A Framework for Identity-Based Encryption with Almost Tight Security

Nuttapong Attrapadung[*1], Goichiro Hanaoka[†1], and Shota Yamada[‡1]

[1]National Institute of Advanced Industrial Science and Technology (AIST).

June 9, 2015

Abstract

We show a framework for constructing identity-based encryption (IBE) schemes that are (almost) tightly secure in the multi-challenge and multi-instance setting. In particular, we formalize a new notion called *broadcast encoding*, analogously to encoding notions by Attrapadung (Eurocrypt '14) and Wee (TCC '14). We then show that it can be converted into such an IBE. By instantiating the framework using several encoding schemes (new or known ones), we obtain the following:

- We obtain (almost) tightly secure IBE in the multi-challenge, multi-instance setting, both in composite and prime-order groups. The latter resolves the open problem posed by Hofheinz et al (PKC '15).
- We obtain the first (almost) tightly secure IBE with sub-linear size public parameters (master public keys). In particular, we can set the size of the public parameters to constant at the cost of longer ciphertexts. This gives a partial solution to the open problem posed by Chen and Wee (Crypto '13).

By applying (a variant of) the Canetti-Halevi-Katz transformation to our schemes, we obtain several CCA-secure PKE schemes with tight security in the multi-challenge, multi-instance setting. One of our schemes achieves very small ciphertext overhead, consisting of less than 12 group elements. This significantly improves the state-of-the-art construction by Libert et al. (in ePrint Archive) which requires 47 group elements. Furthermore, by modifying one of our IBE schemes obtained above, we can make it anonymous. This gives the first anonymous IBE whose security is almost tightly shown in the multi-challenge setting.

Keywords. Tight security reduction, identity-based encryption, multi-challenge security, chosen ciphertext security.

1 Introduction

1.1 Backgrounds

In the context of provable security, we reduce the security of a given scheme to the hardness of a computational problem, in order to gain confidence in the security of the scheme. Namely, we assume an adversary $\mathcal{A}$ who breaks the scheme and then show another adversary $\mathcal{B}$ who solves the (assumed) hard problem using $\mathcal{A}$. Such a reduction should be as *tight* as possible, in the sense that $\mathcal{B}$'s success probability is as large as $\mathcal{A}$. In this paper, we mostly focus on the tight security reduction in identity-based encryption (IBE) [52].

[*]n.attrapadung@aist.go.jp
[†]hanaoka-goichiro@aist.go.jp
[‡]yamada-shota@aist.go.jp

IBE is an advanced form of public key encryption in which one can encrypt a message for a user identity, rather than a public key. The first fully secure (or often called, adaptively secure) construction in the standard model was given in [10]. Later, further developments were made [53, 31, 7, 54]. All the above mentioned papers only deal with the single-challenge, single-instance case. Since it is known that the security in the (much more realistic) multi-challenge and multi-instance setting can be reduced to the security in the single-challenge and single-instance setting [6], these schemes are secure in the former setting in asymptotic sense. However, this reduction incurs $O(\mu Q_c)$ security loss, where Q_c is the number of challenge queries made by the adversary and μ is the number of instances. Since all the above schemes already loose at least $O(Q_k)$ security in the reductions, where Q_k is the number of key extraction queries made by $\mathcal{A}$, theses schemes loose at least $O(\mu Q_c Q_k)$ security in total.

Recently and somewhat surprisingly, Chen and Wee [18, 20] showed the first IBE scheme (CW scheme) whose reduction cost is independent of Q_k, resolving an important open question posed in [53]. Subsequently, Blazy et al. [8] were able to obtain anonymous IBE and hierarchical IBE with the same security guarantee. The drawback of these schemes is its large public parameters (master public keys): It is proportional to the security parameter and thus rather large. Note that they only consider the single-challenge and single-instance setting. Very recently, further important development was made by Hofheinz, Koch, and Striecks [33] who extended the proof technique of Chen and Wee in a novel way and proposed the first IBE scheme (HKS scheme) whose reduction cost is independent from all of μ, Q_c, and Q_k. However, they only give a construction in composite-order groups and explicitly mention that the construction in prime-order groups remains open. We focus on the following two important open problems in this paper:

- *Can we construct a fully, (almost) tightly secure IBE scheme in the multi-challenge and multi-instance setting from a static assumption in the prime-order groups?*
- *Can we construct a fully, (almost) tightly secure IBE scheme from a static assumption with constant-size public parameters even in the single-challenge and single-instance setting?*

1.2 Our Results

New Tightly-Secure IBE Schemes. In this paper, to tackle the above problems, we revisit the proof technique in [18] and [33] and propose a framework for constructing almost tightly secure IBE. The almost tight security means that the reduction cost is independent from μ, Q_c, and Q_k, and is a small polynomial in the security parameter. In particular, we formalize the notion of broadcast encoding analogously to Attrapadung [4] and Wee [55]. Then we show that it can be converted into fully, (almost) tightly secure IBE scheme, in the multi-challenge and multi-instance setting. We propose such conversions both in prime-order and composite-order groups. Furthermore, we propose two broadcast encoding schemes satisfying our requirement. By instantiating our generic conversion with these schemes, we obtain several new IBE schemes. In particular,

- We obtain the first IBE scheme in *prime-order groups* with almost tight security in the multi-challenge and multi-instance setting. The security of our scheme can be shown under the decisional linear (DLIN) assumption. This resolves the first question above.
- We obtain the first IBE scheme with almost tight security in the multi-challenge and multi-instance setting and with *sub-linear public parameter-size* (but at the cost of larger ciphertext-size). An IBE scheme with almost tight security and sub-linear public parameter size is not known, even in the single-challenge setting. This partially answers the second question above.

Application to Chosen-Ciphertext Secure Public Key Encryption. By applying a variant of Canetti-Halevi-Katz transformation to the new IBE schemes, we obtain several new chosen-ciphertext (CCA) secure public key encryption (PKE) schemes. The conversion is tightness-preserving, namely, if the original IBE

is tightly secure in the multi-challenge and multi-instance setting, the resulting PKE scheme is also tightly secure in the same setting. One of our schemes achieves very compact ciphertext size. The ciphertext overhead of the scheme only consists of 10 group elements and 2 elements in $\mathbb{Z}_p$. This is much shorter than the state-of-the-art construction of PKE scheme with the same security guarantee [36]: their scheme requires 47 group elements.

Extension to Anonymous IBE. Furthermore, by modifying one of the new IBE schemes obtained above, we obtain the first anonymous IBE scheme with (almost) tight security reduction in the multi-challenge settings for the first time. The security proof is done by carefully combining information-theoretic argument due to Chen et al. [17] and a computational argument.

See Table 1 for overview of our schemes.

Table 1: Comparison of Almost Tight IBE from Static Assumptions

Schemes	$\lvert\mathsf{pp}\rvert + \lvert\mathsf{mpk}\rvert$	$\lvert\mathsf{CT}\rvert$	$\lvert\mathsf{sk}_{\mathsf{ID}}\rvert$	Anon?	Multi-challenge?	Underlying group	Security Assumption
CW13 [18]	$O(\kappa)$	$O(1)$	$O(1)$	No	No	Composite	SGD, CW
HKS15 [33]	$O(\kappa)$	$O(1)$	$O(1)$	No	Yes	Composite	SGD, HKS
Ours: $\Phi_{\mathsf{cc}}^{\mathsf{comp}}$	$O(\kappa)$	$O(1)$	$O(1)$	No	Yes	Composite	SGD, Problem 5
Ours: $\Phi_{\mathsf{slp}}^{\mathsf{comp}}$	$O(\kappa^{1-c})$	$O(\kappa^c)$	$O(\kappa^c)$	No	Yes	Composite	SGD, Problem 13, 14
CW13 [18]†	$O(\kappa)$	$O(1)$	$O(1)$	No	No	Prime	DLIN
BKP14 [8]*†	$O(\kappa)$	$O(1)$	$O(1)$	Yes	No	Prime	DLIN
Ours: $\Phi_{\mathsf{cc}}^{\mathsf{prime}}$	$O(\kappa)$	$O(1)$	$O(1)$	No	Yes	Prime	DLIN
Ours: $\Phi_{\mathsf{slp}}^{\mathsf{prime}}$	$O(\kappa^{1-c})$	$O(\kappa^c)$	$O(\kappa^c)$	No	Yes	Prime	DLIN
Ours: Φ_{anon}	$O(\kappa)$	$O(1)$	$O(1)$	Yes	Yes	Prime	DLIN

We compare IBE schemes focusing tight security reduction from static assumptions in the standard model. $\lvert\mathsf{pp}\rvert + \lvert\mathsf{mpk}\rvert$, $\lvert\mathsf{CT}\rvert$, and $\lvert\mathsf{sk}_{\mathsf{ID}}\rvert$ show the size of the master public keys and public parameters, ciphertexts, and private keys, respectively. To measure the efficiency, we count the number of group elements. In the table, κ denotes the security parameter. "Anon" shows whether the scheme is anonymous. "Multi-Challenge?" asks whether (almost) tight security reduction in the multi-challenge setting is shown. "SGD" stands for sub-group decision assumptions. "CW" and "HKS" denote specific assumption used in the corresponding paper. For $\Phi_{\mathsf{slp}}^{\mathsf{comp}}$ and $\Phi_{\mathsf{slp}}^{\mathsf{prime}}$, we can assign any $0 \leq c \leq 1$.

* This is the only scheme that can be generalized to HIBE.

† These schemes can be generalized to be secure under the k-linear assumption (k-LIN) [30, 51] for any $k \in \mathbb{N}$. In such a case, $\lvert\mathsf{pp}\rvert + \lvert\mathsf{mpk}\rvert$, $\lvert\mathsf{CT}\rvert$, and $\lvert\mathsf{sk}_{\mathsf{ID}}\rvert$ are changed to be $O(k^2\kappa)$, $O(k)$, and $O(k)$, respectively. Note that the DLIN assumption corresponds to the 2-LIN assumption.

1.3 Our Techniques

Difficulties. To solve the first question above, natural starting point would be trying to apply the frameworks for composite-order-to-prime-order-conversion dedicated to identity/attribute-based encryption [37, 19, 3, 17, 2] to the HKS scheme [33]. However, security proofs for CW and HKS schemes significantly deviate from the most standard form of dual system encryption methodology [40, 42, 55, 4], only for which the above mentioned frameworks can be applied. Another approach is to try to convert specific assumptions they use into prime-order. In fact, Chen and Wee [18] were able to accomplish such a conversion for their scheme. However, their technique is non-generic and therefore it is highly unclear whether the same argument is possible for the assumptions that HKS use.

Next, we explain the difficulty of the second question. The reason why all IBE schemes featuring (almost) tight security reduction in previous works [18, 8, 33] require large public parameters is that they use (randomized version of) Naor-Reingold PRF [43] in their construction. Note that the Naor-Reingold PRF requires seed length which is linear in the input size, which in turn implies very long public parameters in the IBE schemes. A natural approach to improve the efficiency would be, as noted by Chen and Wee [18, 20], to

reduce the seed length of the Naor-Reingold PRF. However, this is a long-standing open problem and turns out to be quite difficult.

Our Strategy. In this paper, we introduce new proof techniques for IBE schemes (with almost tight security) that rely *only on the subgroup decision assumptions* [*]. This allows us to use frameworks for composite-order-to-prime-order conversions in the literature [25, 47, 37, 24, 28, 3, 17, 2] (to name only a few) which converts subgroup decision assumption into a static assumption in prime-order groups, such as the DLIN assumption. Therefore, using these techniques, we are able to convert a variant of HKS scheme into prime-order. This answers the first question above. Note that in the security proof of HKS (and CW), they rely on some specific assumptions in composite-order groups in addition to subgroup decision assumptions. Because of these, it is unclear how to convert HKS scheme into prime-order.

As for the second question, we view Chen and Wee's scheme as being constructed from, somewhat surprisingly, *broadcast encryption* mechanism, instead of (Naor-Reingold) PRF, and hence can avoid the above difficulty regarding PRF. More precisely, we show that the task of constructing almost tightly secure IBE scheme is essentially reduced to a construction of broadcast encryption, and based on this idea, we are able to obtain the first IBE scheme with sub-linear size public parameters and almost tight security. In the following, we explain our technique.

Detailed Overview of Our Technique. Let us start from the following variant of the Chen and Wee's IBE scheme. Let the identity space of the scheme be $\{0,1\}^\ell$. For $i \in \{1,2,3\}$, let g_i be the generator of a subgroup of order p_i of $\mathbb{G}$, which is bilinear groups of composite order $N = p_1p_2p_3$. Let also h be a generator of $\mathbb{G}$. The master public key, a ciphertext, and a private key for an identity ID are in the following form:

$$\begin{aligned}
\mathsf{mpk} &= \left(g_1, g_1^{w_{1,0}}, g_1^{w_{1,1}}, \ldots, g_1^{w_{\ell,0}}, g_1^{w_{\ell,1}}, e(g_1,h)^\alpha \right), \\
\mathsf{CT}_{\mathsf{ID}} &= \left(g_1^s,\ g_1^{s\sum_{i\in[1,\ell]} w_{i,\mathsf{ID}_i}},\ e(g_1,h)^{s\alpha}\cdot \mathsf{M} \right), \quad \mathsf{sk}_{\mathsf{ID}} = \left(h^\alpha \cdot g_1^{r\sum_{i\in[1,\ell]} w_{i,\mathsf{ID}_i}},\ g_1^{-r} \right)
\end{aligned}$$

where ID_i is the i-th bit of ID and M is the message.[†] Now we are going to show the security. We only consider the single-challenge and single-instance case here for simplicity. In the security proof, at first, the challenge ciphertext is changed to the following form using a subgroup decision assumption:

$$\left(g_1^s \cdot g_2^{\hat{s}},\quad g_1^{s\sum_{i\in[1,\ell]} w_{i,\mathsf{ID}_i}} \cdot g_2^{\hat{s}\sum_{i\in[1,\ell]} w_{i,\mathsf{ID}_i}},\quad e(g_1^s\cdot g_2^{\hat{s}}, h^\alpha)\cdot \mathsf{M} \right).$$

Then, we consider ℓ hybrid games. In Game_i, all private keys are in the following form:

$$\left(h^\alpha \cdot \boxed{g_2^{\widehat{\mathsf{R}}_i(\mathsf{ID}|_i)}} \cdot g_1^{r\sum_{i\in[1,\ell]} w_{i,\mathsf{ID}_i}},\quad g_1^{-r} \right)$$

where $\mathsf{ID}|_i$ is the length i prefix of the identity ID and $\widehat{\mathsf{R}}_i : \{0,1\}^i \to N$ is a random function. Intuitively, through these hybrid games, the randomizing part of the key (highlighted in the box) are gradually randomized and made dependent on more and more bits of each identity. Finally, in Game_ℓ, we can argue that any adversary cannot obtain the information on the message M, because these randomizing parts prevent it.

A crucial part of the security proof is to establish the indistinguishability between $\mathsf{Game}_{i^\star-1}$ and $\mathsf{Game}_{i^\star}$ for all $i^\star \in [1,\ell]$. For the target identity $\mathsf{ID}^\star$ (recall that we are considering the single-challenge and single-instance case for now), we assume that $b^\star := \mathsf{ID}^\star_{i^\star}$ is known to the reduction algorithm in advance, since it can

[*] In fact, we also require the decisional bilinear Diffie-Hellman (DBDH) assumption on the composite-order groups (Problem 5) in addition to the subgroup decision assumptions. However, the assumption does not use the power of composite-order groups. In other words, it does *not* imply the factoring assumption. Therefore, it is ready to be converted into prime-order.

[†] In the actual scheme, $\mathsf{sk}_{\mathsf{ID}}$ is randomized by elements of $\mathbb{G}_{p_3}$, but we do not care this point in this overview.

be guessed with probability $1/2$. At the core of the proof for this is an indistinguishability of the following distributions:

$$\begin{aligned}&\text{Given} \quad \left(g_1^s \cdot g_2^{\hat{s}},\ g_1^{s\sum_{i\in[1,\ell]} w_{i,\mathsf{ID}_i^\star}} \cdot g_2^{\hat{s}\sum_{i\in[1,\ell]} w_{i,\mathsf{ID}_i^\star}}\right),\\ &\left(g_1^{r\sum_{i\in[1,\ell]} w_{i,\mathsf{ID}_i}}, g_1^{-r}\right) \stackrel{c}{\approx} \left(\boxed{g_2^{\hat{\alpha}}} \cdot g_1^{r\sum_{i\in[1,\ell]} w_{i,\mathsf{ID}_i}}, g_1^{-r}\right)\end{aligned} \tag{1}$$

for all ID such that $\mathsf{ID}_{i^\star} \neq b^\star$, where $\hat{\alpha} \stackrel{\$}{\leftarrow} \mathbb{Z}_N$. Indistinguishability of $\mathsf{Game}_{i^\star-1}$ and $\mathsf{Game}_{i^\star}$ is reduced to Equation (1). The reduction algorithm can create the challenge ciphertext using the first term in Equation (1). It can also set private key as

$$\begin{cases} h^\alpha \cdot g_2^{\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})} \cdot g_1^{r\sum_{i\in S} w_{i,\mathsf{ID}_i}}, g_1^{-r} & \text{if } \mathsf{ID}_{i^\star} = b^\star \\ h^\alpha \cdot g_2^{\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})} \cdot \boxed{g_2^{\hat{\alpha}}} \cdot g_1^{r\sum_{i\in S} w_{i,\mathsf{ID}_i}}, g_1^{-r} & \text{if } \mathsf{ID}_{i^\star} \neq b^\star \end{cases}$$

where $\hat{\alpha} = 0$ or $\hat{\alpha} \stackrel{\$}{\leftarrow} \mathbb{Z}_N$. It is clear that the game corresponds to $\mathsf{Game}_{i^\star-1}$ if $\hat{\alpha} = 0$. On the other hand, if $\hat{\alpha} \stackrel{\$}{\leftarrow} \mathbb{Z}_N$, it corresponds to $\mathsf{Game}_{i^\star}$ with

$$\widehat{\mathsf{R}}_{i^\star}(\mathsf{ID}|_{i^\star}) = \begin{cases} \widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) & \text{if } \mathsf{ID}_{i^\star} = b^\star \\ \widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) + \hat{\alpha} & \text{if } \mathsf{ID}_{i^\star} \neq b^\star \end{cases}.$$

If $\hat{\alpha}$ is freshly chosen for every distinct $\mathsf{ID}|_{i^\star}$, the simulation is perfect. Therefore, our task of the security proof is reduced to establish Equation (1). To understand better, we decompose the private key in Equation (1) and restate it again in a slightly stronger form:

$$\begin{aligned}&\text{Given} \quad \left(g_1^s \cdot g_2^{\hat{s}},\ g_1^{s\sum_{i\in[1,\ell]} w_{i,\mathsf{ID}_i^\star}} \cdot g_2^{\hat{s}\sum_{i\in[1,\ell]} w_{i,\mathsf{ID}_i^\star}}\right),\\ &\left(g_1^{rw_{i^\star,1-b^\star}}, g_1^{-r}, \{g_1^{rw_{j,b}}\}_{(j,b)\neq(i^\star,1-b^\star)}\right) \stackrel{c}{\approx} \left(\boxed{g_2^{\hat{\alpha}}} \cdot g_1^{rw_{i^\star,1-b^\star}}, g_1^{-r}, \{g_1^{rw_{j,b}}\}_{(j,b)\neq(i^\star,1-b^\star)}\right).\end{aligned}$$

Let us consider a bijection map $f : \{(i,b)\}_{i\in[1,\ell],b\in\{0,1\}} \to [1,2\ell]$ and replace (i,b) with $f((i,b))$. We can further restate the requirement as:

$$\begin{aligned}&\text{Given} \quad \left(g_1^s \cdot g_2^{\hat{s}},\ g_1^{s\sum_{j\in S^\star} w_j} \cdot g_2^{\hat{s}\sum_{j\in S^\star} w_j}\right),\\ &\left(g_1^{rw_{\tau^\star}}, g_1^{-r}, \{g_1^{rw_j}\}_{j\neq\tau^\star}\right) \stackrel{c}{\approx} \left(\boxed{g_2^{\hat{\alpha}}} \cdot g_1^{rw_{\tau^\star}}, g_1^{-r}, \{g_1^{rw_j}\}_{j\neq\tau^\star}\right)\end{aligned} \tag{2}$$

where $S^\star = \{f(i,\mathsf{ID}_i^\star)\}_{i\in[\ell]}$, $\tau^\star = f((i^\star, 1-b^\star))$, and thus $\tau^\star \notin S^\star$. We call the terms in the second line above as the challenge terms. (It should not be confused with challenge ciphertext.) At this point, we can now see a similarity to broadcast encryption. We consider the following broadcast encryption which captures the essence of the above requirement. Let the set of user index be $[1,2\ell]$.

$$\begin{aligned}\mathsf{mpk} &= (g_1, g_1^{w_1}, \ldots, g_1^{w_{2\ell}}, e(g_1,h)^\alpha),\\ \mathsf{CT}_S &= (g_1^s, g_1^{s\sum_{j\in S} w_j}, e(g_1,h)^{s\alpha}\cdot \mathsf{M}), \quad \mathsf{sk}_\tau = (h^\alpha g_1^{rw_\tau}, g_1^{-r}, \{g_1^{rw_j}\}_{j\in[2\ell]\setminus\{\tau\}})\end{aligned}$$

where CT_S is a ciphertext for a set $S \subseteq [2\ell]$ and sk_τ is a private key for a user index $\tau \in [2\ell]$. This is in fact a variant of the broadcast encryption by Gentry and Waters [27]! Indeed, Equation (2) can be interpreted as a security condition for this broadcast encryption scheme (in the sense of encoding analogous to [55, 4]). It says that given semi-functional ciphertext for a set $S^\star$, a normal private key for $\tau^\star \notin S^\star$ is indistinguishable

from a semi-functional private key for $\tau^\star$. At this point, we are able to understand the core technique in Chen and Wee in terms of broadcast encryption scheme.

However, we have not finished yet. In order to make the proof go through, we argue that an adversary cannot distinguish challenge terms in Equation (2), even if these are given to the adversary *unbounded many times* with freshly chosen randomness $\hat{\alpha}, r$. Such an indistinguishability can be shown by a standard technique [38, 55, 4] if the challenge term is given to the adversary *only once*. This can be accomplished by the combination of subgroup decision assumption and the parameter-hiding argument. In parameter-hiding argument, a value which is information-theoretically hidden is used to make normal private key semi-functional [40, 38, 55, 4]. At the first glance, this argument does not seem to be extended to the case where many challenge terms are given to the adversary, since entropy of hidden parameters (in this case, $w_1, \ldots, w_{2\ell} \mod p_2$) is limited. However, we have to simulate unbounded number of challenge terms. Chen and Wee [18] resolve this problem by using computational argument instead of information-theoretic argument as above. Namely, they assume a variant of the DDH assumption on $\mathbb{G}_{p_2}$ [‡] and embed the problem instance into the above challenge terms. Indistinguishability of multiple challenge terms are tightly reduced to the assumption, using the random self-reducibility of the assumption. On the other hand, our technique for boosting to multi-challenge is much simpler. Our key observation is that the challenge term in Equation (2) can be easily randomized by picking $a \xleftarrow{\$} \mathbb{Z}_N$ and computing

$$\left(\left(g_2^{\hat{\alpha}} \cdot g_1^{r w_{\tau^\star}}\right)^a, \left(g_1^{-r}\right)^a, \{\left(g_1^{r w_j}\right)^a\}_{j \neq \tau^\star}\right) = \left(g_2^{\hat{\alpha}'} \cdot g_1^{r' w_{\tau^\star}}, g_1^{-r'}, \{g_1^{r' w_j}\}_{j \neq \tau^\star}\right), \tag{3}$$

where $r' = ar$ and $\hat{\alpha}' = a\hat{\alpha}$. It is easy to see that $r' \mod p_1$ is uniformly random and independent from anything. We can also see that $\hat{\alpha}' \mod p_2 = 0$ if $\hat{\alpha} = 0$ and $\hat{\alpha}' \mod p_2$ is uniformly random if $\hat{\alpha} \neq 0 \mod p_2$. By this argument, we can see that indistinguishability of the single-challenge-term case implies that for the multi-challenge-term case. Based on all the above discussion, we are able to show the security for the above scheme *only using the subgroup decision assumption*.

Overview of Our Framework. We refine the idea above and combine it with the technique by HKS to propose our framework for constructing IBE schemes that are (almost) tightly secure in the multi-challenge and multi-instance setting, in both composite and prime-order groups. We first define a broadcast encoding, which is an abstraction of broadcast encryption. The syntax of it is a special case of "pair encoding" in [4] (also similar to "predicate encoding" in [55]). Then, we define perfect master-key hiding (PMH) security and computational-master-key hiding (CMH) security for it. These security notions are also similar to those of [55, 4]. The former is statistical requirement for the encoding, and the latter is computational requirement. We can easily show that the former implies the latter. Then, we also introduce intermediate notion multi-master-key hiding (MMH) security for the encoding. This is more complex notion compared to the PMH and CMH-security, but implied by these, thanks to our boosting technique above. Then, we show that broadcast encoding satisfying the MMH security requirement can be converted into IBE scheme. All these reductions are (almost) tightness-preserving, namely, if the original broadcast encoding is tightly PMH/CMH secure, the resulting IBE scheme is also tightly secure in the multi-challenge and multi-instance setting. Finally, we provide broadcast encoding schemes that satisfy our requirement. One is implicit in Gentry-Waters broadcast encryption scheme [27] and the other is completely new. By instantiating our general framework with the latter construction, we obtain IBE scheme with almost tight security and with sub-linear master public key size.

[‡] Of course, in symmetric bilinear groups, the DDH assumption does not hold. They considered a DDH assumption on $\mathbb{G}_{p_2}$ where each term is perturbed by a random element in $\mathbb{G}_{p_3}$, which prevents trivial attack against the assumption.

1.4 Related Works

Related Works on IBE. The first realizations of IBE in the random oracle model were given in [12, 50, 22]. Later, realization in the standard model [15, 9] were given. In the random oracle model, it is possible to obtain efficient and tightly secure IBE scheme [5]. Gentry [26] proposed a tightly secure anonymous IBE scheme under a non-static, parametrized assumption. Chen and Wee proposed the first almost tightly secure IBE scheme under static and simple assumptions [18, 20]. Attrapadung [4] proposed an IBE scheme whose security loss only depends on the number of key queries before the challenge phase. Jutla and Roy [34] constructed very efficient IBE scheme from the SXDH assumption, based on a technique related to NIZK. Blazy, Kiltz, and Pan [8] further generalized the idea and show that a message authentication code with a certain specific algebraic structure implies (H)IBE. They further obtained almost tightly secure anonymous IBE and (non-anonymous) HIBE via the framework. Note that all above mentioned schemes only focus on the single-challenge setting.

Related Works on the Multi-Challenge CCA-Secure PKE. Bellare, Boldyreva, and Micali [6] gave a tight reduction for the Cramer-Shoup encryption [23] in the multi-instance (multi-user) and the single-challenge setting. They posed an important open question of whether it is possible to construct tightly CCA-secure PKE scheme in the multi-instance and the multi-challenge setting. The first PKE scheme satisfying the requirement was proposed by Hofheinz and Jager [32]. Their scheme requires hundreds of group elements in the ciphertexts. Subsequently, Abe et al. [1] reduced the size by improving the efficiency of the underlying one-time signature. Libert et al. [35] greatly reduced the ciphertext and made it constant-size for the first time. The ciphertext overhead of their scheme consist of 68 group elements. Very recently, Libert et al. [36] further reduced it to 47 group elements. Concurrently and independently to us, Hofheinz [29] proposes the first PKE scheme with the same security guarantee and fully compact parameters, which means all parameters are constant-size. While the ciphertext-size (which consists of 60 group elements) is longer than construction in [36], it achieves much shorter public parameters. We note that while the technique is very powerful, it is unclear how to extend the technique to the IBE setting.

2 Preliminaries

Notation. Vectors will be treated as either row or column vector matrices. When unspecified, we shall let it be a row vector. We denote by $\mathbf{e}_i$ the i-th unit (row) vector: its i-th component is one, all others are zero. $\mathbf{0}$ denotes the zero vector or zero matrix. For an integer $n \in \mathbb{N}$ and a field $\mathbb{F}$, $\mathbb{GL}_n(\mathbb{F})$ denotes the set of all invertible matrix in $\mathbb{F}^{n\times n}$. For a multiplicative group $\mathbb{G}$, we denote by $\mathbb{G}^*$ a set of all *generators* in $\mathbb{G}$. We also denote by $[a, b]$ a set $\{a, \ldots, b\}$ for any integer a and b and $[n] = [1, n]$ for any $n \in \mathbb{N}$. We denote by $u \xleftarrow{\$} U$ the fact that u is picked uniformly at random from a finite set U.

2.1 Identity-based Encryption

In this section, we define the syntax and security of IBE (in the multi-challenge, multi-instance setting).

Syntax. An IBE scheme with identity space $\mathcal{ID}$ and message space $\mathcal{M}$ consists of the following algorithms:

$\mathsf{Par}(1^\kappa) \to (\mathsf{pp}, \mathsf{sp})$: The parameter sampling algorithm takes as input a security parameter 1^κ and outputs a public parameter pp and a secret parameter sp.

$\mathsf{Gen}(\mathsf{pp}, \mathsf{sp}) \to (\mathsf{mpk}, \mathsf{msk})$: The key generation algorithm takes pp and sp as input and outputs a master public key mpk and master secret key msk.

$\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{ID}) \to \mathsf{sk}_{\mathsf{ID}}$: The user private key extraction algorithm takes as input the master secret key msk, the master public key mpk, and an identity $\mathsf{ID} \in \mathcal{ID}$. It outputs a private key $\mathsf{sk}_{\mathsf{ID}}$.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}, \mathsf{M}) \to \mathsf{CT}$: The encryption algorithm takes as input a master public key mpk, an identity ID, and a message $\mathsf{M} \in \mathcal{M}$. It will output a ciphertext CT.

$\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{CT}) \to \mathsf{M}$: The decryption algorithm takes as input a private key $\mathsf{sk}_{\mathsf{ID}}$ and a ciphertext CT. It outputs a message M or $\perp$ which indicates that the ciphertext is not in a valid form.

We require correctness of decryption: that is, for all κ, all pp and sp produced by $\mathsf{Par}(1^\kappa)$, all $(\mathsf{mpk}, \mathsf{msk})$ produced by $\mathsf{Gen}(\mathsf{pp}, \mathsf{sp})$, all $\mathsf{ID} \in \mathcal{ID}$, all $\mathsf{M} \in \mathcal{M}$, all CT returned by $\mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}, \mathsf{M})$, and all $\mathsf{sk}_{\mathsf{ID}}$ returned by $\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{ID})$, $\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{CT}) = \mathsf{M}$ holds.

In our constructions, we will set identity space $\mathcal{ID} = \{0,1\}^\ell$ for some $\ell \in \mathbb{N}$. Note that the restriction on the identity space can be easily removed by applying a collision resistant hash function $\mathsf{CRH} : \{0,1\}^* \to \{0,1\}^\ell$ to an identity. Typically, we would set $\ell = \Theta(\kappa)$ to avoid the birthday attack.

Security Model. We now define (μ, Q_c, Q_k)-security for an IBE $\Phi = (\mathsf{Par}, \mathsf{Gen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$. This security notion is defined by the following game between a challenger and an attacker $\mathcal{A}$.

Setup. The challenger runs $(\mathsf{pp}, \mathsf{sp}) \xleftarrow{\$} \mathsf{Par}(1^\kappa)$ and $(\mathsf{mpk}^{(j)}, \mathsf{msk}^{(j)}) \xleftarrow{\$} \mathsf{Gen}(\mathsf{pp}, \mathsf{sp})$ for $j \in [\mu]$. The challenger also picks random coin $\mathsf{coin} \xleftarrow{\$} \{0,1\}$ whose value is fixed throughout the game. Then, $(\mathsf{pp}, \{\mathsf{mpk}^{(j)}\}_{j\in[\mu]})$ is given to $\mathcal{A}$.

In the following, $\mathcal{A}$ adaptively makes the following two types of queries in an arbitrary order.

−**Key Extraction Query.** The adversary $\mathcal{A}$ submits $(\texttt{Extraction}, j \in [\mu], \mathsf{ID} \in \mathcal{ID})$ to the challenger. Then, the challenge runs $\mathsf{sk}_{\mathsf{ID}}^{(j)} \xleftarrow{\$} \mathsf{Ext}(\mathsf{msk}^{(j)}, \mathsf{mpk}^{(j)}, \mathsf{ID})$ and returns $\mathsf{sk}_{\mathsf{ID}}^{(j)}$ to $\mathcal{A}$.

−**Challenge Query.** The adversary $\mathcal{A}$ submits $(\texttt{Challenge}, j \in [\mu], \mathsf{ID} \in \mathcal{ID}, \mathsf{M}_0, \mathsf{M}_1 \in \mathcal{M})$ to the challenger. Then, the challenger runs $\mathsf{CT} \xleftarrow{\$} \mathsf{Enc}(\mathsf{mpk}^{(j)}, \mathsf{ID}, \mathsf{M}_{\mathsf{coin}})$ and returns CT to $\mathcal{A}$.

Guess. At last, $\mathcal{A}$ outputs a guess coin' for coin. The advantage of an attacker $\mathcal{A}$ in the game is defined as $\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A},\Phi,(\mu,Q_c,Q_k)}(\kappa) = |\Pr[\mathsf{coin}' = \mathsf{coin}] - \frac{1}{2}|$.

We say that the adversary $\mathcal{A}$ is valid if and only if $\mathcal{A}$ never queries $(\texttt{Extraction}, j, \mathsf{ID})$ such that it has already queried $(\texttt{Challenge}, j, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$ for the same (j, ID) (and vice versa); $\mathcal{A}$ has made at most Q_c challenge queries; and $\mathcal{A}$ has made at most Q_k key extraction queries.

Definition 1. *We say that IBE Φ is secure if $\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A},\Phi,(\mu,Q_c,Q_k)}(\kappa)$ is negligible for any polynomially bounded μ, Q_c, Q_k, and any valid PPT adversary $\mathcal{A}$.*

Anonymity. We also consider anonymity for the IBE scheme. To define (μ, Q_c, Q_k)-anonymity for an IBE scheme, we change the form of challenge queries in the above game as follows.

−**Challenge Query.** The adversary $\mathcal{A}$ submits $(\texttt{Challenge}, j \in [\mu], \mathsf{ID}_0, \mathsf{ID}_1 \in \mathcal{ID}, \mathsf{M}_0, \mathsf{M}_1 \in \mathcal{M})$ to the challenger. Then, the challenger runs $\mathsf{CT} \xleftarrow{\$} \mathsf{Enc}(\mathsf{mpk}^{(j)}, \mathsf{ID}_{\mathsf{coin}}, \mathsf{M}_{\mathsf{coin}})$ and returns CT to $\mathcal{A}$.

We say that the adversary $\mathcal{A}$ is valid if $\mathcal{A}$ never queries $(\texttt{Extraction}, j, \mathsf{ID})$ such that it has already queried $(\texttt{Challenge}, j, \mathsf{ID}_0, \mathsf{ID}_1, \mathsf{M}_0, \mathsf{M}_1)$ for the same j and $\mathsf{ID} \in \{\mathsf{ID}_0, \mathsf{ID}_1\}$ (and vice versa); $\mathcal{A}$ has made at most Q_c challenge queries; and $\mathcal{A}$ has made at most Q_k key extraction queries. We define the advantage of $\mathcal{A}$ in this modified game as $\mathsf{Adv}^{\mathsf{AIBE}}_{\mathcal{A},\Phi,(\mu,Q_c,Q_k)}(\kappa) := |\Pr[\mathsf{coin}' = \mathsf{coin}] - \frac{1}{2}|$.

Definition 2. *We say that IBE Φ is anonymous if $\mathsf{Adv}^{\mathsf{AIBE}}_{\mathcal{A},\Phi,(\mu,Q_c,Q_k)}(\kappa)$ is negligible for any polynomially bounded μ, Q_c, Q_k, and any valid PPT adversary $\mathcal{A}$.*

2.2 Composite-Order Bilinear Groups

We will use bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of composite order $N = p_1p_2p_3p_4$, where p_1, p_2, p_3, p_4 are four distinct prime numbers, with efficiently computable and non-degenerate bilinear map $e(\cdot) : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. For each $d|N$, $\mathbb{G}$ has unique subgroup of order d denoted by $\mathbb{G}_d$. We let g_i be a generator of $\mathbb{G}_{p_i}$. For our purpose, we define a (composite order) bilinear group generator $\mathcal{G}_{\mathsf{comp}}$ that takes as input a security parameter 1^κ and

outputs $(N, \mathbb{G}, \mathbb{G}_T, g_1, g_2, g_3, g_4, e(\cdot))$. Any $h \in \mathbb{G}$ can be expressed as $h = g_1^{a_1} g_2^{a_2} g_3^{a_3} g_4^{a_4}$, where a_i is uniquely determined modulo p_i. We call $g_i^{a_i}$ the $\mathbb{G}_{p_i}$ component of h. We have that $e(g^a, h^b) = e(g, h)^{ab}$ for any $g, h \in \mathbb{G}$, $a, b \in \mathbb{Z}$, $e(g, h) \neq 1_{\mathbb{G}_T}$ for $g, h \neq 1_{\mathbb{G}}$, and $e(g, h) = 1_{\mathbb{G}_T}$ for $g \in \mathbb{G}_{p_i}$ and $h \in \mathbb{G}_{p_j}$ with $i \neq j$.

Let $(N, \mathbb{G}, \mathbb{G}_T, g_1, g_2, g_3, g_4, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{comp}}(1^\kappa)$ and $g \xleftarrow{\$} \mathbb{G}^*$. We define advantage function $\mathsf{Adv}^{\mathsf{Pxx}}_{\mathcal{A}}(\kappa)$ for Problem xx for any adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{Pxx}}_{\mathcal{A}}(\kappa) = |\Pr[\mathcal{A}(g_1, g_4, g, D, T_0) \to 1] - \Pr[\mathcal{A}(g_1, g_4, g, D, T_1) \to 1]|.$$

In each problem, D, T_0, and T_1 are defined as follows. In the following, for $i, j \in [1, 4]$, g_{ij} is chosen as $g_{ij} \xleftarrow{\$} \mathbb{G}^*_{p_i p_j}$.

Problem 1. $D = \emptyset$, $T_0 \xleftarrow{\$} \mathbb{G}^*_{p_1}$, *and* $T_1 \xleftarrow{\$} \mathbb{G}^*_{p_1 p_2}$.

Problem 2. $D = (g_{12}, g_3, g_{24})$, $T_0 \xleftarrow{\$} \mathbb{G}^*_{p_1 p_4}$, *and* $T_1 \xleftarrow{\$} \mathbb{G}^*_{p_1 p_2 p_4}$.

Problem 3. $D = (g_{13}, g_2, g_{34})$, $T_0 \xleftarrow{\$} \mathbb{G}^*_{p_1 p_4}$, *and* $T_1 \xleftarrow{\$} \mathbb{G}^*_{p_1 p_3 p_4}$.

Problem 4. $D = (g_{12}, g_{23})$, $T_0 \xleftarrow{\$} \mathbb{G}^*_{p_1 p_2}$, *and* $T_1 \xleftarrow{\$} \mathbb{G}^*_{p_1 p_3}$.

Problem 5. $D = (g_2, g_3, g_2^x, g_2^y, g_2^z)$, $T_0 = e(g_2, g_2)^{xyz}$, *and* $T_1 = e(g_2, g_2)^{xyz+\gamma}$, *where* $x, y, z \xleftarrow{\$} \mathbb{Z}_N$ *and* $\gamma \xleftarrow{\$} \mathbb{Z}^*_N$.

Problem 1 2, 3, and 4 are called sub-group decision problems. Problem 5 is called the decisional bilinear Diffie-Hellman problem.

Matrix-in-the-Exponent. Given any vector $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{Z}_N^n$ and a group element g, we write $g^{\mathbf{w}} \in \mathbb{G}^n$ to denote $(g^{w_1}, \ldots, g^{w_n}) \in \mathbb{G}^n$: we define $g^{\mathbf{A}}$ for a matrix $\mathbf{A}$ in a similar way. $g^{\mathbf{A}} \cdot g^{\mathbf{B}}$ denotes componentwise product: $g^{\mathbf{A}} \cdot g^{\mathbf{B}} = g^{\mathbf{A}+\mathbf{B}}$. Note that given $g^{\mathbf{A}}$ and a matrix $\mathbf{B}$ of "exponents", one can efficiently compute $g^{\mathbf{BA}}$ and $g^{\mathbf{AB}} = (g^{\mathbf{A}})^{\mathbf{B}}$. Furthermore, if there is an efficiently computable map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, then given $g^{\mathbf{A}}$ and $g^{\mathbf{B}}$, one can efficiently compute $e(g, g)^{\mathbf{A}^\top \mathbf{B}}$ via $(e(g, g)^{\mathbf{A}^\top \mathbf{B}})_{i,j} = \prod_k e(g^{A_{k,i}}, g^{B_{k,j}})$ where $A_{i,j}$ and $B_{i,j}$ denotes the (i, j)-th coefficient of $\mathbf{A}$ and $\mathbf{B}$ respectively. We will use $e(g^{\mathbf{A}}, g^{\mathbf{B}}) = e(g, g)^{\mathbf{A}^\top \mathbf{B}}$ to denote this operation.

3 Broadcast Encoding: Definitions and Reductions

In this section, we define the syntax and the security notions for broadcast encoding. The syntax of our definition corresponds to a special case of "pair encoding" defined in [4] and is also similar to "predicate encoding" in [55]. As for the security requirement for the encoding, ours are slightly different from both. We define several flavours of the security requirement: perfect master-key hiding security (PMH), computational-master-key hiding (CMH) security, and the multi-master-key hiding (MMH) security. The last one is useful, since we can obtain IBE scheme from broadcast encoding scheme satisfying the security notion, as we will show in Section 4. However, MMH security is defined by relatively complex game and may not be easy to show. Later in this section, we will show that MMH security can be tightly reduced to much simpler CMH and PMH security. The reduction is based on very simple but powerful randomness amplification argument (Lemma 2).

3.1 Broadcast Encoding: Syntax

The broadcast encoding Π consists of the following four deterministic algorithms.

$\mathsf{Param}(n, N) \to d_1$: It takes as input an integer n and N and outputs $d_1 \in \mathbb{N}$ which specifies the number of common variables in CEnc and KEnc. For the default notation, $\mathbf{w} = (w_1, \ldots, w_{d_1})$ denotes the list of common variables.

$\mathsf{KEnc}(\tau, N) \to (\mathbf{k}, d_2')$: It takes as input $\tau \in [n]$, $N \in \mathbb{N}$, and outputs a vector of polynomials $\mathbf{k} = (k_1, \ldots, k_{d_2})$ with coefficients in $\mathbb{Z}_N$, and $d_2' \in \mathbb{N}$ that specifies the number of its own variables. We assume that d_2 and d_2' only depend on n and do not depend on τ without loss of generality. We require that each polynomials $\mathbf{k}$ is a *linear combination of monomials* $\alpha, r_j, w_k r_j$ where $\alpha, r_1, \ldots, r_{d_2'}, w_1, \ldots, w_{d_1}$ are variables. More precisely, it outputs $\{b_\iota\}_{\iota \in [d_2]}$, $\{b_{\iota,j}\}_{(\iota,j)\in[d_2]\times[d_2']}$, and $\{b_{\iota,j,k}\}_{(\iota,j,k)\in[d_2]\times[d_2']\times[d_1]}$ in $\mathbb{Z}_N$ such that

$$k_\iota\Big(\alpha, r_1, \ldots, r_{d_2'}, w_1, \ldots, w_{d_1}\Big) = b_\iota \alpha + \Big(\sum_{j\in[d_2']} b_{\iota,j} r_j\Big) + \Big(\sum_{(j,k)\in[d_2']\times[d_1]} b_{\iota,j,k} w_k r_j\Big)$$

for $\iota \in [d_2]$.

$\mathsf{CEnc}(S, N) \to (\mathbf{c}, d_3')$: It takes as input $S \subseteq [n]$, $N \in \mathbb{N}$, and outputs a vector of polynomials $\mathbf{c} = (c_1, \ldots, c_{d_3})$ with coefficients in $\mathbb{Z}_N$, and $d_3' \in \mathbb{N}$ that specifies the number of its own variables. We require that polynomials $\mathbf{c}$ in variables $s_0, s_1, \ldots, s_{d_3'}, w_1, \ldots, w_{d_1}$ has the following form:

There exist (efficiently computable) set of coefficients $\{a_{\iota,j}\}_{(\iota,j)\in[d_3]\times[0,d_3']}$ and $\{a_{\iota,j,k}\}_{(\iota,j,k)\in[d_3]\times[0,d_3']\times[d_1]}$ in $\mathbb{Z}_N$ such that

$$c_\iota\Big(s_0, s_1, \ldots, s_{d_3'}, w_1, \ldots, w_{d_1}\Big) = \Big(\sum_{j\in[0,d_3']} a_{\iota,j} s_j\Big) + \Big(\sum_{(j,k)\in[0,d_3']\times[d_1]} a_{\iota,j,k} w_k s_j\Big)$$

for $\iota \in [d_3]$. We also require that $c_1 = s_0$.

$\mathsf{Pair}(\tau, S, N) \to \mathbf{E}$: It takes as input $\tau \in [n]$, $S \subseteq [n]$, and $N \in \mathbb{N}$ and outputs a matrix $\mathbf{E} = (E_{i,j})_{i\in[d_2],j\in[d_3]} \in \mathbb{Z}_N^{d_2\times d_3}$.

Correctness. The correctness requirement is as follows.

- We require that for any n, N, $d_1 \leftarrow \mathsf{Param}(n, N)$, $\mathbf{k} \leftarrow \mathsf{KEnc}(\tau, N)$, $\mathbf{c} \leftarrow \mathsf{CEnc}(S, N)$, and $\mathbf{E} \leftarrow \mathsf{Pair}(\tau, S, N)$, we have that

  $$\mathbf{k}\mathbf{E}\mathbf{c}^\top = \alpha s_0 \qquad \text{whenever} \qquad \tau \in S.$$

  The equation holds symbolically, or equivalently, as polynomials in variables $\alpha, r_1, \ldots, r_{d_2'}, s_0, s_1, \cdots, s_{d_3'}, w_1, \ldots, w_{d_1}$.

- For p that divides N, if we let $\mathsf{KEnc}(\tau, N) \to (\mathbf{k}, d_2')$ and $\mathsf{KEnc}(\tau, p) \to (\mathbf{k}', d_2'')$, then it holds that $d_2' = d_2''$ and $\mathbf{k} \mod p = \mathbf{k}'$. The requirement for CEnc is similar.

Note that since $\mathbf{k}\mathbf{E}\mathbf{c}^\top = \sum_{(i,j)\in[d_2]\times[d_3]} E_{i,j} k_i c_j$, the first requirement amounts to check if there is a linear combination of $k_i c_j$ terms summed up to αs_0. In the descriptions of proposed broadcast encoding schemes, which will appear later in this paper, we will not explicitly write down $\mathbf{E}$. Instead, we will check this condition.

3.2 Broadcast Encoding: Security

Here, we define two flavours of security notions for broadcast encoding: perfect security and computational security. As we will see, the former implies the latter. In what follows, we denote $\mathbf{w} = (w_1, \ldots, w_{d_1})$, $\mathbf{r} = (r_1, \ldots, r_{d_2'})$, and $\mathbf{s} = (s_0, s_1, \ldots, s_{d_3'})$.

(Perfect Security.) The pair encoding scheme $\Pi = (\mathsf{Param}, \mathsf{KEnc}, \mathsf{CEnc}, \mathsf{Pair})$ is Q-perfectly master-key hiding (Q-PMH) if the following holds. For any $n \in \mathbb{N}$, prime $p \in \mathbb{N}$, $\tau \in [n]$, and $S_1, \ldots, S_Q \subset [n]$ such that $\tau \notin S_j$ for all $j \in [Q]$, let $\mathsf{Param}(n,p) \to d_1$, $(\mathbf{k}_\tau, d_2') \leftarrow \mathsf{KEnc}(\tau, p)$, and $(\mathbf{c}_{S_j}, d_{3,j}') \leftarrow \mathsf{CEnc}(S_j, p)$ for $j \in [Q]$, then the following two distributions are identical:

$$\left\{ \left\{ \mathbf{c}_{S_j}(\mathbf{s}_j, \mathbf{w}) \right\}_{j \in [Q]},\ \mathbf{k}_\tau(0, \mathbf{r}, \mathbf{w}) \right\} \text{ and } \left\{ \left\{ \mathbf{c}_{S_j}(\mathbf{s}_j, \mathbf{w}) \right\}_{j \in [Q]},\ \mathbf{k}_\tau(\alpha, \mathbf{r}, \mathbf{w}) \right\} \tag{4}$$

where $\mathbf{w} \xleftarrow{\$} \mathbb{Z}_p^{d_1}$, $\alpha \xleftarrow{\$} \mathbb{Z}_p$, $\mathbf{r} \xleftarrow{\$} (\mathbb{Z}_p^*)^{d_2'}$, $\mathbf{s}_j \xleftarrow{\$} \mathbb{Z}_p^{d_3'+1}$ for $j \in [Q]$.

(Computational Security on $\mathbb{G}_{p_2}$). We define Q-computational-master-key hiding (Q-CMH [§]) security on $\mathbb{G}_{p_2}$ for a broadcast encoding $\Pi = (\mathsf{Param}, \mathsf{KEnc}, \mathsf{CEnc}, \mathsf{Pair})$ by the following game. At the beginning of the game, an (stateful) adversary $\mathcal{A}$ is given $(1^\kappa, n)$ and chooses $\tau^\star \in [n]$. Then, parameters are chosen as $(N, \mathbb{G}, \mathbb{G}_T, g_1, g_2, g_3, g_4, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{comp}}(1^\kappa)$, $\mathsf{Param}(n, N) \to d_1$, and $\hat{\mathbf{w}} \xleftarrow{\$} \mathbb{Z}_N^{d_1}$. The advantage of $\mathcal{A}$ is defined as

$$\begin{aligned}\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{A},\Pi,Q,\mathbb{G}_{p_2}}(\kappa) = |\Pr[\mathcal{A}(1^\kappa, n) \to \tau^\star,\ \mathcal{A}(g_1, g_2, g_3, g_4)^{\mathcal{O}^{\mathsf{CMH},\mathsf{C}}_{\tau^\star,\hat{\mathbf{w}}}(\cdot), \mathcal{O}^{\mathsf{CMH},\mathsf{K}}_{\tau^\star,\hat{\mathbf{w}},0}(\cdot)} \to 1] - \\ \Pr[\mathcal{A}(1^\kappa, n) \to \tau^\star,\ \mathcal{A}(g_1, g_2, g_3, g_4)^{\mathcal{O}^{\mathsf{CMH},\mathsf{C}}_{\tau^\star,\hat{\mathbf{w}}}(\cdot), \mathcal{O}^{\mathsf{CMH},\mathsf{K}}_{\tau^\star,\hat{\mathbf{w}},1}(\cdot)} \to 1]|.\end{aligned}$$

In the above, $\mathcal{O}^{\mathsf{CMH},\mathsf{K}}_{\tau^\star,\hat{\mathbf{w}},b}(\cdot)$ for $b \in \{0,1\}$ are called only once while $\mathcal{O}^{\mathsf{CMH},\mathsf{C}}_{\tau^\star,\hat{\mathbf{w}}}(\cdot)$ can be called at most Q times. These oracles can be called in any order.

- $\mathcal{O}^{\mathsf{CMH},\mathsf{C}}_{\tau^\star,\hat{\mathbf{w}}}(\cdot)$ takes $S \subset [n]$ such that $\tau^\star \notin S$ as input. It then runs $\mathsf{CEnc}(S, N) \to (\mathbf{c}, d_3')$, picks $\hat{\mathbf{s}} = (\hat{s}_0, \hat{s}_1, \ldots, \hat{s}_{d_3'}) \xleftarrow{\$} \mathbb{Z}_N^{d_3'+1}$, and returns $g_2^{\mathbf{c}(\hat{\mathbf{s}}, \hat{\mathbf{w}})}$. We note that $\hat{\mathbf{s}}$ is *freshly chosen* every time the oracle is called.
- $\mathcal{O}^{\mathsf{CMH},\mathsf{K}}_{\tau^\star,\hat{\mathbf{w}},b}(\cdot)$ ignores its input. When it is called, it first runs $\mathsf{KEnc}(\tau^\star, N) \to (\mathbf{k}, d_2')$ and picks $\hat{\mathbf{r}} = (\hat{r}_1, \ldots, \hat{r}_{d_2'}) \xleftarrow{\$} \mathbb{Z}_N^{d_2'}$ and $\hat{\alpha} \xleftarrow{\$} \mathbb{Z}_N$. Then it returns

$$g_2^{\mathbf{k}(b \cdot \hat{\alpha}, \hat{\mathbf{r}}, \hat{\mathbf{w}})} = \begin{cases} g_2^{\mathbf{k}(0, \hat{\mathbf{r}}, \hat{\mathbf{w}})} & \text{if } b = 0 \\ g_2^{\mathbf{k}(\hat{\alpha}, \hat{\mathbf{r}}, \hat{\mathbf{w}})} & \text{if } b = 1. \end{cases}$$

We say that the broadcast encoding is Q-CMH secure on $\mathbb{G}_{p_2}$ if $\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{A},\Pi,Q,\mathbb{G}_{p_2}}(\kappa)$ is negligible for all PPT adversary $\mathcal{A}$.

(Computational Security on $\mathbb{G}_{p_3}$). We define $\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{A},\Pi,Q,\mathbb{G}_{p_3}}(\kappa)$ and Q-CMH security on $\mathbb{G}_{p_3}$ via similar game. The difference from the above game is that $g_2^{\mathbf{c}(\hat{\mathbf{s}}, \hat{\mathbf{w}})}$ and $g_2^{\mathbf{k}(b \cdot \hat{\alpha}, \hat{\mathbf{r}}, \hat{\mathbf{w}})}$ above are replaced with $g_3^{\mathbf{c}(\hat{\mathbf{s}}, \hat{\mathbf{w}})}$ and $g_3^{\mathbf{k}(b \cdot \hat{\alpha}, \hat{\mathbf{r}}, \hat{\mathbf{w}})}$ respectively.

COMPARISON WITH DEFINITION IN [4]. By setting $Q = 1$, the Q-PMH and the Q-CMH security defined as above almost correspond to the perfect security and the co-selective security defined in [4] respectively. We

[§] Here, we use CMH to stand for "computational-master-key hiding" (for broadcast encoding), while in [4], CMH refers to "co-selective master-key hiding" (for pair encoding). We hope that this should not be confusing, since our notion of 1-CMH security is in fact almost the same as the notion of co-selective master-key hiding security (for broadcast predicate) anyway.

need to deal with the case of $Q \gg 1$ in order to handle the multi-challenge setting. Another difference is that we use groups with the order being a product of four primes, while they deal with a product of three primes.

We have the following lemma which indicates that Q-PMH security unconditionally implies Q-CMH security on both of $\mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$. The proof appears in Appendix A.

Lemma 1. *Assume that a broadcast encoding Π satisfies Q-PMH security for some $Q \in \mathbb{N}$. Then it follows that $\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{A},\Pi,Q,\mathbb{G}_{p_i}}(\kappa) \leq d'_2/p_i$ for $i \in \{2,3\}$.*

3.3 Multi-Master-Key Hiding Security in Composite Order Groups

Here, we define multi-master-key hiding security for a broadcast encoding, which is more complex security notion compared to the CMH security. A broadcast encoding scheme that satisfies the security notion can be converted into an IBE scheme as we will show in Section 4.

Multi-Master-Key Hiding Security (on $\mathbb{G}_{p_2}$). We define (Q_c, Q_k)-multi-master-key hiding ((Q_c, Q_k)-MMH) security on $\mathbb{G}_{p_2}$ for a broadcast encoding $\Pi = (\mathsf{Param}, \mathsf{KEnc}, \mathsf{CEnc}, \mathsf{Pair})$. The security is defined by the following game. At the beginning of the game, $\mathcal{A}$ is given $(1^\kappa, n)$ and chooses $\tau^\star \in [n]$. Then, parameters are chosen as $(N, \mathbb{G}, \mathbb{G}_T, g_1, g_2, g_3, g_4, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{comp}}(1^\kappa)$, $g_{24} \xleftarrow{\$} \mathbb{G}^*_{p_2p_4}$, $d_1 \leftarrow \mathsf{Param}(n, N)$, and $\mathbf{w} \xleftarrow{\$} \mathbb{Z}_N^{d_1}$. The advantage of $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,Q_k),\mathbb{G}_{p_2}}(\kappa) = |\Pr[\mathcal{A}(1^\kappa, n) \to \tau^\star,\ \mathcal{A}(g_1, g_1^{\mathbf{w}}, g_3^{\mathbf{w}}, g_{24}, g_3, g_4)^{\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{w}}(\cdot),\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},0}(\cdot)} \to 1] - \Pr[\mathcal{A}(1^\kappa, n) \to \tau^\star,\ \mathcal{A}(g_1, g_1^{\mathbf{w}}, g_3^{\mathbf{w}}, g_{24}, g_3, g_4)^{\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{w}}(\cdot),\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},1}(\cdot)} \to 1]|.$$

In the above, $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{w}}(\cdot)$ and $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ for $b \in \{0,1\}$ can be called at most Q_c times and Q_k times, respectively. They can be called in any order.

- $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{w}}(\cdot)$ takes $S \subset [n]$ such that $\tau^\star \notin S$ as input. It then runs $\mathsf{CEnc}(S, N) \to (\mathbf{c}, d'_3)$, picks $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_N^{d'_3+1}$ and $\hat{\mathbf{s}} \xleftarrow{\$} \mathbb{Z}_N^{d'_3+1}$ and returns $g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}$.
- $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ ignores its input. When it is called, it first runs $\mathsf{KEnc}(\tau^\star, N) \to (\mathbf{k}, d'_2)$, picks $\hat{\alpha} \xleftarrow{\$} \mathbb{Z}_N$, $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{d'_2}$, $\boldsymbol{\delta} \xleftarrow{\$} \mathbb{Z}_N^{d_2}$. Then it returns

$$g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{w})} \cdot g_2^{\mathbf{k}(b\cdot\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}} = \begin{cases} g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{w})} \cdot g_4^{\boldsymbol{\delta}} & \text{if } b = 0 \\ g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{w})} \cdot g_2^{\mathbf{k}(\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}} & \text{if } b = 1. \end{cases}$$

In the above, $\mathbf{r}$, $\hat{\alpha}$, and $\boldsymbol{\delta}$ as well as $\mathbf{s}$ and $\hat{\mathbf{s}}$ are all *freshly chosen* every time the corresponding oracle is called. We say that the broadcast encoding is (Q_c, Q_k)-MMH secure on $\mathbb{G}_{p_2}$ if $\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,Q_k),\mathbb{G}_{p_2}}(\kappa)$ is negligible for all PPT adversary $\mathcal{A}$.

Multi-Master-Key Hiding Security (on $\mathbb{G}_{p_3}$). We define (Q_c, Q_k)-MMH security on $\mathbb{G}_{p_3}$ and $\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,Q_k),\mathbb{G}_{p_3}}(\kappa)$ similarly to the above. The difference is the following.

- The input to $\mathcal{A}$ is replaced with $(g_1, g_1^{\mathbf{w}}, g_2^{\mathbf{w}}, g_{34}, g_2, g_4)$.
- $g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}$ in the above is replaced with $g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot g_3^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}$.
- $g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{w})} \cdot g_2^{\mathbf{k}(b\cdot\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}}$ is replaced with $g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{w})} \cdot g_3^{\mathbf{k}(b\cdot\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}}$.

3.4 Reduction from MMH security to CMH security

We then prove that the (Q_c, Q_k)-MMH security for a broadcast encoding on $\mathbb{G}_{p_2}$ (resp. $\mathbb{G}_{p_3}$) can be tightly reduced to its Q_c-CMH security on $\mathbb{G}_{p_2}$(resp. $\mathbb{G}_{p_3}$) and the hardness of the Problem 2 (resp. 3).

Theorem 1. *For any $i \in \{2,3\}$, broadcast encoding Π, and adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,Q_k),\mathbb{G}_{p_i}}(\kappa) \leq \mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{B}_1,\Pi,Q_c,\mathbb{G}_{p_i}}(\kappa) + 2\mathsf{Adv}^{\mathsf{P}_{\mathrm{xx}}}_{\mathcal{B}_2} + \frac{1}{p_i}$$

and $\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2)\} \approx \mathsf{Time}(\mathcal{A}) + (Q_k + Q_c)\cdot \mathsf{poly}(\kappa, n)$ *where* $\mathsf{poly}(\kappa, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$. *In the above,* $\mathsf{P}_{\mathrm{xx}} = \mathsf{P}_2$ *if* $i = 2$ *and* $\mathsf{P}_{\mathrm{xx}} = \mathsf{P}_3$ *if* $i = 3$.

Proof. By the symmetry between the case of $i = 2$ and $i = 3$, it suffices to show the theorem for the case of $i = 2$. Here, we introduce Lemma 2 and 3. The former indicates that (Q_c, Q_k)-MMH security can be tightly reduced to $(Q_c, 1)$-MMH security. The latter shows $(Q_c, 1)$-MMH security can be tightly reduced to the hardness of the Problem 2 and Q_c-CMH security. Combining these lemmas, the theorem follows. □

Lemma 2. *For any broadcast encoding Π and adversary $\mathcal{A}$, there exists another adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,Q_k),\mathbb{G}_{p_2}}(\kappa) \leq \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B},\Pi,(Q_c,1),\mathbb{G}_{p_2}}(\kappa) + \frac{1}{p_2}$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + Q_k \cdot \mathsf{poly}(\kappa, n)$ *where* $\mathsf{poly}(\kappa, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We construct $\mathcal{B}$ against $(Q_c, 1)$-MMH security of the broadcast encoding from $\mathcal{A}$ against (Q_c, Q_k)-MMH security of the same encoding. At the beginning of the game, $\mathcal{A}(1^\kappa, n)$ submits its target index $\tau^\star \in [n]$. $\mathcal{B}$ then submits the same index and is given $(g_1, g_1^{\mathbf{w}}, g_3^{\mathbf{w}}, g_{24}, g_3, g_4)$. Then, $\mathcal{B}$ simply pass it to $\mathcal{A}$. $\mathcal{B}$ also calls $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ to obtain $g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{w})} \cdot g_2^{\mathbf{k}(b\cdot\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}}$. In the following, we assume that $\hat{\alpha} \neq 0 \mod p_2$. This happens with probability at least $1 - 1/p_2$. When $\mathcal{A}$ calls $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{w}}(\cdot)$ on input S, $\mathcal{B}$ calls the same oracle on the same input and passes what is given from the oracle to $\mathcal{A}$. When $\mathcal{A}$ calls $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$, $\mathcal{B}$ picks random $a \xleftarrow{\$} \mathbb{Z}_N$, $\mathbf{r}' \xleftarrow{\$} \mathbb{Z}_N^{d_2'}$, and $\boldsymbol{\delta}' \xleftarrow{\$} \mathbb{Z}_N^{d_2}$, computes

$$\left(g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{w})} \cdot g_2^{\mathbf{k}(b\cdot\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}}\right)^a \cdot g_1^{\mathbf{k}(0,\mathbf{r}',\mathbf{w})} \cdot g_4^{\boldsymbol{\delta}'} = g_1^{\mathbf{k}(0,a\mathbf{r}+\mathbf{r}',\mathbf{w})} \cdot g_2^{\mathbf{k}(b\cdot a\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{a\boldsymbol{\delta}+\boldsymbol{\delta}'},$$

and returns it to $\mathcal{A}$. Finally, $\mathcal{B}$ outputs $\mathcal{A}$'s output as its guess. It is easy to see that the simulation by $\mathcal{B}$ is perfect. In particular, $\mathcal{B}$ has made only single call to $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$. Therefore, the lemma follows. □

Lemma 3. *For any broadcast encoding Π and adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,1),\mathbb{G}_{p_2}}(\kappa) \leq \mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{B}_1,\Pi,Q_c,\mathbb{G}_{p_2}}(\kappa) + 2\mathsf{Adv}^{\mathsf{P}_2}_{\mathcal{B}_2}$$

and $\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2)\} \approx \mathsf{Time}(\mathcal{A}) + Q_c\mathsf{poly}(\kappa, n)$ *where* $\mathsf{poly}(\kappa, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We prove the lemma via the following sequence of games. We consider following games for $b \in \{0,1\}$. We write $\mathsf{Event}_{\mathrm{xx}}$ to denote the probability that $\mathcal{A}$ outputs 1 in $\mathsf{Game}_{\mathrm{xx}}$.

$\mathsf{Game}_{0,b}$: This is the $(Q_c, 1)$-MMH security game with oracles $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{w}}(\cdot)$ and $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$.

$\mathsf{Game}_{1,b}$: In this game, we change $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ as follows: When it is called, $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ runs $\mathsf{KEnc}(\tau^\star, N) \to \mathbf{k}$, picks $\hat{\alpha} \xleftarrow{\$} \mathbb{Z}_N$, $\mathbf{r}, \hat{\mathbf{r}} \xleftarrow{\$} \mathbb{Z}_N^{d_2'}$, and $\boldsymbol{\delta} \xleftarrow{\$} \mathbb{Z}_N^{d_2}$, and returns $g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{w})} \cdot \boxed{g_2^{\mathbf{k}(b\hat{\alpha},\hat{\mathbf{r}},\mathbf{w})}} \cdot g_4^{\boldsymbol{\delta}}$ instead of $g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{w})} \cdot g_2^{\mathbf{k}(b\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}}$.

We have that

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,1),\mathbb{G}_{p_2}}(\kappa) = |\Pr[\mathsf{Event}_{0,0}] - \Pr[\mathsf{Event}_{0,1}]|\\ \leq\ & |\Pr[\mathsf{Event}_{1,0}] - \Pr[\mathsf{Event}_{1,1}]| + \sum_{b\in\{0,1\}} |\Pr[\mathsf{Event}_{0,b}] - \Pr[\mathsf{Event}_{1,b}]|.\end{aligned}$$

Therefore, we complete the proof of the lemma by showing the following claims. □

Claim 1. ($\mathsf{Game}_{0,b}$ to $\mathsf{Game}_{1,b}$). *For any $b \in \{0,1\}$ and adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_1$ such that $|\Pr[\mathsf{Event}_{0,b}] - \Pr[\mathsf{Event}_{1,b}]| \leq \mathsf{Adv}^{\mathsf{P}_2}_{\mathcal{B}_1}(\kappa)$ and $\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A}) + Q_c \cdot \mathsf{poly}(\kappa, n)$ where $\mathsf{poly}(\kappa, n)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. We construct $\mathcal{B}_1$ that attacks Problem 2 from $\mathcal{A}$. Given the problem instance $(g_1, g_3, g_4, g_{12}, g_{24}, g, T)$ where $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_4}$ or $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_2p_4}$, $\mathcal{B}_1$ simulates $(Q_c, 1)$-MMH security game for $\mathcal{A}$ as follows.

Setup of Parameters. Given $\tau^\star$ from $\mathcal{A}$, $\mathcal{B}_1$ picks $\mathbf{w} \xleftarrow{\$} \mathbb{Z}_N^{d_1}$ and gives $(g_1, g_1^{\mathbf{w}}, g_3^{\mathbf{w}}, g_{24}, g_3, g_4)$ to $\mathcal{A}$.

Simulating $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{w}}(\cdot)$. Given S, $\mathcal{B}_1$ runs $\mathsf{CEnc}(S, N) \to (\mathbf{c}, d_3')$ and picks $\mathbf{s}' = (s_0', s_1', \ldots, s_{d_3'}') \xleftarrow{\$} \mathbb{Z}_N^{d_3'+1}$. Then it returns $g_{12}^{\mathbf{c}(\mathbf{s}',\mathbf{w})}$ to $\mathcal{A}$. Let g_{12} be $g_{12} = g_1^{a_1} \cdot g_2^{a_2}$ (where $a_1 \in \mathbb{Z}^*_{p_1}$ and $a_2 \in \mathbb{Z}^*_{p_2}$). Then we have $g_{12}^{\mathbf{c}(\mathbf{s}',\mathbf{w})} = g_1^{\mathbf{c}(a_1\mathbf{s}',\mathbf{w})} \cdot g_1^{\mathbf{c}(a_2\mathbf{s}',\mathbf{w})}$. This implicitly set $\mathbf{s} = a_1\mathbf{s}' \mod p_1$ and $\hat{\mathbf{s}} = a_2\mathbf{s}' \mod p_2$. Since $(\mathbf{s} \mod p_1, \hat{\mathbf{s}} \mod p_2)$ in $\mathsf{Game}_{0,b}$ and $\mathsf{Game}_{1,b}$ as well as $(a_1\mathbf{s}' \mod p_1, a_2\mathbf{s}' \mod p_2)$ in the simulation are uniformly distributed over $\mathbb{Z}_{p_1} \times \mathbb{Z}_{p_2}$ due to the Chinese Remainder Theorem, $\mathcal{B}_1$ correctly simulates the oracle.

Simulating $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$. When it is called, $\mathcal{B}_1$ runs $\mathsf{KEnc}(\tau^\star, N) \to \mathbf{k}$, picks $\hat{\alpha}' \xleftarrow{\$} \mathbb{Z}_N$, $\mathbf{r}' \xleftarrow{\$} \mathbb{Z}_N^{d_2'}$, and $\boldsymbol{\delta}' \xleftarrow{\$} \mathbb{Z}_N^{d_2}$. Then it returns $T^{\mathbf{k}(0,\mathbf{r}',\mathbf{w})} \cdot g_{24}^{\mathbf{k}(b\cdot\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}'}$ to $\mathcal{A}$.

We claim that $\mathcal{B}_1$ simulates $\mathsf{Game}_{0,b}$ if $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_4}$ and $\mathsf{Game}_{1,b}$ if $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_2p_4}$. Let T be $T = g_1^{t_1} \cdot g_2^{t_2} \cdot g_4^{t_4}$ and g_{24} be $g_{24} = g_2^{u_2} g_4^{u_4}$. Then we have

$$T^{\mathbf{k}(0,\mathbf{r}',\mathbf{w})} \cdot g_{24}^{\mathbf{k}(b\cdot\hat{\alpha}',\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}'} = g_1^{\mathbf{k}(0,t_1\mathbf{r}',\mathbf{w})} \cdot g_2^{\mathbf{k}(b\cdot u_2\hat{\alpha}',t_2\mathbf{r}',\mathbf{w})} \cdot g_4^{\boldsymbol{\delta}'+\mathbf{k}(bu_4\hat{\alpha}',t_4\mathbf{r}',\mathbf{w})}. \tag{5}$$

At first, we observe that the $\mathbb{G}_{p_4}$ component of the Equation (5) is uniformly random over $\mathbb{G}_{p_4}^{d_2}$ regardless of $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_4}$ or $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_2p_4}$, as desired.

If $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_4}$, we have $t_2 = 0$ and therefore $\mathbb{G}_{p_2}$ component of the Equation (5) is $g_2^{\mathbf{k}(b\cdot u_2\hat{\alpha}',\mathbf{0},\mathbf{w})} = g_2^{\mathbf{k}(b\cdot u_2\hat{\alpha}',\mathbf{0},\mathbf{0})}$. In this case, $\mathcal{B}_1$ implicitly sets $\mathbf{r} = t_1\mathbf{r}' \mod p_1$, $\hat{\alpha} = u_2\hat{\alpha}' \mod p_2$. It can be seen that $t_1\mathbf{r}' \mod p_1$ and $u_2\hat{\alpha}' \mod p_2$ are uniformly distributed over $\mathbb{Z}_{p_1}^{d_2'}$ and $\mathbb{Z}_{p_2}$ respectively. Therefore, $\mathcal{B}_1$ has correctly simulates $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ in $\mathsf{Game}_{0,b}$.

In the case of $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_2p_4}$, $\mathcal{B}_1$ implicitly sets $\mathbf{r} = t_1\mathbf{r}' \mod p_1$, $\hat{\mathbf{r}} = t_2\mathbf{r}' \mod p_2$, and $\hat{\alpha} = u_2\hat{\alpha}' \mod p_2$. Due to the Chinese Remainder Theorem, $(\mathbf{r} \mod p_1, \hat{\mathbf{r}} \mod p_2)$ in $\mathsf{Game}_{1,b}$ as well as $(t_1\mathbf{r}' \mod p_1, t_2\mathbf{r}' \mod p_2)$ in the simulation are uniformly distributed over $\mathbb{Z}_{p_1} \times \mathbb{Z}_{p_2}$. Furthermore, $\hat{\alpha} = u_2\hat{\alpha}' \mod p_2$ is also uniformly random over $\mathbb{Z}_{p_2}$. Therefore, $\mathcal{B}_1$ correctly simulates $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ in $\mathsf{Game}_{1,b}$.

Guess. Finally, $\mathcal{B}_1$ outputs $\mathcal{A}$'s output as its guess. As we have seen, the game corresponds to $\mathsf{Game}_{0,b}$ if $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_4}$ and $\mathsf{Game}_{1,b}$ if $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_2p_4}$. Thus, we may conclude that $|\Pr(\mathsf{Event}_{0,b}) - \Pr(\mathsf{Event}_{1,b})| \leq \mathsf{Adv}^{\mathsf{P}_2}_{\mathcal{B}_1}(\kappa)$. □

Claim 2. ($\mathsf{Game}_{1,0}$ to $\mathsf{Game}_{1,1}$). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_2$ such that $|\Pr[\mathsf{Event}_{1,0}] - \Pr[\mathsf{Event}_{1,1}]| \leq \mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{B}_2,\Pi,Q_c,\mathbb{G}_{p_2}}(\kappa)$ and $\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + Q_c \cdot \mathsf{poly}(\kappa, n)$ where $\mathsf{poly}(\kappa, n)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. We construct $\mathcal{B}_2$ against Q_c-CMH security of the broadcast encoding from $\mathcal{A}$.

Setup of Parameters. At the beginning of the game, $\mathcal{A}(1^\kappa, n)$ submits index $\tau^\star \in [n]$. $\mathcal{B}_2$ submits the same index and is given (g_1, g_2, g_3, g_4). Then, $\mathcal{B}_2$ picks $\bar{\mathbf{w}} \xleftarrow{\$} \mathbb{Z}_N^{d_1}$ and implicitly sets $\mathbf{w} = \bar{\mathbf{w}} \mod p_1$, $\mathbf{w} = \bar{\mathbf{w}} \mod p_3$, and $\mathbf{w} = \hat{\mathbf{w}} \mod p_2$ where $\hat{\mathbf{w}} \xleftarrow{\$} \mathbb{Z}_N^{d_1}$ is chosen by the game (and is not explicitly known to $\mathcal{B}_2$). By the Chinese Remainder Theorem, $\mathbf{w}$ is uniformly distributed over $\mathbb{Z}_N^{d_1}$. $\mathcal{B}_2$ also picks $a \xleftarrow{\$} \mathbb{Z}_N^*$ and gives $(g_1, g_1^{\mathbf{w}} = g_1^{\bar{\mathbf{w}}}, g_3^{\mathbf{w}} = g_3^{\bar{\mathbf{w}}}, g_{24} = (g_2 g_4)^a, g_3, g_4)$ to $\mathcal{A}$.

Simulating $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{w}}(\cdot)$. Given S from $\mathcal{A}$, $\mathcal{B}_2$ calls its oracle $\mathcal{O}^{\mathsf{CMH,C}}_{\tau^\star,\hat{\mathbf{w}}}(\cdot)$ on input S to obtain $g_2^{\mathbf{c}(\hat{\mathbf{s}},\hat{\mathbf{w}})}$. Then, it picks $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_N^{d_3'+1}$ and computes $g_1^{\mathbf{c}(\mathbf{s},\bar{\mathbf{w}})}$. Finally, it returns $g_1^{\mathbf{c}(\mathbf{s},\bar{\mathbf{w}})} \cdot g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})} = g_1^{\mathbf{c}(\mathbf{s},\bar{\mathbf{w}})} \cdot g_2^{\mathbf{c}(\hat{\mathbf{s}},\hat{\mathbf{w}})}$ to $\mathcal{A}$.

Simulating $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$. When it is called, $\mathcal{B}_2$ calls its oracle $\mathcal{O}^{\mathsf{CMH,K}}_{\tau^\star,\hat{\mathbf{w}},b}(\cdot)$ to obtain $g_2^{\mathbf{k}(b\hat{\alpha},\hat{\mathbf{r}},\hat{\mathbf{w}})}$. Then, it picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{d_2'}$, $\boldsymbol{\delta} \xleftarrow{\$} \mathbb{Z}_N^{d_2}$, computes $g_1^{\mathbf{k}(0,\mathbf{r},\bar{\mathbf{w}})}$, and gives $g_1^{\mathbf{k}(0,\mathbf{r},\mathbf{w})} \cdot g_2^{\mathbf{k}(b\hat{\alpha},\hat{\mathbf{r}},\mathbf{w})} \cdot g_4^{\boldsymbol{\delta}} = g_1^{\mathbf{k}(0,\mathbf{r},\bar{\mathbf{w}})} \cdot g_2^{\mathbf{k}(b\hat{\alpha},\hat{\mathbf{r}},\hat{\mathbf{w}})} \cdot g_4^{\boldsymbol{\delta}}$ to $\mathcal{A}$.

Guess. Finally, $\mathcal{B}_2$ outputs $\mathcal{A}$'s output as its guess. It is easy to see that the view of $\mathcal{A}$ corresponds to $\mathsf{Game}_{1,0}$ if $b = 0$ (i.e., $\mathcal{B}_2$ is equipped with the oracle $\mathcal{O}^{\mathsf{CMH,K}}_{\tau^\star,\hat{\mathbf{w}},0}(\cdot)$) and $\mathsf{Game}_{1,1}$ if $b = 1$ (i.e., $\mathcal{B}_2$ is equipped with the oracle $\mathcal{O}^{\mathsf{CMH,K}}_{\tau^\star,\hat{\mathbf{w}},1}(\cdot)$). Thus, we may conclude that $|\Pr[\mathsf{Event}_{1,0}] - \Pr[\mathsf{Event}_{1,1}]| \leq \mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{B}_2,\Pi,Q_c,\mathbb{G}_{p_2}}(\kappa)$. □

4 Almost Tight IBE from Broadcast Encoding in Composite-Order Groups

In this section, we show a generic conversion from a broadcast encoding scheme to an IBE scheme. An important property of the resulting IBE scheme is that (μ, Q_c, Q_k)-security of the scheme can be almost tightly reduced to the Q_c-CMH security of the underlying broadcast encoding scheme (and Problem 1, 2, 3, 4, and 5). In particular, the reduction only incurs small polynomial security loss, which is independent of μ and Q_k. Therefore, if the underlying broadcast encoding scheme is tightly Q_c-CMH secure, which is the case for all of our constructions, the resulting IBE scheme obtained by the conversion is almost tightly secure. Note that in the following construction, we have $\mathsf{sp} = \perp$. This mean that the key generation algorithm Par *does not output any secret parameter*. This property will be needed to convert our IBE scheme into CCA secure PKE scheme in Section 8.

Construction. Here, we construct an IBE scheme Φ^{comp} from a broadcast encoding $\Pi = (\mathsf{Param}, \mathsf{KEnc}, \mathsf{CEnc}, \mathsf{Pair})$. Let the identity space of the scheme be $\mathcal{ID} = \{0,1\}^\ell$ and the message space be $\mathcal{M} = \{0,1\}^m$. We also let $\mathcal{H}$ be a family of pairwise independent hash functions $\mathsf{H} : \mathbb{G}_T \to \mathcal{M}$. We assume that $\sqrt{\frac{2^m}{p_2}} = 2^{-\Omega(\kappa)}$ so that the left-over hash lemma can be applied in the security proof.

$\mathsf{Par}(1^\kappa)$: It first runs $(N, \mathbb{G}, \mathbb{G}_T, g_1, g_2, g_3, g_4, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{comp}}(1^\kappa)$ and $\mathsf{Param}(2\ell, N) \to d_1$. Then it picks $\mathbf{w} \xleftarrow{\$} \mathbb{Z}_N^{d_1}$, $a \xleftarrow{\$} \mathbb{Z}_N^*$, $\mathsf{H} \xleftarrow{\$} \mathcal{H}$ and sets $h := (g_1 g_2 g_3 g_4)^a$. Finally, it outputs $\mathsf{pp} = (g_1, g_1^{\mathbf{w}}, g_4, h, \mathsf{H})$ and $\mathsf{sp} = \perp$.

$\mathsf{Gen}(\mathsf{pp}, \mathsf{sp})$: It picks $\alpha \xleftarrow{\$} \mathbb{Z}_N$ and outputs $\mathsf{mpk} = (\mathsf{pp}, e(g_1, h)^\alpha)$ and $\mathsf{msk} = \alpha$.

$\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{ID})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ where $\mathsf{ID}_i \in \{0,1\}$ is the i-th bit of $\mathsf{ID} \in \{0,1\}^\ell$. Then it runs $\mathsf{KEnc}(j, N) \to (\mathbf{k}_j, d_2')$ and picks $\mathbf{r}_j \xleftarrow{\$} \mathbb{Z}_N^{d_2'}$ and $\boldsymbol{\delta}_j \xleftarrow{\$} \mathbb{Z}_N^{d_2}$ for all $j \in S$. It also picks random $\{\alpha_j \in \mathbb{Z}_N\}_{j \in S}$ subject to constraint that $\alpha = \sum_{j \in S} \alpha_j$. Then, it computes $g_1^{\mathbf{k}_j(0,\mathbf{r}_j,\mathbf{w})}$, $\mathsf{Pair}(j, S, N) \to \mathbf{E}_j$, and

$$\mathsf{sk}_j = h^{\mathbf{k}_j(\alpha_j \mathbf{0}, \mathbf{0})} \cdot g_1^{\mathbf{k}_j(0,\mathbf{r}_j,\mathbf{w})} \cdot g_4^{\boldsymbol{\delta}_j} \tag{6}$$

for all $j \in S$. Note that $g_1^{\mathbf{k}_j(0,\mathbf{r}_j,\mathbf{w})}$ can be computed from $g_1^{\mathbf{w}}$ and $\mathbf{r}_j = (r_{j,1}, \ldots, r_{j,d_2'})$ efficiently because $\mathbf{k}_j(0, \mathbf{r}_j, \mathbf{w})$ contains only linear combinations of monomials $r_{j,i}$, $r_{j,i} w_{j'}$. Finally, it outputs private key $\mathsf{sk}_{\mathsf{ID}} = \prod_{j \in S} (\mathsf{sk}_j)^{\mathbf{E}_j}$.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}, \mathsf{M})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$. Then it runs $\mathsf{CEnc}(S, N) \to (\mathbf{c}, d_3')$, picks $\mathbf{s} = (s_0, s_1, \ldots, s_{d_3'}) \xleftarrow{\$} \mathbb{Z}_N^{d_3'+1}$, and computes $g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})}$. Note that $g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})}$ can be computed from $g_1^{\mathbf{w}}$ and $\mathbf{s}$ efficiently because $\mathbf{c}(\mathbf{s}, \mathbf{w})$ contains only linear combinations of monomials s_i, $s_i w_j$. Finally, it outputs

$$\mathsf{CT} = \Big(C_1 = g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})}, \quad C_2 = \mathsf{H}\big(e(g_1, h)^{s_0\alpha}\big) \oplus \mathsf{M} \Big). \tag{7}$$

Here, $\oplus$ denotes bitwise exclusive OR of two bit strings.

$\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{CT})$: It parses $\mathsf{CT} \to (C_1, C_2)$ and computes $e(\mathsf{sk}_{\mathsf{ID}}^\top, C_1^\top) = e(g_1, h)^{s_0\alpha}$. Then, it recovers the message by $\mathsf{M} = C_2 \oplus \mathsf{H}(e(g_1, h)^{s_0\alpha})$.

CORRECTNESS. We show the correctness of the scheme. It suffices to show the following.

$$\begin{aligned}
e(\mathsf{sk}_{\mathsf{ID}}^\top, C_1^\top) &= e\big((\prod_{j\in S} (\mathsf{sk}_j)^{\mathbf{E}_j})^\top, g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})^\top}\big) \\
&= \prod_{j\in S} e\Big(\big(h^{\mathbf{k}_j(\alpha_j \mathbf{0}, \mathbf{0})} \cdot g_1^{\mathbf{k}_j(0, \mathbf{r}_j, \mathbf{w})} \cdot g_4^{\boldsymbol{\delta}_j}\big)^\top, g_1^{\mathbf{E}_j \mathbf{c}(\mathbf{s},\mathbf{w})^\top}\Big) \\
&= \prod_{j\in S} e\Big(\big(g_1^{\mathbf{k}_j(a\alpha_j, \mathbf{r}_j, \mathbf{w})} \cdot (g_2 g_3 g_4)^{\mathbf{k}(a\alpha_j, \mathbf{0}, \mathbf{0})} \cdot g_4^{\boldsymbol{\delta}_j}\big)^\top, g_1^{\mathbf{E}_j \mathbf{c}(\mathbf{s},\mathbf{w})^\top}\Big) \\
&= \prod_{j\in S} e(g_1, g_1)^{\mathbf{k}_j(a\alpha_j, \mathbf{r}_j, \mathbf{w}) \mathbf{E}_j \mathbf{c}(\mathbf{s},\mathbf{w})^\top} \\
&= \prod_{j\in S} e(g_1, g_1)^{s_0 a \alpha_j} = \prod_{j\in S} e(g_1, h)^{s_0 \alpha_j} = e(g_1, h)^{s_0\alpha}.
\end{aligned}$$

The fifth equation above follows from the correctness of the broadcast encoding.

REMARK. It would be more intuitive to set $\mathsf{sk}_{\mathsf{ID}}$ as $\{\mathsf{sk}_j\}_{j\in S}$ instead of defined as above. In such a case, the decryption algorithm first computes $e((\mathsf{sk}_j^{\mathbf{E}_j})^\top, C_1^\top) = e(g_1, g_1)^{\mathbf{k}(a\alpha_j, \mathbf{r}_j, \mathbf{w})^\top \mathbf{E}_j \mathbf{c}(\mathbf{s},\mathbf{w})} = e(g_1, h)^{s_0\alpha_j}$. Then it computes $\prod_{j\in S} e(g_1, h)^{s_0\alpha_j} = e(g_1, h)^{s_0\alpha}$ and recovers the message. Since the same S and $\{\mathbf{E}_j\}_{j\in S}$ are always used in the algorithm, it can be accelerated by pre-computing $\prod_{j\in S} \mathsf{sk}_j^{\mathbf{E}_j}$. In our scheme described above, we use this value as a private key. This optimization not only accelerate the decryption algorithm, but also makes the private key shorter.

Security. The following theorem indicates that the security of the IBE is (almost) tightly reduced to the MMH security of the underlying broadcast encoding on $\mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$ and Problem 1, 4, and 5. Combining the theorem with Theorem 1, the security of the scheme can be almost tightly reduced to the Q_c-CMH security of the underlying encoding (and Problem 1, 2, 3, 4, and 5). The reduction only incurs $O(\ell)$ security loss. The proof will appear in Appendix B.

Theorem 2. *For any adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}_i$ for $i \in [1, 5]$ such that*

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A}, \Phi^{\mathsf{comp}}, (\mu, Q_c, Q_k)}(\kappa) &\le \mathsf{Adv}^{\mathsf{P}_1}_{\mathcal{B}_1}(\kappa) + \mathsf{Adv}^{\mathsf{P}_5}_{\mathcal{B}_2}(\kappa) + Q_c \cdot 2^{-\Omega(\kappa)} \\
&+ \ell \Big(2\mathsf{Adv}^{\mathsf{P}_4}_{\mathcal{B}_3}(\kappa) + \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B}_4, \Pi, (Q_c, Q_k), \mathbb{G}_{p_2}}(\kappa) + \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B}_5, \Pi, (Q_c, Q_k), \mathbb{G}_{p_3}}(\kappa)\Big)
\end{aligned}$$

and $\max\{\mathsf{Time}(\mathcal{B}_i) | i \in [1, 5]\} \approx \mathsf{Time}(\mathcal{A}) + (\mu + Q_c + Q_k) \cdot \mathsf{poly}(\kappa, \ell)$ *where* $\mathsf{poly}(\kappa, \ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

5 Framework for Constructions in Prime-Order Groups

In Section 3 and 4, we show our framework to construct almost tightly secure IBE in composite-order groups. Since we carefully constructed the framework so that we only use the subgroup decision assumptions and the DBDH assumption in the security proof, we can apply recent composite-order-to-prime-order conversion techniques in the literature [19, 3, 17, 2] to the framework. We choose to use [3], but other choices might be possible. In this section, we show our framework for constructing almost tightly secure IBE in prime-order groups. Our framework is almost parallel to that in composite-order groups. Namely, we define CMH security and MMH security in prime-order groups. Then, we show reduction between them. Finally, we show a generic construction of IBE scheme from broadcast encoding and show that the scheme is (almost) tightly secure if the underlying encoding is tightly CMH secure.

In the following, we will use asymmetric bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ of prime order p with efficiently computable and non-degenerate bilinear map $e(\cdot) : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. For our purpose, we define a prime-order bilinear group generator $\mathcal{G}_{\mathsf{prime}}$ that takes as input a security parameter 1^κ and outputs $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, e(\cdot))$ where g and h are random generator of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Let $\pi_1 : \mathbb{Z}_p^{4\times 4} \to \mathbb{Z}_p^{4\times 2}$, $\pi_2 : \mathbb{Z}_p^{4\times 4} \to \mathbb{Z}_p^{4\times 1}$, and $\pi_3 : \mathbb{Z}_p^{4\times 4} \to \mathbb{Z}_p^{4\times 1}$ be the projection maps that map a 4×4 matrix to the leftmost 2 columns, the third column, and the fourth column, respectively.

Intuition. In prime-order groups, we work with 4×4 matrix. The first two dimensions serve as "normal space" (corresponding to $\mathbb{G}_{p_1}$), while the third and the fourth dimension serve as *double* "semi-functional spaces" (corresponding to $\mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$). There is no corresponding dimension to $\mathbb{G}_{p_4}$. While the use of 4×4 matrices is similar to Chen and Wee [18, 20][¶], conceptually, our techniques are quite different from theirs. They use the first two dimensions as a normal space and the last two dimensions as *single* semi-functional space. In contrast, we introduce additional semi-functional space to be able to prove the multi-challenge security rather than single-challenge security. Furthermore, due to our new proof technique, these semi-functional spaces are smaller compared to those of [18, 20].

5.1 Decisional Linear Assumption and Intermediate Problems

Here, we introduce the decisional linear problem (DLIN). Then, we introduce Problem 7, 8, 9, 10, 11, and 12. As we show in Appendix C.1, all of them are reduced to DLIN. These problems are used to prove the security of our proposed schemes in prime-order groups. The reason why we introduce these intermediate problems is to make security proof of our constructions clearer and modular.

We define the decisional-linear problem (DLIN) as follows.

Problem 6. (DLIN Problem.) *Let* $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{prime}}(1^\kappa)$. *We define advantage function for any adversary* $\mathcal{A}$ *as*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{A}}(\kappa) = |\Pr[\mathcal{A}\big(g, g^{a_1}, g^{a_2}, g^{a_3}, g^{a_1 s_1}, g^{a_2 s_2}, h, h^{a_1}, h^{a_2}, h^{a_3}, h^{a_1 s_1}, h^{a_2 s_2}, T_0\big) \to 1] - \\ \Pr[\mathcal{A}\big(g, g^{a_1}, g^{a_2}, g^{a_3}, g^{a_1 s_1}, g^{a_2 s_2}, h, h^{a_1}, h^{a_2}, h^{a_3}, h^{a_1 s_1}, h^{a_2 s_2}, T_1\big) \to 1]|\end{aligned}$$

where $a_1, a_2, a_3, \gamma \xleftarrow{\$} \mathbb{Z}_p^*$, $s_1, s_2 \xleftarrow{\$} \mathbb{Z}_p$, $T_0 = g^{a_3(s_1+s_2)}$, *and* $T_1 = g^{a_3(s_1+s_2)+\gamma}$. *We say that the* DLIN *assumption holds if* $\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{A}}(\kappa)$ *is negligible for any PPT adversary* $\mathcal{A}$.

We can also define a slight variant of the above problem where T_0 and T_1 are set as $T_0 = h^{a_3(s_1+s_2)}$ and $T_1 = h^{a_3(s_1+s_2)+\gamma}$. We abuse notation and denote the advantage function for $\mathcal{A}$ for the variant also as $\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{A}}(\kappa)$. Note that we can define several weaker variants of the above assumption. For example, one can

[¶]They showed a construction that is secure under the k-LIN assumption for any k, using $2k \times 2k$ matrices. When $k = 2$, the scheme is secure under the DLIN assumption.

remove $h^{a_1 s_1}$ and $h^{a_2 s_2}$ from the input to $\mathcal{A}$. Such a variant is a weaker assumption than the above one. We define the strongest form of the assumption for the sake of simplicity. However, none of our reductions in this paper needs all terms in the problem instance.

REMARKS. Typically, we sample $a_1, a_2, a_3, s_1, s_2, \gamma \xleftarrow{\$} \mathbb{Z}_p$; this only yields a $4/p$ negligible difference in the advantage.

Intermediate Problems. In the following, we introduce intermediate problems. As we show in Appendix C.1, the hardness of these problems can be tightly reduced to the DLIN assumption. We also show that these problems have random self-reducibility in Appendix C.2.

Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{prime}}(1^\kappa)$ and $\mathbf{B} \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$. Also let $\mathbf{D} \in \mathbb{Z}_p^{4\times 4}$ be a random full rank diagonal matrix for which the entries $(3,3)$ and $(4,4)$ are 1. Then, $\mathbf{Z} \in \mathbb{Z}_p^{4\times 4}$ is defined as $\mathbf{Z} = \mathbf{B}^{-\top}\mathbf{D}$. We define advantage function $\mathsf{Adv}^{\mathsf{Pxx}}_{\mathcal{A}}(\kappa)$ for Problem xx for any adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{Pxx}}_{\mathcal{A}}(\kappa) = |\Pr[\mathcal{A}(g, h, D, T_0) \to 1] - \Pr[\mathcal{A}(g, h, D, T_1) \to 1]|.$$

In each problem, D, T_0, and T_1 are defined as follows.

Problem 7. $D = (g^{\mathbf{B}}, h^{\pi_1(\mathbf{Z})})$, $T_0 = g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\0\\0\end{pmatrix}}$, *and* $T_1 = g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\\hat{t}\\0\end{pmatrix}}$ *where* $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ *and* $\hat{t} \xleftarrow{\$} \mathbb{Z}_p^*$.

Problem 8. $D = (g^{\pi_1(\mathbf{B})}, g^{\pi_3(\mathbf{B})}, g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\\hat{t}\\0\end{pmatrix}}, h^{\mathbf{Z}})$, $T_0 = h^{\mathbf{Z}\begin{pmatrix}\mathbf{u}\\0\\0\end{pmatrix}}$, *and* $T_1 = h^{\mathbf{Z}\begin{pmatrix}\mathbf{u}\\\hat{u}\\0\end{pmatrix}}$ *where* $\mathbf{t}, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ *and* $\hat{t}, \hat{u} \xleftarrow{\$} \mathbb{Z}_p^*$.

Problem 9. $D = (g^{\mathbf{B}}, h^{\pi_1(\mathbf{Z})}, h^{\theta_{1,1}\hat{\mathbf{f}}+\theta_{2,1}\tilde{\mathbf{f}}}, h^{\theta_{1,2}\hat{\mathbf{f}}+\theta_{2,2}\tilde{\mathbf{f}}})$, $T_0 = g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\\hat{t}\\0\end{pmatrix}}$, *and* $T_1 = g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\\hat{t}\\\tilde{t}\end{pmatrix}}$ *where* $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\hat{t}, \tilde{t} \xleftarrow{\$} \mathbb{Z}_p^*$, $\mathbf{\Theta} = \begin{pmatrix}\theta_{1,1} & \theta_{1,2}\\ \theta_{2,1} & \theta_{2,2}\end{pmatrix} \xleftarrow{\$} \mathbb{GL}_2(\mathbb{Z}_p)$, $\hat{\mathbf{f}} = \mathbf{Z}\mathbf{e}_3^\top$, *and* $\tilde{\mathbf{f}} = \mathbf{Z}\mathbf{e}_4^\top$.

Problem 10. $D = (g^{\mathbf{B}}, h^{\pi_1(\mathbf{Z})}, h^{\theta_{1,1}\hat{\mathbf{f}}+\theta_{2,1}\tilde{\mathbf{f}}}, h^{\theta_{1,2}\hat{\mathbf{f}}+\theta_{2,2}\tilde{\mathbf{f}}})$, $T_0 = g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\\hat{t}\\0\end{pmatrix}}$, *and* $T_1 = g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\0\\\tilde{t}\end{pmatrix}}$ *where* $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\hat{t}, \tilde{t} \xleftarrow{\$} \mathbb{Z}_p^*$, $\mathbf{\Theta} = \begin{pmatrix}\theta_{1,1} & \theta_{1,2}\\ \theta_{2,1} & \theta_{2,2}\end{pmatrix} \xleftarrow{\$} \mathbb{GL}_2(\mathbb{Z}_p)$, $\hat{\mathbf{f}} = \mathbf{Z}\mathbf{e}_3^\top$, *and* $\tilde{\mathbf{f}} = \mathbf{Z}\mathbf{e}_4^\top$.

Problem 11. $D = (g^x, g^y, h^z)$, $T_0 = e(g,h)^{xyz}$, *and* $T_1 = e(g,h)^{xyz+\gamma}$ *where* $x, y, z \xleftarrow{\$} \mathbb{Z}_p$ *and* $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$.

Problem 12. $D = \emptyset$, $T_0 = g^{\mathbf{X}_0}$, *and* $T_1 = g^{\mathbf{X}_1}$ *where* $\mathbf{X}_0 \xleftarrow{\$} \mathsf{Rk}_2(\mathbb{Z}_p^{6\times 6})$ *and* $\mathbf{X}_1 \xleftarrow{\$} \mathsf{Rk}_6(\mathbb{Z}_p^{6\times 6})$. *Here, we denote by* $\mathsf{Rk}_i(\mathbb{Z}_p^{a\times b})$ *the set of all* $a \times b$ *matrices over* $\mathbb{Z}_p$ *with rank* i.

Note that quite similar problems to Problem 7 and 8 are introduced in [18, 19]. The problem 9 and 10 seem to be new. Roughly speaking, Problem 7, 8, 9, and 10 are analogue of sub-group decision assumption in composite-order groups. The problem 11 is called the decisional bilinear Diffie-Hellman problem in the literature. Problem 12 is the Matrix DLIN assumption, which is introduced in [44].

5.2 Computational-Master-Key Hiding Security on Prime Order Groups

Here, we define Q-computational-master-key hiding (Q-CMH) security on $(\mathbb{G}_1, \mathbb{G}_2)$ for a broadcast encoding $\Pi = (\mathsf{Param}, \mathsf{KEnc}, \mathsf{CEnc}, \mathsf{Pair})$. The definition is quite similar to that on composite-order groups. Consider the following game:

At the beginning of the game, an (stateful) adversary $\mathcal{A}$ is given $(1^\kappa, n)$ and chooses $\tau^\star \in [n]$. Then, parameters are chosen as $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{prime}}(1^\kappa)$, $d_1 \leftarrow \mathsf{Param}(n, p)$, and $\hat{\mathbf{w}} \xleftarrow{\$} \mathbb{Z}_p^{d_1}$. The advantage of $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{A},\Pi,Q,(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) = |\Pr[\mathcal{A}(1^\kappa, n) \to \tau^\star,\ \mathcal{A}(g,h)^{\mathcal{O}^{\mathsf{CMH,C}}_{\tau^\star,\hat{\mathbf{w}}}(\cdot),\mathcal{O}^{\mathsf{CMH,K}}_{\tau^\star,\hat{\mathbf{w}},0}(\cdot)} \to 1] - \Pr[\mathcal{A}(1^\kappa, n) \to \tau^\star,\ \mathcal{A}(g,h)^{\mathcal{O}^{\mathsf{CMH,C}}_{\tau^\star,\hat{\mathbf{w}}}(\cdot),\mathcal{O}^{\mathsf{CMH,K}}_{\tau^\star,\hat{\mathbf{w}},1}(\cdot)} \to 1]|.$$

In the above, $\mathcal{O}^{\mathsf{CMH,K}}_{\tau^\star,\hat{\mathbf{w}},b}(\cdot)$ for $b \in \{0,1\}$ are called only once while $\mathcal{O}^{\mathsf{CMH,C}}_{\tau^\star,\hat{\mathbf{w}}}(\cdot)$ can be called at most Q times. These oracles can be called in any order.

- $\mathcal{O}^{\mathsf{CMH,C}}_{\tau^\star,\hat{\mathbf{w}}}(\cdot)$ takes $S \subset [n]$ such that $\tau^\star \notin S$ as input. It then runs $\mathsf{CEnc}(S, p) \to (\mathbf{c}, d_3')$, picks $\hat{\mathbf{s}} \xleftarrow{\$} \mathbb{Z}_p^{d_3'+1}$, and returns $g^{\mathbf{c}(\hat{\mathbf{s}},\hat{\mathbf{w}})}$.
- $\mathcal{O}^{\mathsf{CMH,K}}_{\tau^\star,\hat{\mathbf{w}},b}(\cdot)$ ignores its input. When it is called, it first runs $\mathsf{KEnc}(\tau^\star, p) \to (\mathbf{k}, d_2')$ and picks $\hat{\mathbf{r}} \xleftarrow{\$} \mathbb{Z}_p^{d_2'}$ and $\hat{\alpha} \xleftarrow{\$} \mathbb{Z}_p$. Then it returns $h^{\mathbf{k}(b\cdot\hat{\alpha},\hat{\mathbf{r}},\hat{\mathbf{w}})} = \begin{cases} h^{\mathbf{k}(0,\hat{\mathbf{r}},\hat{\mathbf{w}})} & \text{if } b = 0 \\ h^{\mathbf{k}(\hat{\alpha},\hat{\mathbf{r}},\hat{\mathbf{w}})} & \text{if } b = 1. \end{cases}$

We say that the broadcast encoding is Q-CMH secure on $(\mathbb{G}_1, \mathbb{G}_2)$ if $\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{A},\Pi,Q,(\mathbb{G}_1,\mathbb{G}_2)}(\kappa)$ is negligible for all PPT adversary $\mathcal{A}$. As in composite-order groups, Q-PMH security unconditionally implies Q-CMH security. The proof of the lemma appears in Appendix D.1.

Lemma 4. *Assume that a broadcast encoding Π satisfies Q-PMH security for some $Q \in \mathbb{N}$. Then it follows that $\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{A},\Pi,Q,(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) \le d_2'/p$.*

5.3 Preparation

Here, we introduce notation needed to define multi-master-key hiding (MMH) security in prime-order groups. Here, we assign each variables in polynomials $\mathbf{k}$ and $\mathbf{c}$ vectors or matrices, rather than scalar values (as in the definition of the CMH security). Let p be a prime number. We assign each variable w_i a matrix $\mathbf{W}_i \in \mathbb{Z}_p^{4\times 4}$ for $i \in [d_1]$, variable α a column vector $\boldsymbol{\alpha} \in \mathbb{Z}_p^{4\times 1}$, variable r_i a vector $\mathbf{x}_i \in \mathbb{Z}_p^{4\times 1}$ for $i \in [d_2']$, and variable s_i a vector $\mathbf{y}_i \in \mathbb{Z}_p^{4\times 1}$ for $i \in [0, d_3']$. The evaluation of polynomials $\mathbf{k}_{\mathbf{Z}}$ and $\mathbf{c}_{\mathbf{B}}$, which are indexed by an invertible matrix $\mathbf{B} \in \mathbb{Z}_p^{4\times 4}$ and $\mathbf{Z} \in \mathbb{Z}_p^{4\times 4}$, are defined as follows. In the following, we denote

$$\begin{aligned} \mathbb{W} &= (\mathbf{W}_1, \dots, \mathbf{W}_{d_1}) \in (\mathbb{Z}_p^{4\times 4})^{d_1}, \quad \mathbf{X} = \left(\mathbf{x}_1, \dots, \mathbf{x}_{d_2'}\right) \in \mathbb{Z}_p^{4\times d_2'} \\ \mathbf{Y} &= \left(\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{d_3'}\right) \in \mathbb{Z}_p^{4\times(d_3'+1)}, \qquad \mathbf{Z} = (\mathbf{B}^{-1})^\top \cdot \mathbf{D}. \end{aligned}$$

where $\mathbf{D} \in \mathbb{Z}_p^{4\times 4}$ is a full-rank diagonal matrix with the entries $(3,3)$ and $(4,4)$ being 1. Let $\mathbf{k} = (k_1, \dots, k_{d_2})$ be a vector of polynomials in variables $\alpha, r_1, \dots, r_{d_2'}, w_1, \dots, w_{d_1}$ with coefficients in $\mathbb{Z}_p$ such that

$$k_\iota\left(\alpha, r_1, \dots, r_{d_2'}, w_1, \dots, w_{d_1}\right) = b_\iota \alpha + \left(\sum_{j\in[d_2']} b_{\iota,j} r_j\right) + \left(\sum_{(j,k)\in[d_2']\times[d_1]} b_{\iota,j,k} w_k r_j\right)$$

for $\iota \in [d_2]$. We define $\mathbf{k}_{\mathbf{Z}}(\boldsymbol{\alpha}, \mathbf{X}, \mathbb{W}) \in \mathbb{Z}_p^{4\times d_2}$ as $\mathbf{k}_{\mathbf{Z}}(\boldsymbol{\alpha}, \mathbf{X}, \mathbb{W}) =$

$$\left\{ k_{\mathbf{Z},\iota}\left(\boldsymbol{\alpha}, \mathbf{X}, \mathbb{W}\right) = b_\iota \boldsymbol{\alpha} + \left(\sum_{j\in[d_2']} b_{\iota,j} \mathbf{Z}\mathbf{x}_j\right) + \left(\sum_{(j,k)\in[d_2']\times[d_1]} b_{\iota,j,k} \mathbf{W}_k^\top \mathbf{Z}\mathbf{x}_j\right) \in \mathbb{Z}_p^{4\times 1} \right\}_{\iota\in[d_2]}.$$

Let $\mathbf{c} = (c_1, \ldots, c_{d_3})$ be a vector of polynomials in variables $s_0, s_1, \ldots, s_{d'_3}, w_1, \ldots, w_{d_1}$ with coefficients in $\mathbb{Z}_p$ such that

$$c_\iota\Big(s_0, s_1, \ldots, s_{d'_3}, w_1, \ldots, w_{d_1}\Big) = \Big(\sum_{j\in[0,d'_3]} a_{\iota,j} s_j\Big) + \Big(\sum_{(j,k)\in[0,d'_3]\times[d_1]} a_{\iota,j,k} w_k s_j\Big)$$

for $\iota \in [d_3]$. We define $\mathbf{c}_{\mathbf{B}}(\mathbf{Y}, \mathbb{W}) \in \mathbb{Z}_p^{4\times d_3}$ as $\mathbf{c}_{\mathbf{B}}(\mathbf{Y}, \mathbb{W}) =$

$$\left\{ c_{\mathbf{B},\iota}(\mathbf{Y}, \mathbb{W}) = \Big(\sum_{j\in[0,d'_3]} a_{\iota,j} \mathbf{B}\mathbf{y}_j\Big) + \Big(\sum_{(j,k)\in[0,d'_3]\times[d_1]} a_{\iota,j,k} \mathbf{W}_k \mathbf{B}\mathbf{y}_j\Big) \in \mathbb{Z}_p^{4\times 1} \right\}_{\iota\in[d_3]}.$$

Restriction on the Encoding. In our framework for prime-order constructions, we define and require *regularity* of encoding similarly to [3], which is needed to prove the security of our IBE obtained from the broadcast encoding. Note that all of broadcast encoding schemes that appear in this paper satisfy the requirement. Compared to the definition in [3], our definition is less general, but simpler and sufficient for our purpose.

Definition 3 (Regularity.)**.** *We call a broadcast encoding* regular *if the following hold:*

1. *For $\iota \in [1, d_2], \iota' \in [1, d_3]$ such that there is $j \in [1, d'_2], k \in [1, d_1], j' \in [0, d'_3], k' \in [1, d_1]$ where $b_{\iota,j,k} \neq 0$ and $a_{\iota',j',k'} \neq 0$, we require that $E_{\iota,\iota'} = 0$.*
2. *For all $j \in [1, d'_2]$, there is $i \in [d_2]$ such that $k_i = r_j$. Similarly, for all $j' \in [0, d'_3]$, there is $i' \in [d_3]$ such that $c_{i'} = s_{j'}$.*

Correctness of Encoding. Let $\tau \in [n]$ and $S \subseteq [n]$ be an index and a set such that $\tau \in S$. Let also $\mathsf{KEnc}(\tau, p) \to (\mathbf{k}, d'_2)$, $\mathsf{CEnc}(S, p) \to (\mathbf{c}, d'_3)$, and $\mathsf{Pair}(\tau, S, p) \to \mathbf{E} = (E_{\eta,\iota})_{(\eta,\iota)\in[d_2]\times[d_3]} \in \mathbb{Z}_p^{d_2\times d_3}$. Then, by the correctness of the broadcast encoding, we have $\sum_{(\eta,\iota)\in[d_2]\times[d_3]} E_{\eta,\iota} k_\eta c_\iota = \alpha s_0$ (the equation holds symbolically). From this, we have the following. (Note that the claim is shown similarly to Claim 15 in [3].) The proof will appear in Appendix D.2.

Lemma 5. *We have $\sum_{(\eta,\iota)\in[d_2]\times[d_3]} E_{\eta,\iota} \cdot k_{\mathbf{Z},\eta}(\boldsymbol{\alpha}, \mathbf{X}, \mathbb{W})^\top c_{\mathbf{B},\iota}(\mathbf{Y}, \mathbb{W}) = \boldsymbol{\alpha}^\top \mathbf{B}\mathbf{y}_0$.*

5.4 Multi-Master-Key Hiding Security in Prime-Order Groups

Here, we define multi-master-key hiding (MMH) security and its variant. The former and the latter are equivalent, but it is convenient to define both. Then, we show that the CMH security implies MMH security, similarly to the case of composite-order groups.

MMH Security in Prime-Order Groups. We define (Q_c, Q_k)-multi-master-key hiding (MMH) security of a broadcast encoding $\Pi = (\mathsf{Param}, \mathsf{KEnc}, \mathsf{CEnc}, \mathsf{Pair})$ (in prime-order groups) by the following game:

At the beginning of the game, $\mathcal{A}$ is given $(1^\kappa, n)$ and chooses $\tau^\star \in [n]$. Then, parameters are chosen as $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{prime}}(1^\kappa)$, $d_1 \leftarrow \mathsf{Param}(n, p)$, $\mathbf{B} \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$, $\mathbb{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_{d_1}) \xleftarrow{\$} (\mathbb{Z}_p^{4\times 4})^{d_1}$. A random full-rank diagonal matrix $\mathbf{D} \in \mathbb{Z}_p^{4\times 4}$ with the entries $(3, 3)$ and $(4, 4)$ being 1, is also chosen. The matrix $\mathbf{Z} \in \mathbb{Z}_p^{4\times 4}$ is set as $\mathbf{Z} := (\mathbf{B}^{-1})^\top \mathbf{D}$. The advantage of $\mathcal{A}$ is defined as

$$\begin{aligned}\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,Q_k),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) = |\Pr[\mathcal{A}(1^\kappa, n) \to \tau^\star,\ \mathcal{A}(\mathsf{params})^{\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot),\ \mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},0}(\cdot)} \to 1] \\ - \Pr[\mathcal{A}(1^\kappa, n) \to \tau^\star,\ \mathcal{A}(\mathsf{params})^{\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot),\ \mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},1}(\cdot)} \to 1]|\end{aligned}$$

where

$$\mathsf{params} = \begin{pmatrix} g, & g^{\pi_1(\mathbf{B})}, & g^{\pi_3(\mathbf{B})}, & g^{\pi_1(\mathbf{W}_1\mathbf{B})}, & \dots, & g^{\pi_1(\mathbf{W}_{d_1}\mathbf{B})}, & g^{\pi_3(\mathbf{W}_1\mathbf{B})}, & \dots, & g^{\pi_3(\mathbf{W}_{d_1}\mathbf{B})} \\ h, & h^{\mathbf{Z}}, & & h^{\pi_1(\mathbf{W}_1^\top\mathbf{Z})}, & \dots, & h^{\pi_1(\mathbf{W}_{d_1}^\top\mathbf{Z})}, & h^{\pi_3(\mathbf{W}_1^\top\mathbf{Z})}, & \dots, & h^{\pi_3(\mathbf{W}_{d_1}^\top\mathbf{Z})} \end{pmatrix}.$$

In the above, $\mathcal{O}^{\mathsf{MMH},\mathsf{C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot)$ is called at most Q_c times, while $\mathcal{O}^{\mathsf{MMH},\mathsf{K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ for $b \in \{0,1\}$ are called at most Q_k times. These oracles can be called in any order.

- $\mathcal{O}^{\mathsf{MMH},\mathsf{C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot)$ takes $S \subseteq [n]\backslash\{\tau^\star\}$ as input. It then picks $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{d_3'} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\hat{s}_0, \hat{s}_1, \dots, \hat{s}_{d_3'} \xleftarrow{\$} \mathbb{Z}_p$, and runs $\mathsf{CEnc}(S,p) \to (\mathbf{c}, d_3')$. Then it sets $\mathbf{S} \in \mathbb{Z}_p^{4\times(d_3'+1)}$ and $\hat{\mathbf{S}} \in \mathbb{Z}_p^{4\times(d_3'+1)}$ as

$$\mathbf{S} = \left(\begin{pmatrix} \mathbf{s}_0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{s}_1 \\ 0 \\ 0 \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{s}_{d_3'} \\ 0 \\ 0 \end{pmatrix} \right) \text{ and } \hat{\mathbf{S}} = \left(\begin{pmatrix} \mathbf{0} \\ \hat{s}_0 \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ \hat{s}_1 \\ 0 \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{0} \\ \hat{s}_{d_3'} \\ 0 \end{pmatrix} \right) \tag{8}$$

  and returns $g^{\mathbf{c}_\mathbf{B}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})}$.

- $\mathcal{O}^{\mathsf{MMH},\mathsf{K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ ignores its input. When it is called, it first runs $\mathsf{KEnc}(\tau^\star,p) \to (\mathbf{k}, d_2')$, picks $\hat{\alpha} \xleftarrow{\$} \mathbb{Z}_p$, $\mathbf{r}_1, \dots, \mathbf{r}_{d_2'} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, and sets $\mathbf{R} = \left(\begin{pmatrix} \mathbf{r}_1 \\ 0 \\ 0 \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{r}_{d_2'} \\ 0 \\ 0 \end{pmatrix} \right) \in \mathbb{Z}_p^{4\times d_2'}$. Finally, it returns

$$h^{\mathbf{k}_\mathbf{Z}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\mathbf{R},\mathbb{W})} = \begin{cases} h^{\mathbf{k}_\mathbf{Z}(\mathbf{0},\mathbf{R},\mathbb{W})} & \text{if } b = 0 \\ h^{\mathbf{k}_\mathbf{Z}(\hat{\alpha}\cdot\hat{\mathbf{f}},\mathbf{R},\mathbb{W})} & \text{if } b = 1 \end{cases}$$

  where $\hat{\mathbf{f}} = \pi_2(\mathbf{Z}) = \mathbf{Z}\cdot\mathbf{e}_3^\top$.

In the above, we note that $\mathbf{S}$ and $\hat{\mathbf{S}}$ as well as $\mathbf{R}$ and $\hat{\alpha}$ are freshly chosen every time the corresponding oracle is called. We say that the broadcast encoding is (Q_c, Q_k)-MMH secure on $(\mathbb{G}_1, \mathbb{G}_2)$ if $\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,Q_k),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa)$ is negligible for all PPT adversary $\mathcal{A}$.

A Variant of MMH-Security. It is convenient to consider a variant of the (Q_c, Q_k)-MMH security, that we call (Q_c, Q_k)-MMH' security. (Q_c, Q_k)-MMH' security is defined by a game that is the same as (Q_c, Q_k)-MMH security game above except that

- params is replaced with

$$\mathsf{params} = \begin{pmatrix} g, & g^{\pi_1(\mathbf{B})}, & g^{\pi_2(\mathbf{B})}, & g^{\pi_1(\mathbf{W}_1\mathbf{B})}, & \dots, & g^{\pi_1(\mathbf{W}_{d_1}\mathbf{B})}, & g^{\pi_2(\mathbf{W}_1\mathbf{B})}, & \dots, & g^{\pi_2(\mathbf{W}_{d_1}\mathbf{B})} \\ h, & h^{\mathbf{Z}}, & & h^{\pi_1(\mathbf{W}_1^\top\mathbf{Z})}, & \dots, & h^{\pi_1(\mathbf{W}_{d_1}^\top\mathbf{Z})}, & h^{\pi_2(\mathbf{W}_1^\top\mathbf{Z})}, & \dots, & h^{\pi_2(\mathbf{W}_{d_1}^\top\mathbf{Z})} \end{pmatrix}.$$

- $\mathcal{O}^{\mathsf{MMH}',\mathsf{C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot)$ returns $g^{\mathbf{c}_\mathbf{B}(\mathbf{S}+\tilde{\mathbf{S}},\mathbb{W})}$ where $\mathbf{S} = \left(\begin{pmatrix} \mathbf{s}_0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{s}_1 \\ 0 \\ 0 \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{s}_{d_3'} \\ 0 \\ 0 \end{pmatrix} \right) \in \mathbb{Z}_p^{4\times(d_3'+1)}$, $\tilde{\mathbf{S}} = \left(\begin{pmatrix} \mathbf{0} \\ 0 \\ \tilde{s}_0 \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ 0 \\ \tilde{s}_1 \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{0} \\ 0 \\ \tilde{s}_{d_3'} \end{pmatrix} \right) \in \mathbb{Z}_p^{4\times(d_3'+1)}$, $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{d_3'} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, and $\tilde{s}_0, \tilde{s}_1, \dots, \tilde{s}_{d_3'} \xleftarrow{\$} \mathbb{Z}_p$.

- $\mathcal{O}^{\mathsf{MMH}',\mathsf{K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ returns $h^{\mathbf{k}_\mathbf{Z}(b\cdot\tilde{\alpha}\tilde{\mathbf{f}},\mathbf{R},\mathbb{W})}$ where $\tilde{\alpha} \xleftarrow{\$} \mathbb{Z}_p$, $\mathbf{r}_1, \dots, \mathbf{r}_{d_2'} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\mathbf{R} = \left(\begin{pmatrix} \mathbf{r}_1 \\ 0 \\ 0 \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{r}_{d_2'} \\ 0 \\ 0 \end{pmatrix} \right) \in \mathbb{Z}_p^{4\times d_2'}$, and $\tilde{\mathbf{f}} = \pi_3(\mathbf{Z}) = \mathbf{Z}\mathbf{e}_4^\top$.

We define the advantage of the adversary in this modified game as $\mathsf{Adv}^{\mathsf{MMH}'}_{\mathcal{A},\Pi,(Q_c,Q_k),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa)$. If we exchange the third and the fourth column of $\mathbf{B}$ and $\mathbf{Z}$, (Q_c, Q_k)-MMH' security game defined as above corresponds to (Q_c, Q_k)-MMH security game. Therefore, the following lemma immediately follows.

Lemma 6. *For any broadcast encoding* Π *and adversary* $\mathcal{A}$*, there exists another adversary* $\mathcal{B}$ *such that* $\mathsf{Adv}^{\mathsf{MMH}'}_{\mathcal{A},\Pi,(Q_c,Q_k),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) = \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B},\Pi,(Q_c,Q_k),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A})$.

CMH Security Implies MMH Security. Similarly to the case in composite-order groups, we have that CMH security tightly implies MMH security (assuming the hardness of the Problem 8). More formally, we have the following theorem:

Theorem 3. *For any broadcast encoding* Π *and adversary* $\mathcal{A}$*, there exist adversaries* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *such that*

$$\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,Q_k),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) \leq \mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{B}_1,\Pi,Q_c,(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) + 2\mathsf{Adv}^{\mathsf{P}_8}_{\mathcal{B}_2} + \frac{1}{p}$$

and $\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2)\} \approx \mathsf{Time}(\mathcal{A}) + (Q_k + Q_c) \cdot \mathsf{poly}(\kappa, n)$ *where* $\mathsf{poly}(\kappa, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

The proof of the theorem is almost parallel to that of Theorem 1 and appears in Appendix D.3.

5.5 Almost Tightly Secure IBE from Broadcast Encoding in Prime Order Groups

Here, we construct an IBE scheme Φ^{prime} from broadcast encoding scheme $\Pi = (\mathsf{Param}, \mathsf{KEnc}, \mathsf{CEnc}, \mathsf{Pair})$. Let the identity space of Φ^{prime} be $\mathcal{ID} = \{0,1\}^\ell$ and the message space $\mathcal{M}$ be $\mathcal{M} = \mathbb{G}_T$. We will not use pairwise independent hash function differently from our construction in composite-order groups. We note that similarly to our construction in composite-order groups, we have $\mathsf{sp} = \bot$ in the following.

$\mathsf{Par}(1^\kappa, \ell)$: It first runs $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{prime}}(1^\kappa)$ and $\mathsf{Param}(2\ell, p) \to d_1$. Then it picks $\mathbf{B} \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$, $\mathbb{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_{d_1}) \xleftarrow{\$} (\mathbb{Z}_p^{4\times 4})^{d_1}$ and a random full-rank diagonal matrix $\mathbf{D} \in \mathbb{Z}_p^{4\times 4}$ with the entries $(3,3)$ and $(4,4)$ being 1. Finally, it sets $\mathbf{Z} = \mathbf{B}^{-\top}\mathbf{D}$ and outputs

$$\mathsf{pp} = \begin{pmatrix} g, & g^{\pi_1(\mathbf{B})}, & g^{\pi_1(\mathbf{W}_1\mathbf{B})}, & \ldots, & g^{\pi_1(\mathbf{W}_{d_1}\mathbf{B})} \\ h, & h^{\pi_1(\mathbf{Z})}, & h^{\pi_1(\mathbf{W}_1^\top\mathbf{Z})}, & \ldots, & h^{\pi_1(\mathbf{W}_{d_1}^\top\mathbf{Z})} \end{pmatrix} \quad \text{and} \quad \mathsf{sp} = \bot.$$

In the following, we will omit subscript $\mathbf{B}$ and $\mathbf{Z}$ from $\mathbf{c}_{\mathbf{B}}(\mathbf{S}, \mathbb{W})$ and $\mathbf{k}_{\mathbf{Z}}(\boldsymbol{\alpha}, \mathbf{R}, \mathbb{W})$ and just denote $\mathbf{c}(\mathbf{S}, \mathbb{W})$ and $\mathbf{k}(\boldsymbol{\alpha}, \mathbf{R}, \mathbb{W})$ for ease of notation. $\mathbf{B}$ and $\mathbf{Z}$ are fixed in the following and clear from the context.

$\mathsf{Gen}(\mathsf{pp})$: It picks $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{Z}_p^{4\times 1}$ and outputs $\mathsf{mpk} = (\mathsf{pp}, e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})})$ and $\mathsf{msk} = \boldsymbol{\alpha}$.

$\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{ID})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ where $\mathsf{ID}_i \in \{0,1\}$ is the i-th bit of $\mathsf{ID} \in \{0,1\}^\ell$. Then it runs $\mathsf{KEnc}(j, p) \to (\mathbf{k}_j, d_2')$, picks $\mathbf{r}_{j,1}, \ldots, \mathbf{r}_{j,d_2'} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, and sets $\mathbf{R}_j = \left(\begin{pmatrix} \mathbf{r}_{j,1} \\ 0 \\ 0 \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{r}_{j,d_2'} \\ 0 \\ 0 \end{pmatrix} \right) \in \mathbb{Z}_p^{4\times d_2'}$ for all $j \in S$. It also picks random $\{\boldsymbol{\alpha}_j \in \mathbb{Z}_p^{4\times 1}\}_{j\in S}$ subject to constraint that $\boldsymbol{\alpha} = \sum_{j\in S} \boldsymbol{\alpha}_j$. Then, it computes $\mathsf{Pair}(j, S, p) \to \mathbf{E}_j = (E_{j,\eta,\iota})_{(\eta,\iota)\in[d_2]\times[d_3]}$ and

$$\mathsf{sk}_j = h^{\mathbf{k}_j(\boldsymbol{\alpha}_j, \mathbf{R}_j, \mathbb{W})} = \{\mathsf{sk}_{j,\eta} = h^{k_{j,\eta}(\boldsymbol{\alpha}_j, \mathbf{R}_j, \mathbb{W})}\}_{\eta\in[d_2]} \tag{9}$$

for all $j \in S$. Note that $h^{\mathbf{k}_j(\boldsymbol{\alpha}_j, \mathbf{R}_j, \mathbb{W})}$ can be computed from $\boldsymbol{\alpha}_j$, $h^{\pi_1(\mathbf{Z})}$, and $\{g^{\pi_1(\mathbf{W}_i^\top \mathbf{Z})}\}_{i\in[d_1]}$ efficiently because $\mathbf{k}_j(\boldsymbol{\alpha}_j, \mathbf{R}_j, \mathbb{W}) = \{k_{j,\iota}(\boldsymbol{\alpha}_j, \mathbf{R}_j, \mathbb{W})\}_{\iota\in[d_2]}$ contains only linear combination of $\boldsymbol{\alpha}_j$, $\mathbf{Z}\begin{pmatrix} \mathbf{r}_i \\ 0 \\ 0 \end{pmatrix} = \pi_1(\mathbf{Z})\mathbf{r}_i$, and $\mathbf{W}_i^\top \mathbf{Z}\begin{pmatrix} \mathbf{r}_{j'} \\ 0 \\ 0 \end{pmatrix} = \pi_1(\mathbf{W}_i^\top \mathbf{Z})\mathbf{r}_{j'}$. Finally, it outputs private key

$$\mathsf{sk}_{\mathsf{ID}} = \left\{ \prod_{j\in S, \eta\in[d_2]} \mathsf{sk}_{j,\eta}^{E_{j,\eta,\iota}} \right\}_{\iota\in[d_3]}. \tag{10}$$

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}, \mathsf{M})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$. Then it runs $\mathsf{CEnc}(S, p) \to (\mathbf{c}, d_3')$, picks $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{d_3'} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, and sets $\mathbf{S} = \left(\begin{pmatrix} \mathbf{s}_0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{s}_1 \\ 0 \\ 0 \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{s}_{d_3'} \\ 0 \\ 0 \end{pmatrix} \right) \in \mathbb{Z}_p^{4\times(d_3'+1)}$. Then it returns

$$\mathsf{CT} = \left(C_1 = g^{\mathbf{c}(\mathbf{S},\mathbb{W})}, \quad C_2 = e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})\mathbf{s}_0} \cdot \mathsf{M} \right).$$

Note that $g^{\mathbf{c}(\mathbf{S},\mathbb{W})}$ can be computed from $g^{\pi_1(\mathbf{B})}$ and $\{g^{\pi_1(\mathbf{W}_i\mathbf{B})}\}_{i\in[d_1]}$ efficiently because $\mathbf{c}(\mathbf{S},\mathbb{W})$ contains only linear combinations of $\mathbf{B}\begin{pmatrix} \mathbf{s}_i \\ 0 \\ 0 \end{pmatrix} = \pi_1(\mathbf{B})\mathbf{s}_i$ and $\mathbf{W}_i\mathbf{B}\begin{pmatrix} \mathbf{s}_j \\ 0 \\ 0 \end{pmatrix} = \pi_1(\mathbf{W}_i\mathbf{B})\mathbf{s}_j$. C_2 can be computed from $e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})}$.

$\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{CT})$: Let CT be $\mathsf{CT} = (C_1, C_2)$. From $C_1 = g^{\mathbf{c}(\mathbf{S},\mathbb{W})} = \{g^{c_\iota(\mathbf{S},\mathbb{W})}\}_{\iota\in[d_3]}$, it computes

$$\prod_{\iota\in[d_3]} e\left(g^{c_\iota(\mathbf{S},\mathbb{W})}, \prod_{j\in S,\eta\in[d_2]} \mathsf{sk}_{j,\eta}^{E_{\eta,\iota}} \right) = e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})\mathbf{s}_0} \tag{11}$$

and recovers the message by $C_2/e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})\mathbf{s}_0} = \mathsf{M}$.

CORRECTNESS. We show the correctness of the scheme. It suffices to show Equation (11).

$$\begin{aligned}
\prod_{\iota\in[d_3]} e\left(g^{c_\iota(\mathbf{S},\mathbb{W})}, \prod_{j\in S,\eta\in[d_2]} \mathsf{sk}_{j,\eta}^{E_{\eta,\iota}} \right) &= \prod_{\iota\in[d_3],j\in S,\eta\in[d_2]} e(g,h)^{E_{\eta,\iota} c_\iota(\mathbf{S},\mathbb{W})^\top k_{j,\eta}(\boldsymbol{\alpha}_j,\mathbf{R}_j,\mathbb{W})} \\
&= \prod_{j\in S} e(g,h)^{\sum_{\iota\in[d_3],\eta\in[d_2]} E_{\eta,\iota} k_{j,\eta}(\boldsymbol{\alpha}_j,\mathbf{R}_j,\mathbb{W})^\top c_\iota(\mathbf{S},\mathbb{W})} \\
&= \prod_{j\in S} e(g,h)^{\boldsymbol{\alpha}_j^\top \mathbf{B} \begin{pmatrix} \mathbf{s}_0 \\ 0 \\ 0 \end{pmatrix}} = e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})\mathbf{s}_0}
\end{aligned}$$

The third equation above follows from the correctness of the underlying broadcast encoding.

Security. The following theorem indicates that the security of the IBE is (almost) tightly reduced to the MMH security of the underlying broadcast encoding on $(\mathbb{G}_1, \mathbb{G}_2)$ and Problem 7, 10, and 11. Combining the theorem with Theorem 3, 7, 8, 10, and 11, the security of the scheme can be almost tightly reduced to the Q_c-CMH security of the underlying encoding and the DLIN assumption. The reduction only incurs $O(\ell)$ security loss. The proof will appear in Appendix D.4.

Theorem 4. *For any adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, and $\mathcal{B}_4$ such that*

$$\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A},\Phi^{\mathsf{prime}},(\mu,Q_c,Q_k)}(\kappa) \le \mathsf{Adv}^{\mathsf{P}_7}_{\mathcal{B}_1}(\kappa) + \mathsf{Adv}^{\mathsf{P}_{11}}_{\mathcal{B}_2}(\kappa) + 2\ell \left(\mathsf{Adv}^{\mathsf{P}_{10}}_{\mathcal{B}_3}(\kappa) + \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B}_4,\Pi,(Q_c,Q_k),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) \right)$$

and $\max\{\mathsf{Time}(\mathcal{B}_i) | i \in [1,4]\} \approx \mathsf{Time}(\mathcal{A}) + (\mu + Q_c + Q_k) \cdot \mathsf{poly}(\kappa, \ell)$ *where* $\mathsf{poly}(\kappa, \ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

6 Construction of Broadcast Encoding Schemes

In this section, we show two broadcast encoding schemes Π_{cc} and Π_{slp}. As we will see, we can tightly prove the Q_c-CMH security for these schemes for any Q_c. Therefore, by applying the conversion in Section 4 and 5, we obtain IBE schemes with almost tight security in the multi-challenge and multi-instance setting

both in prime and composite-order groups. An IBE obtained from Π_{cc} achieves constant-size ciphertexts, but at the cost of requiring public parameters with the number of group elements being linear in the security parameter. Our second broadcast encoding scheme Π_{slp} partially compensate for this. By appropriately setting parameters, we can realize trade-off between size of ciphertexts and public parameters. For example, from the encoding, we obtain the first almost tightly secure IBE with all communication cost (the size of pp and CT) being $O(\sqrt{\kappa})$. Such a scheme is not known even in the single-challenge setting [18, 8]. While the structure of Π_{cc} is implicit in [27], Π_{slp} is new. The construction of Π_{slp} is inspired by recent works on unbounded attribute-based encryption schemes [41, 48, 49]. However, the security proof for the encoding is completely different.

6.1 Broadcast Encoding with Constant-Size Ciphertexts

At first, we show the following broadcast encoding scheme that we call Π_{cc}. The scheme has the same structure as the broadcast encryption scheme proposed by Gentry and Waters [27]. For Π_{cc}, we can prove Q-PMH security for any Q. By Lemma 1, we have that Q-CMH security of Π_{cc} on $\mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$ can be tightly proven unconditionally. Similar implication holds in prime-order groups. See Lemma 4.

$\mathsf{Param}(n, N) \to d_1$: It outputs $d_1 = n$.

$\mathsf{KEnc}(\tau, N) \to (\mathbf{k}, d_2')$: It outputs $\mathbf{k} = (\alpha + rw_\tau, rw_1, \ldots, rw_{\tau-1}, r, rw_{\tau+1}, \ldots, rw_n)$ and $d_2' = 1$ where $\mathbf{r} = r$.

$\mathsf{CEnc}(S, N) \to (\mathbf{c}, d_3')$: Let $S \subseteq [n]$. It outputs $\mathbf{c} = (s,\ \sum_{j\in S} sw_j)$ and $d_3' = 0$ where $\mathbf{s} = s$.

CORRECTNESS. Let $\tau \in S$. Then, we have

$$s \cdot \left((\alpha + rw_\tau) + \left(\sum_{j \in S\setminus\{\tau\}} rw_j \right) \right) - \left(\sum_{j\in S} sw_j \right) \cdot r = s\alpha. \tag{12}$$

Lemma 7. *Π_{cc} defined above is Q-PMH secure for any $Q \in \mathbb{N}$.*

Proof. Let $\tau \notin \cup_{j\in[Q]} S_j$. It is clear that information on w_τ is not leaked given $\{\mathbf{c}_{S_j}(\mathbf{s}_j, \mathbf{w})\}_{j\in[Q]}$. Thus, α is information-theoretically hidden from $\mathbf{k}_\tau(\alpha, \mathbf{r}, \mathbf{w})$, because α is masked by rw_τ which is uniformly random over $\mathbb{Z}_p$. Thus, the lemma follows. □

6.2 Encoding with Sub-linear Parameters

We propose the following broadcast encoding scheme that we call Π_{slp}. We can realize trade-off between sizes of parameters by setting n_1. For the encoding scheme, we are not able to show the Q-PMH security. Instead, we show the Q-CMH security.

$\mathsf{Param}(n, N) \to d_1$: It outputs $d_1 = 2n_1 + 3$. We let $n_2 = \lceil n/n_1 \rceil$. For ease of the notation, we will denote $\mathbf{w} = (u_1, \ldots, u_{n_1}, v, u_1', \ldots, u_{n_1}', v', w)$ in the following.

$\mathsf{KEnc}(\tau, N) \to (\mathbf{k}, d_2')$: It computes unique $\tau_1 \in [n_1]$ and $\tau_2 \in [n_2]$ such that $\tau = \tau_1 + (\tau_2 - 1) \cdot n_1$. Then it sets $d_2' = 1$ and $\mathbf{r} = r$ and outputs

$$\mathbf{k} = \left(\alpha + rw,\ r,\ r(v + \tau_2 u_{\tau_1}),\ \{ru_i\}_{i\in[n_1]\setminus\{\tau_1\}},\ r(v' + \tau_2 u_{\tau_1}'),\ \{ru_i'\}_{i\in[n_1]\setminus\{\tau_1\}} \right).$$

$\mathsf{CEnc}(S, N) \to (\mathbf{c}, d_3')$: It first defines $\tilde{S}_j$ and S_j for $j \in [n_2]$ as

$$\tilde{S}_j = S \cap [(j-1)n_1 + 1, jn_1], \quad S_j = \{j' - (j-1)n_1 \mid j' \in \tilde{S}_j\}, \tag{13}$$

sets $\mathbf{s} = (s_0, t_1, \ldots, t_{n_2}, t_1', \ldots, t_{n_2}')$ and $d_3' = 2n_2 + 1$, and outputs

$$\mathbf{c} = \left(s_0, \left\{ s_0 w + t_i \big(v + i \sum_{j \in S_i} u_j\big) + t_i' \big(v' + i \sum_{j \in S_i} u_j'\big), \quad t_i, \quad t_i' \right\}_{i \in [n_2]} \right)$$

CORRECTNESS. Let $\tau \in S$ and τ_1, τ_2 be defined as above. Then, we have $\tau_1 \in S_{\tau_2}$ and

$$\begin{aligned} & s_0 \cdot (\alpha + rw) - \left(s_0 w + t_{\tau_2} \left(v + \tau_2 \sum_{j \in S_{\tau_2}} u_j \right) + t'_{\tau_2} \left(v' + \tau_2 \sum_{j \in S_{\tau_2}} u'_j \right) \right) \cdot r \\ + \quad & t_{\tau_2} \cdot \left(r(v + \tau_2 u_{\tau_1}) + \tau_2 \cdot \left(\sum_{j \in S_{\tau_2} \setminus \{\tau_1\}} r u_j \right) \right) + t'_{\tau_2} \cdot \left(r(v' + \tau_2 u'_{\tau_1}) + \tau_2 \cdot \left(\sum_{j \in S_{\tau_2} \setminus \{\tau_1\}} r u'_j \right) \right) \\ = \quad & s_0 \alpha. \end{aligned}$$

In Appendix E, we prove the Q-CMH security of Π_{slp} on $(\mathbb{G}_1, \mathbb{G}_2)$ under the DLIN assumption. The Q-CMH security on $\mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$ also can be shown assuming the DLIN assumption on the corresponding group.

6.3 Implications

For Π_{xx}, we call an IBE scheme obtained by applying the conversion in Section 4 to Π_{xx} $\Phi_{\mathrm{xx}}^{\mathsf{comp}}$. Similarly, we call a scheme obtained by the conversion in Section 5.5 $\Phi_{\mathrm{xx}}^{\mathsf{prime}}$. The concrete descriptions of the obtained schemes appear in Appendix H. $\Phi_{\mathsf{cc}}^{\mathsf{prime}}$ and $\Phi_{\mathsf{slp}}^{\mathsf{prime}}$ are the first IBE schemes that are (almost) tightly secure in the multi-challenge and multi-instance setting, from a static assumption in prime-order groups (the DLIN assumption). $\Phi_{\mathsf{cc}}^{\mathsf{comp}}$ and $\Phi_{\mathsf{cc}}^{\mathsf{prime}}$ achieve constant-size ciphertext, meaning the number of group elements in ciphertexts is constant. The drawback of the schemes is their long public parameters. In $\Phi_{\mathsf{slp}}^{\mathsf{comp}}$ and $\Phi_{\mathsf{slp}}^{\mathsf{prime}}$, we can trade-off the size of ciphertexts and public parameters. For example, by setting $n_1 = \sqrt{n}$, we obtain the first almost tightly secure IBE scheme such that all communication cost (the size of the public parameters, the master public keys, and the ciphertexts) is sub-linear in the security parameter. Such a scheme is not known in the literature, even in the single-challenge and single-instance setting. Also see Table 1 in Section 1 for the overview of the obtained schemes.

7 Anonymous IBE with Tight Security Reduction.

All our IBE schemes obtained so far is not anonymous. In these schemes, one can efficiently check that a ciphertext is in a specific form using pairing computation, which leads to an attack against anonymity. In this section, we show that $\Phi_{\mathsf{cc}}^{\mathsf{prime}}$ can be modified to be anonymous, by removing all group elements in $\mathbb{G}_2$ from the public parameter pp and put these in sp instead. We call the resulting scheme Φ_{anon}. This is the first IBE scheme whose anonymity is (almost) tightly proven in the multi-challenge. While our technique for making the scheme anonymous is similar to that in [17], the security proof for our scheme requires some new ideas. This is because [17] only deals with the *single-challenge* setting whereas we prove tight security in the *multi-challenge* setting. The security proof requires new combination of information-theoretic argument (as in [17]) and computational argument.

Construction. Let the identity space of the scheme be $\{0,1\}^\ell$ and the message space be $\mathbb{G}_T$. We note that we have $\mathsf{sp} \neq \perp$ in the following, differently from other constructions in this paper.

$\mathsf{Par}(1^\kappa, \ell)$: It first runs $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{prime}}(1^\kappa)$. Then it picks $\mathbf{B} \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$, $\mathbf{W}_1, \ldots, \mathbf{W}_{2\ell} \xleftarrow{\$} \mathbb{Z}_p^{4\times 4}$ and a random full-rank diagonal matrix $\mathbf{D} \in \mathbb{Z}_p^{4\times 4}$ with the entries $(3,3)$ and $(4,4)$ being 1. Finally, it sets $\mathbf{Z} = \mathbf{B}^{-\top}\mathbf{D}$ and returns $\mathsf{pp} = (g, g^{\pi_1(\mathbf{B})}, g^{\pi_1(\mathbf{W}_1\mathbf{B})}, \ldots, g^{\pi_1(\mathbf{W}_{2\ell}\mathbf{B})})$ and $\mathsf{sp} = (h, h^{\pi_1(\mathbf{Z})}, h^{\pi_1(\mathbf{W}_1^\top\mathbf{Z})}, \ldots, g^{\pi_1(\mathbf{W}_{2\ell}^\top\mathbf{Z})})$.

$\mathsf{Gen}(\mathsf{pp}, \mathsf{sp})$: It picks $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{Z}_p^{4\times 1}$ and outputs $\mathsf{mpk} = (\mathsf{pp}, e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})})$ and $\mathsf{msk} = (\boldsymbol{\alpha}, \mathsf{sp})$.

$\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{ID})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ where $\mathsf{ID}_i \in \{0,1\}$ is the i-th bit of $\mathsf{ID} \in \{0,1\}^\ell$. Then it picks random $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and returns $\mathsf{sk}_{\mathsf{ID}} = (K_1 = h^{\boldsymbol{\alpha} + \sum_{i\in S} \pi_1(\mathbf{W}_i^\top \mathbf{Z})\mathbf{r}},\ K_2 = h^{-\pi_1(\mathbf{Z})\mathbf{r}})$.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}, \mathsf{M})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$. Then it picks random $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and returns $\mathsf{CT} = (C_1 = g^{\pi_1(\mathbf{B})\mathbf{s}},\ C_2 = g^{\sum_{i\in S} \pi_1(\mathbf{W}_i\mathbf{B})\mathbf{s}},\ C_3 = e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})\mathbf{s}} \cdot \mathsf{M})$.

$\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{CT})$: It parses the ciphertext CT as $\mathsf{CT} \to (C_1, C_2, C_3)$, and computes $e(C_1, K_1)e(C_2, K_2) = e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})\mathbf{s}}$. Then, it recovers the message by $C_3 / e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})\mathbf{s}} = \mathsf{M}$.

Remark. We have to ensure that the key extraction algorithm Ext always use the same randomness $\mathbf{r}$ for the same identity, in order to (tightly) prove the security of the scheme. This can be easily accomplished, for example, using PRF [26]. For the sake of simplicity, we do not incorporate this change into the description of our scheme. Instead, in the security proof, we assume that the adversary $\mathcal{A}$ makes at most single key extraction query for the same identity.

Security. The following theorem establishes the security of Φ_{anon} in the single-instance case. The proof appears in Appendix F. While we think that it is not hard to extend the result to the multi-instance setting, we do not treat it in this paper.

Theorem 5. *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\mathsf{AIBE}}_{\mathcal{A},\Phi_{\mathsf{anon}},(1,Q_c,Q_k)}(\kappa) \leq (8\ell+7)\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa) + (12\ell+1)/p$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k) \cdot \mathsf{poly}(\kappa, \ell)$ *where* $\mathsf{poly}(\kappa, \ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

8 Application to CCA Secure Public Key Encryption

Here, we discuss that our IBE schemes with almost tight security reduction in the multi-instance and multi-challenge setting yield almost tightly CCA secure PKE in the same setting via simple modification of Canetti-Halevi-Katz (CHK) transformation [16]. The difference from the ordinary CHK transformation is that we use (tightly secure) Q-fold one-time signature introduced and constructed in [32]. Another difference is that we need a restriction on the original IBE scheme. That is, we require that the key generation algorithm Gen of the IBE scheme does not output any secret parameter. Namely, $\mathsf{sp} = \bot$. Roughly speaking, this is needed since the syntax of the PKE does not allow key generation algorithm to take any secret parameter. Note that this condition is satisfied by all of our constructions except for that in Section 7. See Appendix G for the details about the conversion.

By applying the above conversion to $\Phi^{\mathsf{prime}}_{\mathsf{slp}}$ and $\Phi^{\mathsf{prime}}_{\mathsf{cc}}$, we obtain new PKE schemes that we call $\Psi^{\mathsf{prime}}_{\mathsf{slp}}$ and $\Psi^{\mathsf{prime}}_{\mathsf{cc}}$. The former allows flexible trade-off between the size of public parameters and ciphertexts. The latter achieves significantly short ciphertext-size: The ciphertext overhead of our scheme only consists of 10 group elements and 2 elements in $\mathbb{Z}_p$. This significantly improves previous results [32, 1, 35, 36, 29] on PKE scheme with the same security guarantee in terms of the ciphertext-size. Note that state-of-the-art construction by [36] and [29] require 47 and 59 group elements of ciphertext overhead, respectively. Namely, ciphertext

overhead of our scheme is (at least) 74% shorter, compared to theirs. On the other hand, the size of public parameter of the scheme in [29] is much shorter than ours (and those of [35, 36]). The former only requires 17 group elements, but the latter requires many more.

The reason why we can achieve very short ciphertext size is that our strategy to obtain PKE scheme is quite different from other works. Roughly speaking, all of the previous constructions [32, 1, 35, 36, 29] follow the template established by Hofheinz and Jager [32]. They first construct (almost) tightly-secure signature. Then, they use the signature to construct (almost) tightly-secure unbounded simulation sound (quasi-adaptive) NIZK. Finally, they follow the Naor-Yung paradigm [45] and convert the CPA-secure PKE with tight security reduction [11] into CCA-secure one using the NIZK. On the other hand, our construction is much more direct and simpler. Our conversion only requires very small amount of overhead in public parameters and ciphertexts.

References

[1] M. Abe, B. David, M. Kohlweiss, R. Nishimaki, M. Ohkubo. Tagged One-Time Signatures: Tight Security and Optimal Tag Size. In *PKC*, pp. 312–331, 2013.

[2] S. Agrawal and M. Chase. A study of Pair Encodings: Predicate Encryption in prime order groups. *IACR Cryptology ePrint Archive*, Report 2015/390.

[3] N. Attrapadung. Dual System Encryption Framework in Prime-Order Groups. *IACR Cryptology ePrint Archive*, Report 2015/390.

[4] N. Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In *EUROCRYPT*, pp. 557–577, 2014.

[5] N. Attrapadung, J. Furukawa, T. Gomi, G. Hanaoka, H. Imai, and R. Zhang. Efficient identity-based encryption with tight security reduction. In *CANS*, pp. 19–36, 2006.

[6] M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: security proofs and improvements. In *EUROCRYPT*, pp. 259-274, 2000.

[7] M. Bellare and T. Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for waters' IBE scheme. In *EUROCRYPT*, pp. 407–424, 2009.

[8] O. Blazy, E. Kiltz, and J. Pan. (hierarchical) identity-based encryption from affine message authentication. In *CRYPTO*, pp. 408–425, 2014.

[9] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pp. 223–238, 2004.

[10] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *CRYPTO*, pp. 443–459, 2004.

[11] D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In *CRYPTO*, pp. 41-55, 2004.

[12] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pp. 213–229, 2001.

[13] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, pp. 258–275, 2005.

[14] X. Boyen, B. Waters: Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In *CRYPTO*, pp. 290-307, 2006.

[15] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, pp. 255–271, 2003.

[16] R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *EUROCRYPT*, pp. 207–222, 2004.

[17] J. Chen, R. Gay, and H. Wee. Improved Dual System ABE in Prime-Order Groups via Predicate Encodings. *EUROCRYPT (2)* pp. 595–624, 2015.

[18] J. Chen and H. Wee. Fully, (almost) tightly secure IBE from standard assumptions. *IACR Cryptology ePrint Archive*, Report 2013/803.

[19] J. Chen and H. Wee. Dual system groups and its applications - compact HIBE and more. *IACR Cryptology ePrint Archive*, Report 2014/265.

[20] J. Chen and H. Wee. Fully, (Almost) Tightly Secure IBE and Dual System Groups. *CRYPTO*, pp. 435–460, 2013. A merge of two papers [18, 19].

[21] J. Chen and H. Wee. Semi-adaptive attribute-based encryption and improved delegation for boolean formula. In *SCN*, pp. 277–297, 2014.

[22] C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pp. 360–363, 2001.

[23] R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In *CRYPTO*, pp. 13-25, 1998.

[24] A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. L. Villar. An Algebraic Framework for Diffie-Hellman Assumptions. In *CRYPTO*, pp. 479–499, 2013.

[25] D. M. Freeman. Converting pairing-based Cryptosystems from Composite-Order Groups to Prime-Order Groups. In *EUROCRYPT*, pp. 44–61, 2010.

[26] C. Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pp. 445–464, 2006.

[27] C. Gentry and B. Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In *EUROCRYPT*, pp. 171–188, 2009.

[28] G. Herold, J. Hesse, D. Hofheinz, C. Rfols, A. Rupp. Polynomial spaces: A new framework for composite-to-prime-order transformations. In *CRYPTO*, pp. 261–279, 2014.

[29] Dennis Hofheinz. Algebraic partitioning: fully compact and (almost) tightly secure cryptography. *IACR Cryptology ePrint Archive*, Report 2015/499.

[30] Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO*, pp. 553–571, 2007.

[31] Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In *CRYPTO*, pp. 21–38, 2008.

[32] Dennis Hofheinz, Tibor Jager: Tightly Secure Signatures and Public-Key Encryption. In *CRYPTO*, pp. 590-607, 2012.

[33] D. Hofheinz, J. Koch, and C. Striecks. Identity-based encryption with (almost) tight security in the multi-instance, multi-ciphertext setting. In *PKC*, pp. 799–822, 2015.

[34] C. S. Jutla, A. Roy: Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces. In *ASIACRYPT (1)* pp. 1–20, 2013.

[35] B. Libert, M. Joye, M. Yung, and T. Peters. Concise Multi-challenge CCA-Secure Encryption and Signatures with Almost Tight Security. In *ASIACRYPT (2)* pp. 1-21, 2014.

[36] B. Libert, M. Joye, M. Yung, and T. Peters. Compactly Hiding Linear Spans: Tightly Secure Constant-Size Simulation-Sound QA-NIZK Proofs and Applications. *IACR Cryptology ePrint Archive*, Report 2015/242.

[37] A. B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT*, pp. 318–335, 2012.

[38] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, pp. 62–91, 2010.

[39] A. B. Lewko and B. Waters. Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In *ACM-CCS*, pp. 112–120, 2009.

[40] A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC*, pp. 455–479, 2010.

[41] A. B. Lewko and B. Waters. Unbounded HIBE and attribute-based encryption. In *EUROCRYPT*, pp. 547–567, 2011.

[42] A. B. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO*, pp. 180–198, 2012.

[43] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, Vol. 51, No. 2, pp. 231–262, 2004.

[44] M. Naor and G. Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pp. 18–35, 2009.

[45] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pp. 427-437, 1990.

[46] R. Nishimaki. How to Watermark Cryptographic Functions. In *EUROCRYPT*, pp. 111–125, 2013.

[47] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pp. 191–208, 2010.

[48] T. Okamoto and K. Takashima. Fully secure unbounded inner-product and attribute-based encryption. In *ASIACRYPT*, pp. 349–366, 2012.

[49] Y. Rouselakis and B. Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In *ACM-CCS*, pp. 463–474, 2013.

[50] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing over elliptic curve. In *The 2001 Symposium on Cryptography and Information Security*, 2001. (in Japanese).

[51] H. Shacham. A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants, *IACR Cryptology ePrint Archive*, Report 2007/074.

[52] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pp. 47–53, 1984.

[53] B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pp. 114–127, 2005.

[54] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO*, pp. 619–636, 2009.

[55] H. Wee. Dual system encryption via predicate encodings. In *TCC*, pp. 616–637, 2014.

A Proof of Lemma 1

Proof. It suffices to show the lemma for the case of $i = 2$. By the Q-PMH security, for all $\tau^\star$ and $S_1, \ldots, S_Q$ such that $\tau^\star \notin \cup_{j\in[Q]} S_j$, we have that two distributions $(\{\mathbf{c}_{S_j}(\mathbf{s}_j, \mathbf{w})\}_{j\in[Q]}, \mathbf{k}_{\tau^\star}(0, \mathbf{r}, \mathbf{w}))$ and $(\{\mathbf{c}_{S_j}(\mathbf{s}_j, \mathbf{w})\}_{j\in[Q]}, \mathbf{k}_{\tau^\star}(\alpha, \mathbf{r}, \mathbf{w}))$ are the same, where $(\mathbf{k}_{\tau^\star}, d_2') \leftarrow \mathsf{KEnc}(\tau, p_2)$, and $(\mathbf{c}_{S_j}, d_{3,j}') \leftarrow \mathsf{CEnc}(S_j, p_2)$ for $j \in [Q]$. By a standard argument (e.g., complexity leveraging [55]), this means that the two distributions are the same even if a distinguisher adaptively chooses each S_j depending on $\{\mathbf{c}_{S_{j'}}(\mathbf{s}_{j'}, \mathbf{w})\}_{j'\in[j-1]}$ and $\mathbf{k}_{\tau^\star}(b\alpha, \mathbf{r}, \mathbf{w})$ (where $b = 0$ or $b = 1$). By the correctness of the encoding (second condition), advantage of any adversary in Q-CMH security game would be 0, if $\mathbf{r}$ was chosen random subject to constraint that $(\mathbf{r} \mod p_2) \in (\mathbb{Z}_{p_2}^*)^{d_2'}$. In the actual Q-CMH security game, $\mathbf{r}$ is chosen as $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_N^{d_2'}$. This may alter the advantage at most d_2'/p_2. □

B Security Proof for Our Scheme in Section 4

For the sake of simplicity, we first consider the case of $\mu = 1$ (i.e., the single instance case). Namely, we first show the following theorem. Later in this section, we explain how to modify our proof for the single-instance case to deal with the multi-instance case.

Theorem 6. *For any adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, $\mathcal{B}_4$, and $\mathcal{B}_5$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A},\Phi,(1,Q_c,Q_k)}(\kappa) \le{}& \mathsf{Adv}^{\mathsf{P}_1}_{\mathcal{B}_1}(\kappa) + \mathsf{Adv}^{\mathsf{P}_5}_{\mathcal{B}_2}(\kappa) + Q_c \cdot 2^{-\Omega(\kappa)} \\ &+\ell\left(2\mathsf{Adv}^{\mathsf{P}_4}_{\mathcal{B}_3}(\kappa) + \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B}_4,\Pi,(Q_c,Q_k),\mathbb{G}_{p_2}}(\kappa) + \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B}_5,\Pi,(Q_c,Q_k),\mathbb{G}_{p_3}}(\kappa)\right)\end{aligned}$$

and $\max\{\mathsf{Time}(\mathcal{B}_i) | i \in [1,5]\} \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k) \cdot \mathsf{poly}(\kappa, \ell)$ *where* $\mathsf{poly}(\kappa, \ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. The overall structure of the proof is similar to [33].

Semi-functional Ciphertexts and Private Keys. We define several types of ciphertexts and private keys that are used in the security proof. In the following, we will pick random functions $\widehat{\mathsf{R}}_i : \{0,1\}^i \to \mathbb{Z}_N$ and $\widetilde{\mathsf{R}}_i : \{0,1\}^i \to \mathbb{Z}_N$ (via lazy sampling) for $i = 0, \ldots, \ell$. Here, we use $\{0,1\}^0$ to denote the singleton set containing just the empty string ϵ. For an identity $\mathsf{ID} \in \{0,1\}^\ell$, $\mathsf{ID}|_i$ denotes the first i bits of ID, that is, length i prefix of ID.

- **Semi-functional ciphertexts.** We consider Type 1, Type $(2,i)$, and Type $(3,i)$ of semi-functional ciphertexts for $i \in [0,\ell]$. The form of semi-functional ciphertext are as follows.

$$\mathsf{CT} = \begin{cases} \left(g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot \boxed{g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}},\ \mathsf{H}\left(e\big(g_1^{s_0} \cdot \boxed{g_2^{\hat{s}_0}}, h^{\alpha}\big) \right) \oplus \mathsf{M}_{\mathsf{coin}} \right) & \text{Type 1} \\ \left(g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot \boxed{g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}},\ \mathsf{H}\left(e\big(g_1^{s_0} \cdot \boxed{g_2^{\hat{s}_0}}, h^{\alpha} \cdot \boxed{g_2^{\widehat{\mathsf{R}}_i(\mathsf{ID}|_i)}}\big) \right) \oplus \mathsf{M}_{\mathsf{coin}} \right) & \text{Type } (2,i) \\ \left(g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot \boxed{g_3^{\mathbf{c}(\tilde{\mathbf{s}},\mathbf{w})}},\ \mathsf{H}\left(e\big(g_1^{s_0} \cdot \boxed{g_3^{\tilde{s}_0}}, h^{\alpha} \cdot \boxed{g_3^{\widetilde{\mathsf{R}}_i(\mathsf{ID}|_i)}}\big) \right) \oplus \mathsf{M}_{\mathsf{coin}} \right) & \text{Type } (3,i) \end{cases}$$

where $\hat{\mathbf{s}} = (\hat{s}_0, \hat{s}_1, \ldots \hat{s}_{d_3'})^\top \xleftarrow{\$} \mathbb{Z}_N^{d_3'+1}$ and $\tilde{\mathbf{s}} = (\tilde{s}_0, \tilde{s}_1, \ldots, \tilde{s}_{d_3'})^\top \xleftarrow{\$} \mathbb{Z}_N^{d_3'+1}$.

- **Random ciphertexts.** We consider Type 1 and Type 2 of random ciphertexts. The form of ciphertexts are as follows.

$$\mathsf{CT} = \begin{cases} \left(g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot \boxed{g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}},\ \mathsf{H}\left(e(g_1,h)^{s_0\alpha} \cdot \boxed{e(g_2,g_2)^{z}} \right) \oplus \mathsf{M}_{\mathsf{coin}} \right) & \text{Type 1} \\ \left(g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot \boxed{g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}},\ \mathsf{M}_{\mathsf{rand}} \right) & \text{Type 2} \end{cases}$$

where $z \xleftarrow{\$} \mathbb{Z}_N$ and $\mathsf{M}_{\mathsf{rand}} \xleftarrow{\$} \{0,1\}^m$.

- **Semi-functional Private Keys.** We consider Type $(1,i)$ for $i \in [0,\ell]$ and Type $(2,i)$ for $i \in [\ell-1]$ of semi-functional private keys. To create a semi-functional private key, we replace sk_j in Equation (6) with

$$\mathsf{sk}_j = h^{\mathbf{k}_j(\alpha_j,\mathbf{0},\mathbf{0})} \cdot g_1^{\mathbf{k}_j(0,\mathbf{r}_j,\mathbf{w})} \cdot \boxed{g_2^{\mathbf{k}_j(\hat{\gamma}_j,\mathbf{0},\mathbf{0})} \cdot g_3^{\mathbf{k}_j(\tilde{\gamma}_j,\mathbf{0},\mathbf{0})}} \cdot g_4^{\boldsymbol{\delta}_j} \tag{14}$$

where $\{\hat{\gamma}_j \in \mathbb{Z}_N\}_{j\in S}$ and $\{\tilde{\gamma}_j \in \mathbb{Z}_N\}_{j\in S}$ are random numbers subject to constraint that

$$\sum_{j\in S} \hat{\gamma}_j = \begin{cases} \widehat{\mathsf{R}}_i(\mathsf{ID}|_i) & \text{Type } (1,i) \\ \widehat{\mathsf{R}}_{i+1}(\mathsf{ID}|_{i+1}) & \text{Type } (2,i) \end{cases}, \qquad \sum_{j\in S} \tilde{\gamma}_j = \widetilde{\mathsf{R}}_i(\mathsf{ID}|_i) \quad \text{Type } (1,i) \text{ and } (2,i).$$

Then the semi-functional private key is created as $\mathsf{sk}_{\mathsf{ID}} = \prod_{j\in S} (\mathsf{sk}_j)^{\mathbf{E}_j}$.

Sequence of Games. Next, we define a sequence of games to establish the security of the IBE scheme. We write $\mathsf{Adv}_{\mathrm{xx}}(\kappa)$ to denote the advantage of $\mathcal{A}$ in $\mathsf{Game}_{\mathrm{xx}}$.

Game_0 : This is the real security game.

Game_1: In this game, all challenge ciphertexts created by the challenger are changed to be Type 1.

$\mathsf{Game}_{2,i,1}$ (for $i \in [1,\ell+1]$): In this game, all challenge ciphertexts are of Type $(2,i-1)$ whereas all private keys created by the challenger are Type $(1,i-1)$.

$\mathsf{Game}_{2,i,2}$ (for $i \in [1,\ell]$): This game is the same as $\mathsf{Game}_{2,i,1}$ except that challenge ciphertexts for identities ID such that $\mathsf{ID}_i = 0$ are changed to be Type $(3,i-1)$ where ID_i is the i-th bit of ID.

$\mathsf{Game}_{2,i,3}$ (for $i \in [1,\ell]$): This game is the same as $\mathsf{Game}_{2,i,2}$ except that challenge ciphertexts for identity ID such that $\mathsf{ID}_i = 1$ are changed to be Type $(2,i)$ and all private keys are changed to be Type $(2,i-1)$.

$\mathsf{Game}_{2,i,4}$ (for $i \in [1,\ell]$): This game is the same as $\mathsf{Game}_{2,i,3}$ except that challenge ciphertexts for identity ID such that $\mathsf{ID}_i = 0$ are changed to be Type $(3,i)$ and all private keys are changed to be Type $(1,i)$.

Game_3: This game is the same as $\mathsf{Game}_{2,\ell+1,1}$ except that all ciphertexts are changed to be random ciphertexts of Type 1.

Game_4: This game is the same as Game_3 except that all ciphertexts are changed to be random ciphertexts of Type 2.

Observe that we have $\mathsf{Adv}_4(\kappa) = 0$ since the view of $\mathcal{A}$ is independent from the value of coin in Game_4. We have that

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A},\Phi^{\mathsf{comp}},(1,Q_c,Q_k)}(\kappa) &= \Pr[\mathsf{Adv}_0] \\
&\leq |\Pr[\mathsf{Adv}_0] - \Pr[\mathsf{Adv}_1]| + |\Pr[\mathsf{Adv}_1] - \Pr[\mathsf{Adv}_{2,1,1}]| \\
&+ \sum_{i\in[\ell],j\in[1,3]} |\Pr[\mathsf{Adv}_{2,i,j}] - \Pr[\mathsf{Adv}_{2,i,j+1}]| + \sum_{i\in[\ell]} |\Pr[\mathsf{Adv}_{2,i,4}] - \Pr[\mathsf{Adv}_{2,i+1,1}]| \\
&+ |\Pr[\mathsf{Adv}_{2,\ell+1,1}] - \Pr[\mathsf{Adv}_3]| + |\Pr[\mathsf{Adv}_3] - \Pr[\mathsf{Adv}_4]|
\end{aligned}$$

Therefore, we complete the proof by showing lemma 8, 9, 10, 11, 12, 13, 15, and 16 in the following. □

Lemma 8. (Game_0 to Game_1). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_1$ such that* $|\mathsf{Adv}_0(\kappa) - \mathsf{Adv}_1(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_1}_{\mathcal{B}_1}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k) \cdot \mathsf{poly}(\kappa, \ell)$ *where* $\mathsf{poly}(\kappa, \ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We construct an adversary $\mathcal{B}_1$ who attacks Problem 1 from an adversary $\mathcal{A}$ who distinguishes the games. We note that these games only differ in the creation of challenge ciphertexts. $\mathcal{B}_1$ simulates the challenger for $\mathcal{A}$ as follows.

Setup. At the outset of the game, $\mathcal{B}_1$ is given the problem instance of the assumption (g_1, g_4, g, T) where either $T \xleftarrow{\$} \mathbb{G}_{p_1}$ or $T \xleftarrow{\$} \mathbb{G}_{p_1p_2}$. Then, it runs $\mathsf{Param}(2\ell, N) \to d_1$, picks $\mathbf{w} \xleftarrow{\$} \mathbb{Z}_N^{d_1}$, $\alpha \xleftarrow{\$} \mathbb{Z}_N$, $\mathsf{H} \xleftarrow{\$} \mathcal{H}$, sets $h := g$, and returns the public parameter $\mathsf{pp} = (g_1, g_1^{\mathbf{w}}, g_4, h, \mathsf{H})$ and the master public key $\mathsf{mpk} = (\mathsf{pp}, e(g_1, h)^\alpha)$ to $\mathcal{A}$. $\mathcal{B}_1$ also sets $\mathsf{msk} = \alpha$ and flips a coin $\mathsf{coin} \xleftarrow{\$} \{0, 1\}$.

Key Extraction Queries. When the adversary $\mathcal{A}$ submits $(\mathtt{Extraction}, 1, \mathsf{ID})$ to the challenger, $\mathcal{B}_1$ simply runs $\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{ID}) \to \mathsf{sk}_{\mathsf{ID}}$ and returns $\mathsf{sk}_{\mathsf{ID}}$ to $\mathcal{A}$.

Challenge Queries. When the adversary $\mathcal{A}$ submits $(\mathtt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$ to the challenger, $\mathcal{B}_1$ sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and runs $\mathsf{CEnc}(S, N) \to (\mathbf{c}, d_3')$. Then it picks $\mathbf{s}' = (s_0', s_1', \ldots, s_{d_3'}') \xleftarrow{\$} \mathbb{Z}_N^{d_3'+1}$ and returns $\mathsf{CT} = \left(T^{\mathbf{c}(\mathbf{s}',\mathbf{w})}, \mathsf{H}\big(e(T^{s_0'}, h^\alpha)\big) \oplus \mathsf{M}_{\mathsf{coin}}\right)$ to $\mathcal{A}$. We claim that it is properly distributed normal ciphertext or semi-functional ciphertext of Type 1.

Let T be $T = g_1^{t_1} \cdot g_2^{t_2}$. Then we have

$$\mathsf{CT} = \left(g_1^{\mathbf{c}(t_1\mathbf{s}',\mathbf{w})} \cdot g_2^{\mathbf{c}(t_2\mathbf{s}',\mathbf{w})}, \quad \mathsf{H}\big(e(g_1^{t_1 s_0'} \cdot g_2^{t_2 s_0'}, h^\alpha)\big) \oplus \mathsf{M}_{\mathsf{coin}}\right).$$

If $T \xleftarrow{\$} \mathbb{G}_{p_1}$, we have that $t_1 \xleftarrow{\$} \mathbb{Z}_N^*$ and $t_2 = 0$. In this case, $\mathcal{B}_1$ implicitly sets $\mathbf{s} = t_1\mathbf{s}' \mod p_1$. Since $t_1\mathbf{s}' \mod p_1$ is uniformly distributed over $\mathbb{Z}_{p_1}^{d_3'+1}$ and $e(g_1^{t_1 s_0'} \cdot g_2^{t_2 s_0'}, h^\alpha) = e(g_1, h)^{t_1 s_0' \alpha}$ holds, $\mathcal{B}_1$ has correctly simulated the challenge ciphertext in Game_0. On the other hand, if $T \xleftarrow{\$} \mathbb{G}_{p_1p_2}$, then we have $t_1, t_2 \xleftarrow{\$} \mathbb{Z}_N^*$. In this case, $\mathcal{B}_1$ implicitly sets $\mathbf{s} = t_1\mathbf{s}' \mod p_1$ and $\hat{\mathbf{s}} = t_2\mathbf{s}' \mod p_2$. Both of $(\mathbf{s} \mod p_1, \hat{\mathbf{s}} \mod p_2)$ in Game_1 and $(t_1\mathbf{s}' \mod p_1, t_2\mathbf{s}' \mod p_2)$ in the simulation are uniformly distributed over $\mathbb{Z}_{p_1} \times \mathbb{Z}_{p_2}$ due to the Chinese Remainder Theorem. Therefore, $\mathcal{B}_1$ correctly simulates Game_1.

Guess. When $\mathcal{A}$ outputs coin', $\mathcal{B}_1$ outputs 1 if $\mathsf{coin}' = \mathsf{coin}$ and 0 otherwise.

$\mathcal{B}_1$ has properly simulated Game_0 if $T \xleftarrow{\$} \mathbb{G}_{p_1}$ and Game_1 if $T \xleftarrow{\$} \mathbb{G}_{p_1p_2}$. Hence, we may conclude that $|\mathsf{Adv}_0(\kappa) - \mathsf{Adv}_1(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_1}_{\mathcal{B}_1}(\kappa)$. □

Lemma 9. (Game_1 to $\mathsf{Game}_{2,1,1}$). *For any adversary $\mathcal{A}$, we have* $\mathsf{Adv}_1(\kappa) = \mathsf{Adv}_{2,1,1}(\kappa)$.

Proof. This is purely a conceptual change and thus $\mathcal{A}$'s advantage is not altered. To see this, let us consider a modified version of Game_1 in which we first choose $\alpha', \hat{\gamma}, \tilde{\gamma} \xleftarrow{\$} \mathbb{Z}_N$ and then set (unique) $\alpha \in \mathbb{Z}_N$ such that $h^\alpha = h^{\alpha'} \cdot g_2^{\hat{\gamma}} \cdot g_3^{\tilde{\gamma}}$. Since α is still uniformly distributed over $\mathbb{Z}_N$ due to the Chinese Remainder Theorem, the view of the adversary in the modified game is the same as Game_1. Furthermore, we claim that the view of the adversary in the modified game is also the same as $\mathsf{Game}_{2,1,1}$. This can be seen by regarding α' in the modified game as α in $\mathsf{Game}_{2,1,1}$, $\hat{\gamma}$ as $\widehat{\mathsf{R}}_0(\mathsf{ID}|_0)$, and $\tilde{\gamma}$ as $\widetilde{\mathsf{R}}_0(\mathsf{ID}|_0)$. We check this. At first, it is easy to see that the distribution of the private keys in the modified game is the same as that of $\mathsf{Game}_{2,1,1}$. As for the master public key and the challenge ciphertexts, we have $e(g_1, h^\alpha) = e(g_1, h^{\alpha'} \cdot g_2^{\hat{\gamma}} \cdot g_3^{\tilde{\gamma}}) = e(g_1, h^{\alpha'})$ and $e(g_1^{s_0} \cdot g_2^{\hat{s}_0}, h^\alpha) = e(g_1^{s_0} \cdot g_2^{\hat{s}_0}, h^{\alpha'} \cdot g_2^{\hat{\gamma}} \cdot g_3^{\tilde{\gamma}}) = e(g_1^{s_0} \cdot g_2^{\hat{s}_0}, h^{\alpha'} \cdot g_2^{\hat{\gamma}})$. These indicate that the view of $\mathcal{A}$ in the modified game is also the same as $\mathsf{Game}_{2,1,1}$. Thus, the lemma follows. □

Lemma 10. ($\mathsf{Game}_{2,i^\star,1}$ to $\mathsf{Game}_{2,i^\star,2}$). *For any $i^\star \in [1, \ell]$ and adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_2$ such that* $|\mathsf{Adv}_{2,i^\star,1}(\kappa) - \mathsf{Adv}_{2,i^\star,2}(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_4}_{\mathcal{B}_2}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k) \cdot \mathsf{poly}(\kappa, \ell)$ *where* $\mathsf{poly}(\kappa, \ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We construct an adversary $\mathcal{B}_2$ who attacks Problem 4 from an adversary $\mathcal{A}$ who distinguishes the games. We note that these games only differ in the creation of challenge ciphertexts for identity ID such that $\mathsf{ID}_{i^\star} = 0$. $\mathcal{B}_2$ simulates the challenger for $\mathcal{A}$ as follows.

Setup. At the outset of the game, $\mathcal{B}_2$ is given the problem instance of the assumption $(g_1, g_4, g_{12}, g_{23}, g, T)$ where either $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_2}$ or $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_3}$. Then, it runs $\mathsf{Param}(2\ell, N) \to d_1$, picks $\mathbf{w} \xleftarrow{\$} \mathbb{Z}_N^{d_1}$, $\alpha \xleftarrow{\$} \mathbb{Z}_N$, $\mathsf{H} \xleftarrow{\$} \mathcal{H}$, sets $h := g$, and returns the public parameter $\mathsf{pp} = (g_1, g_1^{\mathbf{w}}, g_4, h, \mathsf{H})$ and the master public key $\mathsf{mpk} = (\mathsf{pp}, e(g_1, h)^\alpha)$ to $\mathcal{A}$. $\mathcal{B}_2$ also sets $\mathsf{msk} = \alpha$ and flips a coin $\mathsf{coin} \xleftarrow{\$} \{0, 1\}$.

Simulating Random Function. Throughout the game, $\mathcal{B}_2$ simulates a random function $\mathsf{R}_{i^\star-1}(\cdot) : \{0,1\}^{i^\star-1} \to \mathbb{Z}_N$ via lazy sampling. Let g_{23} be $g_{23} = g_2^{u_2} g_3^{u_3}$. $\mathcal{B}_2$ will implicitly sets

$$\begin{aligned} \widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) &= u_2 \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) \mod p_2 \quad \text{and} \\ \widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) &= u_3 \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) \mod p_3 \end{aligned}$$

in the following. By the Chinese Remainder Theorem, we have that $\left(u_2 \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) \mod p_2, u_3 \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) \mod p_3\right)$ in the simulation as well as $\left(\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) \mod p_2, \widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) \mod p_3\right)$ in $\mathsf{Game}_{2,i^\star,1}$ and $\mathsf{Game}_{2,i^\star,2}$ are uniformly distributed over $\mathbb{Z}_{p_2} \times \mathbb{Z}_{p_3}$ for all distinct $\mathsf{ID}|_{i^\star-1}$. Therefore, simulation of $\widehat{\mathsf{R}}_{i^\star-1}(\cdot)$ and $\widetilde{\mathsf{R}}_{i^\star-1}(\cdot)$ by $\mathcal{B}_2$ is perfect. (Note that the value of $\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1}) \mod p_1p_3p_4$ and $\widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1}) \mod p_1p_2p_4$ are not specified. This is not a problem, because these values are information-theoretically hidden from the view of $\mathcal{A}$ in both of $\mathsf{Game}_{2,i^\star,1}$ and $\mathsf{Game}_{2,i^\star,2}$.)

Key Extraction Queries. When the adversary $\mathcal{A}$ submits $(\texttt{Extraction}, 1, \mathsf{ID})$ to the challenger, $\mathcal{B}_2$ creates $\{\mathsf{sk}_j\}_{j \in S}$ as Equation (6). This is possible since $\mathcal{B}_2$ has $\mathsf{msk} = \alpha$. Then, $\mathcal{B}_2$ calls the random function $\mathsf{R}_{i^\star-1}(\cdot)$ on input $\mathsf{ID}|_{i^\star-1}$ to obtain $\gamma = \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) \in \mathbb{Z}_N$ and picks random $\{\gamma_j\}_{j \in S}$ subject to constraint that $\sum_{j \in S} \gamma_j = \gamma$. Then, $\mathcal{B}_2$ computes $\mathsf{sk}'_j = \mathsf{sk}_j \cdot g_{23}^{\mathbf{k}_j(\gamma_j, \mathbf{0}, \mathbf{0})}$ and returns the private key $\mathsf{sk}_{\mathsf{ID}} = \prod_{j \in S} (\mathsf{sk}'_j)^{\mathbf{E}_j}$ to $\mathcal{A}$. We claim that $\mathcal{B}_2$ correctly simulates semi-functional private key of Type $(1, i^\star - 1)$. We have that $\mathsf{sk}'_j = \mathsf{sk}_j \cdot g_2^{\mathbf{k}_j(u_2\gamma_j, \mathbf{0}, \mathbf{0})} \cdot g_3^{\mathbf{k}_j(u_3\gamma_j, \mathbf{0}, \mathbf{0})}$. Here, $\mathcal{B}_2$ implicitly sets $\hat{\gamma}_j = u_2\gamma_j$ and $\tilde{\gamma}_j = u_2\gamma_j$ for $j \in S$. Since $\{\hat{\gamma}_j \mod p_2\}_{j \in S}$ and $\{\tilde{\gamma}_j \mod p_3\}_{j \in S}$ are uniformly distributed subject to constraint that

$$\sum_{j \in S} \hat{\gamma}_j = \widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1}) \mod p_2 \quad \text{and} \quad \sum_{j \in S} \tilde{\gamma}_j = \widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1}) \mod p_3,$$

$\mathcal{B}_2$'s simulation is perfect.

Challenge Queries. When the adversary $\mathcal{A}$ submits $(\texttt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$ to the challenger, $\mathcal{B}_2$ sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and runs $\mathsf{CEnc}(S, N) \to (\mathbf{c}, d'_3)$. Then, it proceeds as follows. There are two cases.

– If $\mathsf{ID}_{i^\star} = 1$, $\mathcal{B}_2$ picks $\mathbf{s}' = (s'_0, s'_1, \ldots, s'_{d'_3}) \xleftarrow{\$} \mathbb{Z}_N^{d'_3+1}$ and returns

$$\mathsf{CT} = \Big(g_{12}^{\mathbf{c}(\mathbf{s}',\mathbf{w})},\ \mathsf{H}\big(e(g_{12}^{s'_0}, h^\alpha \cdot g_{23}^{\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})})\big) \oplus \mathsf{M}_{\mathsf{coin}} \Big)$$

to $\mathcal{A}$. We claim that this is properly distributed semi-functional ciphertext of Type $(2, i^\star - 1)$. Let g_{12} be $g_{12} = g_1^{t_1} g_2^{t_2}$. We have $g_{12}^{\mathbf{c}(\mathbf{s}',\mathbf{w})} = g_1^{\mathbf{c}(t_1\mathbf{s}',\mathbf{w})} \cdot g_2^{\mathbf{c}(t_2\mathbf{s}',\mathbf{w})}$ and

$$\begin{aligned} & e(g_{12}^{s'_0}, h^\alpha \cdot g_{23}^{\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})}) = e\big(g_1^{s'_0 t_1} \cdot g_2^{s'_0 t_2}, h^\alpha \cdot g_2^{u_2 \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})} \cdot g_3^{u_3 \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})}\big) \\ = \quad & e\big(g_1^{s'_0 t_1} \cdot g_2^{s'_0 t_2}, h^\alpha \cdot g_2^{u_2 \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})}\big) = e\big(g_1^{s'_0 t_1} \cdot g_2^{s'_0 t_2}, h^\alpha \cdot g_2^{\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})}\big). \end{aligned}$$

Here, $\mathcal{B}_2$ implicitly sets $\mathbf{s} = t_1\mathbf{s}' \mod p_1$ and $\hat{\mathbf{s}} = t_2\mathbf{s}' \mod p_2$. By the Chinese Remainder Theorem, one can see that it is properly distributed semi-functional ciphertext of Type $(2, i^\star - 1)$.

– If $\mathsf{ID}_{i^\star} = 0$, $\mathcal{B}_2$ picks $\mathbf{s}' = (s'_0, s'_1, \ldots, s'_{d'_3}) \xleftarrow{\$} \mathbb{Z}_N^{d'_3+1}$ and returns

$$\mathsf{CT} = \Big(T^{\mathbf{c}(\mathbf{s}',\mathbf{w})},\ \mathsf{H}\big(e(T^{s'_0}, h^\alpha \cdot g_{23}^{\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})})\big) \oplus \mathsf{M}_{\mathsf{coin}} \Big)$$

to $\mathcal{A}$. By a similar argument to the above case, it is not hard to see that the ciphertext is properly distributed semi-functional ciphertext of Type $(2, i^\star - 1)$ if $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_2}$ and semi-functional ciphertext of Type $(3, i^\star - 1)$ if $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_3}$.

Guess. When $\mathcal{A}$ outputs coin', $\mathcal{B}_2$ outputs 1 if $\mathsf{coin}' = \mathsf{coin}$ and 0 otherwise.

$\mathcal{B}_2$ has properly simulated $\mathsf{Game}_{2,i^\star,1}$ if $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_2}$ and $\mathsf{Game}_{2,i^\star,2}$ if $T \xleftarrow{\$} \mathbb{G}^*_{p_1p_3}$. Hence, we may conclude that $|\mathsf{Adv}_{2,i^\star,1}(\kappa) - \mathsf{Adv}_{2,i^\star,2}(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_4}_{\mathcal{B}_2}(\kappa)$.

□

Lemma 11. ($\mathsf{Game}_{2,i^\star,2}$ to $\mathsf{Game}_{2,i^\star,3}$). *For any $i^\star \in [1, \ell]$ and adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_3$ such that $|\mathsf{Adv}_{2,i^\star,2}(\kappa) - \mathsf{Adv}_{2,i^\star,3}(\kappa)| \leq \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B}_3,\Pi,(Q_c,Q_k),\mathbb{G}_{p_2}}(\kappa)$ and $\mathsf{Time}(\mathcal{B}_3) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k) \cdot \mathsf{poly}(\kappa, \ell)$ where $\mathsf{poly}(\kappa, \ell)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. We construct an adversary $\mathcal{B}_3$ who breaks the (Q_c, Q_k)-MMH security of the underlying broadcast encoding on $\mathbb{G}_{p_2}$ from an adversary $\mathcal{A}$ who distinguishes $\mathsf{Game}_{2,i^\star,2}$ and $\mathsf{Game}_{2,i^\star,3}$ for some $i^\star \in [1, \ell]$. We note that these games differ in the creation of ciphertexts for ID such that $\mathsf{ID}_{i^\star} = 1$ and all private keys. In this proof, we first describe $\mathcal{B}_3$ and then analyse the view of $\mathcal{A}$ in the simulation.

Setup. At the outset of the game, $\mathcal{B}_3$ submits $\tau^\star = 2i^\star$ as its target and is given $(g_1, g_1^{\mathbf{w}}, g_3^{\mathbf{w}}, g_{24}, g_3, g_4)$. It first picks $a \xleftarrow{\$} \mathbb{Z}_N^*$ and sets $h := (g_1 g_{24} g_3)^a$. It then picks $\mathsf{H} \xleftarrow{\$} \mathcal{H}$, $\alpha \xleftarrow{\$} \mathbb{Z}_N$ and returns the public parameter $\mathsf{pp} = (g_1, g_1^{\mathbf{w}}, g_4, h, \mathsf{H})$ and the master public key $\mathsf{mpk} = (\mathsf{pp}, e(g_1, h)^\alpha)$. $\mathcal{B}_3$ keeps α, g_3, $g_3^{\mathbf{w}}$, and g_{24} privately. $\mathcal{B}_3$ also flips a random coin $\mathsf{coin} \xleftarrow{\$} \{0, 1\}$.

Programming Random Functions. Throughout the game, $\mathcal{B}_3$ simulates a random functions $\mathsf{R}_{i^\star-1}(\cdot) : \{0,1\}^{i^\star-1} \to \mathbb{Z}_N$ and $\widetilde{\mathsf{R}}_{i^\star-1}(\cdot) : \{0,1\}^{i^\star-1} \to \mathbb{Z}_N$ via lazy sampling. $\mathcal{B}_3$ also maintains a list List of length $i^\star$ prefixes of identities for which key extraction query was made. The list is set as $\mathsf{List} = \emptyset$ at the beginning of the game.

Key Extraction Queries. When the adversary $\mathcal{A}$ submits $(\texttt{Extraction}, 1, \mathsf{ID})$ to the challenger, $\mathcal{B}_3$ first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and runs $\mathsf{KEnc}(j, N) \to (\mathbf{k}_j, d'_2)$ for all $j \in S$. It then calls random functions $\mathsf{R}_{i^\star-1}(\cdot)$ and $\widetilde{\mathsf{R}}_{i^\star-1}(\cdot)$ on input $\mathsf{ID}|_{i^\star-1}$ to obtain $\gamma = \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$ and $\tilde{\gamma} = \widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$. Then, it picks

$\mathbf{r}'_j \xleftarrow{\$} \mathbb{Z}_N^{d'_2}$, $\boldsymbol{\delta}'_j \xleftarrow{\$} \mathbb{Z}_N^{d_2}$ for all $j \in S$. It also picks random $\{\alpha_j \in \mathbb{Z}_N\}_{j\in S}$, $\{\gamma_j \in \mathbb{Z}_N\}_{j\in S}$, and $\{\tilde{\gamma}_j \in \mathbb{Z}_N\}_{j\in S}$ subject to constraint that $\sum_{j\in S}\alpha_j = \alpha$, $\sum_{j\in S}\gamma_j = \gamma$, and $\sum_{j\in S}\tilde{\gamma}_j = \tilde{\gamma}$. Next, $\mathcal{B}_3$ computes $\{\mathsf{sk}_j\}_{j\in S}$ as follows. There are three cases to consider:

– In case of $\mathsf{ID}_{i^\star} = 1$ (or, equivalently, if $\tau^\star \notin S$), it sets $\mathsf{sk}_j = h^{\mathbf{k}_j(\alpha_j,\mathbf{0},\mathbf{0})} \cdot g_1^{\mathbf{k}_j(0,\mathbf{r}'_j,\mathbf{w})} \cdot g_{24}^{\mathbf{k}_j(\gamma_j,\mathbf{0},\mathbf{0})} \cdot g_3^{\mathbf{k}_j(\tilde{\gamma}_j,\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}'_j}$ for all $j \in S$. Let g_{24} be $g_{24} = g_2^{u_2} g_4^{u_4}$. Then, we have that

$$\mathsf{sk}_j = h^{\mathbf{k}_j(\alpha_j,\mathbf{0},\mathbf{0})} \cdot g_1^{\mathbf{k}_j(0,\mathbf{r}'_j,\mathbf{w})} \cdot g_2^{\mathbf{k}_j(u_2\gamma_j,\mathbf{0},\mathbf{0})} \cdot g_3^{\mathbf{k}_j(\tilde{\gamma}_j,\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}'_j + \mathbf{k}_j(u_4\gamma_j,\mathbf{0},\mathbf{0})} \tag{15}$$

Here, $\mathcal{B}_3$ implicitly sets $\hat{\gamma}_j = u_2\gamma_j$ for $j \in S$. $\{\hat{\gamma}_j\}_{j\in S}$ are random subject to constraint that $\sum_{j\in S}\hat{\gamma}_j = u_2 \cdot \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$.

– In case of $\mathsf{ID}_{i^\star} = 0$ (or, equivalently, if $\tau^\star \in S$) and $\mathsf{ID}|_{i^\star} \notin \mathsf{List}$, $\mathcal{B}_3$ first calls $\mathcal{O}^{\mathsf{MMH},\mathsf{K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ to obtain $g_1^{\mathbf{k}_{\tau^\star}(0,\mathbf{r},\mathbf{w})} \cdot g_2^{\mathbf{k}_{\tau^\star}(b\cdot\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}}$ where $\mathbf{r}$, $\hat{\alpha}$, and $\boldsymbol{\delta}$ are randomness chosen by the oracle. Then, $\mathcal{B}_3$ sets sk_j as Equation (15) for $j \in S\backslash\{\tau^\star\}$ and

$$\begin{aligned}\mathsf{sk}_{\tau^\star} &= \left(g_1^{\mathbf{k}_{\tau^\star}(0,\mathbf{r},\mathbf{w})} \cdot g_2^{\mathbf{k}_{\tau^\star}(b\cdot\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}}\right) \cdot \left(h^{\mathbf{k}_{\tau^\star}(\alpha_{\tau^\star},\mathbf{0},\mathbf{0})} \cdot g_1^{\mathbf{k}_{\tau^\star}(0,\mathbf{r}'_{\tau^\star},\mathbf{w})} \cdot g_{24}^{\mathbf{k}_{\tau^\star}(\gamma_{\tau^\star},\mathbf{0},\mathbf{0})} \cdot g_3^{\mathbf{k}_{\tau^\star}(\tilde{\gamma}_{\tau^\star},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}'_{\tau^\star}}\right) \\ &= h^{\mathbf{k}_{\tau^\star}(\alpha_{\tau^\star},\mathbf{0},\mathbf{0})} \cdot g_1^{\mathbf{k}_{\tau^\star}(0,\mathbf{r}+\mathbf{r}'_{\tau^\star},\mathbf{w})} \cdot g_2^{\mathbf{k}_{\tau^\star}(b\cdot\hat{\alpha}+u_2\gamma_{\tau^\star},\mathbf{0},\mathbf{0})} \cdot g_3^{\mathbf{k}_{\tau^\star}(\tilde{\gamma}_{\tau^\star},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}+\boldsymbol{\delta}'_{\tau^\star}+\mathbf{k}_{\tau^\star}(u_4\gamma_{\tau^\star},\mathbf{0},\mathbf{0})}\end{aligned}$$

for $j = \tau^\star$. Here, $\mathcal{B}_3$ implicitly sets $\hat{\gamma}_j = u_2\gamma_j$ for $j \in S\backslash\{\tau^\star\}$ and $\hat{\gamma}_{\tau^\star} = u_2\gamma_{\tau^\star} + b\cdot\hat{\alpha}$. Therefore, $\{\hat{\gamma}_j\}_{j\in S}$ are random subject to constraint that $\sum_{j\in S}\hat{\gamma}_j = u_2 \cdot \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) + b\cdot\hat{\alpha}$. Finally, $\mathcal{B}_3$ updates the list as $\mathsf{List} \leftarrow \mathsf{List} \cup \{\mathsf{ID}|_{i^\star}\}$.

– In case of $\mathsf{ID}_{i^\star} = 0$ and $\mathsf{ID}|_{i^\star} \in \mathsf{List}$, $\mathcal{A}$ must have made a key query for ID' such that $\mathsf{ID}'|_{i^\star} = \mathsf{ID}|_{i^\star}$. Since $\mathsf{ID}'_{i^\star} = \mathsf{ID}_{i^\star} = 0$, $\mathcal{B}_3$ must have called $\mathcal{O}^{\mathsf{MMH},\mathsf{K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ to deal with the *first* such key extraction query made by $\mathcal{A}$. Let $g_1^{\mathbf{k}_{\tau^\star}(0,\mathbf{r},\mathbf{w})} \cdot g_2^{\mathbf{k}_{\tau^\star}(b\cdot\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}}$ be the answer to the oracle call. In this case, $\mathcal{B}_3$ does not call $\mathcal{O}^{\mathsf{MMH},\mathsf{K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ and computes $\{\mathsf{sk}_j\}_{j\in S}$ in the same way as the above case using $g_1^{\mathbf{k}_{\tau^\star}(0,\mathbf{r},\mathbf{w})} \cdot g_2^{\mathbf{k}_{\tau^\star}(b\cdot\hat{\alpha},\mathbf{0},\mathbf{0})} \cdot g_4^{\boldsymbol{\delta}}$. As the above case, $\{\hat{\gamma}_j\}_{j\in S}$ are random subject to constraint that $\sum_{j\in S}\hat{\gamma}_j = u_2 \cdot \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) + b\cdot\hat{\alpha}$.

Finally, $\mathcal{B}_3$ computes $\mathsf{Pair}(j, S, N) \to \mathbf{E}_j$ for all $j \in S$ and returns $\mathsf{sk}_{\mathsf{ID}} = \prod_{j\in S}(\mathsf{sk}_j)^{\mathbf{E}_j}$ to $\mathcal{A}$.

Challenge Queries. When the adversary $\mathcal{A}$ submits $(\mathtt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$ to the challenger, $\mathcal{B}_3$ sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and runs $\mathsf{CEnc}(S, N) \to (\mathbf{c}, d'_3)$. Then, $\mathcal{B}_3$ proceeds as follows. There are two cases.

– If $\mathsf{ID}_{i^\star} = 1$ (or, equivalently, if $\tau^\star \notin S$), it submits S to its oracle $\mathcal{O}^{\mathsf{MMH},\mathsf{C}}_{\tau^\star,\mathbf{w}}(\cdot)$ to obtain $g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}$. Then it returns

$$\mathsf{CT} = \left(g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}, \mathsf{H}\big(e(g_1^{s_0} \cdot g_2^{\hat{s}_0}, h^\alpha \cdot g_{24}^{\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})})\big) \oplus \mathsf{M}_{\mathsf{coin}}\right)$$

to $\mathcal{A}$. Recall that $g_1^{s_0} \cdot g_2^{\hat{s}_0}$ is the first coefficient of the vector $g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}$ (by the restriction we posed on the broadcast encoding scheme). We have that

$$\begin{aligned}e(g_1^{s_0} \cdot g_2^{\hat{s}_0}, h^\alpha \cdot g_{24}^{\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})}) &= e(g_1^{s_0} \cdot g_2^{\hat{s}_0}, h^\alpha \cdot g_2^{u_2\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})} \cdot g_4^{u_4\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})}) \\ &= e(g_1^{s_0} \cdot g_2^{\hat{s}_0}, h^\alpha \cdot g_2^{u_2\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})}).\end{aligned}$$

Then, the ciphertext returned to $\mathcal{A}$ is in the form of

$$\mathsf{CT} = \left(\ g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}, \quad \mathsf{H}\big(e(g_1^{s_0} \cdot g_2^{\hat{s}_0},\ h^\alpha \cdot g_2^{u_2\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})})\big) \oplus \mathsf{M}_{\mathsf{coin}}\ \right).$$

– If $\mathsf{ID}_{i^\star} = 0$ (or, equivalently, if $\tau^\star \in S$), it first computes $\mathsf{CEnc}(S, N) \to (\mathbf{c}, d_3')$, picks $\mathbf{s} = (s_0, s_1, \ldots, s_{d_3'})$, $\tilde{\mathbf{s}} = (\tilde{s}_0, \tilde{s}_1, \ldots, \tilde{s}_{d_3'}) \xleftarrow{\$} \mathbb{Z}_N^{d_3'+1}$ and computes $g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})}$ and $g_3^{\mathbf{c}(\tilde{\mathbf{s}},\mathbf{w})}$ from $g_1^{\mathbf{w}}$ and $g_3^{\mathbf{w}}$. Then it returns $\mathsf{CT} = (g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot g_3^{\mathbf{c}(\tilde{\mathbf{s}},\mathbf{w})}, \mathsf{H}\big(e(g_1^{s_0} \cdot g_3^{\tilde{s}_0},\ h^\alpha \cdot g_3^{\tilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})})\big) \oplus \mathsf{M}_{\mathsf{coin}})$ to $\mathcal{A}$.

Guess. When $\mathcal{A}$ outputs coin', $\mathcal{B}_2$ outputs 1 if $\mathsf{coin}' = \mathsf{coin}$ and 0 otherwise.

Analysis. We claim that $\mathcal{A}$'s view corresponds to that of $\mathsf{Game}_{2,i^\star,2}$ if $b = 0$ (i.e., $\mathcal{B}_3$ is equipped with oracle $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},0}(\cdot)$) and $\mathsf{Game}_{2,i^\star,3}$ if $b = 1$ ($\mathcal{B}_3$ is equipped with $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},1}(\cdot)$). In the case of $b = 0$, it is easily seen that $\mathcal{B}_3$ simulates $\mathsf{Game}_{2,i^\star,2}$ with $\widehat{\mathsf{R}}_{i^\star-1}(\cdot) : \{0,1\}^{i^\star-1} \to \mathbb{Z}_N$ such that $\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) = u_2 \cdot \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$. Since $\mathsf{R}_{i^\star-1}(\cdot)$ is a random function, $\widehat{\mathsf{R}}_{i^\star-1}(\cdot)$ is also a random function. Thus $\mathcal{B}_3$ correctly simulates $\mathsf{Game}_{2,i^\star,2}$. On the other hand, in the case of $b = 1$, one can see that $\mathcal{B}_3$ simulates $\mathsf{Game}_{2,i^\star,3}$ with $\widehat{\mathsf{R}}_{i^\star}(\cdot) : \{0,1\}^{i^\star} \to \mathbb{Z}_N$ such that

$$\widehat{\mathsf{R}}_{i^\star}(\mathsf{ID}|_{i^\star}) = \begin{cases} u_2 \cdot \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) & \text{if } \mathsf{ID}_{i^\star} = 1 \\ u_2 \cdot \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) + \hat{\alpha} & \text{if } \mathsf{ID}_{i^\star} = 0 \end{cases}$$

where $\hat{\alpha}$ *is freshly chosen for every distinct* $\mathsf{ID}|_{i^\star}$. Since $\mathsf{R}_{i^\star-1}(\cdot)$ is a random function, $\widehat{\mathsf{R}}_{i^\star}(\cdot)$ defined above is also a random function. Therefore, $\mathcal{B}_3$'s simulation is perfect. Hence, we may conclude that $|\mathsf{Adv}_{2,i^\star,2}(\kappa) - \mathsf{Adv}_{2,i^\star,3}(\kappa)| \leq \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B}_3,\Pi,(Q_c,Q_k),\mathbb{G}_{p_2}}(\kappa)$. □

Lemma 12. ($\mathsf{Game}_{2,i^\star,3}$ to $\mathsf{Game}_{2,i^\star,4}$). *For any* $i^\star \in [1, \ell]$ *and adversary* $\mathcal{A}$*, there exists an adversary* $\mathcal{B}_4$ *such that* $|\mathsf{Adv}_{2,i^\star,3}(\kappa) - \mathsf{Adv}_{2,i^\star,4}(\kappa)| \leq \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B}_4,\Pi,(Q_c,Q_k),\mathbb{G}_{p_3}}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}_4) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k) \cdot \mathsf{poly}(\kappa, \ell)$ *where* $\mathsf{poly}(\kappa, \ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We construct an adversary $\mathcal{B}_4$ that breaks (Q_c, Q_k)-MMH security of the encoding on $\mathbb{G}_{p_3}$ (instead of $\mathbb{G}_{p_2}$) from an adversary $\mathcal{A}$ who distinguishes the games. The lemma can be shown analogously to Lemma 11. We only highlight the main difference.

- $\mathcal{B}_4$ sets $\tau^\star = 2i^\star - 1$ instead of $\tau^\star = 2i^\star$.
- $\mathcal{B}_4$ simulates $\widehat{\mathsf{R}}_{i^\star}(\cdot)$ and $\mathsf{R}_{i^\star-1}(\cdot)$ throughout the game and use these functions to create challenge ciphertexts and private keys. It will simulate $\mathsf{Game}_{2,i^\star,3}$ with $\widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) = u_3 \cdot \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$ or $\mathsf{Game}_{2,i^\star,4}$ with

  $$\widetilde{\mathsf{R}}_{i^\star}(\mathsf{ID}|_{i^\star}) = \begin{cases} u_3 \cdot \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) & \text{if } \mathsf{ID}_{i^\star} = 0 \\ u_3 \cdot \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) + \tilde{\alpha} & \text{if } \mathsf{ID}_{i^\star} = 1 \end{cases}$$

  where $u_3 \in \mathbb{Z}_N^*$ is defined as a number satisfying $g_{34} = g_3^{u_3} \cdot g_4^{u_4}$ for some $u_4 \in \mathbb{Z}_N^*$ and $\tilde{\alpha}$ is the randomness chosen by $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},1}(\cdot)$. $\tilde{\alpha}$ is freshly chosen for every distinct $\mathsf{ID}|_{i^\star}$.
- $\mathcal{B}_4$ computes the challenge ciphertext by itself if $\mathsf{ID}_{i^\star} = 1$. Otherwise, it makes oracle call to $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{w}}(\cdot)$ and creates the ciphertext using the answer from the oracle.
- $\mathcal{B}_4$ maintains a List of length $i^\star$ prefixes of identities for which key extraction query was made. For key extraction query made by $\mathcal{A}$, $\mathcal{B}_4$ generates private key by itself if $\mathsf{ID}_{i^\star} = 0$. If $\mathsf{ID}_{i^\star} = 1$ and $\mathsf{ID}|_{i^\star} \notin \mathsf{List}$, it calls $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ and creates the private key using the answer. Otherwise, $\mathcal{A}$ must have queried private key for ID' such that $\mathsf{ID}'|_{i^\star} = \mathsf{ID}|_{i^\star}$. $\mathcal{B}_4$ must have called $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ to deal with the *first* such key extraction query made by $\mathcal{A}$. $\mathcal{B}_4$ creates the private key using the answer to the query.

□

Lemma 13. ($\mathsf{Game}_{2,i^\star,4}$ to $\mathsf{Game}_{2,i^\star+1,1}$). *For any $i^\star \in [1,\ell]$ and adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_5$ such that $|\mathsf{Adv}_{2,i^\star,4}(\kappa) - \mathsf{Adv}_{2,i^\star+1,1}(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_4}_{\mathcal{B}_5}(\kappa)$ and $\mathsf{Time}(\mathcal{B}_5) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k) \cdot \mathsf{poly}(\kappa,\ell)$ where $\mathsf{poly}(\kappa,\ell)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. The proof is the same as that of Lemma 10 except that we replace $\mathsf{R}_{i^\star-1}$, $\mathsf{R}'_{i^\star-1}$, $\widehat{\mathsf{R}}_{i^\star-1}$, and $\widetilde{\mathsf{R}}_{i^\star-1}$ with $\mathsf{R}_{i^\star}$, $\mathsf{R}'_{i^\star}$, $\widehat{\mathsf{R}}_{i^\star}$, and $\widetilde{\mathsf{R}}_{i^\star}$. □

The following lemma shows random self-reducibility of the Problem 5. The lemma is needed to prove Lemma 15 in the following.

Lemma 14. (Random Self-Reducibility of Problem 5 [43].) *There exists an efficient algorithm that on input $(g_2, g_2^x, g_2^y, g_2^z, e(g_2,g_2)^{xyz+\gamma})$ outputs $(g_2^{\hat{x}}, e(g_2,g_2)^{\hat{x}yz+\hat{\gamma}})$ where $\begin{cases} \hat{\gamma} \xleftarrow{\$} \mathbb{Z}_N & \text{if } \gamma \neq 0 \mod p_2 \\ \hat{\gamma} = 0 & \text{if } \gamma = 0 \mod p_2 \end{cases}$ and $\hat{x} \xleftarrow{\$} \mathbb{Z}_N$.*

Proof. The algorithm picks $a, b \xleftarrow{\$} \mathbb{Z}_N$ and implicitly sets $\hat{x} = ax + b$ and $\hat{\gamma} = a\gamma$. It is easy to see that $\hat{x} \mod p_2$ is uniformly random over $\mathbb{Z}_{p_2}$ and $\hat{\gamma} = 0 \mod p_2$ in the case of $\gamma = 0 \mod p_2$. It can also be seen that $(\hat{x}, \hat{\gamma}) \mod p_2$ is uniformly random over $\mathbb{Z}_{p_2}^2$ in the case of $\gamma \neq 0 \mod p_2$. Furthermore, the algorithm can efficiently compute $g_2^{\hat{x}} = (g_2^x)^a g_2^b$ and $e(g_2,g_2)^{\hat{x}yz+\hat{\gamma}} = \left(e(g_2,g_2)^{xyz+\gamma}\right)^a e(g_2^y, g_2^z)^b$. □

Lemma 15. ($\mathsf{Game}_{2,\ell+1,1}$ to Game_3). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_6$ such that $|\mathsf{Adv}_{2,\ell+1,1}(\kappa) - \mathsf{Adv}_3(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_5}_{\mathcal{B}_6}(\kappa)$ and $\mathsf{Time}(\mathcal{B}_6) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k) \cdot \mathsf{poly}(\kappa,\ell)$ where $\mathsf{poly}(\kappa,\ell)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. We construct an adversary $\mathcal{B}_6$ who attacks Problem 5 from an adversary $\mathcal{A}$ who distinguishes the games.

Generating Q_c tuples. At the outset of the game, $\mathcal{B}_6$ is given the problem instance $(g_1, g_2, g_3, g_4, g, g_2^x, g_2^y, g_2^z, e(g_2,g_2)^{xyz+\gamma})$ where either $\gamma = 0$ or $\gamma \xleftarrow{\$} \mathbb{Z}_N^*$. $\mathcal{B}_6$ proceeds as follows.

Setup. It runs $\mathsf{Param}(2\ell, N) \to d_1$, picks $\mathbf{w} \xleftarrow{\$} \mathbb{Z}_N^{d_1}$, $\alpha \xleftarrow{\$} \mathbb{Z}_N$, $\mathsf{H} \xleftarrow{\$} \mathcal{H}$, sets $h := g$, and returns the public parameter $\mathsf{pp} = (g_1, g_1^{\mathbf{w}}, g_4, h, \mathsf{H})$ and the master public key $\mathsf{mpk} = (\mathsf{pp}, e(g_1,h)^\alpha)$ to $\mathcal{A}$. $\mathcal{B}_6$ also sets $\mathsf{msk} = \alpha$ and flips a coin $\mathsf{coin} \xleftarrow{\$} \{0,1\}$.

Programming Random Functions. Throughout the game, $\mathcal{B}_6$ simulates random functions $\mathsf{R}_\ell(\cdot) : \{0,1\}^\ell \to \mathbb{Z}_N$ and $\widetilde{\mathsf{R}}_\ell(\cdot) : \{0,1\}^\ell \to \mathbb{Z}_N$ via lazy sampling. In the following, $\mathcal{B}_6$ will implicitly set $\widehat{\mathsf{R}}_\ell(\cdot) : \{0,1\}^\ell \to \mathbb{Z}_N$ as

$$\widehat{\mathsf{R}}_\ell(\mathsf{ID}) = \begin{cases} \mathsf{R}_\ell(\mathsf{ID}) & \text{if } (\texttt{Extraction}, 1, \mathsf{ID}) \text{ is queried} \\ \mathsf{R}_\ell(\mathsf{ID}) + yz & \text{if } (\texttt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}) \text{ is queried for some } \mathsf{M}_0, \mathsf{M}_1. \end{cases}$$

Since $(\texttt{Extraction}, 1, \mathsf{ID})$ and $(\texttt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$ are never queried for the same ID (by the restriction posed on the adversary), $\widehat{\mathsf{R}}_\ell(\cdot)$ above is well-defined. Furthermore, since $\mathsf{R}_\ell(\cdot)$ is a random function, $\widehat{\mathsf{R}}_\ell(\cdot)$ is also a random function.

Key Extraction Queries. When the adversary $\mathcal{A}$ submits $(\texttt{Extraction}, 1, \mathsf{ID})$ to the challenger, $\mathcal{B}_6$ proceeds as follows. $\mathcal{B}_6$ first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and picks $\mathbf{r}_j \xleftarrow{\$} \mathbb{Z}_N^{d_2'}$ for $j \in S$. Then, $\mathcal{B}_6$ calls $\mathsf{R}_\ell(\cdot)$ and $\widetilde{\mathsf{R}}_\ell(\cdot)$ to obtain $\hat{\gamma} = \widehat{\mathsf{R}}_\ell(\mathsf{ID}) = \mathsf{R}_\ell(\mathsf{ID})$ and $\tilde{\gamma} = \widetilde{\mathsf{R}}_\ell(\mathsf{ID})$. It then picks random $\{\alpha_j \in \mathbb{Z}_N\}_{j\in S}$, $\{\hat{\gamma}_j \in \mathbb{Z}_N\}_{j\in S}$, and $\{\tilde{\gamma}_j \in \mathbb{Z}_N\}_{j\in S}$ subject to the constraint that $\sum_{j\in S} \alpha_j = \alpha$, $\sum_{j\in S} \hat{\gamma}_j = \hat{\gamma}$, and $\sum_{j\in S} \tilde{\gamma}_j = \tilde{\gamma}$. Then, $\mathcal{B}_6$ creates $\{\mathsf{sk}_j\}_{j\in S}$ as Equation (14) using the values. Finally, $\mathcal{B}_6$ computes $\mathsf{Pair}(j, S, N) \to \mathbf{E}_j$ for all $j \in S$ and returns the private key $\mathsf{sk}_{\mathsf{ID}} = \prod_{j\in S} (\mathsf{sk}_j)^{\mathbf{E}_j}$ to $\mathcal{A}$.

Challenge Queries. When the adversary $\mathcal{A}$ submits a challenge query $(\mathtt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$ to the challenger, $\mathcal{B}_6$ proceeds as follows. $\mathcal{B}_6$ first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and computes $\mathsf{CEnc}(S, N) \to (\mathbf{c}, d'_3)$. Then, it picks $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_N^{d'_3+1}$, $\hat{s}_1, \ldots, \hat{s}_{d'_3} \xleftarrow{\$} \mathbb{Z}_N$. It also runs the algorithm in Lemma 14 on input $(g_2, g_2^x, g_2^y, g_2^z, e(g_2, g_2)^{xyz+\gamma})$ to obtain $(g_2^{\hat{x}}, e(g_2, g_2)^{\hat{x}yz+\hat{\gamma}})$ where $\hat{\gamma} \xleftarrow{\$} \mathbb{Z}_N$ if $\gamma \neq 0 \mod p_2$ and $\hat{\gamma} = 0$ if $\gamma = 0$. It then implicitly sets $\hat{s}_0 := \hat{x}$ and $\hat{\mathbf{s}} = (\hat{s}_0, \ldots, \hat{s}_{d'_3})$. Next, it computes $g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})}$ and $g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}$. The latter is efficiently computable because $\mathcal{B}_6$ knows $g_2^{\hat{\mathbf{s}}}$ and $\mathbf{w}$, and, $\mathbf{c}(\hat{\mathbf{s}}, \mathbf{w})$ contains only linear combinations of monomials $\hat{s}_i$, $\hat{s}_i w_j$. It also calls $\mathsf{R}_\ell(\cdot)$ on input ID and computes $T = e(g_1^{s_0}, h^\alpha) \cdot e(g_2^{\hat{x}}, h^\alpha \cdot g_2^{\mathsf{R}_\ell(\mathsf{ID})}) \cdot e(g_2, g_2)^{\hat{x}yz+\hat{\gamma}}$. Finally, it returns $\mathsf{CT} = \left(g_1^{\mathbf{c}(\mathbf{s},\mathbf{w})} \cdot g_2^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}, \mathsf{H}(T) \oplus \mathsf{M}_{\mathsf{coin}}\right)$ to $\mathcal{A}$. We have that

$$\begin{aligned} T &= e(g_1^{s_0}, h^\alpha) \cdot e(g_2^{\hat{x}}, h^\alpha \cdot g_2^{\mathsf{R}_\ell(\mathsf{ID})}) \cdot e(g_2, g_2)^{\hat{x}yz+\hat{\gamma}} \\ &= e(g_1^{s_0}, h^\alpha) \cdot e(g_2^{\hat{x}}, h^\alpha \cdot g_2^{\mathsf{R}_\ell(\mathsf{ID})+yz}) \cdot e(g_2, g_2)^{\hat{\gamma}} \\ &= e(g_1^{s_0} \cdot g_2^{\hat{s}_0}, h^\alpha \cdot g_2^{\widehat{\mathsf{R}}_\ell(\mathsf{ID})}) \cdot e(g_2, g_2)^{\hat{\gamma}}. \end{aligned}$$

Therefore, it can be seen that the challenge ciphertext is properly distributed semi-functional ciphertext of Type $(2, \ell)$ if $\gamma = 0$ and random ciphertext of Type 1 if $\gamma \neq 0 \mod p_2$.

Guess. When $\mathcal{A}$ outputs coin', $\mathcal{B}_2$ outputs 1 if $\mathsf{coin}' = \mathsf{coin}$ and 0 otherwise.

$\mathcal{B}_6$ has properly simulated $\mathsf{Game}_{2,\ell+1,1}$ if $\gamma = 0$ and Game_3 if $\gamma \xleftarrow{\$} \mathbb{Z}_N^*$. Hence, we may conclude that $|\mathsf{Adv}_{2,\ell+1,1}(\kappa) - \mathsf{Adv}_3(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_5}_{\mathcal{B}_1}(\kappa)$. □

Lemma 16. (Game_3 to Game_4). *For any adversary $\mathcal{A}$, we have* $|\mathsf{Adv}_3(\kappa) - \mathsf{Adv}_4(\kappa)| \leq Q_c \cdot 2^{-\Omega(\kappa)}$.

Proof. These games only differ in the creation of the challenge ciphertexts. For each challenge ciphertext, we claim that

$$(\mathsf{H}, \mathsf{H}(e(g_1, h)^{s_0\alpha} \cdot e(g_2, g_2)^z)) \qquad \text{and} \qquad (\mathsf{H}, \mathsf{M}_{\mathsf{rand}}) \tag{16}$$

are $2^{-\Omega(\kappa)}$ close even if we fix s_0 and α. This follows from the left-over hash lemma since $e(g_2, g_2)^z$ is distributed uniformly randomly (and independently from the view of $\mathcal{A}$) over the subgroup of $\mathbb{G}_T$ with order p_2 and has at least $\log p_2$ bit of min-entropy. By replacing $\mathsf{H}(e(g_1, h)^{s_0\alpha} \cdot e(g_2, g_2)^z)$ in each challenge ciphertext with a random string one by one, we have that $|\mathsf{Adv}_3(\kappa) - \mathsf{Adv}_4(\kappa)| \leq Q_c \cdot 2^{-\Omega(\kappa)}$. □

Extension to the Multi-Instance Case. So far, we have considered single instance case ($\mu = 1$). We explain how to extend our proof for the single-instance case to deal with multi-instance case ($\mu \gg 1$).

(*Proof Sketch of Theorem 2.*) The main difference from the single instance case is that we have to simulate different random functions for each index of the instance. Namely, we simulate random functions $\widehat{\mathsf{R}}_i^{(j)}(\cdot) : \{0,1\}^i \to \mathbb{Z}_N$ and $\widetilde{\mathsf{R}}_i^{(j)}(\cdot) : \{0,1\}^i \to \mathbb{Z}_N$ for $i \in [0, \ell]$ and each index $j \in [\mu]$. Then, we consider Game_0 to Game_4 exactly the same as the single instance case above except that semi-functional ciphertexts and semi-functional private keys for j-th instance are computed using $\widehat{\mathsf{R}}_i^{(j)}(\cdot)$ and $\widetilde{\mathsf{R}}_i^{(j)}(\cdot)$. We have to bound the difference of the advantage of $\mathcal{A}$ between the games as in the proof of Theorem 6. Only slightly subtle parts in the proof are to bound the difference of the advantage in $\mathsf{Game}_{2,i^\star,2}$ and $\mathsf{Game}_{2,i^\star,3}$, $\mathsf{Game}_{2,i^\star,3}$ and $\mathsf{Game}_{2,i^\star,4}$, and Game_3 and Game_4.

We first bound the difference between $\mathsf{Game}_{2,i^\star,2}$ and $\mathsf{Game}_{2,i^\star,3}$. To do so, we assume an adversary $\mathcal{A}$ that distinguishes these games and construct an adversary $\mathcal{B}$ against the Q_c-CMH security of the underlying broadcast encoding on $\mathbb{G}_{p_2}$. We sketch the reduction.

- $\mathcal{B}$ selects $\tau^\star = 2i^\star$ as its target.

- Simulation of the challenge ciphertexts is analogous to Lemma 11. Namely, for a query $(\texttt{Challenge}, j, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$ from $\mathcal{A}$, it creates the challenge ciphertext by itself if $\mathsf{ID}_{i^\star} = 0$. Otherwise, it makes an oracle call to $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{w}}(\cdot)$ and creates the ciphertext using the answer from the oracle.
- To answer key extraction queries from $\mathcal{A}$, $\mathcal{B}$ maintains a list List of $(j, \mathsf{ID}|_{i^\star})$ such that $\mathcal{A}$ has made a query of the form $(\texttt{Extraction}, j, \mathsf{ID})$ throughout the game. When $\mathcal{A}$ makes a query $(\texttt{Extraction}, j, \mathsf{ID})$, $\mathcal{B}$ proceeds as follows. If $\mathsf{ID}_{i^\star} = 1$, it creates the private key by itself. If $\mathsf{ID}_{i^\star} = 0$ and $(j, \mathsf{ID}|_{i^\star}) \notin \mathsf{List}$, it makes a query to $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ and creates the private key using the answer. Otherwise, $\mathcal{A}$ must have made a query of the form $(\texttt{Extraction}, j, \mathsf{ID}')$ such that $\mathsf{ID}'|_{i^\star} = \mathsf{ID}|_{i^\star}$. $\mathcal{B}_4$ must have called $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{w},b}(\cdot)$ to deal with the *first* such key extraction query made by $\mathcal{A}$. $\mathcal{B}_4$ creates the private key using the answer to the query.

This completes the description of the reduction.

The difference of the advantage of $\mathcal{A}$ in $\mathsf{Game}_{2,i^\star,3}$ and $\mathsf{Game}_{2,i^\star,4}$ can be bounded similarly to the above. The difference of the advantage in $\mathsf{Game}_{2,\ell+1,1}$ and Game_3 can be bounded similarly to Lemma 15. Here, we simulate $\widetilde{\mathsf{R}}^{(j)}_\ell(\cdot) : \{0,1\}^\ell \to \mathbb{Z}_N$ for $j \in [\mu]$ as

$$\widehat{\mathsf{R}}^{(j)}_\ell(\mathsf{ID}) = \begin{cases} \mathsf{R}^{(j)}_\ell(\mathsf{ID}) & \text{if } (\texttt{Extraction}, j, \mathsf{ID}) \text{ is queried} \\ \mathsf{R}^{(j)}_\ell(\mathsf{ID}) + yz & \text{if } (\texttt{Challenge}, j, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}) \text{ is queried for some } \mathsf{M}_0, \mathsf{M}_1 \end{cases}$$

where $\mathsf{R}^{(j)}_\ell(\cdot) : \{0,1\}^\ell \to \mathbb{Z}_N$ is a random function whose value is known to the simulator.

C Random Self-Rducibility of the Intermediate Problems and Reductions from the DLIN Assumption

C.1 Reductions from DLIN

Following theorems show that the hardness of Problem 7, 8, 9, 10, 11, and 12 can be tightly reduced to the DLIN assumption.

Theorem 7. *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that* $\mathsf{Adv}^{\mathsf{P}_7}_{\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + \mathsf{poly}(\kappa)$ *where* $\mathsf{poly}(\kappa)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. Here, $\mathbf{B}$ and $\mathbf{Z}$ are 4-by-4 matrices. A similar lemma is already shown for 3-by-3 matrices in [18]. It is straightforward to extend the result, but we include the proof for completeness. Given the problem instance $(g, h, g^{a_1}, g^{a_2}, g^{a_3}, g^{a_1 s_1}, g^{a_2 s_2}, T = g^{a_3(s_1+s_2)+\gamma})$ where $\gamma = 0$ or $\gamma \xleftarrow{\$} \mathbb{Z}^*_p$, $\mathcal{B}$ proceeds as follows. ($\mathcal{B}$ disregards other terms in the problem instance.)

Programming B, Z. $\mathcal{B}$ picks $\tilde{z}_1, \tilde{z}_2 \xleftarrow{\$} \mathbb{Z}^*_p$ and defines $\mathbf{A}$, $\mathbf{A}^*$, and $\mathbf{D}$ as

$$\mathbf{A} = \begin{pmatrix} a_1 & & & \\ & a_2 & & \\ a_3 & a_3 & 1 & \\ & & & 1 \end{pmatrix} \in \mathbb{Z}^{4\times 4}_p \quad , \quad \mathbf{A}^* = \begin{pmatrix} a_1^{-1} & & -a_1^{-1}a_3 & \\ & a_2^{-1} & -a_2^{-1}a_3 & \\ & & 1 & \\ & & & 1 \end{pmatrix} \in \mathbb{Z}^{4\times 4}_p,$$

$$\text{and} \quad \mathbf{D} = \begin{pmatrix} a_1\tilde{z}_1 & & & \\ & a_2\tilde{z}_2 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}. \tag{17}$$

It is easy to check that $\mathbf{A}^* = \mathbf{A}^{-\top}$. Next, $\mathcal{B}$ samples $\mathbf{B}' \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$ and implicitly sets $\mathbf{B} = \mathbf{B}'\mathbf{A}$. Then, we have $\mathbf{Z} = \mathbf{B}^{-\top}\mathbf{D} = \mathbf{B}'^{-\top}\mathbf{A}^{-\top}\mathbf{D} = \mathbf{B}'^{-\top}(\mathbf{A}^*\mathbf{D})$. It is clear that $\mathbf{B}$ and $\mathbf{Z}$ are properly distributed.

Programming $\mathbf{t}$ and $\hat{t}$. $\mathcal{B}$ implicitly sets $\mathbf{t} = (s_1, s_2)^\top$ and $\hat{t} = \gamma$.

Simulating Input to $\mathcal{A}$. In the following, we check that $\mathcal{B}$ can efficiently compute all input to $\mathcal{A}$. At first, $\mathcal{B}$ can efficiently compute $g^{\mathbf{B}} = g^{\mathbf{B}'\mathbf{A}}$ since it knows $g^{\mathbf{A}}$ and $\mathbf{B}'$. It can also compute $h^{\pi_1(\mathbf{Z})} = h^{\mathbf{B}'^{-\top}\cdot\pi_1(\mathbf{A}^*\mathbf{D})}$ since it knows all entries of $\mathbf{B}'^{-\top}$ and $h^{\pi_1(\mathbf{A}^*\mathbf{D})}$. Furthermore, it can efficiently compute $g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ 0\end{pmatrix}} = g^{\mathbf{B}'\mathbf{A}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ 0\end{pmatrix}}$ from $\mathbf{B}'$ and

$$g^{\mathbf{A}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ 0\end{pmatrix}} = \begin{pmatrix} g^{a_1 s_1} \\ g^{a_2 s_2} \\ T \\ 1_{\mathbb{G}_1} \end{pmatrix}.$$

Therefore, $\mathcal{B}$ can simulate all input to $\mathcal{A}$.

Output. Finally, $\mathcal{B}$ inputs all terms computed above to $\mathcal{A}$ and outputs the same bit as $\mathcal{A}$.

As we can see, $\mathcal{B}$ properly simulates the problem instance of Problem 7 to $\mathcal{A}$. Therefore, we have that $\mathsf{Adv}^{\mathsf{P}_7}_{\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa)$. □

Theorem 8. *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathsf{Adv}^{\mathsf{P}_8}_{\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa)$ and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + \mathsf{poly}(\kappa)$ where $\mathsf{poly}(\kappa)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. Here, $\mathbf{B}$ and $\mathbf{Z}$ are 4-by-4 matrices. A similar lemma is already shown for 3-by-3 matrices in [18]. It is straightforward to extend the result, but we include the proof for completeness. Given the problem instance $(g, h, h^{a_1}, h^{a_2}, h^{a_3}, h^{a_1 s_1}, h^{a_2 s_2}, T = h^{a_3(s_1+s_2)+\gamma})$ where $\gamma = 0$ or $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$, it proceeds as follows. ($\mathcal{B}$ disregards other terms in the problem instance.)

Programming $\mathbf{B}$, $\mathbf{Z}$. $\mathcal{B}$ picks $\tilde{z}_1, \tilde{z}_2 \xleftarrow{\$} \mathbb{Z}_p^*$ and defines $\mathbf{A}$, $\mathbf{A}^*$, and $\mathbf{D}$ as Equation (17). Next, $\mathcal{B}$ samples $\mathbf{B}' \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$ and implicitly sets $\mathbf{B} = \mathbf{B}'\mathbf{A}^*\mathbf{D}$. We have

$$\mathbf{Z} = \mathbf{B}^{-\top}\mathbf{D} = \mathbf{B}'^{-\top}(\mathbf{A}^*)^{-\top}\mathbf{D}^{-\top}\mathbf{D} = \mathbf{B}'^{-\top}\mathbf{A}.$$

It can be seen that $\mathbf{B}$ and $\mathbf{Z}$ are properly distributed.

Programming $\mathbf{u}$ and $\hat{u}$. $\mathcal{B}$ implicitly sets $\mathbf{u} = (s_1, s_2)^\top$ and $\hat{u} = \gamma$.

Programming $\mathbf{t}$ and $\hat{t}$. $\mathcal{B}$ picks random $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_p^{3\times 1}$ and implicitly sets $\begin{pmatrix}\mathbf{t}\\ \hat{t}\end{pmatrix} = \mathbf{E}^{-1}\mathbf{v}$, where $\mathbf{E} \in \mathbb{Z}_p^{3\times 3}$ is the upper left 3×3 submatrix of $\mathbf{A}^*\mathbf{D}$.

Simulating Input to $\mathcal{A}$. In the following, we check that $\mathcal{B}$ can efficiently compute all input to $\mathcal{A}$. At first, $\mathcal{B}$ can efficiently compute $h^{\mathbf{Z}}$ since it knows $h^{\mathbf{A}}$ and $\mathbf{B}'$. It can also compute $g^{\pi_1(\mathbf{B})} = h^{\mathbf{B}'\cdot\pi_1(\mathbf{A}^*\mathbf{D})}$ and $g^{\pi_3(\mathbf{B})} = h^{\mathbf{B}'\cdot\pi_3(\mathbf{A}^*\mathbf{D})}$ since it knows all entries of $\mathbf{B}'$, $h^{\pi_1(\mathbf{A}^*\mathbf{D})}$, and $h^{\pi_3(\mathbf{A}^*\mathbf{D})}$. It can compute $g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ 0\end{pmatrix}} = g^{\mathbf{B}'\mathbf{A}^*\mathbf{D}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ 0\end{pmatrix}} = g^{\mathbf{B}'\begin{pmatrix}\mathbf{v}\\ 0\end{pmatrix}}$ because it knows all entries of $\mathbf{B}'$ and $\mathbf{v}$. Furthermore, we have $h^{\mathbf{A}\begin{pmatrix}\mathbf{u}\\ \hat{u}\\ 0\end{pmatrix}} = \begin{pmatrix} h^{a_1 s_1} \\ h^{a_2 s_2} \\ T \\ 1_{\mathbb{G}_2} \end{pmatrix}$.

Thus $\mathcal{B}$ can efficiently compute all entries of $h^{\mathbf{Z}\begin{pmatrix}\mathbf{u}\\ \hat{u}\\ 0\end{pmatrix}} = h^{\mathbf{B}'^{-\top}\mathbf{A}\begin{pmatrix}\mathbf{u}\\ \hat{u}\\ 0\end{pmatrix}}$ from $\mathbf{B}'$ and $h^{\mathbf{A}\begin{pmatrix}\mathbf{u}\\ \hat{u}\\ 0\end{pmatrix}}$. Therefore, $\mathcal{B}$ can simulate all input to $\mathcal{A}$.

Output. Finally, $\mathcal{B}$ inputs all terms computed above to $\mathcal{A}$ and outputs the same bit as $\mathcal{A}$.

As we can see, $\mathcal{B}$ properly simulates the problem instance of Problem 8 to $\mathcal{A}$. Therefore, we have that $\mathsf{Adv}^{\mathsf{P}_8}_{\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa)$. □

Theorem 9. *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that* $\mathsf{Adv}^{\mathsf{P}_9}_{\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa) + 1/p$ *and* $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + \mathsf{poly}(\kappa)$ *where* $\mathsf{poly}(\kappa)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. At first, $\mathcal{B}$ is given the problem instance of the DLIN assumption $(g, g^{a_1}, g^{a_2}, g^{a_3}, g^{a_1 s_1}, g^{a_2 s_2}, h, h^{a_1}, h^{a_2}, h^{a_3}, T = g^{a_3(s_1+s_2)+\gamma})$ where either $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$ or $\gamma = 0$ and proceeds as follows. ($\mathcal{B}$ disregards other terms in the problem instance.)

Programming $\mathbf{t}$, $\hat{t}$, and $\tilde{t}$. $\mathcal{B}$ picks $\xi, \zeta \xleftarrow{\$} \mathbb{Z}_p$ and implicitly sets $\mathbf{t} = (s_1, s_2)^\top$, $\hat{t} = \xi - a_3 s_1$, and $\tilde{t} = \gamma$. We have that $\hat{t} \neq 0$ with probability at least $1 - 1/p$. Conditioned on the event, $(\mathbf{t}, \hat{t}, \tilde{t})$ is distributed as $(\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}, \hat{t} \xleftarrow{\$} \mathbb{Z}_p^*, \tilde{t} = 0)$ if $\gamma = 0$ and $(\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}, \hat{t} \xleftarrow{\$} \mathbb{Z}_p^*, \tilde{t} \xleftarrow{\$} \mathbb{Z}_p^*)$ if $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$.

Programming $\mathbf{B}$, $\mathbf{Z}$. $\mathcal{B}$ defines $\mathbf{A}$ and $\mathbf{A}^*$ as

$$\mathbf{A} = \begin{pmatrix} a_1 & & & \\ & a_2 & & \\ a_3 & & 1 & \\ a_3 & a_3 & & 1 \end{pmatrix} \in \mathbb{Z}_p^{4\times 4} \quad \text{and} \quad \mathbf{A}^* = \begin{pmatrix} a_1^{-1} & & -a_1^{-1}a_3 & -a_1^{-1}a_3 \\ & a_2^{-1} & & -a_2^{-1}a_3 \\ & & 1 & \\ & & & 1 \end{pmatrix} \in \mathbb{Z}_p^{4\times 4}.$$

It is easy to check that $\mathbf{A}^* = \mathbf{A}^{-\top}$. It also picks $\tilde{z}_1, \tilde{z}_2 \xleftarrow{\$} \mathbb{Z}_p^*$ and implicitly defines

$$\mathbf{D} = \begin{pmatrix} a_1\tilde{z}_1 & & & \\ & a_2\tilde{z}_2 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \in \mathbb{Z}_p^{4\times 4}.$$

It follows immediately that

$$g^{\mathbf{A}\begin{pmatrix} \mathbf{t} \\ \hat{t} \\ \tilde{t} \end{pmatrix}} = \begin{pmatrix} g^{a_1 s_1} \\ g^{a_2 s_2} \\ g^{\xi} \\ T \end{pmatrix} \in \mathbb{Z}_p^4, \qquad \mathbf{A}^*\mathbf{D} = \begin{pmatrix} \tilde{z}_1 & & -a_1^{-1}a_3 & -a_1^{-1}a_3 \\ & \tilde{z}_2 & & -a_2^{-1}a_3 \\ & & 1 & \\ & & & 1 \end{pmatrix} \in \mathbb{Z}_p^{4\times 4},$$

$$\mathbf{A}^*\mathbf{D}\begin{pmatrix} 0 \\ 0 \\ a_1 \\ 0 \end{pmatrix} = \begin{pmatrix} -a_3 \\ 0 \\ a_1 \\ 0 \end{pmatrix}, \quad \text{and} \quad \mathbf{A}^*\mathbf{D}\begin{pmatrix} 0 \\ 0 \\ -a_2 \\ a_2 \end{pmatrix} = \begin{pmatrix} 0 \\ -a_3 \\ -a_2 \\ a_2 \end{pmatrix}.$$

Next, $\mathcal{B}$ samples $\mathbf{B}' \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$ and implicitly sets

$$\mathbf{B} = \mathbf{B}'\mathbf{A}, \quad \text{and} \quad \mathbf{Z} = \mathbf{B}^{-\top}\mathbf{D} = \mathbf{B}'^{-\top}\mathbf{A}^{-\top}\mathbf{D} = \mathbf{B}'^{-\top}(\mathbf{A}^*\mathbf{D}).$$

It is clear that $\mathbf{B}$ and $\mathbf{Z}$ are properly distributed.

Simulating Input to $\mathcal{A}$. In the following, we check that $\mathcal{B}$ can efficiently compute all input to $\mathcal{A}$. At first, $\mathcal{B}$ can efficiently compute $g^{\mathbf{B}}$ since it knows $g^{\mathbf{A}}$ and $\mathbf{B}'$. It can also compute $h^{\pi_1(\mathbf{Z})} = h^{\mathbf{B}'^{-\top} \cdot \pi_1(\mathbf{A}^*\mathbf{D})}$ since it knows all entries of $\mathbf{B}'^{-\top}$ and $h^{\pi_1(\mathbf{A}^*\mathbf{D})}$. Furthermore, $\mathcal{B}$ can compute $g^{\mathbf{B}\begin{pmatrix} \mathbf{t} \\ \hat{t} \\ \tilde{t} \end{pmatrix}}$ since it knows $g^{\mathbf{A}\begin{pmatrix} \mathbf{t} \\ \hat{t} \\ \tilde{t} \end{pmatrix}}$ and $\mathbf{B}'$. It can also compute $h^{a_1\hat{\mathbf{f}}} = h^{\mathbf{B}'^{-\top}(a_1\mathbf{A}^*\mathbf{D}\mathbf{e}_3^\top)}$ and $h^{-a_2\hat{\mathbf{f}}+a_2\tilde{\mathbf{f}}} = h^{\mathbf{B}'^{-\top}(\mathbf{A}^*\mathbf{D}(-a_2\mathbf{e}_3^\top + a_2\mathbf{e}_4^\top))}$ since it knows all entries of $h^{a_1\mathbf{A}^*\mathbf{D}\mathbf{e}_3^\top}$, $h^{(\mathbf{A}^*\mathbf{D}(-a_2\mathbf{e}_3^\top + a_2\mathbf{e}_4^\top))}$, and $\mathbf{B}'^{-\top}$. $\mathcal{B}$ then picks $\begin{pmatrix} \phi_{1,1} & \phi_{1,2} \\ \phi_{2,1} & \phi_{2,2} \end{pmatrix} \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$ and implicitly sets $\mathbf{\Theta} = \begin{pmatrix} \theta_{1,1} & \theta_{1,2} \\ \theta_{2,1} & \theta_{2,2} \end{pmatrix} = \begin{pmatrix} a_1 & -a_2 \\ 0 & a_2 \end{pmatrix} \cdot \begin{pmatrix} \phi_{1,1} & \phi_{1,2} \\ \phi_{2,1} & \phi_{2,2} \end{pmatrix}$. It is easy to see that $\mathbf{\Theta}$ is properly distributed and $\mathcal{B}$ can efficiently compute $h^{\theta_{1,j}\hat{\mathbf{f}} + \theta_{2,j}\tilde{\mathbf{f}}} = \left(h^{a_1\hat{\mathbf{f}}}\right)^{\phi_{1,j}} \cdot \left(h^{-a_2\hat{\mathbf{f}}+a_2\tilde{\mathbf{f}}}\right)^{\phi_{2,j}}$ for $j \in \{1, 2\}$.

Output. Finally, $\mathcal{B}$ inputs all terms computed above to $\mathcal{A}$ and outputs the same bit as $\mathcal{A}$.

As we have seen, $\mathcal{B}$ properly simulates the problem instance of Problem 9 to $\mathcal{A}$, except for probability $1/p$. Therefore, we have that $\mathsf{Adv}^{\mathsf{P}_9}_{\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa) + 1/p$. □

Theorem 10. *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathsf{Adv}^{\mathsf{P}_{10}}_{\mathcal{A}}(\kappa) \leq 2\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa) + 2/p$ and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + \mathsf{poly}(\kappa)$ where $\mathsf{poly}(\kappa)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. The theorem can be shown by applying Theorem 9 twice. We first replace the challenge term $T_0 = g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ 0\end{pmatrix}}$ in the problem instance of Problem 10 with $g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ \tilde{t}\end{pmatrix}}$ where $\mathbf{t} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\hat{t}, \tilde{t} \xleftarrow{\$} \mathbb{Z}_p^*$. By Theorem 9, there exists an adversary $\mathcal{B}$ such that $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + \mathsf{poly}(\kappa)$ and the change of the probability that the adversary $\mathcal{A}$ outputs 1 is bounded by $\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa) + 1/p$. Then, we further replace the challenge term with $T_1 = g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ 0\\ \tilde{t}\end{pmatrix}}$. By the symmetry of the third and the fourth column of $\mathbf{B}$, we can apply Theorem 9 again. The change of the probability that $\mathcal{A}$ outputs 1 is bounded by $\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa) + 1/p$. □

The following theorems are already shown by previous works [14, 44, 46]. Note that these theorems are shown in symmetric pairing groups or groups even without pairing, which are different situations than ours. However, their proofs trivially work also in our setting.

Theorem 11. (Theorem 2.13 in [46]. See also [14].) *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathsf{Adv}^{\mathsf{P}_{11}}_{\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa)$ and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + \mathsf{poly}(\kappa)$ where $\mathsf{poly}(\kappa)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Theorem 12. (Lemma A.1 in [44].) *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathsf{Adv}^{\mathsf{P}_{12}}_{\mathcal{A}}(\kappa) \leq 4\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa)$ and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + \mathsf{poly}(\kappa)$ where $\mathsf{poly}(\kappa)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

C.2 Random Self-Reducibility of the Intermediate Problems

Here, we show that Problem 7, 8, 9, 10, and 11 have random self-reducibility. The following theorem is a slight extension of Lemma 1 in [24].

Lemma 17. (Random Self-Reducibility of Problem 7, 8, 9, and 10.) *There exists an efficient algorithm that on input $(g^{\pi_1(\mathbf{B})}, g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ \tilde{t}\end{pmatrix}})$ outputs $g^{\mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ \tilde{s}\end{pmatrix}}$ where*

$$\begin{cases} \mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1},\ \hat{s} \xleftarrow{\$} \mathbb{Z}_p,\ \tilde{s} = 0 & \text{if } (\hat{t} \neq 0,\ \tilde{t} = 0) \\ \mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1},\ \hat{s} = 0,\ \tilde{s} \xleftarrow{\$} \mathbb{Z}_p & \text{if } (\hat{t} = 0,\ \tilde{t} \neq 0) \\ \mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1},\ \hat{s} = 0,\ \tilde{s} = 0 & \text{if } (\hat{t} = 0,\ \tilde{t} = 0). \end{cases}$$

Similarly, there exists an efficient algorithm that on input $(g^{\pi_1(\mathbf{B})}, g^{\pi_2(\mathbf{B})}, g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ \tilde{t}\end{pmatrix}})$ outputs $g^{\mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ \tilde{s}\end{pmatrix}}$ where

$$\begin{cases} \mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1},\ \hat{s} \xleftarrow{\$} \mathbb{Z}_p,\ \tilde{s} = 0 & \text{if } \tilde{t} = 0 \\ \mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1},\ \hat{s} \xleftarrow{\$} \mathbb{Z}_p,\ \tilde{s} \xleftarrow{\$} \mathbb{Z}_p & \text{if } \tilde{t} \neq 0. \end{cases}$$

Proof. At first, we show the former part of the lemma. The algorithm proceeds as follows. It picks $a \xleftarrow{\$} \mathbb{Z}_p$ and $\mathbf{s}' \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and implicitly sets $\mathbf{s} = \mathbf{s}' + a \cdot \mathbf{t}$, $\hat{s} = a \cdot \hat{t}$, and $\tilde{s} = a \cdot \tilde{t}$. It can be seen that $\mathbf{s}$ is uniformly distributed over $\mathbb{Z}_p^{2\times 1}$ and the value of a is information-theoretically hidden only given $\mathbf{s}$. It is clear that

$(\hat{s} \xleftarrow{\$} \mathbb{Z}_p, \tilde{s} = 0)$ if $(\hat{t} \neq 0, \tilde{t} = 0)$, $(\hat{s} = 0, \tilde{s} \xleftarrow{\$} \mathbb{Z}_p)$ if $(\hat{t} = 0, \tilde{t} \neq 0)$, and $(\hat{s} = 0, \tilde{s} = 0)$ if $(\hat{t} = 0, \tilde{t} = 0)$. The algorithm can compute $g^{\mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ \tilde{s}\end{pmatrix}}$ as

$$g^{\mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ \tilde{s}\end{pmatrix}} = \left(g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ \tilde{t}\end{pmatrix}}\right)^{a} \cdot g^{\mathbf{B}\begin{pmatrix}\mathbf{s}'\\ 0\\ 0\end{pmatrix}} = \left(g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ \tilde{t}\end{pmatrix}}\right)^{a} \cdot g^{\pi_1(\mathbf{B})\mathbf{s}'}.$$

We then show the latter part of the lemma. The algorithm picks $\hat{s}', a \xleftarrow{\$} \mathbb{Z}_p$, $\mathbf{s}' \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and implicitly sets $\mathbf{s} = \mathbf{s}' + a \cdot \mathbf{t}$, $\hat{s} = a \cdot \hat{t} + \hat{s}'$, and $\tilde{s} = a \cdot \tilde{t}$. It can be seen that $\mathbf{s}$ and $\hat{s}$ are uniformly random over $\mathbb{Z}_p^{2\times 1}$ and $\mathbb{Z}_p$ respectively. Furthermore, it can be seen that $\tilde{s} = 0$ if $\tilde{t} = 0$ and $\tilde{s} \xleftarrow{\$} \mathbb{Z}_p$ if $\tilde{t} \neq 0$. The algorithm can compute $g^{\mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ \tilde{s}\end{pmatrix}}$ as

$$g^{\mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ \tilde{s}\end{pmatrix}} = \left(g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ \tilde{t}\end{pmatrix}}\right)^{a} \cdot g^{\mathbf{B}\begin{pmatrix}\mathbf{s}'\\ 0\\ 0\end{pmatrix}} \cdot g^{\mathbf{B}\begin{pmatrix}\mathbf{0}\\ \hat{s}'\\ 0\end{pmatrix}} = \left(g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ \tilde{t}\end{pmatrix}}\right)^{a} \cdot g^{\pi_1(\mathbf{B})\mathbf{s}'} \cdot (g^{\pi_2(\mathbf{B})})^{\hat{s}'}.$$

□

Lemma 18. (Random Self-Reducibility of Problem 11 [43].) *There exists an efficient algorithm that on input* $(g, g^x, g^y, h^z, e(g,h)^{xyz+\gamma})$ *outputs* $(g^{\hat{x}}, e(g,h)^{\hat{x}yz+\hat{\gamma}})$ *where*

$$\begin{cases} \hat{x} \xleftarrow{\$} \mathbb{Z}_p,\ \hat{\gamma} \xleftarrow{\$} \mathbb{Z}_p & \text{if } \gamma \neq 0 \\ \hat{x} \xleftarrow{\$} \mathbb{Z}_p,\ \hat{\gamma} = 0 & \text{if } \gamma = 0. \end{cases}$$

Proof. The lemma can be shown analogously to Lemma 14 and thus we omit here. □

D Omitted Proofs from Section 5

D.1 Proof of Lemma 4

Proof. By the Q-PMH security, for all $\tau^\star$ and $S_1, \ldots, S_Q$ such that $\tau^\star \in \cup_{j\in[Q]} S_j$, we have that two distributions in Equation (4) are the same. By a standard argument (e.g., complexity leveraging [55]), this means that the two distributions are the same even if a distinguisher adaptively chooses each S_j depending on $\{\mathbf{c}_{S_{j'}}(\mathbf{s}_{j'}, \mathbf{w})\}_{j'\in[j-1]}$ and $\mathbf{k}_\tau(b\alpha, \mathbf{r}, \mathbf{w})$ (where $b = 0$ or $b = 1$). Therefore, advantage of any adversary in Q-CMH security game would be 0, if $\mathbf{r}$ was chosen as $\mathbf{r} \xleftarrow{\$} (\mathbb{Z}_p^*)^{d_2'}$ as in the Q-PMH-security game. In the actual Q-CMH security game, $\mathbf{r}$ is chosen as $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^{d_2'}$. This may alter the advantage at most d_2'/p. □

D.2 Proof of Lemma 5

Proof. (sketch.) All monomial terms appear in $\sum_{(\eta,\iota)\in[d_2]\times[d_3]} E_{\eta,\iota} k_\eta c_\iota$ are

$$\begin{gathered} \alpha s_{j'}, \qquad r_j \cdot s_{j'}, \qquad \alpha \cdot (w_{k'} s_{j'}) \\ r_j \cdot (s_{j'} w_k) = (w_k r_j) \cdot s_{j'} \end{gathered}$$

for $j \in [d_2']$, $k \in [d_1]$, $j' \in [0, d_3']$, $k' \in [d_1]$. Coefficients of all terms except for $s_0\alpha$ summed up to 0, because of the correctness of the encoding. We note that the terms of the form $w_{k'} w_k s_{j'} r_j$ do not

appear because of the regularity of the encoding (the first condition). If we expand $\sum_{(\eta,\iota)\in[d_2]\times[d_3]} E_{\eta,\iota} \cdot k_{\mathbf{Z},\eta}(\boldsymbol{\alpha},\mathbf{X},\mathbb{W})^\top c_{\mathbf{B},\iota}(\mathbf{Y},\mathbb{W})$, terms corresponding to above monomials appear:

$$\boldsymbol{\alpha}^\top \mathbf{B}\mathbf{y}_{j'}, \qquad (\mathbf{Z}\mathbf{x}_j)^\top \mathbf{B}\mathbf{y}_{j'}, \qquad \boldsymbol{\alpha}^\top \cdot \mathbf{W}_{k'}\mathbf{B}\mathbf{y}_{j'}$$

$$(\mathbf{Z}\mathbf{x}_j)^\top \cdot \mathbf{W}_k\mathbf{B}\mathbf{y}_{j'} = \mathbf{x}_j^\top \mathbf{Z}^\top \mathbf{W}_k\mathbf{B}\mathbf{y}_{j'} = \left(\mathbf{W}_k^\top \mathbf{Z}\mathbf{x}_j\right)^\top \cdot \mathbf{B}\mathbf{y}_{j'}.$$

The coefficients of all terms other than $\boldsymbol{\alpha}^\top \mathbf{B}\mathbf{y}_0$ are summed up to 0 because of the correctness of the encoding. □

D.3 Proof of Theorem 3

Proof. Here, we prove Theorem 3. We introduce Lemma 19 and 20. The former indicates that (Q_c, Q_k)-MMH security can be tightly reduced to $(Q_c, 1)$-MMH security. The latter shows $(Q_c, 1)$-MMH security can be tightly reduced to the hardness of the Problem 8 and Q_c-CMH security. Combining these lemmas, the theorem follows. □

Lemma 19. *For any broadcast encoding Π and adversary $\mathcal{A}$, there exists another adversary $\mathcal{B}$ such that*

$$\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,Q_k),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) \leq \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B},\Pi,(Q_c,1),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) + \frac{1}{p}$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + Q_k \cdot \mathsf{poly}(\kappa, n)$ *where* $\mathsf{poly}(\kappa, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We construct $\mathcal{B}$ against $(Q_c, 1)$-MMH security of the broadcast encoding from $\mathcal{A}$ against (Q_c, Q_k)-MMH security of the same encoding. At the beginning of the game, $\mathcal{A}(1^\kappa, n)$ submits its target index $\tau^\star \in [n]$. $\mathcal{B}$ then submits the same index and is given the parameter $\mathsf{params} = \big(g, g^{\pi_1(\mathbf{B})}, g^{\pi_3(\mathbf{B})}, \{g^{\pi_1(\mathbf{W}_i\mathbf{B})}\}_{i\in[d_1]}, \{g^{\pi_3(\mathbf{W}_i\mathbf{B})}\}_{i\in[d_1]}, h, h^{\mathbf{Z}}, \{h^{\pi_1(\mathbf{W}_i^\top\mathbf{Z})}\}_{i\in[d_1]}, \{h^{\pi_3(\mathbf{W}_i^\top\mathbf{Z})}\}_{i\in[d_1]}\big)$. Then, $\mathcal{B}$ simply pass it to $\mathcal{A}$. $\mathcal{B}$ also calls $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ to obtain $h^{\mathbf{k}_\mathbf{Z}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\mathbf{R},\mathbb{W})}$. In the following, we assume that $\hat{\alpha} \neq 0$. This happens with probability at least $1 - 1/p$. When $\mathcal{A}$ calls $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot)$ on input S, $\mathcal{B}$ calls the same oracle on the same input and passes what is given from the oracle to $\mathcal{A}$. When $\mathcal{A}$ calls $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$, $\mathcal{B}$ picks (freshly random) $a \xleftarrow{\$} \mathbb{Z}_p$, $\mathbf{r}'_1, \ldots, \mathbf{r}'_{d'_2} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and sets $\mathbf{R}' = \left(\begin{pmatrix}\mathbf{r}'_1\\0\\0\end{pmatrix}, \cdots, \begin{pmatrix}\mathbf{r}'_{d'_2}\\0\\0\end{pmatrix}\right) \in \mathbb{Z}_p^{4\times d'_2}$. Then it computes

$$\left(h^{\mathbf{k}_\mathbf{Z}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\mathbf{R},\mathbb{W})}\right)^a \cdot h^{\mathbf{k}_\mathbf{Z}(\mathbf{0},\mathbf{R}',\mathbb{W})} = h^{\mathbf{k}_\mathbf{Z}(b\cdot a\hat{\alpha}\hat{\mathbf{f}},a\mathbf{R}+\mathbf{R}',\mathbb{W})}$$

and returns it to $\mathcal{A}$. It is easy to see that $a\mathbf{R} + \mathbf{R}'$ is uniformly random subject to constraint that the last two rows are all zero vectors. Furthermore, since the value of a is information-theoretically hidden only given $a\mathbf{R} + \mathbf{R}'$, $a\hat{\alpha}$ is uniformly and independently random over $\mathbb{Z}_p$. Finally, $\mathcal{B}$ outputs $\mathcal{A}$'s output as its guess. It is easy to see that the simulation by $\mathcal{B}$ is perfect. In particular, $\mathcal{B}$ has made only single call to $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$. Therefore, the lemma follows. □

Lemma 20. *For any broadcast encoding Π and adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$\mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,1),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) \leq \mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{B}_1,\Pi,Q_c,(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) + 2\mathsf{Adv}^{\mathsf{P}_8}_{\mathcal{B}_2}$$

and $\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2)\} \approx \mathsf{Time}(\mathcal{A}) + Q_c\mathsf{poly}(\kappa, n)$ *where* $\mathsf{poly}(\kappa, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We prove the lemma via the following sequence of games, which are defined for $b \in \{0, 1\}$. We write $\mathsf{Event}_{\mathrm{xx}}$ to denote the probability that $\mathcal{A}$ outputs 1 in $\mathsf{Game}_{\mathrm{xx}}$.

$\mathsf{Game}_{0,b}$: This is the $(Q_c, 1)$-MMH security game with oracles $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot)$ and $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$.

$\mathsf{Game}_{1,b}$: In this game, we change $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ as follows: When it is called, $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ runs $\mathsf{KEnc}(\tau^\star, p) \to \mathbf{k}$, picks $\hat{\alpha} \xleftarrow{\$} \mathbb{Z}_p$, $\mathbf{r}_1, \ldots, \mathbf{r}_{d_2'} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\hat{r}_1, \ldots, \hat{r}_{d_2'} \xleftarrow{\$} \mathbb{Z}_p$, and returns

$$h^{\mathbf{k}_\mathbf{Z}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\mathbf{R}+\hat{\mathbf{R}},\mathbb{W})} \tag{18}$$

instead of $h^{\mathbf{k}_\mathbf{Z}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\mathbf{R},\mathbb{W})}$. Here, $\mathbf{R} \in \mathbb{Z}_p^{4\times d_2'}$ and $\hat{\mathbf{R}} \in \mathbb{Z}_p^{4\times d_2'}$ are defined as

$$\mathbf{R} = \left(\begin{pmatrix}\mathbf{r}_1\\0\\0\end{pmatrix}, \cdots, \begin{pmatrix}\mathbf{r}_{d_2'}\\0\\0\end{pmatrix}\right) \quad \text{and} \quad \hat{\mathbf{R}} = \left(\begin{pmatrix}\mathbf{0}\\\hat{r}_1\\0\end{pmatrix}, \cdots, \begin{pmatrix}\mathbf{0}\\\hat{r}_{d_2'}\\0\end{pmatrix}\right). \tag{19}$$

$\mathsf{Game}_{2,b}$: This game is the same as $\mathsf{Game}_{1,b}$ except that we change how to choose $\mathbb{W}$. Here, we first pick $\mathbb{W}' = (\mathbf{W}'_1, \ldots, \mathbf{W}'_{d_1}) \xleftarrow{\$} (\mathbb{Z}_p^{4\times 4})^{d_1}$ and $\hat{w}_1, \ldots, \hat{w}_{d_1} \xleftarrow{\$} \mathbb{Z}_p$ and set

$$\mathbf{W}_i = \mathbf{W}'_i + \hat{w}_i \cdot \mathbf{BVB}^{-1} \tag{20}$$

for $i \in [d_1]$. In the above, $\mathbf{V} = \mathbf{e}_3^\top \cdot \mathbf{e}_3 \in \mathbb{Z}_p^{4\times 4}$, namely, $\mathbf{V}$ is the zero matrix with the $(3,3)$ entry is replaced with 1.

$\mathsf{Game}_{3,b}$: This game is the same as $\mathsf{Game}_{2,b}$, except that we change params as

$$\mathsf{params} = \begin{pmatrix} g, & g^{\pi_1(\mathbf{B})}, & g^{\pi_3(\mathbf{B})}, & g^{\pi_1(\mathbf{W}'_1\mathbf{B})}, & \ldots, & g^{\pi_1(\mathbf{W}'_{d_1}\mathbf{B})}, & g^{\pi_3(\mathbf{W}'_1\mathbf{B})}, & \ldots, & g^{\pi_3(\mathbf{W}'_{d_1}\mathbf{B})}, \\ h, & h^{\mathbf{Z}}, & & h^{\pi_1({\mathbf{W}'_1}^\top\mathbf{Z})}, & \ldots, & h^{\pi_1({\mathbf{W}'_{d_1}}^\top\mathbf{Z})}, & h^{\pi_3({\mathbf{W}'_1}^\top\mathbf{Z})}, & \ldots, & h^{\pi_3({\mathbf{W}'_{d_1}}^\top\mathbf{Z})} \end{pmatrix}.$$

Namely, we set up params using $\mathbb{W}'$ instead of $\mathbb{W}$. We remark that oracles $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot)$ and $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ are unchanged from the previous game.

We have that

$$\begin{aligned} & \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{A},\Pi,(Q_c,1),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) = |\Pr[\mathsf{Event}_{0,0}] - \Pr[\mathsf{Event}_{0,1}]| \\ \le\ & |\Pr[\mathsf{Event}_{3,0}] - \Pr[\mathsf{Event}_{3,1}]| + \sum_{b\in\{0,1\},i\in\{0,1,2\}} |\Pr[\mathsf{Event}_{i,b}] - \Pr[\mathsf{Event}_{i+1,b}]|. \end{aligned}$$

Therefore, we complete the proof by showing the following sequence of claims. □

Claim 3. ($\mathsf{Game}_{0,b}$ to $\mathsf{Game}_{1,b}$). *For any $b \in \{0,1\}$ and adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_1$ such that $|\Pr[\mathsf{Event}_{0,b}] - \Pr[\mathsf{Event}_{1,b}]| \le \mathsf{Adv}^{\mathsf{P}_8}_{\mathcal{B}_1}(\kappa)$ and $\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A}) + Q_c \cdot \mathsf{poly}(\kappa, n)$ where $\mathsf{poly}(\kappa, n)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. We construct $\mathcal{B}_1$ that attacks Problem 8 from $\mathcal{A}$. Given the problem instance $(g, h, g^{\pi_1(\mathbf{B})}, g^{\pi_3(\mathbf{B})}, g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\\hat{t}\\0\end{pmatrix}}, h^{\mathbf{Z}}, T = h^{\mathbf{Z}\begin{pmatrix}\mathbf{u}\\\hat{u}\\0\end{pmatrix}})$ where $\hat{u} \xleftarrow{\$} \mathbb{Z}_p^*$ or $\hat{u} = 0$, $\mathcal{B}_1$ simulates $(Q_c, 1)$-MMH security game for $\mathcal{A}$ as follows.

Setup of Parameters. Given $\tau^\star \in [n]$ from $\mathcal{A}$, $\mathcal{B}_1$ picks $\mathbf{W}_1, \ldots, \mathbf{W}_{d_1} \xleftarrow{\$} \mathbb{Z}_p^{4\times 4}$ and computes $g^{\pi_1(\mathbf{W}_j\mathbf{B})} = g^{\mathbf{W}_j\pi_1(\mathbf{B})}$ and $h^{\pi_1(\mathbf{W}_j^\top\mathbf{Z})}$ for $j \in [d_1]$ from $g^{\pi_1(\mathbf{B})}$ and $h^{\mathbf{Z}}$. Then it gives $\mathsf{params} = \big(g, g^{\pi_1(\mathbf{B})}, g^{\pi_3(\mathbf{B})}, \{g^{\pi_1(\mathbf{W}_i\mathbf{B})}\}_{i\in[d_1]}, \{g^{\pi_3(\mathbf{W}_i\mathbf{B})}\}_{i\in[d_1]}, h, h^{\mathbf{Z}}, \{h^{\pi_1(\mathbf{W}_i^\top\mathbf{Z})}\}_{i\in[d_1]}, \{h^{\pi_3(\mathbf{W}_i^\top\mathbf{Z})}\}_{i\in[d_1]}\big)$ to $\mathcal{A}$.

Simulating $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot)$**.** Given $S \subset [n]$, $\mathcal{B}_1$ first runs $\mathsf{CEnc}(S,p) \to (\mathbf{c}, d_3')$. Then it runs the algorithm in Lemma 17 on input $(g^{\pi_1(\mathbf{B})}, g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{t}\\ \hat{t}\\ 0\end{smallmatrix}\right)})$ to obtain $g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_i\\ \hat{s}_i\\ 0\end{smallmatrix}\right)}$ for $i \in [0, d_3']$ where $\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\hat{s}_i \xleftarrow{\$} \mathbb{Z}_p$ (Recall that $\hat{t} \neq 0$). Finally, it computes $g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})}$ and returns it to $\mathcal{A}$, where $\mathbf{S}$ and $\hat{\mathbf{S}}$ are defined as Equation (8). Note that it can be efficiently computable from $\{g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_i\\ \hat{s}_i\\ 0\end{smallmatrix}\right)}\}_{i\in[0,d_3']}$ and $\{\mathbf{W}_j\}_{j\in[d_1]}$ since $\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})$ only contains linear combination of $\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_i\\ \hat{s}_i\\ 0\end{smallmatrix}\right)$ and $\mathbf{W}_i\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_j\\ \hat{s}_j\\ 0\end{smallmatrix}\right)$.

Simulating $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$**.** When it is called, $\mathcal{B}_1$ first runs $\mathsf{KEnc}(\tau^\star,p) \to \mathbf{k}$ and picks $\hat{\alpha} \xleftarrow{\$} \mathbb{Z}_p$. Then, it runs the algorithm in Lemma 17 on input $(h^{\pi_1(\mathbf{Z})}, h^{\mathbf{Z}\left(\begin{smallmatrix}\mathbf{u}\\ \hat{u}\\ 0\end{smallmatrix}\right)})$ to obtain $h^{\mathbf{Z}\left(\begin{smallmatrix}\mathbf{r}_i\\ \hat{r}_i\\ 0\end{smallmatrix}\right)}$ for $i \in [1, d_2']$ where $(\mathbf{r}_i \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}, \hat{r}_i \xleftarrow{\$} \mathbb{Z}_p)$ if $\hat{u} \neq 0$ and $(\mathbf{r}_i \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}, \hat{r}_i = 0)$ if $\hat{u} = 0$. Finally, it computes $h^{\mathbf{k}_\mathbf{Z}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\mathbf{R}+\hat{\mathbf{R}},\mathbb{W})}$ and returns it to $\mathcal{A}$. Here, $\mathbf{R}$ and $\hat{\mathbf{R}}$ are defined as Equation (19). Note that it is efficiently computable from $\hat{\alpha}$,$h^{\mathbf{Z}}$, $\{h^{\mathbf{Z}\left(\begin{smallmatrix}\mathbf{r}_i\\ \hat{r}_i\\ 0\end{smallmatrix}\right)}\}_{i\in[d_2']}$, and $\{\mathbf{W}_j\}_{j\in[d_1]}$ since $\mathbf{k}_\mathbf{Z}(b\cdot\hat{\alpha}\hat{\mathbf{f}}, \mathbf{R}+\hat{\mathbf{R}}, \mathbb{W})$ only contains linear combination of $\mathbf{Z}\left(\begin{smallmatrix}\mathbf{r}_i\\ \hat{r}_i\\ 0\end{smallmatrix}\right)$, $\mathbf{W}_j^\top\mathbf{Z}\left(\begin{smallmatrix}\mathbf{r}_i\\ \hat{r}_i\\ 0\end{smallmatrix}\right)$, and $\hat{\alpha}\hat{\mathbf{f}} = \hat{\alpha}\mathbf{Z}\mathbf{e}_3^\top$.

Guess. Finally, $\mathcal{B}_1$ outputs $\mathcal{A}$'s output as its guess. It can be seen that the game corresponds to $\mathsf{Game}_{0,b}$ if $\hat{u} = 0$ and $\mathsf{Game}_{1,b}$ if $\hat{u} \xleftarrow{\$} \mathbb{Z}_p^*$. Thus, we may conclude that $|\Pr(\mathsf{Event}_{0,b}) - \Pr(\mathsf{Event}_{1,b})| \leq \mathsf{Adv}^{\mathsf{P}_8}_{\mathcal{B}_1}(\kappa)$. □

Claim 4. ($\mathsf{Game}_{1,b}$ to $\mathsf{Game}_{2,b}$). *For any adversary $\mathcal{A}$, we have* $\Pr[\mathsf{Event}_{1,b}] = \Pr[\mathsf{Event}_{2,b}]|$.

Proof. Since each $\mathbf{W}_i$ is uniformly random over $\mathbb{Z}_p^{4\times 4}$ in both games, the change from $\mathsf{Game}_{1,b}$ to $\mathsf{Game}_{2,b}$ is only conceptual. Therefore, the claim follows. □

Claim 5. ($\mathsf{Game}_{2,b}$ to $\mathsf{Game}_{3,b}$). *For any adversary $\mathcal{A}$, we have* $\Pr[\mathsf{Event}_{2,b}] = \Pr[\mathsf{Event}_{3,b}]|$.

Proof. We claim that this change is only conceptual. To see this, it suffices to observe that

$$\begin{aligned}
\pi_i(\mathbf{W}_j\mathbf{B}) &= \pi_i(\mathbf{W}_j'\mathbf{B} + \hat{w}_j\mathbf{B}\mathbf{V}) = \pi_i(\mathbf{W}_j'\mathbf{B}) + \hat{w}_j\mathbf{B}\cdot\pi_i(\mathbf{V}) = \pi_i(\mathbf{W}_j'\mathbf{B})\\
\pi_i(\mathbf{W}_j^\top\mathbf{Z}) &= \pi_i({\mathbf{W}_j'}^\top\mathbf{Z} + \hat{w}_j\mathbf{B}^{-\top}\mathbf{V}^\top\mathbf{B}^\top\mathbf{Z}) = \pi_i({\mathbf{W}_j'}^\top\mathbf{Z} + \hat{w}_j\mathbf{B}^{-\top}\mathbf{V}\mathbf{B}^\top\mathbf{B}^{-\top}\mathbf{D})\\
&= \pi_i({\mathbf{W}_j'}^\top\mathbf{Z}) + \hat{w}_j\mathbf{B}^{-\top}\pi_i(\mathbf{V}^\top\mathbf{D}) = \pi_i({\mathbf{W}_j'}^\top\mathbf{Z})
\end{aligned}$$

hold for $i \in \{1,3\}$, $j \in [d_1]$. In the above, we use the fact that $\pi_i(\mathbf{V})$ and $\pi_i(\mathbf{V}^\top\mathbf{D})$ are zero matrices for $i \in \{1,3\}$. This follows because $\mathbf{D}$ is a diagonal matrix and $\mathbf{V} = \mathbf{e}_3^\top\cdot\mathbf{e}_3$. □

Claim 6. ($\mathsf{Game}_{3,0}$ to $\mathsf{Game}_{3,1}$). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_2$ such that* $|\Pr[\mathsf{Event}_{3,0}] - \Pr[\mathsf{Event}_{3,1}]| \leq \mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{B}_2,\Pi,Q_c,(\mathbb{G}_1,\mathbb{G}_2)}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + Q_c\cdot\mathsf{poly}(\kappa,n)$ *where* $\mathsf{poly}(\kappa,n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. Before going into the proof, we need some definitions and preparations.

Preparations. Let $\hat{\mathbf{w}} = (\hat{w}_1,\ldots,\hat{w}_{d_1})$ and $\mathbf{S}$ and $\hat{\mathbf{S}}$ be as in Equation (8). Then, for $S \subset [n]$ and $\mathsf{CEnc}(S,p) \to (\mathbf{c}, d_3')$, we have that

$$\begin{aligned}
\mathbf{c}_\mathbf{B}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W}) &= \left\{\left(\sum_{j\in[0,d_3']} a_{\iota,j}\mathbf{B}\begin{pmatrix}\mathbf{s}_j\\ \hat{s}_j\\ 0\end{pmatrix}\right) + \left(\sum_{(j,k)\in[0,d_3']\times[d_1]} a_{\iota,j,k}(\mathbf{W}_k' + \hat{w}_k\cdot\mathbf{B}\mathbf{V}\mathbf{B}^{-1})\mathbf{B}\begin{pmatrix}\mathbf{s}_j\\ \hat{s}_j\\ 0\end{pmatrix}\right)\right\}_{\iota\in[d_3]}\\
&= \left\{\left(\sum_{j\in[0,d_3']} a_{\iota,j}\mathbf{B}\begin{pmatrix}\mathbf{s}_j\\ \hat{s}_j\\ 0\end{pmatrix}\right) + \left(\sum_{(j,k)\in[0,d_3']\times[d_1]} a_{\iota,j,k}\left(\mathbf{W}_k'\mathbf{B}\begin{pmatrix}\mathbf{s}_j\\ \hat{s}_j\\ 0\end{pmatrix} + \mathbf{B}\begin{pmatrix}\mathbf{0}\\ \hat{w}_k\hat{s}_j\\ 0\end{pmatrix}\right)\right)\right\}_{\iota\in[d_3]}
\end{aligned}$$

$$= \mathbf{c_B}(\mathbf{S}, \mathbb{W}') + \mathbf{c}'_{\mathbf{B}}(\hat{\mathbf{S}}, \mathbb{W}', \hat{\mathbf{w}})$$

where $\mathbf{c}'_{\mathbf{B}}(\hat{\mathbf{S}}, \mathbb{W}', \hat{\mathbf{w}})$ is defined as

$$\begin{aligned}
\mathbf{c}'_{\mathbf{B}}(\hat{\mathbf{S}}, \mathbb{W}', \hat{\mathbf{w}}) &= \left\{ \left(\sum_{j \in [0, d'_3]} a_{\iota,j} \mathbf{B} \begin{pmatrix} \mathbf{0} \\ \hat{s}_j \\ 0 \end{pmatrix} \right) + \left(\sum_{(j,k) \in [0,d'_3] \times [d_1]} a_{\iota,j,k} \left(\mathbf{W}'_k \mathbf{B} \begin{pmatrix} \mathbf{0} \\ \hat{s}_j \\ 0 \end{pmatrix} + \mathbf{B} \begin{pmatrix} \mathbf{0} \\ \hat{w}_k \hat{s}_j \\ 0 \end{pmatrix} \right) \right) \right\}_{\iota \in [d_3]} \\
&= \left\{ \mathbf{B} \begin{pmatrix} \mathbf{0} \\ c_\iota(\hat{\mathbf{s}}, \hat{\mathbf{w}}) \\ 0 \end{pmatrix} + \sum_{(j,k) \in [0,d'_3] \times [d_1]} a_{\iota,j,k} \mathbf{W}'_k \mathbf{B} \begin{pmatrix} \mathbf{0} \\ \hat{s}_j \\ 0 \end{pmatrix} \right\}_{\iota \in [d_3]}.
\end{aligned}$$

In the above, $\hat{\mathbf{s}} = (\hat{s}_0, \hat{s}_1, \ldots, \hat{s}_{d'_3})$. We observe that $g^{\mathbf{c}'_{\mathbf{B}}(\hat{\mathbf{S}}, \mathbb{W}', \hat{\mathbf{w}})}$ can be efficiently computed given $\mathbf{B}$, $\mathbb{W}'$, and $g^{\mathbf{c}(\hat{\mathbf{s}}, \hat{\mathbf{w}})}$, since $g^{\mathbf{c}(\hat{\mathbf{s}}, \hat{\mathbf{w}})}$ contains the terms $(g^{\hat{s}_0}, \ldots, g^{\hat{s}_{d'_3}})$ because of the regularity of the encoding (the second condition).

Let $\beta \in \mathbb{Z}_p$ and $\mathbf{R}$ and $\hat{\mathbf{R}}$ be as in Equation (19). Then, for $\tau \in [n]$ and $\mathsf{KEnc}(\tau, p) \to (\mathbf{k}, d'_2)$, we have that

$$\begin{aligned}
&\mathbf{k_Z}(\boldsymbol{\alpha} + \beta \hat{\mathbf{f}}, \mathbf{R} + \hat{\mathbf{R}}, \mathbb{W}) \\
&= \left\{ b_\iota(\boldsymbol{\alpha} + \beta \hat{\mathbf{f}}) + \left(\sum_{i \in [d'_2]} b_{\iota,i} \mathbf{Z} \begin{pmatrix} \mathbf{r}_i \\ \hat{r}_i \\ 0 \end{pmatrix} \right) + \left(\sum_{(i,j) \in [d'_2] \times [d_1]} b_{\iota,i,j} (\mathbf{W}'_j + \hat{w}_j \cdot \mathbf{B}\mathbf{V}\mathbf{B}^{-1})^\top \mathbf{Z} \begin{pmatrix} \mathbf{r}_i \\ \hat{r}_i \\ 0 \end{pmatrix} \right) \right\}_{\iota \in [d_2]} \\
&= \left\{ b_\iota(\boldsymbol{\alpha} + \beta \hat{\mathbf{f}}) + \left(\sum_{i \in [d'_2]} b_{\iota,i} \mathbf{Z} \begin{pmatrix} \mathbf{r}_i \\ \hat{r}_i \\ 0 \end{pmatrix} \right) + \left(\sum_{(i,j) \in [d'_2] \times [d_1]} b_{\iota,i,j} \left({\mathbf{W}'_j}^\top \mathbf{Z} \begin{pmatrix} \mathbf{r}_i \\ \hat{r}_i \\ 0 \end{pmatrix} + \hat{w}_j \cdot \mathbf{Z} \begin{pmatrix} \mathbf{0} \\ \hat{r}_i \\ 0 \end{pmatrix} \right) \right) \right\}_{\iota \in [d_2]} \\
&= \mathbf{k_Z}(\boldsymbol{\alpha}, \mathbf{R}, \mathbb{W}') + \mathbf{k}'_{\mathbf{Z}}(\beta \hat{\mathbf{f}}, \hat{\mathbf{R}}, \mathbb{W}', \hat{\mathbf{w}}).
\end{aligned}$$

In the third line above, we used the fact that $(\mathbf{B}\mathbf{V}\mathbf{B}^{-1})^\top \mathbf{Z} = \mathbf{B}^{-\top}\mathbf{V}^\top\mathbf{D} = \mathbf{B}^{-\top}\mathbf{V} = \mathbf{B}^{-\top}\mathbf{D}\mathbf{V} = \mathbf{Z}\mathbf{V}$. Here, $\mathbf{k}'_{\mathbf{Z}}(\beta \hat{\mathbf{f}}, \hat{\mathbf{R}}, \mathbb{W}', \hat{\mathbf{w}})$ is defined as

$$\begin{aligned}
&\mathbf{k}'_{\mathbf{Z}}(\beta \hat{\mathbf{f}}, \hat{\mathbf{R}}, \mathbb{W}', \hat{\mathbf{w}}) \\
&= \left\{ b_\iota \cdot \mathbf{Z} \begin{pmatrix} \mathbf{0} \\ \beta \\ 0 \end{pmatrix} + \left(\sum_{i \in [d'_2]} b_{\iota,i} \mathbf{Z} \begin{pmatrix} \mathbf{0} \\ \hat{r}_i \\ 0 \end{pmatrix} \right) + \left(\sum_{(i,j) \in [d'_2] \times [d_1]} b_{\iota,i,j} \left({\mathbf{W}'_j}^\top \mathbf{Z} \begin{pmatrix} \mathbf{0} \\ \hat{r}_i \\ 0 \end{pmatrix} + \hat{w}_j \cdot \mathbf{Z} \begin{pmatrix} \mathbf{0} \\ \hat{r}_i \\ 0 \end{pmatrix} \right) \right) \right\}_{\iota \in [d_2]} \\
&= \left\{ \mathbf{Z} \begin{pmatrix} \mathbf{0} \\ k_\iota(\beta, \hat{\mathbf{r}}, \hat{\mathbf{w}}) \\ 0 \end{pmatrix} + \sum_{(i,j) \in [d'_2] \times [d_1]} b_{\iota,i,j} {\mathbf{W}'_j}^\top \mathbf{Z} \begin{pmatrix} \mathbf{0} \\ \hat{r}_i \\ 0 \end{pmatrix} \right\}_{\iota \in [d_2]}.
\end{aligned}$$

In the above, $\hat{\mathbf{r}} = (\hat{r}_1, \ldots, \hat{r}_{d'_2})$. Similarly to the case of $g^{\mathbf{c}'_{\mathbf{B}}(\hat{\mathbf{S}}, \mathbb{W}', \hat{\mathbf{w}})}$, we can efficiently compute $h^{\mathbf{k}'_{\mathbf{Z}}(\beta \hat{\mathbf{f}}, \hat{\mathbf{R}}, \mathbb{W}', \hat{\mathbf{w}})}$ given $\mathbf{Z}$, $\mathbb{W}'$, and $h^{\mathbf{k}(\beta \hat{\mathbf{f}}, \hat{\mathbf{r}}, \hat{\mathbf{w}})}$ since $h^{\mathbf{k}(\beta \hat{\mathbf{f}}, \hat{\mathbf{r}}, \hat{\mathbf{w}})}$ contains the terms $(h^{\hat{r}_1}, \ldots, h^{\hat{r}_{d'_2}})$ because of the regularity of the encoding (the second condition).

Reduction. Having finished the preparation, we now construct an adversary $\mathcal{B}_2$ against Q_c-CMH security of the broadcast encoding from an adversary $\mathcal{A}$ who distinguishes $\mathsf{Game}_{3,0}$ from $\mathsf{Game}_{3,1}$.

Setup of Parameters. Given $\tau^\star$ from $\mathcal{A}$, $\mathcal{B}_2$ submits the same index. Then, $\mathcal{B}_2$ is given (g, h). It then picks $\mathbf{B} \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$, $\mathbb{W}' = (\mathbf{W}'_1, \ldots, \mathbf{W}'_{d_1}) \xleftarrow{\$} (\mathbb{Z}_p^{4\times 4})^{d_1}$, and a random full-rank diagonal matrix $\mathbf{D} \in \mathbb{Z}_p^{4\times 4}$ with the entries $(3,3)$ and $(4,4)$ being 1. Then, it sets $\mathbf{Z} = \mathbf{B}^{-\top}\mathbf{D}$. Finally, it computes and gives $\mathsf{params} =$

$\big(g, g^{\pi_1(\mathbf{B})}, g^{\pi_3(\mathbf{B})}, \{g^{\pi_1(\mathbf{W}'_i\mathbf{B})}\}_{i\in[d_1]}, \{g^{\pi_3(\mathbf{W}'_i\mathbf{B})}\}_{i\in[d_1]}, h, h^{\mathbf{Z}}, \{h^{\pi_1(\mathbf{W}'^{\top}_i\mathbf{Z})}\}_{i\in[d_1]}, \{h^{\pi_3(\mathbf{W}'^{\top}_i\mathbf{Z})}\}_{i\in[d_1]}\big)$ to $\mathcal{A}$. In the following, $\mathcal{B}_2$ implicitly sets $\mathbf{W}_i = \mathbf{W}'_i + \hat{w}_i\mathbf{B}\mathbf{V}\mathbf{B}^{-1}$ for $i \in [d_1]$ and $\mathbb{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_{d_1})$ where $\hat{w}_i$ is chosen by the game and is not (explicitly) known to $\mathcal{B}_2$.

Simulating $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot)$. When $\mathcal{A}$ calls $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot)$ on input $S \subset [n]$, $\mathcal{B}_2$ submits the same S to its oracle $\mathcal{O}^{\mathsf{CMH,C}}_{\tau^\star,\hat{\mathbf{w}}}(\cdot)$ to obtain $g^{\mathbf{c}(\hat{\mathbf{s}},\hat{\mathbf{w}})}$, where $\mathbf{c} \leftarrow \mathsf{CEnc}(S, p)$ and $\hat{\mathbf{s}} = (\hat{s}_0, \ldots, \hat{s}_{d'_3}) \xleftarrow{\$} \mathbb{Z}_p^{d'_3+1}$. $\mathcal{B}_2$ then picks $\mathbf{s}_0, \ldots, \mathbf{s}_{d'_3} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and computes $g^{\mathbf{c}_{\mathbf{B}}(\mathbf{S},\mathbb{W}')}$ and $g^{\mathbf{c}'_{\mathbf{B}}(\hat{\mathbf{S}},\mathbb{W}',\hat{\mathbf{w}})}$. The latter can be efficiently computed from $\mathbf{B}$, $\mathbb{W}'$, and $g^{\mathbf{c}(\hat{\mathbf{s}},\hat{\mathbf{w}})}$. Finally, it computes $g^{\mathbf{c}_{\mathbf{B}}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})} = g^{\mathbf{c}_{\mathbf{B}}(\mathbf{S},\mathbb{W}')} \cdot g^{\mathbf{c}'_{\mathbf{B}}(\hat{\mathbf{S}},\mathbb{W}',\hat{\mathbf{w}})}$ and returns it to $\mathcal{A}$.

Simulating $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$. When $\mathcal{A}$ calls $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$, $\mathcal{B}_2$ calls its oracle $\mathcal{O}^{\mathsf{CMH,K}}_{\tau^\star,\hat{\mathbf{w}},b}(\cdot)$ to obtain $h^{\mathbf{k}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\hat{\mathbf{r}},\hat{\mathbf{w}})}$, where $\mathbf{k} \leftarrow \mathsf{KEnc}(\tau, p)$ and $\hat{\mathbf{r}} = (\hat{r}_1, \ldots, \hat{r}_{d'_2}) \xleftarrow{\$} \mathbb{Z}_p^{d'_2}$. $\mathcal{B}_2$ then picks $\mathbf{r}_1, \ldots, \mathbf{r}_{d'_2} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and computes $h^{\mathbf{k}_{\mathbf{Z}}(\mathbf{0},\mathbf{R},\mathbb{W}')}$ and $h^{\mathbf{k}'_{\mathbf{Z}}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\hat{\mathbf{R}},\mathbb{W}',\hat{\mathbf{w}})}$. The latter can be efficiently computed from $\mathbf{Z}$, $\mathbb{W}'$, and $h^{\mathbf{k}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\hat{\mathbf{r}},\hat{\mathbf{w}})}$. Finally, it computes $h^{\mathbf{k}_{\mathbf{Z}}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\mathbf{R}+\hat{\mathbf{R}},\mathbb{W})} = h^{\mathbf{k}_{\mathbf{Z}}(\mathbf{0},\mathbf{R},\mathbb{W}')} \cdot h^{\mathbf{k}'_{\mathbf{Z}}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\hat{\mathbf{R}},\mathbb{W}',\hat{\mathbf{w}})}$ and returns it to $\mathcal{A}$.

Guess. Finally, $\mathcal{B}_2$ outputs $\mathcal{A}$'s output as its guess. It can be seen that the game corresponds to $\mathsf{Game}_{3,0}$ if $b = 0$ (i.e., if $\mathcal{B}_2$ is equipped with the oracle $\mathcal{O}^{\mathsf{CMH,K}}_{\tau^\star,\hat{\mathbf{w}},0}(\cdot)$) and $\mathsf{Game}_{3,1}$ if $b = 1$(i.e., if $\mathcal{B}_2$ is equipped with the oracle $\mathcal{O}^{\mathsf{CMH,K}}_{\tau^\star,\hat{\mathbf{w}},1}(\cdot)$). Thus, we may conclude that $|\Pr[\mathsf{Event}_{3,0}] - \Pr[\mathsf{Event}_{3,1}]| \le \mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{B}_2,\Pi,Q_c,(\mathbb{G}_1,\mathbb{G}_2)}(\kappa)$. □

D.4 Security Proof for Our Scheme in Section 5.5

For the sake of simplicity, we first show the following theorem that establishes the security of the scheme for the case of $\mu = 1$ (i.e., the single instance case). Later, we explain how to modify our proof for the single-instance case to deal with the multi-instance case.

Theorem 13. *For any adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, and $\mathcal{B}_4$ such that*

$$\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A},\Phi^{\mathsf{prime}},(1,Q_c,Q_k)}(\kappa) \le \mathsf{Adv}^{\mathsf{P}_7}_{\mathcal{B}_1}(\kappa) + \mathsf{Adv}^{\mathsf{P}_{11}}_{\mathcal{B}_2}(\kappa) + 2\ell\left(\mathsf{Adv}^{\mathsf{P}_{10}}_{\mathcal{B}_3}(\kappa) + \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B}_4,\Pi,(Q_c,Q_k),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa)\right)$$

and $\max\{\mathsf{Time}(\mathcal{B}_i) | i \in [1,4]\} \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k)\cdot\mathsf{poly}(\kappa,\ell)$ *where* $\mathsf{poly}(\kappa,\ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. (of Theorem 13.) The proof of the theorem is almost parallel to that of Theorem 6.

Semi-functional Ciphertexts and Private Keys. We define several types of ciphertexts and private keys that are used in the security proof. In the following, we will pick random functions $\widehat{\mathsf{R}}_i : \{0,1\}^i \to \mathbb{Z}_p$ and $\widetilde{\mathsf{R}}_i : \{0,1\}^i \to \mathbb{Z}_p$ (via lazy sampling) for $i = 0, \ldots, \ell$. Here, we use $\{0,1\}^0$ to denote the singleton set containing just the empty string ϵ. For an identity $\mathsf{ID} \in \{0,1\}^\ell$, $\mathsf{ID}|_i$ denotes the first i bits of ID, that is, length i prefix of ID.

- **Semi-functional ciphertexts.** We consider Type 1, Type $(2,i)$, Type $(3,i)$, and Type 4 of semi-functional ciphertexts for $i \in [0,\ell]$. The form of semi-functional ciphertext are as follows.

$$\mathsf{CT} = \begin{cases} \left(g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})},\quad e\big(g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_0\\ \hat{s}_0\\ 0\end{pmatrix}}, h^{\boldsymbol{\alpha}}\big)\cdot\mathsf{M}_{\mathsf{coin}}\right) & \text{Type 1} \quad (21)\\ \left(g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})},\quad e\big(g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_0\\ \hat{s}_0\\ 0\end{pmatrix}}, h^{\boldsymbol{\alpha}+\widehat{\mathsf{R}}_i(\mathsf{ID}|_i)\cdot\hat{\mathbf{f}}}\big)\cdot\mathsf{M}_{\mathsf{coin}}\right) & \text{Type } (2,i) \quad (22)\\ \left(g^{\mathbf{c}(\mathbf{S}+\tilde{\mathbf{S}},\mathbb{W})},\quad e\big(g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_0\\ 0\\ \tilde{s}_0\end{pmatrix}}, h^{\boldsymbol{\alpha}+\widetilde{\mathsf{R}}_i(\mathsf{ID}|_i)\cdot\tilde{\mathbf{f}}}\big)\cdot\mathsf{M}_{\mathsf{coin}}\right) & \text{Type } (3,i)\,. \quad (23)\end{cases}$$

In the above,

$$\begin{aligned}\mathbf{S} &= \left(\begin{pmatrix}\mathbf{s}_0\\0\\0\end{pmatrix},\begin{pmatrix}\mathbf{s}_1\\0\\0\end{pmatrix},\cdots,\begin{pmatrix}\mathbf{s}_{d_3'}\\0\\0\end{pmatrix}\right), \quad \hat{\mathbf{S}} = \left(\begin{pmatrix}\mathbf{0}\\\hat{s}_0\\0\end{pmatrix},\begin{pmatrix}\mathbf{0}\\\hat{s}_1\\0\end{pmatrix},\cdots,\begin{pmatrix}\mathbf{0}\\\hat{s}_{d_3'}\\0\end{pmatrix}\right),\\ \tilde{\mathbf{S}} &= \left(\begin{pmatrix}\mathbf{0}\\0\\\tilde{s}_0\end{pmatrix},\begin{pmatrix}\mathbf{0}\\0\\\tilde{s}_1\end{pmatrix},\cdots,\begin{pmatrix}\mathbf{0}\\0\\\tilde{s}_{d_3'}\end{pmatrix}\right), \quad \hat{\mathbf{f}} = \mathbf{Z}\mathbf{e}_3^\top, \quad \tilde{\mathbf{f}} = \mathbf{Z}\mathbf{e}_4^\top \end{aligned} \tag{24}$$

where $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{d_3'} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and $\hat{s}_0, \hat{s}_1, \ldots, \hat{s}_{d_3'}, \tilde{s}_0, \tilde{s}_1, \ldots, \tilde{s}_{d_3'} \xleftarrow{\$} \mathbb{Z}_p$.

- **Random ciphertexts.** We also consider random ciphertexts. The form of ciphertexts are as follows.

$$\mathsf{CT} = \left(g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})}, \quad \mathsf{M}_{\mathsf{rand}} \right) \tag{25}$$

where $\mathsf{M}_{\mathsf{rand}} \xleftarrow{\$} \mathbb{G}_T$ and $\mathbf{S}$ and $\hat{\mathbf{S}}$ are defined as above.

- **Semi-functional Private Keys.** We consider Type $(1,i)$ for $i \in [0,\ell]$ and Type $(2,i)$ for $i \in [0,\ell-1]$ of semi-functional private keys. To create a semi-functional private key, we replace sk_j in Equation (9) with

$$\mathsf{sk}_j = h^{\mathbf{k}_j(\boldsymbol{\alpha}_j+\hat{\gamma}_j\hat{\mathbf{f}}+\tilde{\gamma}_j\tilde{\mathbf{f}},\mathbf{R}_j,\mathbb{W})} = \{\mathsf{sk}_{j,\eta} = h^{k_{j,\eta}(\boldsymbol{\alpha}_j+\hat{\gamma}_j\hat{\mathbf{f}}+\tilde{\gamma}_j\tilde{\mathbf{f}},\mathbf{R}_j,\mathbb{W})}\}_{\eta\in[d_2]} \tag{26}$$

where $\{\hat{\gamma}_j \in \mathbb{Z}_p\}_{j\in S}$ and $\{\tilde{\gamma}_j \in \mathbb{Z}_p\}_{j\in S}$ are random numbers subject to constraint that

$$\sum_{j\in S}\hat{\gamma}_j = \begin{cases}\widehat{\mathsf{R}}_i(\mathsf{ID}|_i) & \text{Type } (1,i)\\ \widehat{\mathsf{R}}_{i+1}(\mathsf{ID}|_{i+1}) & \text{Type } (2,i)\end{cases}, \qquad \sum_{j\in S}\tilde{\gamma}_j = \widetilde{\mathsf{R}}_i(\mathsf{ID}|_i) \quad \text{Type } (1,i) \text{ and } (2,i).$$

Then the semi-functional private key is created as Equation (10).

Sequence of Games. Next, we define a sequence of games to establish the security of the IBE scheme. We write $\mathsf{Adv}_{\mathrm{xx}}(\kappa)$ to denote the advantage of $\mathcal{A}$ in $\mathsf{Game}_{\mathrm{xx}}$.

Game_0 : This is the real security game.

Game_1: In this game, all challenge ciphertexts are changed to be Type 1.

$\mathsf{Game}_{2,i,1}$ (for $i \in [1,\ell+1]$): In this game, all challenge ciphertexts are of Type $(2,i-1)$ whereas all private keys created by the challenger are Type $(1,i-1)$.

$\mathsf{Game}_{2,i,2}$ (for $i \in [1,\ell]$): This game is the same as $\mathsf{Game}_{2,i,1}$ except that challenge ciphertexts for identities ID such that $\mathsf{ID}_i = 0$ are changed to be Type $(3,i-1)$ where ID_i is the i-th bit of ID.

$\mathsf{Game}_{2,i,3}$ (for $i \in [1,\ell]$): This game is the same as $\mathsf{Game}_{2,i,2}$ except that challenge ciphertexts for identity ID such that $\mathsf{ID}_i = 1$ are changed to be Type $(2,i)$ and all private keys are changed to be Type $(2,i-1)$.

$\mathsf{Game}_{2,i,4}$ (for $i \in [1,\ell]$): This game is the same as $\mathsf{Game}_{2,i,3}$ except that challenge ciphertexts for identity ID such that $\mathsf{ID}_i = 0$ are changed to be Type $(3,i)$ and all private keys are changed to be Type $(1,i)$.

Game_3: This game is the same as $\mathsf{Game}_{2,\ell+1,1}$ except that all ciphertexts are changed to be random ciphertexts.

Observe that we have $\mathsf{Adv}_3(\kappa) = 0$ since the view of $\mathcal{A}$ is independent from the value of coin in Game_3. We have that

$$\mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{A},\Phi^{\mathsf{prime}},(1,Q_c,Q_k)}(\kappa) \quad = \quad \Pr[\mathsf{Adv}_0]$$

$$
\begin{aligned}
&\leq |\Pr[\mathsf{Adv}_0] - \Pr[\mathsf{Adv}_1]| + |\Pr[\mathsf{Adv}_1] - \Pr[\mathsf{Adv}_{2,1,1}]| \\
&+ \sum_{i\in[\ell],j\in[1,3]} |\Pr[\mathsf{Adv}_{2,i,j}] - \Pr[\mathsf{Adv}_{2,i,j+1}]| + \sum_{i\in[\ell]} |\Pr[\mathsf{Adv}_{2,i,4}] - \Pr[\mathsf{Adv}_{2,i+1,1}]| \\
&+ |\Pr[\mathsf{Adv}_{2,\ell+1,1}] - \Pr[\mathsf{Adv}_3]| + \Pr[\mathsf{Adv}_3]
\end{aligned}
$$

Therefore, we complete the proof by showing Lemma 21, 22, 23, 24, 25, 26, and 27. □

Lemma 21. (Game_0 to Game_1). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_1$ such that* $|\mathsf{Adv}_0(\kappa) - \mathsf{Adv}_1(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_7}_{\mathcal{B}_1}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k)\cdot\mathsf{poly}(\kappa,\ell)$ *where* $\mathsf{poly}(\kappa,\ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We construct $\mathcal{B}_1$ that attacks Problem 7 from an adversary $\mathcal{A}$ who distinguishes the games. We note that these games only differ in the creation of challenge ciphertexts. $\mathcal{B}_1$ simulates the challenger for $\mathcal{A}$ as follows.

Setup. At the outset of the game, $\mathcal{B}_1$ is given the problem instance of the assumption $(g, h, g^{\mathbf{B}}, h^{\pi_1(\mathbf{Z})}, g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ 0\end{pmatrix}})$ where either $\hat{t} = 0$ or $\hat{t} \xleftarrow{\$} \mathbb{Z}_p^*$. Then, it runs $\mathsf{Param}(2\ell, p) \to d_1$, picks $\mathbf{W}_1, \ldots, \mathbf{W}_{d_1} \xleftarrow{\$} \mathbb{Z}_p^{4\times 4}$ and $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{Z}_p^{4\times 1}$. It also computes $g^{\pi_1(\mathbf{W}_j\mathbf{B})}$ and $h^{\pi_1(\mathbf{W}_j^\top\mathbf{Z})} = h^{\mathbf{W}_j^\top\pi_1(\mathbf{Z})}$ for $j \in [d_1]$. Note that the latter can be efficiently computed from $h^{\pi_1(\mathbf{Z})}$. Finally, it returns the public parameter $\mathsf{pp} = (g, g^{\pi_1(\mathbf{B})}, g^{\pi_1(\mathbf{W}_1\mathbf{B})}, \ldots, g^{\pi_1(\mathbf{W}_{d_1}\mathbf{B})}, h, h^{\pi_1(\mathbf{Z})}, h^{\pi_1(\mathbf{W}_1^\top\mathbf{Z})}, \ldots, h^{\pi_1(\mathbf{W}_{d_1}^\top\mathbf{Z})})$ and the master public key $\mathsf{mpk} = (\mathsf{pp}, e(g,h)^{\boldsymbol{\alpha}^\top\pi_1(\mathbf{B})})$ to $\mathcal{A}$. $\mathcal{B}_1$ also sets $\mathsf{msk} = \boldsymbol{\alpha}$ and flips a coin $\mathsf{coin} \xleftarrow{\$} \{0,1\}$.

Key Extraction Queries. When the adversary $\mathcal{A}$ submits $(\texttt{Extraction}, 1, \mathsf{ID})$ to the challenger, $\mathcal{B}_1$ simply runs $\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{ID}) \to \mathsf{sk}_{\mathsf{ID}}$ and returns $\mathsf{sk}_{\mathsf{ID}}$ to $\mathcal{A}$.

Challenge Queries. When the adversary $\mathcal{A}$ submits $(\texttt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$ to the challenger, $\mathcal{B}_1$ first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and runs $\mathsf{CEnc}(S, p) \to (\mathbf{c}, d_3')$. Then it runs the algorithm in Lemma 17 on input $(g^{\pi_1(\mathbf{B})}, g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ 0\end{pmatrix}})$ to obtain $g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_i\\ \hat{s}_i\\ 0\end{pmatrix}}$ for $i \in [0, d_3']$ where $(\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}, \hat{s}_i \xleftarrow{\$} \mathbb{Z}_p)$ if $\hat{t} \neq 0$ and $(\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}, \hat{s}_i = 0)$ if $\hat{t} = 0$. Finally, it computes $\left(g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})}, e\left(g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_0\\ \hat{s}_0\\ 0\end{pmatrix}}, h^{\boldsymbol{\alpha}}\right)\cdot \mathsf{M}_{\mathsf{coin}}\right)$ and returns it to $\mathcal{A}$, where $\mathbf{S}$ and $\hat{\mathbf{S}}$ are defined as Equation (24). Note that $g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})}$ can be efficiently computable from $\{g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_i\\ \hat{s}_i\\ 0\end{pmatrix}}\}_{i\in[0,d_3']}$ and $\{\mathbf{W}_j\}_{j\in[d_1]}$ since $\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})$ only contains linear combination of $\mathbf{B}\begin{pmatrix}\mathbf{s}_i\\ \hat{s}_i\\ 0\end{pmatrix}$ and $\mathbf{W}_i\mathbf{B}\begin{pmatrix}\mathbf{s}_j\\ \hat{s}_j\\ 0\end{pmatrix}$.

We claim that it is properly distributed normal ciphertext if $\hat{t} = 0$ and semi-functional ciphertext of Type 1 if $\hat{t} \xleftarrow{\$} \mathbb{Z}_p^*$. The latter is trivial. To see the former, it suffices to notice that $e\left(g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_0\\ \hat{s}_0\\ 0\end{pmatrix}}, h^{\boldsymbol{\alpha}}\right) = e(g,h)^{\boldsymbol{\alpha}^\top\pi_1(\mathbf{B})\mathbf{s}_0}$ holds (since $\hat{s}_0 = 0$).

Guess. When $\mathcal{A}$ outputs coin', $\mathcal{B}_1$ outputs 1 if $\mathsf{coin}' = \mathsf{coin}$ and 0 otherwise.

$\mathcal{B}_1$ has properly simulated Game_0 if $\hat{t} = 0$ and Game_1 if $\hat{t} \xleftarrow{\$} \mathbb{Z}_p^*$. Hence, we may conclude that $|\mathsf{Adv}_0(\kappa) - \mathsf{Adv}_1(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_7}_{\mathcal{B}_1}(\kappa)$. □

Lemma 22. (Game_1 to $\mathsf{Game}_{2,1,1}$). *For any adversary $\mathcal{A}$, we have* $\mathsf{Adv}_1(\kappa) = \mathsf{Adv}_{2,1,1}(\kappa)$.

Proof. This is purely a conceptual change and thus $\mathcal{A}$'s advantage is not altered. To see this, let us consider a modified version of Game_1 in which we first choose $\boldsymbol{\alpha}' \xleftarrow{\$} \mathbb{Z}_p^{4\times 1}, \hat{\gamma}, \tilde{\gamma} \xleftarrow{\$} \mathbb{Z}_p$ and then set $\boldsymbol{\alpha} \in \mathbb{Z}_p^{4\times 1}$ as $\boldsymbol{\alpha} = \boldsymbol{\alpha}' + \hat{\gamma}\hat{\mathbf{f}} + \tilde{\gamma}\tilde{\mathbf{f}}$. Since $\boldsymbol{\alpha}$ is still uniformly distributed over $\mathbb{Z}_p^{4\times 1}$, the view of the adversary in the modified

game is the same as Game_1. Furthermore, we claim that the view of the adversary in the modified game is also the same as $\mathsf{Game}_{2,1,1}$. This can be seen by regarding $\boldsymbol{\alpha}'$ in the modified game as $\boldsymbol{\alpha}$ in $\mathsf{Game}_{2,1,1}$, $\hat{\gamma}$ as $\widehat{\mathsf{R}}_0(\mathsf{ID}|_0)$, and $\tilde{\gamma}$ as $\widetilde{\mathsf{R}}_0(\mathsf{ID}|_0)$. It is easy to see that the distribution of the private keys in the modified game is the same as that of $\mathsf{Game}_{2,1,1}$. As for the master public key and the challenge ciphertexts, we have

$$e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})} = e(g,h)^{(\boldsymbol{\alpha}'+\hat{\gamma}\cdot\hat{\mathbf{f}}+\tilde{\gamma}\cdot\tilde{\mathbf{f}})^\top\cdot\pi_1(\mathbf{B})} = e(g,h)^{\boldsymbol{\alpha}'^\top \pi_1(\mathbf{B})}$$

and

$$\begin{aligned} e\big(g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_0\\ \hat{s}_0\\ 0\end{pmatrix}}, h^{\boldsymbol{\alpha}}\big) &= e(g,h)^{(\pi_1(\mathbf{B})\mathbf{s}_0+\hat{s}_0\cdot\pi_2(\mathbf{B}))^\top(\boldsymbol{\alpha}'+\hat{\gamma}\cdot\hat{\mathbf{f}}+\tilde{\gamma}\cdot\tilde{\mathbf{f}})} \\ &= e(g,h)^{(\pi_1(\mathbf{B})\mathbf{s}_0+\hat{s}_0\cdot\pi_2(\mathbf{B}))^\top(\boldsymbol{\alpha}'+\hat{\gamma}\cdot\hat{\mathbf{f}})} \\ &= e\big(g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_0\\ \hat{s}_0\\ 0\end{pmatrix}}, h^{\boldsymbol{\alpha}'+\hat{\gamma}\cdot\hat{\mathbf{f}}}\big). \end{aligned}$$

In the above, we use the fact that $\pi_1(\mathbf{B})^\top\hat{\mathbf{f}} = \pi_1(\mathbf{B})^\top\tilde{\mathbf{f}} = \mathbf{0}$ and $\pi_2(\mathbf{B})^\top\tilde{\mathbf{f}} = 0$. These indicate that the view of $\mathcal{A}$ in the modified game is also the same as $\mathsf{Game}_{2,1,1}$. Thus, the lemma follows. □

Lemma 23. ($\mathsf{Game}_{2,i^\star,1}$ to $\mathsf{Game}_{2,i^\star,2}$). *For any $i^\star \in [1,\ell]$ and adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_2$ such that $|\mathsf{Adv}_{2,i^\star,1}(\kappa) - \mathsf{Adv}_{2,i^\star,2}(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_{10}}_{\mathcal{B}_2}(\kappa)$ and $\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + (Q_c+Q_k)\cdot\mathsf{poly}(\kappa,\ell)$ where $\mathsf{poly}(\kappa,\ell)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. We construct $\mathcal{B}_2$ that attacks Problem 10 from an adversary $\mathcal{A}$ who distinguishes the games. We note that these games only differ in the creation of challenge ciphertexts. $\mathcal{B}_2$ simulates the challenger for $\mathcal{A}$ as follows.

Setup. At the outset of the game, $\mathcal{B}_2$ is given the problem instance of the assumption $(g^{\mathbf{B}}, h^{\pi_1(\mathbf{Z})}, h^{\theta_{1,1}\hat{\mathbf{f}}+\theta_{2,1}\tilde{\mathbf{f}}}, h^{\theta_{1,2}\hat{\mathbf{f}}+\theta_{2,2}\tilde{\mathbf{f}}}, g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ \tilde{t}\end{pmatrix}})$ where either $(\hat{t} \xleftarrow{\$} \mathbb{Z}_p^*, \tilde{t}=0)$ or $(\hat{t}=0, \tilde{t} \xleftarrow{\$} \mathbb{Z}_p^*)$. Then, it runs $\mathsf{Param}(2\ell,p) \to d_1$, picks $\mathbf{W}_1,\ldots,\mathbf{W}_{d_1} \xleftarrow{\$} \mathbb{Z}_p^{4\times 4}$ and $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{Z}_p^{4\times 1}$. It also computes $g^{\pi_1(\mathbf{W}_j\mathbf{B})}$ and $h^{\pi_1(\mathbf{W}_j^\top\mathbf{Z})} = h^{\mathbf{W}_j^\top\pi_1(\mathbf{Z})}$ for $j\in[d_1]$. Note that the latter can be efficiently computed from $h^{\pi_1(\mathbf{Z})}$. Finally, it returns the public parameter $\mathsf{pp} = (g, g^{\pi_1(\mathbf{B})}, g^{\pi_1(\mathbf{W}_1\mathbf{B})},\ldots,g^{\pi_1(\mathbf{W}_{d_1}\mathbf{B})}, h, h^{\pi_1(\mathbf{Z})}, h^{\pi_1(\mathbf{W}_1^\top\mathbf{Z})},\ldots,h^{\pi_1(\mathbf{W}_{d_1}^\top\mathbf{Z})})$ and the master public key $\mathsf{mpk} = (\mathsf{pp}, e(g,h)^{\boldsymbol{\alpha}^\top\pi_1(\mathbf{B})})$ to $\mathcal{A}$. $\mathcal{B}_1$ also sets $\mathsf{msk} = \boldsymbol{\alpha}$ and flips a coin $\mathsf{coin} \xleftarrow{\$} \{0,1\}$.

Programming Random Functions. Throughout the game, $\mathcal{B}_2$ simulates random functions $\mathsf{R}_{i^\star-1}(\cdot): \{0,1\}^{i^\star-1} \to \mathbb{Z}_p$ and $\mathsf{R}'_{i^\star-1}(\cdot): \{0,1\}^{i^\star-1} \to \mathbb{Z}_p$ via lazy sampling. In the following, $\mathcal{B}_2$ will implicitly sets the functions $\widehat{\mathsf{R}}_{i^\star-1}(\cdot): \{0,1\}^{i^\star-1} \to \mathbb{Z}_p$ and $\widetilde{\mathsf{R}}_{i^\star-1}(\cdot): \{0,1\}^{i^\star-1} \to \mathbb{Z}_p$ as

$$\begin{pmatrix}\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})\\ \widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})\end{pmatrix} = \begin{pmatrix}\theta_{1,1} & \theta_{1,2}\\ \theta_{2,1} & \theta_{2,2}\end{pmatrix}\begin{pmatrix}\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})\\ \mathsf{R}'_{i^\star-1}(\mathsf{ID}|_{i^\star-1})\end{pmatrix}. \tag{27}$$

Since $\mathsf{R}_{i^\star-1}(\cdot)$ and $\mathsf{R}'_{i^\star-1}(\cdot)$ are random functions and $\begin{pmatrix}\theta_{1,1} & \theta_{1,2}\\ \theta_{2,1} & \theta_{2,2}\end{pmatrix}$ is an invertible matrix, $\widehat{\mathsf{R}}_{i^\star-1}(\cdot)$ and $\widetilde{\mathsf{R}}_{i^\star-1}(\cdot)$ defined as above are random functions as well.

Key Extraction Queries. When the adversary $\mathcal{A}$ submits $(\mathtt{Extraction}, 1, \mathsf{ID})$ to the challenger, $\mathcal{B}_2$ first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$, runs $\mathsf{KEnc}(j,p) \to (\mathbf{k}_j, d'_2)$, picks $\mathbf{r}_{j,1},\ldots,\mathbf{r}_{j,d'_2} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, and sets $\mathbf{R}_j = \left(\begin{pmatrix}\mathbf{r}_{j,1}\\ 0\\ 0\end{pmatrix},\cdots,\begin{pmatrix}\mathbf{r}_{j,d'_2}\\ 0\\ 0\end{pmatrix}\right) \in \mathbb{Z}_p^{4\times d'_2}$ for all $j\in S$. It also chooses random $\{\boldsymbol{\alpha}_j \in \mathbb{Z}_p^{4\times 1}\}_{j\in S}$ such that $\sum_{j\in S}\boldsymbol{\alpha}_j = \boldsymbol{\alpha}$. Then, $\mathcal{B}_2$ calls the random functions $\mathsf{R}_{i^\star-1}(\cdot)$ and $\mathsf{R}'_{i^\star-1}(\cdot)$ on input $\mathsf{ID}|_{i^\star-1}$ to obtain $\gamma = \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) \in \mathbb{Z}_p$ and $\gamma' = \mathsf{R}'_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) \in \mathbb{Z}_p$. It also picks random $\{\gamma_j \in \mathbb{Z}_p\}_{j\in S}$ and

$\{\gamma'_j \in \mathbb{Z}_p\}_{j\in S}$ subject to constraint that $\sum_{j\in S}\gamma_j = \gamma$ and $\sum_{j\in S}\gamma'_j = \gamma'$. Then it implicitly sets $\hat{\gamma}_j$ and $\tilde{\gamma}_j$ as $\hat{\gamma}_j = \theta_{1,1}\gamma_j + \theta_{1,2}\gamma'_j$ and $\tilde{\gamma}_j = \theta_{2,1}\gamma_j + \theta_{2,2}\gamma'_j$ and computes

$$\left(h^{\theta_{1,1}\hat{\mathbf{f}}+\theta_{2,1}\tilde{\mathbf{f}}}\right)^{\gamma_j} \cdot \left(h^{\theta_{1,2}\hat{\mathbf{f}}+\theta_{2,2}\tilde{\mathbf{f}}}\right)^{\gamma'_j} = h^{\hat{\gamma}_j\hat{\mathbf{f}}+\tilde{\gamma}_j\tilde{\mathbf{f}}}$$

for $j \in S$. Then, it computes

$$\mathsf{sk}_j := h^{\mathbf{k}_j(\boldsymbol{\alpha}_j,\mathbf{R}_j,\mathbb{W})} \cdot h^{\mathbf{k}_j(\hat{\gamma}_j\hat{\mathbf{f}}+\tilde{\gamma}_j\tilde{\mathbf{f}},\mathbf{0},\mathbb{W})} = h^{\mathbf{k}_j(\boldsymbol{\alpha}_j+\hat{\gamma}_j\hat{\mathbf{f}}+\tilde{\gamma}_j\tilde{\mathbf{f}}_j,\mathbf{R}_j,\mathbb{W})}$$

for all $j \in S$. Finally, it computes $\mathsf{sk}_{\mathsf{ID}}$ as Equation (10) and returns it to $\mathcal{A}$. We can see that $\{\hat{\gamma}_j\}_{j\in S}$ and $\{\tilde{\gamma}_j\}_{j\in S}$ are uniformly random subject to constraint that $\sum_{j\in S}\hat{\gamma}_j = \widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$ and $\sum_{j\in S}\tilde{\gamma}_j = \widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$. Therefore, $\mathcal{B}_2$ perfectly simulates a semi-functional private key of Type $(1, i^\star - 1)$.

Challenge Queries. When the adversary $\mathcal{A}$ submits $(\mathtt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$ to the challenger, $\mathcal{B}_2$ first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and runs $\mathsf{CEnc}(S, p) \to (\mathbf{c}, d'_3)$. It also calls $\mathsf{R}_{i^\star-1}(\cdot)$ and $\mathsf{R}'_{i^\star-1}(\cdot)$ on input $\mathsf{ID}_{i^\star-1}$. It can efficiently compute $h^{\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\hat{\mathbf{f}}+\widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\tilde{\mathbf{f}}}$ as $h^{\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\hat{\mathbf{f}}+\widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\tilde{\mathbf{f}}} = \left(h^{\theta_{1,1}\hat{\mathbf{f}}+\theta_{2,1}\tilde{\mathbf{f}}}\right)^{\mathsf{R}_{i^\star-1}(\mathsf{ID}_{i^\star-1})} \cdot \left(h^{\theta_{1,2}\hat{\mathbf{f}}+\theta_{2,2}\tilde{\mathbf{f}}}\right)^{\mathsf{R}'_{i^\star-1}(\mathsf{ID}_{i^\star-1})}$. Then, it proceeds as follows. There are two cases.

– If $\mathsf{ID}_{i^\star} = 0$, it runs the algorithm in Lemma 17 on input $(g^{\pi_1(\mathbf{B})}, g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{t}\\ \hat{t}\\ \tilde{t}\end{smallmatrix}\right)})$ to obtain $g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_i\\ \hat{s}_i\\ \tilde{s}_i\end{smallmatrix}\right)}$ for $i \in [0, d'_3]$ where $(\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}, \hat{s}_i \xleftarrow{\$} \mathbb{Z}_p, \tilde{s}_i = 0)$ if $(\hat{t} \neq 0, \tilde{t} = 0)$ and $(\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}, \hat{s}_i = 0, \tilde{s}_i \xleftarrow{\$} \mathbb{Z}_p)$ if $(\hat{t} = 0, \tilde{t} \neq 0)$. Finally, it computes $\left(g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}}+\tilde{\mathbf{S}},\mathbb{W})}, e\left(g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_0\\ \hat{s}_0\\ \tilde{s}_0\end{smallmatrix}\right)}, h^{\boldsymbol{\alpha}+\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\hat{\mathbf{f}}+\widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\tilde{\mathbf{f}}}\right) \cdot \mathsf{M}_{\mathsf{coin}}\right)$ and returns it to $\mathcal{A}$, where $\mathbf{S}$, $\hat{\mathbf{S}}$, and $\tilde{\mathbf{S}}$ are defined as Equation (24). Note that $g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}}+\tilde{\mathbf{S}},\mathbb{W})}$ can be efficiently computable from $\{g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_i\\ \hat{s}_i\\ \tilde{s}_i\end{smallmatrix}\right)}\}_{i\in[0,d'_3]}$ and $\{\mathbf{W}_j\}_{j\in[d_1]}$ since $\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}}+\tilde{\mathbf{S}},\mathbb{W})$ only contains linear combination of $\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_i\\ \hat{s}_i\\ \tilde{s}_i\end{smallmatrix}\right)$ and $\mathbf{W}_i\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_j\\ \hat{s}_j\\ \tilde{s}_j\end{smallmatrix}\right)$. Furthermore, we have that

$$\begin{aligned} e\left(g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_0\\ \hat{s}_0\\ 0\end{smallmatrix}\right)}, h^{\boldsymbol{\alpha}} \cdot h^{\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\hat{\mathbf{f}}+\widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\tilde{\mathbf{f}}}\right) &= e\left(g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_0\\ \hat{s}_0\\ 0\end{smallmatrix}\right)}, h^{\boldsymbol{\alpha}+\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\hat{\mathbf{f}}}\right) \qquad \text{and} \\ e\left(g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_0\\ 0\\ \tilde{s}_0\end{smallmatrix}\right)}, h^{\boldsymbol{\alpha}} \cdot h^{\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\hat{\mathbf{f}}+\widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\tilde{\mathbf{f}}}\right) &= e\left(g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_0\\ 0\\ \tilde{s}_0\end{smallmatrix}\right)}, h^{\boldsymbol{\alpha}+\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\hat{\mathbf{f}}}\right), \end{aligned}$$

because $\mathbf{e}_i\mathbf{B}^\top\hat{\mathbf{f}} = 0$ for $i \neq 3$ and $\mathbf{e}_i\mathbf{B}^\top\tilde{\mathbf{f}} = 0$ for $i \neq 4$. Therefore, the ciphertext returned to $\mathcal{A}$ is properly distributed semi-functional ciphertext of Type $(2, i^\star - 1)$ if $(\hat{t} \xleftarrow{\$} \mathbb{Z}_p^*, \tilde{t} = 0)$ and Type $(3, i^\star - 1)$ if $(\hat{t} = 0, \tilde{t} \xleftarrow{\$} \mathbb{Z}_p^*)$.

– If $\mathsf{ID}_{i^\star} = 1$, it picks $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{d'_3} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\hat{s}_0, \hat{s}_1, \ldots, \hat{s}_{d'_3} \xleftarrow{\$} \mathbb{Z}_p$, computes $\left(g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})}, e\left(g^{\mathbf{B}\left(\begin{smallmatrix}\mathbf{s}_0\\ \hat{s}_0\\ 0\end{smallmatrix}\right)}, h^{\boldsymbol{\alpha}} \cdot h^{\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\hat{\mathbf{f}}+\widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}_{i^\star-1})\tilde{\mathbf{f}}}\right) \cdot \mathsf{M}_{\mathsf{coin}}\right)$, and returns it to $\mathcal{A}$. Note that $\mathcal{B}_2$ can efficiently compute $g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})}$ from $g^{\mathbf{B}}$, $\{\mathbf{s}_i, \hat{s}_i\}_{i\in[0,d'_3]}$, and $\{\mathbf{W}_j\}_{j\in[d_1]}$. It is easy to see that $\mathcal{B}_2$ properly simulates semi-functional ciphertext of Type $(2, i^\star - 1)$.

Guess. When $\mathcal{A}$ outputs coin', $\mathcal{B}_2$ outputs 1 if $\mathsf{coin}' = \mathsf{coin}$ and 0 otherwise.

$\mathcal{B}_2$ has properly simulated $\mathsf{Game}_{2,i^\star,1}$ if $(\hat{t} \xleftarrow{\$} \mathbb{Z}_p^*, \tilde{t} = 0)$ and $\mathsf{Game}_{2,i^\star,2}$ if $(\hat{t} = 0, \tilde{t} \xleftarrow{\$} \mathbb{Z}_p^*)$. Hence, we may conclude that $|\mathsf{Adv}_{2,i^\star,2}(\kappa) - \mathsf{Adv}_{2,i^\star,3}(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_{10}}_{\mathcal{B}_2}(\kappa)$. □

Lemma 24. ($\mathsf{Game}_{2,i^\star,2}$ to $\mathsf{Game}_{2,i^\star,3}$). *For any $i^\star \in [1, \ell]$ and adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_3$ such that $|\mathsf{Adv}_{2,i^\star,2}(\kappa) - \mathsf{Adv}_{2,i^\star,3}(\kappa)| \leq \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B}_3,\Pi,(Q_c,Q_k),\mathbb{G}_1,\mathbb{G}_2}(\kappa)$ and $\mathsf{Time}(\mathcal{B}_3) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k) \cdot \mathsf{poly}(\kappa, \ell)$ where $\mathsf{poly}(\kappa, \ell)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. We construct an adversary $\mathcal{B}_3$ who breaks the (Q_c, Q_k)-MMH security of the underlying broadcast encoding from an adversary $\mathcal{A}$ who distinguishes $\mathsf{Game}_{2,i^\star,2}$ and $\mathsf{Game}_{2,i^\star,3}$. We note that these games differ in the creation of ciphertexts for ID such that $\mathsf{ID}_{i^\star} = 1$ and all private keys. In this proof, we first describe $\mathcal{B}_3$ and then analyse the view of $\mathcal{A}$ in the simulation.

Setup. At the outset of the game, $\mathcal{B}_3$ submits $\tau^\star = 2i^\star$ as its target and is given $\mathsf{params} = \big(g, g^{\pi_1(\mathbf{B})}, g^{\pi_3(\mathbf{B})}, \{g^{\pi_1(\mathbf{W}_i\mathbf{B})}\}_{i\in[d_1]}, \{g^{\pi_3(\mathbf{W}_i\mathbf{B})}\}_{i\in[d_1]}, h, h^{\mathbf{Z}}, \{h^{\pi_1(\mathbf{W}_i^\top\mathbf{Z})}\}_{i\in[d_1]}, \{h^{\pi_3(\mathbf{W}_i^\top\mathbf{Z})}\}_{i\in[d_1]}\big)$. It then picks $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{Z}_p^{4\times 1}$ and returns the public parameter $\mathsf{pp} = (g, g^{\pi_1(\mathbf{B})}, \{g^{\pi_1(\mathbf{W}_i\mathbf{B})}\}_{i\in[d_1]}, h, h^{\pi_1(\mathbf{Z})}, \{h^{\pi_1(\mathbf{W}_i^\top\mathbf{Z})}\}_{i\in[d_1]})$ and the master public key $\mathsf{mpk} = (\mathsf{pp}, e(g,h)^{\boldsymbol{\alpha}^\top\pi_1(\mathbf{B})})$. $\mathcal{B}_3$ keeps $\boldsymbol{\alpha}$ and params privately.

Programming Random Functions. Throughout the game, $\mathcal{B}_3$ simulates random functions $\mathsf{R}_{i^\star-1}(\cdot) : \{0,1\}^{i^\star-1} \to \mathbb{Z}_p$ and $\widetilde{\mathsf{R}}_{i^\star-1}(\cdot) : \{0,1\}^{i^\star-1} \to \mathbb{Z}_p$ via lazy sampling. $\mathcal{B}_3$ also maintains a list List of length $i^\star$ prefixes of identities for which a key extraction query was made. The list is set as $\mathsf{List} = \emptyset$ at the beginning of the game. $\mathcal{B}_3$ also flips a random coin $\mathsf{coin} \xleftarrow{\$} \{0,1\}$.

Key Extraction Queries. When the adversary $\mathcal{A}$ submits $(\texttt{Extraction}, 1, \mathsf{ID})$ to the challenger, $\mathcal{B}_3$ first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and runs $\mathsf{KEnc}(j,p) \to (\mathbf{k}_j, d_2')$ for all $j \in S$. It then calls random functions $\mathsf{R}_{i^\star-1}(\cdot)$ and $\widetilde{\mathsf{R}}_{i^\star-1}(\cdot)$ on input $\mathsf{ID}|_{i^\star-1}$ to obtain $\gamma = \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$ and $\tilde{\gamma} = \widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$. Then, it picks $\mathbf{r}'_{j,1}, \ldots, \mathbf{r}'_{j,d_2'} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and sets $\mathbf{R}'_j = \left(\begin{pmatrix} \mathbf{r}'_{j,1} \\ 0 \\ 0 \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{r}'_{j,d_2'} \\ 0 \\ 0 \end{pmatrix}\right)$ for all $j \in S$. It also picks random $\{\boldsymbol{\alpha}_j \in \mathbb{Z}_p^{4\times 1}\}_{j\in S}$, $\{\gamma_j \in \mathbb{Z}_p\}_{j\in S}$, and $\{\tilde{\gamma}_j \in \mathbb{Z}_p\}_{j\in S}$ subject to constraint that $\sum_{j\in S}\boldsymbol{\alpha}_j = \boldsymbol{\alpha}$, $\sum_{j\in S}\gamma_j = \gamma$, and $\sum_{j\in S}\tilde{\gamma}_j = \tilde{\gamma}$. Then, it computes $h^{\mathbf{k}_j(\boldsymbol{\alpha}_j, \mathbf{R}'_j, \mathbb{W})}$, $h^{\mathbf{k}_j(\gamma_j\hat{\mathbf{f}} + \tilde{\gamma}_j\tilde{\mathbf{f}}, \mathbf{0}, \mathbf{0})}$, and

$$h^{\mathbf{k}_j(\boldsymbol{\alpha}_j, \mathbf{R}'_j, \mathbb{W})} \cdot h^{\mathbf{k}_j(\gamma_j\hat{\mathbf{f}} + \tilde{\gamma}_j\tilde{\mathbf{f}}, \mathbf{0}, \mathbf{0})} = h^{\mathbf{k}_j(\boldsymbol{\alpha}_j + \gamma_j\hat{\mathbf{f}} + \tilde{\gamma}_j\tilde{\mathbf{f}}, \mathbf{R}'_j, \mathbb{W})}$$

for all $j \in S$. Note that these values can be efficiently computed from γ_j, $\tilde{\gamma}_j$, $\boldsymbol{\alpha}_j$, $\mathbf{R}'_j$, and $h^{\mathbf{Z}}$. Then, $\mathcal{B}_3$ sets $\{\mathsf{sk}_j\}_{j\in S}$ as follows. There are three cases to consider:

– In case of $\mathsf{ID}_{i^\star} = 1$ (or, equivalently, if $\tau^\star \notin S$), it sets

$$\mathsf{sk}_j = h^{\mathbf{k}_j(\boldsymbol{\alpha}_j + \gamma_j\hat{\mathbf{f}} + \tilde{\gamma}_j\tilde{\mathbf{f}}, \mathbf{R}'_j, \mathbb{W})}$$

for all $j \in S$. Here, $\mathcal{B}_3$ sets $\hat{\gamma}_j = \gamma_j$ for $j \in S$. Thus, $\{\hat{\gamma}_j\}_{j\in S}$ are random subject to constraint that $\sum_{j\in S}\hat{\gamma}_j = \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$.

– In case of $\mathsf{ID}_{i^\star} = 0$ (or, equivalently, if $\tau^\star \in S$) and $\mathsf{ID}|_{i^\star} \notin \mathsf{List}$, $\mathcal{B}_3$ first calls $\mathcal{O}^{\mathsf{MMH},\mathsf{K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ to obtain $h^{\mathbf{k}_{\tau^\star}(b\cdot\hat{\alpha}\hat{\mathbf{f}}, \mathbf{R}, \mathbb{W})}$ where $\mathbf{R}$ and $\hat{\alpha}$ are randomness chosen by the oracle. Then, $\mathcal{B}_3$ computes

$$h^{\mathbf{k}_{\tau^\star}(\boldsymbol{\alpha}_{\tau^\star} + \gamma_{\tau^\star}\hat{\mathbf{f}} + \tilde{\gamma}_{\tau^\star}\tilde{\mathbf{f}},\ \mathbf{R}'_{\tau^\star}, \mathbb{W})} \cdot h^{\mathbf{k}_{\tau^\star}(b\cdot\hat{\alpha}\hat{\mathbf{f}}, \mathbf{R}, \mathbb{W})} = h^{\mathbf{k}_{\tau^\star}(\boldsymbol{\alpha}_{\tau^\star} + (\gamma_{\tau^\star} + b\cdot\hat{\alpha})\hat{\mathbf{f}} + \tilde{\gamma}_{\tau^\star}\tilde{\mathbf{f}},\ \mathbf{R} + \mathbf{R}'_{\tau^\star}, \mathbb{W})}.$$

and sets

$$\mathsf{sk}_j = \begin{cases} h^{\mathbf{k}_j(\boldsymbol{\alpha}_j + \gamma_j\hat{\mathbf{f}} + \tilde{\gamma}_j\tilde{\mathbf{f}},\ \mathbf{R}'_j, \mathbb{W})} & \text{If } j \neq \tau^\star \\ h^{\mathbf{k}_{\tau^\star}(\boldsymbol{\alpha}_{\tau^\star} + (\gamma_{\tau^\star} + b\cdot\hat{\alpha})\hat{\mathbf{f}} + \tilde{\gamma}_{\tau^\star}\tilde{\mathbf{f}},\ \mathbf{R} + \mathbf{R}'_{\tau^\star}, \mathbb{W})} & \text{If } j = \tau^\star \end{cases} \tag{28}$$

for $j \in S$. Here, $\mathcal{B}_3$ implicitly sets $\hat{\gamma}_j = \gamma_j$ for $j \in S \backslash \{\tau^\star\}$ and $\hat{\gamma}_{\tau^\star} = \gamma_{\tau^\star} + b \cdot \hat{\alpha}$. Therefore, $\{\hat{\gamma}_j\}_{j\in S}$ are random subject to constraint that $\sum_{j\in S}\hat{\gamma}_j = \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) + b \cdot \hat{\alpha}$. Finally, $\mathcal{B}_3$ updates the list as $\mathsf{List} \leftarrow \mathsf{List} \cup \{\mathsf{ID}|_{i^\star}\}$.

– In case of $\mathsf{ID}_{i^\star} = 0$ and $\mathsf{ID}|_{i^\star} \in \mathsf{List}$, $\mathcal{A}$ must have made a key query for ID' such that $\mathsf{ID}'|_{i^\star} = \mathsf{ID}|_{i^\star}$. Since $\mathsf{ID}'_{i^\star} = \mathsf{ID}_{i^\star} = 0$, $\mathcal{B}_3$ must have called $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ to deal with the *first* such key extraction query made by $\mathcal{A}$. Let $h^{\mathbf{k}_{\tau^\star}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\mathbf{R},\mathbb{W})}$ be the answer to the oracle call. In this case, $\mathcal{B}_3$ does not call $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ and computes $\{\mathsf{sk}_j\}_{j\in S}$ as Equation (28) using $h^{\mathbf{k}_{\tau^\star}(b\cdot\hat{\alpha}\hat{\mathbf{f}},\mathbf{R},\mathbb{W})}$. As the above case, $\{\hat{\gamma}_j\}_{j\in S}$ are random subject to constraint that $\sum_{j\in S}\hat{\gamma}_j = \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) + b\cdot\hat{\alpha}$.

Finally, $\mathcal{B}_3$ computes $\mathsf{sk}_{\mathsf{ID}}$ as Equation (10) and returns it to $\mathcal{A}$.

Challenge Queries. When the adversary $\mathcal{A}$ submits $(\mathtt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$ to the challenger, $\mathcal{B}_3$ sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and runs $\mathsf{CEnc}(S, p) \to (\mathbf{c}, d'_3)$. Then, $\mathcal{B}_3$ proceeds as follows. There are two cases.

– If $\mathsf{ID}_{i^\star} = 1$ (or, equivalently, if $\tau^\star \notin S$), it submits S to its oracle $\mathcal{O}^{\mathsf{MMH,C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot)$ to obtain $g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})}$. It then calls $\mathsf{R}_{i^\star-1}(\cdot)$ on input $\mathsf{ID}|_{i^\star-1}$ and returns $\mathsf{CT} = (g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})}, e\big(g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_0\\ \hat{s}_0\\ 0\end{pmatrix}}, h^{\boldsymbol{\alpha}+\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})\cdot\hat{\mathbf{f}}}\big)\cdot\mathsf{M}_{\mathsf{coin}})$ to $\mathcal{A}$. The latter part of CT can be efficiently computable because $g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_0\\ \hat{s}_0\\ 0\end{pmatrix}}$ is included in $g^{\mathbf{c}_{\mathbf{B}}(\mathbf{S}+\hat{\mathbf{S}},\mathbb{W})}$ (by the regularity of the broadcast encoding scheme) and $\mathcal{B}_3$ knows $\boldsymbol{\alpha}$ and $h^{\mathbf{Z}}$.

– If $\mathsf{ID}_{i^\star} = 0$ (or, equivalently, if $\tau^\star \in S$), it picks $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{d'_3} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\tilde{s}_0, \tilde{s}_1, \ldots, \tilde{s}_{d'_3} \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_i\\ 0\\ \tilde{s}_i\end{pmatrix}} = g^{\pi_1(\mathbf{B})\mathbf{s}_i}\cdot g^{\tilde{s}_i\cdot\pi_3(\mathbf{B})} \quad \text{and} \quad g^{\mathbf{W}_j\mathbf{B}\begin{pmatrix}\mathbf{s}_i\\ 0\\ \tilde{s}_i\end{pmatrix}} = g^{\pi_1(\mathbf{W}_j\mathbf{B})\mathbf{s}_i}\cdot g^{\tilde{s}_i\cdot\pi_3(\mathbf{W}_j\mathbf{B})}$$

for $i \in [0, d'_3]$ and $j \in [d_1]$. Finally, it computes $\mathsf{CT} = \big(g^{\mathbf{c}(\mathbf{S}+\tilde{\mathbf{S}},\mathbb{W})}, e\big(g^{\mathbf{B}\begin{pmatrix}\mathbf{s}_0\\ 0\\ \tilde{s}_0\end{pmatrix}}, h^{\boldsymbol{\alpha}+\tilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})\cdot\tilde{\mathbf{f}}}\big)\cdot \mathsf{M}_{\mathsf{coin}}\big)$ and returns it to $\mathcal{A}$. Note that $\mathcal{B}_3$ can efficiently compute $g^{\mathbf{c}(\mathbf{S}+\tilde{\mathbf{S}},\mathbb{W})}$ because $\mathbf{c}(\mathbf{S}+\tilde{\mathbf{S}},\mathbb{W})$ only contains linear combination of $\mathbf{B}\begin{pmatrix}\mathbf{s}_i\\ 0\\ \tilde{s}_i\end{pmatrix}$ and $\mathbf{W}_i\mathbf{B}\begin{pmatrix}\mathbf{s}_j\\ 0\\ \tilde{s}_j\end{pmatrix}$.

Guess. When $\mathcal{A}$ outputs coin', $\mathcal{B}_3$ outputs 1 if $\mathsf{coin}' = \mathsf{coin}$ and 0 otherwise.

Analysis. We claim that $\mathcal{A}$'s view corresponds to that of $\mathsf{Game}_{2,i^\star,2}$ if $b = 0$ (i.e., $\mathcal{B}_3$ is equipped with oracle $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},0}(\cdot)$) and $\mathsf{Game}_{2,i^\star,3}$ if $b = 1$ ($\mathcal{B}_3$ is equipped with $\mathcal{O}^{\mathsf{MMH,K}}_{\tau^\star,\mathbf{Z},\mathbb{W},1}(\cdot)$). In the case of $b = 0$, it is easily seen that $\mathcal{B}_3$ simulates $\mathsf{Game}_{2,i^\star,2}$ with $\widehat{\mathsf{R}}_{i^\star-1}(\cdot) : \{0,1\}^{i^\star-1} \to \mathbb{Z}_p$ being $\widehat{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) = \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$. Therefore $\mathcal{B}_3$ correctly simulates $\mathsf{Game}_{2,i^\star,2}$. On the other hand, in the case of $b = 1$, one can see that $\mathcal{B}_3$ simulates $\mathsf{Game}_{2,i^\star,3}$ with $\widehat{\mathsf{R}}_{i^\star}(\cdot) : \{0,1\}^{i^\star} \to \mathbb{Z}_p$ being

$$\widehat{\mathsf{R}}_{i^\star}(\mathsf{ID}|_{i^\star}) = \begin{cases}\mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) & \text{if } \mathsf{ID}_{i^\star} = 1\\ \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) + \hat{\alpha} & \text{if } \mathsf{ID}_{i^\star} = 0\end{cases}$$

where $\hat{\alpha}$ *is freshly chosen for every distinct* $\mathsf{ID}|_{i^\star}$. Since $\mathsf{R}_{i^\star-1}(\cdot)$ is a random function, $\widehat{\mathsf{R}}_{i^\star}(\cdot)$ defined above is also a random function. Therefore, $\mathcal{B}_3$ correctly simulates $\mathsf{Game}_{2,i^\star,3}$ in this case. Hence, we may conclude that $|\mathsf{Adv}_{2,i^\star,2}(\kappa) - \mathsf{Adv}_{2,i^\star,3}(\kappa)| \le \mathsf{Adv}^{\mathsf{MMH}}_{\mathcal{B}_3,\Pi,(Q_c,Q_k),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa)$. □

Lemma 25. ($\mathsf{Game}_{2,i^\star,3}$ to $\mathsf{Game}_{2,i^\star,4}$). *For any $i^\star \in [1,\ell]$ and adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_4$ such that $|\mathsf{Adv}_{2,i^\star,3}(\kappa) - \mathsf{Adv}_{2,i^\star,4}(\kappa)| \le \mathsf{Adv}^{\mathsf{MMH}'}_{\mathcal{B}_4,\Pi,(Q_c,Q_k),(\mathbb{G}_1,\mathbb{G}_2)}(\kappa)$ and $\mathsf{Time}(\mathcal{B}_4) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k)\cdot\mathsf{poly}(\kappa,\ell)$ where $\mathsf{poly}(\kappa,\ell)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. We construct an adversary $\mathcal{B}_4$ that breaks (Q_c, Q_k)-MMH' security of the encoding from an adversary $\mathcal{A}$ who distinguishes the games. The lemma can be shown analogously to Lemma 24. We only highlight the main difference.

- $\mathcal{B}_4$ sets $\tau^\star = 2i^\star - 1$ instead of $\tau^\star = 2i^\star$.
- $\mathcal{B}_4$ simulates $\widehat{\mathsf{R}}_{i^\star}(\cdot)$ and $\mathsf{R}_{i^\star-1}(\cdot)$ throughout the game and use these functions to create challenge ciphertexts and private keys. It will simulate $\mathsf{Game}_{2,i^\star,3}$ with $\widetilde{\mathsf{R}}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) = \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1})$ or $\mathsf{Game}_{2,i^\star,4}$ with

$$\widetilde{\mathsf{R}}_{i^\star}(\mathsf{ID}|_{i^\star}) = \begin{cases} \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) & \text{if } \mathsf{ID}_{i^\star} = 0 \\ \mathsf{R}_{i^\star-1}(\mathsf{ID}|_{i^\star-1}) + \tilde{\alpha} & \text{if } \mathsf{ID}_{i^\star} = 1 \end{cases}$$

  where $\tilde{\alpha}$ is the randomness chosen by $\mathcal{O}^{\mathsf{MMH}',\mathsf{K}}_{\tau^\star,\mathbf{Z},\mathbb{W},1}(\cdot)$. $\tilde{\alpha}$ is freshly chosen for every distinct $\mathsf{ID}|_{i^\star}$.
- $\mathcal{B}_4$ computes the challenge ciphertext by itself if $\mathsf{ID}_{i^\star} = 1$. Otherwise, it makes oracle call to $\mathcal{O}^{\mathsf{MMH}',\mathsf{C}}_{\tau^\star,\mathbf{B},\mathbb{W}}(\cdot)$ and creates the ciphertext using the answer from the oracle.
- $\mathcal{B}_4$ maintains a List of length $i^\star$ prefixes of identities for which key extraction was made. For key extraction query made by $\mathcal{A}$, $\mathcal{B}_4$ generates private key by itself if $\mathsf{ID}_{i^\star} = 0$. If $\mathsf{ID}_{i^\star} = 1$ and $\mathsf{ID}|_{i^\star} \notin \mathsf{List}$, it calls $\mathcal{O}^{\mathsf{MMH}',\mathsf{K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ and creates the private key using the answer. Otherwise, $\mathcal{A}$ must have queried private key for ID' such that $\mathsf{ID}'|_{i^\star} = \mathsf{ID}|_{i^\star}$. $\mathcal{B}_4$ must have called $\mathcal{O}^{\mathsf{MMH}',\mathsf{K}}_{\tau^\star,\mathbf{Z},\mathbb{W},b}(\cdot)$ to deal with the *first* such key extraction query made by $\mathcal{A}$. $\mathcal{B}_4$ creates the private key using the answer to the query.

□

Lemma 26. ($\mathsf{Game}_{2,i^\star,4}$ to $\mathsf{Game}_{2,i^\star+1,1}$). *For any $i^\star \in [1,\ell]$ and adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_5$ such that $|\mathsf{Adv}_{2,i^\star,4}(\kappa) - \mathsf{Adv}_{2,i^\star+1,1}(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_{10}}_{\mathcal{B}_5}(\kappa)$ and $\mathsf{Time}(\mathcal{B}_5) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k) \cdot \mathsf{poly}(\kappa,\ell)$ where $\mathsf{poly}(\kappa,\ell)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. The proof is the same as that of Lemma 23 except that we replace $\mathsf{R}_{i^\star-1}$, $\mathsf{R}'_{i^\star-1}$, $\widehat{\mathsf{R}}_{i^\star-1}$, and $\widetilde{\mathsf{R}}_{i^\star-1}$ with $\mathsf{R}_{i^\star}$, $\mathsf{R}'_{i^\star}$, $\widehat{\mathsf{R}}_{i^\star}$, and $\widetilde{\mathsf{R}}_{i^\star}$. □

Lemma 27. ($\mathsf{Game}_{2,\ell+1,1}$ to Game_3). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_6$ such that $|\mathsf{Adv}_{2,\ell+1,1}(\kappa) - \mathsf{Adv}_3(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_{11}}_{\mathcal{B}_6}(\kappa)$ and $\mathsf{Time}(\mathcal{B}_6) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k) \cdot \mathsf{poly}(\kappa,\ell)$ where $\mathsf{poly}(\kappa,\ell)$ is independent of $\mathsf{Time}(\mathcal{A})$.*

Proof. We construct an adversary $\mathcal{B}_6$ who attacks Problem 11 from an adversary $\mathcal{A}$ who distinguishes the games. $\mathcal{B}_6$ is given the problem instance $(g, h, g^x, g^y, h^z, e(g,h)^{xyz+\gamma})$ where either $\gamma = 0$ or $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$ and proceeds as follows.

Setup. $\mathcal{B}_6$ runs $\mathsf{Param}(2\ell, p) \to d_1$ and picks $\mathbf{B} \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$, $\mathbb{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_{d_1}) \xleftarrow{\$} (\mathbb{Z}_p^{4\times 4})^{d_1}$ and a random full-rank diagonal matrix $\mathbf{D} \in \mathbb{Z}_p^{4\times 4}$ with the entries $(3,3)$ and $(4,4)$ being 1. Finally, it sets $\mathbf{Z} = \mathbf{B}^{-\top}\mathbf{D}$ and returns the public parameter $\mathsf{pp} = (g, g^{\pi_1(\mathbf{B})}, g^{\pi_1(\mathbf{W}_1\mathbf{B})}, \ldots, g^{\pi_1(\mathbf{W}_{d_1}\mathbf{B})}, h, h^{\pi_1(\mathbf{Z})}, h^{\pi_1(\mathbf{W}_1^\top\mathbf{Z})}, \ldots, h^{\pi_1(\mathbf{W}_{d_1}^\top\mathbf{Z})})$ and the master public key $\mathsf{mpk} = (\mathsf{pp}, e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})})$ to $\mathcal{A}$. $\mathcal{B}_6$ also flips a coin $\mathsf{coin} \xleftarrow{\$} \{0,1\}$.

Programming Random Functions. Throughout the game, $\mathcal{B}_6$ simulates random functions $\mathsf{R}_\ell(\cdot) : \{0,1\}^\ell \to \mathbb{Z}_p$ and $\widetilde{\mathsf{R}}_\ell(\cdot) : \{0,1\}^\ell \to \mathbb{Z}_p$ via lazy sampling. In the following, $\mathcal{B}_6$ will implicitly set $\widehat{\mathsf{R}}_\ell(\cdot) : \{0,1\}^\ell \to \mathbb{Z}_p$ as

$$\widehat{\mathsf{R}}_\ell(\mathsf{ID}) = \begin{cases} \mathsf{R}_\ell(\mathsf{ID}) & \text{if } (\texttt{Extraction}, 1, \mathsf{ID}) \text{ is queried} \\ \mathsf{R}_\ell(\mathsf{ID}) + yz & \text{if } (\texttt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}) \text{ is queried for some } \mathsf{M}_0, \mathsf{M}_1. \end{cases}$$

Since $(\texttt{Extraction}, 1, \mathsf{ID})$ and $(\texttt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$ are never queried for the same ID (by the restriction posed on the adversary), $\widehat{\mathsf{R}}_\ell(\cdot)$ is well-defined. Furthermore, since $\mathsf{R}_\ell(\cdot)$ is a random function, $\widehat{\mathsf{R}}_\ell(\cdot)$ is also a random function.

Key Extraction Queries. When the adversary $\mathcal{A}$ submits $(\mathtt{Extraction}, 1, \mathsf{ID})$ to the challenger, $\mathcal{B}_6$ calls $\mathsf{R}_\ell(\cdot)$ and $\widetilde{\mathsf{R}}_\ell(\cdot)$ on input ID to obtain $\hat{\gamma} = \widehat{\mathsf{R}}_\ell(\mathsf{ID}) = \mathsf{R}_\ell(\mathsf{ID})$ and $\tilde{\gamma} = \widetilde{\mathsf{R}}_\ell(\mathsf{ID})$. Then it sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ where $\mathsf{ID}_i \in \{0,1\}$ is the i-th bit of $\mathsf{ID} \in \{0,1\}^\ell$. Then it runs $\mathsf{KEnc}(j, p) \to (\mathbf{k}_j, d_2')$, picks $\mathbf{r}_{j,1}, \ldots, \mathbf{r}_{j,d_2'} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, and sets $\mathbf{R}_j = \left(\begin{pmatrix} \mathbf{r}_{j,1} \\ 0 \\ 0 \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{r}_{j,d_2'} \\ 0 \\ 0 \end{pmatrix} \right) \in \mathbb{Z}_p^{4\times d_2'}$ for all $j \in S$. It also picks random $\{\boldsymbol{\alpha}_j\}_{j\in S}$, $\{\hat{\gamma}_j\}_{j\in S}$, and $\{\tilde{\gamma}_j\}_{j\in S}$ subject to constraints that $\sum_{j\in S} \boldsymbol{\alpha}_j = \boldsymbol{\alpha}$, $\sum_{j\in S} \hat{\gamma}_j = \hat{\gamma}$, and $\sum_{j\in S} \tilde{\gamma}_j = \tilde{\gamma}$. Next, it computes $\mathbf{k}_j(\boldsymbol{\alpha}_j + \hat{\gamma}_j \hat{\mathbf{f}} + \tilde{\gamma}_j \tilde{\mathbf{f}}, \mathbf{R}_j, \mathbb{W})$ and sets $\mathsf{sk}_j = h^{\mathbf{k}_j(\boldsymbol{\alpha}_j + \hat{\gamma}_j \hat{\mathbf{f}} + \tilde{\gamma}_j \tilde{\mathbf{f}}, \mathbf{R}_j, \mathbb{W})}$ for all $j \in S$. Finally, $\mathcal{B}_6$ computes $\mathsf{sk}_{\mathsf{ID}}$ from sk_j as Equation (10) and returns it to $\mathcal{A}$.

Challenge Queries. When the adversary $\mathcal{A}$ makes challenge query $(\mathtt{Challenge}, 1, \mathsf{ID}, \mathsf{M}_0, \mathsf{M}_1)$, $\mathcal{B}_6$ proceeds as follows. $\mathcal{B}_6$ first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and computes $\mathsf{CEnc}(S, p) \to (\mathbf{c}, d_3')$. Then, it picks $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{d_3'} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\hat{s}_1, \ldots, \hat{s}_{d_3'} \xleftarrow{\$} \mathbb{Z}_p$ and runs the algorithm in Lemma 18 on input $(g, h, g^x, g^y, h^z, e(g,h)^{xyz+\gamma})$ to obtain $(g^{\hat{x}}, e(g,h)^{\hat{x}yz+\hat{\gamma}})$ where $\hat{\gamma} \xleftarrow{\$} \mathbb{Z}_p$ if $\gamma \neq 0$ and $\hat{\gamma} = 0$ if $\gamma = 0$. Then, it implicitly sets $\hat{s}_0 = \hat{x}$ and computes

$$g^{\mathbf{B}\begin{pmatrix} \mathbf{s}_0 \\ \hat{s}_0 \\ 0 \end{pmatrix}} = g^{\pi_1(\mathbf{B})\mathbf{s}_0} \cdot (g^{\hat{x}})^{\pi_2(\mathbf{B})}$$

and $g^{\mathbf{B}\begin{pmatrix} \mathbf{s}_i \\ \hat{s}_i \\ 0 \end{pmatrix}}$ for $i \in [1, d_3']$. Then, it computes $g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}}, \mathbb{W})}$ from $\{g^{\mathbf{B}\begin{pmatrix} \mathbf{s}_i \\ \hat{s}_i \\ 0 \end{pmatrix}}\}_{i\in[0,d_3']}$ and $\mathbb{W}$, where $\mathbf{S}$ and $\hat{\mathbf{S}}$ are defined as Equation (24). Finally, it returns the challenge ciphertext

$$\mathsf{CT} = \left(g^{\mathbf{c}(\mathbf{S}+\hat{\mathbf{S}}, \mathbb{W})}, \quad e(g^{\mathbf{B}\begin{pmatrix} \mathbf{s}_0 \\ \hat{s}_0 \\ 0 \end{pmatrix}}, h^{\boldsymbol{\alpha}}) \cdot e(g^{\hat{x}}, h^{\mathsf{R}_\ell(\mathsf{ID})}) \cdot e(g,h)^{\hat{x}yz+\hat{\gamma}} \cdot \mathsf{M}_{\mathsf{coin}} \right)$$

to $\mathcal{A}$. We have that

$$\begin{aligned} e(g^{\mathbf{B}\begin{pmatrix} \mathbf{s}_0 \\ \hat{s}_0 \\ 0 \end{pmatrix}}, h^{\boldsymbol{\alpha}}) \cdot e(g^{\hat{x}}, h^{\mathsf{R}_\ell(\mathsf{ID})}) \cdot e(g,h)^{\hat{x}yz+\hat{\gamma}} &= e(g^{\mathbf{B}\begin{pmatrix} \mathbf{s}_0 \\ \hat{s}_0 \\ 0 \end{pmatrix}}, h^{\boldsymbol{\alpha}}) \cdot e(g,h)^{\hat{s}_0 \widehat{\mathsf{R}}_\ell(\mathsf{ID})} \cdot e(g,h)^{\hat{\gamma}} \\ &= e(g^{\mathbf{B}\begin{pmatrix} \mathbf{s}_0 \\ \hat{s}_0 \\ 0 \end{pmatrix}}, h^{\boldsymbol{\alpha}}) \cdot e(g^{\mathbf{B}\begin{pmatrix} \mathbf{s}_0 \\ \hat{s}_0 \\ 0 \end{pmatrix}}, h^{\widehat{\mathsf{R}}_\ell(\mathsf{ID})\mathbf{Z}\mathbf{e}_3^\top}) \cdot e(g,h)^{\hat{\gamma}} \\ &= e(g^{\mathbf{B}\begin{pmatrix} \mathbf{s}_0 \\ \hat{s}_0 \\ 0 \end{pmatrix}}, h^{\boldsymbol{\alpha}+\widehat{\mathsf{R}}_\ell(\mathsf{ID})\hat{\mathbf{f}}}) \cdot e(g,h)^{\hat{\gamma}} \end{aligned}$$

where we use the fact $\mathbf{B}^\top \mathbf{Z} \mathbf{e}_3^\top = \mathbf{D}\mathbf{e}_3^\top = \mathbf{e}_3^\top$ in the second line. It can be seen that the challenge ciphertext is a properly distributed semi-functional ciphertext of Type $(2, \ell)$ if $\hat{\gamma} = 0$ and a random ciphertext if $\hat{\gamma} \xleftarrow{\$} \mathbb{Z}_p$.

Guess. When $\mathcal{A}$ outputs coin', $\mathcal{B}_6$ outputs 1 if $\mathsf{coin}' = \mathsf{coin}$ and 0 otherwise.

$\mathcal{B}_6$ has properly simulated $\mathsf{Game}_{2,\ell+1}$ if $\gamma = 0$ and Game_3 if $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$. Hence, we may conclude that $|\mathsf{Adv}_{2,\ell+1}(\kappa) - \mathsf{Adv}_3(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_{11}}_{\mathcal{B}_6}(\kappa)$. □

Extension to the Multi-Instance Case. As in the case of the constructions in composite order groups, we can easily extend the above proof to the case of the multi-instance case.

(*Proof Sketch of Theorem 4.*) To prove Theorem 4, we have to change the proof of Theorem 13 so that we simulate $\widehat{\mathsf{R}}_i^{(j)}(\cdot) : \{0,1\}^i \to \mathbb{Z}_p$ and $\widetilde{\mathsf{R}}_i^{(j)}(\cdot) : \{0,1\}^i \to \mathbb{Z}_p$ for $i \in [0, \ell]$ and each $j \in [\mu]$. Then, we consider Game_0 to Game_3 exactly the same as the single instance case except that semi-functional ciphertexts and semi-functional private keys for j-th instance are computed using $\widehat{\mathsf{R}}_i^{(j)}(\cdot)$ and $\widetilde{\mathsf{R}}_i^{(j)}(\cdot)$. The difference of advantage of $\mathcal{A}$ between them can be bounded accordingly. This can be done almost parallel to the case of composite-order groups and thus we omit the details. See the proof sketch of Theorem 2.

E CMH-Security of Π_{slp}

Here, we discuss the Q-CMH security of Π_{slp}. We first prove the following theorem that indicates that Π_{slp} is Q-CMH secure on prime-order groups ($\mathbb{G}_1$ and $\mathbb{G}_2$) assuming the DLIN assumption. Then, we discuss the security of Π_{slp} on composite-order groups.

Theorem 14. *For any Q_c and any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that* $\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{A},\Pi_{\mathsf{slp}},Q_c,(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + Q_c \cdot \mathsf{poly}(\kappa, n)$ *where* $\mathsf{poly}(\kappa, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We construct $\mathcal{B}$ that attacks the DLIN problem from $\mathcal{A}$ that violates the Q_c-CMH security of the above broadcast encoding. Given the problem instance $(g, g^{a_1}, g^{a_2}, h^{a_3}, g^{a_1 c_1}, g^{a_2 c_2}, T = h^{a_3(c_1+c_2)+\gamma})$ where $\gamma = 0$ or $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$, $\mathcal{B}$ proceeds as follows. ($\mathcal{B}$ disregards other terms in the problem instance.)

Setup of Parameters. At the beginning of the game, $\mathcal{A}$ outputs its target $\tau^\star \in [n]$. Let $\tau^\star = \tau_1^\star + (\tau_2^\star - 1) \cdot n_1$ where $\tau_1^\star \in [n_1]$ and $\tau_2^\star \in [n_2]$. $\mathcal{B}$ first picks $\theta_1, \ldots, \theta_{n_1}, \theta'_1, \ldots, \theta'_{n_1}, \phi, \phi' \xleftarrow{\$} \mathbb{Z}_p$. Then it implicitly sets $\hat{\mathbf{w}} = (\hat{u}_1, \ldots, \hat{u}_{n_1}, \hat{v}, \hat{u}'_1, \ldots, \hat{u}'_{n_1}, \hat{v}', \hat{w})$ as

$$\hat{u}_i := \begin{cases} \theta_i & \text{if } i \in [n_1] \backslash \{\tau_1^\star\} \\ \theta_{\tau_1^\star} + c_1 & \text{if } i = \tau_1^\star \end{cases}, \quad \hat{v} := -\tau_2^\star c_1 + \phi, \quad \hat{w} := c_1 + c_2$$

$$\hat{u}'_i := \begin{cases} \theta'_i & \text{if } i \in [n_1] \backslash \{\tau_1^\star\} \\ \theta'_{\tau_1^\star} + c_2 & \text{if } i = \tau_1^\star \end{cases}, \quad \hat{v}' := -\tau_2^\star c_2 + \phi'.$$

We can see that $\hat{\mathbf{w}}$ is correctly distributed.

Simulating $\mathcal{O}^{\mathsf{CMH},\mathsf{C}}_{\tau^\star,\hat{\mathbf{w}}}(\cdot)$. Given S such that $\tau^\star \notin S$, $\mathcal{B}$ first defines S_j for $j \in [n_2]$ as Equation (13) and picks $\hat{s}_0, \xi_1, \ldots, \xi_{n_2}, \xi'_1, \ldots, \xi'_{n_2} \xleftarrow{\$} \mathbb{Z}_p$. It then implicitly sets $\hat{t}_i$ and $\hat{t}'_i$ as

$$\hat{t}_i := a_1 \xi_i + \hat{s}_0/\nu_i, \quad \hat{t}'_i := a_2 \xi'_i + \hat{s}_0/\nu_i, \quad \text{where} \quad \nu_i := \begin{cases} \tau_2^\star & \text{if } \tau_1^\star \notin S_i \\ \tau_2^\star - i & \text{if } \tau_1^\star \in S_i. \end{cases}$$

for $i \in [n_2]$. Note that $\tau_1^\star \notin S_{\tau_2^\star}$, since $\tau^\star \notin S$. Furthermore, we also have $|\tau_2^\star - i| < n_2 < p$ and $1 \leq \tau_2^\star \leq n_2 < p$ for all $i \in [n_2]$. Thus, for all $i \in [n_2]$, ν_i defined as above are non zero. Therefore, $\hat{t}_i$ and $\hat{t}'_i$ are well-defined for all $i \in [n_2]$.

It can be seen that we have

$$\begin{aligned}
& \hat{s}_0 \hat{w} + \hat{t}_i \big(\hat{v} + i \sum_{j \in S_i} \hat{u}_j\big) + \hat{t}'_i \big(\hat{v}' + i \sum_{j \in S_i} \hat{u}'_j\big) \\
= \; & \hat{s}_0 (c_1 + c_2) + (a_1 \xi_i + \hat{s}_0/\nu_i) \cdot (\phi + i \sum_{j \in S_i} \theta_j - \nu_i c_1) + (a_2 \xi'_i + \hat{s}_0/\nu_i) \cdot (\phi' + i \sum_{j \in S_i} \theta'_j - \nu_i c_2) \\
= \; & \cancel{\hat{s}_0 c_1} + \cancel{\hat{s}_0 c_2} + \big(\xi_i (\phi + i \sum_{j \in S_i} \theta_j)\big) a_1 - (\xi_i \nu_i) a_1 c_1 + \big(\hat{s}_0 (\phi + i \sum_{j \in S_i} \theta_j)/\nu_i\big) - \cancel{\hat{s}_0 c_1} \\
& + \big(\xi'_i (\phi' + i \sum_{j \in S_i} \theta'_j)\big) a_2 - (\xi'_i \nu_i) a_2 c_2 + \big(\hat{s}_0 (\phi' + i \sum_{j \in S_i} \theta'_j)/\nu_i\big) - \cancel{\hat{s}_0 c_2} \\
= \; & \Phi_{i,0} + \Phi_{i,1} a_1 + \Phi_{i,2} a_1 c_1 + \Phi_{i,3} a_2 + \Phi_{i,4} a_2 c_2
\end{aligned}$$

where $\Phi_{i,0} = \hat{s}_0 (\phi + \phi' + i \sum_{j \in S_i} (\theta_j + \theta'_j))/\nu_i$, $\Phi_{i,1} = \xi_i (\phi + i \sum_{j \in S_i} \theta_j)$, $\Phi_{i,2} = -\xi_i \nu_i$, $\Phi_{i,3} = \xi'_i (\phi' + i \sum_{j \in S_i} \theta'_j)$, and $\Phi_{i,4} = -\xi'_i \nu_i$ in the above. $\mathcal{B}$ can efficiently compute

$$g^{\hat{s}_0 \hat{w} + \hat{t}_i \left(\hat{v} + i \sum_{j \in S_i} \hat{u}_j\right) + \hat{t}'_i \left(\hat{v}' + i \sum_{j \in S_i} \hat{u}'_j\right)} = g^{\Phi_{i,0}} \cdot (g^{a_1})^{\Phi_{i,1}} \cdot (g^{a_1 c_1})^{\Phi_{i,2}} \cdot (g^{a_2})^{\Phi_{i,3}} \cdot (g^{a_2 c_2})^{\Phi_{i,4}},$$

$$g^{\hat{t}_i} = (g^{a_1})^{\xi_i} \cdot g^{\hat{s}_0/\nu_i}, \quad \text{and} \quad g^{\hat{t}'_i} = (g^{a_2})^{\xi'_i} \cdot g^{\hat{s}_0/\nu_i}$$

for $i \in [n_2]$, since $\{\Phi_{i,j}\}_{i\in[n_2],j\in[5]}$ is known to $\mathcal{B}$. It computes $g^{\mathbf{c}(\hat{\mathbf{s}},\mathbf{w})}$ as above and returns it to $\mathcal{A}$.

Simulating $\mathcal{O}^{\mathsf{CMH},\mathsf{K}}_{\tau^\star,\hat{\mathbf{w}},b}(\cdot)$. When $\mathcal{A}$ calls the oracle, $\mathcal{B}$ implicitly sets $\hat{r} = a_3$ and $\hat{\alpha} = \gamma$ and returns

$$h^{\mathbf{k}(\hat{\alpha},\hat{\mathbf{r}},\hat{\mathbf{w}})} = \begin{pmatrix} h^{\hat{r}\hat{w}+\hat{\alpha}} = T, & h^{\hat{r}(\hat{v}+\tau_2^\star\hat{u}_{\tau_1^\star})} = (h^{a_3})^{\phi+\tau_2^\star\theta_{\tau_1^\star}}, & \{h^{\hat{r}\hat{u}_i} = (h^{a_3})^{\theta_i}\}_{i\in[n_1]\setminus\{\tau_1^\star\}}, \\ h^{\hat{r}} = h^{a_3}, & h^{\hat{r}(\hat{v}'+\tau_2^\star\hat{u}'_{\tau_1^\star})} = (h^{a_3})^{\phi'+\tau_2^\star\theta'_{\tau_1^\star}}, & \{h^{\hat{r}\hat{u}'_i} = (h^{a_3})^{\theta'_i}\}_{i\in[n_1]\setminus\{\tau_1^\star\}} \end{pmatrix}$$

to $\mathcal{A}$. It can be seen that $\mathcal{B}$ simulates $\mathcal{O}^{\mathsf{CMH},\mathsf{K}}_{\tau^\star,\hat{\mathbf{w}},0}(\cdot)$ if $\hat{\alpha} = \gamma = 0$ and $\mathcal{O}^{\mathsf{CMH},\mathsf{K}}_{\tau^\star,\hat{\mathbf{w}},1}(\cdot)$ if $\hat{\alpha} = \gamma \xleftarrow{\$} \mathbb{Z}_p^*$.

Guess. Finally, $\mathcal{B}$ outputs $\mathcal{A}$'s output as its guess. As we have seen, the game corresponds to the case of $b = 1$ if $\gamma \xleftarrow{\$} \mathbb{Z}_p^*$ and $b = 0$ if $\gamma = 0$. Thus, we may conclude that $\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{B},\Pi_{\mathsf{slp}},Q_c,(\mathbb{G}_1,\mathbb{G}_2)}(\kappa) \leq \mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}}(\kappa)$. □

We then introduce two problems in order to discuss the Q-CMH security of Π_{slp} on composite-order groups. We define advantage function $\mathsf{Adv}^{\mathsf{P}_{\mathrm{xx}}}_{\mathcal{A}}(\kappa)$ for Problem xx for any adversary $\mathcal{A}$ as follows. Let $(N, \mathbb{G}, \mathbb{G}_T, g_1, g_2, g_3, g_4, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{comp}}(1^\kappa)$ in the following.

Problem 13. (DLIN Problem on $\mathbb{G}_{p_2}$.) *We define advantage function for any adversary $\mathcal{A}$ as*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{P}_{13}}_{\mathcal{A},\mathbb{G}_{p_2}}(\kappa) = |\Pr[\mathcal{A}(g_1, g_2, g_3, g_4, g_2^{a_1}, g_2^{a_2}, g_2^{a_3}, g_2^{a_1 s_1}, g_2^{a_2 s_2}, T_0) \to 1] - \\ \Pr[\mathcal{A}(g_1, g_2, g_3, g_4, g_2^{a_1}, g_2^{a_2}, g_2^{a_3}, g_2^{a_1 s_1}, g_2^{a_2 s_2}, T_1) \to 1]|\end{aligned}$$

where $a_1, a_2, a_3, \gamma \xleftarrow{\$} \mathbb{Z}_N^*$, $s_1, s_2 \xleftarrow{\$} \mathbb{Z}_N$, $T_0 = g_2^{a_3(s_1+s_2)}$, and $T_1 = g_2^{a_3(s_1+s_2)+\gamma}$.

Problem 14. (DLIN Problem on $\mathbb{G}_{p_3}$.) *We define advantage function for any adversary $\mathcal{A}$ as*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{P}_{14}}_{\mathcal{A},\mathbb{G}_{p_3}}(\kappa) = |\Pr[\mathcal{A}(g_1, g_2, g_3, g_4, g_3^{a_1}, g_3^{a_2}, g_3^{a_3}, g_3^{a_1 s_1}, g_3^{a_2 s_2}, T_0) \to 1] - \\ \Pr[\mathcal{A}(g_1, g_2, g_3, g_4, g_3^{a_1}, g_3^{a_2}, g_3^{a_3}, g_3^{a_1 s_1}, g_3^{a_2 s_2}, T_1) \to 1]|\end{aligned}$$

where $a_1, a_2, a_3, \gamma \xleftarrow{\$} \mathbb{Z}_N^*$, $s_1, s_2 \xleftarrow{\$} \mathbb{Z}_N$, $T_0 = g_3^{a_3(s_1+s_2)}$, and $T_1 = g_3^{a_3(s_1+s_2)+\gamma}$.

By replacing g and h with g_2 (resp. g_3) in the proof of Theorem 14, we immediately obtain the security theorem for the Q-CMH security of Π_{slp} on $\mathbb{G}_{p_2}$ (resp. $\mathbb{G}_{p_3}$) as follows;

Theorem 15. *For $i \in \{2, 3\}$, any Q_c, and any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that* $\mathsf{Adv}^{\mathsf{CMH}}_{\mathcal{A},\Pi_{\mathsf{slp}},Q_c,\mathbb{G}_{p_i}}(\kappa) \leq \mathsf{Adv}^{\mathsf{P}_{\mathrm{xx}}}_{\mathcal{B}}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + Q_c \cdot \mathsf{poly}(\kappa, n)$ *where* $\mathsf{poly}(\kappa, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$. *In the above,* $\mathsf{P}_{\mathrm{xx}} = \mathsf{P}_{13}$ *if* $i = 2$ *and* $\mathsf{P}_{\mathrm{xx}} = \mathsf{P}_{14}$ *if* $i = 3$.

F Proof of Theorem 5

Proof. We assume that the adversary $\mathcal{A}$ makes at most single key extraction query for the same ID in the security game. This restriction can be easily removed by using a PRF. See remark in Section 7.

Sequence of Games. We define a sequence of games to establish the security of the IBE scheme. We write $\mathsf{Adv}_{\mathrm{xx}}(\kappa)$ to denote the advantage of $\mathcal{A}$ in $\mathsf{Game}_{\mathrm{xx}}$. In the following, we will pick random functions $\widehat{\mathsf{R}}_\ell : \{0,1\}^\ell \to \mathbb{Z}_p$ and $\widetilde{\mathsf{R}}_\ell : \{0,1\}^\ell \to \mathbb{Z}_p$ (via lazy sampling).

Game_0 : This is the real security game for anonymous IBE.

Game_1: In this game, all challenge ciphertexts are changed to a ciphertext for a random message. Namely, when the adversary submits $(\texttt{Challenge}, 1, (\mathsf{ID}_0, \mathsf{ID}_1), (\mathsf{M}_0, \mathsf{M}_1))$ to the challenger,

$$\mathsf{CT} = \left(C_1 = g^{\mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ 0\end{pmatrix}}, \quad C_2 = g^{\sum_{i\in S_{\mathsf{coin}}} \mathbf{W}_i \mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ 0\end{pmatrix}}, \quad C_3 = \mathsf{M}_{\mathsf{rand}} \right) \tag{29}$$

is returned. Here, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\hat{s} \xleftarrow{\$} \mathbb{Z}_p$, and $\mathsf{M}_{\mathsf{rand}} \xleftarrow{\$} \mathbb{G}_T$. $S_{\mathsf{coin}} \subset [2\ell]$ in the above is defined as $S_{\mathsf{coin}} = \{2i - \mathsf{ID}_{\mathsf{coin},i} | i \in [\ell]\}$ where $\mathsf{ID}_{\mathsf{coin},i}$ is the i-th bit of $\mathsf{ID}_{\mathsf{coin}}$. Note that while the distribution of the ciphertext is independent from $\mathsf{M}_{\mathsf{coin}}$, it still depends on $\mathsf{ID}_{\mathsf{coin}}$ in this game.

Furthermore, for key extraction query $(\texttt{Extraction}, 1, \mathsf{ID})$ made by $\mathcal{A}$, the challenger returns

$$\mathsf{sk}_{\mathsf{ID}} = \left(K_1 = h^{\boldsymbol{\alpha} + \sum_{i\in S} \mathbf{W}_i^\top \mathbf{Z}\begin{pmatrix}\mathbf{r}\\ 0\\ 0\end{pmatrix} + \widehat{\mathsf{R}}_\ell(\mathsf{ID})\hat{\mathbf{f}} + \widetilde{\mathsf{R}}_\ell(\mathsf{ID})\tilde{\mathbf{f}}}, \quad K_2 = g^{-\mathbf{Z}\begin{pmatrix}\mathbf{r}\\ 0\\ 0\end{pmatrix}} \right) \tag{30}$$

where $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\hat{\mathbf{f}} = \mathbf{Z}\mathbf{e}_3^\top$, $\tilde{\mathbf{f}} = \mathbf{Z}\mathbf{e}_4^\top$, and $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$. Namely, the challenger returns semi-functional key of Type $(1, \ell)$ (in terminology of the proof of Theorem 13).

Game_2: In this game, the challenge ciphertexts are further modified as

$$\mathsf{CT} = \left(C_1 = g^{\mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ \tilde{s}\end{pmatrix}}, \quad C_2 = g^{\sum_{i\in S_{\mathsf{coin}}} \mathbf{W}_i \mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ \tilde{s}\end{pmatrix}}, \quad C_3 = \mathsf{M}_{\mathsf{rand}} \right) \tag{31}$$

where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\hat{s}, \tilde{s} \xleftarrow{\$} \mathbb{Z}_p$, and $\mathsf{M}_{\mathsf{rand}} \xleftarrow{\$} \mathbb{G}_T$.

Game_3: This game is the same as Game_2 except that we change how to choose $\mathbf{W}_1, \ldots, \mathbf{W}_{2\ell}$. Here, we first pick $\mathbf{W}'_1, \mathbf{W}'_2, \ldots, \mathbf{W}'_{2\ell} \xleftarrow{\$} \mathbb{Z}_p^{4\times 4}$ and $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_p^{4\times 2}$. We denote the i-th column of $\mathbf{A}$ as $\mathbf{a}_i \in \mathbb{Z}_p^{4\times 1}$ for $i \in \{1, 2\}$. Then, we set another matrix $\mathbf{A}' \in \mathbb{Z}_p^{4\times 4}$ so that $\pi_1(\mathbf{A}') = \mathbf{0}$, $\pi_2(\mathbf{A}') = \mathbf{a}_1$, and $\pi_3(\mathbf{A}) = \mathbf{a}_2$. $\mathbf{W}_1, \ldots, \mathbf{W}_{2\ell}$ are set as

$$\mathbf{W}_1 = \mathbf{W}'_1 + \mathbf{B}\mathbf{A}'\mathbf{B}^{-1}, \ \mathbf{W}_2 = \mathbf{W}'_2 + \mathbf{B}\mathbf{A}'\mathbf{B}^{-1}, \text{ and } \mathbf{W}_i = \mathbf{W}'_i \text{ for } i \in [3, 2\ell].$$

Game_4: In this game, public parameter pp and all private keys are computed using $\mathbf{W}'_1, \ldots, \mathbf{W}'_{2\ell}$ instead of $\mathbf{W}_1, \ldots, \mathbf{W}_{2\ell}$. (The challenge ciphertexts are unchanged.) Namely, pp is set as $\mathsf{pp} = (g, g^{\pi_1(\mathbf{B})}, g^{\pi_1(\mathbf{W}'_1\mathbf{B})}, \ldots, g^{\pi_1(\mathbf{W}'_{2\ell}\mathbf{B})})$ and all private keys are created as

$$\mathsf{sk}_{\mathsf{ID}} = \left(K_1 = h^{\boldsymbol{\alpha} + \sum_{i\in S} \mathbf{W}_i'^\top \mathbf{Z}\begin{pmatrix}\mathbf{r}\\ 0\\ 0\end{pmatrix} + \widehat{\mathsf{R}}_\ell(\mathsf{ID})\hat{\mathbf{f}} + \widetilde{\mathsf{R}}_\ell(\mathsf{ID})\tilde{\mathbf{f}}}, \quad K_2 = g^{-\mathbf{Z}\begin{pmatrix}\mathbf{r}\\ 0\\ 0\end{pmatrix}} \right). \tag{32}$$

Game_5: In this game, we change all challenge ciphertexts to be uniformly random group elements. Namely, the challenger gives $\mathsf{CT} = (C_1, C_2, C_3)$ such that $C_1, C_2 \xleftarrow{\$} \mathbb{Z}_p^{4\times 1}$ and $C_3 \xleftarrow{\$} \mathbb{G}_T$ to $\mathcal{A}$ when $\mathcal{A}$ requests a challenge ciphertext.

Observe that we have $\mathsf{Adv}_5(\kappa) = 0$ since the view of $\mathcal{A}$ is independent from the value of coin in Game_5. We have that

$$\mathsf{Adv}^{\mathsf{AIBE}}_{\mathcal{A},\Phi_{\mathsf{anon}},(1,Q_c,Q_k)}(\kappa) = \Pr[\mathsf{Adv}_0] \leq \sum_{i\in[0,4]} |\Pr[\mathsf{Adv}_i] - \Pr[\mathsf{Adv}_{i+1}]| + \Pr[\mathsf{Adv}_5].$$

Therefore, we complete the proof by showing Lemma 28, 29, 30, 31, and 32 in the following. □

Lemma 28. (Game_0 to Game_1). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_1$ such that* $|\mathsf{Adv}_0(\kappa) - \mathsf{Adv}_1(\kappa)| \leq (8\ell+2)\mathsf{Adv}^{\mathsf{DLIN}}_{\mathcal{B}_1}(\kappa) + 12\ell/p$ *and* $\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A}) + (Q_c+Q_k)\cdot\mathsf{poly}(\kappa,\ell)$ *where* $\mathsf{poly}(\kappa,\ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. This can be shown by repeating the proof of Theorem 13 for the case of the underlying broadcast encoding is Π_{cc}. There are only two differences. The first one is that the simulator does not give $\mathsf{sp} = (h, h^{\pi_1(\mathbf{Z})}, h^{\pi_1(\mathbf{W}_1^\top \mathbf{Z})}, \ldots, g^{\pi_1(\mathbf{W}_{2\ell}^\top \mathbf{Z})})$ to $\mathcal{A}$. The second one is that the adversary submits two identities in the challenge query. The proof of the Theorem 13 can easily be modified accordingly. □

Lemma 29. (Game_1 to Game_2). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_2$ such that* $|\mathsf{Adv}_1(\kappa) - \mathsf{Adv}_2(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_9}_{\mathcal{B}_2}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + (Q_c + Q_k)\cdot\mathsf{poly}(\kappa,\ell)$ *where* $\mathsf{poly}(\kappa,\ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We construct an adversary $\mathcal{B}_2$ who attacks the Problem 9 from an adversary $\mathcal{A}$ who distinguishes the games. The proof of the lemma is similar to that of Lemma 23.

Setup of Parameters. At the beginning of the game, $\mathcal{B}_2$ is given the problem instance $(g, h, g^{\mathbf{B}}, h^{\pi_1(\mathbf{Z})}, h^{\mathbf{Z}(\theta_{1,1}\hat{\mathbf{f}}+\theta_{2,1}\tilde{\mathbf{f}})}, h^{\mathbf{Z}(\theta_{1,2}\hat{\mathbf{f}}+\theta_{2,2}\tilde{\mathbf{f}})}, g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ \tilde{t}\end{pmatrix}})$ where $\tilde{t} = 0$ or $\tilde{t} \xleftarrow{\$} \mathbb{Z}_p$. It first picks $\mathbf{W}_1, \ldots, \mathbf{W}_{2\ell} \xleftarrow{\$} \mathbb{Z}_p^{4\times 4}$ and $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{Z}_p^{4\times 1}$. Then, it computes the public parameter $\mathsf{pp} = (g, g^{\pi_1(\mathbf{B})}, g^{\pi_1(\mathbf{W}_1\mathbf{B})}, \ldots, g^{\pi_1(\mathbf{W}_{2\ell}\mathbf{B})})$ and the master public key $\mathsf{mpk} = e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})}$ and returns it to $\mathcal{A}$. $\mathcal{B}_2$ also flips a coin $\mathsf{coin} \xleftarrow{\$} \{0,1\}$.

Simulating Random Functions. Throughout the game, $\mathcal{B}_2$ simulates random functions $\mathsf{R}_\ell(\cdot) : \{0,1\}^\ell \to \mathbb{Z}_p$ and $\mathsf{R}'_\ell(\cdot) : \{0,1\}^\ell \to \mathbb{Z}_p$ via lazy sampling. It implicitly sets $\widehat{\mathsf{R}}_\ell(\mathsf{ID})$ and $\widetilde{\mathsf{R}}_\ell(\mathsf{ID})$ as Equation (27). Since $\mathsf{R}_\ell(\cdot)$ and $\mathsf{R}'_\ell(\cdot)$ are random functions, $\widehat{\mathsf{R}}_\ell(\cdot)$ and $\widetilde{\mathsf{R}}_\ell(\cdot)$ are random functions as well.

Challenge Queries. When the adversary $\mathcal{A}$ makes challenge query $(\mathtt{Challenge}, 1, \mathsf{ID}_0, \mathsf{ID}_1, \mathsf{M}_0, \mathsf{M}_1)$, $\mathcal{B}_2$ proceeds as follows. $\mathcal{B}_2$ first sets $S_{\mathsf{coin}} = \{2i - \mathsf{ID}_{\mathsf{coin},i} | i \in [\ell]\}$ where $\mathsf{ID}_{\mathsf{coin},i}$ is the i-th bit of $\mathsf{ID}_{\mathsf{coin}}$. Then, it runs the algorithm in Lemma 17 on input $(g^{\pi_1(\mathbf{B})}, g^{\pi_3(\mathbf{B})}, g^{\mathbf{B}\begin{pmatrix}\mathbf{t}\\ \hat{t}\\ \tilde{t}\end{pmatrix}})$ to obtain $g^{\mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ \tilde{s}\end{pmatrix}}$ where $(\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}, \hat{s} \xleftarrow{\$} \mathbb{Z}_p, \tilde{s} = 0)$ if $\tilde{t} = 0$ and $(\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}, \hat{s} \xleftarrow{\$} \mathbb{Z}_p^*, \tilde{s} \xleftarrow{\$} \mathbb{Z}_p)$ if $\tilde{t} \neq 0$. Then it picks $\mathsf{M}_{\mathsf{rand}} \xleftarrow{\$} \mathbb{G}_T$ and returns the challenge ciphertext $\mathsf{CT} = \left(C_1 = g^{\mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ \tilde{s}\end{pmatrix}}, C_2 = g^{\sum_{i\in S_{\mathsf{coin}}} \mathbf{W}_i \mathbf{B}\begin{pmatrix}\mathbf{s}\\ \hat{s}\\ \tilde{s}\end{pmatrix}}, C_3 = \mathsf{M}_{\mathsf{rand}},\right)$ to $\mathcal{A}$.

Key Extraction Queries. When the adversary $\mathcal{A}$ submits $(\mathtt{Extraction}, 1, \mathsf{ID})$ to the challenger, $\mathcal{B}_3$ first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ and $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$. Then it computes $h^{\widehat{\mathsf{R}}_\ell(\mathsf{ID}_\ell)\hat{\mathbf{f}} + \widetilde{\mathsf{R}}_\ell(\mathsf{ID}_\ell)\tilde{\mathbf{f}}} = \left(h^{\theta_{1,1}\hat{\mathbf{f}}+\theta_{2,1}\tilde{\mathbf{f}}}\right)^{\mathsf{R}_\ell(\mathsf{ID}_\ell)} \cdot \left(h^{\theta_{1,2}\hat{\mathbf{f}}+\theta_{2,2}\tilde{\mathbf{f}}}\right)^{\mathsf{R}'_\ell(\mathsf{ID}_\ell)}$. Finally, it computes the private key as

$$\mathsf{sk}_{\mathsf{ID}} = \left(K_1 = h^{\boldsymbol{\alpha} + \sum_{i\in S} \mathbf{W}_i^\top \pi_1(\mathbf{Z})\mathbf{r} + \widehat{\mathsf{R}}_\ell(\mathsf{ID})\hat{\mathbf{f}} + \widetilde{\mathsf{R}}_\ell(\mathsf{ID})\tilde{\mathbf{f}}}, \quad K_2 = g^{-\pi_1(\mathbf{Z})\mathbf{r}}\right)$$

and returns it to $\mathcal{A}$.

Guess. When $\mathcal{A}$ outputs coin', $\mathcal{B}_2$ outputs 1 if $\mathsf{coin}' = \mathsf{coin}$ and 0 otherwise.

$\mathcal{B}_2$ has properly simulated Game_1 if $\tilde{t} = 0$ and Game_2 if $\tilde{t} \xleftarrow{\$} \mathbb{Z}_p^*$. Hence, we may conclude that $|\mathsf{Adv}_1(\kappa) - \mathsf{Adv}_2(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_9}_{\mathcal{B}_2}(\kappa)$. □

Lemma 30. (Game_2 to Game_3). *For any adversary $\mathcal{A}$, we have* $\mathsf{Adv}_2 = \mathsf{Adv}_3$.

Proof. Since each $\mathbf{W}_i$ is uniformly random over $\mathbb{Z}_p^{4\times 4}$ in both games, the change from Game_2 to Game_3 is only conceptual. Therefore, the lemma follows. □

Lemma 31. (Game_3 to Game_4). *For any adversary $\mathcal{A}$, we have* $\mathsf{Adv}_3 = \mathsf{Adv}_4$.

Proof. We claim that the change from Game_3 to Game_4 is conceptual. At first, we have that

$$\pi_1(\mathbf{W}_i\mathbf{B}) = \pi_1(\mathbf{W}'_i\mathbf{B}) + \pi_1(\mathbf{B}\mathbf{A}') = \pi_1(\mathbf{W}'_i\mathbf{B}) + \mathbf{B}\cdot\pi_1(\mathbf{A}') = \pi_1(\mathbf{W}'_i\mathbf{B})$$

for $i \in \{1,2\}$ and $\mathbf{W}_i = \mathbf{W}'_i$ for $i \in [3, 2\ell]$. Therefore, pp is not altered by this change.

The private keys returned to $\mathcal{A}$ in Game_3 are in the form of $\mathsf{sk}_{\mathsf{ID}} = (K_1, K_2)$ where $K_2 = h^{-\pi_1(\mathbf{Z})\mathbf{r}}$ and $K_1 = h^{\boldsymbol{\alpha}+\sum_{i\in S}\pi_1(\mathbf{W}_i^\top\mathbf{Z})\mathbf{r}+\widehat{\mathsf{R}}_\ell(\mathsf{ID})\hat{\mathbf{f}}+\widetilde{\mathsf{R}}_\ell(\mathsf{ID})\tilde{\mathbf{f}}}$. We have that

$$\begin{aligned}
K_1 &= h^{\boldsymbol{\alpha}+\sum_{i\in S}\mathbf{W}_i^\top\mathbf{Z}\left(\begin{smallmatrix}\mathbf{r}\\0\\0\end{smallmatrix}\right)+\widehat{\mathsf{R}}_\ell(\mathsf{ID})\hat{\mathbf{f}}+\widetilde{\mathsf{R}}_\ell(\mathsf{ID})\tilde{\mathbf{f}}}\\
&= h^{\boldsymbol{\alpha}+((\sum_{i\in S}\mathbf{W}'_i)+\mathbf{B}\mathbf{A}'\mathbf{B}^{-1})^\top\mathbf{Z}\left(\begin{smallmatrix}\mathbf{r}\\0\\0\end{smallmatrix}\right)+\widehat{\mathsf{R}}_\ell(\mathsf{ID})\hat{\mathbf{f}}+\widetilde{\mathsf{R}}_\ell(\mathsf{ID})\tilde{\mathbf{f}}}\\
&= h^{\boldsymbol{\alpha}+\sum_{i\in S}\mathbf{W}'^\top_i\mathbf{Z}\left(\begin{smallmatrix}\mathbf{r}\\0\\0\end{smallmatrix}\right)+(\widehat{\mathsf{R}}_\ell(\mathsf{ID})+\pi_2(\mathbf{A}')^\top\pi_1(\mathbf{D})\mathbf{r})\hat{\mathbf{f}}+(\widetilde{\mathsf{R}}_\ell(\mathsf{ID})+\pi_3(\mathbf{A}')^\top\pi_1(\mathbf{D})\mathbf{r})\tilde{\mathbf{f}}}\\
&= h^{\boldsymbol{\alpha}+\sum_{i\in S}\mathbf{W}'^\top_i\mathbf{Z}\left(\begin{smallmatrix}\mathbf{r}\\0\\0\end{smallmatrix}\right)+\widehat{\mathsf{R}}'_\ell(\mathsf{ID})\hat{\mathbf{f}}+\widetilde{\mathsf{R}}'_\ell(\mathsf{ID})\tilde{\mathbf{f}}}.
\end{aligned}$$

In the second line above, we used the fact that $\big((1\in S)\wedge(2\notin S)\big)\vee\big((1\notin S)\wedge(2\in S)\big)$. In the third line above, we used the fact that

$$(\mathbf{B}\mathbf{A}'\mathbf{B}^{-1})^\top\mathbf{Z}\begin{pmatrix}\mathbf{r}\\0\\0\end{pmatrix} = \mathbf{B}^{-\top}\mathbf{A}'^\top\mathbf{D}\begin{pmatrix}\mathbf{r}\\0\\0\end{pmatrix} = \mathbf{B}^{-\top}\begin{pmatrix}\mathbf{0}\\\pi_2(\mathbf{A}')^\top\pi_1(\mathbf{D})\mathbf{r}\\\pi_3(\mathbf{A}')^\top\pi_1(\mathbf{D})\mathbf{r}\end{pmatrix} = \mathbf{Z}\begin{pmatrix}\mathbf{0}\\\pi_2(\mathbf{A}')^\top\pi_1(\mathbf{D})\mathbf{r}\\\pi_3(\mathbf{A}')^\top\pi_1(\mathbf{D})\mathbf{r}\end{pmatrix}.$$

In the fourth line, we defined two functions $\widehat{\mathsf{R}}'_\ell(\cdot):\{0,1\}^\ell\to\mathbb{Z}_p$ and $\widetilde{\mathsf{R}}'_\ell(\cdot):\{0,1\}^\ell\to\mathbb{Z}_p$ as

$$\begin{aligned}
\widehat{\mathsf{R}}'_\ell(\mathsf{ID}) &= \widehat{\mathsf{R}}_\ell(\mathsf{ID}) + \pi_2(\mathbf{A}')^\top\pi_1(\mathbf{D})\mathbf{r} \quad \text{and}\\
\widetilde{\mathsf{R}}'_\ell(\mathsf{ID}) &= \widetilde{\mathsf{R}}_\ell(\mathsf{ID}) + \pi_3(\mathbf{A}')^\top\pi_1(\mathbf{D})\mathbf{r}.
\end{aligned}$$

These functions are well-defined (throughout the game), because $\mathcal{A}$ makes key extraction query for the same ID at most once. Furthermore, these functions are random functions because $\widehat{\mathsf{R}}_\ell(\cdot)$ and $\widetilde{\mathsf{R}}_\ell(\cdot)$ are so. One can observe that the distribution of private keys is also the same as that in Game_4. The lemma follows readily. □

Lemma 32. (Game_4 to Game_5). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_3$ such that* $|\mathsf{Adv}_4(\kappa) - \mathsf{Adv}_5(\kappa)| \le \mathsf{Adv}^{\mathsf{P}_{12}}_{\mathcal{B}_3}(\kappa) + 1/p$ *and* $\mathsf{Time}(\mathcal{B}_3) \approx \mathsf{Time}(\mathcal{A}) + (Q_c+Q_k)\cdot\mathsf{poly}(\kappa,\ell)$ *where* $\mathsf{poly}(\kappa,\ell)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We show an adversary $\mathcal{B}_3$ that attacks the Problem 12 from $\mathcal{A}$ who distinguishes the games. Given the problem instance $(g, h, g^{\mathbf{X}})$ where either $\mathbf{X} \xleftarrow{\$} \mathsf{Rk}_2(\mathbb{Z}_p^{6\times6})$ or $\mathbf{X} \xleftarrow{\$} \mathsf{Rk}_6(\mathbb{Z}_p^{6\times6})$, it proceeds as follows. Let the upper two rows of $\mathbf{X}$ be $\mathbf{U} \in \mathbb{Z}_p^{2\times6}$ and lower four rows be $\mathbf{L} \in \mathbb{Z}_p^{4\times6}$. We assume that $\mathbf{U}$ is full-rank. This happens with probability at least $1-1/p$ regardless of $\mathbf{X} \xleftarrow{\$} \mathsf{Rk}_2(\mathbb{Z}_p^{6\times6})$ or $\mathbf{X} \xleftarrow{\$} \mathsf{Rk}_6(\mathbb{Z}_p^{6\times6})$. If $\mathbf{X} \xleftarrow{\$} \mathsf{Rk}_2(\mathbb{Z}_p^{6\times6})$, we have that there exists a random matrix $\mathbf{A} \in \mathbb{Z}_p^{4\times2}$ such that $\mathbf{L} = \mathbf{A}\mathbf{U}$.

Setup. At the outset of the game, $\mathcal{B}_3$ picks $\mathbf{B} \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$, $\mathbf{W}'_1, \ldots, \mathbf{W}'_{2\ell} \xleftarrow{\$} \mathbb{Z}_p^{4\times4}$, $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{Z}_p^{4\times1}$ and a random full-rank diagonal matrix $\mathbf{D} \in \mathbb{Z}_p^{4\times4}$ with the entries $(3,3)$ and $(4,4)$ being 1. It then sets $\mathbf{Z} = \mathbf{B}^{-\top}\mathbf{D}$ and returns $\mathsf{pp} = (g, g^{\pi_1(\mathbf{B})}, g^{\pi_1(\mathbf{W}'_1\mathbf{B})}, \ldots, g^{\pi_1(\mathbf{W}'_{2\ell}\mathbf{B})})$ and $\mathsf{mpk} = e(g,h)^{\boldsymbol{\alpha}^\top\pi_1(\mathbf{B})}$ to $\mathcal{A}$.

Simulating Random Functions. Throughout the game, $\mathcal{B}_3$ simulates random functions $\widehat{\mathsf{R}}_\ell(\cdot):\{0,1\}^\ell\to\mathbb{Z}_p$ and $\widetilde{\mathsf{R}}_\ell(\cdot):\{0,1\}^\ell\to\mathbb{Z}_p$ via lazy sampling.

Key Extraction Queries. When $\mathcal{A}$ submits $(\mathtt{Extraction}, 1, \mathsf{ID})$ to the challenger, $\mathcal{B}_3$ picks $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^{2\times1}$, computes $\mathsf{sk}_{\mathsf{ID}}$ as Equation (32) and returns it to $\mathcal{A}$. Note that $\mathsf{sk}_{\mathsf{ID}}$ can be efficiently computed using $\boldsymbol{\alpha}$, $\{\mathbf{W}'_i\}_{i\in[2\ell]}$, and $\mathbf{Z}$.

Challenge Queries. When $\mathcal{A}$ submits $(\texttt{Challenge}, 1, \mathsf{ID}_0, \mathsf{ID}_1, \mathsf{M}_0, \mathsf{M}_1)$ to the challenger, $\mathcal{B}_3$ picks $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$, $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_p^{6\times 1}$, and $\mathsf{M}_{\mathsf{rand}} \xleftarrow{\$} \mathbb{G}_T$ and computes $g^{\mathbf{Uv}}$ and $g^{\mathbf{Lv}}$. Then it returns

$$\mathsf{CT} = \left(C_1 = g^{\mathbf{B}\binom{\mathbf{s}}{\mathbf{Uv}}}, \quad C_2 = g^{\sum_{i\in S_{\mathsf{coin}}} \mathbf{W}_i'\mathbf{B}\binom{\mathbf{s}}{\mathbf{Uv}}} \cdot g^{\mathbf{BLv}}, \quad C_3 = \mathsf{M}_{\mathsf{rand}} \right)$$

to $\mathcal{A}$ where $S_{\mathsf{coin}} = \{2i - \mathsf{ID}_{\mathsf{coin},i} | i \in [\ell]\}$. Note that $g^{\mathbf{B}\binom{\mathbf{s}}{\mathbf{Uv}}}$, $g^{\sum_{i\in S_{\mathsf{coin}}} \mathbf{W}_i'\mathbf{B}\binom{\mathbf{s}}{\mathbf{Uv}}}$, and $g^{\mathbf{BLv}}$ can be efficiently computed from $g^{\mathbf{Uv}}$, $g^{\mathbf{Lv}}$, $\mathbf{B}$, $\mathbf{s}$, and $\{\mathbf{W}_i'\}_{i\in[2\ell]}$. We claim that it is properly distributed ciphertext in Game_4 or Game_5, depending on whether $\mathbf{X} \xleftarrow{\$} \mathsf{Rk}_2(\mathbb{Z}_p^{6\times 6})$ or $\mathbf{X} \xleftarrow{\$} \mathsf{Rk}_6(\mathbb{Z}_p^{6\times 6})$.

In the case of $\mathbf{X} \xleftarrow{\$} \mathsf{Rk}_2(\mathbb{Z}_p^{6\times 6})$, $\mathcal{B}_3$ implicitly sets $\binom{\hat{s}}{\tilde{s}} := \mathbf{Uv}$. Since $\mathbf{U}$ is a full-rank matrix, $\binom{\hat{s}}{\tilde{s}}$ is uniformly random over $\mathbb{Z}_p^{2\times 1}$. We also have that $\mathbf{Lv} = \mathbf{AUv} = \mathbf{A}\binom{\hat{s}}{\tilde{s}}$. Thus, it holds that $C_1 = g^{\mathbf{B}\binom{\mathbf{s}}{\hat{s}}{\tilde{s}}}$ and

$$C_2 = g^{\sum_{i\in S_{\mathsf{coin}}} \mathbf{W}_i'\mathbf{B}\binom{\mathbf{s}}{\hat{s}}{\tilde{s}}} \cdot g^{\mathbf{BA}\binom{\hat{s}}{\tilde{s}}} = g^{((\sum_{i\in S_{\mathsf{coin}}} \mathbf{W}_i'\mathbf{B}) + \mathbf{BA}')\binom{\mathbf{s}}{\hat{s}}{\tilde{s}}} = g^{\sum_{i\in S_{\mathsf{coin}}} \mathbf{W}_i\mathbf{B}\binom{\mathbf{s}}{\hat{s}}{\tilde{s}}}.$$

Here, $\mathcal{B}_3$ implicitly sets $\mathbf{W}_i = \mathbf{W}_i' + \mathbf{BA}'\mathbf{B}^{-1}$ for $i \in [1, 2]$ and $\mathbf{W}_i = \mathbf{W}_i'$ for $i \in [3, 2\ell]$ where $\mathbf{A}' \in \mathbb{Z}_p^{4\times 4}$ is defined so that $\pi_1(\mathbf{A}') = \mathbf{0}$ and the right half of $\mathbf{A}'$ corresponds to $\mathbf{A}$. Therefore, in this case, $\mathcal{B}_3$ simulates the ciphertext in Game_4. (Note that $\mathbf{A}'$ is fixed throughout the game.)

On the other hand, if $\mathbf{X} \xleftarrow{\$} \mathsf{Rk}_6(\mathbb{Z}_p^{6\times 6})$, $(\mathbf{Uv}, \mathbf{Lv})$ is uniformly random over $\mathbb{Z}_p^{2\times 1} \times \mathbb{Z}_p^{4\times 1}$. Since $\mathbf{B}$ is full-rank, it can be seen that (C_1, C_2) is also uniformly random over $\mathbb{Z}_p^{4\times 1} \times \mathbb{Z}_p^{4\times 1}$. Therefore, the distribution of challenge ciphertexts is the same as that of Game_5.

Guess. Finally, $\mathcal{B}_3$ outputs $\mathcal{A}$'s output as its guess.

As we have seen, the game corresponds to the case of Game_4 if $\mathbf{X} \xleftarrow{\$} \mathsf{Rk}_2(\mathbb{Z}_p^{6\times 6})$ and Game_5 if $\mathbf{X} \xleftarrow{\$} \mathsf{Rk}_6(\mathbb{Z}_p^{6\times 6})$. Thus, we may conclude that $|\mathsf{Adv}_4(\kappa) - \mathsf{Adv}_5(\kappa)| \leq \mathsf{Adv}^{\mathsf{P}_{12}}_{\mathcal{B}_3}(\kappa)$. □

G Omitted Details from Section 8

G.1 Definitions

Q-fold One-Time Signature. We define Q-fold one-time signature scheme introduced in [32], which consists of the following five algorithms $\Sigma = (\Sigma.\mathsf{Par}, \Sigma.\mathsf{Gen}, \Sigma.\mathsf{Sign}, \Sigma.\mathsf{Verify})$. The parameter generation algorithm $\Sigma.\mathsf{Par}$ takes 1^κ as input and outputs a public parameter pp_Σ. The key generation algorithm $\Sigma.\mathsf{Gen}$ takes pp_Σ as input and outputs a verification key vk and a signing key sigk. The signing algorithm $\Sigma.\mathsf{Sign}$ takes sigk and a message M as input and outputs a signature σ on M. The verification algorithm $\Sigma.\mathsf{Verify}$ takes a verification key vk, a message M, and a (purported) signature σ and outputs 1, which indicates that the signature is valid, or 0.

Security. We now define the security for a Q-fold OTS scheme $\Sigma = (\Sigma.\mathsf{Par}, \Sigma.\mathsf{Gen}, \Sigma.\mathsf{Sign}, \Sigma.\mathsf{Verify})$ by the following game between a challenger and an attacker $\mathcal{A}$.

Setup. The challenger runs $\mathsf{pp}_\Sigma \xleftarrow{\$} \Sigma.\mathsf{Par}(1^\kappa)$ and $(\mathsf{vk}_j, \mathsf{sigk}_j) \xleftarrow{\$} \Sigma.\mathsf{Gen}(\mathsf{pp}_\Sigma)$ for $j \in [Q]$. Then, $(\mathsf{pp}_\Sigma, \{\mathsf{vk}_j\}_{j\in[Q]})$ is given to $\mathcal{A}$.

Signing Queries. In the game, $\mathcal{A}$ adaptively makes signing queries at most Q times. When $\mathcal{A}$ submits $(\texttt{Sign}, j \in [Q], \mathsf{M})$, the challenger runs $\sigma \xleftarrow{\$} \Sigma.\mathsf{Sign}(\mathsf{sigk}_j, \mathsf{M})$ to obtain the signature σ and returns it to $\mathcal{A}$. $\mathcal{A}$ may request (up to) one signature for each $j \in [Q]$.

Forgery. At last, $\mathcal{A}$ outputs a forgery $(j^\star \in [Q], \mathsf{M}^\star, \sigma^\star)$. We say that $\mathcal{A}$ wins the game if $\Sigma.\mathsf{Verify}(\mathsf{vk}_{j^\star}, \mathsf{M}^\star, \sigma^\star) = 1$ and one of the following conditions holds.

- $\mathcal{A}$ have never made a signing query for $j^\star$.

- $\mathcal{A}$ made a signing query for $j^\star$. Let the query be $(\mathtt{Sign}, j^\star, \mathsf{M})$ and the response to it by the challenger be σ. Then, $(\mathsf{M}, \sigma) \neq (\mathsf{M}^\star, \sigma^\star)$.

We define the advantage of the adversary $\mathcal{A}$ $\mathsf{Adv}^{\mathsf{OTS}}_{\mathcal{A},\Sigma,Q}(\kappa)$ as the probability that $\mathcal{A}$ wins in the above game.

Definition 4. *We say that a Q-fold OTS scheme Σ is secure if $\mathsf{Adv}^{\mathsf{OTS}}_{\mathcal{A},\Sigma,Q}(\kappa)$ is negligible for any polynomially bounded Q and any PPT adversary $\mathcal{A}$.*

CONCRETE CONSTRUCTION. Hofheinz and Jager [32] gave a concrete construction of tightly secure Q-fold one-time signature. In their scheme, the public parameter (pp_Σ) consists of two group elements and description of a collision resistant hash function. vk consists of two group elements and σ consists of two elements in $\mathbb{Z}_p$. The scheme can be used to sign any bit strings. The security of the scheme is reduced to the discrete logarithm problem in the underlying group of the scheme. The corresponding reduction looses only a factor of 2, which is independent of Q.

Public Key Encryption. A public key encryption scheme consists of the following five algorithms $\Psi = (\Psi.\mathsf{Par}, \Psi.\mathsf{Gen}, \Psi.\mathsf{Enc}, \Psi.\mathsf{Dec})$. The parameter generation algorithm $\Psi.\mathsf{Par}$ takes 1^κ as input and outputs a public parameter pp. The key generation algorithm $\Psi.\mathsf{Gen}$ takes pp as input and outputs a encryption key ek and a decryption key dk. The encryption algorithm $\Psi.\mathsf{Enc}$ takes ek and a message M and outputs a ciphertext CT. The decryption algorithm takes dk and a ciphertext CT and outputs a message M or $\perp$ which indicates that the ciphertext is not in a valid form. We require the usual correctness properties.

Security. We now define (μ, Q_c, Q_k)-security for a PKE scheme $\Psi = (\Psi.\mathsf{Par}, \Psi.\mathsf{Gen}, \Psi.\mathsf{Enc}, \Psi.\mathsf{Dec})$ by the following game between a challenger and an attacker A.

Setup. The challenger runs $\mathsf{pp} \xleftarrow{\$} \Psi.\mathsf{Par}(1^\kappa)$ and $(\mathsf{ek}^{(j)}, \mathsf{dk}^{(j)}) \xleftarrow{\$} \Psi.\mathsf{Gen}(\mathsf{pp})$ for $j \in [\mu]$. The challenger also picks random coin $\mathsf{coin} \xleftarrow{\$} \{0,1\}$ whose value is fixed throughout the game. Then, $(\mathsf{pp}, \{\mathsf{ek}^{(j)}\}_{j\in[\mu]})$ is given to $\mathcal{A}$.

In the following, $\mathcal{A}$ adaptively makes the following two types of queries in an arbitrary order.

−**Decryption Queries.** The adversary $\mathcal{A}$ submits $(\mathtt{Decryption}, j \in [\mu], \mathsf{CT})$ to the challenger. Then, the challenger runs $\Psi.\mathsf{Dec}(\mathsf{dk}^{(j)}, \mathsf{CT}) \to \mathsf{M}/\perp$ and returns the result to $\mathcal{A}$.

−**Challenge Queries.** The adversary $\mathcal{A}$ submits $(\mathtt{Challenge}, j \in [\mu], \mathsf{M}_0, \mathsf{M}_1 \in \mathcal{M})$ to the challenger. Then, the challenger runs $\mathsf{CT} \xleftarrow{\$} \Psi.\mathsf{Enc}(\mathsf{ek}^{(j)}, \mathsf{M}_{\mathsf{coin}})$ and returns CT to $\mathcal{A}$.

Guess. At last, $\mathcal{A}$ outputs a guess coin' for coin. The advantage of an attacker $\mathcal{A}$ is defined as $\mathsf{Adv}^{\mathsf{PKE}}_{\mathcal{A},\Psi,(\mu,Q_c,Q_k)}(\kappa) = |\Pr[\mathsf{coin}' = \mathsf{coin}] - \frac{1}{2}|$.

We say that the adversary $\mathcal{A}$ is valid if and only if $\mathcal{A}$ never queries $(\mathtt{Decryption}, j, \mathsf{CT})$ such that CT was obtained as an answer to the query $(\mathtt{Challenge}, j, \mathsf{M}_0, \mathsf{M}_1)$ for the same j and some $\mathsf{M}_0, \mathsf{M}_1 \in \mathcal{M}$; $\mathcal{A}$ has made at most Q_c challenge queries; and $\mathcal{A}$ has made at most Q_k key extraction queries.

Definition 5. *We say that a PKE scheme Ψ is secure if $\mathsf{Adv}^{\mathsf{PKE}}_{\mathcal{A},\Psi,(\mu,Q_c,Q_k)}(\kappa)$ is negligible for any polynomially bounded μ, Q_c, Q_k, and any valid PPT adversary $\mathcal{A}$.*

G.2 CCA Secure PKE from IBE

Here, we show a generic construction of CCA-secure PKE from an IBE scheme and a Q-fold OTS scheme. We require that in the original IBE scheme, the key generation algorithm does not output any secret parameter. Namely, we require that $\mathsf{sp} = \perp$. This requirement is satisfied in all of our IBE schemes except for that in Section 7. If the original IBE and Q-fold OTS scheme are tightly secure, the resulting PKE is tightly secure as well. We construct PKE scheme Ψ from an IBE scheme $\Phi = (\Phi.\mathsf{Par}, \Phi.\mathsf{Gen}, \Phi.\mathsf{Ext}, \Phi.\mathsf{Enc}, \Phi.\mathsf{Dec})$ and a Q-fold OTS scheme $\Sigma = (\Sigma.\mathsf{Par}, \Sigma.\mathsf{Gen}, \Sigma.\mathsf{Sign}, \Sigma.\mathsf{Verify})$ as follows. Without loss of generality, we assume that identity space of Φ contains all possible vk output by $\Sigma.\mathsf{Gen}$.

$\Psi.\mathsf{Par}(1^\kappa)$: It runs $\Sigma.\mathsf{Par}(1^\kappa) \to \mathsf{pp}_\Sigma$ and $\Phi.\mathsf{Par}(1^\kappa) \to (\mathsf{pp}_\Phi, \mathsf{sp} = \bot)$. Then, it outputs $\mathsf{pp}_\Psi = (\mathsf{pp}_\Phi, \mathsf{pp}_\Sigma)$.

$\Psi.\mathsf{Gen}(\mathsf{pp}_\Psi)$ It parses $\mathsf{pp}_\Psi \to (\mathsf{pp}_\Phi, \mathsf{pp}_\Sigma)$ and runs $\Phi.\mathsf{Gen}(\mathsf{pp}_\Phi, \mathsf{sp} = \bot) \to (\mathsf{mpk}, \mathsf{msk})$. Then, it outputs the encryption key $\mathsf{ek} = (\mathsf{pp}_\Phi, \mathsf{pp}_\Sigma, \mathsf{mpk})$ and the decryption key $\mathsf{dk} = (\mathsf{mpk}, \mathsf{msk})$.

$\Psi.\mathsf{Enc}(\mathsf{ek}, \mathsf{M})$ It first parses $\mathsf{ek} \to (\mathsf{pp}_\Phi, \mathsf{pp}_\Sigma, \mathsf{mpk})$. Then, it runs $\Sigma.\mathsf{Gen}(\mathsf{pp}_\Sigma) \to (\mathsf{vk}, \mathsf{sigk})$, $\Phi.\mathsf{Enc}(\mathsf{mpk}, \mathsf{vk}, \mathsf{M}) \to \mathsf{CT}_\Phi$, and $\Phi.\mathsf{Sign}(\mathsf{sigk}, \mathsf{CT}_\Phi) \to \sigma$. Finally, it outputs $\mathsf{CT}_\Psi = (\mathsf{vk}, \mathsf{CT}_\Phi, \sigma)$.

$\Psi.\mathsf{Dec}(\mathsf{dk}, \mathsf{CT}_\Psi)$ It first parses the ciphertext as $\mathsf{CT}_\Psi \to (\mathsf{vk}, \mathsf{CT}_\Phi, \sigma)$. Any ciphertext not satisfying this format is rejected (i.e., the decryption algorithm outputs $\bot$). Then, it checks whether σ is a valid signature on CT_Φ by running $\Sigma.\mathsf{Verify}(\mathsf{vk}, \mathsf{CT}_\Phi, \sigma)$. If it is 0, the decryption algorithm outputs $\bot$. Otherwise, it runs $\Phi.\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{vk}) \to \mathsf{sk}_{\mathsf{vk}}$ and outputs $\Phi.\mathsf{Dec}(\mathsf{sk}_{\mathsf{vk}}, \mathsf{CT}_\Phi) \to \mathsf{M}$ or $\bot$.

Theorem 16. *For any valid adversary $\mathcal{A}$ against the above PKE scheme, there exist adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that* $\mathsf{Adv}^{\mathsf{PKE}}_{\mathcal{A},\Psi,(\mu,Q_c,Q_k)}(\kappa) \le \mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{B}_1,\Phi,(\mu,Q_c,Q_k)}(\kappa) + \mathsf{Adv}^{\mathsf{OTS}}_{\mathcal{B}_2,\Sigma,Q_c}(\kappa)$ *and* $\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2)\} \approx \mathsf{Time}(\mathcal{A}) + (\mu + Q_k + Q_c) \cdot \mathsf{poly}(\kappa)$ *where* $\mathsf{poly}(\kappa)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We prove the theorem by the following sequence of the games. We write $\mathsf{Adv}_{\mathrm{xx}}(\kappa)$ to denote the advantage of $\mathcal{A}$ in $\mathsf{Game}_{\mathrm{xx}}$.

Game_0 : This is the real security game.

Game_1 : In this game, the challenger runs $\Sigma.\mathsf{Par}(1^\kappa) \to \mathsf{pp}_\Sigma$ and $(\mathsf{vk}_i, \mathsf{sigk}_i) \xleftarrow{\$} \Sigma.\mathsf{Gen}(\mathsf{pp}_\Sigma)$ for $i \in [Q_c]$ at the outset of the game and use $(\mathsf{vk}_i, \mathsf{sigk}_i)$ to create the i-th challenge ciphertext.

Game_2 : In this game, the challenger stops the experiment and forces $\mathcal{A}$ to output a random bit if $\mathcal{A}$ submits $(\mathtt{Decryption}, j', \mathsf{CT}'_\Psi = (\mathsf{vk}', \mathsf{CT}'_\Phi, \sigma'))$ that satisfies $\Sigma.\mathsf{Verify}(\mathsf{vk}', \mathsf{CT}'_\Phi, \sigma') = 1$ and one of the following conditions:

(Case A) There exists $i^\star \in [Q_c]$ such that $\mathsf{vk}' = \mathsf{vk}_{i^\star}$ and $\mathcal{A}$ has not made the $i^\star$-th challenge query yet.

(Case B) There exists $i^\star \in [Q_c]$ such that $\mathsf{vk}' = \mathsf{vk}_{i^\star}$ and $\mathcal{A}$'s $i^\star$-th challenge query is in the form of $(\mathtt{Challenge}, j', \mathsf{M}_0, \mathsf{M}_1)$ for the same j'.

Since the change from Game_0 to Game_1 is only conceptual, we have $\mathsf{Adv}_0(\kappa) = \mathsf{Adv}_1(\kappa)$. Therefore, we have $\mathsf{Adv}^{\mathsf{PKE}}_{\mathcal{A},\Psi,(\mu,Q_c,Q_k)}(\kappa) = \mathsf{Adv}_0(\kappa) \le |\mathsf{Adv}_1(\kappa) - \mathsf{Adv}_2(\kappa)| + \mathsf{Adv}_2(\kappa)$ and it suffices to show Lemma 33 and 34 in the following. □

Lemma 33. (Game_1 to Game_2). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_1$ such that* $|\mathsf{Adv}_1(\kappa) - \mathsf{Adv}_2(\kappa)| \le \mathsf{Adv}^{\mathsf{OTS}}_{\mathcal{B}_1,\Sigma,Q_c}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}_1) \approx Q_c \cdot \mathsf{poly}(\kappa) + \mathsf{Time}(\mathcal{A})$ *where* $\mathsf{poly}(\kappa)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. The Game_2 differs from Game_1 only if $\mathcal{A}$ makes a decryption query of the specific form defined as above (Case A and B). We let the probability of this event in Game_1 be ϵ. We construct an adversary $\mathcal{B}_1$ against the Q_c-fold OTS scheme whose advantage is ϵ from $\mathcal{A}$.

Setup. At the outset of the game, $\mathcal{B}_1$ is given $(\mathsf{pp}_\Sigma, \{\mathsf{vk}_i\}_{i\in[Q_c]})$. Then, it runs $\Phi.\mathsf{Par}(1^\kappa) \to \mathsf{pp}_\Phi$ and $\Phi.\mathsf{Gen}(\mathsf{pp}_\Phi) \to (\mathsf{mpk}^{(j)}, \mathsf{msk}^{(j)})$ for $j \in [\mu]$ and returns $\mathsf{pp}_\Psi = (\mathsf{pp}_\Phi, \mathsf{pp}_\Sigma)$ and $\{\mathsf{ek}^{(j)} = (\mathsf{pp}_\Phi, \mathsf{pp}_\Sigma, \mathsf{mpk}^{(j)})\}_{j\in[\mu]}$ to $\mathcal{A}$. It also picks $\mathsf{coin} \xleftarrow{\$} \{0,1\}$.

Challenge Queries. For the i-th challenge query $(\mathtt{Challenge}, j, \mathsf{M}_0, \mathsf{M}_1)$ made by $\mathcal{A}$, $\mathcal{B}_1$ proceeds as follows. It first runs $\Phi.\mathsf{Enc}(\mathsf{mpk}^{(j)}, \mathsf{vk}_i, \mathsf{M}_{\mathsf{coin}}) \to \mathsf{CT}_\Phi$ and then submits $(\mathtt{Sign}, i, \mathsf{CT}_\Phi)$ to its challenger. Then, $\Sigma.\mathsf{Sign}(\mathsf{sigk}_i, \mathsf{CT}_\Phi) \to \sigma$ is returned to $\mathcal{B}_3$. Finally, $\mathcal{B}_3$ returns $\mathsf{CT}_\Psi = (\mathsf{vk}_i, \mathsf{CT}_\Phi, \sigma)$ to $\mathcal{A}$.

Decryption Queries. When $\mathcal{A}$ makes query $(\texttt{Decryption}, j', \mathsf{CT}'_\Psi = (\mathsf{vk}', \mathsf{CT}'_\Phi, \sigma'))$, $\mathcal{B}_1$ proceeds as follows. If $\Sigma.\mathsf{Verify}(\mathsf{vk}', \mathsf{CT}'_\Phi, \sigma') = 0$, it returns $\perp$ to $\mathcal{A}$. If not, $\mathcal{B}_1$ searches for $i^\star$ such that $\mathsf{vk}' = \mathsf{vk}_{i^\star}$. If there is such $i^\star$, $\mathcal{B}_1$ checks whether (Case A) or (Case B) holds. If it holds, $\mathcal{B}_1$ stops the game and outputs $(i^\star, \mathsf{CT}'_\Phi, \sigma')$ as its forgery. Otherwise, it answers the decryption query using $\{\mathsf{dk}^{(j)} = (\mathsf{mpk}^{(j)}, \mathsf{msk}^{(j)})\}_{j\in[\mu]}$.

Analysis. Let $(i^\star, \mathsf{CT}'_\Phi, \sigma')$ be the output of $\mathcal{B}_1$. If (Case A) holds, $\mathcal{B}_1$ has not made signing query for $i^\star$. Therefore, $\mathcal{B}_1$ wins the game in this case. We then consider (Case B). Let the $i^\star$-th challenge query be $(\texttt{Challenge}, j', \mathsf{M}_0, \mathsf{M}_1)$ and answer to the query be $\mathsf{CT}''_\Psi = (\mathsf{vk}_{i^\star}, \mathsf{CT}''_\Phi, \sigma'')$. Note that $\mathcal{B}_1$ has made a signing query $(\texttt{Sign}, i^\star, \mathsf{CT}''_\Phi)$ to obtain σ''. Since $\mathcal{A}$ is a valid adversary, we have that $(j', \mathsf{CT}'_\Psi) \neq (j', \mathsf{CT}''_\Psi)$. In particular, we have $(\mathsf{CT}'_\Phi, \sigma') \neq (\mathsf{CT}''_\Phi, \sigma'')$. Therefore, $\mathcal{B}_1$ wins the game also in this case. □

Lemma 34. *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}_2$ such that* $\mathsf{Adv}_2(\kappa) \leq \mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{B}_2,\Phi,(\mu,Q_c,Q_k)}(\kappa)$ *and* $\mathsf{Time}(\mathcal{B}_2) \approx (Q_c + Q_k) \cdot \mathsf{poly}(\kappa) + \mathsf{Time}(\mathcal{A})$ *where* $\mathsf{poly}(\kappa)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. We construct an adversary $\mathcal{B}_2$ against (μ, Q_c, Q_k)-security of the IBE scheme from $\mathcal{A}$. $\mathcal{B}_2$ simulates Game_2 for $\mathcal{A}$ as follows.

Setup. At the outset of the game, $\mathcal{B}_2$ is given pp_Φ and $\{\mathsf{mpk}^{(j)}\}_{j\in[\mu]}$. Then, it runs $\Sigma.\mathsf{Par}(1^\kappa) \to \mathsf{pp}_\Sigma$ and returns $\mathsf{pp}_\Psi = (\mathsf{pp}_\Phi, \mathsf{pp}_\Sigma)$ and $\{\mathsf{ek}^{(j)} = (\mathsf{pp}_\Phi, \mathsf{pp}_\Sigma, \mathsf{mpk}^{(j)})\}_{j\in[\mu]}$ to $\mathcal{A}$. $\mathcal{B}_2$ also picks $(\mathsf{vk}_i, \mathsf{sigk}_i) \xleftarrow{\$} \Sigma.\mathsf{Gen}(\mathsf{pp}_\Sigma)$ for $i \in [Q_c]$.

Challenge Queries. When the adversary $\mathcal{A}$ makes the i-th challenge query $(\texttt{Challenge}, j, \mathsf{M}_0, \mathsf{M}_1)$, $\mathcal{B}_2$ first requests $(\texttt{Challenge}, j, \mathsf{vk}_i, \mathsf{M}_0, \mathsf{M}_1)$ for its challenger and receives $\Phi.\mathsf{Enc}(\mathsf{mpk}^{(j)}, \mathsf{vk}_i, \mathsf{M}_{\mathsf{coin}}) \to \mathsf{CT}_\Phi$. Then, $\mathcal{B}_2$ runs $\Sigma.\mathsf{Sign}(\mathsf{sigk}, \mathsf{CT}_\Phi) \to \sigma$ and returns the challenge ciphertext $(\mathsf{vk}, \mathsf{CT}_\Phi, \sigma)$ to $\mathcal{A}$.

Decryption Queries. When $\mathcal{A}$ makes query $(\texttt{Decryption}, j, \mathsf{CT}'_\Psi = (\mathsf{vk}', \mathsf{CT}'_\Phi, \sigma'))$, $\mathcal{B}_2$ first checks the validity of σ' by $\Sigma.\mathsf{Verify}(\mathsf{vk}', \mathsf{CT}'_\Phi, \sigma')$. If it is 0, it returns $\perp$. Otherwise, $\mathcal{B}_2$ checks whether $(j', \mathsf{CT}'_\Psi = (\mathsf{vk}', \mathsf{CT}'_\Phi, \sigma'))$ satisfies (Case A) or (Case B) condition. If it satisfies, $\mathcal{B}_2$ aborts and outputs a random bit. Otherwise, $\mathcal{B}_2$ makes key extraction query $(\texttt{Extraction}, j', \mathsf{vk}')$ to its challenger to obtain $\mathsf{sk}^{(j')}_{\mathsf{vk}'}$ and returns $\Phi.\mathsf{Dec}(\mathsf{CT}'_\Phi, \mathsf{sk}^{(j')}_{\mathsf{vk}'}) \to \mathsf{M}/\perp$.

Output. Finally, $\mathcal{B}_2$ outputs the same bit as $\mathcal{A}$ as its guess.

Analysis. It is clear that we have $\mathsf{Adv}_2(\kappa) \leq \mathsf{Adv}^{\mathsf{IBE}}_{\mathcal{B}_2,\Phi,(\mu,Q_c,Q_k)}(\kappa)$. Here, we check that $\mathcal{B}_2$ is a valid adversary. At first, we check that $\mathcal{B}_2$ never makes any prohibited key extraction query. For a decryption query $(\texttt{Decryption}, j', \mathsf{CT}'_\Psi = (\mathsf{vk}', \mathsf{CT}'_\Phi, \sigma'))$ that satisfies neither (Case A) nor (Case B) condition, we have that $\mathsf{vk}' \notin \{\mathsf{vk}_i\}_{i\in[Q_c]}$, or, for all $i^\star \in [Q_c]$ such that $\mathsf{vk}_{i^\star} = \mathsf{vk}'$, we have that the $i^\star$-th challenge query made by $\mathcal{A}$ is $(\texttt{Challenge}, j'', \mathsf{M}''_0, \mathsf{M}''_1)$ for some $j'' \neq j'$. In any case, $\mathcal{B}_2$ is allowed to make key extraction query of the form $(\texttt{Extraction}, j', \mathsf{vk}')$. Next, we check that $\mathcal{B}_2$ never makes any prohibited challenge query. Let us assume that $\mathcal{B}_2$ makes the i-th challenge query $(\texttt{Challenge}, j, \mathsf{vk}_i, \mathsf{M}_0, \mathsf{M}_1)$ for some j, M_0, and M_1. Then, since $\mathcal{B}_2$ has not aborted until then, $\mathcal{A}$ has not made any decryption query that satisfies (Case A). Therefore, for all key extraction query $(\texttt{Extraction}, j', \mathsf{vk}')$ made by $\mathcal{B}_2$ until then, we have that $\mathsf{vk}' \neq \mathsf{vk}_i$. □

H Concrete Descriptions of Our Schemes

Here, we show concrete description of our proposed schemes. In all of the following schemes, we let the identity space be $\{0,1\}^\ell$.

H.1 Description of IBE Scheme $\Phi^{\mathsf{comp}}_{\mathsf{cc}}$

Let the message space be $\mathcal{M} = \{0,1\}^m$. We also let $\mathcal{H}$ be a family of pairwise independent hash functions $\mathsf{H} : \mathbb{G}_T \to \mathcal{M}$. We assume that $\sqrt{\frac{|\mathcal{M}|}{p_2}} = 2^{-\Omega(\kappa)}$.

$\mathsf{Par}(1^\kappa)$: It first runs $(N, \mathbb{G}, \mathbb{G}_T, g_1, g_2, g_3, g_4, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{comp}}(1^\kappa)$ and picks $\mathbf{w} = (w_1 \ldots, w_{2\ell}) \xleftarrow{\$} \mathbb{Z}_N^{2\ell}$, $a \xleftarrow{\$} \mathbb{Z}_N^*$, $\mathsf{H} \xleftarrow{\$} \mathcal{H}$. Then it sets $h := (g_1 g_2 g_3 g_4)^a$ and outputs $\mathsf{pp} = (g_1, g_1^{\mathbf{w}}, g_4, h, \mathsf{H})$ and $\mathsf{sp} = \perp$.

$\mathsf{Gen}(\mathsf{pp}, \mathsf{sp})$: It picks $\alpha \xleftarrow{\$} \mathbb{Z}_N$ and outputs master public key $\mathsf{mpk} = (\mathsf{pp}, e(g_1, h)^\alpha)$ and $\mathsf{msk} = \alpha$.

$\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{ID})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ where $\mathsf{ID}_i \in \{0,1\}$ is the i-th bit of $\mathsf{ID} \in \{0,1\}^\ell$. It then picks $r, \delta_1, \delta_2 \xleftarrow{\$} \mathbb{Z}_N$ and returns

$$\mathsf{sk}_{\mathsf{ID}} = \left(K_1 = h^\alpha g_1^{r \sum_{j \in S} w_j} g_4^{\delta_1}, \quad K_2 = g_1^{-r} g_4^{\delta_2} \right).$$

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}, \mathsf{M})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$. Then it picks $s \xleftarrow{\$} \mathbb{Z}_N$ and outputs

$$\mathsf{CT} = \left(C_1 = g_1^s, \quad C_2 = g_1^{s \sum_{j \in S} w_j}, \quad C_3 = \mathsf{H}\big(e(g_1, h)^{s\alpha}\big) \oplus \mathsf{M} \right).$$

$\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{CT})$: It parses $\mathsf{CT} \to (C_1, C_2, C_3)$ and computes $e(C_1, K_1) e(C_2, K_2) = e(g_1, h)^{s\alpha}$. Then, it recovers the message M by $\mathsf{M} = C_3 \oplus \mathsf{H}(e(g_1, h)^{s\alpha})$.

REMARK. There is a slight gap from the description of the above scheme to the resulting scheme obtained by our conversion in Section 4 to Π_{cc}. We call the former scheme (A) and the latter scheme (B). In particular, the description of the key extraction algorithm Ext in scheme (A) is slightly simplified compared to that of scheme (B). We explain this. In the key extraction algorithm of scheme (B), sk_j defined as Equation (6) is computed for all $j \in S$. We have

$$\mathsf{sk}_j = (h^{\alpha_j} g_1^{r_j w_j} g_4^{\delta_{j,j}}, \ \{g_1^{r_j w_k} g_4^{\delta_{j,k}}\}_{k \in S \setminus \{j\}}, \ g_1^{r_j} g_4^{\delta_{j,0}})$$

where $r_j, \delta_{j,0}, \delta_{j,k} \xleftarrow{\$} \mathbb{Z}_N$ for all $k \in S$. From the Equation (12), we have that

$$\begin{aligned} \mathsf{sk}_j^{\mathbf{E}_j} &= \big(h^{\alpha_j} g_1^{r_j w_j} g_4^{\delta_{j,j}} \cdot \prod_{k \in S \setminus \{j\}} (g_1^{r_j w_k} g_4^{\delta_{j,k}}), \ g_1^{-r_j} g_4^{\delta_{j0}}\big) \\ &= \big(h^{\alpha_j} g_1^{\sum_{k \in S} r_j w_k} g_4^{\sum_{k \in S} \delta_{j,k}}, \ g_1^{-r_j} g_4^{\delta_{j,0}}\big). \end{aligned}$$

Therefore, we have that $\mathsf{sk}_{\mathsf{ID}}$ in scheme (B) is in the form of

$$\mathsf{sk}_{\mathsf{ID}} = \prod_{j \in S} \mathsf{sk}_j^{\mathbf{E}_j} = \big(h^\alpha g_1^{(\sum_{j \in S} r_j)(\sum_{k \in S} w_k)} g_4^{\sum_{j \in S, k \in S} \delta_{j,k}}, \ g_1^{-\sum_{j \in S} r_j} g_4^{\sum_{j \in S} \delta_{j,0}} \big).$$

The above private key corresponds to that of scheme (A) if we replace $\sum_{j \in S, k \in S} \delta_{j,k}$, $\sum_{j \in S} r_j$, and $\sum_{j \in S} \delta_{j,0}$ with δ_1, r, and δ_2, respectively. It is clear that this does not change the distribution of the private key and thus does not harm the security at all.

We note that we will apply similar simplification to the key extraction algorithms that appear in this Appendix.

H.2 Description of IBE Scheme $\Phi_{\mathsf{slp}}^{\mathsf{comp}}$

Let the message space be $\mathcal{M} = \{0,1\}^m$. We also let $\mathcal{H}$ be a family of pairwise independent hash functions $\mathsf{H} : \mathbb{G}_T \to \mathcal{M}$. We assume that $\sqrt{\frac{|\mathcal{M}|}{p_2}} = 2^{-\Omega(\kappa)}$. We set ℓ_1 and ℓ_2 be integers such that $\ell_1 \ell_2 \geq 2\ell$.

$\mathsf{Par}(1^\kappa)$: It first runs $(N, \mathbb{G}, \mathbb{G}_T, g_1, g_2, g_3, g_4, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{comp}}(1^\kappa)$ and picks $\mathbf{w} = (u_1, \ldots, u_{\ell_1}, v, u'_1, \ldots, u'_{\ell_1}, v', w) \xleftarrow{\$} \mathbb{Z}_N^{2\ell_1+3}$, $a \xleftarrow{\$} \mathbb{Z}_N^*$, $\mathsf{H} \xleftarrow{\$} \mathcal{H}$. Then it sets $h := (g_1 g_2 g_3 g_4)^a$ and outputs $\mathsf{pp} = (g_1, g_1^{\mathbf{w}}, g_4, h, \mathsf{H})$ and $\mathsf{sp} = \perp$.

$\mathsf{Gen}(\mathsf{pp}, \mathsf{sp})$: It picks $\alpha \xleftarrow{\$} \mathbb{Z}_N$ and outputs $\mathsf{mpk} = (\mathsf{pp}, e(g_1, h)^\alpha)$ and $\mathsf{msk} = \alpha$.

$\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{ID})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ where $\mathsf{ID}_i \in \{0,1\}$ is the i-th bit of $\mathsf{ID} \in \{0,1\}^\ell$. It then defines $\tilde{S}_j$ and S_j for $j \in [\ell_2]$ as

$$\tilde{S}_j = S \cap [(j-1)\ell_1 + 1, j\ell_1], \quad S_j = \{j' - (j-1)\ell_1 \mid j' \in \tilde{S}_j\}. \tag{33}$$

Finally, it picks $\delta_1 \xleftarrow{\$} \mathbb{Z}_N$ and $r_i, \delta_{2,i}, \delta_{3,i}, \delta'_{3,i} \xleftarrow{\$} \mathbb{Z}_N$ for $i \in [\ell_2]$ and outputs

$$\mathsf{sk}_{\mathsf{ID}} = \left(K_1 = h^\alpha g_1^{\sum_{i\in[\ell_2]} w r_i} g_4^{\delta_1}, \left\{ K_{2,i} = g_1^{-r_i} g_4^{\delta_{2,i}}, \begin{array}{l} K_{3,i} = g_1^{(v+i\sum_{j\in S_i} u_j) r_i} g_4^{\delta_{3,i}}, \\ K'_{3,i} = g_1^{(v'+i\sum_{j\in S_i} u'_j) r_i} g_4^{\delta'_{3,i}} \end{array} \right\}_{i\in[\ell_2]} \right).$$

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}, \mathsf{M})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$. Then, it picks $s, t_1, \ldots, t_{\ell_2}, t'_1, \ldots, t'_{\ell_2} \xleftarrow{\$} \mathbb{Z}_N$ and outputs

$$\mathsf{CT} = \left(\begin{array}{l} C_1 = g_1^s, \\ C_4 = \mathsf{H}\big(e(g_1,h)^{s\alpha}\big) \oplus \mathsf{M}, \end{array} \left\{ \begin{array}{ll} C_{2,i} = g_1^{ws + (v+i\sum_{j\in S_i} u_j) t_i + (v'+i\sum_{j\in S_i} u'_j) t'_i}, & \\ C_{3,i} = g_1^{t_i}, & C'_{3,i} = g_1^{t'_i} \end{array} \right\}_{i\in[\ell_2]} \right).$$

$\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{CT})$: It parses the ciphertext CT as $\mathsf{CT} \to (C_1, \{C_{2,i}, C_{3,i}, C'_{3,i}\}_{i\in[\ell_2]}, C_4)$. It then computes

$$e(C_{2,i}, K_{2,i}) \cdot e(C_{3,i}, K_{3,i}) \cdot e(C'_{3,i}, K'_{3,i}) = e(g_1, g_1)^{-wsr_i}$$

Finally, it computes $e(C_1, K_1) \cdot \prod_{i\in S} e(g_1, g_1)^{-wsr_i} = e(g_1, h)^{s\alpha}$ and recovers the message M by $\mathsf{M} = C_4 \oplus \mathsf{H}(e(g_1, h)^{s\alpha})$.

H.3 Description of IBE Scheme $\Phi_{\mathsf{cc}}^{\mathsf{prime}}$

Let the message space be $\mathcal{M} = \mathbb{G}_T$.

$\mathsf{Par}(1^\kappa, \ell)$: It first runs $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{prime}}(1^\kappa)$. Then it picks $\mathbf{B} \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$, $\mathbf{W}_1, \ldots, \mathbf{W}_{2\ell} \xleftarrow{\$} \mathbb{Z}_p^{4\times 4}$ and a random full-rank diagonal matrix $\mathbf{D} \in \mathbb{Z}_p^{4\times 4}$ with the entries $(3,3)$ and $(4,4)$ being 1. Finally, it sets $\mathbf{Z} = \mathbf{B}^{-\top}\mathbf{D}$ and outputs

$$\mathsf{pp} = \begin{pmatrix} g, & g^{\pi_1(\mathbf{B})}, & g^{\pi_1(\mathbf{W}_1\mathbf{B})}, & \ldots, & g^{\pi_1(\mathbf{W}_{2\ell}\mathbf{B})} \\ h, & h^{\pi_1(\mathbf{Z})}, & h^{\pi_1(\mathbf{W}_1^\top\mathbf{Z})}, & \ldots, & h^{\pi_1(\mathbf{W}_{2\ell}^\top\mathbf{Z})} \end{pmatrix} \quad \text{and} \quad \mathsf{sp} = \perp.$$

$\mathsf{Gen}(\mathsf{pp}, \mathsf{sp})$: It picks $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{Z}_p^{4\times 1}$ and outputs $\mathsf{mpk} = (\mathsf{pp}, e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})})$ and $\mathsf{msk} = \boldsymbol{\alpha}$.

$\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{ID})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ where $\mathsf{ID}_i \in \{0,1\}$ is the i-th bit of $\mathsf{ID} \in \{0,1\}^\ell$. Then it picks random $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and returns

$$\mathsf{sk}_{\mathsf{ID}} = \left(K_1 = h^{\boldsymbol{\alpha} + \sum_{i\in S} \pi_1(\mathbf{W}_i^\top \mathbf{Z})\mathbf{r}}, \quad K_2 = h^{-\pi_1(\mathbf{Z})\mathbf{r}} \right).$$

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}, \mathsf{M})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$. Then it picks random $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and returns

$$\mathsf{CT} = \left(C_1 = g^{\pi_1(\mathbf{B})\mathbf{s}}, \quad C_2 = g^{\sum_{i\in S}\pi_1(\mathbf{W}_i\mathbf{B})\mathbf{s}}, \quad C_3 = e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})\mathbf{s}} \cdot \mathsf{M}\right).$$

$\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{CT})$: It parses the ciphertext CT as $\mathsf{CT} \to (C_1, C_2, C_3)$. It then computes

$$\begin{aligned} e(C_1, K_1)e(C_2, K_2) &= e(g^{\pi_1(\mathbf{B})\mathbf{s}}, h^{\boldsymbol{\alpha}+\sum_{i\in S}\pi_1(\mathbf{W}_i^\top \mathbf{Z})\mathbf{r}}) \cdot e(g^{\sum_{i\in S}\pi_1(\mathbf{W}_i\mathbf{B})\mathbf{s}}, h^{-\pi_1(\mathbf{Z})\mathbf{r}}) \\ &= e(g,h)^{\mathbf{s}^\top \pi_1(\mathbf{B})^\top(\boldsymbol{\alpha}+\sum_{i\in S}\pi_1(\mathbf{W}_i^\top\mathbf{Z})\mathbf{r}) - \mathbf{s}^\top\pi_1(\mathbf{B})^\top(\sum_{i\in S}\pi_1(\mathbf{W}_i^\top\mathbf{Z})\mathbf{r})} \\ &= e(g,h)^{\boldsymbol{\alpha}^\top\pi_1(\mathbf{B})\mathbf{s}} \end{aligned}$$

Finally, it recovers the message by $C_3/e(g,h)^{\boldsymbol{\alpha}^\top\pi_1(\mathbf{B})\mathbf{s}} = \mathsf{M}$.

H.4 Description of IBE Scheme $\Phi_{\mathsf{slp}}^{\mathsf{prime}}$

Let the message space be $\mathcal{M} = \mathbb{G}_T$. We also let ℓ_1 and ℓ_2 be integers such that $\ell_1\ell_2 \geq 2\ell$.

$\mathsf{Par}(1^\kappa, \ell)$: It first runs $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, e(\cdot)) \xleftarrow{\$} \mathcal{G}_{\mathsf{prime}}(1^\kappa)$. Then it picks $\mathbf{B} \xleftarrow{\$} \mathbb{GL}_4(\mathbb{Z}_p)$, $\mathbf{U}_1, \ldots, \mathbf{U}_{\ell_1}$, $\mathbf{V}, \mathbf{U}'_1, \ldots, \mathbf{U}'_{\ell_1}, \mathbf{V}', \mathbf{W} \xleftarrow{\$} \mathbb{Z}_p^{4\times 4}$ and a random full-rank diagonal matrix $\mathbf{D} \in \mathbb{Z}_p^{4\times 4}$ with the entries $(3,3)$ and $(4,4)$ being 1. Finally, it sets $\mathbf{Z} = \mathbf{B}^{-\top}\mathbf{D}$ and outputs

$$\mathsf{pp} = \begin{pmatrix} g, & g^{\pi_1(\mathbf{B})}, & g^{\pi_1(\mathbf{U}_1\mathbf{B})}, & \ldots, & g^{\pi_1(\mathbf{U}_{\ell_1}\mathbf{B})}, & g^{\pi_1(\mathbf{V}\mathbf{B})}, & g^{\pi_1(\mathbf{W}\mathbf{B})} \\ & & g^{\pi_1(\mathbf{U}'_1\mathbf{B})}, & \ldots, & g^{\pi_1(\mathbf{U}'_{\ell_1}\mathbf{B})}, & g^{\pi_1(\mathbf{V}'\mathbf{B})}, & \\ h, & h^{\pi_1(\mathbf{Z})}, & h^{\pi_1(\mathbf{U}_1^\top\mathbf{Z})}, & \ldots, & h^{\pi_1(\mathbf{U}_{\ell_1}^\top\mathbf{Z})} & h^{\pi_1(\mathbf{V}^\top\mathbf{Z})}, & h^{\pi_1(\mathbf{W}^\top\mathbf{Z})} \\ & & h^{\pi_1(\mathbf{U'}_1^\top\mathbf{Z})}, & \ldots, & h^{\pi_1(\mathbf{U'}_{\ell_1}^\top\mathbf{Z})} & h^{\pi_1(\mathbf{V'}^\top\mathbf{Z})} & \end{pmatrix} \quad \text{and} \quad \mathsf{sp} = \perp.$$

$\mathsf{Gen}(\mathsf{pp}, \mathsf{sp})$: It picks $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{Z}_p^{4\times 1}$ outputs $\mathsf{mpk} = (\mathsf{pp}, e(g,h)^{\boldsymbol{\alpha}^\top\pi_1(\mathbf{B})})$ and $\mathsf{msk} = \boldsymbol{\alpha}$.

$\mathsf{Ext}(\mathsf{msk}, \mathsf{mpk}, \mathsf{ID})$: It first sets $S = \{2i - \mathsf{ID}_i | i \in [\ell]\}$ where $\mathsf{ID}_i \in \{0,1\}$ is the i-th bit of $\mathsf{ID} \in \{0,1\}^\ell$. It then defines S_j for $j \in [\ell_2]$ as Equation (33). Finally, it picks $\mathbf{r}_1, \ldots, \mathbf{r}_{\ell_2} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and outputs $\mathsf{sk}_{\mathsf{ID}} =$

$$\left(K_1 = h^{\boldsymbol{\alpha}+\sum_{i\in[\ell_2]}\pi_1(\mathbf{W}^\top\mathbf{Z})\mathbf{r}_i}, \left\{K_{2,i} = h^{-\pi_1(\mathbf{Z})\mathbf{r}_i}, \begin{matrix} K_{3,i} = h^{\pi_1(\mathbf{V}^\top\mathbf{Z}+i\sum_{j\in S_i}\mathbf{U}_j^\top\mathbf{Z})\mathbf{r}_i} \\ K'_{3,i} = h^{\pi_1(\mathbf{V}'^\top\mathbf{Z}+i\sum_{j\in S_i}\mathbf{U}'^\top_j\mathbf{Z})\mathbf{r}_i} \end{matrix}\right\}_{i\in[\ell_2]}\right).$$

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}, \mathsf{M})$: It first sets S_j for $j \in [n_2]$ as above. Then, it picks $\mathbf{s}, \mathbf{t}_1, \ldots, \mathbf{t}_{\ell_2}, \mathbf{t}'_1, \ldots, \mathbf{t}'_{\ell_2} \xleftarrow{\$} \mathbb{Z}_p^{2\times 1}$ and outputs $\mathsf{CT} =$

$$\left(\begin{matrix} C_1 = g^{\pi_1(\mathbf{B})\mathbf{s}}, \\ C_4 = e(g,h)^{\boldsymbol{\alpha}^\top\pi_1(\mathbf{B})\mathbf{s}} \cdot \mathsf{M}, \end{matrix} \left\{\begin{matrix} C_{2,i} = g^{\pi_1(\mathbf{W}\mathbf{B})\mathbf{s}+\pi_1(\mathbf{V}\mathbf{B}+i\sum_{j\in S_i}\mathbf{U}_j\mathbf{B})\mathbf{t}_i+\pi_1(\mathbf{V}'\mathbf{B}+i\sum_{j\in S_i}\mathbf{U}'_j\mathbf{B})\mathbf{t}'_i} \\ C_{3,i} = g^{\pi_1(\mathbf{B})\mathbf{t}_i}, \qquad C'_{3,i} = g^{\pi_1(\mathbf{B})\mathbf{t}'_i} \end{matrix}\right\}_{i\in[\ell_2]}\right).$$

$\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{CT})$: It parses the ciphertext CT as $\mathsf{CT} \to (C_1, \{C_{2,i}, C_{3,i}, C'_{3,i}\}_{i\in[\ell_2]}, C_4)$. Observe that

$$\begin{aligned} & e(C_{2,i}, K_{2,i}) \cdot e(C_{3,i}, K_{3,i}) \cdot e(C'_{3,i}, K'_{3,i}) \\ = \; & e(g^{\pi_1(\mathbf{W}\mathbf{B})\mathbf{s}+\pi_1(\mathbf{V}\mathbf{B}+i\sum_{j\in S_i}\mathbf{U}_j\mathbf{B})\mathbf{t}_i+\pi_1(\mathbf{V}'\mathbf{B}+i\sum_{j\in S_i}\mathbf{U}'_j\mathbf{B})\mathbf{t}'_i}, h^{-\pi_1(\mathbf{Z})\mathbf{r}_i}) \\ & e(g^{\pi_1(\mathbf{B})\mathbf{t}_i}, h^{\pi_1(\mathbf{V}^\top\mathbf{Z}+i\sum_{j\in S_i}\mathbf{U}_j^\top\mathbf{Z})\mathbf{r}_i}) \cdot e(g^{\pi_1(\mathbf{B})\mathbf{t}'_i}, h^{\pi_1(\mathbf{V}'^\top\mathbf{Z}+i\sum_{j\in S_i}\mathbf{U}'^\top_j\mathbf{Z})\mathbf{r}_i}) \end{aligned}$$

$$= \quad e(g,h)^{-\mathbf{s}^\top \pi_1(\mathbf{WB})^\top \pi_1(\mathbf{Z})\mathbf{r}_i} = e(g,h)^{-\mathbf{s}^\top \pi_1(\mathbf{B})^\top \pi_1(\mathbf{W}^\top \mathbf{Z})\mathbf{r}_i}$$

for $i \in [\ell_2]$. It computes

$$\begin{aligned} & e(C_1, K_1) \prod_{i\in[\ell_2]} \big(e(C_{2,i}, K_{2,i}) \cdot e(C_{3,i}, K_{3,i}) \cdot e(C'_{3,i}, K'_{3,i})\big) \\ = \quad & e(g^{\pi_1(\mathbf{B})\mathbf{s}}, h^{\boldsymbol{\alpha}+\sum_{i\in[\ell_2]} \pi_1(\mathbf{W}^\top \mathbf{Z})\mathbf{r}_i}) \cdot \prod_{i\in[\ell_2]} e(g,h)^{-\mathbf{s}^\top \pi_1(\mathbf{B})^\top \pi_1(\mathbf{W}^\top \mathbf{Z})\mathbf{r}_i} \\ = \quad & e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})\mathbf{s}}. \end{aligned}$$

Finally, it recovers the message by $C_4/e(g,h)^{\boldsymbol{\alpha}^\top \pi_1(\mathbf{B})\mathbf{s}} = \mathsf{M}$.

Contents

1 Introduction **1**
- 1.1 Backgrounds . . . 1
- 1.2 Our Results . . . 2
- 1.3 Our Techniques . . . 3
- 1.4 Related Works . . . 7

2 Preliminaries **7**
- 2.1 Identity-based Encryption . . . 7
- 2.2 Composite-Order Bilinear Groups . . . 8

3 Broadcast Encoding: Definitions and Reductions **9**
- 3.1 Broadcast Encoding: Syntax . . . 10
- 3.2 Broadcast Encoding: Security . . . 11
- 3.3 Multi-Master-Key Hiding Security in Composite Order Groups . . . 12
- 3.4 Reduction from MMH security to CMH security . . . 13

4 Almost Tight IBE from Broadcast Encoding in Composite-Order Groups **15**

5 Framework for Constructions in Prime-Order Groups **17**
- 5.1 Decisional Linear Assumption and Intermediate Problems . . . 17
- 5.2 Computational-Master-Key Hiding Security on Prime Order Groups . . . 18
- 5.3 Preparation . . . 19
- 5.4 Multi-Master-Key Hiding Security in Prime-Order Groups . . . 20
- 5.5 Almost Tightly Secure IBE from Broadcast Encoding in Prime Order Groups . . . 22

6 Construction of Broadcast Encoding Schemes **23**
- 6.1 Broadcast Encoding with Constant-Size Ciphertexts . . . 24
- 6.2 Encoding with Sub-linear Parameters . . . 24
- 6.3 Implications . . . 25

7 Anonymous IBE with Tight Security Reduction. **25**

8 Application to CCA Secure Public Key Encryption **26**

References **27**

A Proof of Lemma 1 **29**

B Security Proof for Our Scheme in Section 4 **29**

C Random Self-Rducibility of the Intermediate Problems and Reductions from the DLIN Assumption **38**
- C.1 Reductions from DLIN . . . 38
- C.2 Random Self-Reducibility of the Intermediate Problems . . . 41

D Omitted Proofs from Section 5 **42**
D.1 Proof of Lemma 4 . . . 42
D.2 Proof of Lemma 5 . . . 42
D.3 Proof of Theorem 3 . . . 43
D.4 Security Proof for Our Scheme in Section 5.5 . . . 47

E CMH-Security of Π_{slp} **56**

F Proof of Theorem 5 **57**

G Omitted Details from Section 8 **61**
G.1 Definitions . . . 61
G.2 CCA Secure PKE from IBE . . . 62

H Concrete Descriptions of Our Schemes **64**
H.1 Description of IBE Scheme $\Phi_{\mathsf{cc}}^{\mathsf{comp}}$ . . . 64
H.2 Description of IBE Scheme $\Phi_{\mathsf{slp}}^{\mathsf{comp}}$ . . . 65
H.3 Description of IBE Scheme $\Phi_{\mathsf{cc}}^{\mathsf{prime}}$ . . . 66
H.4 Description of IBE Scheme $\Phi_{\mathsf{slp}}^{\mathsf{prime}}$ . . . 67