A New Approach for Practical Function-Private Inner Product Encryption

Sungwook Kim\textsuperscript{a}, Jinsu Kim\textsuperscript{a}, Jae Hong Seo\textsuperscript{b,∗}

\textsuperscript{a}Security Lab., Software R&D Center, Samsung Electronics, Republic of Korea
\textsuperscript{b}Department of Mathematics, Myongji University, Republic of Korea

Abstract

Functional Encryption (FE) is a new paradigm supporting restricted decryption keys of function $f$ that allows one to learn $f(x_j)$ from encryptions of messages $x_j$. A natural and practical security requirements for FE is to keep not only messages $x_1,\ldots,x_q$ but also functions $f_1,\ldots,f_q$ confidential from encryptions and decryptions keys, except inevitable information \{\(f_i(x_j)\)\}_{i,j\in[q]} for any polynomial a-priori unknown number $q$, where $f_i$’s and $x_j$’s are adaptively chosen by adversaries. Such the security requirement is called full function privacy. In this paper, we particularly focus on function-private FE for inner product functionality in the private key setting (simply called Inner Product Encryption (IPE)). To the best of our knowledge, there are two approaches for fully function-private IPE schemes in the private key setting. One of which is to employ a general transformation from (non-function-private) FE for general circuits (Brakerski and Segev, TCC 2015). This approach requires heavy crypto tools such as indistinguishability obfuscation (for non-function-private FE for general circuits) and therefore inefficient. The other approach is relatively practical; it directly constructs IPE scheme by using dual pairing vector spaces (DPVS) (Bishop et al. ASIACRYPT 2015, Datta et al. PKC 2016, and Tomida et al. ISC 2016).

We present a new approach for practical function-private IPE schemes that does not employ DPVS but generalizations of Brakerski-Segev transformation. Our generalizations of Brakerski-Segev transformation are easily combinable with existing (non-function-private) IPE schemes as well as (non-function-private) FE schemes for general circuits in several levels of security. Our resulting IPE schemes achieve better performance in comparison with Bishop et al. IPE scheme as well as Datta et al. IPE scheme while preserving the same security notion under the same complexity assumption. In comparison with Tomida et al. IPE scheme, ours have comparable performance in the size of both ciphertext and decryption key, but better performance in the size of master key.

Keywords: Functional Encryption, Function Privacy, Inner Product

∗Corresponding author
Email addresses: sv14.kim@samsung.com (Sungwook Kim), jinsu86.kim@samsung.com (Jinsu Kim), jaehongseo@mju.ac.kr (Jae Hong Seo)
1. **Introduction**

The usage of classical encryption schemes and its security requirement are simple; one who has the decryption key can recover all encrypted messages and all the other users including adversaries cannot obtain any information from the encrypted messages. That is, the decryption policy in classical encryption schemes is *all-or-nothing*. In Functional Encryption (FE), it provides more fine-grained handling of decryption policy by supporting restricted decryption keys associated with specific function $f$ that allow one to learn $f(x_j)$ from encryption of message $x_j$.

There are numerous realizations of FE schemes for either general or specific functionality. The first FE scheme for simple functionality (beyond classical encryption scheme) is identity-based encryption scheme [1, 2, 3]. Subsequently, several attribute-based encryption schemes [4, 5, 6] and predicate encryption schemes are proposed [7, 6, 8]. The general concept and the security notion (in terms of message privacy) of FE for general functionality is later formalized by Boneh, Sahai, Waters and O’Neill [9, 10]. Until recently, there has been extensively researched on the security of FE schemes and the methodology to construct FE schemes from a variety of assumptions [9, 10, 11, 12, 13, 14, 15, 16, 17, 18].

**Function Privacy in Private Key Setting.** Contrary to classical encryption scheme, there are several decryption keys in FE scheme and each decryption key is empowered to obtain restricted information from encrypted messages. Due to this difference, the decryption keys as well as the encrypted messages may be subject to keep confidential and we call such a security notion *function privacy* [19].

In this paper, we focus on the function privacy in the private key setting like many previous works [19, 20, 21, 22, 23] since the function privacy has much advantage in the private key setting. In the public key setting, given a decryption key associated with function $f$, the adversary can always learn $f(x_j)$ for any chosen $x_j$ by herself.¹ Therefore, to define meaningful function privacy in the public key setting, it seems necessary to assume the functions are sampled from somewhat unpredictable distributions [24, 25]. In the private key setting, we do not need such the restriction and we can consider the privacy of encrypted messages and the privacy of functional keys in a completely symmetric manner.

**Full Function Privacy.** There are several levels of function privacy in private key functional encryption. The strongest function privacy notion (and the message privacy as well) can be formalized in a simulation model. There are impossi-

---

¹The adversary first encrypts $x_j$, decrypts it by using the decryption key associated with $f$, and then finally obtains $f(x_j)$. 
bility results for such the simulation-based security (against adaptive unbounded queries attack) notion in the standard model, though some functionality can be achieved in ideal models such as random oracles [9] and generic group models [26, 23]. Unlike to simulation-based security, the indistinguishability-based strongest notion for function privacy, called full function privacy, is achievable in the standard model. We consider Agrawal et al.’s formalization for full function privacy [26], which is a generalization of predicate-privacy in attribute-based encryption [19]. Let us briefly explain the model of full function privacy. Adversaries are allowed to interact with two left-or-right oracles $KG_b(mk, \cdot, \cdot)$ and $Enc_b(mk, \cdot, \cdot)$ for randomly chosen coin $b \in \{0, 1\}$, where $KG_b$ takes two functions $f_0$ and $f_1$ as input and then returns a decryption key $sk_{f_b} \leftarrow KG(mk, f_b)$ and $Enc_b$ takes two messages $x_0$ and $x_1$ as input and then outputs an encryption $ct_{x_b} \leftarrow Enc(mk, x_b)$. In particular, adversaries can adaptively interact with oracles for any polynomial a-priori unbounded number of queries. To exclude inherently inevitable attacks, there is a condition in adversarial queries that for all queries $(x_0, x_1)$ and $(f_0, f_1)$, it should satisfy that

$$f_0(x_0) = f_1(x_1). \quad (1)$$

We also consider a slightly weaker notion for function privacy, called weak function privacy, which was introduced by Bishop, Jain, and Kowalczyk [21]. Basically, the weak function privacy is equivalent to the full function privacy, except the condition for adversarial queries. The weak function privacy relaxes the condition in Equation (1) to the following condition.

$$f_0(x_0) = f_0(x_1) = f_1(x_1) = f_1(x_0). \quad (2)$$

**Inner Product Encryption.** In this paper, we mainly focus on FE for inner-product functionality, which simply called Inner Product Encryption (IPE) [27]. In IPE scheme, the domain of plaintexts is an $n$-dimensional inner product space $M$ and each decryption key is associated with a function $f(\cdot) := \langle \cdot, y \rangle$ for some vector $y \in M$. The inner product functionality has numerous applications (See well-written previous works [27, 21, 28, 29, 22, 23].) In particular, function-private IPE scheme could be useful for statistical analysis on encrypted data, where the statistical analysis itself has sensitive information.

To the best of our knowledge, there are two approaches for fully function-private IPE schemes in the private key setting. One of which is to employ Brakerski and Segev’s general transformation from (non-function-private) FE schemes for general circuits [20]. Although the transformation itself is quite efficient in the sense that it just combines symmetric key encryption and functional encryption in a natural way, this approach requires heavy crypto tools such as indistinguishability obfuscation for realizing non-function-private FE for general circuits and therefore quite inefficient. The other approach is relatively practical; it directly constructs IPE scheme by using dual pairing vector spaces (DPVS) introduced by Okamoto and Takashima [30, 31]; Bishop, Jain, and Kowalczyk [21] proposed the first function-private IPE scheme and they proved the weak privacy
Table 1: Performance Comparison of IPE Schemes in the Standard Model

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Ciphertext</th>
<th>Dec. Key</th>
<th>Master Key</th>
<th>Privacy</th>
<th>Assum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BJK [21]</td>
<td>$(2n + 2)\ell_{G_1}$</td>
<td>$(2n + 2)\ell_{G_2}$</td>
<td>$(8n^2 + 8)\ell_{\mathbb{Z}_q}$</td>
<td>Weak</td>
<td>SXDH</td>
</tr>
<tr>
<td>DDM [22]</td>
<td>$(4n + 8)\ell_{G_1}$</td>
<td>$(4n + 8)\ell_{G_2}$</td>
<td>$(8n^2 + 12n + 28)\ell_{\mathbb{Z}_q}$</td>
<td>Full</td>
<td>SXDH</td>
</tr>
<tr>
<td>TAO [34]</td>
<td>$(2n + 5)\ell_{G_1}$</td>
<td>$(2n + 5)\ell_{G_2}$</td>
<td>$(4n^2 + 18n + 20)\ell_{\mathbb{Z}_q}$</td>
<td>Full</td>
<td>XDLIN</td>
</tr>
<tr>
<td>BJK [21]+LV Trans [35]</td>
<td>$(4n + 2)\ell_{G_1}$</td>
<td>$(4n + 2)\ell_{G_2}$</td>
<td>$(32n^2 + 8)\ell_{\mathbb{Z}_q}$</td>
<td>Full</td>
<td>SXDH</td>
</tr>
<tr>
<td>Ours 1</td>
<td>$(n + 4)\ell_{G_1}$</td>
<td>$(n + 4)\ell_{G_2}$</td>
<td>$(4n + 4)\ell_{\mathbb{Z}_q}$</td>
<td>Weak</td>
<td>SXDH</td>
</tr>
<tr>
<td>Ours 2</td>
<td>$(2n + 8)\ell_{G_1}$</td>
<td>$(2n + 8)\ell_{G_2}$</td>
<td>$(6n + 4)\ell_{\mathbb{Z}_q}$</td>
<td>Full</td>
<td>SXDH</td>
</tr>
</tbody>
</table>

All schemes employ asymmetric bilinear maps over two groups $G_1$ and $G_2$ of order $q$. $n$: the dimension of vector, $\ell_{G}$: the bit length to represent an element in group $G$
As aforementioned, there are only four realized function-private IPE schemes BJK-IPE and DDM-IPE prior to our IPE scheme in the standard model. All these schemes essentially employ the information-theoretic property of the DPVS which is implemented with asymmetric bilinear maps. Contrary to previous schemes, we achieve function-private IPE schemes via a different technique; that is, we do not use DPVS but a generalization of Brakerski and Segev transformation from (non-function-private) FE for general functionality [20]. Our approach consists of two steps. First, we generalize the Brakerski-Segev transformation and then analyze the security of our generalizations in several levels of function privacy. We believe that our transformations are of independent interest since our approach may be applicable to other functionalities. We will consider the detailed advantages of our transformations later. In the second step, we modify an existing non-function-private IPE scheme and also propose a new FE scheme for multi-exponentiation, to be well harmonized with our transformations. Then, by applying our transformation to these FE schemes, we directly obtain new practical function-private IPE schemes. Let us explain overviews of two our steps below.

**Step 1 - General Transformation.** We first propose two transformations Trans1 and Trans2 from a message-private FE scheme FE1 for functionality $F$ to a function-private FE for the same functionality. In our transformations, we use another message-private functional encryption FE2 for specific functionality, which is related to decryption circuit of FE1. We give a pictorial description of our first transformation Trans1 in Figure 1. Our second transformation Trans2 is obtained by applying the Naor-Yung double encryption methodology [36] to Trans1.

The proposed transformations Trans1 and Trans2 are generalizations of Brakerski-Segev.
erski and Segev’s basic\textsuperscript{2} and final transformations, respectively [20].\textsuperscript{3} Let us briefly explain how to consider the Brakerski-Segev transformation as an instance of ours. We focus on their basic scheme since both their final scheme and our Trans2 are obtained by applying the Naor-Yung double encryption methodology to their basic scheme and our Trans1, respectively. One difference between ours and their basic scheme is that Brakerski and Segev use a semantically secure symmetric encryption scheme instead of \text{FE1} in Trans1. In fact, a semantically secure symmetric encryption scheme can be utilized as a functional encryption scheme for arbitrary functionality, in particular, secure against non-key query attack.\textsuperscript{4} More precisely, given a symmetric encryption scheme \( (S.\text{Setup}, S.\text{Enc}, S.\text{Dec}) \), an FE scheme \( (\text{FE.\text{Setup}}, \text{FE.\text{Enc}}, \text{FE.\text{KG}}, \text{FE.\text{Dec}}) \) for arbitrary functionality \( F \) can be constructed as follows.

\[
\begin{align*}
\text{FE.\text{Setup}} &= S.\text{Setup}(1^\lambda) \rightarrow \text{mk}, & \text{FE.\text{Enc}} &= S.\text{Enc}(\text{mk}, x) \rightarrow \text{ct}_x, \\
\text{FE.\text{KG}}(\text{mk}, f \in F) & \rightarrow \text{sk}_f = \{\text{mk}, f\}, & \text{FE.\text{Dec}}(\text{sk}_f, \text{ct}_x) & \rightarrow f(S.\text{Dec}(\text{mk}, \text{ct}_x)).
\end{align*}
\]

The semantic security of the symmetric encryption scheme directly implies the message privacy of the FE scheme against non-key query attack. Therefore, the Brakerski-Segev scheme can be considered as an instantiation of our transformation (of course, only if the security against non-key query attack is enough for \text{FE1}).

Our analysis shows that several levels of function privacy can be achieved via our transformations. Informally, we achieve that

1. if \text{FE1} and \text{FE2} satisfy IND-based message privacy (against adaptive query attack), then the result of Trans1 achieves IND-based weak function privacy.\textsuperscript{5}

2. if \text{FE1} is SIM-based message-private against non-key query attack and \text{FE2} satisfy SIM-based message privacy, then the result of Trans1 achieves SIM-based full function privacy.\textsuperscript{6}

3. if \text{FE1} is IND-based message-private against non-key query attack and \text{FE2}
satisfies IND-based message privacy, then the result of Trans2 achieves IND-based full function privacy.

Here, for the second and third results, message-private FE1 against non-key query attack is enough. As for the first result, we require FE1 to satisfy the message privacy against adaptive (key and encryption) query attack.

Our generalizations have several advantages in terms of both security and applicability. Brakerski and Segev showed that their basic scheme is not function-private, and then they applied the Naor-Yung double encryption methodology [36] to the basic scheme for achieving IND-based function privacy. By extending Brakerski and Segev’s argument about their basic transformation, we can show that their basic scheme does not achieve even (IND-based) weak privacy. Contrary to their basic transformation, we show that our first transformation Trans1 achieves (IND-based) weak function privacy if FE1 satisfies the message-privacy against adaptive query attack. Furthermore, we prove that Trans1 satisfies SIM-based full function privacy if the underlying FE scheme satisfies SIM-based message privacy. At first glance this result looks counterintuitive since Trans1 (and Brakerski and Segev’s basic transformation) can achieve a stronger security notion (that is, SIM-based full function privacy), but not achieve a weaker security notion (that is, IND-based message privacy).

However, this is due to the underlying functional encryptions; the SIM-based security notion implies the IND-based security notion and there exists a class of schemes that satisfy IND-based security but not SIM-based security. Although the transformation works well for SIM-based secure schemes, but it does not work for such the class of (IND-secure but not SIM-secure) schemes. Therefore, there is no contradiction in the above results.

To obtain a function-private FE for functionality \( F \), the Brakerski-Segev transformation requires a message-private FE for more expressive functionality than \( F \). (Roughly speaking, the underlying FE scheme has to support a composite function of \( F \) and decryption algorithm of the underlying symmetric encryption scheme.) In our transformation, we requires a message-private FE scheme FE1 for the same functionality \( F \) and another message-private FE for decryption circuit of the first FE scheme FE2. Therefore, if there exists an efficient realization of FE2 for the decryption circuit of FE1, then we can realize an efficient function-private FE scheme for \( F \). In the next paragraph, we present our idea for a pair of FE schemes satisfying the above requirements, in particular, for inner product functionality.

**Step 2 - FE1 and FE2 Constructions for Inner Product.** As we mentioned above, in order to apply our first transformation Trans1 for inner product functionality, we need two message-private FE schemes as building blocks; the first scheme FE1 is an IPE whose decryption is relatively simple and the second scheme FE2 is an FE scheme whose functionality supports the decryption of FE1.

To this end, we begin with Agrawal et al.’s group-based IPE (ALS-IPE) scheme [28]. The decryption process of ALS-IPE consists of two steps; (1) a
multi-exponentiation and then (2) a solving a discrete log problem (DLP).\footnote{The final process of solving DLP imposes a restriction on the distribution of message space. However, we note that all previous function-private IPE schemes \cite{21, 22, 23} has the same process of solving DLP in decryption algorithms. Nevertheless, there are still numerous applications under this restriction. See \cite{21, 22, 23}.} To the best of our knowledge, unfortunately, there is no practical FE construction for such operations, except FE schemes for general circuits, which is inefficient as we mentioned before. To resolve this, we make several tricks, essentially for modifying the decryption algorithm so that we can construct another efficient FE scheme for the decryption circuit of the first FE scheme. First, we remove the process of solving a DLP procedure at the moment. (We will recover it after applying transformation.) Second, we add an asymmetric bilinear map operation at the end of decryption process.\footnote{Indeed, the modified scheme is no longer IPE scheme. It outputs $e(g, \bar{g})^{(x,y)}$ instead of $(x,y)$, where $x$ and $y$ are vectors in input ciphertext and decryption key and $g$ is a generator of $G_1$. However, it does not matter to our purpose, function-private IPE. The resulting scheme transformed by $\text{Trans1}$ will also output $e(g, \bar{g})^{(x,y)}$. Then, we can add a post process of solving a DLP based on $e(g, \bar{g})$ to the decryption algorithm of the transformed scheme.} This process seemingly redundant, but necessary to combine with our second FE scheme.\footnote{Indeed, we could not construct an FE scheme for multi-exponentiation without bilinear map. If one can construct an efficient FE scheme for such functionality, then we can remove a bilinear map from the first FE scheme.} Next, we propose another FE scheme for multi-exponentiation with bilinear map, by again modifying the ALS-IPE scheme.

In order to apply our second transformation $\text{Trans2}$, we need another idea to consider the decryption process with “double encrypted” ciphertexts of the modified ALS-IPE scheme. (Recall that $\text{Trans2}$ applies the Naor-Yung double encryption methodology to $\text{Trans1}$.) A problem of the decryption process with double encrypted ciphertexts is that this process contains a branching statement. (For the detailed functionality, see Figure 5.) In general, we don’t know how to represent this process by a simple circuit, as much as we can realize an FE scheme for it. Instead, we show that if the decryption algorithm of $\text{FE1}$ satisfies some mild conditions, which are also satisfied by our modification of the ALS-IPE scheme, then there exists an equivalent decryption process not containing branching statements. The resulting decryption process becomes a multi-exponentiation with bilinear. Therefore, similar to $\text{Trans1}$, we can utilize the proposed FE for multi-exponentiation with bilinear.

1.2. Road Map

In the next section, we provide short preliminaries including definitions of FE scheme, complexity assumptions, weak/full function privacy. In Section 3 and 4, we present two our transformations $\text{Trans1}$ and $\text{Trans2}$, respectively, where $\text{Trans2}$ is obtained by applying Naor-Yung double encryption methodology to $\text{Trans1}$. Furthermore, we also provide analyses for two our transformations in the corresponding sections, respectively. In Section 5, we propose several modifications of an existing IPE scheme to construct non-function-private FE schemes.
that could be appropriately combinable with two our transformations. Finally, the resulting schemes are two practical function-private IPE scheme, which have shortest master secret key in comparison with previous works.

2. Preliminaries

We first define notations used in the paper. For a message \( x \) and a function \( f \), \(|x|\) and \(|f|\) denote the bit-length to represent \( x \) and \( f \), respectively, where such the representation is uniquely fixed (by the cryptosystem). For example, let \( x \) be a vector in \( \mathbb{Z}_q^n \) and \( f \) be an inner-product function with a length-\( n \) vector \( y \in \mathbb{Z}_q^n \), that is, \( f(\cdot) := \langle \cdot, y \rangle \). Then, we can specify \( x \) by \( n\lceil \log p \rceil \) bits and \( f \) by \( y \), so that \(|x| = |f| = |y| = n\lceil \log p \rceil \).

2.1. Private-Key Functional Encryption

We provide the syntax and the security notion for functional encryption schemes, in particular, in the private-key setting. We basically follows definitions of Brakerski and Segev [20] and Bishop, Jain, and Kowalczyk [21].

**Definition 1** (Private-Key Functional Encryption). A private-key functional encryption \( \text{FE} \) over a message space \( \mathcal{M} \) and a function space \( \mathcal{F} \) consists of the following four algorithms \( \{\text{Setup}, \text{Enc}, \text{KG}, \text{Dec}\} \).

- \( \text{Setup}(1^\lambda, (\mathcal{F}, \mathcal{M})) \) takes as input \( \mathcal{F}, \mathcal{M} \), and the unary representation \( 1^\lambda \) of the security parameter \( \lambda \in \mathbb{N} \) and outputs a master secret key \( \text{mk} \) and a system parameter \( \text{pp} \). (If \( (\mathcal{M}, \mathcal{F}) \) is clear from the context, we omit it from the input of \( \text{Setup} \).)

- \( \text{Enc}(\text{mk}, x) \) takes as input \( \text{mk} \) and a message \( x \in \mathcal{M} \) and output a ciphertext \( \text{ct} \).

- \( \text{KG}(\text{mk}, f) \) takes as input \( \text{mk} \) and a function \( f \in \mathcal{F} \) and output a decryption key \( \text{sk}_f \).

- \( \text{Dec}(\text{pp}, \text{ct}, \text{sk}) \) takes as input \( \text{pp}, \text{ct} \) and \( \text{sk} \) output a result satisfying the following correctness condition.

\[
\text{We usually omit pp from the output of Setup algorithm and the input of Dec algorithm if it is clear from the context. For } \text{Setup}(1^\lambda) \rightarrow \text{mk}, \forall x \in \mathcal{M}, \forall f \in \mathcal{F}, \text{there exists a negligible function } \varepsilon \text{ in } \lambda \text{ such that }
\]

\[
\Pr [\text{Dec(Enc(mk, x), KG(mk, f)) \rightarrow f(x)}] \geq 1 - \varepsilon(\lambda),
\]

where the probability goes over all internal randomness used in \( \text{Setup} \), \( \text{Enc} \), \( \text{KG} \), and \( \text{Dec} \).

**Remark 1.** In Definition 1, we define private-key functional encryption over a message space \( \mathcal{M} \) and a function space \( \mathcal{F} \). In some applications, we may need to define \( \mathcal{M} \) and \( \mathcal{F} \) simultaneously with the other parameters. For example, an
inner-product between two vectors in $\mathbb{Z}_q^n$ for some $n$, where $q$ is a prime larger than $2^\lambda$. To cover these applications, we may relax the input $(M, F)$ of Setup algorithm as additional information to define $(M, F)$, instead of the exact form of $M$ and $F$. As for the inner-product functionality, the dimension of vectors $n$ can be taken as input of Setup and $\mathbb{Z}_q$ is defined as output of Setup. Then, $M$ and $F$ are defined as $\mathbb{Z}_q^n$ and inner-product over $\mathbb{Z}_q$, respectively.

We define message privacy of FE schemes in the private-key setting, which is the strongest indistinguishability-based definition given in [20].

**Definition 2** (Message Privacy). A private-key functional encryption FE over a message space $M$ and a function space $F$ is message private (denoted by $\text{IND}_{\text{dp}}^\text{MP}$) if for any probabilistic polynomial-time algorithm $A$, which has two oracle accesses $\text{KG}(\text{mk}, f)$ and $\text{Enc}_0(\text{mk}, x_0, x_1)$ for her choices $f \in F$ and $x_0, x_1 \in M$ with the restriction $f(x_0) = f(x_1)$ and $|x_0| = |x_1|$, there exists a negligible function $\varepsilon$ in $\lambda$ such that

$$\text{Setup}(1^\lambda) \rightarrow \text{mk}$$

and

$$\Pr[A^{\text{KG}(\text{mk}, \cdot), \text{Enc}_0(\text{mk}, \cdot)}(\lambda) \rightarrow 1] - \Pr[A^{\text{KG}(\text{mk}, \cdot), \text{Enc}_1(\text{mk}, \cdot)}(\lambda) \rightarrow 1] \leq \varepsilon,$$

where the probability is taken over all the randomness used in algorithms/oracles of FE and $A$.

We also define a weaker notion of message privacy against non-key query attack that is equivalent to the message privacy in Definition 2, except that the adversary is given encryption oracle only. That is, the adversary is not allowed to issue decryption key queries for any functions in the non-key query attack model. For the sake of the simplicity, we use a notation $\text{IND}_{\text{non}}^\text{MP}$ to denote the IND-based message privacy against non-key query attack.

Next, we give two definitions for function privacy used for our IPE proposals as well as our generic transformations. For the following full function privacy of private-key FE schemes, we use Agrawal et al.’s definition [26] (and so Brakerski et al.’s definition [20]).

**Definition 3** (Full Function Privacy). A private-key functional encryption FE over a message space $M$ and a function space $F$ is fully function private (denoted by $\text{IND}_{\text{gfp}}^\text{FFP}$) if for any probabilistic polynomial-time algorithm $A$, which has two oracle accesses $\text{KG}_0(\text{mk}, f_0, f_1) := \text{KG}(\text{mk}, f_0)$ and $\text{Enc}_0(\text{mk}, x_0, x_1) := \text{Enc}(\text{mk}, x_0)$ for her choices $f_0, f_1 \in F$ and $x_0, x_1 \in M$ with the restriction $f_0(x_0) = f_1(x_1)$, $|x_0| = |x_1|$, and $|f_0| = |f_1|$, there exists a negligible function $\varepsilon$ in $\lambda$ such that

$$\text{Setup}(1^\lambda) \rightarrow \text{mk}$$

and

$$\Pr[A^{\text{KG}_0(\text{mk}, \cdot), \text{Enc}_0(\text{mk}, \cdot)}(\lambda) \rightarrow 1] - \Pr[A^{\text{KG}_1(\text{mk}, \cdot), \text{Enc}_1(\text{mk}, \cdot)}(\lambda) \rightarrow 1] \leq \varepsilon,$$

where the probability is taken over all the randomness used in algorithms/oracles of FE and $A$. 

10
We also define a weaker notion of function privacy, which is firstly introduced by Bishop, Jain, and Kowalczyk [21]. Bishop et al. used this notion for the security of their functional encryption scheme with inner-product functionality. We extend Bishop et al.'s notion for inner-product functionality to general functionality.

**Definition 4** (Weak Function Privacy). A private-key functional encryption \( FE \) over a message space \( \mathcal{M} \) and a function space \( \mathcal{F} \) is weakly function private (denoted by \( \text{IND}_{\text{wFP}} \)) if for any probabilistic polynomial-time algorithm \( \mathcal{A} \), which has two oracle accesses \( \mathcal{KG}(\mathsf{mk}, f_0, f_1) := \mathcal{KG}(\mathsf{mk}, f_b) \) and \( \mathcal{Enc}_\mathsf{c}(\mathsf{mk}, x_0, x_1) := \mathcal{Enc}(\mathsf{mk}, x_b) \) for her choices \( f_0, f_1 \in \mathcal{F} \) and \( x_0, x_1 \in \mathcal{M} \) with the restriction \( f_0(x_0) = f_1(x_1) = f_0(x_1) = f_1(x_0) \), \( |x_0| = |x_1|, \) and \( |f_0| = |f_1| \), there exists a negligible function \( \varepsilon \) in \( \lambda \) such that

\[
\left| \Pr \left[ \mathcal{A}_{\mathcal{KG}}(\mathsf{mk}, \cdot), \mathcal{Enc}_\mathsf{c}(\mathsf{mk}, \cdot) (\lambda) \rightarrow 1 \right] - \Pr \left[ \mathcal{A}_{\mathcal{KG}}(\mathsf{mk}, \cdot), \mathcal{Enc}_\mathsf{c}(\mathsf{mk}, \cdot) (\lambda) \rightarrow 1 \right] \right| \leq \varepsilon,
\]

where the probability is taken over all the randomness used in algorithms/oracles of \( FE \) and \( \mathcal{A} \).

**2.2. Background on Mathematical Structure and Assumption**

We provide a background on mathematical structure and complexity assumption used in our IPE construction and its security analysis.

**Definition 5** (Asymmetric Bilinear Map). We say an algorithm \( \mathcal{G}_{\mathcal{ABG}} \) is an asymmetric bilinear group generator if it takes a security parameter \( \lambda \) as input and outputs \( (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \) where \( \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \) are groups of prime order \( q \) of \( \lambda \)-bit length and a map \( e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T \) satisfies the following properties:

1. (bilinearity) \( \forall a, b \in \mathbb{Z}_q \) and \( g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2, e(g_1^a, g_2^b) = e(g_1, g_2)^{ab} \)

2. (non-degeneracy) \( \exists g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2, e(g_1, g_2) \) is a generator of \( \mathbb{G}_T \).

We assume that group operations in \( \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \) and bilinear operation are efficiently computable in polynomial time with respect to \( \lambda \).

**Definition 6** (Symmetric eXternal Diffie-Hellman assumption (SXDH)). Let \( (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \mathcal{G}_{\mathcal{ABG}}(1^\lambda) \) be an asymmetric bilinear group. The Decisional Diffie-Hellman (DDH) problem in \( \mathbb{G}_1 \) is to distinguish between the two distributions \( \mathcal{D}_0 := \{(g_1, g_1^a, g_1^b, g_2) : g_1, g_2 \overset{\$}{\leftarrow} \mathbb{G}_1, g_2 \overset{\$}{\leftarrow} \mathbb{G}_2, a, b \overset{\$}{\leftarrow} \mathbb{Z}_q\} \) and \( \mathcal{D}_1 := \{(g_1, g_1^a, g_1^{a+b+r}, g_2) : g_1, g_2 \overset{\$}{\leftarrow} \mathbb{G}_1, g_2 \overset{\$}{\leftarrow} \mathbb{G}_2, a, b, r \overset{\$}{\leftarrow} \mathbb{Z}_q\} \). We can similarly define the DDH problem in \( \mathbb{G}_2 \) by changing roles between \( g_1 \) and \( g_2 \). We say the Symmetric eXternal Diffie-Hellman (SXDH) assumption holds if for any polynomial time adversary \( \mathcal{A} \), both advantages of \( \mathcal{A} \) in solving the DDH problem in \( \mathbb{G}_1 \) and the DDH problem in \( \mathbb{G}_2 \) are negligible in \( \lambda \), where the advantage \( \text{Adv}_{\mathcal{A}, \mathbb{G}_1}^{\text{DDH}}(\lambda) \) is defined as follows.

\[
\text{Adv}_{\mathcal{A}, \mathbb{G}_1}^{\text{DDH}}(\lambda) := \left| \Pr[\mathcal{A}(\mathcal{D}_\beta) \rightarrow \beta] - \frac{1}{2} \right|.
\]
Setup(1^λ): Run Setup1(1^λ) → mk1 and Setup2(1^λ) → mk2. Output mk = \{mk1, mk2\}.

Enc(mk, x): Run Enc1(mk1, x) → ct1, and then KG2(mk2, Fct1) → sk2_{Fct1}, where the function Fct1 is defined in Figure 3. Output ct = sk2_{Fct1}.

KG(mk, f): Run KG1(mk1, f) → sk1_f, and then Enc2(mk2, sk1_f) → ct2. Output sk = ct2.

Dec(ct, sk): Run Dec2(sk, ct) and output its result.

\[ F_{ct1}(\cdot) = Dec1(ct1, \cdot) \]

Figure 2: The 1st Transformation Trans1

Figure 3: The function F_{ct1}(\cdot)

where the probability goes over all the randomness used in A and the generation \( \beta \leftarrow \{0, 1\} \) and \( D_\beta \).

3. Our Transformation for Weak Function Privacy

In this section, we first present our first transformation Trans1. We then analyze security of the transformed FE scheme; we basically utilize Trans1 for achieving IND_{WFP}^\text{ad} security. Interestingly, we also show that Trans1 can achieve the simulation-based full function privacy if the underlying FE schemes satisfy the simulation-based message privacy.

3.1. Description of Trans1

Let FE1 and FE2 be private-key FE schemes over message spaces \( \mathcal{M}_1 \) and \( \mathcal{M}_2 \), respectively, and function spaces \( \mathcal{F}_1 \) and \( \mathcal{F}_2 \), respectively, where each scheme consists of four algorithms \{Setup_i, Enc_i, KG_i, Dec_i\} for \( i \in \{1, 2\} \), respectively. We show that if \( \mathcal{F}_2 \) contains some specific forms of functions defined in Figure 3, then we can transfer FE1 to a private-key FE scheme \( FE = \{\text{Setup, Enc, KG, Dec}\} \) over a message space \( \mathcal{M} = \mathcal{M}_1 \) and a function space \( \mathcal{F} = \mathcal{F}_1 \). We provide the precise description of Trans1 in Figure 2.

Assumption on the output of KG1. We assume that for each \( f \in \mathcal{F}_1 \), the output sk1_f of the key generation algorithm KG1(mk1, f) has the same bit-length. If FE1 does not satisfy this assumption, we can easily modify KG1 and Dec1; pad empty symbols to the output of KG1 and remove it at the beginning of Dec1.
Correctness. Let $\text{Enc}(mk, x) \rightarrow ct$ and $\text{KG}(mk, f) \rightarrow sk$. Then, $ct = sk_{F_{ct1}}$ and $sk = ct2$, where $F_{ct1}$ is defined as in Figure 3 and $\text{Enc2}(mk2, sk1_f) \rightarrow ct2$. If we run $\text{Dec2}(sk, ct)$, it is equivalent to running $\text{Dec2}(ct2, sk_{F_{ct1}})$ for an encryption $ct2$ of $sk1_f$ and a decryption key $sk_{F_{ct1}}$ of the function $F_{ct1}$. Therefore, by the correctness of $\text{FE}_2$, we obtain $F_{ct1}(sk1_f) = \text{Dec1}(ct1, sk1_f) \rightarrow f(x)$.

3.2. IND-based Security of Transformed FE Schemes

The IND-based security of the resulting scheme via $\text{Trans1}$ is given in the following theorem:

**Theorem 1.** If both $\text{FE}_1$ and $\text{FE}_2$ are IND$_{ad}^{MP}$-secure, then the resulting scheme $\text{FE}$ via $\text{Trans1}$ is IND$_{ad}^{MP}$-secure.

**Proof.** We prove the theorem by the standard hybrid argument. First, we define a series of games as follows:

$\text{Game}_0$: In this game, $A$ has oracle accesses to $\text{KG}_0(mk, \cdot, \cdot)$ and $\text{Enc}_0(mk, \cdot, \cdot)$.

$\text{Game}_1$: This is a game with oracles $\text{KG}_0(mk, \cdot, \cdot)$ and $\text{Enc}_1(mk, \cdot, \cdot)$.

$\text{Game}_2$: In this game, the oracles $\text{KG}_1(mk, \cdot, \cdot)$ and $\text{Enc}_1(mk, \cdot, \cdot)$ are used for all $A$’s queries.

We will prove that the difference of $A$’s output distribution between any consecutive games is negligible. Then, by the triangle inequality, we conclude that $A$’s advantage is negligible.

**Difference between $\text{Game}_0$ and $\text{Game}_1$.** We assume that there exists an adversary $A$ having a non-negligible advantage in distinguishing two game $\text{Game}_0$ and $\text{Game}_1$. More precisely, for some non-negligible function $\epsilon_0$,

$$\left| \Pr [A^{\text{KG}_0(mk, \cdot, \cdot), \text{Enc}_0(mk, \cdot, \cdot)}(\lambda) \rightarrow 1] - \Pr [A^{\text{KG}_0(mk, \cdot, \cdot), \text{Enc}_1(mk, \cdot, \cdot)}(\lambda) \rightarrow 1] \right| > \epsilon_0.$$

Then, we show that there exists an adversary $B$ breaking the message privacy IND$_{ad}^{MP}$ of $\text{FE}_1$. We first describe $B$ and analyze later. $B$ generates $\text{FE}_2$ by itself; that is, it runs $\text{Setup2}(1^n) \rightarrow mk2$ and uses $mk2$ for $\text{Enc2}$ and $\text{KG2}$ algorithms. Note that $B$ has oracle accesses $\text{Enc}_1(b(mk1, \cdot, \cdot))$ and $\text{KG1}(mk1, \cdot, \cdot)$, where $b$ is randomly chosen by the challenger.

$\text{KG}_0(mk, \cdot, \cdot)$: On $A$’s query $(f_0, f_1)$, $B$ sends $f_0$ to its own oracle $\text{KG1}(mk1, \cdot, \cdot)$ oracle and receive $sk1_{f_0}$. Then $B$ runs $\text{Enc2}(mk2, sk1_{f_0}) \rightarrow ct$ and returns the result $sk = ct2$.

$\text{Enc}_0(b(mk, \cdot, \cdot))$: On $A$’s query $(x_0, x_1)$, $B$ delivers it to its own oracle $\text{Enc}_0(mk1, \cdot, \cdot) \rightarrow ct1^*$. Then, $B$ runs $\text{KG2}(mk2, F_{ct1^*}) \rightarrow sk_{F_{ct1^*}}$ and returns the result $ct = sk_{F_{ct1^*}}$. 

13
If $A$ outputs her guess $b'$, then $B$ sends $b'$ to the challenger.

Now, we analyze $B$’s advantage. It is trivial to check the correctness of oracle answers; KG0’s output is well distributed regardless of $b$ since mk1 and mk2 are normally used by KG1 oracle and $B$, respectively. Enc0’s output is also well distributed according to the challenger’s choice $b$. Next, we have to show that all $B$’s queries hold the necessary condition $f(x_0) = f(x_1)$. This directly comes from the condition $f_0(x_0) = f_0(x_1) = f_1(x_0) = f_1(x_1)$ for weak function privacy of FE. Therefore, $B$’s advantage in distinguishing two oracles Enc1 and Enc2 is exactly the same as $A$’s advantage in distinguishing two games Game1 and Game2.

**Difference between Game1 and Game2.** Similar to the above proof, we also assume that there exists an adversary $A$ having a non-negligible advantage in distinguishing two game Game1 and Game2. More precisely, for some non-negligible function $\varepsilon_1$,

$$\left| \Pr[A^{KG_0(mk,\cdot,\cdot)}(\cdot) \rightarrow 1] - \Pr[A^{KG_1(mk,\cdot,\cdot)}(\cdot) \rightarrow 1] \right| > \varepsilon_1.$$ 

Then, we show that there exists an adversary $B$ breaking the message privacy IND_{mp} of FE2. We first describe $B$ and analyze later, $B$ generates FE1 by itself; that is, it runs Setup1($1^\lambda$) $\rightarrow$ mk1 and uses mk1 for Enc1 and KG1 algorithms. Note that $B$ has its own oracle accesses Enc2b(mk2, $\cdot, \cdot$) and KG2(mk2, $\cdot, \cdot$), where $b$ is randomly chosen by the challenger.

**KG0(mk, $\cdot, \cdot$):** On $A$’s query $(f_0, f_1), B$ runs KG1(mk1, $f_0$) $\rightarrow$ sk1$_{f_0}$ and KG1(mk1, $f_1$) $\rightarrow$ sk1$_{f_1}$. Then $B$ sends them to its own oracle Enc2b(mk2, $\cdot, \cdot$) and receives ct2. Finally, $B$ returns sk = ct2 to $A$.

**Enc1(mk, $\cdot, \cdot$):** On $A$’s query $(x_0, x_1), B$ runs Enc1(mk1, $x_1$) $\rightarrow$ ct1, and delivers it to its own oracle KG2(mk2, ct1) $\rightarrow$ sk2$_{ct1}, B$ returns the result ct = sk2$_{ct1}$ to $A$.

If $A$ outputs her guess $b'$, then $B$ sends $b'$ to the challenger.

Now, we analyze $B$’s advantage. It is trivial to check the correctness of oracle answers; Enc1’s output is well distributed regardless of $b$ since mk1 and mk2 are normally used by $B$ and KG2 oracle, respectively. KG0’s output is also well distributed according to the challenger’s choice $b$. Next, we should show that all $B$’s oracle queries satisfy the necessary conditions. $B$ issues message queries (sk1$_{f_0}$, sk1$_{f_1}$) and function queries Fct1, where ct1 is an encryption of $x_1$, so that we have to check whether $F_{ct1}(sk1_{f_0}) = F_{ct1}(sk1_{f_1})$ holds for all such queries. By definition of Fct1, $F_{ct1}(sk1_{f_0}) = Dec1(ct1, sk1_{f_0}) = f_0(x_1)$ and $F_{ct1}(sk1_{f_1}) = Dec1(ct1, sk1_{f_1}) = f_1(x_1)$. By the necessary condition for queries of weak function privacy, we know that $f_0(x_1) = f_1(x_1)$ for all $A$’s queries $(f_0, f_1)$ and $(x_0, x_1)$. Furthermore, our assumption on the output of KG1, we know that $|sk1_{f_0}| = |sk1_{f_1}|$, so that all $B$’s queries satisfy the necessary conditions for queries of message privacy. Therefore, $B$’s advantage in distinguishing
two oracles $\text{Enc}_0$ and $\text{Enc}_1$ is exactly the same as $A$’s advantage in distinguishing two games $\text{Game}_1$ and $\text{Game}_2$.

**Remark 2.** We note that the basic version of the Brakerski-Segev transformation, which is what they called ‘a failed attempt’, does not satisfy the weak function privacy. As explained in the introduction, their basic scheme is an instantiation of our transformation, where $\text{FE}_1$ is implemented by using a semantically secure symmetric encryption scheme. In (the proof of) Theorem 1, $\text{IND}_{\text{adv}}$ security seems necessary for $\text{FE}_1$. (If not, the simulator cannot correctly simulate $\text{KG}$ oracle in the proof of the difference between $\text{Game}_i$ and $\text{Game}_1$.) However, $\text{FE}$ schemes implemented by a symmetric encryption scheme can achieve only $\text{IND}_{\text{non}}$ security.

### 3.3. Simulation-based Security of Transformed FE Schemes

We consider the SIM-based security of $\text{FE}$ via $\text{Trans}_1$. Intuitively, the SIM-based security for message privacy of $\text{FE}$ captures the ideal security that any information the adversary is able to learn from the decryption keys and ciphertexts can be obtained by a simulator with access only to the decryption key queries and the outputs of the functionality, i.e., $f(x)$. For function privacy, SIM-security even requests a simulator to generate the decryption key and ciphertext with access only to the outputs of the functionality.

In analyzing of SIM-based security of our construction, we also separate abilities of adversaries into two cases, as we did for IND-based security notion; adversaries have adaptive access to decryption key generation and encryption oracles or no access at all to decryption key generation oracle. When a functional encryption $\text{FE}$ achieves simulation-based message privacy against adaptive query, we say that the scheme satisfies *simulation-based message privacy against adaptive query* ($\text{SIM}_{\text{ad}}$ - secure, for short). In the case with non-key query, we say that the scheme satisfies *simulation-based message privacy against non-key query* ($\text{SIM}_{\text{non}}$ - secure, for short). Note that $\text{SIM}_{\text{non}}$ - security only cares about the semantic security of $\text{Enc}$ of $\text{FE}$. When a functional encryption $\text{FE}$ furthermore achieves simulation-based function privacy against adaptive query, we say that the scheme satisfies *simulation-based function privacy against adaptive query* ($\text{SIM}_{\text{ad}}^{\text{FFP}}$ - secure, for short). For the formal definitions for the above SIM-security notions, refer to Appendix Appendix A, which we borrow from Kim et al. [23]. We now analyze SIM-security of the transformed schemes via $\text{Trans}_1$ in the following theorems. We refer the reader to Appendix Appendix B for the proof.

**Theorem 2.** If $\text{FE}_1$ is $\text{SIM}_{\text{non}}$- secure and $\text{FE}_2$ is $\text{SIM}_{\text{ad}}$- secure, then the transformed scheme $\text{FE}$ via $\text{Trans}_1$ is $\text{SIM}_{\text{ad}}^{\text{FFP}}$- secure.

**Remark 3.** In the above theorem, $\text{SIM}_{\text{ad}}^{\text{FFP}}$ - security of the resulting scheme only relies on the semantic security of $\text{Enc}$ in $\text{FE}_1$ differently from $\text{FE}_2$. This points out that the basic version of the Brakerski-Segev transformation also generates $\text{SIM}_{\text{ad}}^{\text{FFP}}$ - secure $\text{FE}$. 

15
Setup($\lambda$): Run Setup1($\lambda$) twice and obtain mk1 and $\hat{mk}_1$. Run Setup2($\lambda$) → mk2. Output $mk = \{mk_1, \hat{mk}_1, mk_2\}$.

Enc($mk, x$): Run Enc1($mk_1, x$) → ct1 and Enc1($\hat{mk}_1, x$) → $\hat{ct}_1$, and then run KG2($mk_2, F_{ct1, \hat{ct}_1}$) → sk2$\leftarrow F_{ct1, \hat{ct}_1}$. The function $F_{ct1, \hat{ct}_1}$ is defined in Figure 5. Output $ct = sk2\leftarrow F_{ct1, \hat{ct}_1}$.

KG($mk, f$): Run KG1($mk_1, f$) → sk1$\leftarrow F_{ct1, \hat{ct}_1}$, and then run Enc2($mk_2, (sk1, \perp)$) → ct2. Output $sk = ct2$.

Dec($ct, sk$): Run Dec2($sk, ct$) and output its result.

**Figure 4:** Fully Function-Private Functional Encryption

The domain of $F_{ct1, \hat{ct}_1}(\cdot, \cdot)$ is $(R \times \{\perp\}) \cup (\{\perp\} \times R)$, where $R$ is the range of KG1 and $\perp \in R$.

$$F_{ct1, \hat{ct}_1}(k, \hat{k}) \text{ outputs } \begin{cases} \perp & \text{if } (k, \hat{k}) = (\perp, \perp) \\ \text{Dec1($ct1, k$)} & \text{if } k \neq \perp \\ \text{Dec1($\hat{ct}_1, \hat{k}$)} & \text{if } \hat{k} \neq \perp \end{cases}$$

**Figure 5:** The function $F_{ct1, \hat{ct}_1}$

### 4. Our Transformation for Full Function Privacy

In this section, we present our second transformation Trans2 that is obtained by applying Naor-Yung double encryption technique [36] to Trans1. We then analyze security of transformed schemes under IND-based security notion.

#### 4.1. Description of Trans2

Similar to our weak function-private construction, we use two private-key functional encryption schemes FE1 and FE2, where each consists of four algorithms $\{\text{Setup$_i$, Enc$_i$, KG$_i$, Dec$_i$}\}$ for $i \in \{1, 2\}$.

**Assumption on the output of KG1.** Similar to weakly function-private FE construction, we assume that for each $f \in F_1$, the output of the key generation algorithm KG1(·, ·) has the same bit-length, regardless of the input. (We may employ an appropriate padding process.) In addition, we assume that $\perp$ is a special symbol with the same bit-length representation as other decryption keys sk$1_f$. 

---

16
Correctness. Let \( \text{Enc}(\text{mk}, x) \rightarrow \text{ct} \) and \( \text{KG}(\text{mk}, f) \rightarrow \text{sk} \). Then, \( \text{ct} = \text{sk}_2 \cdot F_{\text{ct}, \text{ct}1} \) and \( \text{sk} = \text{ct}2 \), where \( F_{\text{ct}, \text{ct}1} \) is defined as in Figure 3 and \( \text{Enc2}(\text{mk}_2, (\text{sk}_1, \bot)) \rightarrow \text{ct}2 \). If we run \( \text{Dec2}(\text{sk}, \text{ct}) \), it is equivalent to running \( \text{Dec2}(\text{ct}2, \text{sk}2 \cdot F_{\text{ct}, \text{ct}1}) \) for an encryption \( \text{ct}2 \) of \( (\text{sk}1, \bot) \) and a decryption key \( \text{sk}2 \cdot F_{\text{ct}, \text{ct}1} \) of the function \( F_{\text{ct}, \text{ct}1} \). Therefore, by the correctness of FE2, we obtain \( F_{\text{ct}, \text{ct}1}(\text{sk}1, \bot) = \text{Dec1}(\text{ct}1, \text{sk}1) \rightarrow f(x) \).

### 4.2. IND-base Security of Transformed FE Schemes

The IND-based security of the transformed FE schemes by \( \text{Trans2} \) is captured as follows:

**Theorem 3.** If \( \text{FE1} \) is IND\(_\text{non-mp} \)-secure and \( \text{FE2} \) is IND\(_\text{ad-mp} \)-secure, then the transformed scheme FE in Figure 4 is IND\(_\text{ad-ffp} \)-secure.

**Proof.** We begin with defining a series of games, in which where \( A \) has different oracle accesses. The \( i \)-th game is completely specified by its key generation oracle, denoted by \( \text{KG}^{(i)}(\text{mk}, \cdot, \cdot) \) and its encryption oracle, denoted by \( \text{Enc}^{(i)}(\text{mk}, \cdot, \cdot) \). In Table 2, we give the description of the oracle outputs in a series of games \( \text{Game}_0, \ldots, \text{Game}_4 \).

<table>
<thead>
<tr>
<th>Games</th>
<th>KG Query ((f_0, f_1))</th>
<th>Enc Query ((x_0, x_1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{Game}_0</td>
<td>( \text{KG}(\text{mk}_1, f_0) \rightarrow \text{sk}_1 \cdot f_0 \cdot \text{Enc}(\text{mk}_2, (\text{sk}_1, f_1, \bot)) )</td>
<td>( \text{Enc}(\text{mk}_1, x_0) \rightarrow \text{ct}1, \text{Enc}(\text{mk}_1, x_1) \rightarrow \text{ct}1, \text{KG}(\text{mk}<em>2, F</em>{\text{ct}, \text{ct}1}) )</td>
</tr>
<tr>
<td>\text{Game}_1</td>
<td>( \text{KG}(\text{mk}_1, f_0) \rightarrow \text{sk}_1 \cdot f_0 \cdot \text{Enc}(\text{mk}_2, (\text{sk}_1, f_1, \bot)) )</td>
<td>( \text{Enc}(\text{mk}_1, x_0) \rightarrow \text{ct}1, \text{Enc}(\text{mk}_1, x_1) \rightarrow \text{ct}1, \text{KG}(\text{mk}<em>2, F</em>{\text{ct}, \text{ct}1}) )</td>
</tr>
<tr>
<td>\text{Game}_2</td>
<td>( \text{KG}(\text{mk}_1, f_1) \rightarrow \text{sk}_1 \cdot f_1 \cdot \text{Enc}(\text{mk}_2, (\bot, \text{sk}_1, f_1)) )</td>
<td>( \text{Enc}(\text{mk}_1, x_0) \rightarrow \text{ct}1, \text{Enc}(\text{mk}_1, x_1) \rightarrow \text{ct}1, \text{KG}(\text{mk}<em>2, F</em>{\text{ct}, \text{ct}1}) )</td>
</tr>
<tr>
<td>\text{Game}_3</td>
<td>( \text{KG}(\text{mk}_1, f_1) \rightarrow \text{sk}_1 \cdot f_1 \cdot \text{Enc}(\text{mk}_2, (\bot, \text{sk}_1, f_1)) )</td>
<td>( \text{Enc}(\text{mk}_1, x_0) \rightarrow \text{ct}1, \text{Enc}(\text{mk}_1, x_1) \rightarrow \text{ct}1, \text{KG}(\text{mk}<em>2, F</em>{\text{ct}, \text{ct}1}) )</td>
</tr>
<tr>
<td>\text{Game}_4</td>
<td>( \text{KG}(\text{mk}_1, f_1) \rightarrow \text{sk}_1 \cdot f_1 \cdot \text{Enc}(\text{mk}_2, (\bot, \text{sk}_1, f_1)) )</td>
<td>( \text{Enc}(\text{mk}_1, x_0) \rightarrow \text{ct}1, \text{Enc}(\text{mk}_1, x_1) \rightarrow \text{ct}1, \text{KG}(\text{mk}<em>2, F</em>{\text{ct}, \text{ct}1}) )</td>
</tr>
</tbody>
</table>

Table 2: The description of oracle outputs in games \( \text{Game}_0, \ldots, \text{Game}_4 \). \( F_{\text{ct}, \text{ct}1} \) is defined in Figure 5.

### Difference between \text{Game}_0 and \text{Game}_1

Assuming the existence of an adversary \( A \) having a non-negligible advantage in distinguishing two game \( \text{Game}_0 \) and \( \text{Game}_1 \), we construct a simulator \( B \) breaking the message privacy IND\(_\text{mp} \) of FE1. More precisely, we assume that there exists \( A \) and a non-negligible function \( \varepsilon_0 \) in \( \lambda \) such that

\[
\left| \Pr \left[ A^{\text{KG}^{(0)}(\text{mk}, \cdot, \cdot), \text{Enc}^{(0)}(\text{mk}, \cdot, \cdot)}(\lambda) \rightarrow 1 \right] - \Pr \left[ A^{\text{KG}^{(1)}(\text{mk}, \cdot, \cdot), \text{Enc}^{(1)}(\text{mk}, \cdot, \cdot)}(\lambda) \rightarrow 1 \right] \right| > \varepsilon_0.
\]

17
Then, we will show that our $B$ construction also has the same advantage $\varepsilon_0$ against $\text{IND}_\text{MP}^{\text{non}}$ security of $\text{FE}_1$.

Now, we describe $B$'s oracle simulations. At the very beginning, $B$ runs Setup2$(\lambda^1) \rightarrow \text{mk}_2$ and Setup1$(\lambda^1) \rightarrow \text{mk}_1$, and keep the master secret keys $\text{mk}_2$ and $\text{mk}_1$ during the simulation.

$\text{KG}(\text{mk}, \ldots)$: On input $(f_0, f_1)$, $B$ returns the output of $\text{Enc}_2(\text{mk}_2, (\text{KG}1(\text{mk}_1, f_0), \perp))$. ($B$ already knows $\text{mk}_2$ and $\text{mk}_1$.)

$\text{Enc}(\text{mk}, \ldots)$: On input $(x_0, x_1)$, $B$ queries $(x_0, x_1)$ to its own oracle $\text{Enc}_b(\text{mk}_1, \ldots)$ $\rightarrow x_1$. $B$ computes $\text{Enc}_1(\text{mk}_1, x_0) \rightarrow \text{ct}_1$ and $\text{KG}_2(\text{mk}_2, F_{\text{ct}_1, \text{ct}_1}) \rightarrow \text{sk}_2_{\text{ct}_1, \text{ct}_1}$, and then delivers $\text{sk}_2_{\text{ct}_1, \text{ct}_1}$ to $A$.

Finally, if $A$ outputs $b'$, then $B$ sends it to the challenger. It is trivial that the challenge bit $b'$, which is used in the oracle $\text{Enc}_b(\text{mk}_1, \ldots)$, exactly determines whether $B$ perfectly simulates $\text{Game}_0$ or $\text{Game}_1$. Therefore, we conclude that $B$'s advantage is exactly equal to $\varepsilon_0$.

**Difference between $\text{Game}_1$ and $\text{Game}_2$.** Assuming the existence of an adversary $A$ having a non-negligible advantage in distinguishing two game $\text{Game}_1$ and $\text{Game}_2$, we construct a simulator $B$ breaking the message privacy $\text{IND}_\text{MP}^{\text{ad}}$ of $\text{FE}_2$. More precisely, we assume that there exists $A$ and a non-negligible function $\varepsilon_1$ in $\lambda$ such that

$$|\Pr[A^{\text{KG}_1(\ldots), \text{Enc}_1(\ldots)}(\lambda) \rightarrow 1] - \Pr[A^{\text{KG}_2(\ldots), \text{Enc}_2(\ldots)}(\lambda) \rightarrow 1]| > \varepsilon_1.$$ 

Then, we will show that our $B$ construction also has the same advantage $\varepsilon_1$ against the message privacy $\text{IND}_\text{MP}^{\text{ad}}$ of $\text{FE}_2$.

Now, we describe $B$'s oracle simulations. At the very beginning, $B$ runs Setup1$(\lambda) \rightarrow \text{mk}_1$ and Setup1$(\lambda) \rightarrow \text{mk}_1$, and keep the master secret keys $\text{mk}_1$ and $\text{mk}_1$ during the simulation.

$\text{KG}(\ldots)$: On input $(f_0, f_1)$, $B$ computes $\text{KG}_1(\text{mk}_1, f_0) \rightarrow \text{sk}_1_{f_0}$ and $\text{KG}_1(\text{mk}_1, f_1) \rightarrow \text{sk}_1_{f_1}$. $B$ sends $(\text{sk}_1_{f_0}, \perp)$ and $(\perp, \text{sk}_1_{f_1})$ to its own oracle $\text{Enc}_b(\text{mk}_2, \ldots)$. $B$ delivers the query result to $A$.

$\text{Enc}(\ldots)$: On input $(x_0, x_1)$, $B$ runs $\text{Enc}_1(\text{mk}_1, x_0) \rightarrow \text{ct}_1$ and $\text{Enc}_1(\text{mk}_1, x_1) \rightarrow \text{ct}_1$. Next, $B$ sends $F_{\text{ct}_1, \text{ct}_1}$ to $\text{KG}_2(\text{mk}_2, \ldots)$ oracle and obtains $\text{sk}_2_{\text{ct}_1, \text{ct}_1}$.

Then, $B$ returns the query result to $A$.

In $B$'s simulation for $\text{KG}$ oracle, $B$ used the output of the oracle $\text{Enc}_2(\text{mk}_2, \ldots)$. For a while, let us assume that all $B$'s queries satisfy the necessary conditions for oracle queries, and show it later. If $b = 0$, $B$'s simulation for $\text{KG}$ perfectly becomes $\text{KG}^{(1)}$. Otherwise ($b = 1$), $B$ simulates perfectly $\text{KG}^{(2)}$. Regardless of $b$, $B$ perfectly simulates $\text{Enc}^{(1)}(\text{mk}, \ldots) = \text{Enc}^{(2)}(\text{mk}, \ldots)$ oracle. Therefore, we conclude that $B$'s advantage is exactly equal to $A$'s advantage $\varepsilon_1$.
Now, we check whether all $B$’s queries satisfy the necessary conditions for oracle queries. First, by the assumption on the output of KG1, we know that $B$’s two inputs on $Enc_2(\text{mk}2, \cdot, \cdot)$ are of the equal size. Second, we show that for all queries $(f_0, f_1)$ and $(x_0, x_1)$ issued by $A$, $F_{c1\cdot c1}(sk_{1\cdot f_0}, \bot) = F_{c1\cdot c1}(\bot, sk_{1\cdot f_1})$, where $(c1\cdot c1)$ is a pair of encryptions $(Enc1(mk1, x_0), Enc1(mk1, x_1))$. By the definition of the function $F_{c1\cdot c1}$, we have $F_{c1\cdot c1}(sk_{1\cdot f_0}, \bot) = Dec1(ct1, sk_{1\cdot f_0}) = f_0(x_0) = f_1(x_1) = Dec1(c1\cdot sk_{1\cdot f_1}) = F_{c1\cdot c1}(\bot, sk_{1\cdot f_1})$. This completes the proof for the difference between $\text{Game}_1$ and $\text{Game}_2$.

**Difference between $\text{Game}_2$ and $\text{Game}_3$.** The proof for the difference between $\text{Game}_2$ and $\text{Game}_3$ is exactly the same as that for the difference between $\text{Game}_0$ and $\text{Game}_1$, except for changing the roles of $\text{mk}1$ and $\text{mk}1$. That is, $B$ generates $\text{mk}1$ and $\text{mk}2$ by itself, and plays a game for distinguishing a bit $b$ used in the challenge oracle $Enc1_b(\text{mk}1, \cdot, \cdot)$. Similarly to the proof between $\text{Game}_0$ and $\text{Game}_1$, from the message privacy IND$_{\text{mp}}$ of FE1, we can prove that the advantage of arbitrary adversary distinguishing two games $\text{Game}_2$ and $\text{Game}_3$ should be bounded by $B$’s advantage against breaking the message privacy IND$_{\text{mp}}$ of FE1.

**Difference between $\text{Game}_3$ and $\text{Game}_4$.** Assuming the existence of an adversary $A$ having a non-negligible advantage in distinguishing two game $\text{Game}_3$ and $\text{Game}_4$, we construct a simulator $B$ breaking the message privacy IND$_{\text{mp}}$ of FE2. More precisely, we assume that there exists $A$ and a non-negligible function $\varepsilon_3$ in $\lambda$ such that

$$\left| \Pr[A^{KG(3)(\cdot, \cdot), Enc(3)(\cdot, \cdot)}(\lambda) \rightarrow 1] - \Pr[A^{KG(4)(\cdot, \cdot), Enc(4)(\cdot, \cdot)}(\lambda) \rightarrow 1] \right| > \varepsilon_3.$$  

Then, we will show that our $B$ construction also has the same advantage $\varepsilon_3$ against the message privacy IND$_{\text{mp}}$ of FE2.

Now, we describe $B$’s oracle simulations. At the very beginning, $B$ runs $\text{Setup}1(1^{\lambda}) \rightarrow \text{mk}1$ and $\text{Setup}1(1^{\lambda}) \rightarrow \text{mk}1$, and keep the master secret keys $\text{mk}1$ and $\text{mk}1$ during the simulation.

**KG(\cdot, \cdot):** On input $(f_0, f_1)$, $B$ computes $KG1(\text{mk}1, f_1) \rightarrow sk_{1\cdot f_1}$ and $KG1(\text{mk}1, f_1) \rightarrow sk_{1\cdot f_1}$. $B$ sends $(sk_{1\cdot f_1}, \bot, (\bot, sk_{1\cdot f_1}))$ to its own oracle $Enc_2(\text{mk}2, \cdot, \cdot)$. $B$ delivers the query result to $A$.

**Enc(\cdot, \cdot):** On input $(x_0, x_1)$, $B$ runs $Enc1(\text{mk}1, x_1) \rightarrow ct1$ and $Enc1(\text{mk}1, x_1) \rightarrow ct1$. Next, $B$ sends $ct1\cdot c1$ to $KG2(\text{mk}2, \cdot)$ oracle and obtains $sk2_{ct1\cdot c1}$.

Then, $B$ returns the query result to $A$.

In $B$’s simulation for KG oracle, $B$ used the output of the oracle $Enc_2(\text{mk}2, \cdot, \cdot)$. For a while, let us assume that all $B$’s queries satisfy the necessary conditions for oracle queries, and show it later. If $b = 0$, $B$’s simulation for KG perfectly becomes KG(4). Otherwise ($b = 1$), $B$ simulates perfectly KG(3). Regardless of
5. Application to Inner-Product Functionality

In this section, we show that our transformations can be applied for inner-product functionality. Finally, we obtain a weakly function-private IPE scheme and a fully function-private IPE scheme, both under the SXDH assumption.

For a multi-exponentiation, we use the following notation: let \( G \) be a group of order \( q \), \( g = (g_1, \ldots, g_n) \in G^n \), \( h \in G \), and \( x = (x_1, \ldots, x_n) \in \mathbb{Z}_q^n \). Then, \( g^x \) and \( h^x \) denote \( \prod_{i=1}^n g_i^{x_i} \in G \) and \( (h^{x_1}, \ldots, h^{x_n}) \), respectively.

5.1. Weakly Function-Private IPE Construction

In this subsection, we propose two modifications of Agrawal, Libert and Stehlé’s IPE (ALS-IPE) scheme [28]. Then, our first transformation directly implies weakly function-private IPE scheme from these two FE schemes.

**First Modification: FE for Inner Product in the Exponent.** The first modification is quite simple: it is a private-key version and the decryption algorithm outputs \( e(g, \bar{g})^{\langle x, y \rangle} \) instead of \( \langle x, y \rangle \), where \( e \) is an asymmetric bilinear map and \( g \) and \( \bar{g} \) are generators. We provide our modification of the ALS-IPE scheme in Figure 6.

Indeed, the modified scheme is no longer IPE scheme. It outputs \( e(g, \bar{g})^{\langle x, y \rangle} \) instead of \( \langle x, y \rangle \), where \( x \) and \( y \) are vectors in input ciphertext and decryption key and \( g \) is a generator of \( G_1 \). However, it does not matter to our purpose, function-private IPE. The resulting scheme transformed by \( \text{Trans}1 \) will also output \( e(g, \bar{g})^{\langle x, y \rangle} \). Then, we can add a post process of solving a DLP based on \( e(g, \bar{g}) \) to the decryption algorithm of the transformed scheme.

**Theorem 4.** [28] The scheme given in Figure 6 is \( \text{IND}_{\text{ad}}^{\text{MP}} \)-secure under the DDH assumption in \( G_1 \).

**Second Modification: FE for Multi-exponentiation with Bilinear Map.** Then, we propose another FE for FE2 supporting multi-exponentiation with bilinear map; as described in Figure 6, the domain of \( F_{\text{ct1}} = \text{Dec1}(\text{ct1}, \cdot) \) is \( \mathbb{Z}_q^{n+2} \), and \( F_{\text{ct1}}(\text{sk}) = e(\text{ctsk}, \bar{g}) \) is a multi-exponentiation with bilinear map, where \( \text{ct1} \) is defined in a function and \( \text{sk} \) is a message.
Then, the output of $\text{Dec}$ can be rewritten by $(g, h, \{h_i, s_i, t_i\}_{i=1}^n)$ and $\text{pp} := (G, g)$. We again modify the key generation and decryption algorithm of the ALS-IPE scheme to construct an FE scheme for $G_1$. In the key generation algorithm, it takes as input $\bar{g} = (g_1, \ldots, g_{n+2}) \in G_1^{n+2}$ (instead of $\bar{g} = (y_1, \ldots, y_{n+2}) \in Z_q^{n+2}$) and outputs a decryption key for $\bar{g}$

$$
\bar{S} = \prod_{i=1}^{n+2} g_i^{-u_i} \prod_{i=1}^{n+2} g_i^{-v_i}, g_1, \ldots, g_{n+2}) \in G_1^{n+4},
$$

(3)

(instead of $-\sum_{i=1}^{n+2} u_i y_i, -\sum_{i=1}^{n+2} v_i y_i, y_1, \ldots, y_{n+2}) \in Z_q^{n+4}$), where $\bar{u} = (u_1, \ldots, u_{n+2})$ and $\bar{v} = (v_1, \ldots, v_{n+2})$ are master secret key. In the decryption procedure, it computes and outputs $\prod_{i=1}^{n+4} e(S_i, C_i)$ where $\bar{C} = (C_1, \ldots, C_{n+4})$ is a ciphertext of $\bar{u}$. The full description of functional encryption for a multi-exponentiation is given in Figure 7.

**Correctness.** We check the correctness of the scheme in Figure 7. An encryption for a vector $w$ is the form $ct_w = (g^r, \bar{h}^r, g_{\bar{z}_1}h_{\bar{z}_1}, \ldots, g_{\bar{z}_{n+z}}h_{\bar{z}_{n+z}})$ and a decryption key for a vector $g$ is the form $sk_g = (g^{-u}, g^{-v}, g_1, \ldots, g_{n+2})$. Let $g_i = g^z_i$ for some $z_i \in Z_q$; that is, $g = g^z$ for $z = (z_1, \ldots, z_{n+z})$. Then, $sk_g$ can be rewritten by $(g^{-\langle u, z \rangle}, g^{-\langle v, z \rangle}, g_1, \ldots, g_{n+z}) = g^{(-\langle u, z \rangle,-\langle v, z \rangle, z_1, \ldots, z_{n+z})}$. Then, the output of $\text{Dec}$ algorithm is

$$
\prod_{i=1}^{n+4} e(S_i, C_i) = e(g, ct^{(-\langle u, z \rangle,-\langle v, z \rangle, z_1, \ldots, z_{n+z})}) = e(g, \bar{g}^{\langle w, z \rangle}) = \prod_{i=1}^{n+2} e(g_i, \bar{g}).
$$

Figure 6: A Modified Agrawal-Libert-Stehlé IPE
Setup \((1^\lambda, (1^{n+2}, \{G, \bar{g}\})_\lambda)\): Choose \(\bar{h} \xleftarrow{\$} G_2\), \(u = (u_1, \ldots, u_{n+2})\), \(v = (v_1, \ldots, v_{n+2})\) \(\xleftarrow{\$} \mathbb{Z}_q^{n+2}\) and compute \(\bar{h}_i = \bar{g}^{u_i} \bar{h}^{v_i}\) for \(i = 1, \ldots, n+2\).

Output \(\text{mk} := (G, \bar{g}, \bar{h}, \{\bar{h}_i, u_i, v_i\}_{i=1}^{n+2})\) and \(\text{pp} := G\).

Enc \((\text{mk}, w)\): Given a vector \(w = (w_1, \ldots, w_{n+2}) \in \mathbb{Z}^{n+2}_q\), choose \(r \xleftarrow{\$} \mathbb{Z}_q\) and output \(ct_w := (\bar{g}^r, \bar{h}_1^r, \bar{g}^{w_1} \bar{h}_1^r, \ldots, \bar{g}^{w_{n+2}} \bar{h}_{n+2}^r) \in G^{n+4}_2\).

KG \((\text{mk}, g)\): Given a vector \(g = (g_1, \ldots, g_{n+2}) \in G^{n+2}_1\), compute and output \(sk_g := (g^{-u}, g^{-v}, g_1, \ldots, g_{n+2}) \in G^{n+4}_1\).

Dec \((\text{pp}, ct, sk)\): Given \(ct = (C_1, \ldots, C_{n+4}) \in G^{n+4}_2\) and \(sk = (S_1, \ldots, S_{n+4}) \in G^{n+4}_1\), compute and output \(\prod_{i=1}^{n+4} e(S_i, C_i)\).

The second equality holds since \(C_i\) and \((- \langle u, z \rangle, - \langle v, z \rangle, z_1, \ldots, z_{n+2})\) can be considered as an encryption of \(w\) and a decryption key for \(z\), respectively, of ALS-IPE over \((n+2)\)-length vectors, where the roles of \(G_1\) and \(G_2\) are changed.

Theorem 5. The scheme given in Figure 7 is IND\textsuperscript{MP}\textsubscript{ad} - secure under the DDH assumption in \(G_2\).

The proof is essentially the same as that of ALS-IPE scheme. We give the detailed proof in Appendix Appendix B.

Weakly Function-Private IPE. By the correctness of the transformation in Figure 2, the output of the decryption algorithm of IPE in Figure 6 is equal to the transformed scheme; that is, it outputs \(e(g, \bar{g})^\langle x, y \rangle\) instead of \(\langle x, y \rangle\). Therefore, to obtain an IPE scheme, we need a post process of solving the DLP with the base \(e(g, \bar{g})\).\(^{10}\) The final construction for our weakly function-private IPE is given in Figure 8.

The weak function privacy of the scheme in Figure 8 is directly obtained by Theorem 1, Theorem 4, and Theorem 5. We give the resulting theorem as follows.

---

\(^{10}\)To this end, we also require that the range of the inner product functionality should be polynomial in \(\lambda\). We note that all previous function-private functional encryption schemes for inner-product functionality [21, 22, 23] also have the same process of computing discrete logarithm in their decryption algorithms and so the condition for the range of inner product.

---

22
Setup($1^\lambda, 1^n$): Choose an asymmetric bilinear group $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e)$ of prime order $q > 2^\lambda$ with generators $g, h \overset{R}{\in} \mathbb{G}_1$ and $\tilde{g}, \tilde{h} \overset{R}{\in} \mathbb{G}_2$. Choose $s = (s_1, \ldots, s_n), t = (t_1, \ldots, t_n) \overset{R}{\in} \mathbb{Z}_q^n$ and $u = (u_1, \ldots, u_{n+2}), v = (v_1, \ldots, v_{n+2}) \overset{R}{\in} \mathbb{Z}_q^{n+2}$. Compute $h_i = g^{s_i}h_i$ for $i = 1, \ldots, n$ and $\tilde{h}_i = \tilde{g}^{s_i}\tilde{h}_i$ for $i = 1, \ldots, n + 2$. Output

$$mk := (\{h_i, s_i, t_i\}_{i=1}^n, \{\tilde{h}_i, u_i, v_i\}_{i=1}^{n+2})$$ and $pp := (\mathcal{G}, g, h, \tilde{g}, \tilde{h})$.

Enc(mk, x): Given a vector $x = (x_1, \ldots, x_n) \in \mathbb{Z}_q^n$, choose $r \overset{R}{\in} \mathbb{Z}_q$ and compute

$$ct_1_x := (g^r, h^r, g^{s_1}h_1^r, \ldots, g^{s_n}h_n^r) \in \mathbb{G}_1^{n+2}.$$ Output

$$ct_x := (ct_1_x^{-u}, ct_1_x^{-v}, ct_1_x^{-1}) \in \mathbb{G}_1^{n+4}.$$ KG(mk, y): Given a vector $y = (y_1, \ldots, y_n) \in \mathbb{Z}_q^n$, choose $r \overset{R}{\in} \mathbb{Z}_q$ and compute and output

$$sk_y := (\tilde{g}^r, \tilde{h}^r, \tilde{g}^{-(s_i,y)}\tilde{h}^r, \tilde{g}^{-(t_i,y)}\tilde{h}^r, \tilde{g}^{v_1}\tilde{h}_3, \ldots, \tilde{g}^{v_n}\tilde{h}_{n+2}) \in \mathbb{G}_2^{n+4}.$$ Dec(pp, ct, sk): Given $ct = (C_1, \ldots, C_{n+4}) \in \mathbb{G}_1^{n+4}$ and $sk = (S_1, \ldots, S_{n+4}) \in \mathbb{G}_2^{n+4}$, compute and output the discrete logarithm of $\prod_{i=1}^{n+4} e(C_i, S_i)$ to the base $e(g, \tilde{g})$.

Figure 8: Weakly Function-Private IPE

**Theorem 6.** If we apply the transformation given in Figure 2 to two functional encryptions in Figures 6 and 7, then the resulting functional encryption is weakly function-private under the SXDH assumption.

### 5.2. Full Function-Private IPE Construction

To apply our second transformation Trans2 for full function-privacy, we again use the scheme in Figure 6. However, contrary to the first transformation Trans1, it is not straightforward to apply Trans2 to this scheme. In contrast to Trans1, Trans2 in Figure 4 requires a functional encryption $FE_2$ for special functionality containing a branching statement, as in Figure 5. To the best of our knowledge, there is no efficient construction for message-private FE supporting such functionality under reasonable assumptions, except for indistinguishability obfuscation or multilinear maps. Therefore, it is not trivial to find an efficient construction for $FE_2$ in Trans2.

**Removing Branching Statement.** Although it is hard to find an efficient FE for such the functionality containing branching statement in general, we show that if we can assume that the range of $Dec_1$ has a group structure and $Dec_1(\cdot, \bot)$ is equal to the identity of the group, then we can remove such the branching statement.
 Setup$(1^λ, 1^n)$: Choose an asymmetric bilinear group $G = (G_1, G_2, G_t, e)$ of prime order $q > 2^λ$ with generators $g, h \in G_1$ and $g, h \in G_2$. Choose $s = (s_1, \ldots, s_n)$, $\hat{s} = (\hat{s}_1, \ldots, \hat{s}_n)$, $t = (t_1, \ldots, t_n)$, $\hat{t} = (\hat{t}_1, \ldots, \hat{t}_n) \neq \mathbb{Z}_q^n$ and $u = (u_1, \ldots, u_{n+2})$, $\mathbf{v} = (v_1, \ldots, v_{n+2}) \in \mathbb{Z}_q^{n+2}$. Compute $h_{i} = g^{v_{i}}h^{u_{i}}$, $\hat{h}_{i} = g^{v_{i}}h^{u_{i}}$ for $i = 1, \ldots, n$ and $\hat{h}_{1} = \hat{g}^{v_{1}}\hat{h}^{u_{1}}$ for $i = 1, \ldots, n + 2$. Output $\mathbf{mk} := \{(h_{i}, \hat{h}_{i}, s_{i}, \hat{s}_{i}, t_{i}, \hat{t}_{i})\}_{i=1}^{n+2}$ and $\mathbf{pp} := (\mathcal{G}, g, h, \hat{g}, \hat{h})$.

 Enc($\mathbf{mk}, \mathbf{x}$): Given a vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_q^n$, choose $r, \hat{r} \leftarrow \mathbb{Z}_q$ and compute
\[
\begin{align*}
\mathbf{ct}_1 & := (g^{x_1}h^{r}, g^{x_2}h^{r}, \ldots, g^{x_n}h^{r}) \in G_1^{n+2}, \\
\hat{\mathbf{ct}}_1 & := (g^{x_1}\hat{h}^{r}, g^{x_2}\hat{h}^{r}, \ldots, g^{x_n}\hat{h}^{r}) \in G_1^{n+2}.
\end{align*}
\]
Compute $\mathbf{C}_x := (\hat{\mathbf{ct}}_1^{-u}, \mathbf{ct}_1^{-v}, \mathbf{ct}_1), \hat{\mathbf{C}}_x := (\mathbf{ct}_1^{-u}, \hat{\mathbf{ct}}_1^{-v}, \hat{\mathbf{ct}}_1) \in G_1^{n+4}$. Output $\mathbf{ct}_x := (\mathbf{C}_x, \hat{\mathbf{C}}_x) \in G_1^{2(n+4)}$.

 KG($\mathbf{mk}, \mathbf{y}$): Given a vector $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{Z}_q^n$, choose $r, \hat{r} \leftarrow \mathbb{Z}_q$ and compute
\[
\begin{align*}
\mathbf{S}_y & := (\hat{y}^r\hat{h}^{r}, \hat{y}^r\hat{h}^{r}, \hat{y}^r\hat{h}^{r}, \ldots, g^{y_n}\hat{h}^{r}) \in G_2^{n+4}, \\
\mathbf{S}'_y & := (\hat{y}^r\hat{h}^{r}, \hat{y}^r\hat{h}^{r}, \hat{y}^r\hat{h}^{r}, \ldots, g^{y_n}\hat{h}^{r}) \in G_2^{n+4}.
\end{align*}
\]
Output $\mathbf{sk}_y := (\mathbf{S}_y, \mathbf{S}'_y) \in G_2^{2(n+4)}$.

 Dec($\mathbf{pp}, \mathbf{ct}, \mathbf{sk}$): Given $\mathbf{ct} = (C_1, \ldots, C_{n+4}, \hat{C}_1, \ldots, \hat{C}_{n+4}) \in G_1^{2(n+4)}$ and $\mathbf{sk} = (S_1, \ldots, S_{n+4}, S'_1, \ldots, S'_{n+4}) \in G_2^{2(n+4)}$, compute and output the discrete logarithm of $\prod_{i=1}^{n+4} c(C_i, S_i) \cdot c(\hat{C}_i, S'_i)$ to the base $e(g, \hat{g})$.

Figure 9: Fully Function-Private Functional Encryption for Inner-Product

We assume that the decryption algorithm Dec1 in Figure 4 satisfies the following conditions

1. Let $G_t$ be a (multiplicative) group. For any valid ciphertext $\mathbf{ct}_1$ and any key $k$, Dec1($\mathbf{ct}_1, k$) $\in G_t$.

2. $\text{Dec1}(\mathbf{ct}_1, \bot) = 1_t$, where $1_t$ is the identity of $G_t$.

Then, the above conditions immediately imply that for the function in Figure 5, if $(k, \hat{k}) \neq (\bot, \bot)$, then
\[
F_{\mathbf{ct}_1, \mathbf{ct}_1}(k, \hat{k}) = \text{Dec1}(\mathbf{ct}_1, k) \cdot \text{Dec1}(\mathbf{ct}_1, \hat{k}),
\] (4)

where $\cdot$ is a group operation in $G_t$. However, we note that although we have defined the case $(k, \hat{k}) \neq (\bot, \bot)$ for completeness, this case is not used in the transformation nor the proof of Theorem 3. Therefore, Equation (4) is sufficient for the definition of $F_{\mathbf{ct}_1, \mathbf{ct}_1}$ under the above two conditions.
We note that the above conditions do not affect the proof of Theorem 3 since these conditions are only about the output format of Dec1 algorithm and Dec1( ct1, ⊥) is not used in both the scheme and the proof at all. Therefore, if there is a message-private FE scheme satisfying the above two conditions and it is used as FE1 in Figure 4, then we can still apply Theorem 3, so that the resulting scheme becomes fully function-private.

**Construction for FE2.** We can slightly modify ALS-IPE to satisfy the above conditions; redefine Dec algorithm for a special input sk =⊥ to output the identity of Gt. An easy way to implement this modification is to define ⊥ as a zero vector in Zn+2.11 We now consider the function $F_{ct1, \widehat{ct1}}(k, \widehat{k})$ when we use ALS-IPE as FE1 in our full function-privacy transformation. We have

$$F_{ct1,\widehat{ct1}}(k, \widehat{k}) = \text{Dec1}(ct1, k) \cdot \text{Dec1}(\widehat{ct1}, \widehat{k})$$

$$= e( ct1^k, \bar{g}) \cdot e(\widehat{ct1}^k, \bar{g})$$

$$= e( ct1^k \cdot \widehat{ct1}^k, \bar{g})$$

$$= e( (ct1, \widehat{ct1})^{(k, \widehat{k})}, \bar{g}).$$

Note that either k or $\widehat{k}$ is ⊥, but we represent ⊥ by a zero vector in $\mathbb{Z}^{n+2}$. so that multi-exponentiation using ⊥ is well defined. The function $F_{ct1,\widehat{ct1}}(k, \widehat{k})$ eventually becomes a multi-exponentiation with bilinear map; mapping from $((ct1, \widehat{ct1}), (k, \widehat{k}))$ to $e((ct1, \widehat{ct1})^{(k, \widehat{k})}, \bar{g})$. Therefore, similar to weak function-privacy transformation, we can use the FE scheme for multi-exponentiation with bilinear map (Figure 7) as FE2 of full function-privacy transformation in Figure 4.

**Fully Function-Private IPE.** We give the full description of the transformed scheme in Figure 9. The full function privacy of the scheme in Figure 9 is directly obtained by Theorem 3, Theorem 4, and Theorem 5. We give the resulting theorem as follows.

**Theorem 7.** If we apply the transformation given in Figure 4 to two functional encryptions in Figure 6 and 7, then the the resulting functional encryption is fully function-private under the SXDH assumption.

**References**


11In fact, a zero vector in $\mathbb{Z}^{n+2}$ is reserved for the decryption key of a zero vector $\mathbb{Z}^{n+2}$. We note that this collision does not make any contradictions in both the scheme and the proof. Indeed, the proof of Theorem 3 does not care the output of Dec1( ct1, ⊥), in particular, if Dec1( ct1, ⊥) = i_t.


Appendix A. Definitions of Simulation-based Security

In the below definitions of SIM-based security we use the the notation $A^{B(O^{(1)\cdot\tau})}$. It means that $A$ can issue a query $q$ to $B$, which forwards a query $q$ to a stateful oracle $O$. $O(q;\tau)$ is then executed and outputs a pair $(q',\tau')$, where $q'$ is sent to $B$ and $\tau$ is updated to $\tau'$ for the next invocation. $B(q',\sigma)$ is then executed and outputs $(r_q,\sigma')$. Finally $B$ sends $r_q$ to $A$ as the response to the query $q$, and updates $\sigma$ to $\sigma'$ for the next invocation.

Definition 7 (SIM\textsubscript{non} and SIM\textsubscript{ad} - Security for Private-Key FE). Let $FE$ be a private-key functional encryption for function space $\mathcal{F}$ over a message space $\mathcal{M}$. For every probabilistic polynomial-time adversary $A$ and a probabilistic polynomial-time simulator $S = (S\cdot\text{setup}, S\cdot\text{kg}, S\cdot\text{enc})$, consider the following two experiments:
Real\(_A(1^\lambda)\):

1. \((pp, mk) \leftarrow \text{Setup}(1^\lambda)\)
2. \(\alpha \leftarrow A^{KG(mk,:)\cdot Enc(mk,:)\cdot (1^\lambda, pp)}\)
   Let \(\{f_{i'} : i' \in [i]\}\) be queries to KG made by \(A\)
   Let \(\{x_{j'} : j' \in [j]\}\) be queries to Enc made by \(A\)
3. Output \(\{\alpha, f_{i'}, x_{j'} : i' \in [i], j' \in [j]\}\)

Ideal\(_A,S(1^\lambda)\):

1. \((pp, \sigma) \leftarrow S.\text{setup}(1^\lambda)\)
2. \(\alpha \leftarrow A^{S.kg(O^{\text{SIM}} : \cdot ; \sigma) \cdot S.enc(O^{\text{SIM}} : \cdot ; \sigma) \cdot (1^\lambda, pp)}\)
   Let \(\{f_{i'} : i' \in [i]\}\) be queries to \(S.kg\) made by \(A\)
   Let \(\{x_{j'} : j' \in [j]\}\) be queries to \(S.enc\) made by \(A\)
3. Output \(\{\alpha, f_{i'}, x_{j'} : i' \in [i], j' \in [j]\}\)

1. In the adaptive query case, the stateful algorithms \(S.kg\) and \(S.enc\) simulate the ideal key generation and encryption, respectively. They share the state \(\sigma\), which is initialized by \(S_1\) at the beginning of the game. They work as follows:

   - Given the decryption key query of \(f_{i+1} \in \mathcal{F}\) made by \(A\) (in adaptive manner), \(S.kg\) forwards it to \(O^{\text{SIM}}\) to obtain its inputs \(|f_{i+1}|\) and \(\{(f_{i+1}, j', f_{i+1}(x_{j'})) : j' \in [j]\}\). \(x_{j'}\)'s are queries of messages until then. It outputs \((sk_{f_{i+1}}, \sigma')\). It updated \(\sigma\) by adding a pair \((f_{i+1}, sk_{f_{i+1}})\) into \(\sigma\).

   - Given the ciphertext query of \(x_{j+1} \in \mathcal{M}\) made by \(A\) (in adaptive manner), \(S.enc\) forwards it to \(O^{\text{SIM}}\) to obtain its inputs \(|x_{j+1}|\) and \(\{(f_{i'}, j+1, f_{i'}(x_{j+1})) : i' \in [i]\}\), where \(f_{i'}\)'s are queries of functions until then. It outputs \((ct_{x_{j+1}}, \sigma')\).

The private-key functional encryption \(\text{FE}\) is then said to be satisfied with the simulation-based message privacy against adaptive query attack (\(\text{SIM}^{\text{MP}}_{\text{ad}}\) - secure, for short) if two distributions above are computationally indistinguishable.

2. In the non-key query case, the oracle KG in the real game and \(S.kg\) in the ideal game are both empty oracles. Thus two games above simply mean \(\text{SIM}\) security of the encryption algorithm of \(\text{FE}\). That is, given the ciphertext query of \(x_{j+1} \in \mathcal{M}\) made by \(A\), \(S.enc\) obtains its input \(|x_{j+1}|\) from \(O^{\text{SIM}}\) and outputs \((ct_{x_{j+1}}, \sigma')\). The private-key functional encryption \(\text{FE}\) is then said to be satisfied with the simulation-based message privacy against non-key query attack (\(\text{SIM}^{\text{MP}}_{\text{non}}\) - secure, for short) if two distributions above are computationally indistinguishable.

**Definition 8 (\(\text{SIM}^{\text{FFP}}_{\text{ad}}\) - Security for Private-Key FE).** Let \(\text{FE}\) be a private-key functional encryption for function space \(\mathcal{F}\) over a message space \(\mathcal{M}\). For every probabilistic polynomial-time adversary \(A\) and a probabilistic polynomial-time simulator \(S = (S.\text{setup}, S.kg, S.enc)\), consider the following two experiments:
\[ \text{Real}_A(1^\lambda) : \]

1. \( (pp, mk) \leftarrow \text{Setup}(1^\lambda) \)
2. \( \alpha \leftarrow A^{\text{KG}(mk, \cdot), \text{Enc}(\cdot, \cdot)}(1^\lambda, pp) \)
   Let \( \{ f_{i'} : i' \in [i] \} \) be queries to KG made by \( A \)
   Let \( \{ x_{j'} : j' \in [j] \} \) be queries to Enc made by \( A \)
3. Output \( \{ \alpha, f_{i'}, x_{j'} : i' \in [i], j' \in [j] \} \)

\[ \text{Ideal}_{A,S}(1^\lambda) : \]

1. \( (pp, \sigma) \leftarrow S.\text{setup}(1^\lambda) \)
2. \( \alpha \leftarrow A^{S.\text{kg}(\cdot; \cdot; \cdot), S.\text{enc}(\cdot; \cdot; \cdot)}(1^\lambda, pp) \)
   Let \( \{ f_{i'} : i' \in [i] \} \) be queries to S.kg made by \( A \)
   Let \( \{ x_{j'} : j' \in [j] \} \) be queries to S.enc made by \( A \)
3. Output \( \{ \alpha, f_{i'}, x_{j'} : i' \in [i], j' \in [j] \} \)

The stateful algorithms \( S.\text{kg} \) and \( S.\text{enc} \) share the state \( \sigma \), which is initialized by \( S_1 \) at the beginning of the game, and simulate the ideal key generation and encryption, respectively. They work as follows:

- Given the decryption key query of \( f_{i+1} \in F \) made by \( A \) (in adaptive manner), \( S.\text{kg} \) forwards it to \( O^{\text{FFP}} \) to obtain its inputs \( |f_{i+1}| \) and \( \{(i + 1, j', f_{i+1}(x_{j'})) : j' \in [j]\} \), \( x_{j'} \)'s are queries of messages until then. It outputs \( (sk_{f_{i+1}}, \sigma') \).

- Given the ciphertext query of \( x_{j+1} \in M \) made by \( A \) (in adaptive manner), \( S.\text{enc} \) forwards it to \( O^{\text{FFP}} \) to obtain its inputs \( |x_{j+1}| \) and \( \{(i', j + 1, f_{i'}(x_{j+1})) : i' \in [i]\} \), where \( f_{i'} \)'s are queries of functions until then. It outputs \( (ct_{x_{j+1}}, \sigma') \).

The private-key functional encryption \( FE \) is then said to be satisfied with the simulation-based full function privacy against adaptive query attack (\( \text{SIM}_{\text{FFP}}^{\text{ad}} \)-secure, for short) if two distributions above are computationally indistinguishable.

Appendix B. Missing Proof

Appendix B.1. Proof for Theorem 2

Let \( S_1 = (S_1.\text{setup}, S_1.\text{kg}, S_1.\text{enc}) \) be the \( \text{SIM}_{\text{non}}^{\text{MP}} \)-simulator for \( \text{FE1} \) which accesses to the oracle \( O_{\text{MP}_1} \), and \( S_2 = (S_2.\text{setup}, S_2.\text{kg}, S_2.\text{enc}) \) be the \( \text{SIM}_{\text{ad}}^{\text{MP}} \)-simulator for \( \text{FE2} \) which accesses to the oracle \( O_{\text{MP}_2} \). Given \( S_1 \) and \( S_2 \), we construct the \( \text{SIM}_{\text{FFP}}^{\text{ad}} \)-simulator \( S = (S.\text{setup}, S.\text{kg}, S.\text{enc}) \) for \( \text{FE} \) under the transformation in Figure 2. In the below we denote a decryption key for \( f_{i'} \) and a ciphertext for \( x_{j'} \) of \( \text{FE1} \) by \( sk_{f_{i'}} \) and \( ct_{x_{j'}} \), respectively. \( S \) works as follows:

- \( S.\text{setup}(1^\lambda) \) simply executes \( (pp_1, \sigma_1) \leftarrow S_1.\text{setup}(1^\lambda) \) and \( (pp_2, \sigma_2) \leftarrow S_2.\text{setup}(1^\lambda) \). It outputs \( (pp, \sigma) \), where \( pp := (pp_1, pp_2) \) and \( \sigma := (\sigma_1, \sigma_2) \).
The description of $S_{\text{kg}}(O_{\text{FPP}}(\cdot ; \tau); \sigma)$ is depicted in Figure B.10. Given the decryption key query of $f_{i+1} \in F$ made by $A$, $S_{\text{kg}}$ works as follows:

1. $S_{\text{kg}}$ forwards it to $O_{\text{FPP}}$ to get $f_{i+1}$ and $\{(i+1, j', f_{i+1}(x_{j'})) : j' \in [j]\}$.
2. It invokes $S_{\text{enc}}$ on the behalf of $O_{\text{MP}}$ in order to obtain a ciphertext $ct_{2}$ of $FE_{2}$ on the message $x_{j+1}$. It generates inputs $|sk_{f_{i+1}}| = |f_{i+1}|$ and $\{(F_{ct_{1}j'}, i+1, f_{i+1}(x_{j'})) : j' \in [j]\}$ for $S_{\text{enc}}$. $S_{\text{enc}}$ then outputs $ct_{2}$, $\sigma_{2}^{\prime}$. Note that since $x_{j'}^{\prime}$ has been queried before, its $FE_{1}$ ciphertext $ct_{1}j'$ is already known to $S_{\text{kg}}$, which has been obtained by $S_{\text{enc}}$ in the below.
3. It provides $sk_{f_{i+1}} := ct_{2}$ to $A$ as the response of the query.

The description of $S_{\text{enc}}(O_{\text{FPP}}(\cdot ; \tau); \sigma)$ is depicted in Figure B.11. Given the encryption query of $x_{j+1} \in M$ made by $A$, $S_{\text{enc}}$ works as follows:

1. $S_{\text{enc}}$ forwards it to $O_{\text{FPP}}$ to get $x_{j+1}$ and $\{(i', j'+1, f_{i+1}(x_{j'})) : i' \in [i]\}$.
2. It invokes $S_{\text{enc}}$ on the behalf of $O_{\text{MP}}$ in order to obtain a ciphertext $ct_{1,j+1}$ of $FE_{1}$ on the message $x_{j+1}$. It feeds an input $x_{j+1}$ to $S_{\text{enc}}$, which outputs $ct_{1,j+1}$, $\sigma_{1}^{\prime}$. It then invokes $S_{\text{kg}}$ on the behalf of $O_{\text{MP}}$ in order to obtain a decryption key $sk_{2}$ of $FE_{2}$ on the function $F_{ct_{1,j+1}}$. It generates inputs
\(|F_{ct1}| \) and \( \{(F_{ct1,i+1}, i', f'_i(x_{j+1})) : i' \in [i]\} \) for \( S_{2,kg} \) by combining the given inputs of the previous steps. \( S_{2,kg} \) then outputs \((sk2, \sigma')\).

4. It provides \( ct_{x_{j+1}} := sk2 \) to \( A \) as the response of the query.

We now claim that \( S_{kg} \) and \( S_{enc} \) properly simulate ideal decryption key generation and encryption, respectively. For \( A \)'s key generation query \( f_{i+1} \), the real FE key generation produces \( sk_{f_{i+1}} := ct2_{sk1,f_{i+1}} \leftarrow Enc2(mk2, sk_{f_{i+1}}) \), where \( sk_{f_{i+1}} := KG1(mk1, f_{i+1}) \). \( S_{kg} \) invokes \( S_{2,enc} \) to obtain the simulated FE2 ciphertext \( sk'_{f_{i+1}} := ct2'_{sk1,f_{i+1}} \) for \( sk_{f_{i+1}} \), which is computationally indistinguishable from \( sk_{f_{i+1}} \). By the correctness of \( S_{2,enc} \) and \( ct_{x_{j'}} = sk2'_{x_{j'}} \), 
\[
\text{Dec}(ct_{x_{j'}}, sk_{f_{i+1}}) = \text{Dec}(ct_{sk1,f_{i+1}}, sk'_{x_{j'}}) = f_{i+1}(x_{j'}) = \text{Dec}(ct2'_{sk1,f_{i+1}}, sk'_{x_{j'}})
\]
\[
= \text{Dec}(ct_{x_{j'}}, sk_{f_{i+1}}'), \text{ for } j' \in [j].
\]

For \( A \)'s encryption query \( x_{j+1} \), the real FE encryption produces \( ct_{x_{j+1}} := sk2'_{x_{j+1}} \leftarrow KG2(mk2, F_{ct1}) \), where \( F_{ct1} \) is obtained from \( ct1 \leftarrow Enc1(mk1, x_{j+1}) \). \( S_{2} \) exploits \( S_{1,enc} \) to obtain the simulated FE1 ciphertext \( ct_{1}' \) for \( x_{j+1} \), which is computationally indistinguishable from \( ct1 \). \( S_{kg} \) then uses \( S_{2,enc} \) to obtain \( ct_{x_{j+1}} \) from \( F_{ct1}' \), which is computationally indistinguishable from \( ct_{x_{j+1}} \). By the correctness of \( S_{1,enc} \) and \( S_{2,kg} \), 
\[
\text{Dec}(ct_{x_{j+1}}, sk_{f_{i+1}}) = \text{Dec}(ct1, sk_{f_{i+1}}) = f_{i+1}(x_{j+1}) = \text{Dec}(ct1', sk_{f_{i+1}}) = \text{Dec}(ct'_{x_{j+1}}, sk_{f_{i+1}}) \text{ for } i' \in [i].
\]

\( \square \)

Appendix B.2. Proof Sketch for Theorem 5

The proof is essentially the same as that of ALS-FE scheme \[28\]. We use sequential games that begins with real game and ends with the game where the adversary has no advantage in distinguishing challenge ciphertext. We denote the event that the adversary wins in \( Game_i \) by \( S_i \).

**Game_0.** This is the real game. In the challenger phase, the adversary \( A \) chooses and sends challenge messages \( \bar{w}_0, \bar{w}_1 \in Z_q^n \) and obtains encryption of \( \bar{w}_b = (w_{b1}, \ldots, w_{bn+2}) \) from challenger \( B \). We observe that for \( \bar{g} = (g_1, \ldots, g_{n+2}) \in G_2^{n+2} \) that submitted to key generation oracle, the equation \( e(\bar{g}^{\bar{w}_0}, \bar{g}) = e(\bar{g}^{\bar{w}_1}, \bar{g}) \) must hold in \( G_T \).

**Game_1** This game is the same with \( Game_0 \) except that the challenger ciphertext \( ct_{\bar{w}_b} = (C, D, E_1, \ldots, E_n) \) is computed as follows:

\[
C = \bar{g}^r, \ D = \bar{h}^r \text{ and } E_i = \bar{g}^{w_{b,i}} \cdot C^{w_i} \cdot D^{v_i}, \quad (B.1)
\]

where \( r \leftarrow Z_q \). Then the challenge ciphertext \( ct_{\bar{w}_b} \) has the same distribution with \( Game_0 \). Therefore, we have \( \Pr[S_0] = \Pr[S_1] \).

**Game_2.** This game is the same with \( Game_1 \) except that the challenger ciphertext \( ct_{\bar{w}_b} = (C, D, E_1, \ldots, E_n) \) is computed as follows:

\[
C = \bar{g}^r, \ D = \bar{h}^{r + r'} \text{ and } E_i = \bar{g}^{w_{b,i}} \cdot C^{w_i} \cdot D^{v_i}, \quad (B.2)
\]

where \( r, r' \leftarrow Z_q \). We can easily verify that the distribution of \( (C, D, E_1, \ldots, E_n) \) in equations \( (B.1) \) and \( (B.2) \) are computationally indistinguishable under DDH.
As in [28], we can show that the challenger ciphertext \(ct_{\pi}\) in \(Game_2\) is perfectly hide the information on \(b \in \{0, 1\}\). Therefore, we have 
\[
|\Pr[S_0] - \frac{1}{2}| = |\Pr[S_0] - \Pr[S_2]| = |\Pr[S_1] - \Pr[S_2]| \leq \text{Adv}^{\text{DDH}}_{\mathcal{B}, \mathcal{C}_2}(\lambda).
\]
\(\square\)