbb{B}, \mathbb{B}^*) \leftarrow \mathsf{Dual}(\mathbb{Z}_p^{4n+2})$ and*

$$\hat{\mathbb{B}} = (\boldsymbol{b}_0, \cdots, \boldsymbol{b}_n, \boldsymbol{b}_{2n+1}, \cdots, \boldsymbol{b}_{4n+1}),\ \ \hat{\mathbb{B}}^* = (\boldsymbol{b}_0^*, \cdots, \boldsymbol{b}_n^*, \boldsymbol{b}_{2n+1}^*, \cdots, \boldsymbol{b}_{4n+1}^*)$$

$$\begin{array}{ll} \tau, \delta_0, \omega, \omega^{'}, \omega^{''}, \kappa^{'}, \kappa^{''} \leftarrow \mathbb{Z}_p,\ \ \textit{for } i \in [\![1, n]\!] & \\ \boldsymbol{h}_{0,i}^* = g_2^{\tau \boldsymbol{b}_{n+i}^* + \delta_0 \boldsymbol{b}_{3n+i}^*} & \boldsymbol{h}_{1,i}^* = g_2^{\tau \boldsymbol{b}_{2n+i}^* + \delta_0 \boldsymbol{b}_{3n+i}^*} \\ \boldsymbol{e}_i = g_1^{\omega^{'} \boldsymbol{b}_{n+i} + \omega^{''} \boldsymbol{b}_{2n+i}} & \boldsymbol{f}_i = g_1^{\kappa^{'} \boldsymbol{b}_{n+i} + \kappa^{''} \boldsymbol{b}_{2n+i}}. \end{array}$$

For a probabilistic adversary $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 3 is defined as

$$\mathsf{Adv}_{\mathcal{B}}^{P3}(\lambda) = \left| \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_0^{P3}(1^\lambda, n)\right] - \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_1^{P3}(1^\lambda, n)\right] \right|$$

Lemma 6. *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}_{\mathcal{B}}^{P3}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLin2}}(\lambda) + 7/p$.*

B Security proofs

In this appendix we give the full security proofs of our schemes.

B.1 Constant size ciphertext WIBE security proof

First we prove the indistinguishability between the security games presented in section 4.1, by proving following lemmas 7, 8 and 9.

Lemma 7. *If there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}_{\mathcal{A}}^0 - \mathsf{Adv}_{\mathcal{A}}^1$ is non-negligible, then there exists a PPT algorithm $\mathcal{B}$ with non-negligible advantage against DS1 with $k = 2$ and $n = 4$.*

Proof. `INIT`: $\mathcal{B}$ is given $D = (\mathit{\Gamma}, g_2^{\boldsymbol{b}_1^*}, g_2^{\boldsymbol{b}_2^*}, g_1^{\boldsymbol{b}_1}, g_1^{\boldsymbol{b}_2}, g_1^{\boldsymbol{b}_3}, g_1^{\boldsymbol{b}_4}, \boldsymbol{u}_1, \boldsymbol{u}_2, \mu_2)$

along with $\boldsymbol{t}_1, \boldsymbol{t}_2$, distributed either as $g_1^{\tau_1 \boldsymbol{b}_1}$, $g_1^{\tau_1 \boldsymbol{b}_2}$ or $g_1^{\tau_1 \boldsymbol{b}_1 + \tau_2 \boldsymbol{b}_3}$, $g_1^{\tau_1 \boldsymbol{b}_2 + \tau_2 \boldsymbol{b}_4}$.

`SETUP`: $\mathcal{B}$ first chooses a random invertible matrix $\boldsymbol{A} \in \mathbb{Z}_p^{2\times 2}$. It implicitly sets dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to: $\boldsymbol{d}_1 = \boldsymbol{b}_1, \boldsymbol{d}_2 = \boldsymbol{b}_2$, $(\boldsymbol{d}_3, \boldsymbol{d}_4) = (\boldsymbol{b}_3, \boldsymbol{b}_4) \cdot \boldsymbol{A}$, $\boldsymbol{d}_1^* = \boldsymbol{b}_1^*$, $\boldsymbol{d}_2^* = \boldsymbol{b}_2^*$, $(\boldsymbol{d}_3^*, \boldsymbol{d}_4^*) = (\boldsymbol{b}_3^*, \boldsymbol{b}_4^*) \cdot (\boldsymbol{A}^{-1})^\top$.

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed and reveal no information about $\boldsymbol{A}$. Notice also that $\mathcal{B}$ cannot produce $g_2^{\boldsymbol{d}_3^*}, g_2^{\boldsymbol{d}_4^*}$, but these will not be needed to create normal keys. $\mathcal{B}$ chooses random values $\alpha, a_1, \cdots, a_L \in \mathbb{Z}_p$. $\mathcal{A}$ is given the public key

$$\mathsf{pk} = (\mathit{\Gamma}, e(g_1, g_2)^{\alpha \boldsymbol{d}_1 \cdot \boldsymbol{d}_1^*}, g_1^{\boldsymbol{d}_1}, \boldsymbol{h}_1 = g_1^{a_1 \boldsymbol{d}_2}, \cdots, \boldsymbol{h}_L = g_1^{a_L \boldsymbol{d}_2}).$$

The master key is $\mathsf{msk} = (\alpha, g_2^{\boldsymbol{d}_1^*}, g_2^{\boldsymbol{d}_2^*}, a_1, \cdots, a_L)$.

`KEY QUERY`: msk is known to $\mathcal{B}$, which allows $\mathcal{B}$ to respond to all of $\mathcal{A}$'s key queries by calling the normal key generation algorithm.

`CHALLENGE`: $\mathcal{A}$ sends $\mathcal{B}$ a challenge pattern $\boldsymbol{P}$ and two messages $(\mathsf{m}_0, \mathsf{m}_1)$. $\mathcal{B}$ chooses a random bit $b \in \{0, 1\}$ and encrypts m_b under $\boldsymbol{P}$ as follows:

$$c_1 = \mathsf{m}_b \cdot (e(\boldsymbol{t}_1, g_2^{\boldsymbol{b}_1^*}))^{\alpha} \ , \boldsymbol{c}_2 = \boldsymbol{t}_1 \cdot \boldsymbol{t}_2^{\sum_{i \in W(\boldsymbol{P})} a_i}.$$

It gives the ciphertext $\mathsf{ct}^* = (c_1, \boldsymbol{c}_2)$ to $\mathcal{A}$.

- If $(\boldsymbol{t}_1, \boldsymbol{t}_2) = (g_1^{\tau_1 \boldsymbol{b}_1}, g_1^{\tau_1 \boldsymbol{b}_2})$, we have a normal ciphertext with randomness τ_1: $c_1 = (\mathsf{m}_b(e(g_1, g_2)^{\boldsymbol{b}_1 \cdot \boldsymbol{b}_1^* \alpha})^{\tau_1}$, and $\boldsymbol{c}_2 = g_1^{\tau_1 \boldsymbol{b}_1 + \tau_1 \boldsymbol{b}_2 \sum_{i \in W(\boldsymbol{P})} a_i}$. Thus $\mathcal{B}$ has properly simulated Game_0.

- If $(\boldsymbol{t}_1, \boldsymbol{t}_2) = g_1^{\tau_1 \boldsymbol{b}_1 + \tau_2 \boldsymbol{b}_3}, g_1^{\tau_1 \boldsymbol{b}_2 + \tau_2 \boldsymbol{b}_4}$, $c_1 = \mathsf{m}_b \cdot (e(g_1, g_2)^{\boldsymbol{b}_1 \cdot \boldsymbol{b}_1^* \alpha})^{\tau_1} \cdot e(g_1, g_2)^{\tau_2 \boldsymbol{b}_3 \boldsymbol{b}_1^* \alpha}$ $= \mathsf{m}_b \cdot (e(g_1, g_2)^{\boldsymbol{b}_1 \cdot \boldsymbol{b}_1^* \alpha})^{\tau_1}$ and $\boldsymbol{c}_2 = g_1^{\tau_1 \boldsymbol{b}_1 + \tau_1 \boldsymbol{b}_2 \sum_{i \in W(\boldsymbol{P})} a_i + \tau_2 \boldsymbol{b}_3 + \tau_2 \boldsymbol{b}_4 \sum_{i \in W(\boldsymbol{P})} a_i}$.

This ciphertext has an additional term with coefficients in basis $\boldsymbol{b}_3, \boldsymbol{b}_4$, which form the vector $\tau_2(1, \sum_{i \in W(\boldsymbol{P})} a_i)$. To compute coefficients in the basis $(\boldsymbol{d}_3, \boldsymbol{d}_4)$ we multiply the matrix $\boldsymbol{A}^{-1}$ by the transpose of this vector. Since $\boldsymbol{A}$ is random, these new coefficients are uniformly random. Thus in this case the ciphertext is SF (with coefficients in the base $\mathbb{D}$) and $\mathcal{B}$ has properly simulated Game_1. This allows $\mathcal{B}$ to leverage $\mathcal{A}$'s non-negligible difference in advantage between Game_0 and Game_1 to achieve a non-negligible advantage against DS1.

Lemma 8. *If there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}_{\mathcal{A}}^{2-(j-1)} - \mathsf{Adv}_{\mathcal{A}}^{2-j}$ is non-negligible, then there exists a PPT algorithm $\mathcal{B}$ with non-negligible advantage against DS2 with $k = 2$ and $n = 4$.*

Proof. `INIT`: $\mathcal{B}$ is given $D = (\mathit{\Gamma}, g_1^{\boldsymbol{b}_1}, g_1^{\boldsymbol{b}_2}, g_2^{\boldsymbol{b}_1^*}, g_2^{\boldsymbol{b}_2^*}, g_2^{\boldsymbol{b}_3^*}, g_2^{\boldsymbol{b}_4^*}, \boldsymbol{u}_1, \boldsymbol{u}_2, \mu_2)$

along with $\boldsymbol{t}_1, \boldsymbol{t}_2$, distributed either as $g_2^{\tau_1 \boldsymbol{b}_1^*}, g_2^{\tau_1 \boldsymbol{b}_2^*}$ or $g_2^{\tau_1 \boldsymbol{b}_1^* + \tau_2 \boldsymbol{b}_3^*}$, $g_2^{\tau_1 \boldsymbol{b}_2^* + \tau_2 \boldsymbol{b}_4^*}$.

`SETUP`: $\mathcal{B}$, chooses a random invertible matrix $\boldsymbol{A} \in \mathbb{Z}_q^{2 \times 2}$. Then it implicitly sets dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to: $\boldsymbol{d}_1 = \boldsymbol{b}_1$, $\boldsymbol{d}_2 = \boldsymbol{b}_2$, $(\boldsymbol{d}_3, \boldsymbol{d}_4) = (\boldsymbol{b}_3, \boldsymbol{b}_4) \cdot \boldsymbol{A}$, $\boldsymbol{d}_1^* = \boldsymbol{b}_1^*$, $\boldsymbol{d}_2^* = \boldsymbol{b}_2^*$,$(\boldsymbol{d}_3^*, \boldsymbol{d}_4^*) = (\boldsymbol{b}_3^*, \boldsymbol{b}_4^*) \cdot (\boldsymbol{A}^{-1})^{\top}$.

We note that $\mathbb{D}, \mathbb{D}^*$ are properly distributed and reveal no information about $\boldsymbol{A}$. $\mathcal{B}$ chooses random values $\alpha, a_1, \cdots, a_L \in \mathbb{Z}_p$. $\mathcal{A}$ is given the public key

$$\mathsf{pk} = (\mathit{\Gamma}, e(g_1, g_2)^{\alpha \boldsymbol{d}_1 . \boldsymbol{d}_1^*}, g_1^{\boldsymbol{d}_1}, \boldsymbol{h}_1 = g_1^{a_1 \boldsymbol{d}_2}, \cdots, \boldsymbol{h}_L = g_1^{a_L \boldsymbol{d}_2}).$$

The master key is $\mathsf{msk} = (\alpha, g_2^{\boldsymbol{d}_1^*}, g_2^{\boldsymbol{d}_2^*}, a_1, \cdots, a_L)$.

`KEY QUERY`: $\mathcal{B}$ knows msk and $g_2^{\boldsymbol{d}_3^*}, g_2^{\boldsymbol{d}_4^*}$, thus can easily call the key generation algorithm or produce semi-functional keys. It allows $\mathcal{B}$ to answer to all $\mathcal{A}$'s key queries.

- To answer the first j-1 key queries that $\mathcal{A}$ makes, $\mathcal{B}$ runs the semi-functional key generation algorithm to produce semi-functional keys.
- To answer to the j-th key query for $\boldsymbol{P}^j$, $\mathcal{B}$ responds with:

  $$\boldsymbol{a} = (g_2^{\boldsymbol{b}_1^*})^\alpha \cdot \boldsymbol{t}_1^{\sum_{i \in \mathcal{I}} a_i} \cdot \boldsymbol{t}_2^{-1}, \;\; \boldsymbol{b}_i = \boldsymbol{t}_1^{a_i} \;\text{ for } i \in \mathcal{O}.$$

  - If $\boldsymbol{t}_1, \boldsymbol{t}_2 = g_2^{\tau_1 \boldsymbol{b}_1^*}, g_2^{\tau_1 \boldsymbol{b}_2^*}$, then $\mathsf{sk}_{\boldsymbol{P}^j}$ is a normal key with randomness τ_1.
  - If $\boldsymbol{t}_1, \boldsymbol{t}_2 = g_2^{\tau_1 \boldsymbol{b}_1^* + \tau_2 \boldsymbol{b}_3^*}, g_2^{\tau_1 \boldsymbol{b}_2^* + \tau_2 \boldsymbol{b}_4^*}$, then it is a semi-functional key.
- For the remaining key queries, $\mathcal{B}$ runs the normal key generation algorithm.

`CHALLENGE`: At some point, $\mathcal{A}$ sends $\mathcal{B}$ two messages $\mathsf{m}_0, \mathsf{m}_1$ and a challenge pattern $\boldsymbol{P}$. $\mathcal{B}$ chooses a random bit $b \in \{0, 1\}$ and encrypts m_b under $\boldsymbol{P}$ as follows: $c_1 = \mathsf{m}_b \cdot (e(\boldsymbol{u}_1, g_2^{\boldsymbol{b}_1^*}))^\alpha$, $\boldsymbol{c}_2 = \boldsymbol{u}_1 \cdot \boldsymbol{u}_2^{\sum_{i \in W(\boldsymbol{P})} a_i}$.

Suppose that $\mathcal{B}$ decides not to be honest, and find the nature of the j-th key by itself. To do so, it creates a ciphertext for a pattern $\boldsymbol{P}^*$ such that $\boldsymbol{P}^j \in_\star \boldsymbol{P}^*$. He tries to decrypt it with $\mathsf{sk}_{\boldsymbol{P}^j}$ to learn if $\mathsf{sk}_{\boldsymbol{P}^j}$ is a normal or a SF key (a normal key will decrypt correctly while a SF key will with high probability fail to decrypt). Let's see that by construction even if $\mathsf{sk}_{\boldsymbol{P}^j}$ is SF it will decrypt correctly.

Suppose that $\boldsymbol{t}_1, \boldsymbol{t}_2 = (g_2^{\tau_1 \boldsymbol{b}_1^* + \tau_2 \boldsymbol{b}_3^*}, g_2^{\tau_1 \boldsymbol{b}_2^* + \tau_2 \boldsymbol{b}_4^*})$. During decryption, $\mathcal{B}$ obtains the term $e\left(g_1^{\mu_2 \boldsymbol{b}_3 + \mu_2 \sum_{i \in W(\boldsymbol{P}^*)} a_i \boldsymbol{b}_4}, g_2^{\tau_2 \boldsymbol{b}_3^* \sum_{i \in \mathcal{I}} a_i - \tau_2 \boldsymbol{b}_4^*} \cdot g_2^{\tau_2 \boldsymbol{b}_3^* \sum_{i \in W(\boldsymbol{P}^*) \cap \mathcal{O}} a_i}\right)$. In the exponent we have $\mu_2(\boldsymbol{b}_3 + \boldsymbol{b}_4 \sum_{i \in W(\boldsymbol{P}^*)} a_i \boldsymbol{b}_4) \cdot \tau_2(\boldsymbol{b}_3^* \sum_{i \in W(\boldsymbol{P}^*)} a_i - \boldsymbol{b}_4^*)$ because $\boldsymbol{P}^j \in_\star \boldsymbol{P}^*$ implies $\mathcal{I} \cap (W(\boldsymbol{P}^*) \cup \mathcal{O}) = W(\boldsymbol{P}^*)$. The term in the exponent is: $\mu_2 \tau_2 \psi \sum_{i \in W(\boldsymbol{P}^*)} a_i - \mu_2 \tau_2 \, \psi \sum_{i \in W(\boldsymbol{P}^*)} a_i = 0$. Thus it will decrypt, and $\mathcal{B}$ will have no information about the j-th key 's nature.

In the authorized case, $\boldsymbol{P}^j \notin_\star \boldsymbol{P}$. Let's see that the extra coefficients in basis $(\boldsymbol{b}_3, \boldsymbol{b}_4)$ of the ciphertext and the extra coefficients in basis $(\boldsymbol{b}_3^*, \boldsymbol{b}_4^*)$ of the key are distributed as random vectors in the spans of $(\boldsymbol{d}_3, \boldsymbol{d}_4)$ and $(\boldsymbol{d}_3^*, \boldsymbol{d}_4^*)$ respectively. To express them in basis $(\boldsymbol{d}_3, \boldsymbol{d}_4)$ and $(\boldsymbol{d}_3^*, \boldsymbol{d}_4^*)$ respectively, we multiply them by $\boldsymbol{A}^\top$ and $\boldsymbol{A}^{-1}$ respectively. Since the distribution of everything given to $\mathcal{A}$ except for the $j-$th key and the challenge ciphertext is independent of the random matrix $\boldsymbol{A}$ and $\boldsymbol{P}^j \notin_\star \boldsymbol{P}$, we can conclude that these coefficients are uniformly random. Thus $\mathcal{B}$ has properly simulated Game_{2-j} in this case.

If $\boldsymbol{t}_1, \boldsymbol{t}_2 = g_2^{\tau_1 \boldsymbol{b}_1^*}, g_2^{\tau_1 \boldsymbol{b}_2^*}$ then the coefficients of the semi functional part of the ciphertext are uniformly random. Thus $\mathcal{B}$ has properly simulated $\mathsf{Game}_{2-(j-1)}$ in this case. Therefore $\mathcal{B}$ can leverage $\mathcal{A}$'s non-negligible difference in advantage between these games to obtain a non-negligible advantage against DS2.

Lemma 9. *If there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}_\mathcal{A}^{2-Q} - \mathsf{Adv}_\mathcal{A}^3$ is non-negligible, then there exists a PPT algorithm $\mathcal{B}$ with non-negligible advantage against DS1 with $k = 1$ and $n = 4$.*

We prove this lemma in two steps, by randomizing each appearance of s in the $\boldsymbol{c}_2$ term of the ciphertext, thereby severing its link with the blinding factor. The end result is a SF encryption of a random message. As a first step, we consider an intermediary game, called $\mathsf{Game}_{2-Q'}$, that is exactly like Game_{2-Q}, except that in the $\boldsymbol{c}_2$ term of the challenge ciphertext the coefficient of $\boldsymbol{d}_2$ is changed from being $s\sum_{i\in W(\boldsymbol{P})} a_i$ to a fresh random value in $\mathbb{Z}_p$. We denote the advantage of an algorithm $\mathcal{A}$ in this game by $\mathsf{Adv}_{\mathcal{A}}^{Q^{'}}$. We first prove the following lemma.

Lemma 10. *If there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}_{\mathcal{A}}^{2-Q} - \mathsf{Adv}_{\mathcal{A}}^{2-Q^{'}}$ is non-negligible, then there exists a PPT algorithm $\mathcal{B}$ with non-negligible advantage against DS1 with $k = 1$ and $n = 4$.*

Proof. `INIT`: $\mathcal{B}$ is given $D = (\mathit{\Gamma}, g_2^{\boldsymbol{b}_1^*}, g_2^{\boldsymbol{b}_3^*}, g_2^{\boldsymbol{b}_4^*}, g_1^{\boldsymbol{b}_1}, g_1^{\boldsymbol{b}_2}, g_1^{\boldsymbol{b}_3}, g_1^{\boldsymbol{b}_4}, \boldsymbol{u}_1, \mu_2)$, along with $\boldsymbol{t}_1$ either equal to $g_1^{\tau_1\boldsymbol{b}_1}$ or $g_1^{\tau_1\boldsymbol{b}_1+\tau_2\boldsymbol{b}_2}$.

`SETUP`: $\mathcal{B}$ implicitly sets $\boldsymbol{d}_1 = \boldsymbol{b}_3, \boldsymbol{d}_2 = \boldsymbol{b}_2, \boldsymbol{d}_3 = \boldsymbol{b}_1, \boldsymbol{d}_4 = \boldsymbol{b}_4$, and $\boldsymbol{d}_1^* = \boldsymbol{b}_3^*, \boldsymbol{d}_2^* = \boldsymbol{b}_2^*, \boldsymbol{d}_3^* = \boldsymbol{b}_1^*, \boldsymbol{d}_4^* = \boldsymbol{b}_4^*$.

This enables $\mathcal{B}$ to produce $g_1^{\boldsymbol{d}_1}, g_1^{\boldsymbol{d}_2}, g_1^{\boldsymbol{d}_3}, g_1^{\boldsymbol{d}_4}$. We note also that $\mathbb{D}, \mathbb{D}^*$ are properly distributed dual orthonormal bases, and that $\mathcal{B}$ can produce $g_2^{\boldsymbol{d}_1^*}, g_2^{\boldsymbol{d}_3^*}$ and $g_2^{\boldsymbol{d}_4^*}$ but does not know $g_2^{\boldsymbol{d}_2^*}$. $\mathcal{B}$ chooses random values $\alpha, a_1, \cdots, a_L \in \mathbb{Z}_p$. It gives $\mathcal{A}$ the public key

$$\mathsf{pk} = (\mathit{\Gamma}, p, e(g_1, g_2)^{\alpha\boldsymbol{d}_1\cdot\boldsymbol{d}_1^*}, g_1^{\boldsymbol{d}_1}, \boldsymbol{h}_1 = \boldsymbol{g}_1^{a_1\boldsymbol{d}_2}, \cdots, \boldsymbol{h}_L = g_1^{a_L\boldsymbol{d}_2}).$$

`KEY QUERY`: We note that $\mathcal{B}$ does not know the full master secret key, but he knows $\boldsymbol{u}_1 = g_2^{\mu_1\boldsymbol{b}_1^*+\mu_2\boldsymbol{b}_2^*}$, μ_2 and $a_1, \cdots, a_L$. This allows him to produce SF keys as follows: when $\mathcal{A}$ requests a key for some pattern $\boldsymbol{P}^{'}$, $\mathcal{B}$ chooses random values $r^{'}, t_4 \in \mathbb{Z}_p$. It sets $r = \mu_2 r^{'}$ and forms the secret key as: $\boldsymbol{a} = (\boldsymbol{u}_1)^{-r^{'}} \cdot g_2^{\alpha\boldsymbol{d}_1^*+\mu_2 r^{'}\sum_{i\in\mathcal{I}} a_i\boldsymbol{d}_1^*+t_4\boldsymbol{d}_4^*}$, $\boldsymbol{b}_i = g_2^{\mu_2 r^{'} a_i + t_{b,i}\boldsymbol{d}_3^*}$.

We obtain that $\boldsymbol{a} = g_2^{\alpha\boldsymbol{d}_1^*+r\boldsymbol{d}_1^*\sum_{i\in\mathcal{I}} a_i - r\boldsymbol{d}_2^* + (-r^{'}\mu_1)\boldsymbol{d}_3^*+t_4\boldsymbol{d}_4^*}$. The coefficients of $\boldsymbol{d}_3^*, \boldsymbol{d}_4^*$ are uniformly random thus it is a SF key.

`CHALLENGE`: $\mathcal{A}$ submits two messages $\mathsf{m}_0, \mathsf{m}_1$ and a challenge pattern $\boldsymbol{P}$. $\mathcal{B}$ chooses $b \in \{0, 1\}$ and forms the challenge ciphertext as follows:

$$c_1 = \mathsf{m}_b \cdot (e(g_1, g_2)^{\alpha\boldsymbol{d}_1\cdot\boldsymbol{d}_1^*})^s, \;\; \boldsymbol{c}_2 = g_1^{s\boldsymbol{d}_1+s\boldsymbol{d}_2\sum_{i\in W(\boldsymbol{P})} a_i} \cdot \boldsymbol{t}_1 \cdot g_1^{z\boldsymbol{d}_4}$$

where $s, z \leftarrow \mathbb{Z}_p$.

– If $\boldsymbol{t}_1$ is equal to $g_1^{\tau_1\boldsymbol{b}_1}$ then $\boldsymbol{c}_2 = g_1^{s\boldsymbol{d}_1+s\boldsymbol{d}_2\sum_{i\in W(\boldsymbol{P})} a_i+\tau_1\boldsymbol{d}_3+z\boldsymbol{d}_4}$ which is a semi functional ciphertext and $\mathcal{B}$ simulates Game_{2-Q}.

- If $\boldsymbol{t}_1 = g_1^{\tau_1 \boldsymbol{b}_1 + \tau_2 \boldsymbol{b}_2}$ then $\boldsymbol{c}_2 = g_1^{s\boldsymbol{d}_1 + (s\sum_{i \in W(\boldsymbol{P})} a_i + \tau_2)\boldsymbol{d}_2 + \tau_1 \boldsymbol{d}_3 + z\boldsymbol{d}_4}$ is a semi functional ciphertext with randomized coefficients for $\boldsymbol{d}_2$, thus $\mathcal{B}$ simulates $\mathsf{Game}_{2-Q'}$.

Therefore, $\mathcal{B}$ can leverage $\mathcal{A}$'s non-negligible difference of advantage between these two games to achieve a non-negligible advantage against DS1.

Note 10. All queried keys shared μ_1, μ_2 in their randomness. However, as it is in exponent and "randomized" by other random elements, then for an adversary it is indistinguishable from a truly random element.

Lemma 11. *If there exists a PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}_{\mathcal{A}}^{2-Q'} - \mathsf{Adv}_{\mathcal{A}}^{3}$ is non-negligible, then there exists a PPT algorithm $\mathcal{B}$ with non-negligible advantage against DS1 with $k = 1$ and $n = 4$.*

Proof. `INIT`: $\mathcal{B}$ is given $D = (\mathit{\Gamma}, g_2^{\boldsymbol{b}_1^*}, g_2^{\boldsymbol{b}_3^*}, g_2^{\boldsymbol{b}_4^*}, g_1^{\boldsymbol{b}_1}, g_1^{\boldsymbol{b}_2}, g_1^{\boldsymbol{b}_3}, g_1^{\boldsymbol{b}_4}, \boldsymbol{u}_1, \mu_2)$, along with $\boldsymbol{t}_1$ either equal to $g_1^{\tau_1 \boldsymbol{b}_1}$ or $g_1^{\tau_1 \boldsymbol{b}_1 + \tau_2 \boldsymbol{b}_2}$.

`SETUP`: $\mathcal{B}$ implicitly sets $\boldsymbol{d}_1 = \boldsymbol{b}_2, \boldsymbol{d}_2 = \boldsymbol{b}_3, \boldsymbol{d}_3 = \boldsymbol{b}_1, \boldsymbol{d}_4 = \boldsymbol{b}_4$, and $\boldsymbol{d}_1^* = \boldsymbol{b}_2^*, \boldsymbol{d}_2^* = \boldsymbol{b}_3^*, \boldsymbol{d}_3^* = \boldsymbol{b}_1^*, \boldsymbol{d}_4^* = \boldsymbol{b}_4^*$.

This enables $\mathcal{B}$ to produce $g_1^{\boldsymbol{d}_1}, g_1^{\boldsymbol{d}_2}, g_1^{\boldsymbol{d}_3}, g_1^{\boldsymbol{d}_4}$, but not $\boldsymbol{d}_2$. We note also that $\mathbb{D}, \mathbb{D}^*$ are properly distributed dual orthonormal bases, and that $\mathcal{B}$ can produce $g_2^{\boldsymbol{d}_2^*}, g_2^{\boldsymbol{d}_3^*}$ and $g_2^{\boldsymbol{d}_4^*}$ but does not know $g_2^{\boldsymbol{d}_1^*}$. $\mathcal{B}$ chooses random values $\alpha', a_1, \cdots, a_L \in \mathbb{Z}_p$. It computes $e(g_1^{\boldsymbol{b}_3}, g_2^{\boldsymbol{b}_3^*})^\alpha = e(g_1, g_2)^{\alpha \boldsymbol{d}_2 \cdot \boldsymbol{d}_2^*} = e(g_1, g_2)^{\alpha\psi} = e(g_1, g_2)^{\alpha \boldsymbol{d}_1 \cdot \boldsymbol{d}_1^*}$. It gives $\mathcal{A}$ the public key

$$\mathsf{pk} = (\mathit{\Gamma}, p, e(g_1, g_2)^{\alpha \boldsymbol{d}_1 \cdot \boldsymbol{d}_1^*}, g_1^{\boldsymbol{d}_1}, \boldsymbol{h}_1 = g_1^{a_1 \boldsymbol{d}_2}, \cdots, \boldsymbol{h}_L = g_1^{a_L \boldsymbol{d}_2}).$$

`KEY QUERY`: We note that $\mathcal{B}$ does not know the full master secret key, but he knows $\boldsymbol{u}_1 = g_2^{\mu_1 \boldsymbol{b}_1^* + \mu_2 \boldsymbol{b}_2^*}$, μ_2 and $a_1, \cdots, a_L$. This allows it to produce SF keys as follows: when $\mathcal{A}$ requests a key for some pattern $\boldsymbol{P}'$, $\mathcal{B}$ chooses random values $r', t_4 \in \mathbb{Z}_p$. It sets $r = \mu_2 r'$ and forms the secret key as: $\boldsymbol{a} = (\boldsymbol{u}_1)^{(\alpha' + r' \sum_{i \in \mathcal{I}} a_i)} . g_2^{-\mu_2 r' \boldsymbol{d}_2^* + t_4 \boldsymbol{d}_4^*}$, $\boldsymbol{b}_i = \boldsymbol{u}_1^{r' a_i}$.

We obtain that $\boldsymbol{a} = g_2^{\alpha \boldsymbol{d}_1^* + r\boldsymbol{d}_1^* \sum_{i \in \mathcal{I}} a_i - r\boldsymbol{d}_2^* + (\alpha' \mu_1 + r' \mu_1 \sum_{i \in \mathcal{I}} a_i)\boldsymbol{d}_3^* + t_4 \boldsymbol{d}_4^*}$ and $\boldsymbol{b}_i = g_2^{r\boldsymbol{d}_1^* a_i + r' \mu_1 a_i \boldsymbol{d}_3^*}$. The coefficients of $\boldsymbol{d}_3^*, \boldsymbol{d}_4^*$ are uniformly random thus it is a SK key.

`CHALLENGE`: $\mathcal{A}$ submits messages $\mathsf{m}_0, \mathsf{m}_1$ and challenge pattern $\boldsymbol{P}$, $\mathcal{B}$ chooses $b \in \{0, 1\}$ and forms the challenge ciphertext as follows: $s, w, z \leftarrow \mathbb{Z}_p$,

$$c_1 = \mathsf{m}_b \cdot (e(g_1, g_2)^{\alpha \boldsymbol{d}_1 \cdot \boldsymbol{d}_1^*})^s, \;\; \boldsymbol{c}_2 = g_1^{s\boldsymbol{d}_1 + w\boldsymbol{d}_2} \cdot \boldsymbol{t}_1 \cdot g_1^{z\boldsymbol{d}_4}$$

- If $\boldsymbol{t}_1$ is equal to $g_1^{\tau_1 \boldsymbol{b}_1}$ then $\boldsymbol{c}_2 = g_1^{s\boldsymbol{d}_1 + w\boldsymbol{d}_2 + \tau_1 \boldsymbol{d}_3 + z\boldsymbol{d}_4}$ is a semi functional ciphertext with the second appearance of s randomised. In this case $\mathcal{B}$ simulates $\mathsf{Game}_{2-Q'}$.

- If $\boldsymbol{t}_1$ is equal to $g_1^{\tau_1 \boldsymbol{b}_1 + \tau_2 \boldsymbol{b}_2}$ then $\boldsymbol{c}_2 = g_1^{(s+\tau_2)\boldsymbol{d}_1 + w\boldsymbol{d}_2 + \tau_1 \boldsymbol{d}_3 + z\boldsymbol{d}_4}$ which is a semi functional ciphertext with randomised coefficients for $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$. Thus in this case $\mathcal{B}$ simulates Game_3.

Therefore, $\mathcal{B}$ can leverage $\mathcal{A}$'s non-negligible difference of advantage between these two games to achieve a non-negligible advantage against DS1.

Combining lemmas 10 and 11 we obtain lemma 9. Along with lemmas 1, 7 and 8, this completes the proof of theorem 8.

B.2 Second WIBE security proofs

We now prove the security of our pattern-hiding WIBE (section 4.2). We start by proving indistinguishability between games presented for $t = 0$, by proving following lemmas 12, 14, 15 and 16.

Lemma 12. *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_0$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $\left|\mathsf{Adv}_{\mathcal{A}}^{(0^{'})}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)\right| = \mathsf{Adv}_{\mathcal{B}_0}^{P1^b}(\lambda)$.*

Proof. In order to prove lemma 12, we construct a probabilistic machine $\mathcal{B}_0$ against Problem 1 bis by using any adversary $\mathcal{A}$ in a security game ($\mathsf{Game}_{0'}$ or Game_1) as a black box as follows:

1. $\mathcal{B}_0$ is given a Problem 1 bis instance $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{B}, \hat{\mathbb{B}}^*,\ \boldsymbol{e}_{\beta,1}, \{\boldsymbol{e}_i\ \}_{i \in [\![1,n]\!]})$.
2. $\mathcal{B}_0$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.
3. At the first step of the game, $\mathcal{B}_0$ returns $\mathsf{pk} = (\mathit{\Gamma}, p, e(g_1, g_2)^{\alpha \boldsymbol{b}_0 . \boldsymbol{b}_0^*}, g_1^{\boldsymbol{b}_0}, g_1^{\boldsymbol{b}_{4L+1}},$ $\boldsymbol{h}_1 = g_1^{\boldsymbol{b}_1}, \cdots, \boldsymbol{h}_L = g_1^{\boldsymbol{b}_L})$ to $\mathcal{A}$.
4. When a key queried is issued, $\mathcal{B}_0$ answers a correct secret key computed by using $\hat{\mathbb{B}}^*$, i.e. a normal key.
5. When $\mathcal{B}_0$ gets challenge plaintexts $\mathsf{m}_0, \mathsf{m}_1$ and pattern $\boldsymbol{P}^*$ (with $\boldsymbol{P}_1^* \neq \star$) from $\mathcal{A}$), $\mathcal{B}_0$ calculates and returns $(c_1, \boldsymbol{c}_2)$ such that $c_1 = \mathsf{m}_b {\cdot} e(g_1, g_2)^{\alpha \boldsymbol{b}_0 \cdot \boldsymbol{b}_0^* s_1}$ and $\boldsymbol{c}_2 = g_1^{s_1 \boldsymbol{b}_0} {\cdot} \boldsymbol{e}_{\beta,1} {\cdot} \prod_{i \in \bar{W}(\boldsymbol{P}^*), i \geq 2} \boldsymbol{e}_i$, where $\boldsymbol{e}_{\beta,1}$ and $\boldsymbol{e}_i$ are from the Problem 1 bis instance, $s_1 \leftarrow \mathbb{Z}_p$ and $b\,\{0,1\}$.
6. After the challenge encryption query, another key query step is executed in the same manner as step 4.
7. $\mathcal{A}$ outputs a bit $b^{'}$. If $b = b^{'}$, $\mathcal{B}_0$ outputs $\beta^{'} = 1$. Otherwise, $\mathcal{B}_0$ outputs $\beta^{'} = 0$.

Let's see that if $\beta = 0$, then the distribution of $(c_1, \boldsymbol{c}_2)$ in step 5 is the same as that in $\mathsf{Game}_{0'}$. If $\beta = 1$, the distribution of $(c_1, \boldsymbol{c}_2)$ in step 5 is the same as that in Game_1.

If $\beta = 0$,

$$\begin{aligned}\boldsymbol{c}_2 &= g_1^{s_1 \boldsymbol{b}_0} \cdot \boldsymbol{e}_{\beta,1} \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^*), i \geq 2} \boldsymbol{e}_i \\ &= g_1^{s_1 \boldsymbol{b}_0} \cdot g_1^{\omega \boldsymbol{b}_1 + \gamma \boldsymbol{b}_{4L+1}} \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^*), i \geq 2} g_1^{\omega b_i} \\ &= g_1^{s_1 \boldsymbol{b}_0 + \omega \sum_{i \in \bar{W}(\boldsymbol{P}^*)} \boldsymbol{b}_i + \gamma \boldsymbol{b}_{4L+1}}.\end{aligned}$$

This is the challenge ciphertext in Game_0.
If $\beta = 1$,

$$\begin{aligned}\boldsymbol{c}_2 &= g_1^{s_1 \boldsymbol{b}_0} \cdot \boldsymbol{e}_{\beta,1} \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^*), i \geq 2} \boldsymbol{e}_i \\ &= g_1^{s_1 \boldsymbol{b}_0} \cdot g_1^{\omega \boldsymbol{b}_1 + \sum_{l=1}^{L} z_l \boldsymbol{b}_{L+l} + \gamma \boldsymbol{b}_{4L+1}} \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^*), i \geq 2} g_1^{\omega b_i} \\ &= g_1^{s_1 \boldsymbol{b}_0 + \omega \sum_{i \in \bar{W}(\boldsymbol{P}^*)} \boldsymbol{b}_i + \sum_{l=1}^{L} z_l \boldsymbol{b}_{L+l} + \gamma \boldsymbol{b}_{4L+1}}.\end{aligned}$$

Because $(z_1, \cdots, z_L) \leftarrow \mathbb{Z}_p^L \setminus \{\boldsymbol{0}^L\}$ and γ are independently uniform, this is the challenge ciphertext in Game_1.

When $\beta = 0$, the advantage of $\mathcal{A}$ in the above game is equal to that in Game_0, i.e., $\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda)$, and is also equal to $\mathrm{Pr}_0 = \Pr\left[\mathcal{B}_0(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_0^{P1b}(1^\lambda, L)\right]$. Similarly, when $\beta = 1$, we see that the advantage of $\mathcal{A}$ in the above game is equal to $\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)$, and is also equal to $\mathrm{Pr}_1 = \Pr\left[\mathcal{B}_0(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_1^{P1b}(1^\lambda, L)\right]$. Therefore,
$\left|\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)\right| = |\mathrm{Pr}_0 - \mathrm{Pr}_1| = \mathsf{Adv}_{\mathcal{B}_0}^{P1b}(\lambda)$. This completes the proof.

To prove lemma 14, we need the following lemma from [23], that we admit.

Lemma 13. *[23] Let $C = \{(\boldsymbol{x}, \boldsymbol{v}) | \boldsymbol{x} \cdot \boldsymbol{v} \neq 0\} \subset V \times V^*$, where V is n-dimensional vector space $\mathbb{Z}_p^n$, and V^* its dual. For all $(\boldsymbol{x}, \boldsymbol{v}) \in C$, for all $(\boldsymbol{r}, \boldsymbol{w}) \in C$,*

$$\Pr\left[\boldsymbol{x}(\rho \boldsymbol{U}) = \boldsymbol{r} \wedge \boldsymbol{v}(\tau \boldsymbol{Z}) = \boldsymbol{w}\right] = 1/s,$$

where $\boldsymbol{Z} \leftarrow \mathsf{GL}(n, \mathbb{Z}_p)$, $\rho, \tau \leftarrow \mathbb{Z}_p^*$, $\boldsymbol{U} = (\boldsymbol{Z}^{-1})^\top$ and $s = \#C (= (p^n - 1)(p^n - p^{n-1}))$.

Lemma 14. *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_k$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $\left|\mathsf{Adv}_{\mathcal{A}}^{(2-(k-1))}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2-k)}(\lambda)\right| \leq \mathsf{Adv}_{\mathcal{B}_k}^{P2^b}(\lambda) + 1/p$.*

Proof. In order to prove lemma 14, we construct a probabilistic machine $\mathcal{B}_k$ against Problem 2 bis by using any adversary $\mathcal{A}$ in a security game ($\mathsf{Game}_{2-(k-1)}$ or Game_{2-k}) as a black box as follows:

1. $\mathcal{B}_k$ is given a Problem 2 bis instance $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \mathbb{B}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{e}_i\}_{i \in [\![1,n]\!]})$.
2. $\mathcal{B}_k$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.
3. At the first step of the game, $\mathcal{B}_k$ returns $\mathsf{pk} = (\mathit{\Gamma}, p, e(g_1, g_2)^{\alpha \boldsymbol{b}_0 . \boldsymbol{b}_0^*}, g_1^{\boldsymbol{b}_0}, g_1^{\boldsymbol{b}_{4L+1}}, \boldsymbol{h}_1 = g_1^{\boldsymbol{b}_1}, \cdots, \boldsymbol{h}_L = g_1^{\boldsymbol{b}_L})$ to $\mathcal{A}$.
4. When the s-th key query is issued for predicate $\boldsymbol{P}$, $\mathcal{B}_k$ answers as follows:
   - When $1 \leq s \leq k-1$, $\mathcal{B}_k$ calculates and answers by using $\hat{\mathbb{B}}^*$
   $$\mathsf{sk}_{\boldsymbol{P}} = g_2^{\alpha \boldsymbol{b}_0^* + \sum_{j \in \mathcal{I}} r_j \boldsymbol{b}_j^* + \sum_{l=1}^{L} x_l \boldsymbol{b}_{L+l}^* + \sum_{l=1}^{L} \eta_l \boldsymbol{b}_{3L+l}^*}.$$
   - When $s = k$, $\mathcal{B}_k$ calculates and answers $\mathsf{sk}_{\boldsymbol{P}}$ as follows:$\{\xi_i \leftarrow \mathbb{Z}_p\}_{i \in \mathcal{I}}$,
   $$\mathsf{sk}_{\boldsymbol{P}} = g_2^{\alpha \boldsymbol{b}_0^*} \cdot \prod_{i \in \mathcal{I}} \boldsymbol{h}_{\beta,i}^{*\xi_i},$$
   - When $q \geq k+1$, $\mathcal{B}_k$ answers a correct secret key computed by using $\mathbb{B}^*$, i.e. normal key.
5. When $\mathcal{B}_k$ gets challenge plaintexts $\mathsf{m}_0, \mathsf{m}_1$ and pattern $\boldsymbol{P}^*$ from $\mathcal{A}$, $\mathcal{B}_k$ calculates and returns $(c_1, \boldsymbol{c}_2)$ such that $c_1 = \mathsf{m}_b \cdot e(g_1, g_2)^{\alpha \boldsymbol{b}_0 \cdot \boldsymbol{b}_0^* s_1}$ and $\boldsymbol{c}_2 = g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1}} \cdot \prod_{i \in \bar{W}(\boldsymbol{P}^*)} \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ are from the Problem 2 bis instance, $s_1, s_2 \leftarrow \mathbb{Z}_p$ and $b \in \{0, 1\}$.
6. After the challenge encryption query, another key query step is executed in the same manner as step 4.
7. $\mathcal{A}$ outputs a bit $b^{'}$. If $b = b^{'}$, $\mathcal{B}_k$ outputs $\beta^{'} = 1$. Otherwise, $\mathcal{B}_k$ outputs $\beta^{'} = 0$.

Let's see that if $\beta = 0$, then the distribution of $(c_1, \boldsymbol{c}_2)$ in step 5 and $\mathsf{sk}_{\boldsymbol{P}}$ is the same as that in $\mathsf{Game}_{2-(k-1)}$ except with probability $1/p$. If $\beta = 1$, the distribution of $(c_1, \boldsymbol{c}_2)$ in step 5 and $\boldsymbol{sk}_{\boldsymbol{P}}$ is the same as that in Game_{2-k} except with probability $1/p$.
We consider the joint distribution of $\boldsymbol{c}_2$ and $\mathsf{sk}_{\boldsymbol{P}}$. Ciphertext $\boldsymbol{c}_2$ generated in step 5 is

$$\begin{aligned}
\boldsymbol{c}_2 &= g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1}} \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^*)} \boldsymbol{e}_i \\
&= g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1}} \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^*)} g_1^{\omega \boldsymbol{b}_i + \tau \sum_{j=1}^{L} z_{i,j} \boldsymbol{b}_{L+j}} \\
&= g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + \omega \sum_{i \in \bar{W}(\boldsymbol{P}^*)} \boldsymbol{b}_i + \tau \sum_{j=1}^{L} \sum_{i \in \bar{W}(\boldsymbol{P}^*)} z_{i,j} \boldsymbol{b}_{L+j}} \\
&= g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + \omega \sum_{i \in \bar{W}(\boldsymbol{P}^*)} \boldsymbol{b}_i + \sum_{j=1}^{L} \tilde{t}_j \boldsymbol{b}_{L+j}}
\end{aligned}$$

where $s_1, s_2, \omega \in \mathbb{Z}_p$, $\tilde{t}_j = \sum_{i \in \bar{W}(\boldsymbol{P}^*)} \tau z_{i,j}$ and $(\tilde{t}_1, \cdots, \tilde{t}_L) \leftarrow \mathbb{Z}_p^L \setminus \{\boldsymbol{0}\}$ are independently uniform.
If $\beta = 0$, secret key generated in case b of step 4 or 6 is

$$\mathsf{sk}_{\boldsymbol{P}} = g_2^{\alpha \boldsymbol{b}_0^*} \cdot \prod\nolimits_{i \in \mathcal{I}} \boldsymbol{h}_{\beta,i}^{*\xi_i} = g_2^{\alpha \boldsymbol{b}_0^* + \sum_{i \in \mathcal{I}} \xi_i \delta \boldsymbol{b}_i^* + \sum_{i \in \mathcal{I}} \sum_{j=1}^{L} \xi_i \delta_{i,j} \boldsymbol{b}_{3L+j}^*}$$

This is a normal secret key, thus distribution of $(c_1, \boldsymbol{c}_2), \mathsf{sk}_{\boldsymbol{P}}$ are as in $\mathsf{Game}_{2-(k-1)}$ (i.e. temporal ciphertext and normal key).
If $\beta = 1$,

$$\mathsf{sk}_{\boldsymbol{P}} = g_2^{\alpha \boldsymbol{b}_0^*} \cdot \prod_{i \in \mathcal{I}} \boldsymbol{h}_{\beta,i}^{*\xi_i}$$

$$= g_2^{\alpha \boldsymbol{b}_0^* + \sum_{i \in \mathcal{I}} \xi_i \delta \boldsymbol{b}_i^* + \sum_{i \in \mathcal{I}} \sum_{j=1}^{L} \xi_i u_{i,j} \boldsymbol{b}_{L+j}^* + \sum_{i \in \mathcal{I}} \sum_{j=1}^{L} \xi_i \delta_{i,j} \boldsymbol{b}_{3L+j}^*}$$

$$= g_2^{\alpha \boldsymbol{b}_0^* + \sum_{i \in \mathcal{I}} \xi_i \delta \boldsymbol{b}_i^* + \sum_{j=1}^{L} \tilde{x}_j \boldsymbol{b}_{L+j}^* + \sum_{i \in \mathcal{I}} \sum_{j=1}^{L} \xi_i \delta_{i,j} \boldsymbol{b}_{3L+j}^*}$$

where $\tilde{x}_j = \sum_{i \in \mathcal{I}} \xi_i u_{i,j}$ and $(\tilde{x}_1, \cdots, \tilde{x}_L) \leftarrow \mathbb{Z}_p^L \setminus \{\mathbf{0}\}$.

Since $\boldsymbol{Z} = (\boldsymbol{U}^{-1})^\top$ where $\boldsymbol{Z} = (z_{i,j})$ and $U = (u_{i,j})$, we should verify the independence of coefficient vectors $\tilde{\boldsymbol{t}} = (\tilde{t}_1, \cdots, \tilde{t}_l)$ in $\boldsymbol{c}_2$ and $\tilde{\boldsymbol{x}} = (\tilde{x}_1, \cdots, \tilde{x}_l)$ in $\mathsf{sk}_{\boldsymbol{P}}$. Notice that we can rewrite $\tilde{\boldsymbol{t}}$ and $\tilde{\boldsymbol{x}}$ respectively as $\overrightarrow{\boldsymbol{y}} \cdot \boldsymbol{U}$ and $\overrightarrow{\boldsymbol{x}} \cdot \boldsymbol{Z}$, where $\overrightarrow{\boldsymbol{y}}$ and $\overrightarrow{\boldsymbol{x}}$ are vectors such that $\overrightarrow{y}_i = \begin{cases} \xi_i & \text{if } i \in \mathcal{I} \\ 0 & \text{otherwise} \end{cases}$ for $i \in [\![1, L]\!]$ and $\overrightarrow{x}_i = \begin{cases} \tau & \text{if } i \in \bar{W}(\boldsymbol{P}^*) \\ 0 & \text{otherwise} \end{cases}$ for $i \in [\![1, L]\!]$. Since $\mathcal{I} \cap \bar{W}(\boldsymbol{P}^*) \neq \varnothing$ from condition on keys and challenge ciphertext, coefficients vectors $\tilde{\boldsymbol{t}}$ and $\tilde{\boldsymbol{x}}$ are (pairwise)-independently and uniformly distributed under the condition that $\overrightarrow{\boldsymbol{y}} \cdot \overrightarrow{\boldsymbol{x}} \neq 0$ (from lemma 13). Since $(x_1, \cdots, x_l), (t_1, \cdots, t_l) \leftarrow \mathbb{Z}_p^L$ in Game_{2-k}, the event that$(x_, \cdots, x_l) \cdot (t_1, \cdots, t_l) = 0$ occurs in the game with probability $1/p$.
Thus this is a temporal 1 secret key, and the distribution of $(c_1, \boldsymbol{c}_2), \mathsf{sk}_{\boldsymbol{P}}$ are as in Game_{2-k}, except with probability $1/p$.

When $\beta = 0$, the advantage of $\mathcal{A}$ in the above game is equal to that in $\mathsf{Game}_{2-(k-1)}$, i.e., $\mathsf{Adv}_{\mathcal{A}}^{(2-(k-1))}(\lambda)$, and is also equal to $\Pr_0 = \Pr\left[\mathcal{B}_k(1^\lambda, \varrho) \rightarrow 1 | \varrho \leftarrow \mathcal{G}_0^{P2^b}(1^\lambda, L)\right]$. Similarly, when $\beta = 1$, we see that the advantage of $\mathcal{A}$ in the above game is equal to $\mathsf{Adv}_{\mathcal{A}}^{(2-k)}(\lambda)$, and is also equal to $\Pr_1 = \Pr\left[\mathcal{B}_k(1^\lambda, \varrho) \rightarrow 1 | \varrho \leftarrow \mathcal{G}_1^{P2^b}(1^\lambda, L)\right]$. Therefore, $\left|\mathsf{Adv}_{\mathcal{A}}^{(2-(k-1))}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2-h)}(\lambda)\right| \leq |\Pr_0 - \Pr_1| + 1/p = \mathsf{Adv}_{\mathcal{B}_k}^{P2^b}(\lambda) + 1/p$. This completes the proof.

Lemma 15. *For any adversary $\mathcal{A}$,* $\mathsf{Adv}_{\mathcal{A}}^{(2-Q)}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)$.

Proof. To prove lemma 15, we will show that distribution $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, p, \hat{\mathbb{B}}, \left\{\mathsf{sk}^{(j)}\right\}_{j \in [\![1,Q]\!]}, c_1, \boldsymbol{c}_2)$ in Game_{2-Q} and that in Game_3 are equivalent. For that purpose, we define new bases $\mathbb{D}, \mathbb{D}^*$ as follows: we generate randoms $\{\xi_{i,s}\}_{i,s \in [\![1,L]\!]}$, $\{\theta_i\}_{i=1,\cdots L}$ and set,$\boldsymbol{d}_{L+i} = \boldsymbol{b}_{L+i} - \sum_{s=1}^{L} \xi_{i,s} \boldsymbol{b}_s - \theta_i \boldsymbol{b}_0$, $\boldsymbol{d}_i^* = \boldsymbol{b}_i^* + \sum_{s=1}^{L} \xi_{s,i} \boldsymbol{b}_{L+s}^*$ for $i \in [\![1, L]\!]$ and $\boldsymbol{d}_0^* = \boldsymbol{b}_0^* + \sum_{s=1}^{L} \theta_s \boldsymbol{b}_{L+s}^*$. We set

$$\mathbb{D} = (\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_L, \boldsymbol{d}_{L+1}, \cdots, \boldsymbol{d}_{2L}, \boldsymbol{b}_{2L+1}, \cdots, \boldsymbol{b}_{4L+1}),$$
$$\mathbb{D}^* = (\boldsymbol{d}_0^*, \boldsymbol{d}_1^*, \cdots, \boldsymbol{d}_L^*, \boldsymbol{b}_{L+1}^*, \cdots, \boldsymbol{b}_{2L}^*, \boldsymbol{b}_{2L+1}^*, \cdots, \boldsymbol{b}_{4L+1}^*).$$

We then easily verify that $\mathbb{D}$ and $\mathbb{D}^*$ are dual orthonormal, and are distributed the same as the original bases, $\mathbb{B}, \mathbb{B}^*$. Keys and challenge ciphertext $(\left\{\mathsf{sk}^{(j)}\right\}_{j \in [\![1,Q]\!]}, c_1, \boldsymbol{c}_2)$ in Game_{2-Q} are expressed over bases $\mathbb{B}$ and $\mathbb{B}^*$ as

$$\begin{aligned}
\mathsf{sk}^{(j)} &= g_2^{\alpha \boldsymbol{b}_0^* + \sum_{i \in \mathcal{I}} r_i^{(j)} \boldsymbol{b}_i^* + \sum_{l=1}^{L} x_l^{(j)} \boldsymbol{b}_{L+l}^* + \sum_{l=1}^{L} \eta_l^{(j)} \boldsymbol{b}_{3L+l}^*}, \\
c_1 &= \mathsf{m}_b \cdot e(g_1, g_2)^{\alpha \boldsymbol{b}_0 \cdot \boldsymbol{b}_0^* s_1}, \\
\boldsymbol{c}_2 &= g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + s_3 \sum_{i \in \bar{W}(\boldsymbol{P}^*)} \boldsymbol{b}_i + \sum_{l=1}^{L} t_l \boldsymbol{b}_{L+l}}
\end{aligned}$$

Then,

$$\begin{aligned}
\mathsf{sk}^{(j)} &= g_2^{\alpha \boldsymbol{b}_0^* + \sum_{i \in \mathcal{I}} r_i^{(j)} \boldsymbol{b}_i^* + \sum_{l=1}^{L} x_l^{(j)} \boldsymbol{b}_{L+l}^* + \sum_{l=1}^{L} \eta_l^{(j)} \boldsymbol{b}_{3L+l}^*} \\
&= g_2^{\alpha(\boldsymbol{d}_0^* - \sum_{s=1}^{L} \theta_s \boldsymbol{d}_{L+s}^*) + \sum_{i \in \mathcal{I}} r_i^{(j)} (\boldsymbol{d}_i^* - \sum_{s=1}^{L} \xi_{i,s} \boldsymbol{d}_{L+s}^*) + \sum_{l=1}^{L} x_l^{(j)} \boldsymbol{d}_{L+l}^* + \sum_{l=1}^{L} \eta_l^{(j)} \boldsymbol{d}_{3L+l}^*} \\
&= g_2^{\alpha \boldsymbol{d}_0^* + \sum_{i \in \mathcal{I}} r_i^{(j)} \boldsymbol{d}_i^* + \sum_{l=1}^{L} \tilde{x}_l^{(j)} \boldsymbol{d}_{L+l}^* + \sum_{l=1}^{L} \eta_l^{(j)} \boldsymbol{d}_{3L+l}^*}
\end{aligned}$$

where $\tilde{x}_l^{(j)} = -\alpha\theta_l - \sum_{i \in \mathcal{I}} r_i^{(j)} \xi_{i,l} + x_l^{(j)}$ for $l \in [\![1, L]\!]$, which are uniformly, independently distributed since $x_l^{(j)} \leftarrow \mathbb{Z}_p$.

$$\begin{aligned}
\boldsymbol{c}_2 &= g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + s_3 \sum_{i \in \bar{W}(\boldsymbol{P}^*)} \boldsymbol{b}_i + \sum_{l=1}^{L} t_l \boldsymbol{b}_{L+l}} \\
&= g_1^{s_1 \boldsymbol{d}_0 + s_2 \boldsymbol{d}_{4L+1} + s_3 \sum_{i \in \bar{W}(\boldsymbol{P}^*)} \boldsymbol{d}_i + \sum_{l=1}^{L} t_l (\boldsymbol{d}_{L+l} + \sum_{s=1}^{L} \xi_{l,s} \boldsymbol{d}_s + \theta_l \boldsymbol{d}_0)} \\
&= g_1^{\boldsymbol{d}_0 + (s_1 + \sum_{l=1}^{L} t_l \theta_l) + s_2 \boldsymbol{d}_{4L+1} + s_3 \sum_{i \in \bar{W}(\boldsymbol{P}^*)} \boldsymbol{d}_i + \sum_{l=1}^{L} t_l \sum_{s=1}^{L} \xi_{l,s} \boldsymbol{d}_s + \sum_{l=1}^{L} t_l \boldsymbol{d}_{L+l}} \\
&= g_1^{s_1^{'} \boldsymbol{d}_0 + s_2 \boldsymbol{d}_{4L+1} + \sum_{i=1}^{L} \tilde{s}_i \boldsymbol{d}_i + \sum_{l=1}^{L} t_l \boldsymbol{d}_{L+l}}
\end{aligned}$$

where $s_1^{'} = s_1 + \sum_{l=1}^{L} t_l \theta_l$ and $\tilde{s}_i = \begin{cases} \sum_{l=1}^{L} t_l \xi_{l,i} & \text{if } i \notin \bar{W}(\boldsymbol{P}^*) \\ \sum_{l=1}^{L} t_l \xi_{l,i} + s_3 & \text{if } i \in \bar{W}(\boldsymbol{P}^*) \end{cases}$ for $k \in [\![1, L]\!]$.

which are uniformly, independently distributed since $(t_1, \cdots, t_l) \leftarrow \mathbb{Z}_p^L \setminus \{\boldsymbol{0}\}$, $\{\xi_{t,i}\} \leftarrow \mathbb{Z}_p$.

In the light of the adversary's view, both $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{D}, \mathbb{D}^*)$ are consistent with public key $\mathsf{pk} = (\mathit{\Gamma}, e(g_1, g_2)^{\alpha \boldsymbol{b}_0 \cdot \boldsymbol{b}_0^*}, g_1^{\boldsymbol{b}_0}, g_1^{\boldsymbol{b}_{4L+1}}, \boldsymbol{h}_1 = g_1^{\boldsymbol{b}_1}, \cdots, \boldsymbol{h}_L = g_1^{\boldsymbol{b}_L})$. Therefore, $\left\{\mathsf{sk}^{(j)}\right\}_{j \in [\![1,Q]\!]}$ and $\boldsymbol{c}_2$ can be expressed as keys and ciphertext in two ways, in Game_{2-Q} over bases $(\mathbb{B}, \mathbb{B}^*)$ and in Game_3 over bases $(\mathbb{D}, \mathbb{D}^*)$. Thus, Game_{2-Q} can be conceptually changed to Game_3.

Lemma 16. *For any adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda) = 0$.*

Proof. The value of b is independent from the adversary's view in Game_3. Hence, $\mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda) = 0$.

Combining all theses proofs, we obtain that any adversary has no advantage in winning the security game. Adding to these the fact that Problem 1 bis and Problem 2 bis hold if $\mathsf{XDLin1}, \mathsf{XDLin2}$ hold, we have proven theorem 10 when $t = 0$.

Finally we prove the indistinguishability between games presented for $t = 1$, by proving following lemmas 17, 18, 19, 20, 21 and 22.

Lemma 17. *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_1$ against Problem 1, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $\left|\mathsf{Adv}_{\mathcal{A}}^{(0^{'})}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)\right| \leq \mathsf{Adv}_{\mathcal{B}_1}^{P1}(\lambda)$.*

To prove lemma 17, we construct a probabilistic machine $\mathcal{B}_1$ against Problem 1 using an adversary $\mathcal{A}$ in a security game ($\mathsf{Game}_{0'}$ or Game_1) as a black box as follows:

1. $\mathcal{B}_1$ is given a Problem 1 instance $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{B}, \hat{\mathbb{B}}^*, \boldsymbol{e}_{\beta,1}, \{\boldsymbol{e}_i\}_{i \in [\![2,n]\!]})$.
2. $\mathcal{B}_1$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.
3. At the fist step of the game, $\mathcal{B}_1$ provides $\mathcal{A}$ a public key $\mathsf{pk} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, p, e(g_1, g_2)^{\alpha \boldsymbol{b}_0 . \boldsymbol{b}_0^*}, g_1^{\boldsymbol{b}_0}, g_1^{\boldsymbol{b}_{4L+1}}, \boldsymbol{h}_1 = g_1^{\boldsymbol{b}_1}, \cdots, \boldsymbol{h}_L = g_1^{\boldsymbol{b}_L})$ of $\mathsf{Game}_{0'}$ (and Game_1).
4. When a key query is issued for a pattern $\boldsymbol{P}$, $\mathcal{B}_1$ answers normal key $\mathsf{sk}_{\boldsymbol{P}}$, that is computed using $\hat{\mathbb{B}}^*$ of the Problem 1 instance.
5. When $\mathcal{B}_1$ receives an encryption query with challenge plaintext m and patterns $\boldsymbol{P}^0, \boldsymbol{P}^1$ from $\mathcal{A}$, $\mathcal{B}_1$ computes the challenge ciphertext $(c_1, \boldsymbol{c}_2)$ s.t.,

   $$c_1 = \mathsf{m} \cdot e(g_1, g_2)^{s_1 \alpha \boldsymbol{b}_0 \boldsymbol{b}_0^*} \;\; \boldsymbol{c}_2 = g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1}} \cdot \boldsymbol{e}_{\beta,1} \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^b) \setminus \{1\}} \boldsymbol{e}_i,$$

   where $s_1, s_2 \leftarrow \mathbb{Z}_p$, $b \leftarrow \{0,1\}$ and $\{\boldsymbol{b}_i\}_{i=0,4L+1}, \boldsymbol{e}_{\beta,1}, \{\boldsymbol{e}_i\}_{i=2,\cdots,L}$ is part of the Problem 1 instance.
6. When a key query is issued by $\mathcal{A}$ after the encryption query, $\mathcal{B}_1$ executes the same procedure as that of step 4.
7. $\mathcal{A}$ finally outputs bit $b^{'}$. If $b = b^{'}$, $\mathcal{B}_1$ outputs $\beta^{'} = 1$. Otherwise, $\mathcal{B}_1$ outputs $\beta^{'} = 0$.

Now let's see that the distribution of the view of adversary $\mathcal{A}$ in the abovementioned game simulated by $\mathcal{B}_1$ given a Problem 1 instance with $\beta \in \{0,1\}$ is the same as that in $\mathsf{Game}_{0'}$ (resp. Game_1) if $\beta = 0$ (resp. $\beta = 1$).

We will consider the distribution of $\boldsymbol{c}_2$. When $\beta = 0$, $\boldsymbol{c}_2$ generated in step 5 is

$$\boldsymbol{c}_2 = g_1^{s_1\boldsymbol{b}_0+s_2\boldsymbol{b}_{4L+1}} \cdot \boldsymbol{e}_{\beta,1} \cdot \prod_{i \in \bar{W}(\boldsymbol{P}^b)\setminus\{1\}} \boldsymbol{e}_i$$

$$= g_1^{s_1\boldsymbol{b}_0+s_2\boldsymbol{b}_{4L+1}+\omega\boldsymbol{b}_1+\gamma\boldsymbol{b}_{4L+1}+\omega\sum_{i \in \bar{W}(\boldsymbol{P}^b)\setminus\{1\}} \boldsymbol{b}_i}$$

$$= g_1^{s_1\boldsymbol{b}_0+s_3\sum_{i \in \bar{W}(\boldsymbol{P}^b)}+s_2^{'}\boldsymbol{b}_{4L+1}}$$

where $s_3 = \omega, s_2^{'} = s_2 + \gamma, s_1 \in \mathbb{Z}_p$ are uniformly and independently distributed.
When $\beta = 1$, $\boldsymbol{c}_2$ generated in step 5 is

$$\boldsymbol{c}_2 = g_1^{s_1\boldsymbol{b}_0+s_2\boldsymbol{b}_{4L+1}} \cdot \boldsymbol{e}_{\beta,1} \cdot \prod_{i \in \bar{W}(\boldsymbol{P}^b)\setminus\{1\}} \boldsymbol{e}_i$$

$$= g_1^{s_1\boldsymbol{b}_0+s_2\boldsymbol{b}_{4L+1}+\omega\boldsymbol{b}_1+z\boldsymbol{b}_{L+1}+\gamma\boldsymbol{b}_{4L+1}+\omega\sum_{i \in \bar{W}(\boldsymbol{P}^b)\setminus\{1\}} \boldsymbol{b}_i}$$

$$= g_1^{s_1\boldsymbol{b}_0+s_3\sum_{i \in \bar{W}(\boldsymbol{P}^b)}+t\boldsymbol{b}_{L+1}+s_2^{'}\boldsymbol{b}_{4L+1}}$$

where $t = z, s_3 = \omega, s_2^{'} = s_2 + \gamma, s_1 \in \mathbb{Z}_p$ are uniformly and independently distributed.
Therefore, the above $c_1, \boldsymbol{c}_2$ give a challenge ciphertext in $\mathsf{Game}_{0'}$ when $\beta = 0$ and that in Game_1 when $\beta = 1$. Thus,

$$\left|\mathsf{Adv}_{\mathcal{A}}^{(0^{'})}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)\right|$$
$$= \left|\Pr\left[\mathcal{B}_1(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_0^{P1}(1^\lambda, L)\right] - \Pr\left[\mathcal{B}_1(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_1^{P1}(1^\lambda, L)\right]\right|$$
$$\leq \mathsf{Adv}_{\mathcal{B}_1}^{P1}(\lambda).$$

This complete the proof of lemma 17.

Lemma 18. *For any adversary $\mathcal{A}$,*

$$\left|\mathsf{Adv}_{\mathcal{A}}^{(2-(h-1)-4)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2-h-1)}(\lambda)\right| \leq \epsilon,$$

for $\epsilon = 4/p$ when $h = 1$ and $\epsilon = 3/p$ when $h \geq 2$.

We start with the case $h = 1$, i.e. the proof for $\left|\mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2-1-1)}(\lambda)\right| \leq 4/p$.

We define an intermediate game, $\mathsf{Game}_{1'}$, and will show the equivalence of the distribution of the views of $\mathcal{A}$ in Game_1 and that in $\mathsf{Game}_{1'}$ and those in Game_{2-1-1} and in $\mathsf{Game}_{1'}$.

$\mathsf{Game}_{1'}$:$\mathsf{Game}_{1'}$ is the same as Game_1 except that the $\boldsymbol{c}_2$ of the challenge ciphertext for (challenge plaintext m and) patterns $\boldsymbol{P}^0, \boldsymbol{P}^1$ is:

$$\boldsymbol{c}_2 = g_1^{s_1\boldsymbol{b}_0+s_3\sum_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i+\sum_{i=1}^{2L} r_i\boldsymbol{b}_{L+i}+s_2\boldsymbol{b}_{4L+1}}$$

where $r_i \leftarrow \mathbb{Z}_p$ for $i \in [\![1, 2L]\!]$, $\boldsymbol{r} = (r_1, \cdots, r_{2L}) \neq \mathbf{0}^{2L}$, and all the other variables are generated as in Game_1.

Let's see that the distribution of $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, p, \mathbb{B}^*, \hat{\mathbb{B}}, \left\{\mathsf{sk}^{(j)}\right\}_{j\in[\![1,Q]\!]}, c_1,$ $\boldsymbol{c}_2)$ in Game_1 and that in $\mathsf{Game}_{1'}$ are equivalent except with negligible probability.

We will consider the distribution in Game_1. We define new dual orthonormal bases $(\mathbb{D}, \mathbb{D}^*)$ below. Pick $\boldsymbol{F} \leftarrow \mathsf{GL}(2L, \mathbb{Z}_p)$, and set

$$\begin{pmatrix} \boldsymbol{d}_{L+1} \\ \vdots \\ \boldsymbol{d}_{3L} \end{pmatrix} = \boldsymbol{F}^{-1} \cdot \begin{pmatrix} \boldsymbol{b}_{L+1} \\ \vdots \\ \boldsymbol{b}_{3L} \end{pmatrix}, \quad \begin{pmatrix} \boldsymbol{d}^*_{L+1} \\ \vdots \\ \boldsymbol{d}^*_{3L} \end{pmatrix} = \boldsymbol{F}^{\top} \cdot \begin{pmatrix} \boldsymbol{b}^*_{L+1} \\ \vdots \\ \boldsymbol{b}^*_{3L} \end{pmatrix},$$

$\mathbb{D} = (\boldsymbol{b}_0, \cdots, \boldsymbol{b}_L, \boldsymbol{d}_{L+1}, \cdots, \boldsymbol{d}_{3L}, \boldsymbol{b}_{3L+1}, \cdots, \boldsymbol{b}_{4L+1})$ and $\mathbb{D}^* = (\boldsymbol{b}^*_0, \cdots, \boldsymbol{b}^*_L, \boldsymbol{d}^*_{L+1},$ $\cdots, \boldsymbol{d}^*_{3L}, \boldsymbol{b}^*_{3L+1}, \cdots, \boldsymbol{b}^*_{4L+1})$. Then, $\mathbb{D}, \mathbb{D}^*$ are dual orthonormal bases. Notice that then $\boldsymbol{b}_{L+1}$ is equal to $\boldsymbol{F} \cdot \begin{pmatrix} \boldsymbol{d}_{L+1} \\ \vdots \\ \boldsymbol{d}_{3L} \end{pmatrix}$, thus can be written as $\boldsymbol{b}_{L+1} = f_{1,1}\boldsymbol{d}_{L+1} + f_{1,2}\boldsymbol{d}_{L+2} + \cdots + f_{1,2L}\boldsymbol{d}_{3L}$, with

$$\boldsymbol{F} = \begin{pmatrix} f_{1,1} & f_{1,2}, & \cdots & f_{1,2L} \\ f_{2,1} & f_{2,2}, & \cdots & f_{2,2L} \\ \vdots & & & \\ f_{2L,1} & f_{2L,2}, & \cdots & f_{2L,2L} \end{pmatrix}.$$

Challenge ciphertext $\boldsymbol{c}_2$ is expressed as

$$g_1^{s_1\boldsymbol{b}_0 + s_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i + t\boldsymbol{b}_{L+1} + s_2\boldsymbol{b}_{4L+1}}$$

$$= g_1^{s_1\boldsymbol{d}_0 + s_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{d}_i + t(f_{1,1}\boldsymbol{d}_{L+1} + f_{1,2}\boldsymbol{d}_{L+2} + \cdots + f_{1,2L}\boldsymbol{d}_{3L}) + s_2\boldsymbol{d}_{4L+1}}$$

$$= g_1^{s_1\boldsymbol{d}_0 + \boldsymbol{s}_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{d}_i + \sum_{i=1}^{2L} r_i\boldsymbol{d}_{L+i} + s_2\boldsymbol{d}_{4L+1}}$$

where $s_1, s_2, s_3 \leftarrow \mathbb{Z}_p$ and $\boldsymbol{r} = (r_i = tf_{1,i})_{i\in[\![1,2L]\!]}$. Vector $\boldsymbol{r}$ is uniformly distributed in $\mathbb{Z}_p^{2L} \setminus \left\{\mathbf{0}^{2L}\right\}$ except for probability $1/p$ and independent of all the other variables.

In Game_1, $\mathsf{sk}_{\boldsymbol{P}}$ is $g_2^{\alpha\boldsymbol{b}^*_0 + \sum_{j\in\mathcal{I}} r_j\boldsymbol{b}^*_j + \sum_{l=1}^{L} \eta_l \cdot \boldsymbol{b}^*_{3L+l}} = g_2^{\alpha\boldsymbol{d}^*_0 + \sum_{j\in\mathcal{I}} r_j\boldsymbol{d}^*_j + \sum_{l=1}^{L} \eta_l \cdot \boldsymbol{d}^*_{3L+l}}$, where $r, \{\eta_l\}_{l\in[\![1,L]\!]} \leftarrow \mathbb{Z}_p$, for every queried key.

In the light of the adversary's view, $(\mathbb{D}, \mathbb{D}^*)$ is consistent with public key $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, p, \boldsymbol{b}_0, \boldsymbol{b}_{4L+1}, g_1^{\boldsymbol{b}_1}, \cdots, g_1^{\boldsymbol{b}_L})$. Moreover, the challenge ciphertext in Game_1 can be conceptually changed to that in $\mathsf{Game}_{1'}$ except with probability $1/p$.

Let's see that the distribution of $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, p, \mathbb{B}^*, \hat{\mathbb{B}}, \left\{\mathsf{sk}^{(j)}\right\}_{j \in [\![1,Q]\!]},$ $c_1, \boldsymbol{c}_2)$ in Game_{2-1-1} and that in $\mathsf{Game}_{1'}$ are equivalent except with probability $3/p$.

We will consider the distribution in Game_{2-1-1}. We define new dual orthonormal bases $(\mathbb{D}, \mathbb{D}^*)$ as above. Challenge ciphertext $\boldsymbol{c}_2$ is expressed as

$$g_1^{s_1 \boldsymbol{b}_0 + s_3 \sum_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i + t \sum_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_{L+i} + u \sum_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i} + s_2 \boldsymbol{b}_{4L+1}}$$

$$= g_1^{s_1 \boldsymbol{d}_0 + s_3 \sum_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{d}_i + t \sum_{i \in \bar{W}(\boldsymbol{P}^b)} (\sum_{j=1}^{2N} f_{i,j} \boldsymbol{d}_{L+j}) + u \sum_{i \in \bar{W}(\boldsymbol{P}^0)} (\sum_{j=1}^{2L} f_{i,j} \boldsymbol{d}_{L+j})}$$

$$\cdot g_1^{\tilde{u} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} (\sum_{j=1}^{2L} f_{i,j} \boldsymbol{d}_{L+j}) + s_2 \boldsymbol{d}_{4L+1}}$$

$$= g_1^{s_1 \boldsymbol{d}_0 + \boldsymbol{s}_3 \sum_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{d}_i + \sum_{i=1}^{2L} r_i \boldsymbol{d}_{L+i} + s_2 \boldsymbol{d}_{4L+1}}$$

where $s_1, s_2, s_3 \leftarrow \mathbb{Z}_p$ and vector $\boldsymbol{r}$ such that for $i \in [\![1, 2L]\!]$, $r_i = t \sum_{j \in \bar{W}(\boldsymbol{P}^b)} f_{j,i}$ $+ u \sum_{j \in \bar{W}(\boldsymbol{P}^0)} f_{j,i} + \tilde{u} \sum_{j \in \bar{W}(\boldsymbol{P}^1)} f_{j,i}$.
Vector $\boldsymbol{r} \neq \boldsymbol{0}^{2L}$ except with probability $3/p$, is uniformly distributed in $\mathbb{Z}_p^{2L}$ $\setminus \left\{\boldsymbol{0}^{2L}\right\}$, and independent of all the other variables. For the queried keys, the same as above holds also in Game_{2-1-1}.

In the light of the adversary's view, $(\mathbb{D}, \mathbb{D}^*)$ is consistent with public key $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T,\ g_1, g_2, e, p, \boldsymbol{b}_0, \boldsymbol{b}_{4L+1}, g_1^{\boldsymbol{b}_1}, \cdots, g_1^{\boldsymbol{b}_L})$. Moreover, the challenge ciphertext in Game_{2-1-1} can be conceptually changed to that in $\mathsf{Game}_{1'}$ except with probability $3/p$.

This completes the proof when $h = 1$.

Now $h \geq 2$, i.e. proof for $\left|\mathsf{Adv}_{\mathcal{A}}^{(2-(h-1)-4)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2-h-1)}(\lambda)\right| \leq 3/p$.

We define an intermediate game, $\mathsf{Game}_{2-(h-1)-4'}$, and will show the equivalence of the distribution of the views of $\mathcal{A}$ in $\mathsf{Game}_{2-(h-1)-4}$ and that in $\mathsf{Game}_{2-(h-1)-4'}$ and those in Game_{2-h-1} and in $\mathsf{Game}_{2-(h-1)-4'}$.

$\mathsf{Game}_{2-(h-1)-4'}$: $\mathsf{Game}_{2-(h-1)-4'}$ is the same as $\mathsf{Game}_{2-(h-1)-4}$ except that the $\boldsymbol{c}_2$ of the challenge ciphertext for (challenge plaintext m and) patterns $\boldsymbol{P}^0, \boldsymbol{P}^1$ is:

$$\boldsymbol{c}_2 = g_1^{s_1\boldsymbol{b}_0+s_3\sum_{i\in\bar{W}(\boldsymbol{P}^b)}\boldsymbol{b}_i+\sum_{i=1}^{L} r_i\boldsymbol{b}_{L+i}+u\sum_{i\in\bar{W}(\boldsymbol{P}^0)}\boldsymbol{b}_{2L+i}+\tilde{u}\sum_{i\in\bar{W}(\boldsymbol{P}^1)}\boldsymbol{b}_{2L+i}+s_2\boldsymbol{b}_{4L+1}}$$

where $r_i \leftarrow \mathbb{Z}_p$ for $i \in [\![1, L]\!]$, $\boldsymbol{r} = (r_1, \cdots, r_L) \neq \boldsymbol{0}^L$, and all the other variables are generated as in $\mathsf{Game}_{2-(h-1)-4}$.

Let's see that the distribution of $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, p, \mathbb{B}^*, \hat{\mathbb{B}}, \left\{\mathsf{sk}^{(j)}\right\}_{j\in[\![1,Q]\!]},$ $c_1, \boldsymbol{c}_2)$ in $\mathsf{Game}_{2-(h-1)-4}$ and that in $\mathsf{Game}_{2-(h-1)-4'}$ are equivalent except with probability $2/p$.

We will consider the distribution in $\mathsf{Game}_{2-(h-1)-4}$. We define new dual orthonormal bases $(\mathbb{D}, \mathbb{D}^*)$ below.

We generate $\boldsymbol{F} \leftarrow \mathsf{GL}(L, \mathbb{Z}_p)$, and set

$$\begin{pmatrix}\boldsymbol{d}_{L+1}\\ \vdots \\ \boldsymbol{d}_{2L}\end{pmatrix} = \boldsymbol{F}^{-1}\begin{pmatrix}\boldsymbol{b}_{L+1}\\ \vdots \\ \boldsymbol{b}_{2L}\end{pmatrix} \quad \begin{pmatrix}\boldsymbol{d}^*_{L+1}\\ \vdots \\ \boldsymbol{d}^*_{2L}\end{pmatrix} = \boldsymbol{F}^{\top}\begin{pmatrix}\boldsymbol{b}^*_{L+1}\\ \vdots \\ \boldsymbol{b}^*_{2L}\end{pmatrix}$$

$\mathbb{D} = (\boldsymbol{b}_0, \cdots, \boldsymbol{b}_L, \boldsymbol{d}_{L+1}, \cdots, \boldsymbol{d}_{2L}, \boldsymbol{b}_{2L+1}, \cdots, \boldsymbol{b}_{4L+1})$, and $\mathbb{D}^* = (\boldsymbol{b}^*_0, \cdots, \boldsymbol{b}^*_L, \boldsymbol{d}^*_{L+1},$ $\cdots, \boldsymbol{d}^*_{2L}, \boldsymbol{b}^*_{2L+1}, \cdots, \boldsymbol{b}^*_{4L+1})$. Then $\mathbb{D}$ and $\mathbb{D}^*$ are dual orthonormal bases. Challenge ciphertext $\boldsymbol{c}_2$ is expressed as

$$g_1^{s_1\boldsymbol{b}_0+s_3\sum_{i\in\bar{W}(\boldsymbol{P}^b)}\boldsymbol{b}_i+t\sum_{i\in\bar{W}(\boldsymbol{P}^0)}\boldsymbol{b}_{L+i}+\tilde{t}\sum_{i\in\bar{W}(\boldsymbol{P}^1)}\boldsymbol{b}_{L+i}+u\sum_{i\in\bar{W}(\boldsymbol{P}^0)}\boldsymbol{b}_{2L+i}+\tilde{u}\sum_{i\in\bar{W}(\boldsymbol{P}^0)}\boldsymbol{b}_{2L+i}+s_2\boldsymbol{b}_{4L+1}}$$

$$= g_1^{s_1\boldsymbol{d}_0+s_3\sum_{i\in\bar{W}(\boldsymbol{P}^b)}\boldsymbol{d}_i+t\sum_{i\in\bar{W}(\boldsymbol{P}^0)}(\sum_{j=1}^{L} f_{i,j}\boldsymbol{d}_{L+j})+\tilde{t}\sum_{i\in\bar{W}(\boldsymbol{P}^1)}(\sum_{j=1}^{L} f_{i,j}\boldsymbol{d}_{L+j})+u\sum_{i\in\bar{W}(\boldsymbol{P}^0)}\boldsymbol{d}_{2L+i}}$$
$$\cdot g_1^{\tilde{u}\sum_{i\in\bar{W}(\boldsymbol{P}^0)}\boldsymbol{d}_{2L+i}+s_2\boldsymbol{d}_{4L+1}}$$

$$= g_1^{s_1\boldsymbol{d}_0+\boldsymbol{s}_3\sum_{i\in\bar{W}(\boldsymbol{P}^b)}\boldsymbol{d}_i+\sum_{i=1}^{L} r_i\boldsymbol{d}_{L+i}+u\sum_{i\in\bar{W}(\boldsymbol{P}^0)}\boldsymbol{d}_{2L+i}+\tilde{u}\sum_{i\in\bar{W}(\boldsymbol{P}^0)}\boldsymbol{d}_{2L+i}+s_2\boldsymbol{d}_{4L+1}}$$

where $s_1, s_2, s_3, u, \tilde{u} \leftarrow \mathbb{Z}_p$ and $\boldsymbol{r}$ is defined such that for $i \in [\![1, L]\!]$, $r_i = t\sum_{j\in\bar{W}(\boldsymbol{P}^0)} f_{j,i} + \tilde{t}\sum_{j\in\bar{W}(\boldsymbol{P}^1)} f_{j,i}$. Thus $\boldsymbol{r} \neq \boldsymbol{0}^L$ except with probability $2/p$, is uniformly distributed and independent of all the other variables.

When $1 \le j \le h-1$, the j-th queried key $\mathsf{sk}_{\boldsymbol{P}^{(j)}}$ is g_2 with exponent $\alpha\boldsymbol{b}^*_0 + \sum_{j\in\mathcal{I}} r_j\boldsymbol{b}^*_j + \sum_{j\in\mathcal{I}} x_j\boldsymbol{b}_{2L+j} + \sum_{l=1}^{L}\eta_l\cdot\boldsymbol{b}^*_{3L+l} = \alpha\boldsymbol{d}^*_0 + \sum_{j\in\mathcal{I}} r_j\boldsymbol{d}^*_j + \sum_{j\in\mathcal{I}} x_j\boldsymbol{d}_{2L+j} + \sum_{l=1}^{L}\eta_l\cdot\boldsymbol{d}^*_{3L+l}$, where $\{x_j, r_j\}_{i,j\in[\![1,L]\!]}, \{\eta_l\}_{l\in[\![1,L]\!]} \leftarrow \mathbb{Z}_p$. When $h \le j \le Q$, the j-th queried key $\mathsf{sk}_{\boldsymbol{P}^{(j)}}$ is g_2 with exponent $\alpha\boldsymbol{b}^*_0 + \sum_{j\in\mathcal{I}} r_j\boldsymbol{b}^*_j + \sum_{l=1}^{L}\eta_l\cdot\boldsymbol{b}^*_{3L+l} =$

$\alpha \boldsymbol{d}_0^* + \sum_{j\in\mathcal{I}} r_j \boldsymbol{d}_j^* + \sum_{l=1}^{L} \eta_l \cdot \boldsymbol{d}_{3L+l}^*$, where$\{r_j\}_{j\in[\![1,L]\!]}, \{\eta_l\}_{l\in[\![1,L]\!]} \leftarrow \mathbb{Z}_p$.

In the light of the adversary's view, $(\mathbb{D},\mathbb{D}^*)$ is consistent with public key $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, p, \boldsymbol{b}_0, \boldsymbol{b}_{4L+1}, g_1^{\boldsymbol{b}_1}, \cdots, g_1^{\boldsymbol{b}_L})$. Moreover, the challenge ciphertext in $\mathsf{Game}_{2-(h-1)-4}$ can be conceptually changed to that in $\mathsf{Game}_{2-(h-1)-4'}$ except with probability $2/p$.

Let us see that the distribution of $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, p, \mathbb{B}^*, \hat{\mathbb{B}}, \left\{\mathsf{sk}^{(j)}\right\}_{j\in[\![1,Q]\!]}, c_1, \boldsymbol{c}_2)$ in Game_{2-h-1} and that in $\mathsf{Game}_{2-(h-1)-4'}$ are equivalent except with probability $1/p$.

We will consider the distribution in Game_{2-h-1}. We define new dual orthonormal bases $(\mathbb{D},\mathbb{D}^*)$ as above. Challenge ciphertext $\boldsymbol{c}_2$ is expressed as

$$g_1^{s_1\boldsymbol{b}_0 + s_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i + t \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_{L+i} + u \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + s_2 \boldsymbol{b}_{4L+1}}$$

$$= g_1^{s_1\boldsymbol{d}_0 + s_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{d}_i + t \sum_{i\in\bar{W}(\boldsymbol{P}^b)} (\sum_{j=1}^{L} \boldsymbol{d}_{L+j}) + u \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{d}_{2L+i} + \tilde{u} \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{d}_{2L+i} + s_2 \boldsymbol{d}_{4L+1}}$$

$$= g_1^{s_1\boldsymbol{d}_0 + \boldsymbol{s}_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{d}_i + \sum_{i=1}^{L} r_i \boldsymbol{d}_{L+i} + \sum_{i=1}^{L} \boldsymbol{d}_{L+i} + u \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{d}_{2L+i} + \tilde{u} \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{d}_{2L+i} + s_2 \boldsymbol{d}_{4L+1}}$$

where $s_1, s_2, s_3, u, \tilde{u} \leftarrow \mathbb{Z}_p$ and vector $\boldsymbol{r}$ such that for $i \in [\![1,L]\!]$, $r_i = t\sum_{j\in\bar{W}(\boldsymbol{P}^b)} f_{j,i}$. Vector $\boldsymbol{r} \neq 0$ except with probability $1/p$, then is uniformly distributed in $\mathbb{Z}_p^L \setminus \{\boldsymbol{0}^L\}$, and independent of all the other variables.
For the queried keys, the same as above holds also in Game_{2-h-1}.

In the light of the adversary's view, $(\mathbb{D},\mathbb{D}^*)$ is consistent with public key $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, p, \boldsymbol{b}_0, \boldsymbol{b}_{4L+1}, g_1^{\boldsymbol{b}_1}, \cdots, g_1^{\boldsymbol{b}_L})$. Moreover, the challenge ciphertext in Game_{2-h-1} can be conceptually changed to that in $\mathsf{Game}_{2-(h-1)-4'}$ except with probability $1/p$.

This completes the proof when $h \geq 2$, and thus also the proof of lemma 18.

Lemma 19. *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_{2-1}$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $\left|\mathsf{Adv}_{\mathcal{A}}^{2-h-1}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{2-h-2}(\lambda)\right| \leq \mathsf{Adv}_{\mathcal{B}_{2-h-1}}^{P2}(\lambda)$, where $\mathcal{B}_{2-h-1}(.)$ $= B_{2-1}(h,.)$.*

Proof. We construct a probabilistic adversary $\mathcal{B}_{2-1}$ against Problem 2 using an adversary $\mathcal{A}$ in a security game (Game_{2-h-1} or Game_{2-h-2}) as a black box as follows:

1. $\mathcal{B}_{2-1}$ is given an integer h and $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \mathbb{B}^*, \left\{\boldsymbol{h}_{\beta,i}^*, \boldsymbol{e}_i\right\}_{i\in[\![1,L]\!]})$.

2. $\mathcal{B}_{2-1}$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.
3. At the first step of the game, $\mathcal{B}_{2-1}$ provides $\mathcal{A}$ elements for public key $1^\lambda, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}'$ of $\mathsf{Game}_{2-(h-1)-4}$ (and Game_{2-h-1}), where $\hat{\mathbb{B}}' = (\boldsymbol{b}_0, \cdots, \boldsymbol{b}_L, \boldsymbol{b}_{4L+1})$ is obtained from the Problem 2 instance. $\mathcal{A}$ can now creates pk.
4. When the ι-th key query is issued for a pattern $\boldsymbol{P}$, $\mathcal{B}_{2-1}$ answers as follows:
   - When $1 \le \iota \le h-1$, $\mathcal{B}_{2-1}$ answers keys of temporal 2 form, that are computed using $\mathbb{B}^*$ of the Problem 2 instance.
   - When $\iota = h$, $\mathcal{B}_{2-1}$ calculates $\mathsf{sk}_{\boldsymbol{P}}$ using $\left\{\boldsymbol{h}^*_{\beta,i}\right\}_{i \in [\![1,L]\!]}, \{\boldsymbol{b}^*_i\}_{i=0,3L+1,\cdots,4L}$ of the Problem 2 instance as follows: $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_L) \leftarrow \mathbb{Z}_p^L, \xi_i \leftarrow \mathbb{Z}_p$ for $i \in [\![1, L]\!]$
   $$\mathsf{sk}_{\boldsymbol{P}} = g_2^{\alpha \boldsymbol{b}^*_0} \cdot \prod_{i \in \mathcal{I}} \boldsymbol{h}^{*\xi_i}_{\beta,i} \cdot g_2^{\sum_{i \in [\![1,L]\!]} \eta_i \boldsymbol{b}^*_{3L+i}}$$
   - When $\iota \ge h+1$, $\mathcal{B}_{2-1}$ answers normal keys using $\mathbb{B}^*$ of the Problem 2 instance.
5. When $\mathcal{B}_{2-1}$ receives an encryption query with challenge plaintext m and patterns $\boldsymbol{P}^0, \boldsymbol{P}^1$ from $\mathcal{A}$, $\mathcal{B}_{2-1}$ computes challenge ciphertext $(c_1, \boldsymbol{c}_2)$ s.t.
   $$c_1 = \mathsf{m} \cdot e(g_1, g_2)^{s_1 \boldsymbol{b}_0 \cdot \boldsymbol{b}^*_0}$$
   $$\boldsymbol{c}_2 = g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1}} \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{e}_i \cdot g_1^{u \sum_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i}}$$
   where $s_1, s_2, u, \tilde{u} \leftarrow \mathbb{Z}_p, b \leftarrow \{0,1\}$ and $\{\boldsymbol{b}_i\}_{i=0,2L+1,\cdots,3L,4L+1}, \{\boldsymbol{e}_i\}_{i \in [\![1,L]\!]}$ is a part of the Problem 2 instance.
6. When a key query is issued by $\mathcal{A}$, $\mathcal{B}_{2-1}$ executes the same as in step 4.
7. $\mathcal{A}$ outputs bit b'. If $b = b'$, $\mathcal{B}_{2-1}$ outputs $\beta' = 1$. Otherwise, $\mathcal{B}_{2-1}$ outputs $\beta' = 0$.

Now let us see that if $\beta = 0$, then the distribution of the view of adversary $\mathcal{A}$ in the above mentioned game simulated by $\mathcal{B}_{2-1}$ is the same that in Game_{2-h-1}, and that if $\beta = 1$ it is the same that in Game_{2-h-2}. Ciphertext $\boldsymbol{c}_2$ is

$$\begin{aligned}
&g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1}} \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{e}_i \cdot g_1^{u \sum_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i}} \\
&= g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1}} \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^b)} g_1^{\omega \boldsymbol{b}_i + \sigma \boldsymbol{b}_{L+i}} \cdot g_1^{u \sum_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i}} \\
&= g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + \omega \sum_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i + \sigma \sum_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_{L+i} + u \sum_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i}}
\end{aligned}$$

where $s_1, s_2, \omega, \sigma, u, \tilde{u} \in \mathbb{Z}_p$ are uniformly distributed.
Now let us see the value of $\mathsf{sk}_{\boldsymbol{P}}$. When $\beta = 0$, $\mathsf{sk}_{\boldsymbol{P}}$ in case (b) of step 4 or 6 is

$$\begin{aligned}
g_2^{\alpha \boldsymbol{b}^*_0} \cdot \prod_{i \in \mathcal{I}} \boldsymbol{h}^{*\xi_i}_{\beta,i} \cdot g_2^{\sum_{i \in [\![1,L]\!]} \eta_i \boldsymbol{b}^*_{3L+i}} &= g_2^{\alpha \boldsymbol{b}^*_0} \cdot \prod_{i \in \mathcal{I}} g_2^{\delta \xi_i \boldsymbol{b}^*_i + \xi_i \delta_0 \boldsymbol{b}^*_{3L+i}} \cdot g_2^{\sum_{i \in [\![1,L]\!]} \eta_i \boldsymbol{b}^*_{3L+i}} \\
&= g_2^{\alpha \boldsymbol{b}^*_0 + \sum_{j \in \mathcal{I}} \delta \xi_i \boldsymbol{b}^*_j + \sum_{i \in [\![1,L]\!]} \phi_i \boldsymbol{b}^*_{3L+i}}
\end{aligned}$$

where α, δ are uniformly and independently distributed and $\phi_i = \xi_i \delta_0 + \eta_i$ if $i \in \mathcal{I}$ and $\phi_i = \eta_i$ otherwise. Therefore, generated $\boldsymbol{c}_2, \mathsf{sk}_{\boldsymbol{P}}$ have the same joint distribution as in Game_{2-h-1}. When $\beta = 1$, $\mathsf{sk}_{\boldsymbol{P}}$ in case (b) of step 4 or 6 is

$$g_2^{\alpha \boldsymbol{b}_0^*} \cdot \prod_{i \in \mathcal{I}} \boldsymbol{h}_{\beta,i}^{*\xi_i} \cdot g_2^{\sum_{i \in [\![1,L]\!]} \eta_i \boldsymbol{b}_{3L+i}^*}$$

$$= g_2^{\alpha \boldsymbol{b}_0^*} \cdot \prod_{i \in \mathcal{I}} g_2^{\xi_i \delta \boldsymbol{b}_i^* + \xi_i \tau \boldsymbol{b}_{L+i} + \xi_i \delta_0 \boldsymbol{b}_{3L+i}^*} \cdot g_2^{\sum_{i \in [\![1,L]\!]} \eta_i \boldsymbol{b}_{3L+i}^*}$$

$$= g_2^{\alpha \boldsymbol{b}_0^* + \sum_{i \in \mathcal{I}} \delta \xi_i \boldsymbol{b}_j^* + \sum_{i \in \mathcal{I}} \tau \xi_i \boldsymbol{b}_{L+i}^* + \sum_{i \in [\![1,L]\!]} \phi_i \boldsymbol{b}_{3L+i}^*}$$

where α, δ, τ are uniformly and independently distributed and $\phi_i = \xi_i \delta_0 + \eta_i$ if $i \in \mathcal{I}$ and $x_i = \eta_i$ otherwise. Therefore, generated $\boldsymbol{c}_2, \mathsf{sk}_{\boldsymbol{P}}$ have the same joint distribution as in Game_{2-h-2}. Thus $\left|\mathsf{Adv}_{\mathcal{A}}^{2-h-1}(\lambda) - \mathcal{A}^{2-h-2}(\lambda)\right| \leq \mathsf{Adv}_{\mathcal{B}_{2-h-1}}^{P2}(\lambda)$.

Lemma 20. *For any adversary $\mathcal{A}$,* $\left|\mathsf{Adv}_{\mathcal{A}}^{(2-h-2)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2-h-3)}\right| \leq \frac{4}{p^{|\mathcal{I}|}} + 5/p$.

Proof. We will show that distribution $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{B}, \mathbb{B}^*, \left\{\mathsf{sk}^{(j)}\right.\ \}_{j \in [\![1,Q]\!]}$ $, c_1, \boldsymbol{c}_2)$ in Game_{2-h-2} and that in Game_{2-h-3} are equivalent. For that purpose, we define an intermediate game: $\mathsf{Game}_{2-h-2'}$, is the same as Game_{2-h-2} except that $\boldsymbol{c}_2$ of the challenge ciphertext for challenge plaintext m and patterns $\boldsymbol{P}^0, \boldsymbol{P}^1$ is:

$$\boldsymbol{c}_2 = g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + \sum_{i \in \bar{W}(\boldsymbol{P}^b)} \tilde{s}_i \boldsymbol{b}_i + \sum_{i=1}^{l} \nu_i \boldsymbol{b}_{L+i}} \cdot g_1^{u \sum_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i}}$$

where $\{\tilde{s}_i \in \mathbb{Z}_p\}_{i \in [\![1,L]\!]}, \boldsymbol{\nu} \leftarrow \mathbb{Z}_p^L$ and all the other variables are generated as in Game_{2-h-2}. Notice that $\boldsymbol{\nu}$ is equal to zero at position i such that $i \notin \bar{W}(\boldsymbol{P}^0) \wedge i \notin \bar{W}(\boldsymbol{P}^1)$.
Let us see that the distribution $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{B}, \mathbb{B}^*, \left\{\mathsf{sk}^{(j)}\right\}_{j \in [\![1,Q]\!]}, c_1,$ $\boldsymbol{c}_2)$ in Game_{2-h-2} and that in $\mathsf{Game}_{2-h-2'}$ are equivalent except with negligible probability.

Here we cannot do as in the original proof. Indeed, otherwise with the change of bases $\mathbb{B}, \mathbb{B}^*$ to $\mathbb{D}, \mathbb{D}^*$, the h-th key can no longer decrypt the ciphertext. Thus, the adversary can distinguish the different games as in one case the h-th key decrypts the challenge ciphertext but not in the other case.
That is because, with the definition of $\mathbb{D}, \mathbb{D}^*$, some elements of $\mathbb{B}$ (resp.$\mathbb{B}^*$) are now linear combination of elements of $\mathbb{D}$ (resp.$\mathbb{D}^*$). Thus, the set $\bar{W}(\boldsymbol{P}^b) \cap \mathcal{I}$ is no longer equal to $\varnothing$ but is equal to $\bar{W}(\boldsymbol{P}^b)$.

We will consider the distribution in Game_{2-h-2}. We define new dual orthonormal bases $(\mathbb{D}, \mathbb{D}^*)$, following the idea of the last lemma in the original proof. For $i \in [\![1,L]\!]$ let $\theta_i, \tau_i \leftarrow \mathbb{Z}_p$ and set

$$\boldsymbol{d}_i = \tau_i^{-1} \boldsymbol{b}_i + \theta_i \boldsymbol{b}_{L+i},\ \boldsymbol{d}_i^* = \tau_i \boldsymbol{b}_i^*\ \boldsymbol{d}_{L+i} = \tau_i \boldsymbol{b}_{L+i}\ \boldsymbol{d}_{L+i}^* = -\theta_i \boldsymbol{b}_i^* + \tau_i^{-1} \boldsymbol{b}_{L+i}^*,$$

$$\mathbb{D} = (\boldsymbol{b}_0, \boldsymbol{d}_1 \cdots, \boldsymbol{d}_L, \boldsymbol{d}_{L+1}, \cdots, \boldsymbol{d}_{2L}, \boldsymbol{b}_{2L+1} \cdots \boldsymbol{b}_{4L+1}),$$
$$\mathbb{D}^* = (\boldsymbol{b}_0^*, \boldsymbol{d}_1^*, \cdots, \boldsymbol{d}_L^*, \boldsymbol{b}_L^*, \boldsymbol{d}_{L+1}^*, \cdots, \boldsymbol{d}_{2L}^*, \boldsymbol{b}_{2L+1}^*, \cdots, \boldsymbol{b}_{4L+1}^*).$$

We then easily verify that $\mathbb{D}$ and $\mathbb{D}^*$ are dual orthonormal. The h-th queried key and challenge ciphertext $(\mathsf{sk}^{(h)}, c_1, \boldsymbol{c}_2)$ in Game_{2-h-2} are expressed over bases $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{D}, \mathbb{D}^*)$ as

$$\begin{aligned}\mathsf{sk}^{(h)} &= g_2^{\alpha \boldsymbol{b}_0^* + \sum_{j\in\mathcal{I}} r_j \boldsymbol{b}_j^* + \sum_{j\in\mathcal{I}} z_j \boldsymbol{b}_{L+j}^* + \sum_{l=1}^{L} \eta_l \boldsymbol{b}_{3L+l}^*} \\ &= g_2^{\alpha \boldsymbol{d}_0^* + \sum_{j\in\mathcal{I}} r_j \tau_j \boldsymbol{d}_j^* + \sum_{j\in\mathcal{I}} z_j (\tau_j \boldsymbol{d}_{L+j}^* + \theta_j \boldsymbol{d}_j^*) + \sum_{l=1}^{L} \eta_l \boldsymbol{d}_{3L+l}^*} \\ &= g_2^{\alpha \boldsymbol{d}_0^* + \sum_{j\in\mathcal{I}} (r_j \tau_j + z_j \theta_j) \boldsymbol{d}_j^* + \sum_{j\in\mathcal{I}} z_j \tau_j \boldsymbol{d}_{L+j}^* + \sum_{l=1}^{L} \eta_l \boldsymbol{d}_{3L+l}^*}\end{aligned}$$

$$\begin{aligned}\boldsymbol{c}_2 &= g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + s_3 \sum_{i\in\overline{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i + t \sum_{i\in\overline{W}(\boldsymbol{P}^b)} \boldsymbol{b}_{L+i} + u \sum_{i\in\overline{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i\in\overline{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i}} \\ &= g_1^{s_1 \boldsymbol{d}_0 + s_2 \boldsymbol{d}_{4L+1} + s_3 \sum_{i\in\overline{W}(\boldsymbol{P}^b)} (\tau_i \boldsymbol{d}_i - \theta_i \boldsymbol{d}_{L+i}) + t \sum_{i\in\overline{W}(\boldsymbol{P}^b)} \boldsymbol{d}_{L+i}^* + u \sum_{i\in\overline{W}(\boldsymbol{P}^0)} \boldsymbol{d}_{2L+i}} \\ &\qquad \cdot g_1^{\tilde{u} \sum_{i\in\overline{W}(\boldsymbol{P}^1)} \boldsymbol{d}_{2L+i}} \\ &= g_1^{s_1 \boldsymbol{d}_0 + s_2 \boldsymbol{d}_{4L+1} + s_3 \sum_{i\in\overline{W}(\boldsymbol{P}^b)} \tau_i \boldsymbol{d}_i + \sum_{i\in\overline{W}(\boldsymbol{P}^b)} (t - s_3 \theta_i) \boldsymbol{d}_{L+i}^* + u \sum_{i\in\overline{W}(\boldsymbol{P}^0)} \boldsymbol{d}_{2L+i}} \\ &\qquad \cdot g_1^{\tilde{u} \sum_{i\in\overline{W}(\boldsymbol{P}^1)} \boldsymbol{d}_{2L+i}}\end{aligned}$$

$$c_1 = \mathsf{m}.e(g_1, g_2)^{s \boldsymbol{b}_0^* \boldsymbol{b}_0} = \mathsf{m}.e(g_1, g_2)^{s \boldsymbol{d}_0^* \boldsymbol{d}_0}$$

where $\tilde{r}_j = r_j \tau_j + z_j \theta_j$, $\tilde{z}_j = z_j \tau_j$ for $j \in \mathcal{I}$ and $\tilde{r}_j = w_j = 0$ otherwise, and $\tilde{s}_i = s_3 \tau_i$, $\nu_i = t - s_3 \theta_i$ for $i \in \overline{W}(\boldsymbol{P}^b)$ and $\tilde{s}_i = \nu_i = 0$ otherwise. $\tilde{s}_i, \tilde{r}_i, \boldsymbol{\nu}, \boldsymbol{w}$ are uniformly distributed for the position different of 0 and independent of all the other variables except with probability $\frac{2}{p^{|\mathcal{I}|}} + 2/p$.

In the light of the adversary view, both $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{D}, \mathbb{D}^*)$ are consistent with public key $\mathsf{pk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{B}, \mathbb{B}^*)$ and the answered keys $\left\{\mathsf{sk}^{(j)}\right\}_{j\neq h}$. Therefore, by using the above result for the distribution of $(\mathsf{sk}^{(h)}, c_1, \boldsymbol{c}_2)$, $\left\{\mathsf{sk}^{(j)}\right\}_{j\in[\![1,Q]\!]}$ and $\boldsymbol{c}_2$ can be expressed as keys and ciphertext in two ways, in Game_{2-h-2} over bases $(\mathbb{B}, \mathbb{B}^*)$ and in $\mathsf{Game}_{2-h-2'}$ over bases $(\mathbb{D}, \mathbb{D}^*)$. Thus, Game_{2-h-2} can be conceptually changed to $\mathsf{Game}_{2-h-2'}$, except with probability $\frac{2}{p^{|\mathcal{I}|}} + 2/p$.

Now let us see that the distribution $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{B}, \mathbb{B}^*, \left\{\mathsf{sk}^{(j)}\right\}_{j\in[\![1,Q]\!]}$ $, c_1, \boldsymbol{c}_2)$ in Game_{2-h-3} and that in $\mathsf{Game}_{2-h-2'}$ are equivalent except with negligible probability. As above, we set new bases $(\mathbb{D}, \mathbb{D}^*)$. The h-th queried key $\mathsf{sk}^{(h)}$ in Game_{2-h-3} is expressed as above in bases $\mathbb{B}^*$ and $\mathbb{D}^*$, and the part of the ciphertext c_1 in Game_{2-h-3} is given as above. $\boldsymbol{c}_2$ in Game_{2-h-3} is expressed over bases $\mathbb{B}$ and $\mathbb{D}$ as g_1 with exponent

$$\begin{aligned}
&s_1\boldsymbol{b}_0 + s_2\boldsymbol{b}_{4L+1} + s_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i + t \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{L+i} \\
&+\tilde{t} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{L+i} + u \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i} \\
&= s_1\boldsymbol{d}_0 + s_2\boldsymbol{d}_{4L+1} + s_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} (\tau_i\boldsymbol{d}_i - \theta_i\boldsymbol{d}_{L+i}) + t \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \tau_i^{-1}\boldsymbol{d}_{L+i} \\
&+\tilde{t} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \tau_i^{-1}\boldsymbol{d}_{L+i} + u \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{d}_{2L+i} + \tilde{u} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \boldsymbol{d}_{2L+i} \\
&= s_1\boldsymbol{d}_0 + s_2\boldsymbol{d}_{4L+1} + s_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \tau_i\boldsymbol{d}_i - s_3 \sum_{i\in\bar{W}(\boldsymbol{P}^b)} \theta_i\boldsymbol{d}_{L+i} + t \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \tau_i^{-1} \\
&\boldsymbol{d}_{L+i} + \tilde{t} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \tau_i^{-1}\boldsymbol{d}_{L+i} + u \sum_{i\in\bar{W}(\boldsymbol{P}^0)} \boldsymbol{d}_{2L+i} + \tilde{u} \sum_{i\in\bar{W}(\boldsymbol{P}^1)} \boldsymbol{d}_{2L+i}
\end{aligned}$$

We can define $\boldsymbol{\nu}$ the coefficient vector of $(\boldsymbol{d}_{L+1}, \cdots, \boldsymbol{d}_{2L})$ as for $i \in [\![1, L]\!]$:

$$\nu_i = \begin{cases}
\tau_i^{-1}t & \text{if } i \in \bar{W}(\boldsymbol{P}^0) \wedge i \notin \bar{W}(\boldsymbol{P}^1) \wedge b = 1 \\
\tau_i^{-1}\tilde{t} & \text{if } i \in \bar{W}(\boldsymbol{P}^1) \wedge i \notin \bar{W}(\boldsymbol{P}^0) \wedge b = 0 \\
-s_3\theta_i + \tau_i^{-1}t & \text{if } i \in \bar{W}(\boldsymbol{P}^0) \wedge i \notin \bar{W}(\boldsymbol{P}^1) \wedge b = 0 \\
-s_3\theta_i + \tau_i^{-1}\tilde{t} & \text{if } i \in \bar{W}(\boldsymbol{P}^1) \wedge i \notin \bar{W}(\boldsymbol{P}^0) \wedge b = 1 \\
-s_3\theta_i + \tau_i^{-1}t + \tau_i^{-1}\tilde{t} & \text{if } i \in \bar{W}(\boldsymbol{P}^0) \wedge i \in \bar{W}(\boldsymbol{P}^1) \\
0 & \text{otherwise}
\end{cases}$$

and $\tilde{s}_i = s_3\tau_i$ for $i \in \bar{W}(\boldsymbol{P}^b)$. $\boldsymbol{\nu}, \{\tilde{s}_i\}_{i=1,\cdots,L}$ are uniformly distributed for the position different of 0 and independent of the other variables except with probability $3/p$.

Similar as above, we see that $\left\{\mathsf{sk}^{(j)}\right\}_{j\in[\![1,Q]\!]}$ and $\boldsymbol{c}_2$ can be expressed as keys and ciphertext in two ways, in Game_{2-h-3} over bases $(\mathbb{B}, \mathbb{B}^*)$ and in $\mathsf{Game}_{2-h-2'}$ over bases $(\mathbb{D}, \mathbb{D}^*)$. Thus Game_{2-h-3} can be conceptually changed to $\mathsf{Game}_{2-h-2'}$ except with probability $\frac{2}{p^{|\mathcal{I}|}} + 3/p$. Combining both, we obtain lemma 20.

Lemma 21. *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_{2-2}$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,*

$$\left|\mathsf{Adv}_{\mathcal{A}}^{2-h-3}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{2-h-4}(\lambda)\right| \leq \mathsf{Adv}_{\mathcal{B}_{2-h-2}}^{P3}(\lambda)$$

where $\mathcal{B}_{2-h-2}(.) = B_{2-2}(h, .)$.

Proof. We construct a probabilistic adversary $\mathcal{B}_{2-2}$ against Problem 3 using an adversary $\mathcal{A}$ in a security game (Game_{2-h-3} or Game_{2-h-4}) as a black box as follows:

1. $\mathcal{B}_{2-2}$ is given an integer h and a Problem 3 instance, $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \hat{\mathbb{B}}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{e}_i, \boldsymbol{f}_i\}_{i \in [\![1,n]\!]})$.
2. $\mathcal{B}_{2-2}$ plays a role of the challenger in the security game against adversary $\mathcal{A}$.
3. At the first step of the game, $\mathcal{B}_{2-h}$ provides $\mathcal{A}$ a public key $\mathsf{pk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e(g_1, g_2)^{\alpha \boldsymbol{b}_0 \boldsymbol{b}_0^*}, g_1^{\boldsymbol{b}_0}, g_1^{\boldsymbol{b}_1}, \cdots, g_1^{\boldsymbol{b}_L}, g_1^{\boldsymbol{b}_{4L+1}}$ of Game_{2-h-3} (and Game_{2-h-4}), obtained from the Problem 3 instance.
4. When the ι-th key query is issued for a pattern $\boldsymbol{P}$, $\mathcal{B}_{2-2}$ answers as follows:
   - When $1 \leq \iota \leq h-1$, $\mathcal{B}_{2-2}$ answers keys of temporal 2 form, that are computed using $\mathbb{B}^*$ of the Problem 3 instance.
   - When $\iota = h$, $\mathcal{B}_{2-2}$ calculates $\mathsf{sk}_{\boldsymbol{P}}$ using $(\{\boldsymbol{h}^*_{\beta,i}\}_{i \in [\![1,L]\!]}, \{\boldsymbol{b}_i^*\}_{i=0,3L+1,\cdots,4L})$ of the Problem 3 instance as follows: $\{\sigma_i \xi_i\}_{i \in [\![1,L]\!]} \leftarrow \mathbb{Z}_p, \boldsymbol{\eta} = (\eta_1, \cdots, \eta_L) \leftarrow \mathbb{Z}_p^L$

     $$\mathsf{sk}_{\boldsymbol{P}} = g_2^{\alpha \boldsymbol{b}_0^* + \sum_{i \in \mathcal{I}} \sigma_i \boldsymbol{b}_i^* + \sum_{i \in [\![1,L]\!]} \eta_i \boldsymbol{b}^*_{3L+i}} \cdot \prod_{i \in \mathcal{I}} \boldsymbol{h}^{*\xi_i}_{\beta,i}$$

   - When $\iota \geq h+1$, $\mathcal{B}_{2-2}$ answers normal keys using $\mathbb{B}^*$ of the Problem 3 instance.
5. When $\mathcal{B}_{2-1}$ receives an encryption query with the challenge plaintext m and patterns $\boldsymbol{P}^0, \boldsymbol{P}^1$ from $\mathcal{A}$, $\mathcal{B}_{2-2}$ computes the challenge ciphertext $(c_1, \boldsymbol{c}_2)$ such that

   $$\boldsymbol{c}_2 = g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + s_3 \sum_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i} \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{e}_i \cdot \prod\nolimits_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{f}_i$$

   $$c_1 = \mathsf{m} \cdot (e(g_1, g_2)^{\alpha \boldsymbol{b}_0 \cdot \boldsymbol{b}_0^*})^{s_1}$$

   where $s_1, s_2, s_3 \leftarrow \mathbb{Z}_p, b \leftarrow \{0,1\}$ and $(\{\boldsymbol{b}_i\}_{i=0,2L+1,\cdots,3L,4L+1}, \{\boldsymbol{e}_i\}_{i \in [\![1,L]\!]}, \{\boldsymbol{e}_i, \boldsymbol{f}_i\}_{i \in [\![1,L]\!]})$ is a part of the Problem 3 instance.
6. When a key query is issued by $\mathcal{A}$ after the encryption query, $\mathcal{B}_{2-2}$ executes the same procedure as in that of step 4.
7. $\mathcal{A}$ finally outputs bit b'. If $b = b'$, $\mathcal{B}_{2-2}$ outputs $\beta' = 1$. Otherwise, $\mathcal{B}_{2-2}$ outputs $\beta' = 0$.

Let us see that the distribution of the view of adversary $\mathcal{A}$ in the above-mentioned game simulated by $\mathcal{B}_{2-2}$ given a Problem 3 instance with $\beta \in \{0,1\}$ is the same as that in Game_{2-h-3} (resp. Game_{2-h-4}) if $\beta = 0$ (resp. $\beta = 1$).

We consider the joint distribution of $\boldsymbol{c}_2$ and $\mathsf{sk}_{\boldsymbol{P}}$. Ciphertext $\boldsymbol{c}_2$ is

$$g_1^{s_1\boldsymbol{b}_0+s_2\boldsymbol{b}_{4L+1}+s_3\sum_{i\in\bar{W}(\boldsymbol{P}^b)}\boldsymbol{b}_i}\cdot\prod_{i\in\bar{W}(\boldsymbol{P}^0)}\boldsymbol{e}_i\cdot\prod_{i\in\bar{W}(\boldsymbol{P}^1)}\boldsymbol{f}_i$$

$$=g_1^{s_1\boldsymbol{b}_0+s_2\boldsymbol{b}_{4L+1}+s_3\sum_{i\in\bar{W}(\boldsymbol{P}^b)}\boldsymbol{b}_i+\sum_{i\in\bar{W}(\boldsymbol{P}^0)}(\omega^{'}b_{L+i}+\omega^{"}b_{2L+i})+\sum_{i\in\bar{W}(\boldsymbol{P}^1)}(\kappa^{'}b_{L+i}+\kappa^{"}b_{2L+i})}$$

$$=g_1^{s_1\boldsymbol{b}_0+s_2\boldsymbol{b}_{4L+1}+s_3\sum_{i\in\bar{W}(\boldsymbol{P}^b)}\boldsymbol{b}_i+\sum_{i\in\bar{W}(\boldsymbol{P}^0)}\omega^{'}b_{L+i}+\sum_{i\in\bar{W}(\boldsymbol{P}^1)}\kappa^{'}b_{L+i}}$$
$$\cdot g_1^{\sum_{i\in\bar{W}(\boldsymbol{P}^0)}\omega^{"}b_{2L+i}+\sum_{i\in\bar{W}(\boldsymbol{P}^1)}\kappa^{"}b_{2L+i}}$$

where $s_1,s_2,s_3,\omega^{'},\omega^{"},\kappa^{'},\kappa^{"}\in\mathbb{Z}_p$ are uniformly distributed.
Now let's see the value of $\mathsf{sk}_{\boldsymbol{P}}$. When $\beta=0$, $\mathsf{sk}_{\boldsymbol{P}}$ is case (b) of step 4 or 6 is

$$g_2^{\alpha\boldsymbol{b}_0^*+\sum_{i\in\mathcal{I}}\sigma_i\boldsymbol{b}_i^*+\sum_{i\in[\![1,L]\!]}\eta_i\boldsymbol{b}_{3L+i}^*}\cdot\prod_{i\in\mathcal{I}}\boldsymbol{h}_{\beta,i}^{*\xi_i}$$

$$=g_2^{\alpha\boldsymbol{b}_0^*+\sum_{i\in\mathcal{I}}\sigma_i\boldsymbol{b}_i^*+\sum_{i\in[\![1,L]\!]}\eta_i\boldsymbol{b}_{3L+i}^*+\sum_{i\in\mathcal{I}}(\tau\xi_i\boldsymbol{b}_{L+i}^*+\xi_i\delta_0\boldsymbol{b}_{3L+i}^*)}$$

$$=g_2^{\alpha\boldsymbol{b}_0^*+\sum_{i\in\mathcal{I}}\sigma_i\boldsymbol{b}_i^*+\sum_{i\in\mathcal{I}}\tau\xi_i\boldsymbol{b}_{L+i}^*+\sum_{i\in\mathcal{I}}\delta_0\xi_i\eta_i\boldsymbol{b}_{3L+i}^*+\sum_{i\in\mathcal{O}}\eta_i\boldsymbol{b}_{3L+i}^*}$$

where $\alpha,\sigma,\tau,\delta_0,\{\eta_i\}_{i\in[L]}$ are uniformly and independently distributed. Therefore, generated $\boldsymbol{c}_2,\mathsf{sk}_{\boldsymbol{P}}$ have the same joint distribution as in Game_{2-h-3}.

When $\beta=1$, $\mathsf{sk}_{\boldsymbol{P}}$ in case (b) of step 4 or 6 is

$$g_2^{\alpha\boldsymbol{b}_0^*+\sum_{i\in\mathcal{I}}\sigma_i\boldsymbol{b}_i^*+\sum_{i\in[\![1,L]\!]}\eta_i\boldsymbol{b}_{3L+i}^*}\cdot\prod_{i\in\mathcal{I}}\boldsymbol{h}_{\beta,i}^{*\xi_i}$$

$$=g_2^{\alpha\boldsymbol{b}_0^*+\sum_{i\in\mathcal{I}}\sigma_i\boldsymbol{b}_i^*+\sum_{i\in[\![1,L]\!]}\eta_i\boldsymbol{b}_{3L+i}^*+\sum_{i\in\mathcal{I}}(\tau\xi_i\boldsymbol{b}_{2L+i}^*+\xi_i\delta_0\boldsymbol{b}_{3L+i}^*)}$$

$$=g_2^{\alpha\boldsymbol{b}_0^*+\sum_{i\in\mathcal{I}}\sigma_i\boldsymbol{b}_i^*+\sum_{i\in\mathcal{I}}\tau\xi_i\boldsymbol{b}_{2L+i}^*+\sum_{i\in\mathcal{I}}\xi_i\delta_0\eta_i\boldsymbol{b}_{3L+i}^*+\sum_{i\in\mathcal{O}}\eta_i\boldsymbol{b}_{3L+i}^*}$$

where $\alpha,\sigma,\tau,\delta_0,\{\eta_i\}_{i\in[L]}$ are uniformly and independently distributed. Therefore, generated $\boldsymbol{c}_2,\mathsf{sk}_{\boldsymbol{P}}$ have the same joint distribution as in Game_{2-h-4}.

Thus, $\left|\mathsf{Adv}_{\mathcal{A}}^{2-h-3}(\lambda)-\mathcal{A}^{2-h-4}(\lambda)\right|\leq\mathsf{Adv}_{\mathcal{B}_{2-h-2}}^{P3}(\lambda)$.

Lemma 22. *For any adversary $\mathcal{A}$, $\left|\mathsf{Adv}_{\mathcal{A}}^{2-Q-4}(\lambda)-\mathsf{Adv}_{\mathcal{A}}^{3}(\lambda)\right|\leq\frac{2}{p^{|\mathcal{I}|}}+3/p$.*

Proof. To prove this lemma, we will show that distribution $(\mathsf{pk},\{\mathsf{sk}_{\boldsymbol{P}^j}\}_{j\in[\![1,Q]\!]},c_1,$ $\boldsymbol{c}_2)$ in Game_{2-Q-4} and that in Game_3 are equivalent. For that purpose, we define new dual orthonormal bases $(\mathbb{D},\mathbb{D}^*)$ as follows:

We generate $\theta_i \leftarrow \mathbb{Z}_p$ for $i \in \llbracket 1, L \rrbracket$ and set for $i \in \llbracket 1, L \rrbracket$

$$\begin{aligned} \boldsymbol{d}_{2L+i} = \boldsymbol{b}_{2L+i} - \theta_i \boldsymbol{b}_i,\ \ \boldsymbol{d}^*_i = \boldsymbol{b}^*_i + \theta_i \boldsymbol{b}^*_{2L+i} \\ \mathbb{D} = (\boldsymbol{b}_0, \cdots, \boldsymbol{b}_{2L}, \boldsymbol{d}_{2L+1}, \cdots, \boldsymbol{d}_{3L}, \boldsymbol{d}_{3L+1}, \cdots, \boldsymbol{d}_{4L+1}), \\ \mathbb{D}^* = (\boldsymbol{b}^*_0, \boldsymbol{d}^*_1, \cdots, \boldsymbol{d}^*_L, \boldsymbol{b}^*_{L+1}, \cdots, \boldsymbol{b}^*_{4L+1}) \end{aligned}$$

We then easily verify that $\mathbb{D}$ and $\mathbb{D}^*$ are dual orthonormal, and are distributed the same as the original bases $(\mathbb{B}, \mathbb{B}^*)$.

Keys and challenge ciphertext $\{\mathsf{sk}_{\boldsymbol{P}^j}\}_{j \in \llbracket 1, Q \rrbracket}, c_1, \boldsymbol{c}_2$ in Game$_{2-Q-4}$ are expressed over bases $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{D}, \mathbb{D}^*)$ as

$$\begin{aligned} \mathsf{sk}_{\boldsymbol{P}^j} &= g_2^{\alpha \boldsymbol{b}^*_0 + \sum_{i \in \mathcal{I}} r^j_i \boldsymbol{b}^*_i + \sum_{i \in \mathcal{I}} x^j_i \boldsymbol{b}^*_{2L+i} + \sum_{l=1}^{L} \eta^j_l \cdot \boldsymbol{b}^*_{3L+l}} \\ &= g_2^{\alpha \boldsymbol{d}^*_0 + \sum_{i \in \mathcal{I}} r^j_i (\boldsymbol{d}^*_i - \theta_i \boldsymbol{b}^*_{2L+i}) + \sum_{i \in \mathcal{I}} x^j_i \boldsymbol{d}^*_{2L+i} + \sum_{l=1}^{L} \eta^j_l \cdot \boldsymbol{d}^*_{3L+l}} \\ &= g_2^{\alpha \boldsymbol{d}^*_0 + \sum_{i \in \mathcal{I}} r^j_i \boldsymbol{d}^*_i + \sum_{j \in \mathcal{I}} (x^j_i - r^j_i \theta_i) \boldsymbol{d}^*_{2L+i} + \sum_{l=1}^{L} \eta^j_l \cdot \boldsymbol{d}^*_{3L+l}} \end{aligned}$$

$$\begin{aligned} \boldsymbol{c}_2 &= g_1^{s_1 \boldsymbol{b}_0 + s_2 \boldsymbol{b}_{4L+1} + s_3 \sum_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{b}_i + t \sum_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{L+i} + \tilde{t} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{L+i}} \\ &\quad \cdot g_1^{u \sum_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{b}_{2L+i} + \tilde{u} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{b}_{2L+i}} \\ &= g_1^{s_1 \boldsymbol{d}_0 + s_2 \boldsymbol{d}_{4L+1} + s_3 \sum_{i \in \bar{W}(\boldsymbol{P}^b)} \boldsymbol{d}_i + t \sum_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{d}_{L+i} + \tilde{t} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{d}_{L+i}} \\ &\quad \cdot g_1^{u \sum_{i \in \bar{W}(\boldsymbol{P}^0)} (\boldsymbol{d}_{2L+i} + \theta_i \boldsymbol{d}_i) + \tilde{u} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} (\boldsymbol{d}_{2L+i} + \theta_i \boldsymbol{d}_i)} \\ &= g_1^{s_1 \boldsymbol{d}_0 + s_2 \boldsymbol{d}_{4L+1} + \sum_{i=1}^{L} \nu_i \boldsymbol{d}_i + t \sum_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{d}_{L+i} + \tilde{t} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{d}_{L+i} + u \sum_{i \in \bar{W}(\boldsymbol{P}^0)} \boldsymbol{d}_{2L+i}} \\ &\quad \cdot g_1^{\tilde{u} \sum_{i \in \bar{W}(\boldsymbol{P}^1)} \boldsymbol{d}_{2L+i}} \end{aligned}$$

$$c_1 = \mathsf{m} \cdot e(g_1, g_2)^{\alpha \boldsymbol{b}_0 \boldsymbol{b}^*_0 s} = \mathsf{m} \cdot e(g_1, g_2)^{\alpha \boldsymbol{d}_0 \boldsymbol{d}^*_0 s}$$

where for $i \in \llbracket 1, L \rrbracket$:

$$\tilde{x}_i = \begin{cases} x^j_i - r^j_i \theta_i & \text{if } i \in \mathcal{I} \\ 0 & \text{otherwise} \end{cases}$$

and

$$\nu_i = \begin{cases} 0 & \text{if } i \notin \bar{W}(\boldsymbol{P}^0) \wedge i \notin \bar{W}(\boldsymbol{P}^1) \\ \theta_i u & \text{if } i \in \bar{W}(\boldsymbol{P}^0) \wedge i \notin \bar{W}(\boldsymbol{P}^1) \wedge b = 1 \\ \theta_i \tilde{u} & \text{if } i \in \bar{W}(\boldsymbol{P}^1) \wedge i \notin \bar{W}(\boldsymbol{P}^0) \wedge b = 0 \\ s_3 + u\theta_i & \text{if } i \in \bar{W}(\boldsymbol{P}^0) \wedge i \notin \bar{W}(\boldsymbol{P}^1) \wedge b = 0 \\ s_3 + \tilde{u}\theta_i & \text{if } i \in \bar{W}(\boldsymbol{P}^1) \wedge i \notin \bar{W}(\boldsymbol{P}^0) \wedge b = 1 \\ s_3 + (u + \tilde{u})\theta_i & \text{if } i \in \bar{W}(\boldsymbol{P}^1) \wedge i \in \bar{W}(\boldsymbol{P}^0) \end{cases}$$

are uniformly, independently (from other variables) distributed since $s_3, \theta, t^j \leftarrow \mathbb{Z}_p$, except with probability $\frac{2}{p^{|\mathcal{I}|}} + 3/p$.

In the light of the adversary's view, both $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{D}, \mathbb{D}^*)$ are consistent with public key pk. Therefore, $\{\mathsf{sk}_{\boldsymbol{P}^j}\}_{j \in [\![1,Q]\!]}, c_1, \boldsymbol{c}_2$ can be expressed as keys and ciphertext in two ways, in Game$_{2-Q-4}$ over bases $(\mathbb{B}, \mathbb{B}^*)$ and in Game$_3$ over bases $(\mathbb{D}, \mathbb{D}^*)$.

Thus, Game_{2-Q-4} can be conceptually changed to Game_3.

Combining all theses proofs, we obtain that any adversary has no advantage in winning the security game. Adding to these the fact that Problem 1, Problem 2 and Problem 3 hold if $\mathsf{XDLin}1, \mathsf{XDLin}2$ hold, we have proven theorem 10 when $t = 1$.

C Reductions proofs

The appendix presents the reductions proofs of our problems to $\mathsf{XDLin}\ _1$ and $\mathsf{XDLin}\ _2$. Reductions of DS1 and DS2 to DDH in respectively $\mathbb{G}_1, \mathbb{G}_2$ are done in [14].

C.1 Security reductions for Problems 1 and 1 bis

Proofs of lemmas 2 and 3, can be done as in [28], using the following intermediate problem (based on the one of [27], Annex B) and lemmas 23, 24, and 25.

Definition 24. ***Problem 0*** *is to guess β, given $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{B}, \hat{\mathbb{B}}^*, \boldsymbol{y}_\beta, \boldsymbol{f}^*, g_2^\kappa, g_1^\xi, g_1^{\delta\xi}) \leftarrow \mathcal{G}_\beta^{P0}(1^\lambda, n)$, where $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ is an asymmetric bilinear pairing group, and $\kappa, \xi, \rho, \tau \leftarrow \mathbb{Z}_p^*, \delta, \sigma, \omega \leftarrow \mathbb{Z}_p$, $\psi = \kappa \cdot \xi$, $(\mathbb{B}, \mathbb{B}^*) \leftarrow$ $\mathsf{Dual}(\mathbb{Z}_p^4)$, $\hat{\mathbb{B}^*} = (\boldsymbol{b}_1^*, \boldsymbol{b}_3^*, \boldsymbol{b}_4^*)$*

$$\boldsymbol{y}_0 = g_1^{\delta \boldsymbol{b}_1 + \sigma \boldsymbol{b}_4}, \boldsymbol{y}_1 = g_1^{\delta \boldsymbol{b}_1 + \rho \boldsymbol{b}_2 + \sigma \boldsymbol{b}_4}, \boldsymbol{f}^* = g_2^{\omega \boldsymbol{b}_1^* + \tau \boldsymbol{b}_2^*}.$$

For a probabilistic adversary $\mathcal{B}$, the advantage of $\mathcal{B}$ for Problem 0 is defined as

$$\mathsf{Adv}_{\mathcal{B}}^{P0}(\lambda) = \left| \Pr\left[\mathcal{B}(1^\lambda, \varrho) \rightarrow 1 | \varrho \leftarrow \mathcal{G}_0^{P0}(1^\lambda, n)\right] - \Pr\left[\mathcal{B}(1^\lambda, \varrho) \rightarrow 1 | \varrho \leftarrow \mathcal{G}_1^{P0}(1^\lambda, n)\right] \right|$$

Lemma 23. *For any adversary $\mathcal{D}$, there is a probabilistic machine $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{E}$, such that for any security parameter λ,*
$\mathsf{Adv}_{\mathcal{D}}^{P0}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLin}1}(\lambda) + 5/p$.

Proof. Given a $\mathsf{XDLin1}$ instance $g_1, g_1^\xi, g_1^\kappa, g_1^{\delta\xi}, g_1^{\sigma\kappa}, g_2, g_2^\xi, g_2^\kappa, g_2^{\delta\xi}$, $g_2^{\sigma\kappa}, Y_\beta$, where $Y_\beta = g_1^{\delta+\sigma}$ or g_1^z where $z \leftarrow \mathbb{Z}_p$ is chosen randomly, $\mathcal{E}$ calculates $g_T = e(g_1^\kappa, g_2^\xi)$

and sets 4×4 matrices $\boldsymbol{\Pi}^*, \boldsymbol{\Pi}$ as follows:

$$\boldsymbol{\Pi} = \begin{pmatrix} \xi & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & \kappa & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \boldsymbol{\Pi}^* = \begin{pmatrix} \kappa & 0 & 0 & 0 \\ -\kappa & -\xi & 0 & \kappa\xi \\ 0 & \xi & 0 & 0 \\ 0 & 0 & \kappa\xi & 0 \end{pmatrix}.$$

Then, $\boldsymbol{\Pi}.(\boldsymbol{\Pi}^*)^\top = \kappa\xi \cdot Id_4$. By using $\boldsymbol{\Pi}, \boldsymbol{\Pi}^*$, $\mathcal{E}$ sets

$$\begin{array}{llll} \boldsymbol{u}_1 = (g_1^\xi, 0, 0, g_1), & \boldsymbol{u}_2 = (0, 0, 0, g_1), & \boldsymbol{u}_3 = (0, g_1^\kappa, 0, g_1) & \boldsymbol{u}_4 = (0, 0, g_1, 0), \\ \boldsymbol{u}_1^* = (g_2^\kappa, 0, 0, 0), & \boldsymbol{u}_2^* = (g_2^{-\kappa}, g_2^{-\xi}, 0, g_2^{\kappa\xi}), & \boldsymbol{u}_3^* = (0, g_2^\xi, 0, 0) & \boldsymbol{u}_4^* = (0, 0, g_2^{\kappa\xi}, 0) \end{array}$$

$\mathcal{E}$ can compute $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \boldsymbol{u}_4, \boldsymbol{u}_1^*, \boldsymbol{u}_3^*$ from the above $\mathsf{XDLin1}$ instance above. Let bases $\mathbb{U} = (\boldsymbol{u}_i)_{i=1,2,3,4}$ of $\mathbb{G}_1^4$ and $\mathbb{U}^* = (\boldsymbol{u}_i^*)_{i=1,2,3,4}$ of $\mathbb{G}_2^4$. $\mathcal{E}$ chooses $\eta, \phi \leftarrow \mathbb{Z}_p$ such that $\eta \neq 0$, and sets

$$\boldsymbol{v}^* = (g_2^\phi, g_2^{-\eta}, 0, g_2^{\eta\kappa}) \quad \boldsymbol{w}_\beta = (g_1^{\delta\xi}, g_1^{\sigma\kappa}, 0, Y_\beta)$$

$\mathcal{E}$ generates random linear transformation $\boldsymbol{W} = (w_{i,j})_{i,j=1,\cdots,4} \leftarrow GL(4, \mathbb{Z}_p)$, $\boldsymbol{Z} = (W^\top)^{-1} = (z_i, j)_{i,j=1,\cdots,4}$ then calculates

$$\begin{array}{ll} \boldsymbol{b}_i^* = \sum_{j=1}^4 w_{i,j} \boldsymbol{u}_j^* \text{ for } i = 1, 3 & \boldsymbol{b}_i = \sum_{j=1}^4 z_{i,j} \boldsymbol{u}_j \text{ for } i = 1, 2, 3, 4 \\ \hat{\mathbb{B}}^* = (\boldsymbol{b}_1^*, \boldsymbol{b}_3^*, \boldsymbol{b}_4^*) & \mathbb{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \boldsymbol{b}_4), \\ \boldsymbol{f}^* = \boldsymbol{W}(\boldsymbol{v}^*), & \boldsymbol{y}_\beta = \boldsymbol{Z}(\boldsymbol{w}_\beta) \end{array}$$

$\mathcal{E}$ then gives $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}^*}, \mathbb{B}, \boldsymbol{y}_\beta, \boldsymbol{f}^*, g_2^\kappa, g_1^\xi, g_1^{\delta\xi})$ where $g_2^\kappa, g_1^\xi, g_1^{\delta\xi}$ are contained in the $\mathsf{XDLin1}$ instance, and outputs $\beta^{'} \in \{0, 1\}$ if $\mathcal{D}$ outputs $\beta^{'}$.

If we set $\tau = \xi^{-1}\eta$, $\omega = \tau + \kappa^{-1}\phi$ then $\kappa \neq 0$ (since $\eta \neq 0$),

$$\begin{aligned} \boldsymbol{v}^* &= (g_2^\phi, g_2^{-\eta}, 0, g_2^{\eta\kappa}) = (g_2^{(\omega-\tau)\kappa}, g_2^{-\tau\xi}, 0, g_2^{\tau\kappa\xi}) = \boldsymbol{u}_1^{*\omega} + \boldsymbol{u}_2^{*\tau}, \\ \boldsymbol{f}^* &= W \cdot \boldsymbol{v}^* = W \cdot (\boldsymbol{u}_1^{*\omega} + \boldsymbol{u}_2^{*\tau}) = g_2^{\omega \boldsymbol{b}_1^* + \tau \boldsymbol{b}_2^*} \end{aligned}$$

If $\beta = 0$, i.e. $Y_\beta = Y_0 = g_1^{\delta+\sigma}$, then

$$\begin{aligned} \boldsymbol{w}_0 &= (g_1^{\delta\xi}, g_1^{\sigma\kappa}, 0, g_1^{\delta+\sigma}) = \boldsymbol{u}_1^\delta + \boldsymbol{u}_3^\sigma \\ \boldsymbol{y}_0 &= \boldsymbol{Z} \cdot \boldsymbol{w}_0 = \boldsymbol{Z} \cdot (\boldsymbol{u}_1^\delta + \boldsymbol{u}_3^\sigma) = g_1^{\delta \boldsymbol{b}_1 + \sigma \boldsymbol{b}_4}. \end{aligned}$$

Thus, the distribution of $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}^*, \mathbb{B}, \boldsymbol{y}_0, \ \boldsymbol{f}^*, g_2^\kappa, g_1^\xi, g_1^{\delta\xi})$ is exactly the same as $\left\{\varrho | \varrho \leftarrow \mathcal{G}_0^{P0}(1^\lambda)\right\}$ when $\kappa \neq 0$ and $\xi \neq 0$, i.e. except with probability $2/p$.

If $\beta = 1$, i.e. $Y_\beta = Y_1 = g_1^\psi$ is uniformly distributed in $\mathbb{G}_1$, we set $\rho = \psi_\delta - \sigma$. Then

$$\begin{aligned} \boldsymbol{w}_1 &= (g_1^{\delta\xi}, g_1^{\sigma\kappa}, 0, g_1^{\delta+\rho+\sigma}) = \boldsymbol{u}_1\delta + \boldsymbol{u}_2^\rho + \boldsymbol{u}_3^\sigma, \\ \boldsymbol{y}_1 &= \boldsymbol{Z} \cdot \boldsymbol{w}_1 = \boldsymbol{Z} \cdot (\boldsymbol{u}_1\delta + \boldsymbol{u}_2^\rho + \boldsymbol{u}_3^\sigma) = g_1^{\delta \boldsymbol{b}_1 + \rho \boldsymbol{b}_2 + \sigma \boldsymbol{b}_4}. \end{aligned}$$

Therefore, the distribution of $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}^*, \mathbb{B}, \boldsymbol{y}_1, \boldsymbol{f}^*, g_2^{\kappa}, g_1^{\xi}, g_1^{\delta\xi})$ is exactly the same as $\left\{\varrho | \varrho \leftarrow \mathcal{G}_1^{P0}(1^{\lambda})\right\}$ when $\kappa \neq 0, \xi \neq 0$ and $\rho \neq 0$, i.e. except with probability $3/p$.

Therefore, $\mathsf{Adv}_{\mathcal{D}}^{P0}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLin1}}(\lambda) + 5/p$.

Lemma 24. *For any adversary $\mathcal{C}$, there is a probabilistic machine $\mathcal{D}$, whose running time is essentially the same as that of $\mathcal{C}$, such that for any security parameter λ,*
$\mathsf{Adv}_{\mathcal{C}}^{P1}(\lambda) = \mathsf{Adv}_{\mathcal{D}}^{P0}(\lambda)$.

Proof. Lemma 24. $\mathcal{D}$ is given a Problem 0 instance $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{B}, \hat{\mathbb{B}}^*, \boldsymbol{y}_{\beta}, \boldsymbol{f}^*, g_2^{\kappa}, g_1^{\xi}, g_1^{\delta\xi})$. $\mathcal{D}$ generates random linear transformation $\boldsymbol{W} = (w_{i,j})_{i,j=1,\cdots,4n+2}$ $\leftarrow GL(4n+2, \mathbb{Z}_p)$, $\boldsymbol{Z} = (\boldsymbol{W}^{\top})^{-1} = (z_{i,j})_{i,j=1,\cdots,4n+2}$, then sets

$$\begin{array}{ll}
g_1^{\boldsymbol{d}_0} = \boldsymbol{W} \cdot (g_1^{\xi}, \mathbf{0}^{4n+1}) & g_2^{\boldsymbol{d}_0^*} = \boldsymbol{Z} \cdot (g_2^{\kappa}, \mathbf{0}^{4n+1}) \\
g_1^{\boldsymbol{d}_1} = \boldsymbol{W} \cdot (0, \boldsymbol{b}_1, \mathbf{0}^{4n+1-4}) & g_2^{\boldsymbol{d}_1^*} = \boldsymbol{Z} \cdot (0, \boldsymbol{b}_1^*, \mathbf{0}^{4n+1-4}) \\
g_1^{\boldsymbol{d}_{n+1}} = \boldsymbol{W} \cdot (0, \boldsymbol{b}_2, \mathbf{0}^{4n-4}) & g_2^{\boldsymbol{d}_{n+1}^*} = \boldsymbol{Z} \cdot (0, \boldsymbol{b}_2^*, \mathbf{0}^{4n-4}) \\
g_1^{\boldsymbol{d}_{3n+1}^*} = \boldsymbol{W} \cdot (0, \boldsymbol{b}_3, \mathbf{0}^{4n-4}) & g_2^{\boldsymbol{d}_{3n}^*} = \boldsymbol{Z} \cdot (0, \boldsymbol{b}_3^*, g_2^{\kappa}, \mathbf{0}^{4n-4}) \\
g_1^{\boldsymbol{d}_{4n+1}} = \boldsymbol{W} \cdot (0, \boldsymbol{b}_4, \mathbf{0}^{4n+1-4}) & g_2^{\boldsymbol{d}_{4n+1}^*} = \boldsymbol{Z} \cdot (0, \boldsymbol{b}_4^*, \mathbf{0}^{4n+1-4}))
\end{array}$$

and

$$g_1^{\boldsymbol{d}_i} = \begin{cases} \boldsymbol{W} \cdot (0, \mathbf{0}^{i+2}, g_1^{\xi}, \mathbf{0}^{4n+1-(i+2)-1}) & \text{for } i \in [\![2, n]\!] \\ \boldsymbol{W} \cdot (0, \mathbf{0}^{i+1}, g_1^{\xi}, \mathbf{0}^{4n-(i+1)-1}) & \text{for } i \in [\![n+2, 3n]\!] \\ \boldsymbol{W} \cdot (0, \mathbf{0}^{i}, g_1^{\xi}, \mathbf{0}^{4n-i-1}) & \text{for } i \in [\![3n+2, 4n]\!] \end{cases}$$

$$g_2^{\boldsymbol{d}_i^*} = \begin{cases} \boldsymbol{Z} \cdot (0, \mathbf{0}^{i+2}, g_2^{\kappa}, \mathbf{0}^{4n+1-(i+2)-1}) & \text{for } i \in [\![2, n]\!] \\ \boldsymbol{Z} \cdot (0, \mathbf{0}^{i+1}, g_2^{\kappa}, \mathbf{0}^{4n-(i+1)-1}) & \text{for } i \in [\![n+2, 3n]\!] \\ \boldsymbol{Z} \cdot (0, \mathbf{0}^{i}, g_2^{\kappa}, \mathbf{0}^{4n-i-1}) & \text{for } i \in [\![3n+2, 4n]\!] \end{cases}$$

Then $(\mathbb{D}, \mathbb{D}^*)$ are dual orthonormal bases. $\mathcal{D}$ can compute $\mathbb{D}, \hat{\mathbb{D}}^*$ from $\mathbb{B}, \hat{\mathbb{B}}^*, g_2^{\kappa}$ and g_1^{ξ}.

Note 11. Here we directly give $(\mathbb{D}, \mathbb{D}^*)$ in exponent of g_1, g_2 respectively, as we cannot compute them directly.

It computes:

$$\boldsymbol{g}_{\beta,1} = W \cdot (\boldsymbol{y}_{\beta}, \mathbf{0}^{4n-4}) \; \boldsymbol{g}_i = W \cdot (0, \mathbf{0}^{i+2}, g_1^{\delta\xi}, \mathbf{0}^{4n-(i+2)-1})$$

for $i = 2, \cdots, n$. $\mathcal{D}$ then gives $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{D}, \hat{\mathbb{D}}^*, \boldsymbol{g}_{\beta,1}, \{\boldsymbol{g}_i\}_{i \in [\![1,n]\!]})$ to $\mathcal{C}$, and outputs $\beta^{'} \in \{0, 1\}$ if $\mathcal{C}$ outputs $\beta^{'}$. We can see that

$$\boldsymbol{g}_{0,1} = g_1^{\omega^{'}\boldsymbol{d}_1 + \gamma^{'}\boldsymbol{d}_{4n+1}} \quad \boldsymbol{g}_{1,1} = g_1^{\omega^{'}\boldsymbol{d}_1 + \tau^{'}\boldsymbol{d}_{n+1} + \gamma^{'}\boldsymbol{d}_{4n+1}} \quad \boldsymbol{g}_i = g_1^{\omega^{'}\boldsymbol{d}_i} \text{ for } i = 2, \cdots, n$$

where $\omega^{'} = \delta, \gamma^{'} = \sigma$ and $\tau^{'} = \rho$ which are distributed uniformly in $\mathbb{Z}_p$. Therefore the distribution of $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{D}, \hat{\mathbb{D}^*}, \boldsymbol{g}_{\beta,1}, \{\boldsymbol{g}_i\}_{i \in [\![1,n]\!]})$ is exactly the same as $\left\{\varrho | \varrho \leftarrow \mathcal{G}_{\beta}^{P1}(1^{\lambda}, n)\right\}$.

Combining lemmas 23 and 24 we have proven lemma 2. To complete the proof of lemma 3, we prove lemma 25.

Lemma 25. *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{A}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}_{\mathcal{B}}^{P1b}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{P1}(\lambda)$.*

Proof. Lemma 25. Let $\mathcal{B}$ be an adversary against Problem 1 bis, that wins with non negligible advantage. We construct $\mathcal{A}$ an adversary against Problem 1.

$\mathcal{A}$ is given a Problem 1 instance $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{B}, \hat{\mathbb{B}}^*, \boldsymbol{e}_{\beta,1}, \{\boldsymbol{e}_i\}_{i=2,\cdots,n})$. He picks $\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n \leftarrow \mathbb{Z}_p^n$ randomly and sets

$$\begin{array}{ll} \boldsymbol{d}_{n+1} = \boldsymbol{z}_1^{-1}\boldsymbol{b}_{n+1} + \sum_{j=2}^{n} \boldsymbol{z}_j \boldsymbol{b}_{n+j} & \boldsymbol{d}_{n+i} = \boldsymbol{z}_1 \boldsymbol{b}_{n+i} \text{ for } i = 2, \cdots, n \\ \boldsymbol{d}_{n+1}^* = \boldsymbol{z}_1 \boldsymbol{b}_{n+1}^* & \boldsymbol{d}_{n+i}^* = \boldsymbol{z}_1^{-1}\boldsymbol{b}_{n+i}^* - \boldsymbol{z}_i \boldsymbol{b}_{n+1}^* \text{ for } i = 2, \cdots, n \end{array}$$

Finally, $\mathcal{A}$ sets dual orthonormal bases

$$\begin{array}{l} \mathbb{D} = (\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_n, \boldsymbol{d}_{n+1}, \boldsymbol{d}_{n+2}, \cdots, \boldsymbol{d}_{2n}, \boldsymbol{b}_{2n+1}, \cdots, \boldsymbol{b}_{4n+1}) \\ \mathbb{D}^* = (\boldsymbol{b}_0^*, \boldsymbol{b}_1^*, \cdots, \boldsymbol{b}_n^*, \boldsymbol{d}_{n+1}^*, \cdots, \boldsymbol{d}_{2n}^*, \boldsymbol{b}_{2n+1}^*, \cdots, \boldsymbol{b}_{4n+1}^*) \\ \hat{\mathbb{D}}^* = (\boldsymbol{d}_0^*, \cdots, \boldsymbol{d}_n^*, \boldsymbol{d}_{2n+1}^*, \cdots, \boldsymbol{d}_{4n+1}^*). \end{array}$$

$\boldsymbol{e}_{\beta,1}, \{\boldsymbol{e}_i\}_{i=2,\cdots,n}$ are expressed in bases $\mathbb{B}, \mathbb{B}^*$ as

$$\boldsymbol{e}_{0,1} = g_1^{\omega \boldsymbol{b}_1 + \gamma \boldsymbol{b}_{4n+1}} \quad \boldsymbol{e}_{1,1} = g_1^{\omega \boldsymbol{b}_1 + z \boldsymbol{b}_{n+1} + \gamma \boldsymbol{b}_{4n+1}} \quad \boldsymbol{e}_i = g_1^{\omega \boldsymbol{b}_i} \text{ for } i = 2, \cdots, n.$$

and can be expressed in bases $\mathbb{D}, \mathbb{D}^*$ as

$$\boldsymbol{f}_{0,1} = g_1^{\omega \boldsymbol{d}_1 + \gamma \boldsymbol{d}_{4n+1}}, \quad \boldsymbol{f}_{1,1} = g_1^{\omega \boldsymbol{d}_1 + z \cdot \boldsymbol{z}_1 \boldsymbol{d}_{n+1} - z \sum_{j=2}^{n} \boldsymbol{z}_j \boldsymbol{d}_{n+j} + \gamma \boldsymbol{d}_{4n+1}}, \quad \boldsymbol{f}_i = g_1^{\omega \boldsymbol{d}_i}$$

for $i = 2, \cdots, n$.

$\mathcal{A}$ gives $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{D}, \hat{\mathbb{D}}^*, \boldsymbol{f}_{\beta,1}, \{\boldsymbol{f}_i\}_{i=2,\cdots,n})$ to $\mathcal{B}$ as a Problem 1 bis instance, and outputs $\beta^{'} \in \{0, 1\}$ if $\mathcal{B}$ outputs $\beta^{'}$.
By construction, the distribution of $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \mathbb{D}, \hat{\mathbb{D}}^*, \boldsymbol{f}_{\beta,1}, \{\boldsymbol{f}_i\}_{i=2,\cdots,n})$ is exactly the same as $\left\{\varrho | \varrho \leftarrow \mathcal{G}_{\beta}^{P1^{'}}(1^{\lambda}, n)\right\}$.

Combining lemmas 25 and 2 we prove lemma 3.

C.2 Security reductions for Problems 2 and 2 bis

Proofs of lemmas 4 and 5 can be done as in [28], using following intermediate problem (based on the one of [27], Annex B), and lemmas 26, 27 and 28.

Definition 25. ***Problem 0'*** *is to guess* β*, given* $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \mathbb{B}^*, \boldsymbol{y}^*_\beta, \boldsymbol{f}, g_1^\kappa, g_2^\xi, g_2^{\delta\xi}) \leftarrow \mathcal{G}_\beta^{P0'}(1^\lambda, n)$*, where* $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$ *is an asymmetric bilinear pairing group, and* $\kappa, \xi, \rho, \tau \leftarrow \mathbb{Z}_p^*, \delta, \sigma, \omega \leftarrow \mathbb{Z}_p$*,* $\psi = \kappa \cdot \xi$*,* $(\mathbb{B}, \mathbb{B}^*) \leftarrow \mathsf{Dual}(\mathbb{Z}_p^4)$*,* $\hat{\mathbb{B}} = (\boldsymbol{b}_1, \boldsymbol{b}_3, \boldsymbol{b}_4)$*,* $\boldsymbol{y}_0^* = g_2^{\delta \boldsymbol{b}_1^* + \sigma \boldsymbol{b}_4^*}$*,* $\boldsymbol{y}_1^* = g_2^{\delta \boldsymbol{b}_1^* + \rho \boldsymbol{b}_2^* + \sigma \boldsymbol{b}_4^*}$*,* $\boldsymbol{f} = g_1^{\omega \boldsymbol{b}_1 + \tau \boldsymbol{b}_2}$*.*
For a probabilistic adversary $\mathcal{B}$*, the advantage of* $\mathcal{B}$ *for Problem 0' is defined as*

$$\mathsf{Adv}_{\mathcal{B}}^{P0'}(\lambda) = \left| \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_0^{P0'}(1^\lambda, n)\right] - \Pr\left[\mathcal{B}(1^\lambda, \varrho) \to 1 | \varrho \leftarrow \mathcal{G}_1^{P0'}(1^\lambda, n)\right] \right|$$

Lemma 26. *For any adversary* $\mathcal{D}$*, there is a probabilistic machine* $\mathcal{E}$*, whose running time is essentially the same as that of* $\mathcal{E}$*, such that for any security parameter* λ*,*
$\mathsf{Adv}_{\mathcal{D}}^{P0'}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLin2}}(\lambda) + 5/p$*.*

Proof. Given a $\mathsf{XDLin2}$ instance $g_1, g_1^\xi, g_1^\kappa, g_1^{\delta\xi}, g_1^{\sigma\kappa}, g_2, g_2^\xi, g_2^\kappa, g_2^{\delta\xi}$, $g_2^{\sigma\kappa}, Y_\beta$, where $Y_\beta = g_2^{\delta+\sigma}$ or g_2^z where $z \leftarrow \mathbb{Z}_p$ is chosen randomly, $\mathcal{E}$ calculates $g_T = e(g_1^\kappa, g_2^\xi)$. $\mathcal{E}$ sets 4×4 matrices $\boldsymbol{\Pi}^*, \boldsymbol{\Pi}$ as follows:

$$\boldsymbol{\Pi}^* = \begin{pmatrix} \xi & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & \kappa & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \boldsymbol{\Pi} = \begin{pmatrix} \kappa & 0 & 0 & 0 \\ -\kappa & -\xi & 0 & \kappa\xi \\ 0 & \xi & 0 & 0 \\ 0 & 0 & \kappa\xi & 0 \end{pmatrix}.$$

Then, $\boldsymbol{\Pi}.(\boldsymbol{\Pi}^*)^\top = \kappa\xi \cdot Id_4$. By using $\boldsymbol{\Pi}, \boldsymbol{\Pi}^*$, $\mathcal{E}$ sets

$$\begin{aligned} &\boldsymbol{u}_1^* = (g_2^\xi, 0, 0, g_2), \boldsymbol{u}_2^* = (0, 0, 0, g_2), \qquad \boldsymbol{u}_3^* = (0, g_2^\kappa, 0, g_2), \boldsymbol{u}_4^* = (0, 0, g_2, 0), \\ &\boldsymbol{u}_1 = (g_1^\kappa, 0, 0, 0), \ \boldsymbol{u}_2 = (g_1^{-\kappa}, g_1^{-\xi}, 0, g_1^{\kappa\xi}), \boldsymbol{u}_3 = (0, g_1^\xi, 0, 0), \ \boldsymbol{u}_4 = (0, 0, g_1^{\kappa\xi}, 0) \end{aligned}$$

$\mathcal{E}$ can computes $\boldsymbol{u}_1^*, \boldsymbol{u}_2^*, \boldsymbol{u}_3^*, \boldsymbol{u}_4^*, \boldsymbol{u}_1, \boldsymbol{u}_3$ from the above $\mathsf{XDLin2}$ instance above. Let bases $\mathbb{U} = (\boldsymbol{u}_i)_{i \in [\![1,4]\!]}$ of $\mathbb{G}_1^4$ and $\mathbb{U}^* = (\boldsymbol{u}_i^*)_{\in [\![1,4]\!]}$ of $\mathbb{G}_2^4$. $\mathcal{E}$ chooses $\eta, \phi \leftarrow \mathbb{Z}_p$ such that $\eta \neq 0$, and sets

$$\boldsymbol{v} = (g_1^\phi, g_1^{-\eta}, 0, g_1^{\eta\kappa}), \ \boldsymbol{w}_\beta^* = (g_2^{\delta\xi}, g_2^{\sigma\kappa}, 0, Y_\beta)$$

$\mathcal{E}$ generates random linear transformation $\boldsymbol{W} = (w_{i,j})_{i,j \in [\![1,4]\!]} \leftarrow GL(4, \mathbb{Z}_p)$, $\boldsymbol{Z} = (\boldsymbol{W}^\top)^{-1} = (z_i, j)_{i,j = \in [\![1,4]\!]}$, then calculates $\boldsymbol{b}_i = \sum_{j=1}^4 w_{i,j} \boldsymbol{u}_i$ for $i = 1, 3$, $\boldsymbol{b}_i^* = \sum_{j=1}^4 z_{i,j} \boldsymbol{u}_i^*$ for $i \in [\![1, 4]\!]$, $\hat{\mathbb{B}} = (\boldsymbol{b}_1, \boldsymbol{b}_3, \boldsymbol{b}_4)$, $\mathbb{B}^* = (\boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \boldsymbol{b}_3^*, \boldsymbol{b}_4^*)$, $\boldsymbol{f} = \boldsymbol{W} \cdot \boldsymbol{v}$, $\boldsymbol{y}_\beta^* = \boldsymbol{Z} \cdot \boldsymbol{w}_\beta^*$.

$\mathcal{E}$ then gives $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \mathbb{B}^*, \boldsymbol{y}_\beta^*, \boldsymbol{f}, g_1^\kappa, g_2^\xi, g_2^{\delta\xi})$ where $g_2^\xi, g_2^{\delta\xi}$ are contained in the $\mathsf{XDLin2}$ instance, and outputs $\beta' \in \{0, 1\}$ if $\mathcal{D}$ outputs β'.

If we set $\tau = \xi^{-1}\eta$, $\omega = \tau + \kappa^{-1}\phi$ then $\kappa \neq 0$ (since $\eta \neq 0$),

$$\begin{aligned} &\boldsymbol{v} = (g_1^\phi, g_1^{-\eta}, 0, g_1^{\eta\kappa}) = (g_1^{(\omega-\tau)\kappa}, g_1^{-\tau\xi}, 0, g_1^{\tau\kappa\xi}) = \boldsymbol{u}_1^\omega + \boldsymbol{u}_2^\tau \\ &\boldsymbol{f} = W \cdot \boldsymbol{v} = W \cdot (\boldsymbol{u}_1^\omega + \boldsymbol{u}_2^\tau) = g_1^{\omega \boldsymbol{b}_1 + \tau \boldsymbol{b}_2} \end{aligned}$$

If $\beta = 0$, i.e. $Y_\beta = Y_0 = g_2^{\delta+\sigma}$, then

$$\begin{aligned}
&\boldsymbol{w}_0^* = (g_2^{\delta\xi}, g_2^{\sigma\kappa}, 0, g_2^{\delta+\sigma}) = \boldsymbol{u}_1^{*\delta} + \boldsymbol{u}_4^{*\sigma}\\
&\boldsymbol{y}_0^* = \boldsymbol{Z} \cdot \boldsymbol{w}_0^* = \boldsymbol{Z} \cdot (\boldsymbol{u}_1^{*\delta} + \boldsymbol{u}_4^{*\sigma}) = (\delta \boldsymbol{b}_1^* + \sigma \boldsymbol{b}_4^*).
\end{aligned}$$

Therefore, the distribution of $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \mathbb{B}^*, \boldsymbol{y}_0^*, \boldsymbol{f},\ g_1^\kappa, g_2^\xi, g_2^{\delta\xi})$ is exactly the same as $\left\{\varrho | \varrho \leftarrow \mathcal{G}_0^{P0'}(1^\lambda)\right\}$ when $\kappa \neq 0$ and $\xi \neq 0$, i.e. except with probability $2/p$.
If $\beta = 1$, i.e. $Y_\beta = Y_1 = g_2^\psi$ is uniformly distributed in $\mathbb{G}_2$, we set $\rho = \psi_\delta - \sigma$. Then

$$\begin{aligned}
&\boldsymbol{w}_1^* = (g_2^{\delta\xi}, g_2^{\sigma\kappa}, 0, g_2^{\delta+\rho+\sigma}) = \boldsymbol{u}_1^{*\delta} + \boldsymbol{u}_2^{*\rho} + \boldsymbol{u}_4^{*\sigma}\\
&\boldsymbol{y}_1^* = \boldsymbol{Z} \cdot \boldsymbol{w}_1^* = \boldsymbol{Z}(\boldsymbol{u}_1^{*\delta} + \boldsymbol{u}_2^{*\rho} + \boldsymbol{u}_4^{*\sigma}) = (\delta \boldsymbol{b}_1^* + \rho \boldsymbol{b}_2^* + \sigma \boldsymbol{b}_4^*).
\end{aligned}$$

Therefore, the distribution of $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \mathbb{B}^*, \boldsymbol{y}_1^*, \boldsymbol{f},\ g_1^\kappa, g_2^\xi, g_2^{\delta\xi})$ is exactly the same as $\left\{\varrho | \varrho \leftarrow \mathcal{G}_1^{P0'}(1^\lambda)\right\}$ when $\kappa \neq 0, \xi \neq 0$ and $\rho \neq 0$, i.e. except with probability $3/p$.

Therefore, $\mathsf{Adv}_{\mathcal{D}}^{P0'}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLin2}}(\lambda) + 5/p$.

Lemma 27. *For any adversary $\mathcal{C}$, there is a probabilistic machine $\mathcal{D}$, whose running time is essentially the same as that of $\mathcal{C}$, such that for any security parameter λ,*
$\mathsf{Adv}_{\mathcal{C}}^{P2}(\lambda) = \mathsf{Adv}_{\mathcal{D}}^{P0'}(\lambda)$.

Proof. $\mathcal{D}$ is given a Problem 0' instance $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \mathbb{B}^*, \boldsymbol{y}_\beta^*, \boldsymbol{f}, g_1^\kappa, g_2^\xi, g_2^{\delta\xi})$ and generates random linear transformation $\boldsymbol{W} = (w_{i,j})_{i,j \in [\![1,4]\!]} \leftarrow GL(4, \mathbb{Z}_p)$, $\boldsymbol{Z} = (\boldsymbol{W}^\top)^{-1} = (z_i, j)_{i,j \in [\![1,4]\!]}$, then sets for $i \in [\![1, n]\!]$:

$$\begin{aligned}
g_1^{\boldsymbol{d}_0} &= \boldsymbol{W} \cdot (g_1^\kappa, \boldsymbol{0}^{4n+1}), & g_2^{\boldsymbol{d}_0^*} &= \boldsymbol{Z} \cdot (g_2^\xi, \boldsymbol{0}^{4n+1})\\
g_1^{\boldsymbol{d}_i} &= W \cdot (0, \boldsymbol{0}^{4(i-1)}, \boldsymbol{b}_1, \boldsymbol{0}^{4(n-i)}, 0), & g_2^{\boldsymbol{d}_i^*} &= \boldsymbol{Z} \cdot (0, \boldsymbol{0}^{4(i-1)}, \boldsymbol{b}_1^*, \boldsymbol{0}^{4(n-i)}, 0)\\
g_1^{\boldsymbol{d}_{n+i}} &= \boldsymbol{W} \cdot (0, \boldsymbol{0}^{4(i-1)}, \boldsymbol{b}_2, \boldsymbol{0}^{4(n-i)}, 0), & g_2^{\boldsymbol{d}_{n+i}^*} &= \boldsymbol{Z} \cdot (0, \boldsymbol{0}^{4(i-1)}, \boldsymbol{b}_2^*, \boldsymbol{0}^{4(n-i)}, 0)\\
g_1^{\boldsymbol{d}_{2n+i}} &= \boldsymbol{W} \cdot (0, \boldsymbol{0}^{4(i-1)}, \boldsymbol{b}_3, \boldsymbol{0}^{4(n-i)}, 0), & g_2^{\boldsymbol{d}_{2n+i}^*} &= \boldsymbol{Z} \cdot (0, \boldsymbol{0}^{4(i-1)}, \boldsymbol{b}_3^*, \boldsymbol{0}^{4(n-i)}, 0)\\
g_1^{\boldsymbol{d}_{3n+i}} &= \boldsymbol{W} \cdot (0, \boldsymbol{0}^{4(i-1)}, \boldsymbol{b}_4, \boldsymbol{0}^{4(n-i)}, 0), & g_2^{\boldsymbol{d}_{3n+i}^*} &= \boldsymbol{Z} \cdot (0, \boldsymbol{0}^{4(i-1)}, \boldsymbol{b}_4^*, \boldsymbol{0}^{4(n-i)}, 0)\\
g_1^{\boldsymbol{d}_{4n+1}} &= \boldsymbol{W} \cdot (\boldsymbol{0}^{4n+1}, g_1^\kappa), & g_2^{\boldsymbol{d}_{4n+1}^*} &= \boldsymbol{Z} \cdot (\boldsymbol{0}^{4n+1}, g_2^\xi)
\end{aligned}$$

Note 12. Again we cannot give directly $(\mathbb{D}, \mathbb{D}^*)$ that is why we give them in the exponent of g_1 and g_2 respectively.

Then $\mathbb{D} = (\boldsymbol{d}_i)_{i \in [\![0,4n+1]\!]}$ and $\mathbb{D}^* = (\boldsymbol{d}_i^*)_{i \in [\![0,4n+1]\!]}$ are dual orthonormal bases. Then, $\mathcal{D}$ sets, for $i \in [\![1, n]\!]$,

$$\begin{aligned}
&\boldsymbol{p}_{\beta,i}^* = (W^{-1})^\top (0, \boldsymbol{0}^{4(i-1)}, \boldsymbol{y}_\beta^*, \boldsymbol{0}^{4(n-i)}, 0)\\
&\boldsymbol{g}_i = W \cdot (0, \boldsymbol{0}^{4(i-1)}, \boldsymbol{f}, \boldsymbol{0}^{4(n-i)}, 0)
\end{aligned}$$

$\mathcal{D}$ can compute $\hat{\mathbb{D}} = (\boldsymbol{d}_0, \cdots, \boldsymbol{d}_n, \boldsymbol{d}_{2n+1}, \cdots, \boldsymbol{d}_{4n+1})$ from $\hat{\mathbb{B}}, \mathbb{B}, g_1^{\kappa}, g_2^{\xi}$. $\mathcal{D}$ then gives $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{D}}, \mathbb{D}^*, \left\{\boldsymbol{p}^*_{\beta,i}, \boldsymbol{g}_i\right\}_{i \in [\![1,n]\!]})$ to $\mathcal{C}$, and outputs $\beta^{'} \in \{0,1\}$ if $\mathcal{C}$ outputs $\beta^{'}$.
We can see that for $i \in [\![1,n]\!]$, $\boldsymbol{p}_{0,i} = g_2^{\delta \boldsymbol{d}^*_i + \sigma \boldsymbol{d}^*_{3n+i}}$, $\boldsymbol{p}_{1,i} = g_2^{\delta \boldsymbol{d}^*_i + \rho \boldsymbol{d}^*_{n+i} + \sigma \boldsymbol{d}^*_{3n+i}}$, $\boldsymbol{g}_i = g_1^{\omega \boldsymbol{d}_i + \tau \boldsymbol{d}_{n+i}}$.
Therefore the distribution of $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{D}}, \mathbb{D}^*, \left\{\boldsymbol{p}^*_{\beta,i}, \boldsymbol{g}_i\right\}_{i \in [\![1,n]\!]})$ is exactly the same as $\left\{\rho | \rho \leftarrow \mathcal{G}_{\beta}^{P2}(1^{\lambda})\right\}$.

Combining lemmas 26 and 27 we prove lemma 4. To complete the proof of lemma 5, we prove lemma 28.

Lemma 28. *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{A}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{P2b}_{\mathcal{B}}(\lambda) = \mathsf{Adv}^{P2}_{\mathcal{A}}(\lambda)$.*

Proof. Let $\mathcal{B}$ be an adversary against Problem 2 bis, that wins with non negligible advantage. We construct $\mathcal{A}$ an adversary against Problem 2.

$\mathcal{A}$ is given a Problem 2 instance $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \mathbb{B}^*, \{\boldsymbol{h}_{\beta,i}, \boldsymbol{e}_i\}_{i \in [\![1,n]\!]})$. $\mathcal{A}$ generates $\boldsymbol{U} = (u_{i,j}) \leftarrow \mathsf{GL}(n, \mathbb{Z}_p)$ and calculates $\boldsymbol{Z} = (z_{i,j}) = (U^{-1})^{\top}$. Then $\mathcal{A}$ calculates $\{\boldsymbol{d}_{n+1}, \cdots, \boldsymbol{d}_{2n}\}$ and $\{\boldsymbol{d}^*_{n+1}, \cdots, \boldsymbol{d}^*_{2n}\}$ from $\{\boldsymbol{b}_{n+1}, \cdots, \boldsymbol{b}_{2n}\}$ and $\{\boldsymbol{b}^*_{n+1}, \ \cdots, \boldsymbol{b}^*_{2n}\}$ respectively as

$$\boldsymbol{d}_{n+j} = \textstyle\sum_{i=1}^{n} z_{i,j} \boldsymbol{b}_{n+i}, \ \ \boldsymbol{d}^*_{n+j} = \textstyle\sum_{i=1}^{n} u_{i,j} \boldsymbol{b}^*_{n+i}$$

for $j \in [\![1,n]\!]$. Then, $\boldsymbol{b}_{n+i} = \sum_{j=1}^{n} u_{i,j} \boldsymbol{d}_{n+j}$, $\boldsymbol{b}^*_{n+i} = \sum_{j=1}^{n} z_{i,j} \boldsymbol{d}^*_{n+j}$ for $i \in [\![1,n]\!]$ since $UZ^{\top} = I_n$. $\mathcal{A}$ picks $\boldsymbol{r}_i = r_{i,1} \boldsymbol{b}^*_{3n+1} + \cdots + r_{i,n} \boldsymbol{b}^*_{4n}$ and for $i \in [\![1,n]\!]$ we have

$$\boldsymbol{h}^*_{0,i} \cdot g_2^{\boldsymbol{r}_i} = g_2^{\delta \boldsymbol{b}^*_i + \delta_0 \boldsymbol{b}^*_{3n+i} + \boldsymbol{r}_i} = g_2^{\delta \boldsymbol{d}^*_i + \sum_{j=1}^{n} \tilde{r}_i \boldsymbol{d}^*_{3n+j}}$$

$$\boldsymbol{h}^*_{1,i} \cdot g_2^{\boldsymbol{r}_i} = g_2^{\delta \boldsymbol{b}^*_i + \tau \boldsymbol{b}_{n+i} + \delta_0 \boldsymbol{b}^*_{3n+i} + \boldsymbol{r}_i} = g_2^{\delta \boldsymbol{d}^*_i + \tau \sum_{j=1}^{n} z_{i,j} \boldsymbol{d}^*_{n+j} + \sum_{j=1}^{n} \tilde{r}_i \boldsymbol{d}^*_{3n+j}}$$

$$\boldsymbol{e}_i = g_1^{\omega \boldsymbol{b}_i + \sigma \boldsymbol{b}_{n+i}} = g_1^{\omega \boldsymbol{d}_i + \sigma \sum_{j=1}^{n} u_{i,j} \boldsymbol{d}_{n+i}}$$

where $\tilde{r}_i = r_{i,j}$ if $i \neq j$ and $\tilde{r} = r_{i,i} + \delta_0$ otherwise.

Let $\mathbb{D} = (\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_n, \boldsymbol{d}_{n+1}, \cdots, \boldsymbol{d}_{2n}, \boldsymbol{b}_{2n+1}, \cdots, \boldsymbol{b}_{4n+1})$, $\mathbb{D}^* = (\boldsymbol{b}^*_0, \boldsymbol{b}^*_1, \cdots, \boldsymbol{b}^*_n, \boldsymbol{d}^*_{n+1}, \cdots, \boldsymbol{d}^*_{2n}, \boldsymbol{d}^*_{2n+1}, \cdots, \boldsymbol{b}^*_{4n+1})$ and $\hat{\mathbb{D}} = (\boldsymbol{d}_0, \cdots, \boldsymbol{d}_n, \boldsymbol{d}_{2n+1}, \cdots, \boldsymbol{d}_{4n+1})$.

$\mathcal{A}$ gives $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{D}}, \mathbb{D}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{r}_i\}_{i \in [\![1,n]\!]})$ to $\mathcal{B}$ as a Problem 2 bis instance, and outputs $\beta^{'} \in \{0,1\}$ if $\mathcal{B}$ outputs $\beta^{'}$.
By construction, the distribution of $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{D}}, \mathbb{D}^*, \{\boldsymbol{h}^*_{\beta,i}, \boldsymbol{r}_i\}_{i \in [\![1,n]\!]})$ is exactly the same as $\left\{\varrho | \varrho \leftarrow \mathcal{G}_{\beta}^{P2b}(1^{\lambda}, n)\right\}$.

Combining lemmas 28 and 4 we prove lemma 5.

C.3 Security reduction for Problem 3

To prove lemma 6 we use the following experiment and lemmas 29 and 30, as in [28] (Annex B).

Definition 26. ***Experiment 3**-α ($\alpha = 0, 1, 2$). We define the Exp-3-α instance generator, denoted $\mathcal{G}_{\alpha}^{Exp-3}(1^{\lambda}, n)$, where, $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T,\ g_1, g_2, e, p)$ is an asymmetric bilinear prime order group, $(\mathbb{B}, \mathbb{B}^*) \leftarrow \mathsf{Dual}(\mathbb{Z}_p^{4n+2})$ are dual orthonormal bases,*

$$\hat{\mathbb{B}} = (\boldsymbol{b}_0, \cdots, \boldsymbol{b}_n, \boldsymbol{b}_{3n+1}, \cdots, \boldsymbol{b}_{4n+1}),\ \hat{\mathbb{B}}^* = (\boldsymbol{b}_0^*, \cdots, \boldsymbol{b}_n^*, \boldsymbol{b}_{3n+1}^*, \cdots, \boldsymbol{b}_{4n+1}^*),$$

$\tau, \tau^{'}, \delta_0, \omega^{'}, \omega^{''}, \kappa^{'}, \kappa^{''} \leftarrow \mathbb{Z}_p$, and for $i \in [\![1, n]\!]$

$$\begin{array}{lll} \boldsymbol{h}_{0,i}^* = g_2^{\tau \boldsymbol{b}_{n+i}^* + \delta_0 \boldsymbol{b}_{3n+i}^*} & \boldsymbol{h}_{1,i}^* = g_2^{\tau \boldsymbol{b}_{n+i}^* + \tau^{'} \boldsymbol{b}_{2n+i}^* + \delta_0 \boldsymbol{b}_{3n+i}^*} & \boldsymbol{h}_{2,i}^* = g_2^{\tau^{'} \boldsymbol{b}_{2n+i}^* + \delta_0 \boldsymbol{b}_{3n+i}^*} \\ \boldsymbol{e}_i = g_1^{\omega^{'} \boldsymbol{b}_{n+i} + \omega^{''} \boldsymbol{b}_{2n+i}} & \boldsymbol{f}_i = g_1^{\kappa^{'} \boldsymbol{b}_{n+i} + \kappa^{''} \boldsymbol{b}_{2n+i}} & \end{array}$$

$$\textit{return}\ (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, p, \hat{\mathbb{B}}, \hat{\mathbb{B}}^*, \left\{\boldsymbol{h}_{\alpha,i}^*, \boldsymbol{e}_i, \boldsymbol{f}_i\right\}_{i \in [\![1,n]\!]}).$$

For a probabilistic adversary $\mathcal{B}$, we define 3 experiments $Exp_{\mathcal{B}}^{3-\alpha}$ ($\alpha = 0, 1, 2$) as follows: a) $\mathcal{C}$ is given $\varrho \leftarrow \mathcal{G}_{\alpha}^{Exp-3}(1^{\lambda}, n)$. b) Output $\beta^{'} \leftarrow \mathcal{B}(1^{\lambda}, \varrho)$.

Lemma 29. *For any adversary $\mathcal{B}$, for any security parameter λ,*

$$|\Pr\left[Exp_{\mathcal{B}}^{3-0}(\lambda) \rightarrow 1\right] - \Pr\left[Exp_{\mathcal{B}}^{3-1}(\lambda) \rightarrow 1\right]| \leq 1/p.$$

Proof. Let $\theta \leftarrow \mathbb{Z}_p$. If we set, for $i \in [\![1, n]\!]$, $\boldsymbol{d}_{2n+i} = \boldsymbol{b}_{2n+i} - \theta \boldsymbol{b}_{n+i}$, $\boldsymbol{d}_{n+i}^* = \boldsymbol{b}_{n+i}^* + \theta \boldsymbol{b}_{2n+i}^*$, then

$$\begin{array}{ll} \boldsymbol{h}_{0,i}^* = g_2^{\tau \boldsymbol{b}_{n+i}^* + \delta_0 \boldsymbol{b}_{3n+i}^*} & = g_2^{\tau \boldsymbol{d}_{n+i}^* + \tau^{'} \boldsymbol{d}_{2n+i}^* + \delta_0 \boldsymbol{d}_{3n+i}^*} \\ \boldsymbol{e}_i = g_1^{\omega^{'} \boldsymbol{b}_{n+i} + \omega^{''} \boldsymbol{b}_{2n+i}} & = g_1^{\tilde{\omega}^{'} \boldsymbol{d}_{n+i} + \omega^{''} \boldsymbol{d}_{2n+i}} \\ \boldsymbol{f}_i = g_1^{\kappa^{'} \boldsymbol{b}_{n+i} + \kappa^{''} \boldsymbol{b}_{2n+i}} & = g_1^{\tilde{\kappa}^{'} \boldsymbol{d}_{n+i} + \omega^{''} \boldsymbol{d}_{2n+i}} \end{array}$$

where $\tau^{'} = -\theta\tau, \tilde{\omega}^{'} = \omega^{'} + \theta\omega^{''}$ and $\tilde{\kappa}^{'} = \kappa^{'} + \theta\kappa^{''}$, which are independently and uniformly distributed since $\theta, \omega^{'}, \kappa^{'} \leftarrow \mathbb{Z}_p$ except for the case $\tau = 0$. That is, the joint distribution for Exp.3-0 and that for Exp.3-1 are equivalent except with probability $1/p$.

Lemma 30. *For any adversary $\mathcal{B}$, there is a probabilistic machine $\mathcal{C}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ,*

$$\left|\left|\Pr\left[Exp_{\mathcal{B}}^{3-1}(\lambda) \rightarrow 1\right] - \Pr\left[Exp_{\mathcal{B}}^{3-2}(\lambda) \rightarrow 1\right]\right| - \mathsf{Adv}_{\mathcal{C}}^{P2}(\lambda)\right| \leq 1/p.$$

Proof. To prove lemma 30, we construct a probabilistic machine $\mathcal{C}$ against Problem 2 using a machine $\mathcal{B}$ distinguishing the experiment $\mathsf{Exp}_{\mathcal{B}}^{3-1}$ from $\mathsf{Exp}_{\mathcal{B}}^{3-2}$ as a black box as follows: $\mathcal{C}$ is given a Problem 2 instance $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \hat{\mathbb{B}}, \mathbb{B}^*, \left\{\boldsymbol{h}_{\beta,i}^*, \boldsymbol{e}_i\right\}_{i \in \llbracket 1,n \rrbracket})$, and sets $\boldsymbol{f}_i = \eta_1 \boldsymbol{b}_i + \eta_2 \boldsymbol{e}_i$ for $i \in \llbracket 1, n \rrbracket$,

$$\mathbb{D} = (\boldsymbol{b}_0, \boldsymbol{b}_{2n+1}, \cdots, \boldsymbol{b}_{3n}, \boldsymbol{b}_{n+1}, \cdots, \boldsymbol{b}_{2n}, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_n, \boldsymbol{b}_{3n+1}, \cdots, \boldsymbol{b}_{4n+1}),$$

$$\mathbb{D}^* = (\boldsymbol{b}_0^*, \boldsymbol{b}_{2n+1}^*, \cdots, \boldsymbol{b}_{3n}^*, \boldsymbol{b}_{n+1}^*, \cdots, \boldsymbol{b}_{2n}^*, \boldsymbol{b}_1^*, \cdots, \boldsymbol{b}_n^*, \boldsymbol{b}_{3n+1}^*, \cdots, \boldsymbol{b}_{4n+1}^*),$$

$$\hat{\mathbb{D}} = (\boldsymbol{d}_0, \cdots, \boldsymbol{d}_n, \boldsymbol{d}_{3n+1}, \cdots, \boldsymbol{d}_{4n+1}) = (\boldsymbol{b}_0, \boldsymbol{b}_{2n+1}, \cdots, \boldsymbol{b}_{3n}, \boldsymbol{b}_{3n+1}, \cdots, \boldsymbol{b}_{4n+1})$$

$$\hat{\mathbb{D}}^* = (\boldsymbol{d}_0^*, \cdots, \boldsymbol{d}_n^*, \boldsymbol{d}_{3n+1}^*, \cdots, \boldsymbol{d}_{4n+1}^*) = (\boldsymbol{b}_0^*, \boldsymbol{b}_{2n+1}^*, \cdots, \boldsymbol{b}_{3n}^*, \boldsymbol{b}_{3n+1}^*, \cdots, \boldsymbol{b}_{4n+1}^*)$$

where $\mathcal{C}$ can calculate $\hat{\mathbb{D}}$ and $\hat{\mathbb{D}}^*$ from a part of the Problem 2 instance, i.e. $(\hat{\mathbb{B}}, \mathbb{B}^*)$, while $\mathcal{C}$ cannot calculate a part of the basis $\mathbb{D}$, i.e., $(\boldsymbol{d}_{n+1}, \cdots, \boldsymbol{d}_{2n})$, from the Problem 2 instance. $\mathcal{C}$ gives $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, p, \hat{\mathbb{D}}, \hat{\mathbb{D}}^*, \left\{\boldsymbol{h}_{\beta,i}^*, \boldsymbol{e}_i, \boldsymbol{f}_i\right\}_{i \in \llbracket 1,n \rrbracket})$ to $\mathcal{B}$, and receives $\beta^{'} \in \{0, 1\}$. $\mathcal{C}$ then outputs $\beta^{'}$. Then,

$$\boldsymbol{h}_{0,i}^* = g_2^{\delta \boldsymbol{b}_i^* + \delta_0 \boldsymbol{b}_{3n+i}^*} = g_2^{\delta \boldsymbol{d}_{2n+i}^* + \delta_0 \boldsymbol{d}_{3n+i}^*}$$

$$\boldsymbol{h}_{1,i}^* = g_2^{\delta \boldsymbol{b}_i^* + \tau \boldsymbol{b}_{n+i}^* \delta_0 \boldsymbol{b}_{3n+i}^*} = g_2^{\tau \boldsymbol{d}_{n+i}^* + \delta \boldsymbol{d}_{2n+i}^* + \delta_0 \boldsymbol{d}_{3n+i}^*}$$

$$\boldsymbol{e}_i = g_1^{\omega \boldsymbol{b}_i + \sigma \boldsymbol{b}_{n+i}} = g_1^{\sigma \boldsymbol{d}_{n+i} + \omega \boldsymbol{d}_{2n+i}}$$

$$\boldsymbol{f}_i = g_1^{\eta_1 \boldsymbol{b}_i + \eta_2 \sigma \boldsymbol{e}_i} = g_1^{\eta_2 \sigma \boldsymbol{d}_{n+i} + (\eta_1 + \eta_2 \sigma) \boldsymbol{d}_{2n+i}}$$

where $\delta, \tau, \omega, \sigma, \eta_1 + \eta_2 \omega$ and $\eta_2 \sigma$ are independently and uniformly distributed in $\mathbb{Z}_p$ since $\delta, \tau, \omega, \sigma, \eta_1, \eta_2 \leftarrow \mathbb{Z}_p$ except for the case $\sigma = 0$.
The above $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, p, \hat{\mathbb{D}}, \hat{\mathbb{D}}^*, \left\{\boldsymbol{h}_{\beta,i}^*, \boldsymbol{e}_i, \boldsymbol{f}_i\right\}_{i \in \llbracket 1,n \rrbracket})$ has the same distribution as the output of the generator $\mathcal{G}_1^{Exp-3}(1^\lambda, n)$ (resp. $\mathcal{G}_2^{Exp-3}(1^\lambda, n)$) when $\beta = 1$ (resp. $\beta = 0$) except with probability $1/p$. This completes the proof of lemma 30.

Now let's prove lemma 6.

Proof. Problem 3 is the hybrid of Experiment $3-0, 3-1$ and $3-2$, i.e., $\mathsf{Adv}_{\mathcal{B}}^{P3}(\lambda) = \left|\Pr\left[\mathsf{Exp}_{\mathcal{B}}^{3-0}(\lambda) \to 1\right] - \Pr\left[\mathsf{Exp}_{\mathcal{B}}^{3-2}(\lambda) \to 1\right]\right|$. Therefore, from lemmas 30, 29 and 4, there exist probabilistic machines $\mathcal{C}, \mathcal{E}$, whose running time are essentially the same as that of $\mathcal{B}$, such that for any security parameter λ,

$$\begin{aligned} &\mathsf{Adv}_{\mathcal{B}}^{P3}(\lambda) = \left|\Pr\left[\mathsf{Exp}_{\mathcal{B}}^{3-0}(\lambda) \to 1\right] - \Pr\left[\mathsf{Exp}_{\mathcal{B}}^{3-2}(\lambda) \to 1\right]\right| \\ &\leq \left|\Pr\left[\mathsf{Exp}_{\mathcal{B}}^{3-0}(\lambda) \to 1\right] - \Pr\left[\mathsf{Exp}_{\mathcal{B}}^{3-1}(\lambda) \to 1\right]\right| + \left|\Pr\left[\mathsf{Exp}_{\mathcal{B}}^{3-1}(\lambda) \to 1\right]\right. \\ &\left. - \Pr\left[\mathsf{Exp}_{\mathcal{B}}^{3-2}(\lambda) \to 1\right]\right| \leq \mathsf{Adv}_{\mathcal{C}}^{P2}(\lambda) + 2/p \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLin2}}(\lambda) + 7/p. \end{aligned}$$

This completes the proof of lemma 6.

D Anonymity

Within the broadcast encryption setting, the anonymity property, introduced in [6], further requests to hide the used user set, and is useful in some practical cases. In this appendix, we prove that our generic construction of (Aug) BE from a pattern hiding WIBE gives us an anonymous scheme.

Anonymous Broadcast encryption.

Definition 27. ***Anonymous BE scheme*** *(ANO-BE) [6, 26]*
We say that a BE scheme is adaptively anonymous *(or satisfies ANO-BE security) if all polynomial time adaptive adversaries $\mathcal{A}$ have at most negligible advantage in the game presented in Figure 14, where $\mathcal{A}$'s advantage is defined as*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{ANO\text{-}BE}}(\lambda) = \left|\Pr\left[b^{'} = b\right] - 1/2\right|.$$

SETUP: challenger $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, 1^N)$ to generate pk and msk, and gives pk to $\mathcal{A}$.
KEY QUERY: $\mathcal{A}$ issues queries to $\mathcal{C}$ for index $i \in [N]$. $\mathcal{C}$ returns $\mathsf{sk}_i \leftarrow \mathsf{KeyGen}(\mathsf{msk}, i)$.
CHALLENGE: $\mathcal{A}$ selects two messages $\mathsf{m}_0, \mathsf{m}_1$ and two distinct sets $S^0, S^1 \subseteq [N]$ of users. We impose the restriction that $\mathcal{A}$ has not issued key queries for any i such that $i \in S^0 \wedge i \notin S^1$ or $i \in S^1 \wedge i \notin S^0$. Further, if there exists an $i \in S^0 \cap S^1$ for which $\mathcal{A}$ has queried the key, then we require that $\mathsf{m}_0 = \mathsf{m}_1$. $\mathcal{A}$ passes $\mathsf{m}_0, \mathsf{m}_1$ and S^0, S^1 to $\mathcal{C}$. The latter picks $b \in \{0, 1\}$ random and computes $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, S^b, \mathsf{m}_b)$ which is returned to $\mathcal{A}$.
KEY QUERY: $\mathcal{A}$ continues to make queries for index $i \in [N]$ with the restriction that $i \notin S^0 \wedge i \notin S^1$ or $i \in S^0 \wedge i \in S^1$ and if $\mathsf{m}_0 \neq \mathsf{m}_1$ then $i \notin S^0 \cap S^1$.
GUESS: $\mathcal{A}$ outputs its guess $b^{'} \in \{0, 1\}$ for b, and wins the game if $b^{'} = b$.

Fig. 14. ANO-BE security game.

Note 13. Many BE schemes require the encryption set S to be publicly given as an input of decryption algorithm. Otherwise even authorized users will not be able to decrypt. However, anonymous schemes does not need the encryption set description as an input for the decryption algorithm.

Our first generic construction gives us a broadcast encryption scheme from a WIBE. Let us show that if the underlying WIBE is also pattern-hiding, then the obtained BE is anonymous.

Theorem 11. *If $\mathcal{WIBE}$ satisfies adaptive (resp. selective) pattern hiding security, then the obtained BE scheme is adaptive (resp. selective) anonymous.*

Proof. Let $\mathcal{B}$ be an adversary against anonymous security, that wins with non negligible advantage. In Figure 15 we construct $\mathcal{A}$, an adversary against pattern hiding security that uses $\mathcal{B}$ and wins with non negligible advantage. Let $\mathcal{C}$ be a challenger.

If all $\mathcal{B}$'s queries satisfy the game constraints, then all $\mathcal{A}$'s queries have the same property. Thus $\mathcal{A}$'s simulation is perfect, and the advantage of $\mathcal{A}$ is the same as $\mathcal{B}$'s. This concludes the proof.

SETUP: $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda, 1^N) \rightarrow (\mathsf{msk}, \mathsf{pk})$ and sends pk to $\mathcal{A}$, who sends it to $\mathcal{B}$.

KEY QUERY: $\mathcal{B}$ chooses $i \in [N]$, sends it to $\mathcal{A}$ who creates the pattern $\boldsymbol{P}^i$ such that for $j \in [\![1, N]\!]$, $P^i_j = 1$ if $i = j$ and $P^i_j = 0$ otherwise. $\mathcal{A}$ sends $\boldsymbol{P}^i$ to $\mathcal{C}$ who runs $\mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{P}^i) \rightarrow \mathsf{sk}_{\boldsymbol{P}^i}$. $\mathcal{A}$ receives $\mathsf{sk}_{\boldsymbol{P}^i}$ and sends it to $\mathcal{B}$ as sk_i.

CHALLENGE: $\mathcal{B}$ chooses message m, two sets S_0, S_1 and sends them to $\mathcal{A}$ who creates patterns $\boldsymbol{P}^0, \boldsymbol{P}^1$ s.t. for $j \in [\![1, N]\!]$, $P^0_j = \star$ if $j \in S_0$, $P^0_j = 0$ otherwise, and $P^1_j = \star$ if $j \in S_1$, $P^1_j = 0$ otherwise. $\mathsf{m}, \boldsymbol{P}^0, \boldsymbol{P}^1$ are sent to $\mathcal{C}$. If for any queried $\boldsymbol{P}^i$ during previous step, $\boldsymbol{P}^i \in_\star \boldsymbol{P}^0 \wedge \boldsymbol{P}^i \notin_\star \boldsymbol{P}^1$ or $\boldsymbol{P}^i \notin_\star \boldsymbol{P}^0 \wedge \boldsymbol{P}^i \in_\star \boldsymbol{P}^1$, $\mathcal{C}$ aborts. Otherwise, it chooses $b \leftarrow \{0, 1\}$ and runs $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \boldsymbol{P}^b, \mathsf{m})$. $\mathcal{C}$ sends ct^* to $\mathcal{A}$, who sends it to $\mathcal{B}$.

KEY QUERY: $\mathcal{B}$ and $\mathcal{A}$ proceeds as in the first KEY QUERY step. If $\boldsymbol{P}^i \in_\star \boldsymbol{P}^0 \wedge \boldsymbol{P}^i \notin_\star \boldsymbol{P}^1$ or $\boldsymbol{P}^i \notin_\star \boldsymbol{P}^0 \wedge \boldsymbol{P}^i \in_\star \boldsymbol{P}^1$, $\mathcal{C}$ aborts, otherwise it runs $\mathsf{KeyDer}(\mathsf{msk}, \boldsymbol{P}^i) \rightarrow \mathsf{sk}_{\boldsymbol{P}^i}$. $\mathsf{sk}_{\boldsymbol{P}^i}$ is sent to $\mathcal{A}$, who sends it to $\mathcal{B}$ as sk_i.

GUESS: $\mathcal{B}$ outputs its guess $b^{'}$ to $\mathcal{A}$, who outputs it as its guess.

Fig. 15. Construction of pattern hiding adversary from anonymous BE adversary.

State of art. In the anonymous BE setting, the Libert *et al.* scheme ([26]) is the best known option so far, even if [25] proposes a slight improvement regarding the ciphertext size. The main practical problem with both constructions is however that each user has to try each element of the ciphertext to find the one he/she can truly decrypt. In 2014, [4] proposed a generic construction of anonymous BT from anonymous BE, and an instantiation based on [26]'s anonymous BE scheme.
Notice that [26] said that achieving shorter than linear size for ciphertext is impossible when considering the used users set description as part of the ciphertext. With our second WIBE and our generic constructions, setting $L = N + 1$, we obtain a new anonymous broadcast encryption scheme. Our scheme does not improve the efficiency of the Libert *et al.* scheme [26], which is the best known so far. In particular, their scheme has pk and sk_i sizes in respectively $O(N)$ and $O(1)$ while in our scheme the same parameters have sizes in $O(N^2)$ and $O(N)$ respectively. Regarding security, their scheme achieves the stronger CCA security in the standard model while we only reach a CPA security. However, in Libert *et al.*'s scheme, each user has to try each element of the ciphertext to find the one he can truly decrypt, while this is not necessary in our construction.

Anonymous Augmented Broadcast encryption. As notice in section **??**, the AugBE scheme obtained from our second WIBE instantiation is actually the first known anonymous AugBE. Eventually, the derivation of our anonymous AugBE to an anonymous BT scheme is quite direct from the generic construction given in [10]. A formal definition of an anonymous BT scheme is quite straightforward from the one of anonymous AugBE and can be found in *e.g.*, [4]. The only existing anonymous BT is the one of [4], which is based on the anonymous BE scheme of [26]: it directly inherit advantages and drawbacks compare to our resulting scheme.