1-1-2013

Fully secure hidden vector encryption under standard assumptions

Jong Hwan Park
Korea University - Korea

Kwangsu Lee
Korea University - Korea

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Dong Hoon Lee
Korea University - Korea

Fully secure hidden vector encryption under standard assumptions

Abstract

Hidden Vector Encryption (HVE) is a special type of predicate encryption that can support conjunctive equality and range searches on encrypted data. All previous HVE schemes were proven to be either selectively secure or weakly attribute-hiding. In this paper, we first construct a new HVE scheme that is fully secure under standard assumptions. Our HVE scheme, which is based on bilinear maps (pairings), provides efficiency advantages in that it requires O(1)-sized private keys and O(1) pairing computations for decryption, regardless of both the number of conjunctives and the dimension of vectors. To achieve our goal, we develop a novel technique to realize a tag-based dual system encryption in prime-order groups and show how to hide vector components and compress tag values into one.

Keywords

assumptions, hidden, fully, vector, secure, encryption, under, standard

Disciplines

Engineering | Science and Technology Studies

Publication Details

Park, J. Hwan., Lee, K., Susilo, W. & Lee, D. Hoon. (2013). Fully secure hidden vector encryption under standard assumptions. Information Sciences, 232 188-207.

Fully secure hidden vector encryption under standard assumptions

Jong Hwan Park [a], Kwangsu Lee [a], Willy Susilo [b], Dong Hoon Lee [a,*]

[a] Center for Information Security and Technologies, Korea University, Seoul, Republic of Korea
[b] School of Computer Science and Software Engineering, University of Wollongong, Northfields Avenue, Wollongong, Australia

Abstract

Hidden Vector Encryption (HVE) is a special type of predicate encryption that can support conjunctive equality and range searches on encrypted data. All previous HVE schemes were proven to be either selectively secure or weakly attribute-hiding. In this paper, we first construct a new HVE scheme that is fully secure under standard assumptions. Our HVE scheme, which is based on bilinear maps (pairings), provides efficiency advantages in that it requires O(1)-sized private keys and O(1) pairing computations for decryption, regardless of both the number of conjunctives and the dimension of vectors. To achieve our goal, we develop a novel technique to realize a tag-based dual system encryption in prime-order groups and show how to hide vector components and compress tag values into one.

1. Introduction

Recently, predicate encryption [28] has received considerable attention as a new vision in public key encryption. In a predicate encryption scheme, an encryptor uses a public key PK to generate a ciphertext $\mathsf{CT}_{x,M}$, which is an encryption of an arbitrary access control policy $x \in X$ as well as a message M, and an authority who has a master secret key generates a secret key sk_y for another access control policy $y \in Y$. Using sk_y, the ciphertext $\mathsf{CT}_{x,M}$ is successfully decrypted, i.e., the decryption outputs the right message M if and only if $P(x, y) = 1$, where P is a predicate function defined as $P{:}X \times Y \to \{0, 1\}$. A primary security property of predicate encryption is that the ciphertext $\mathsf{CT}_{x,M}$ leaks no information about either x or M.[1] Nevertheless, the possibility of computing the predicate $P(x, y)$ without revealing x from the ciphertext can provide a good solution for searching encrypted data.

One application of predicate encryption could be an electronic health record system where patients' sensitive data should be securely encrypted. When patients' data needs to be accessed by an outside entity, access should be limited to only the minimum necessary amount of data. In the health record system, each doctor has its own public/private key pair, and encrypts a patient's data M each time the doctor treats a patient. The data M is encrypted along with an access policy $x$that could be comprised of a set of attributes like the name of the patient, name of a disease, date of treatment, etc. If an outside entity later requests an access token associated with a particular access policy y, the doctor generates a private key sk_y and gives it to the entity as a token. Within the security of predicate encryption, the outside entity is able to access the set of

* Corresponding author.
E-mail addresses: decartian@korea.ac.kr (J.H. Park), guspin@korea.ac.kr (K. Lee), wsusilo@uow.edu.au (W. Susilo), donghlee@korea.ac.kr (D.H. Lee).

[1] Functional encryption [11] is a broader concept including predicate encryption, which encompasses the case in which $\mathsf{CT}_{x,M}$ does not reveal only information about M but not about x. The sub-classes of functional encryption with the revelation of x include Identity-Based Encryption (IBE) [43,9,14,6,47,22,16,2], Hierarchical IBE [26,23,6,7,21,46,32,33], Attribute-Based Encryption (ABE) [41,25,37,3,35,4,31], and Ciphertext-Policy ABE [5,24,30,35,45]. We refer to [11] for more precise definition and classification about functional encryption (including predicate encryption).

ciphertexts $\{\mathsf{CT}_{x,M}\}$ such that $P(x, y) = 1$, and not beyond it. This can be an exact realization of the minimum necessary requirement.

Predicate encryption can be realized in a variety of ways, depending on how the predicate function is explored over $X \times Y$. Until now, there have been three sub-classes of predicate encryption: anonymous Identity-Based Encryption (IBE), Hidden Vector Encryption (HVE), and Inner-Product Encryption (IPE). Anonymous IBE [1,20,12] supports a simple equality predicate and thus gives a simple equality search on encrypted data. HVE [13] provides a conjunctive equality predicate, which can be extended to give a conjunctive combination of equality, comparison, and range searches. IPE [28] employs an inner-product predicate, and this enables more complex access controls such as conjunctions, disjunctions, and polynomial evaluations. The relation between these three primitives forms a hierarchy: anonymous IBE $\leftarrow$ HVE $\leftarrow$ IPE, where $A \leftarrow B$ signfies that B implies A.

1.1. Efficiency of HVE

The predicate in an HVE scheme is defined over ℓ-dimensional vectors $\vec{x} \in X$ and $\vec{y} \in Y$. Most previous HVE schemes [13,44,27,28,36,30,18,35,38] (including HVE derived from IPE), which are all pairing-based, require not only $O(\ell)$ pairing computations to perform one decryption, but also a size $O(\ell)$ of private keys. From the perspective of efficiency, it is desirable that the searching cost per one ciphertext is not proportional to the number ℓ, i.e., the cost required to decrypt one ciphertext $\mathsf{CT}_{\vec{x},M}$ using $\mathsf{sk}_{\vec{y}}$[2] becomes $O(1)$. To search for suitable ciphertexts holding $P(\vec{x}, \vec{y}) = 1$, the decryptor should perform decryption on all ciphertexts $\{\mathsf{CT}_{\vec{x},M}\}$ in a storage server. This is because the decryptor does not know any information on the stored or incoming ciphertexts in advance and each ciphertext could possibly become the one that matches $\mathsf{sk}_{\vec{y}}$. The $O(\ell)$ pairing computations will become burdensome for the decryptor if the number ℓ increases to deal with more expressive access control, and become seriously problematic if a large number of users can have access to the storage system.

The size of $\mathsf{sk}_{\vec{y}}$ becomes an important factor since each $\mathsf{sk}_{\vec{y}}$ should be transmitted in a secure channel from the authority to the decryptor. In a storage system with a large number of users, the transmission can be viewed as a reverse situation of broadcast encryption [19] where a central authority broadcasts encrypted messages to many receivers. Shortening the size of broadcast ciphertexts has long been a central issue in designing broadcast encryption schemes [10,39]. Thus, like in broadcast encryption, it is necessary to shorten the transmission size of $\mathsf{sk}_{\vec{y}}$ as the number of users increases. Also, this is especially the case when the authority is based on a device with restricted resources like a smart phone. Until now, only a few HVE schemes [40,29] have achieved both $O(1)$ pairing computations and $O(1)$ size of private keys in a weaker security model (described below).

1.2. Security of HVE

It is better for an HVE scheme to be fully (or adaptively) secure. Full security means that an adversary is allowed to make both matching and non-matching private key queries for two target pairs $(\vec{x}_b, M_b)$ for $b = 0, 1$. In other words, any private key query for $\vec{y}$ is permitted as long as $P(\vec{x}_0, \vec{y}) = P(\vec{x}_1, \vec{y})$. In fact, this is the complete security notion of HVE that was suggested in [11], but no previous HVE (or even IPE) schemes have achieved full security. Most earlier HVE schemes [13,44,27,28,36,40,29,38] have argued their security in a selective security model (originated from [15]), albeit permitting the two type of key queries. Recently, several constructions [30,35,18] have overcome the barrier of selective security by adapting the technique of dual system encryption [46], but are unfortunately not yet fully secure since their security models allow an adversary to make only non-matching private key queries. This incomplete security is described as 'weakly attribute-hiding'.

There is a strict difference between weakly attribute-hiding security and full security. In the former case, the adversary is allowed to make only non-matching queries so that it cannot employ queried keys to decrypt a challenge ciphertext that is an encryption of $(\vec{x}_b, M_b)$ for a randomly chosen $b \in \{0, 1\}$. This ensures that the adversary does not know any information about (the whole of) $\vec{x}_b$ and M_b, provided that any matching key is not given. In contrast, full security considers an adversary that is able to ask both matching and non-matching queries. Naturally, full security encompasses weakly attributing security by additionally considering the case where an adversary is able to have matching keys. The additional security guarantees that even if the adversary knows information about the message[3] and (partial) $\vec{x}_b$ that involves the same vector components $x_{0,i} = x_{1,i}$ in $\vec{x}_b = (x_{b,0}, \ldots, x_{b,\ell})$ $(b = 0, 1)$, the adversary does not gain any information about the pairwise-distinct vector components in $\vec{x}_b$ from the ciphertext.

Although we have powerful tools like dual system encryption [46] for achieving adaptive security, the resulting HVE schemes [30,35,18] have been limited to weak attribute hiding. A natural direction of research would be to provide an answer to the open problem by presenting an HVE scheme that can be proven to be fully secure. Another challenge is that it is clearly desirable for HVE security to rely on well-known standard assumptions. Of all suggested HVE schemes (which are all pairing-based), only a few constructions [27,38,35] have demonstrated security under the Decision Bilinear Diffie-Hellman (DBDH) and Decision Linear (DLIN) assumptions. These constructions are all based on prime-order groups and can be instantiated using either symmetric or asymmetric bilinear maps.

[2] It is often called 'token', denoted as $\mathsf{TK}_{\vec{y}}$.

[3] In this case, two challenge messages should be equal, i.e., $M_0 = M_1$.

1.3. Our contribution

We present the first HVE scheme that is fully secure under the DBDH and DLIN assumptions, and additionally achieves $O(1)$ pairing computations and $O(1)$-sized private keys. Table 1 in Section 5 will compare our scheme with previous HVE schemes in terms of efficiency and security. We have developed a new method to realize dual system encryption in prime-order groups. Our method is similar to the original Waters' method [46] in the sense that tag values are critically used to solve a paradox in our dual system technology. Based on the new dual system encryption, we suggest our HVE scheme by introducing two techniques to hide each component of $\vec{x}$ from the ciphertext and also to compress tag values that would otherwise be associated with each component. Fortunately, combining these new techniques leads to improvements in efficiency and thus we can avoid the dependance on the dimension ℓ in terms of both private key size and pairing computations.

When encrypting $\vec{x} = (x_1, \ldots, x_\ell)$, a ciphertext component corresponding to x_i is encoded in the form of $(u_i h_i^{x_i} v^{\mathrm{tag}_i})^{s_1} Y_i^{s_2}$, where u_i, h_i, v and $Y_i = g^{y_i}$ are public parameters, and s_1, s_2, and tag_i are exponents randomly chosen by an encryptor. The element $Y_i^{s_2}$ is a blinding factor that plays a key role in preventing the component x_i from being revealed in groups with bilinear maps. In decryption, ciphertext components that need to match $\mathsf{sk}_{\vec{y}}$ for the vector $\vec{y} = (y_1, \ldots, y_\ell)$ are multiplied together, resulting in $\left(\prod_i u_i h_i^{x_i} \cdot v^{\Sigma_i \mathrm{tag}_i}\right)^{s_1} g^{(\Sigma_i y_i) s_2}$. The important point here is that the tag values $\{\mathrm{tag}_i\}$ are compressed into one. Thus, if the secret key $\mathsf{sk}_{\vec{y}}$ is constructed into the similar compressed form of $\left(\prod_i u_i h_i^{y_i} \cdot v^{\mathrm{tag}_k}\right)^r$ for a randomly chosen exponent r, we can make the size of $\mathsf{sk}_{\vec{y}}$ constant, irrelevant to the number of vector components embedded into $\mathsf{sk}_{\vec{y}}$. Moreover since $\mathsf{sk}_{\vec{y}}$ consists of a constant number of group elements, the number of pairing computations necessary for decryption also becomes constant.

To achieve full security, our proof is divided into two cases: (1) *all* private key queries are non-matching and (2) *at least one* private key query is matching. In the first case, we can apply the hybrid argument of dual system encryption [46] to prove the confidentiality of M_b, and on top of that we need to consider an additional hybrid argument to prove the confidentiality of $\vec{x}_b$. In the second case, at least one queried key can be used for successful decryption so that the message-hiding property is no longer necessary. At first glance, the adversary in the second case can ask all private key queries that are matching ones, which might make the second case proof seem challenging. However, for an index $i \in \{1, \ldots, \ell\}$ such that $x_{0,i} \neq x_{1,i}$ in both challenge vectors $\vec{x}_b = (x_{b,0}, \ldots, x_{b,\ell})(b = 0, 1)$, any key query for $\vec{y}$ that includes an ith component can not be matching, i.e., should be non-matching as in the first case. Using this fact, we can create a variant of the hybrid argument applied in the first case proof.

Since any HVE implies an anonymous IBE, our HVE construction can yield a new anonymous IBE scheme that is fully secure under the standard assumptions. Full security is straightforwardly achieved by using the same strategy as in the first case above, since all private key queries for identities should all be non-matching. Prior to our new result, several schemes [20,30,35] have been presented to offer full security without random oracles, and only [35] is fully secure under the DLIN assumption. Compared to [35], our anonymous IBE scheme is more efficient in all respects.

2. Preliminaries

2.1. Hidden vector encryption

Let Σ be an arbitrary set of attributes, and let $*$ be a wildcard character which is not involved with any attribute. We let $\mathcal{I} = \Sigma \cup \{*\}$. We then use two ℓ-dimensional vectors, $\vec{x} = (x_1, \ldots, x_\ell) \in \Sigma^\ell$ in the encryption phase and $\vec{\sigma} = (\sigma_1, \ldots, \sigma_\ell) \in \mathcal{I}^\ell$ in

Table 1
Comparison between other HVE schemes and ours.

Scheme	Group order	PK size	Ciphertext size	Token size	Decryption cost	Selective or full	Standard assumptions
BW-HVE [13]	p_1p_2	$O(\ell)$	$(2\ell+1)\mathsf{G} + 1\ \mathsf{G}_\mathsf{T}$	$(2\ell+1)\mathsf{G}$	$(2\ell+1)\mathsf{p}$	S	No
KSW-HVE$_{\mathrm{IPE}}$ [28]	$p_1p_2p_3$	$O(\ell)$	$2(2\ell+1)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$(2\ell+1)\mathsf{G}$	$2(2\ell+1)\mathsf{p}$	S	No
SW-HVE [44]$^\diamondsuit$	$p_1p_2p_3$	$O(\ell)$	$(\ell+3)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$(\ell+3)\mathsf{G}$	$(\ell+3)\mathsf{p}$	S	No
IP-HVE [27]$^\clubsuit$	p_1	$O(\ell)$	$(2\ell+1)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$(2\ell)\mathsf{G}$	$(2\ell)\mathsf{p}$	S	DBDH, DLIN
OT-HVE$_{\mathrm{IPE}}$ [36]$^\diamondsuit$	p_1	$O(\ell)$	$(2\ell+3)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$(2\ell+3)\mathsf{G}$	$(2\ell+3)\mathsf{p}$	S	No
Park-HVE$_{\mathrm{IPE}}$ [38]	p_1	$O(\ell)$	$2(4\ell+2)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$2(4\ell+2)\mathsf{G}$	$2(4\ell+2)\mathsf{p}$	S	DBDH, DLIN
PL-HVE [40]	p_1p_2	$O(\ell)$	$(2\ell+2)\mathsf{G} + 1\ \mathsf{G}_\mathsf{T}$	$4\mathsf{G}$	$4\mathsf{p} + 2(\ell-1)\mathsf{m}$	S	No
LL-HVE-1 [29]	$p_1p_2p_3$	$O(\ell)$	$(\ell+3)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$4\mathsf{G}$	$4\mathsf{p} + (\ell-1)\mathsf{m}$	S	No
LL-HVE-2 [29]$^\heartsuit$	p_1	$O(\ell)$	$(\ell+3)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$4\mathsf{G}$	$4\mathsf{p} + (\ell-1)\mathsf{m}$	S	No
LOS$^+$-HVE$_{\mathrm{IPE}}$ [30]$^\diamondsuit$	p_1	$O(\ell)$	$(4\ell+3)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$(4\ell+3)\mathsf{G}$	$(4\ell+3)\mathsf{p}$	wF$^\spadesuit$	No
DIP-HVE [18]$^{\clubsuit\ \diamondsuit}$	$p_1p_2p_3p_4$	$O(\ell)$	$(\ell)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$(\ell)\mathsf{G}$	$(\ell)\mathsf{p}$	wF$^\spadesuit$	No
OT-HVE$_{\mathrm{IPE}}$ [35]$^\diamondsuit$	p_1	$O(\ell)$	$(6\ell+6)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$(6\ell+6)\mathsf{G}$	$(6\ell+6)\mathsf{p}$	wF$^\spadesuit$	DLIN
OT-HVE$_{\mathrm{IPE}}$-1 [34]$^\diamondsuit$	p_1	$O(\ell^2)$	$2(4\ell+2)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$2(4\ell+2)\mathsf{G}$	$2(4\ell+2)\mathsf{p}$	F	DLIN
OT-HVE$_{\mathrm{IPE}}$-2 [34]$^\diamondsuit$	p_1	$O(\ell)$	$2(5\ell+1)\mathsf{G} + 1\mathsf{G}_\mathsf{T}$	$11\mathsf{G} + \ell\mathsf{Z}_p$	$11\mathsf{p} + 5(\ell-1)\mathsf{e}$	F	DLIN
Our HVE	p_1	$O(\ell)$	$(2\ell+6)\mathsf{G} + 1\mathsf{G}_\mathsf{T} + 1\mathsf{Z}_\mathsf{P}$	$9\mathsf{G} + 1\mathsf{Z}_\mathsf{P}$	$9\mathsf{p} + 3(\ell-1)\mathsf{m}$	F	DBDH, DLIN

p_i: prime numbers; ℓ: the dimension of vectors; $\{\mathsf{G},\mathsf{G}_\mathsf{T},\mathsf{Z}_p\}$: length of an element in $\{\mathbb{G}, \mathbb{G}_T, \mathbb{Z}_p\}$; $\{\mathsf{p},\mathsf{m},\mathsf{e}\}$: pairing, multiplication, and exponentiation in $\mathbb{G}$, respectively.

$^\clubsuit$ In [27,18], the components of vectors are defined over {0,1} in encryption and {0,1,*} in token generation.

$^\spadesuit$ weakly attribute-hiding.

$^\diamondsuit$ [44,36,30,18,35,34] provide delegation mechanism.

$^\heartsuit$ The second construction of [29] is based upon asymmetric bilinear maps.

the token generation phase. For the vector $\vec{\sigma} \in \mathcal{I}^\ell$, let $S(\vec{\sigma})$ be the set of indexes i such that σ_i is not a wildcard character. We define a predicate function $P_\ell : \mathcal{I}^\ell \times \Sigma^\ell \to \{0,1\}$ as follows:

$$P_\ell(\vec{\sigma} \in \mathcal{I}^\ell,\ \vec{x} \in \Sigma^\ell) = \begin{cases} 1 & \text{if for all } i \in S(\vec{\sigma}), x_i = \sigma_i, \\ 0 & \text{otherwise.} \end{cases}$$

In HVE, the sender encrypts a pair $(\vec{x}, M) \in \Sigma^\ell \times \mathcal{M}$ where $\mathcal{M}$ is a message space, and the receiver releases a token for a vector $\vec{\sigma}$. Then, the token can decrypt a ciphertext if and only if $P_\ell(\vec{\sigma}, \vec{x}) = 1$. With the predicate function described above, we formally define HVE by the following four algorithms:

Setup (k, ℓ) takes as input a security parameter k and a dimension ℓ of vector consisting of attributes. It outputs a public key PK and a secret key SK.

Encrypt $(\mathsf{PK}, (\vec{x}, M))$ takes as input the public key PK, a vector $\vec{x} \in \Sigma^\ell$ of attributes, and a message $M \in \mathcal{M}$. It outputs a ciphertext CT.

GenToken $(\mathsf{SK}, \vec{\sigma})$ takes as input the secret key SK and a vector $\vec{\sigma} \in \mathcal{I}^\ell$ of attributes. It outputs a token $\mathsf{TK}_{\vec{\sigma}}$.

Decrypt $(\mathsf{TK}_{\vec{\sigma}}, \mathsf{CT})$ takes as input the token $\mathsf{TK}_{\vec{\sigma}}$ and a ciphertext CT. It outputs a message M if $P_\ell(\vec{\sigma}, \vec{x}) = 1$ and outputs $\perp$ otherwise.

Correctness. For all $\vec{x} \in \Sigma^\ell$, all $\vec{\sigma} \in \mathcal{I}^\ell$, and all $M \in \mathcal{M}$, let $(\mathsf{PK}, \mathsf{SK}) \xleftarrow{R}$ **Setup**(k, ℓ), $\mathsf{CT} \xleftarrow{R}$ **Encrypt**$(\mathsf{PK}, (\vec{x}, M))$, and $\mathsf{TK}_{\vec{\sigma}} \xleftarrow{R}$ **GenToken**$(\mathsf{SK}, \vec{\sigma})$. If we have $P_\ell(\vec{\sigma}, \vec{x}) = 1$, $M \leftarrow$ **Decrypt**$(\mathsf{TK}_{\vec{\sigma}}, \mathsf{CT})$, otherwise $\Pr[\perp \leftarrow$ **Decrypt**$(\mathsf{TK}_{\vec{\sigma}}, \mathsf{CT})] > 1 - \epsilon(k)$ where $\epsilon(k)$ is a negligible function.

In the above definition, the message M is a real message that the encryptor wishes to send to recipients. In practice, M can also be used as a symmetric key with which authenticated encryption works to check the validity of the ciphertext.

2.2. Security for hidden vector encryption

Following [13,28,11,35], we describe the security for HVE that captures the intuition that the ciphertext CT reveals no information about $(\vec{x}, M)$. The security is defined in the following interaction between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, where ℓ is given to $\mathcal{A}$.

Setup: $\mathcal{C}$ runs the setup algorithm to obtain the public key PK and the secret key SK. It gives PK to $\mathcal{A}$.

Query Phase 1: $\mathcal{A}$ adaptively issues a polynomial number of token queries for vectors, $\vec{\sigma}_i$. $\mathcal{C}$ responds with the corresponding tokens $\mathsf{TK}_{\vec{\sigma}_i} \leftarrow$ **GenToken**$(\mathsf{SK}, \vec{\sigma}_i)$.

Challenge: $\mathcal{A}$ outputs $\vec{x}_0^*$, $\vec{x}_1^*$ and two messages M_0, M_1 under the two constraints that:

- $P_\ell(\vec{\sigma}_i, \vec{x}_0^*) = P_\ell(\vec{\sigma}_i, \vec{x}_1^*) = 0$ for all queried vectors, σ_i.
- $P_\ell(\vec{\sigma}_i, \vec{x}_0^*) = P_\ell(\vec{\sigma}_i, \vec{x}_1^*) = 1$ for at least one queried vector, σ_i, in which case $M_0 = M_1$.

$\mathcal{C}$ flips a coin $b \in \{0,1\}$ and gives $\mathsf{CT}^* \leftarrow$ **Encrypt**$(\mathsf{PK}, (\vec{x}_b^*, M_b))$ to $\mathcal{A}$.

Query Phase 2: $\mathcal{A}$ adaptively issues additional token queries for vectors, σ_i, subject to the restriction in **Challenge** above. $\mathcal{C}$ responds with the corresponding tokens $\mathsf{TK}_{\vec{\sigma}_i} \leftarrow$ **GenToken**$(\mathsf{SK}, \vec{\sigma}_i)$.

Guess: $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. $\mathcal{A}$ wins if $b' = b$.

The advantage of the adversary $\mathcal{A}$ in breaking the HVE scheme is defined as $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HVE}} = |\Pr[b' = b] - 1/2|$.

Definition 1. We say that a Hidden Vector Encryption (HVE) scheme is (attribute-hiding) secure if for any polynomial time adversaries $\mathcal{A}$ attacking the HVE scheme, the advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HVE}}$ is negligible.

2.3. Bilinear maps and complexity assumptions

Bilinear Maps: We adopt the notation in [9,6]. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two (multiplicative) cyclic groups of prime order p. We assume that g is a generator of $\mathbb{G}$. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a function that has the following properties:

1. Bilinear: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degenerate: $e(g, g) \neq 1$.
3. Computable: there is an efficient algorithm to compute the map e.

Then, we say that the map e is a bilinear map in $\mathbb{G}$. Note that $e(,)$ is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

The Decisional Bilinear Diffie-Hellman (**DBDH**) **Problem**: The DBDH problem [9] is defined as follows: given $(g, g^a, g^b, g^c, Z) \in \mathbb{G}^4 \times \mathbb{G}_T$ as input, determine whether $Z = e(g, g)^{abc}$ or Z is random in $\mathbb{G}_T$.

The Decision Linear (**DLIN**) **Problem:** The DLIN problem [8] was originally stated as follows: given $(g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, Z) \in \mathbb{G}^6$ as input, determine whether $Z = g^{z_3 + z_4}$ or Z is random in $\mathbb{G}$. We consider an equivalently modified version such as: given $(g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_4}, Z) \in \mathbb{G}^6$ as input, determine whether $Z = g^{z_2(z_3 + z_4)}$ or Z is random in $\mathbb{G}$. This was already used in [12].

Definition 2. We say that the {DBDH, DLIN} assumption holds in $\mathbb{G}$ if the advantage of any polynomial time algorithm in solving the {DBDH, DLIN} problem is negligible.

Remark 1. In the groups equipped with symmetric bilinear maps $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, we can see that the DBDH assumption is weaker than the DLIN assumption. To show this, let us assume that there is an adversary to solve DBDH problem. If an instance $(g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_4}, Z)$ as a DLIN problem is given, we first compute $Z' = e(g^{z_1}, Z)/e(g^{z_2}, g^{z_1 z_3})$ and next we give an instance $(g, g^{z_1}, g^{z_2}, g^{z_4}, Z')$ of a DBDH problem to the adversary. Clearly, if $Z' = e(g,g)^{z_1 z_2 z_4}$, then $Z = g^{z_2(z_3+z_4)}$, and otherwise, Z is random. It seems that the opposite direction does not hold, and also the relation between the n-DLIN assumption ($n > 2$) (which is also weaker than the DLIN assumption) and the DBDH is not clear.

3. Fully secure HVE scheme

3.1. Construction

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime order p, and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be the bilinear map. We assume that each attribute x_i belongs to $\Sigma = \mathbb{Z}_p$ and our scheme deals with ℓ-dimensional vector $\vec{x} = (x_1, \ldots, x_\ell) \in \Sigma^\ell$. If necessary, we can extend our construction to handle arbitrary attributes in $\{0,1\}^*$ by first hashing each x_i using a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p$. Note that $\mathcal{I} = \mathbb{Z}_p \cup \{*\}$.

Setup (λ): Given a security parameter $\lambda \in \mathbb{Z}^+$, the setup algorithm runs $\mathcal{G}(\lambda)$ to obtain a tuple $(p, \mathbb{G}, \mathbb{G}_T, e)$. The algorithm picks a random generator $g \in \mathbb{G}$, random elements $g_1, v, \varphi, \{u_i, h_i, \tau_i\}_{i=1}^{\ell}$ in $\mathbb{G}$, random exponents $\Omega, \gamma, \{y_i\}_{i=1}^{\ell}$, $\{w_i, \delta_i, \phi_i\}_{i=1}^{2}, \{f_i\}_{i=1}^{3}$ in $\mathbb{Z}_p$. It obtains $\alpha(\neq 0) \in \mathbb{Z}_p$ such that $w_1\delta_1 + w_2\delta_2 = \alpha\Omega$. If $\alpha = 0$, the algorithm tries again with new random exponents. It sets $w_1\phi_1 + w_2\phi_2 = \beta \in \mathbb{Z}_p$. The algorithm sets

$$W_1 = g^{w_1},\ W_2 = g^{w_2},\ F_1 = g^{f_1},\ F_2 = g^{f_2},\ F_3 = g^{f_3},$$
$$Y_i = g^{y_i} (i = 1, \ldots, \ell),\ g_2 = g^{\alpha},\ g_3 = g^{\beta},\ g_4 = g^{\gamma},\ \Lambda = e(g_1, g_2).$$

The public key PK (along with the description of $(p, \mathbb{G}, \mathbb{G}_T, e)$) and the secret key msk are set to be

$$\mathsf{PK} = \left(g, v, \varphi, \{u_i, h_i, \tau_i, Y_i\}_{i=1}^{\ell}, g_2, g_3, g_4, \{W_i\}_{i=1}^{2}, \{F_i\}_{i=1}^{3}, \Lambda\right) \in \mathbb{G}^{4\ell+11} \times \mathbb{G}_T,$$

$$\mathsf{SK} = \left(\Omega, \{y_i\}_{i=1}^{\ell}, \gamma, \{\delta_i, \phi_i\}_{i=1}^{2}, \{f_i\}_{i=1}^{3}, g_1\right) \in \mathbb{Z}_p^{\ell+9} \times \mathbb{G}.$$

Encrypt $(\mathsf{PK}, (\vec{x}, M))$: Let $\vec{x} = (x_1, \ldots, x_\ell) \in \Sigma^\ell$. To encrypt a message $M \in \mathcal{M} \subseteq \mathbb{G}_T$ and the vector $\vec{x}$ under the public key PK, the encryption algorithm picks random exponents $s_1, s_2, s_3, \{\mathrm{tag}_{c,i}\}_{i=1}^{\ell}$ in $\mathbb{Z}_p$ and computes the ciphertext $\mathsf{CT} = \left(C_1, \ldots, \{C_{6,i}, C_{7,i}\}_{i=1}^{\ell}, C_8, C_9, \{\mathrm{tag}_{c,i}\}_{i=1}^{\ell}\right) \in \mathbb{G}^{2\ell+6} \times \mathbb{G}_T \times \mathbb{Z}_p^{\ell}$ as follows:

$$C_1 = W_1^{s_1} F_1^{s_2},\ C_2 = W_2^{s_1} F_2^{s_2},\ C_3 = g_2^{s_1},\ C_4 = g_3^{s_1} F_3^{s_2},\ C_5 = g^{s_2},$$

$$\left\{C_{6,i} = \left(u_i h_i^{x_i} v^{\mathrm{tag}_{c,i}}\right)^{s_2} Y_i^{s_3},\ C_{7,i} = \left(\tau_i \varphi^{\mathrm{tag}_{c,i}}\right)^{s_2}\right\}_{i=1}^{\ell},\ C_8 = g_4^{s_3},\ C_9 = \Lambda^{s_1} M.$$

GenToken $(\mathsf{SK}, \vec{\sigma})$: Let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_\ell) \in \mathcal{I}^\ell$. Let $S(\vec{\sigma})$ be the set of all indexes i such that $\sigma_i \neq *$. To generate a token $\mathsf{TK}_{\vec{\sigma}}$ for the vector $\vec{\sigma}$, the token generation algorithm picks random exponents $r_1, r_2, r_3, r_4, \mathrm{tag}_k \in \mathbb{Z}_p$ and obtains $r_5 \in \mathbb{Z}_p$ such that $(\Sigma_{i \in S(\vec{\sigma})} y_i) r_3 = \gamma r_5$. The algorithm computes the token $\mathsf{TK}_{\vec{\sigma}} = (K_1, \ldots, K_9, \mathrm{tag}_k) \in \mathbb{G}^9 \times \mathbb{Z}_p$ as follows:

$$K_1 = g^{\delta_1 r_1} g^{\phi_1 r_2},\ K_2 = g^{\delta_2 r_1} g^{\phi_2 r_2},\ K_3 = g_1 g^{\Omega r_1},\ K_4 = g^{r_2},\ K_5 = K_1^{f_1} K_2^{f_2} K_4^{-f_3} v^{r_3} \varphi^{r_4},\ K_6$$
$$= \left(\prod_{i \in S(\vec{\sigma})} u_i h_i^{\sigma_i} \cdot v^{\mathrm{tag}_k}\right)^{r_3} \left(\prod_{i \in S(\vec{\sigma})} \tau_i \cdot \varphi^{\mathrm{tag}_k}\right)^{r_4},\ K_7 = g^{r_3},\ K_8 = g^{r_4},\ K_9 = g^{r_5}.$$

Decrypt $(\mathsf{CT}, \mathsf{TK}_{\vec{\sigma}})$: To decrypt a ciphertext $\mathsf{CT} = \left(C_1, \ldots, \{C_{6,i}, C_{7,i}\}_{i=1}^{\ell}, C_8, C_9, \{\mathrm{tag}_{c,i}\}_{i=1}^{\ell}\right)$ using a private key $\mathsf{TK}_{\vec{\sigma}} = (K_1, \ldots, K_9, \mathrm{tag}_k)$, the decryption algorithm sets

$$C_6 = \prod_{i \in S(\vec{\sigma})} C_{6,i},\ C_7 = \prod_{i \in S(\vec{\sigma})} C_{7,i},\ \mathrm{tag}_c = \sum_{i \in S(\vec{\sigma})} \mathrm{tag}_{c,i}.$$

If $\mathrm{tag}_c \neq \mathrm{tag}_k$, the decryption algorithm proceeds as follows:

1. Compute $A_1 = e(C_1, K_1) \cdot e(C_2, K_2)/e(C_3, K_3) \cdot e(C_4, K_4) \cdot e(C_5, K_5)$.
2. Compute

$$A_2 = \left(e(C_6, K_7) \cdot e(C_7, K_8)/e(C_5, K_6) \cdot e(C_8, K_9)\right)^{1/(\mathrm{tag}_c - \mathrm{tag}_k)}.$$

3. Output $M = C_9 \cdot A_1 \cdot A_2$.

Performance: Note that a token consists of 9 group elements in $\mathbb{G}$ plus 1 group element in $\mathbb{Z}_p$, and the decryption algorithm requires 9 pairing operations. These two efficiency factors are independent of the dimension ℓ of the attribute vectors.

3.2. Correctness

We first check that $A_1 = \Lambda^{-s_1} \cdot e(g,v)^{-s_2 r_3} \cdot e(g,\varphi)^{-s_2 r_4}$ as follows:

$$\begin{aligned}
\frac{e(C_1,K_1)\cdot e(C_2,K_2)}{e(C_3,K_3)\cdot e(C_4,K_4)\cdot e(C_5,K_5)} &= \frac{e\left(g^{w_1 s_1} g^{f_1 s_2}, g^{\delta_1 r_1} g^{\phi_1 r_2}\right)\cdot e\left(g^{w_2 s_1} g^{f_2 s_2}, g^{\delta_2 r_1} g^{\phi_2 r_2}\right)}{e(g^{\alpha s_1}, g_1 g^{\Omega r_1})\cdot e(g^{\beta s_1} g^{f_3 s_2}, g^{r_2})\cdot e\left(g^{s_2}, K_1^{f_1} K_2^{f_2} K_4^{-f_3} v^{r_3}\varphi^{r_4}\right)} \\
&= \frac{e\left(g^{(w_1\delta_1 + w_2\delta_2)s_1}, g^{r_1}\right)\cdot e\left(g^{(w_1\phi_1 + w_2\phi_2)s_1}, g^{r_2}\right)\cdot e\left(g^{f_1 s_2}, K_1\right)\cdot e\left(g^{f_2 s_2}, K_2\right)}{e(g^{\alpha s_1}, g_1 g^{\Omega r_1})\cdot e(g^{\beta s_1} g^{f_3 s_2}, g^{r_2})\cdot e\left(g^{s_2}, K_1^{f_1} K_2^{f_2} K_4^{-f_3} v^{r_3}\varphi^{r_4}\right)} \\
&= \frac{e\left(g^{\alpha\Omega s_1}, g^{r_1}\right)\cdot e\left(g^{\beta s_1}, g^{r_2}\right)\cdot e\left(g^{s_2}, K_1^{f_1} K_2^{f_2}\right)}{e(g^{\alpha s_1}, g_1 g^{\Omega r_1})\cdot e(g^{\beta s_1} g^{f_3 s_2}, g^{r_2})\cdot e\left(g^{s_2}, K_1^{f_1} K_2^{f_2} K_4^{-f_3} v^{r_3}\varphi^{r_4}\right)} \\
&= \frac{1}{e(g,g_1)^{\alpha s_1}\cdot e(g,v)^{s_2 r_3}\cdot e(g,\varphi)^{s_2 r_4}}.
\end{aligned}$$

Next, notice that

$$C_6 = \left(\prod_{i\in S(\vec{\sigma})} u_i h_i^{x_i}\cdot v^{\mathrm{tag}_c}\right)^{s_2}\cdot g^{(\Sigma_{i\in S(\vec{\sigma})} y_i) s_3},\ C_7 = \left(\prod_{i\in S(\vec{\sigma})}\tau_i\cdot\varphi^{\mathrm{tag}_c}\right)^{s_2}.$$

Then, if $P_\ell(\vec{\sigma},\vec{x}) = 1$, (i.e., $\sigma_i = x_i$ for all $i \in S(\vec{\sigma})$), we can see that $A_2 = e(g,v)^{s_2 r_3}\cdot e(g,\varphi)^{s_2 r_4}$ by the following computation:

$$\begin{aligned}
\frac{e(C_6,K_7)\cdot e(C_7,K_8)}{e(C_5,K_6)\cdot e(C_8,K_9)} &= \frac{e\left(\left(\prod_{i\in S(\vec{\sigma})} u_i h_i^{x_i}\cdot v^{\mathrm{tag}_c}\right)^{s_2} g^{(\Sigma_{i\in S(\vec{\sigma})} y_i) s_3},\ g^{r_3}\right)\cdot e\left(\left(\prod_{i\in S(\vec{\sigma})}\tau_i\cdot\varphi^{\mathrm{tag}_c}\right)^{s_2}, g^{r_4}\right)}{e\left(g^{s_2}, \left(\prod_{i\in S(\vec{\sigma})} u_i h_i^{\sigma_i}\cdot v^{\mathrm{tag}_k}\right)^{r_3}\left(\prod_{i\in S(\vec{\sigma})}\tau_i\cdot\varphi^{\mathrm{tag}_k}\right)^{r_4}\right)\cdot e(g^{\gamma s_3}, g^{r_5})} \\
&= \frac{e\left(\left(\prod_{i\in S(\vec{\sigma})} u_i h_i^{x_i}\cdot v^{\mathrm{tag}_c}\right)^{r_3}\left(\prod_{i\in S(\vec{\sigma})}\tau_i\cdot\varphi^{\mathrm{tag}_c}\right)^{r_4}, g^{s_2}\right)\cdot e\left(g^{(\Sigma_{i\in S(\vec{\sigma})} y_i) s_3},\ g^{r_3}\right)}{e\left(g^{s_2}, \left(\prod_{i\in S(\vec{\sigma})} u_i h_i^{\sigma_i}\cdot v^{\mathrm{tag}_k}\right)^{r_3}\left(\prod_{i\in S(\vec{\sigma})}\tau_i\cdot\varphi^{\mathrm{tag}_k}\right)^{r_4}\right)\cdot e(g^{\gamma s_3}, g^{r_5})} = \frac{e\left(v^{r_3}\varphi^{r_4}, g^{s_2}\right)^{\mathrm{tag}_c}}{e\left(g^{s_2}, v^{r_3}\varphi^{r_4}\right)^{\mathrm{tag}_k}} \\
&= \left(e(g,v)^{s_2 r_3}\cdot e(g,\varphi)^{s_2 r_4}\right)^{\mathrm{tag}_c - \mathrm{tag}_k}.
\end{aligned}$$

Finally, the message M is correctly recovered as

$$C_9 \cdot A_1 \cdot A_2 = \Lambda^{s_1} M \cdot \Lambda^{-s_1}\cdot e(g,v)^{-s_2 r_3}\cdot e(g,\varphi)^{-s_2 r_4}\cdot e(g,v)^{s_2 r_3}\cdot e(g,\varphi)^{s_2 r_4} = M.$$

Otherwise, if $P_\ell(\vec{\sigma},\vec{x}) = 0$, this means that there is at least one component $\sigma_i \neq x_i$ for some $i \in S(\vec{\sigma})$. Let D be the set of indexes $i \in S(\vec{\sigma})$ such that $\sigma_i \neq x_i$. In this case, the computation above becomes

$$\begin{aligned}
\frac{e(C_6,K_7)\cdot e(C_7,K_8)}{e(C_5,K_6)\cdot e(C_8,K_9)} &= e\left(g^{s_2 r_3}, \prod_{i\in D} h_i^{x_i-\sigma_i}\right)\cdot\left(e(g,v)^{s_2 r_3}\cdot e(g,\varphi)^{s_2 r_4}\right)^{\mathrm{tag}_c - \mathrm{tag}_k} \\
&= e(g,g)^{s_2 r_3 \Sigma_{i\in D}(\log_g h_i)(x_i-\sigma_i)}\cdot\left(e(g,v)^{s_2 r_3}\cdot e(g,\varphi)^{s_2 r_4}\right)^{\mathrm{tag}_c - \mathrm{tag}_k}.
\end{aligned}$$

Thus, the final output becomes M if $\Sigma_{i\in D}(\log_g h_i)(x_i - \sigma_i) = 0$ in $\mathbb{Z}_p$. However, it is computationally hard to find pairs (x_i, σ_i) for $i \in D$ for which such an equality holds. In fact, the probability of a false positive is at most $1/p$ in each decryption.

3.3. Fully secure anonymous IBE scheme

Any HVE scheme implies an anonymous IBE scheme if the vectors $\vec{x}$ and $\vec{\sigma}$ are limited to one dimension. Thus, our HVE scheme provides a new anonymous IBE scheme that is fully secure under standard assumptions such as the DLIN and DBDH assumptions. Prior to our result, several works [9,20,12,42,16,2,17] (including all previous HVE and IPE schemes) have been proposed, but until now there were few anonymous IBE schemes [20,30,35] that achieve full security without using random

oracles. Our new scheme is another example, but is fully secure under the standard assumptions. Compared to [35], which has comparable security, our construction is more efficient in all efficiency respects. Table 2 in Section 5 presents the result by simply assigning the dimension to 1. To demonstrate security of any anonymous IBE scheme, testing for weak attribute hiding is sufficient since two target pairs (ID_b, M_b) for $b = 0,1$ should be equal as long as at least one matching query is asked. Thus, security of our new anonymous IBE scheme is straightforwardly obtained from the proof of Case 1 (defined in the next section) where only non-matching token queries are permitted.[4]

The hierarchical extension of our anonymous IBE scheme can not be realized due to the relation $(\Sigma_{i\in S(\vec{\sigma})} y_i) r_3 = \gamma r_5$. Such a relation plays a key role in the elimination of blinding factors $Y_i^{s_3} = g^{y_i s_3}$. In constructing our anonymous IBE scheme, only y_1 is necessary for one identity and choosing r_3 and r_5 satisfying the relation $y_1 r_3 = \gamma r_5$ can be easily done by the key generation center who knows the exponents $\{y_i\}$ and γ. However, if a key owner (as a parent) wants to generate private keys for its descendants of depth 2, the parent has to select exponents r'_3 and r'_5 such that $y_2 r'_3 = \gamma r'_5$ without knowing y_2 and γ, which is computationally infeasible. Thus, it is still an open problem to construct an anonymous Hierarchical IBE scheme that is fully secure under standard assumptions.

4. Security proof

4.1. Semi-functional algorithms

We now describe the semi-functional ciphertexts and tokens. Their main purpose is to define the structures that will be used in our proof.

4.1.1. Semi-functional ciphertexts

The algorithm first runs the encryption algorithm to generate a normal ciphertext $\mathsf{CT} = \left(C'_1, \ldots, \left\{C'_{6,i}, C'_{7,i}\right\}_{i=1}^{\ell}, C'_8, C'_9, \{\mathrm{tag}_{c,i}\}_{i=1}^{\ell}\right)$ for a vector $\vec{x}$ and a message M. The algorithm selects a random exponent $x \in \mathbb{Z}_p$ and sets

$$C_1 = C'_1 \cdot g^{\delta_2 x}, \quad C_2 = C'_2 \cdot g^{-\delta_1 x}, \quad C_3 = C'_3, \quad C_4 = C'_4 \cdot g^{(\delta_2\phi_1 - \delta_1\phi_2)x},$$

$$C_5 = C'_5, \quad \left\{C_{6,i} = C'_{6,i}, \quad C_{7,i} = C'_{7,i}\right\}_{i=1}^{\ell}, \; C_8 = C'_8, \quad C_9 = C'_9.$$

The semi-functional ciphertext is $\mathsf{CT}^{\mathrm{sf}} = \left(C_1, \ldots, \{C_{6,i}, C_{7,i}\}_{i=1}^{\ell}, C_8, C_9, \{\mathrm{tag}_{c,i}\}_{i=1}^{\ell}\right)$. If one tries to decrypt the semi-functional ciphertext with a normal token $\mathsf{TK}_{\vec{\sigma}}$ for $\vec{\sigma}$, then the decryption would be correctly performed. This stems from the fact that

$$\frac{e(g^{\delta_2 x}, K_1) \cdot e(g^{-\delta_1 x}, K_2)}{e(g^{(\delta_2\phi_1 - \delta_1\phi_2)x}, K_4)} = \frac{e(g^{\delta_2 x}, g^{\delta_1 r_1} g^{\phi_1 r_2}) \cdot e(g^{-\delta_1 x}, g^{\delta_2 r_1} g^{\phi_2 r_2})}{e(g^{(\delta_2\phi_1 - \delta_1\phi_2)x}, g^{r_2})} = 1,$$

where K_1, K_2, and K_4 are components of the normal token.

4.1.2. Semi-functional tokens

The algorithm first runs the token generation algorithm to generate a normal token $\mathsf{TK}_{\vec{\sigma}} = (K'_1, \ldots, K'_9, \mathrm{tag}_k)$ for a vector $\vec{\sigma}$. Next the algorithm picks a random exponent $\lambda \in \mathbb{Z}_p$ and sets

$$K_1 = K'_1 \cdot g^{-w_2\lambda}, \quad K_2 = K'_2 \cdot g^{w_1\lambda}, \quad K_3 = K'_3, \quad K_4 = K'_4,$$

$$K_5 = K'_5 \cdot g^{(f_2 w_1 - f_1 w_2)\lambda}, \quad K_6 = K'_6, \quad K_7 = K'_7, \quad K_8 = K'_8, \quad K_9 = K'_9.$$

Then, the semi-functional token is $\mathsf{TK}^{\mathrm{sf}}_{\vec{\sigma}} = (K_1, \ldots, K_9, \mathrm{tag}_k)$. Note that the element K_5 becomes $K_5 = K_1^{f_1} K_2^{f_2} K_4^{-f_3} v^{r_3} \varphi^{r_4}$. If one tries to decrypt a normal ciphertext encrypted under $\vec{x}$ with the semi-functional token $\mathsf{TK}^{\mathrm{sf}}_{\vec{\sigma}}$, the decryption would be also correctly performed. This can be checked from the fact that

$$\frac{e(C_1, g^{-w_2\lambda}) \cdot e(C_2, g^{w_1\lambda})}{e(C_5, g^{(f_2 w_1 - f_1 w_2)\lambda})} = \frac{e(g^{w_1 s_1} g^{f_1 s_2}, g^{-w_2\lambda}) \cdot e(g^{w_2 s_1} g^{f_2 s_2}, g^{w_1\lambda})}{e(g^{s_2}, g^{(f_2 w_1 - f_1 w_2)\lambda})} = 1,$$

where C_1, C_2, and C_5 are components of the normal ciphertext.

We note that when a semi-functional token is used to decrypt a semi-functional ciphertext, the semi-functional components in two parts will be computed as follows:

$$\frac{e\left(C'_1 \cdot g^{\delta_2 x}, K'_1 \cdot g^{-w_2\lambda}\right) \cdot e\left(C'_2 \cdot g^{-\delta_1 x}, K'_2 \cdot g^{w_1\lambda}\right)}{e\left(C'_4 \cdot g^{(\delta_2\phi_1 - \delta_1\phi_2)x}, K'_4\right) \cdot e\left(C'_5, K'_5 \cdot g^{(f_2 w_1 - f_1 w_2)\lambda}\right)} = e(g,g)^{-(w_1\delta_1 + w_2\delta_2)x\lambda} = e(g,g)^{-\alpha\Omega x\lambda},$$

which is not equal to 1 in $\mathbb{G}_T$.

[4] The condition should be different when considering security of anonymous Hierarchical IBE, where both matching and non-matching token queries are justified for two identity vectors upon which an adversary wants to challenge.

4.2. Proof of security

In the security game defined in Section 2, the adversary $\mathcal{A}$ outputs two vectors $\vec{x}_0^* = \left(x_{0,1}^*, \ldots, x_{0,\ell}^*\right), \vec{x}_1^* = \left(x_{1,1}^*, \ldots, x_{1,\ell}^*\right) \in \Sigma^\ell$ and two messages $M_0, M_1 \in \mathcal{M}$ as its challenge. The goal of $\mathcal{A}$ is to decide which one of the two pairs $(\vec{x}_0^*, M_0)$ and $(\vec{x}_1^*, M_1)$ is associated with the challenge ciphertext. All tokens will be normal and the challenge ciphertext will also be normal. This is the real security game Game_{Real}. Under the rules of the security game, $\mathcal{A}$ that makes at most q token queries will behave in one of two different ways:

Case 1 $\mathcal{A}$ will make token queries for vectors $\vec{\sigma}_i$ such that $P_\ell(\vec{\sigma}_i, \vec{x}_0^*) = P_\ell(\vec{\sigma}_i, \vec{x}_1^*) = 0$ for *all* $i = 1, \ldots, q$.
Case 2 $\mathcal{A}$ will make token queries for vectors $\vec{\sigma}_i$ such that $P_\ell(\vec{\sigma}_i, \vec{x}_0^*) = P_\ell(\vec{\sigma}_i, \vec{x}_1^*) = 1$ for *at least one* $i \in \{1, \ldots, q\}$. In this case, it should be the case that $M_0 = M_1$.

In our security proof, the simulator needs to guess which case it will be in by flipping a coin. If the guess is wrong, the simulator aborts the simulation and outputs a random bit as its answer. Since the simulator's guess will be independent of which case $\mathcal{A}$ behaves in, the simulation is able to proceed with probability 1/2. Depending on the case by guess, the simulator prepares its simulation differently. We describe the simulator's strategy in two cases.

Case 1: (Proof idea) We first give an idea behind the security proof in Case 1. Since $\mathcal{A}$ cannot make any matching token query, we can adapt a similar proof strategy to one in the Waters' original dual system encryption [46]. That is, we create a sequence of hybrid games, where the challenge ciphertext and all tokens are changed into semi-functional ones, and we can then change the message M_b for a random bit $b \in \{0, 1\}$ into a random message. This is the same as in [46], but the difference is that we create an additional sequence of hybrid games, based on the result of randomizing M_b, in order to change each component of the vector $\vec{x}_b^*$ into a random one. During these two sequences of hybrid games, tag values are crucially used to solve the paradox that happens inevitably when proving full security.

The simulator considers a sequence of hybrid games as follows:

Game_{Real}^1: This is the actual HVE security game in Case 1. All tokens will be normal and the challenge ciphertext will be a normal challenge ciphertext on a pair $(\vec{x}_b^*, M_b)$, where $b \in \{0,1\}$ is a random bit.
Game_0^1: All tokens will be normal, but the challenge ciphertext will be a semi-functional ciphertext on a pair $(\vec{x}_b^*, M_b)$.
$\vdots$ $\vdots$
Game_k^1: The first k token queries will return semi-functional tokens, and the rest of the tokens will be normal. The challenge ciphertext will be a semi-functional ciphertext on a pair $(\vec{x}_b^*, M_b)$.
$\vdots$ $\vdots$
Game_q^1: All tokens will be semi-functional, and the challenge ciphertext will be a semi-functional ciphertext on a pair $(\vec{x}_b^*, M_b)$.
Game_M^1: All tokens will be semi-functional, and the challenge ciphertext will be a semi-functional ciphertext on a pair $(\vec{x}_b^*, R)$, where R is a random message from $\mathcal{M}$.
$\mathsf{Game}_{\vec{x},1}^1$: All tokens will be semi-functional, and the challenge ciphertext will be a semi-functional ciphertext on a pair $\left((r_1, x_{b,2}^*, \ldots, x_{b,\ell}^*), R\right)$, where r_1 is a random element from Σ.
$\vdots$ $\vdots$
$\mathsf{Game}_{\vec{x},\ell}^1$: All tokens will be semi-functional, and the challenge ciphertext will be a semi-functional ciphertext on a pair $((r_1, r_2, \ldots, r_\ell), R)$, where all r_i for $i = 1, \ldots, \ell$ are random elements from Σ.

In Game_{Real}^1, the normal challenge ciphertext corresponding to $(\vec{x}_b^*, M_b)$ is given to the adversary. On the other hand, in $\mathsf{Game}_{\vec{x},\ell}^1$, the challenge ciphertext given to the adversary is a semi-functional ciphertext corresponding to $((r_1, \ldots, r_\ell), R)$ that leaks no information about $(\vec{x}_b^*, M_b)$. We will show that no polynomial time adversary is able to distinguish between Game_{Real}^1 and $\mathsf{Game}_{\vec{x},\ell}^1$ by proving that the transitions between the sequence of games above are all computationally indistinguishable under the DLIN and DBDH assumptions.

Lemma 1. *Suppose that the DLIN assumption holds. Then no polynomial time adversary $\mathcal{A}$ can distinguish between Game_{Real}^1 and Game_0^1 with non-negligible advantage.*

Proof. Suppose that there exists an adversary $\mathcal{A}$ which can attack our HVE scheme with non-negligible advantage ϵ. We describe an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the DLIN problem with advantage ϵ. On input $(g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_4}, Z) \in \mathbb{G}^6$, $\mathcal{B}$'s goal is to output 1 if $Z = g^{z_2(z_3+z_4)}$ and 0 otherwise. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

Setup $\mathcal{B}$ selects random exponents $\Omega, \gamma, \{y_i, \mu_i, t_i, \zeta_i\}_{i=1}^{\ell}, \{\gamma_i\}_{i=1}^{3}, \{w_i, \delta_i, \phi_i\}_{i=1}^{2}, \{f_i\}_{i=1}^{3}$ in $\mathbb{Z}_p$, such that $w_1\delta_1 + w_2\delta_2 = \Omega$. $\mathcal{B}$ sets

$$W_1 = (g^{z_2})^{\delta_2}(g^{z_1})^{w_1}, \quad W_2 = (g^{z_2})^{-\delta_1}(g^{z_1})^{w_2},$$

$$F_1 = (g^{z_2})^{\delta_2} g^{f_1}, \quad F_2 = (g^{z_2})^{-\delta_1} g^{f_2}, \quad F_3 = (g^{z_2})^{\delta_2\phi_1 - \delta_1\phi_2} g^{f_3},$$

$$g_2 = g^{z_1}, \quad g_3 = (g^{z_2})^{\delta_2\phi_1 - \delta_1\phi_2}(g^{z_1})^{w_1\phi_1 + w_2\phi_2}, \quad g_4 = g^{\gamma},$$

$$Y_i = g^{y_i}, \quad u_i = g^{\mu_i}, \quad h_i = g^{t_i}, \quad \tau_i = g^{\zeta_i} \quad (i = 1, \ldots, \ell),$$

$$v = g^{\gamma_2}, \quad \varphi = g^{\gamma_3}, \quad \Lambda = e(g^{\gamma_1}, g^{z_1}).$$

$\mathcal{B}$ (implicitly) sets

$$\tilde{w}_1 = \delta_2 z_2 + w_1 z_1, \quad \tilde{w}_2 = -\delta_1 z_2 + w_2 z_1, \quad \tilde{f}_1 = \delta_2 z_2 + f_1, \; \tilde{f}_2 = -\delta_1 z_2 + f_2$$

$$\tilde{f}_3 = (\delta_2\phi_1 - \delta_1\phi_2) z_2 + f_3, \quad \beta = (\delta_2\phi_1 - \delta_1\phi_2) z_2 + (w_1\phi_1 + w_2\phi_2) z_1,$$

$$\alpha = z_1, \quad g_1 = g^{\gamma_1}.$$

Notice that each public key element is independently and uniformly distributed as in the actual construction. Also, we can see that

$$\tilde{w}_1\delta_1 + \tilde{w}_2\delta_2 = (\delta_2 z_2 + w_1 z_1)\delta_1 + (-\delta_1 z_2 + w_2 z_1)\delta_2 = (\delta_1 w_1 + \delta_2 w_2) z_1 = \Omega\alpha,$$

$$\tilde{w}_1\phi_1 + \tilde{w}_2\phi_2 = (\delta_2 z_2 + w_1 z_1)\phi_1 + (-\delta_1 z_2 + w_2 z_1)\phi_2 = (\delta_2\phi_1 - \delta_1\phi_2) z_2 + (w_1\phi_1 + w_2\phi_2) z_1 = \beta.$$

Key Generation Phases 1 and 2 $\mathcal{A}$ issues token queries for vectors $\{\vec{\sigma}_i\}$. For any queried vector $\vec{\sigma}_i$, it is easy for $\mathcal{B}$ to generate a normal token $\mathsf{TK}_{\vec{\sigma}_i}$, since it knows exponents $\Omega, \gamma, \{y_i\}_{i=1}^{\ell}, \{\delta_i, \phi_i\}_{i=1}^{2}$. It selects random r_1, r_2, r_3, r_4, r_5 (subject to the equation $(\sum_{i \in S(\vec{\sigma}_i)} y_i) r_3 = \gamma r_5$), $\mathsf{tag}_k \in \mathbb{Z}_p$ and computes a normal token.

Challenge Ciphertext $\mathcal{A}$ outputs two vectors $\vec{x}_0^*, \vec{x}_1^*$ and two messages M_0, M_1. $\mathcal{B}$ flips a random coin $b \in \{0,1\}$ and picks random $s_3, \{\mathsf{tag}_{c,i}\}_{i=1}^{\ell}$ in $\mathbb{Z}_p$. To generate a challenge ciphertext for $\left(\vec{x}_b^* = (x_{b,1}^*, \ldots, x_{b,\ell}^*), M_b\right)$, $\mathcal{B}$ implicitly sets $s_1 = z_3$ and $s_2 = z_4$. $\mathcal{B}$ computes $C_3, C_5, \{C_{6,i}, C_{7,i}\}_{i=1}^{\ell}, C_8$, and C_9 elements as

$$C_3 = g^{z_1 z_3} = g_2^{s_1}, \quad C_5 = g^{z_4} = g^{s_2},$$

$$C_{6,i} = (g^{z_4})^{\mu_i + t_i x_{b,i}^* + \gamma_2 \mathsf{tag}_{c,i}} g^{y_i s_3} = (u_i h_i^{x_{b,i}^*} v^{\mathsf{tag}_{c,i}})^{s_2} Y_i^{s_3},$$

$$C_{7,i} = (g^{z_4})^{\zeta_i + \gamma_3 \mathsf{tag}_{c,i}} = (\tau_i \varphi^{\mathsf{tag}_{c,i}})^{s_2},$$

$$C_8 = g^{\gamma s_3} = g_4^{s_3}, \quad C_9 = e(g^{z_1 z_3}, g^{\gamma_1}) M_b = e(g_1, g_2)^{s_1} M_b.$$

Next, $\mathcal{B}$ computes C_1, C_2, and C_4 elements as follows:

$$C_1 = (g^{z_1 z_3})^{w_1} (g^{z_4})^{f_1} Z^{\delta_2},$$

$$C_2 = (g^{z_1 z_3})^{w_2} (g^{z_4})^{f_2} Z^{-\delta_1},$$

$$C_4 = (g^{z_1 z_3})^{w_1\phi_1 + w_2\phi_2} (g^{z_4})^{f_3} Z^{(\delta_2\phi_1 - \delta_1\phi_2)}.$$

If $Z = g^{z_2(z_3+z_4)}$, then we have that

$$C_1 = (g^{z_1 z_3})^{w_1} (g^{z_4})^{f_1} (g^{z_2(z_3+z_4)})^{\delta_2} = g^{(\delta_2 z_2 + w_1 z_1) z_3} g^{(\delta_2 z_2 + f_1) z_4} = W_1^{s_1} F_1^{s_2},$$

$$C_2 = (g^{z_1 z_3})^{w_2} (g^{z_4})^{f_2} (g^{z_2(z_3+z_4)})^{-\delta_1} = g^{(-\delta_1 z_2 + w_2 z_1) z_3} g^{(-\delta_1 z_2 + f_2) z_4} = W_2^{s_1} F_2^{s_2},$$

$$C_4 = (g^{z_1 z_3})^{w_1\phi_1 + w_2\phi_2} (g^{z_4})^{f_3} (g^{z_2(z_3+z_4)})^{(\delta_2\phi_1 - \delta_1\phi_2)} = g^{[(\delta_2\phi_1 - \delta_1\phi_2) z_2 + (w_1\phi_1 + w_2\phi_2) z_1] z_3} g^{[(\delta_2\phi_1 - \delta_1\phi_2) z_2 + f_3] z_4} = g_3^{s_1} F_3^{s_2}.$$

In this case, the ciphertext will have the same distribution as a normal ciphertext. Thus, $\mathcal{B}$ is playing Game_{Real} with $\mathcal{A}$. On the other hand, if $Z = g^{z_2(z_3+z_4)} g^{\pi}$ for some (non-zero) random $\pi \in \mathbb{Z}_p$, then

$$C_1 = (g^{z_1 z_3})^{w_1} (g^{z_4})^{f_1} (g^{z_2(z_3+z_4)} g^{\pi})^{\delta_2} = W_1^{s_1} F_1^{s_2} \cdot g^{\delta_2 x},$$

$$C_2 = (g^{z_1 z_3})^{w_2} (g^{z_4})^{f_2} (g^{z_2(z_3+z_4)} g^{\pi})^{-\delta_1} = W_2^{s_1} F_2^{s_2} \cdot g^{-\delta_1 x},$$

$$C_4 = (g^{z_1 z_3})^{w_1\phi_1 + w_2\phi_2} (g^{z_4})^{f_3} (g^{z_2(z_3+z_4)} g^{\pi})^{(\delta_2\phi_1 - \delta_1\phi_2)} = g_3^{s_1} F_3^{s_2} \cdot g^{(\delta_2\phi_1 - \delta_1\phi_2)x},$$

where the exponent π plays the role of x. In this case, the ciphertext will have the same distribution as a semi-functional ciphertext. Thus, $\mathcal{B}$ is playing Game_0 with $\mathcal{A}$.

Guess $\mathcal{B}$ receives a bit $b' \in \{0, 1\}$ and outputs 0 if $b' = b$.

Analysis As mentioned above, if $Z = g^{z_2(z_3+z_4)}$ the challenge ciphertext is distributed exactly as in Game_{Real}, whereas if $Z = g^{z_2(z_3+z_4)} g^{\pi}$ the challenge ciphertext is distributed exactly as in Game_0. It follows that under the DLIN assumption, these two games are indistinguishable. □

Let $k = 1, \ldots, q$.

Lemma 2. *Suppose that the DLIN assumption holds. Then no polynomial time adversary $\mathcal{A}$ can distinguish between* Game_{k-1}^1 *and* Game_k^1 *with non-negligible advantage.*

Proof. Suppose that there exists an adversary $\mathcal{A}$ which can attack our HVE scheme with non-negligible advantage ϵ. We describe an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the DLIN problem with advantage ϵ. On input $(g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_4}, Z) \in \mathbb{G}^6$, $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

Setup $\mathcal{B}$ selects random exponents $\alpha,\ \{A_i,\ B_i,\ y_i,\ \mu_i,\ t_i,\ \zeta_i\}_{i=1}^{\ell},\ \gamma,\ \{\gamma_i\}_{i=1}^{3},\ \{w_i,\ \delta_i,\ \phi_i\}_{i=1}^{2},\ \{f_i\}_{i=1}^{3}$ in $\mathbb{Z}_p$, such that $w_1\delta_1 + w_2\delta_2 = \alpha$. (Here, we can exclude the unlikely event that $\gamma = 0$ and $f_3 = 0$ in $\mathbb{Z}_p$.) $\mathcal{B}$ sets $w_1\phi_1 + w_2\phi_2 = \beta$ and

$$W_1 = g^{w_1}, \quad W_2 = g^{w_2},$$

$$F_1 = g^{f_1}, \quad F_2 = g^{f_2}, \quad F_3 = (g^{z_1})^{f_3},$$

$$g_2 = g^{\alpha}, \quad g_3 = g^{\beta}, \quad g_4 = g^{\gamma},$$

$$Y_i = g^{y_i}, \quad u_i = (g^{z_1})^{-A_i} g^{\mu_i}, \quad h_i = (g^{z_1})^{-B_i} g^{t_i}, \quad \tau_i = g^{\zeta_i} \quad i = 1, \ldots, \ell),$$

$$v = g^{z_1} g^{\gamma_2}, \quad \varphi = g^{\gamma_3}, \quad \Lambda = e(g^{\gamma_1}, g^{\alpha}).$$

$\mathcal{B}$ (implicitly) sets

$$\tilde{\delta}_1 = -w_2 z_2 + \delta_1 z_1, \quad \tilde{\delta}_2 = w_1 z_2 + \delta_2 z_1, \quad \tilde{\phi}_1 = -w_2 z_2 + \phi_1, \quad \tilde{\phi}_2 = w_1 z_2 + \phi_2, \quad \tilde{f}_3 = f_3 z_1, \quad \Omega = z_1,\ g_1 = g^{\gamma_1}.$$

Notice that each public key element is independently and uniformly distributed as in the actual construction. Also, we can see that

$$w_1\tilde{\delta}_1 + w_2\tilde{\delta}_2 = w_1(-w_2 z_2 + \delta_1 z_1) + w_2(w_1 z_2 + \delta_2 z_1) = (w_1\delta_1 + w_2\delta_2) z_1 = \alpha\Omega,$$

$$w_1\tilde{\phi}_1 + w_2\tilde{\phi}_2 = w_1(-w_2 z_2 + \phi_1) + w_2(w_1 z_2 + \phi_2) = w_1\phi_1 + w_2\phi_2 = \beta.$$

Key Generation Phases $\mathcal{A}$ issues token queries for vectors $\{\vec{\sigma}_i\}$. $\mathcal{B}$ breaks the token generation phases into three cases. Consider ith query issued by $\mathcal{A}$.

Case I: $i > k$.
$\mathcal{B}$ generates a normal token for the requested vector $\vec{\sigma}_i$. $\mathcal{B}$ picks random exponents r_1, r_2, r_3, r_4, r_5 (subject to the equation $(\Sigma_{i \in S(\vec{\sigma}_i)} y_i) r_3 = \gamma r_5$), $\mathrm{tag}_k \in \mathbb{Z}_p$ and performs the usual token generation procedures. Note that even though $\tilde{\delta}_1, \tilde{\delta}_2, \tilde{\phi}_1, \tilde{\phi}_2, \Omega$, and $\tilde{f}_3$ are unknown to $\mathcal{B}$, it is easy to compute a normal token.

Case II: $i < k$.
$\mathcal{B}$ generates a semi-functional token for the requested vector $\vec{\sigma}_i$. $\mathcal{B}$ picks random exponents r_1, r_2, r_3, r_4, r_5 (subject to the equation $(\Sigma_{i \in S(\vec{\sigma}_i)} y_i) r_3 = \gamma r_5$), $\mathrm{tag}_k, \lambda \in \mathbb{Z}_p$ and generates the semi-functional token generation procedures. Since $\mathcal{B}$ knows exponents w_1, w_2, f_1, and f_2, it is easy to compute a semi-functional token.

Case III: $i = k$.
For the requested vector $\vec{\sigma}_i$, $\mathcal{B}$ picks random exponents $r_3, r_4 \in \mathbb{Z}_p$ and sets

$$\mathrm{tag}_k = \Sigma_{j \in S(\vec{\sigma}_i)} (A_j + B_j \cdot \sigma_{i,j}),$$

where $\sigma_{i,j}$ is a non-wildcard component in $\vec{\sigma}_i$. It (implicitly) sets $\tilde{r}_1 = z_3$, $\tilde{r}_2 = z_4$, $\tilde{r}_3 = f_3 z_4 + r_3$, and $\tilde{r}_5 = (\Sigma_{j \in S(\vec{\sigma}_i)} y_j)(f_3 z_4 + r_3)/\gamma$. Note that the equation $(\Sigma_{j \in S(\vec{\sigma}_i)} y_j)\tilde{r}_3 = \gamma \tilde{r}_5$ is satisfied. $\mathcal{B}$ generates K_3, K_4, K_6, K_7, K_8, and K_9 elements as follows:

$$K_3 = g^{\gamma_1} g^{z_1 z_3} = g_1 g^{\Omega \tilde{r}_1}, \quad K_4 = g^{z_4} = g^{\tilde{r}_2},$$

$$K_6 = (g^{z_4})^{[(\Sigma_{j\in S(\vec{\sigma}_i)}(\mu_i + t_i\sigma_{i,j}))+\gamma_2 \mathrm{tag}_k]f_3} g^{[(\Sigma_{j\in S(\vec{\sigma}_i)}(\mu_i + t_i\sigma_{ij}))+\gamma_2 \mathrm{tag}_k]r_3} g^{[(\Sigma_{j\in S(\vec{\sigma}_i)}\zeta_i)+\gamma_3 \mathrm{tag}_k]r_4} = \left(od_{j\in S(\vec{\sigma}_i)} u_i h_i^{\sigma_{i,j}} \cdot v^{\mathrm{tag}_k}\right)^{\tilde{r}_3} \left(\prod_{j\in S(\vec{\sigma}_i)} \tau_i \cdot \varphi^{\mathrm{tag}_k}\right)^{r_4},$$

$$K_7 = (g^{z_4})^{f_3} g^{r_3} = g^{\tilde{r}_3}, \quad K_8 = g^{r_4},$$

$$K_9 = (g^{z_4})^{(\Sigma_{j\in S(\vec{\sigma}_i)} y_i) f_3/\gamma} g^{(\Sigma_{j\in S(\vec{\sigma}_i)} y_i) r_3/\gamma} = g^{\tilde{r}_5}.$$

Next, $\mathcal{B}$ generates K_1, K_2, and K_5 elements as

$$K_1 = Z^{-w_2} (g^{z_1 z_3})^{\delta_1} (g^{z_4})^{\phi_1}, \quad K_2 = Z^{w_1} (g^{z_1 z_3})^{\delta_2} (g^{z_4})^{\phi_2},$$

$$K_5 = K_1^{f_1} K_2^{f_2} (g^{z_1})^{r_3} (g^{z_4})^{\gamma_2 f_3} g^{\gamma_2 r_3} g^{\gamma_3 r_4} = K_1^{f_1} K_2^{f_2} K_4^{-\tilde{f}_3} v^{\tilde{r}_3} \varphi^{r_4}.$$

If $Z = g^{z_2(z_3+z_4)}$, then $\mathcal{B}$ has that

$$K_1 = (g^{z_2(z_3+z_4)})^{-w_2} (g^{z_1 z_3})^{\delta_1} (g^{z_4})^{\phi_1} = g^{(-w_2 z_2 + \delta_1 z_1) z_3} g^{(-w_2 z_2 + \phi_1) z_4} = g^{\tilde{\delta}_1 r_1} g^{\tilde{\phi}_1 r_2},$$

$$K_2 = (g^{z_2(z_3+z_4)})^{w_1} (g^{z_1 z_3})^{\delta_2} (g^{z_4})^{\phi_2} = g^{(w_1 z_2 + \delta_2 z_1) z_3} g^{(w_1 z_2 + \phi_2) z_4} = g^{\tilde{\delta}_2 r_1} g^{\tilde{\phi}_2 r_2}.$$

In this case, $\mathcal{B}$ generates the ith token as a normal token, so $\mathcal{B}$ plays Game_{k-1} with $\mathcal{A}$. On the other hand, if $Z = g^{z_2(z_3+z_4)} g^{\pi}$ for some (non-zero) $\pi \in \mathbb{Z}_p$, then $\mathcal{B}$ has that

$$K_1 = (g^{z_2(z_3+z_4)} g^{\pi})^{-w_2} (g^{z_1 z_3})^{\delta_1} (g^{z_4})^{\phi_1} = g^{\tilde{\delta}_1 r_1} g^{\tilde{\phi}_1 r_2} \cdot g^{-w_2 \pi},$$

$$K_2 = (g^{z_2(z_3+z_4)} g^{\pi})^{w_1} (g^{z_1 z_3})^{\delta_2} (g^{z_4})^{\phi_2} = g^{\tilde{\delta}_2 r_1} g^{\tilde{\phi}_2 r_2} \cdot g^{w_1 \pi},$$

where π plays a random exponent λ in $\mathbb{Z}_p$. In this case, $\mathcal{B}$ generates the ith token as a semi-functional token, so $\mathcal{B}$ plays Game_k with $\mathcal{A}$.

Challenge Ciphertext $\mathcal{A}$ outputs two vectors $\vec{x}_0^*, \vec{x}_1^*$ and two messages M_0, M_1. $\mathcal{B}$ picks a random bit $b \in \{0,1\}$ and random exponents $s_1, s_2, s_3, x \in \mathbb{Z}_p$. $\mathcal{B}$ (implicitly) sets $\tilde{s}_2 = f_3^{-1} \alpha x z_2 + s_2$ and $\mathrm{tag}_{c,i}^* = A_i + B_i \cdot x_{b,i}^* \in \mathbb{Z}_p$ for $i = 1, \ldots, \ell$, where $\vec{x}_b^* = \left(x_{b,1}^*, \ldots, x_{b,\ell}^*\right)$. $\mathcal{B}$ computes a semi-functional ciphertext for $(\vec{x}_b^*, M_b)$ as follows:

$$C_1 = g^{w_1 s_1} (g^{z_2})^{f_1 f_3^{-1} \alpha x} g^{f_1 s_2} \cdot (g^{z_2})^{w_1 x} (g^{z_1})^{\delta_2 x} = W_1^{s_1} F_1^{\tilde{s}_2} \cdot g^{\tilde{\delta}_2 x},$$

$$C_2 = g^{w_2 s_1} (g^{z_2})^{f_2 f_3^{-1} \alpha x} g^{f_2 s_2} \cdot (g^{z_2})^{w_2 x} (g^{z_1})^{-\delta_1 x} = W_2^{s_1} F_2^{\tilde{s}_2} \cdot g^{-\tilde{\delta}_1 x},$$

$$C_3 = g^{\alpha s_1} = g_2^{s_1},$$

$$C_4 = g^{\beta s_1} (g^{z_1})^{f_3 s_2} \cdot (g^{z_2})^{(w_1\phi_1 + w_2\phi_2)x} (g^{z_1})^{(\delta_2\phi_1 - \delta_1\phi_2)x} = g_3^{s_1} F_3^{\tilde{s}_2} \cdot g^{(\tilde{\delta}_2\tilde{\phi}_1 - \tilde{\delta}_1\tilde{\phi}_2)x},$$

$$C_5 = (g^{z_2})^{f_3^{-1}\alpha x} g^{s_2} = g^{\tilde{s}_2},$$

$$C_{6,i} = (g^{z_2})^{(\mu_i + t_i x_{b,i}^* + \gamma_2 \mathrm{tag}_{c,i}^*) f_3^{-1} \alpha x} g^{(\mu_i + t_i x_{b,i}^* + \gamma_2 \mathrm{tag}_{c,i}^*) s_2} g^{y_i s_3} = \left(u_i h_i^{x_{b,i}^*} v^{\mathrm{tag}_{c,i}^*}\right)^{\tilde{s}_2} Y_i^{s_3},$$

$$C_{7,i} = (g^{z_2})^{(\zeta_i + \mathrm{tag}_{c,i}^* \gamma_3) \alpha x f_3^{-1}} g^{(\zeta_i + \mathrm{tag}_{c,i}^* \gamma_3) s_2} = \left(\tau_i \varphi^{\mathrm{tag}_{c,i}^*}\right)^{\tilde{s}_2},$$

$$C_8 = g^{\gamma s_3} = g_4^{s_3},$$

$$C_9 = e(g,g)^{\gamma_1 \alpha s_1} M_b = e(g_1, g_2)^{s_1} M_b.$$

In computing C_4, we have that

$$\tilde{\delta}_2\tilde{\phi}_1 - \tilde{\delta}_1\tilde{\phi}_2 = (w_1 z_2 + \delta_2 z_1)(-w_2 z_2 + \phi_1) - (-w_2 z_2 + \delta_1 z_1)(w_1 z_2 + \phi_2) = (w_1\phi_1 + w_2\phi_2) z_2 + (\delta_2\phi_1 - \delta_1\phi_2) z_1 - \alpha z_1 z_2,$$

where $\alpha = w_1\delta_1 + w_2\delta_2$. Then, the unknown term $(g^{z_1 z_2})^{-\alpha x}$ is canceled out by the opposite term $(g^{z_1 z_2})^{\alpha x}$ that comes from the calculation of $F_3^{\tilde{s}_2} = (g^{f_3 z_1})^{f_3^{-1}\alpha x z_2 + s_2}$. Also, in computing $C_{6,i}$, the term g^{z_1} inserted into u_i, h_i, and v_i is also canceled out by setting $\mathrm{tag}_{c,i}^* = A_i + B_i \cdot x_{b,i}^*$ for $i = 1, \ldots, \ell$. Note that the values $\{A_i, B_i\}_{i=1}^{\ell}$ are information-theoretically hidden to the adversary, and

because of the restriction $P_\ell(\vec{\sigma}_i, \vec{x}_0^*) = P_\ell(\vec{\sigma}_i, \vec{x}_1^*) = 0$ in the security model, there must be at least one component $\sigma_{i,j}$ such that $\sigma_{i,j} \neq x_{b,j}^*$, so that the value $\mathrm{tag}_k = \Sigma_{j \in S(\vec{\sigma}_i)}(A_j + B_j \cdot \sigma_{i,j})$ from the ith query must include at least one value $A_j + B_j \cdot \sigma_{i,j}$ for the component $\sigma_{i,j} \neq x_{b,j}^*$. Because of this existence, the adversary cannot identify any special relationship between $\left\{\mathrm{tag}_{c,i}^*\right\}_{i=1}^{\ell}$ and tag_k (from the ith query).

Guess $\mathcal{B}$ receives a bit $b' \in \{0,1\}$ and outputs 0 if $b' = b$.

Analysis As mentioned above, if $Z = g^{z_2(z_3+z_4)}, \mathcal{B}$ is in Game_{k-1}, whereas if $Z = g^{z_2(z_3+z_4)}g^{\pi}, \mathcal{B}$ is in Game_k. It follows that under the DLIN assumption, these two games are indistinguishable. □

Lemma 3. *Suppose that the DBDH assumption holds. Then no polynomial time adversary $\mathcal{A}$ can distinguish between* Game_q^1 *and* Game_M^1 *with non-negligible advantage.*

Proof. Suppose that there exists an adversary $\mathcal{A}$ which can attack our HVE scheme with non-negligible advantage ϵ. We describe an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the DBDH problem with advantage ϵ. On input $(g,\ g^a,\ g^b,\ g^c,\ Z) \in \mathbb{G}^4 \times \mathbb{G}_T$, $\mathcal{B}$'s goal is to output 1 if $Z = e(g,g)^{abc}$ and 0 otherwise. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

Setup $\mathcal{B}$ selects random exponents $\alpha,\ \Omega,\ \gamma,\ \{y_i,\ \mu_i,\ t_i,\ \zeta_i\}_{i=1}^{\ell},\ \{\gamma_i\}_{i=1}^{3}, \{w_i,\ \delta_i,\ \phi_i\}_{i=1}^{2},\ \{f_i\}_{i=1}^{3}$ in $\mathbb{Z}_p$, such that $w_1\delta_1 + w_2\delta_2 = \alpha\Omega$ (where $\Omega \neq 0$).

$\mathcal{B}$ sets

$$W_1 = (g^b)^{\delta_2} g^{w_1}, \quad W_2 = (g^b)^{-\delta_1} g^{w_2}, \quad F_1 = g^{f_1}, \quad F_2 = g^{f_2}, \quad F_3 = g^{f_3},$$

$$g_2 = g^{\alpha}, \quad g_3 = (g^b)^{\delta_2\phi_1 - \delta_1\phi_2} g^{w_1\phi_1 + w_2\phi_2}, \quad g_4 = g^{\gamma},$$

$$Y_i = g^{y_i}, \quad u_i = g^{\mu_i}, \quad h_i = g^{t_i}, \quad \tau_i = g^{\zeta_i} \quad (i = 1, \ldots, \ell),$$

$$v = g^{\gamma_2}, \quad \varphi = g^{\gamma_3}, \quad \Lambda = e(g^a, g^b)^{\Omega} \cdot e(g,g)^{\gamma_1}.$$

$\mathcal{B}$ (implicitly) sets

$$\tilde{w}_1 = \delta_2 b + w_1, \quad \tilde{w}_2 = -\delta_1 b + w_2, \quad \beta = (\delta_2\phi_1 - \delta_1\phi_2)b + (w_1\phi_1 + w_2\phi_2),$$

$$g_1 = g^{\Omega ab} g^{\gamma_1}.$$

Notice that each public key element is independently and uniformly distributed as in the actual construction. Also, we can see that

$$\tilde{w}_1\delta_1 + \tilde{w}_2\delta_2 = (\delta_2 b + w_1)\delta_1 + (-\delta_1 b + w_2)\delta_2 = w_1\delta_1 + w_2\delta_2 = \alpha\Omega,$$

$$\tilde{w}_1\phi_1 + \tilde{w}_2\phi_2 = (\delta_2 b + w_1)\phi_1 + (-\delta_1 b + w_2)\phi_2 = (\delta_2\phi_1 - \delta_1\phi_2)b + (w_1\phi_1 + w_2\phi_2) = \beta.$$

Key Generation Phases 1 and 2 $\mathcal{A}$ issues token queries for vectors. For any queried vector $\vec{\sigma}_i, \mathcal{B}$ generates a semi-functional token $\mathsf{TK}_{\vec{\sigma}_i}^{\mathrm{sf}}$. It selects random r_1, r_2, r_3, r_4, r_5 (subject to the equation $(\Sigma_{i \in S(\vec{\sigma}_i)} y_i) r_3 = \gamma r_5$), tag_k, $\lambda \in \mathbb{Z}_p$. $\mathcal{B}$ implicitly sets

$$\tilde{r}_1 = -ab + r_1, \quad \tilde{\lambda} = a + \lambda.$$

It computes the token as follows:

$$K_1 = (g^b)^{\delta_1\lambda}(g^a)^{-w_2} g^{\delta_1 r_1} g^{\phi_1 r_2} g^{-w_2\lambda}, \quad K_2 = (g^b)^{\delta_2\lambda}(g^a)^{w_1} g^{\delta_2 r_1} g^{\phi_2 r_2} g^{w_1\lambda},$$

$$K_3 = g^{\gamma_1} g^{\Omega r_1}, \quad K_4 = g^{r_2}, \quad K_5 = K_1^{f_1} K_2^{f_2} K_4^{-f_3} v^{r_3} \varphi^{r_4},$$

$$K_6 = \left(\prod_{j \in S(\vec{\sigma}_i)} u_j h_j^{\sigma_j} \cdot v^{\mathrm{tag}_k} \right)^{r_3} \left(\prod_{j \in S(\vec{\sigma}_i)} \tau_j \cdot \varphi^{\mathrm{tag}_k} \right)^{r_4},$$

$$K_7 = g^{r_3}, \quad K_8 = g^{r_4}, \quad K_9 = g^{r_5}.$$

The validity of K_1, K_2, K_3, K_5, and K_6 elements can be checked as follows:

$$K_1 = (g^b)^{\delta_1\lambda}(g^a)^{-w_2} g^{\delta_1 r_1} g^{\phi_1 r_2} g^{-w_2\lambda} = g^{\delta_1(-ab+r_1)} g^{\phi_1 r_2} \cdot g^{-(-\delta_1 b + w_2)(a+\lambda)} = g^{\delta_1 \tilde{r}_1} g^{\phi_1 r_2} \cdot g^{-\tilde{w}_2\tilde{\lambda}},$$

$$K_2 = (g^b)^{\delta_2\lambda}(g^a)^{w_1} g^{\delta_2 r_1} g^{\phi_2 r_2} g^{w_1\lambda} = g^{\delta_2(-ab+r_1)} g^{\phi_2 r_2} \cdot g^{(\delta_2 b + w_1)(a+\lambda)} = g^{\delta_2 \tilde{r}_1} g^{\phi_2 r_2} \cdot g^{\tilde{w}_1\tilde{\lambda}},$$

$$K_3 = g^{\gamma_1} g^{\Omega r_1} = g^{\Omega ab} g^{\gamma_1} g^{\Omega(-ab+r_1)} = g_1 g^{\Omega \tilde{r}_1}.$$

Observe that the unknown term g^{ab} is canceled out in K_1, K_2, and K_3 elements, respectively.

Challenge Ciphertext $\mathcal{A}$ outputs two vectors $\vec{x}_0^*, \vec{x}_1^*$ and two messages M_0, M_1. $\mathcal{B}$ then flips a random coin $\beta \in \{0,1\}$. $\mathcal{B}$ picks random exponents $\{\mathrm{tag}_{c,i}^*\}_{i=1}^{\ell}, s_2, s_3, x \in \mathbb{Z}_p$. $\mathcal{B}$ implicitly sets $\tilde{s}_1 = c$ and $\tilde{x} = -bc + x$. $\mathcal{B}$ computes a semi-functional ciphertext under $(\vec{x}_\beta^* = \left(x_{\beta,1}^*, \ldots, x_{\beta,\ell}^*\right), M_\beta)$ as follows:

$$C_1 = (g^c)^{w_1} g^{f_1 s_2} g^{\delta_2 x}, \quad C_2 = (g^c)^{w_2} g^{f_2 s_2} g^{-\delta_1 x}, \quad C_3 = (g^c)^{\alpha} = g_2^{\tilde{s}_1},$$

$$C_4 = (g^c)^{w_1\phi_1 + w_2\phi_2} g^{f_3 s_2} g^{(\delta_2\phi_1 - \delta_1\phi_2)x}, \quad C_5 = g^{s_2},$$

$$C_{6,i} = \left(u_i h_i^{x_{\beta,i}^*} v^{\mathrm{tag}_{c,i}^*}\right)^{s_2} Y_i^{s_3}, \quad C_{7,i} = \left(\tau_i \varphi^{\mathrm{tag}_{c,i}^*}\right)^{s_2}, \quad C_8 = g_4^{s_3},$$

$$C_9 = Z^{\Omega} \cdot e(g, g^c)^{\gamma_1} \cdot M_\beta.$$

The validity of elements C_1, C_2, and C_4 can be checked as follows:

$$C_1 = (g^c)^{w_1} g^{f_1 s_2} g^{\delta_2 x} = g^{(\delta_2 b + w_1)c} g^{f_1 s_2} \cdot g^{\delta_2(-bc+x)} = W_1^{\tilde{s}_1} F_1^{s_2} \cdot g^{\delta_2 \tilde{x}},$$

$$C_2 = (g^c)^{w_2} g^{f_2 s_2} g^{-\delta_1 x} = g^{(-\delta_1 b + w_2)c} g^{f_2 s_2} \cdot g^{-\delta_1(-bc+x)} = W_2^{\tilde{s}_1} F_2^{s_2} \cdot g^{-\delta_1 \tilde{x}},$$

$$C_4 = (g^c)^{w_1\phi_1 + w_2\phi_2} g^{f_3 s_2} g^{(\delta_2\phi_1 - \delta_1\phi_2)x} = g^{[(\delta_2\phi_1 - \delta_1\phi_2)b + (w_1\phi_1 + w_2\phi_2)]c} g^{f_3 s_2} \cdot g^{(\delta_2\phi_1 - \delta_1\phi_2)(-bc+x)} = g_2^{\tilde{s}_1} F_3^{s_2} \cdot g^{(\delta_2\phi_1 - \delta_1\phi_2)\tilde{x}}.$$

If $Z = e(g,g)^{abc}$, then the element C_9 can be computed as follows:

$$C_9 = Z^{\Omega} \cdot e(g, g^c)^{\gamma_1} \cdot M_\beta = e(g,g)^{\Omega abc} \cdot e(g,g)^{\gamma_1 c} \cdot M_\beta = e(g_1, g_2)^{\tilde{s}_1} \cdot M_\beta.$$

In this case, the challenge ciphertext is a valid encryption under $(\vec{x}_\beta^*, M_\beta)$. Thus, $\mathcal{B}$ is playing Game_q with $\mathcal{A}$. On the other hand, if Z is random, C_9 is randomly distributed. Thus, $\mathcal{B}$ is playing Game_{Final^M} with $\mathcal{A}$.

Guess $\mathcal{B}$ receives a bit $\beta' \in \{0,1\}$ and outputs 0 if $\beta' = \beta$.

Analysis As mentioned above, if $Z = e(g,g)^{abc}$, the challenge ciphertext is distributed as in Game_q, whereas if Z is random, the challenge ciphertext is distributed as in Game_{Final^M}. It follows that under the DBDH assumption, these two games are indistinguishable. □

Let $k = 1, \ldots, \ell$ and let $\mathsf{Game}_M^1 = \mathsf{Game}_{\vec{x},0}^1$.

Lemma 4. *Suppose that the DLIN assumption holds. Then no polynomial time adversary $\mathcal{A}$ can distinguish between* $\mathsf{Game}_{\vec{x},k-1}^1$ *and* $\mathsf{Game}_{\vec{x},k}^1$ *with non-negligible advantage.*

Proof. Suppose that there exists an adversary $\mathcal{A}$ which can attack our HVE scheme with non-negligible advantage ϵ. We describe an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the DLIN problem with advantage ϵ. On input $(g,\ g^{z_1},\ g^{z_2},\ g^{z_1 z_3},\ g^{z_4},\ Z) \in \mathbb{G}^6$, $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

Setup $\mathcal{B}$ selects random exponents $\alpha,\ \gamma,\ \{A_i,\ y_i,\ \mu_i,\ t_i,\ \zeta_i\}_{i=1}^{\ell},\ \{\gamma_i\}_{i=1}^{3}, \{w_i,\ \delta_i,\ \phi_i\}_{i=1}^{2},\ \{f_i\}_{i=1}^{3}$ in $\mathbb{Z}_p$, $\mathcal{B}$ sets

$$W_1 = g^{z_1 w_1}, \quad W_2 = g^{w_2}, \quad F_1 = g^{f_1}, \quad F_2 = g^{f_2}, \quad F_3 = (g^{z_1})^{f_3},$$

$$g_2 = g^{\alpha}, \quad g_3 = (g^{z_1})^{\phi_1 + w_2} g^{w_1\phi_1 + w_2\phi_2}, \quad g_4 = g^{z_1},$$

$$Y_i = (g^{z_1})^{y_i} \quad (i = 1, \ldots, k-1, k+1, \ldots, \ell), \quad Y_k = (g^{z_2})^{y_k},$$

$$u_i = g^{\mu_i} \quad (i = 1, \ldots, k-1, k+1, \ldots, \ell), \quad u_k = (g^{z_2})^{y_k} g^{\mu_k},$$

$$h_i = g^{t_i}, \quad \tau_i = (g^{z_2})^{-A_i} g^{\zeta_i}, \quad v = g^{\gamma_2}, \quad \varphi = g^{z_2} g^{\gamma_3}, \quad \Lambda = e(g^{\gamma_1}, g^{\alpha}).$$

Note that the values $\{A_i\}_{i=1}^{\ell}$ are information-theoretically hidden. $\mathcal{B}$ (implicitly) sets

$$\gamma = z_1, \quad \tilde{y}_i = y_i z_1 \quad (i = 1, \ldots, k-1, k+1, \ldots, \ell), \quad \tilde{y}_k = y_k z_2,$$

$$\tilde{f}_3 = f_3 z_1, \quad \tilde{w}_1 = z_1 + w_1, \quad \tilde{\phi}_2 = z_1 + \phi_2, \quad \Omega = (\delta_1 z_1 + w_1\delta_1 + w_2\delta_2)/\alpha,$$

$$\beta = (\phi_1 + w_2) z_1 + (w_1\phi_1 + w_2\phi_2), \quad g_1 = g^{\gamma_1}.$$

Notice that each public key element is independently and uniformly distributed as in the actual construction. Also, we can see that

$$\tilde{w}_1\delta_1 + w_2\delta_2 = (z_1 + w_1)\delta_1 + w_2\delta_2 = \delta_1 z_1 + w_1\delta_1 + w_2\delta_2 = \alpha\Omega,$$

$$\tilde{w}_1\phi_1 + w_2\tilde{\phi}_2 = (z_1 + w_1)\phi_1 + w_2(z_1 + \phi_2) = (\phi_1 + w_2)z_1 + (w_1\phi_1 + w_2\phi_2) = \beta.$$

Key Generation Phases $\mathcal{A}$ issues token queries for vectors. $\mathcal{B}$ generates a semi-functional token for the requested vector $\vec{\sigma}_i$. $\mathcal{B}$ handles this in one of two way.

Case I: $k \notin S(\vec{\sigma}_i)$.

$\mathcal{B}$ picks random exponents $r_1,\ r_2,\ r_3,\ r_4,\ \lambda,\ \mathrm{tag}_k(\neq \Sigma_{j\in S(\vec{\sigma}_i)} A_j)$ in $\mathbb{Z}_p$. It implicitly sets

$$\tilde{r}_3 = r_3 z_1, \quad \tilde{r}_5 = r_3(\Sigma_{j\in S(\vec{\sigma}_i)} y_j z_1).$$

Note that the equation

$$(\Sigma_{j\in S(\vec{\sigma}_i)} \tilde{y}_j)\tilde{r}_3 = (\Sigma_{j\in S(\vec{\sigma}_i)} y_j z_1) r_3 z_1 = z_1 \cdot r_3(\Sigma_{j\in S(\vec{\sigma}_i)} y_j z_1) = \gamma\tilde{r}_5$$

is satisfied. $\mathcal{B}$ generates the semi-functional token as follows:

$$K_1 = g^{\delta_1 r_1} g^{\phi_1 r_2 - w_2\lambda} = g^{\delta_1 r_1} g^{\phi_1 r_2} \cdot g^{-w_2\lambda} = g^{\delta_1 r_1} g^{\phi_1 r_2} \cdot g^{-w_2\lambda},$$

$$K_2 = g^{\delta_2 r_1} (g^{z_1})^{r_2+\lambda} g^{\phi_2 r_2 + w_1\lambda} = g^{\delta_2 r_1} g^{(z_1+\phi_2) r_2} \cdot g^{(z_1+w_1)\lambda} = g^{\delta_2 r_1} g^{\tilde{\phi}_2 r_2} \cdot g^{\tilde{w}_1\lambda},$$

$$K_3 = g^{\gamma_1} (g^{z_1})^{\delta_1 r_1/\alpha} g^{(w_1\delta_1 + w_2\delta_2) r_1/\alpha} = g_1 g^{\Omega r_1},$$

$$K_4 = g^{r_2},$$

$$K_5 = K_1^{f_1} K_2^{f_2} (g^{z_1})^{-f_3 r_2} (g^{z_1})^{\gamma_2 r_3} (g^{z_2})^{r_4} g^{\gamma_3 r_4} = K_1^{f_1} K_2^{f_2} (g^{r_2})^{-f_3 z_1} (g^{\gamma_2})^{r_3 z_1} (g^{z_2+\gamma_3})^{r_4} = K_1^{f_1} K_2^{f_2} K_4^{-\tilde{f}_3} v^{\tilde{r}_3} \varphi^{r_4},$$

$$\begin{aligned} K_6 &= (g^{z_1})^{r_3 \left[\Sigma_{j\in S(\vec{\sigma}_i)} (\mu_j + t_j\sigma_j) + \gamma_2 \mathrm{tag}_k\right]} (g^{z_2})^{r_4(-\Sigma_{j\in S(\vec{\sigma}_i)} A_j + \mathrm{tag}_k)} g^{r_4(\Sigma_{j\in S(\vec{\sigma}_i)} \zeta_j + \gamma_3 \mathrm{tag}_k)} \\ &= \left(\prod_{j\in S(\vec{\sigma}_i)} g^{\mu_j} g^{t_j\sigma_j} \cdot g^{\gamma_2 \mathrm{tag}_k}\right)^{r_3 z_1} \left(\prod_{j\in S(\vec{\sigma}_i)} (g^{z_2})^{-A_j} g^{\zeta_j} \cdot (g^{z_2} g^{\gamma_3})^{\mathrm{tag}_k}\right)^{r_4} = \left(\prod_{j\in S(\vec{\sigma}_i)} u_j h_j^{\sigma_j} \cdot v^{\mathrm{tag}_k}\right)^{\tilde{r}_3} \left(\prod_{j\in S(\vec{\sigma}_i)} \tau_j \cdot \varphi^{\mathrm{tag}_k}\right)^{r_4}, \end{aligned}$$

$$K_7 = (g^{z_1})^{r_3} = g^{\tilde{r}_3}, \quad K_8 = g^{r_4},$$

$$K_9 = (g^{z_1})^{r_3(\Sigma_{j\in S(\vec{\sigma}_i)} y_j)} = g^{r_3(\Sigma_{j\in S(\vec{\sigma}_i)} y_j z_1)} = g^{\tilde{r}_5}.$$

Note that $\mathcal{B}$ can generate a normal token[5] for vectors in Case I, even though $\mathcal{B}$ computes the semi-functional token. This is because $\mathcal{B}$ can simply set the exponent $\tilde{r}_4$ to not include the unknown terms z_1 or z_2. However, the semi-functionality of tokens is necessary for handling queries in the next case.

Case II: $k \in S(\vec{\sigma}_i)$.

For notational convenience, we define $\widehat{S} = S(\vec{\sigma}_i)/\{k\}$. $\mathcal{B}$ picks random exponents $r_1,\ r_2,\ r_3,\ r_4,\ \lambda,\ \mathrm{tag}_k(\neq \Sigma_{j\in S(\vec{\sigma}_i)} A_j)$ in $\mathbb{Z}_p$. It implicitly sets

$$\tilde{\lambda} = \lambda - \frac{r_3 y_k z_2}{f_3(\Sigma_{j\in S(\vec{\sigma}_i)} A_j - \mathrm{tag}_k)}, \quad \tilde{r}_2 = r_2 + \frac{r_3 y_k z_2}{f_3(\Sigma_{j\in S(\vec{\sigma}_i)} A_j - \mathrm{tag}_k)},$$

$$\tilde{r}_3 = r_3 z_1, \quad \tilde{r}_4 = r_4 + \frac{r_3 y_k z_1}{\Sigma_{j\in S(\vec{\sigma}_i)} A_j - \mathrm{tag}_k}, \quad \tilde{r}_5 = r_3\left(y_k z_2 + \Sigma_{j\in\widehat{S}} y_j z_1\right).$$

Note that the equation

$$(\Sigma_{j\in S(\vec{\sigma}_i)} \tilde{y}_j)\tilde{r}_3 = \left(y_k z_2 + \Sigma_{j\in\widehat{S}} y_j z_1\right) r_3 z_1 = z_1 \cdot r_3\left(y_k z_2 + \Sigma_{j\in\widehat{S}} y_j z_1\right) = \gamma\tilde{r}_5,$$

is satisfied. For notational convenience, we let $\Phi = y_k/(\Sigma_{j\in S(\vec{\sigma}_i)} A_j - \mathrm{tag}_k)$. $\mathcal{B}$ generates the semi-functional token as follows:

[5] This fact will be used in proving Claim 2.

$$K_1 = g^{\delta_1 r_1} g^{\phi_1 r_2 - w_2 \lambda} (g^{z_2})^{(r_3\phi_1 + r_3 w_2)\Phi/f_3} = g^{\delta_1 r_1} g^{\phi_1 (r_2 + r_3 \Phi z_2/f_3)} \cdot g^{-w_2(\lambda - r_3 \Phi z_2/f_3)} = g^{\delta_1 r_1} g^{\phi_1 \tilde{r}_2} \cdot g^{-w_2 \tilde{\lambda}},$$

$$K_2 = g^{\delta_2 r_1} (g^{z_1})^{r_2 + \lambda} g^{\phi_2 r_2 + w_1 \lambda} (g^{z_2})^{(r_3\phi_2 - r_3 w_1)\Phi/f_3} = g^{\delta_2 r_1} g^{(z_1 + \phi_2)(r_2 + r_3 \Phi z_2/f_3)} \cdot g^{(z_1 + w_1)(\lambda - r_3 \Phi z_2/f_3)} = g^{\delta_2 r_1} g^{\tilde{\phi}_2 \tilde{r}_2} \cdot g^{\tilde{w}_1 \tilde{\lambda}},$$

$$K_3 = g^{\gamma_1} (g^{z_1})^{\delta_1 r_1/\alpha} g^{(w_1\delta_1 + w_2\delta_2) r_1/\alpha} = g_1 g^{\Omega r_1},$$

$$K_4 = g^{r_2} (g^{z_2})^{r_3 \Phi/f_3} = g^{\tilde{r}_2},$$

$$K_5 = K_1^{f_1} K_2^{f_2} (g^{z_1})^{-f_3 r_2} (g^{z_1})^{\gamma_2 r_3} (g^{z_2})^{r_4} g^{\gamma_3 r_4} (g^{z_1})^{\gamma_3 r_3 \Phi} = K_1^{f_1} K_2^{f_2} \left(g^{r_2} (g^{z_2})^{r_3 \Phi/f_3} \right)^{-f_3 z_1} (g^{\gamma_2})^{r_3 z_1} (g^{z_2 + \gamma_3})^{r_4 + r_3 \Phi z_1} = K_1^{f_1} K_2^{f_2} K_4^{-\tilde{f}_3} v^{\tilde{r}_3} \varphi^{\tilde{r}_4},$$

$$\begin{aligned} K_6 &= (g^{z_1})^{r_3 \left[\Sigma_{j \in S(\vec{\sigma}_i)} (\mu_j + t_j \sigma_j) + \gamma_2 \mathrm{tag}_k \right]} (g^{z_2})^{r_4 (-\Sigma_{j \in S(\vec{\sigma}_i)} A_j + \mathrm{tag}_k)} g^{r_4 (\Sigma_{j \in S(\vec{\sigma}_i)} \zeta_j + \gamma_3 \mathrm{tag}_k)} (g^{z_1})^{r_3 \Phi (\Sigma_{j \in S(\vec{\sigma}_i)} \zeta_j + \gamma_3 \mathrm{tag}_k)} \\ &= \left(\prod_{j \in \widehat{S}} g^{\mu_j} g^{t_j \sigma_j} \cdot (g^{z_2})^{y_k} g^{\mu_k} g^{t_k \sigma_k} \cdot g^{\gamma_2 \mathrm{tag}_k} \right)^{r_3 z_1} \left(\prod_{j \in S(\vec{\sigma}_i)} (g^{z_2})^{-A_j} g^{\zeta_j} \cdot (g^{z_2} g^{\gamma_3})^{\mathrm{tag}_k} \right)^{r_4 + r_3 \Phi z_1} \\ &= \left(\prod_{j \in S(\vec{\sigma}_i)} u_j h_j^{\sigma_j} \cdot v^{\mathrm{tag}_k} \right)^{\tilde{r}_3} \left(\prod_{j \in S(\vec{\sigma}_i)} \tau_j \cdot \varphi^{\mathrm{tag}_k} \right)^{\tilde{r}_4}, \end{aligned}$$

$$K_7 = (g^{z_1})^{r_3} = g^{\tilde{r}_3}, \quad K_8 = g^{r_4} (g^{z_1})^{r_3 \Phi} = g^{\tilde{r}_4},$$

$$K_9 = (g^{z_2})^{r_3 y_k} (g^{z_1})^{r_3 (\Sigma_{j \in S} \widetilde{\ } y_j)} = g^{r_3 (y_k z_2 + \Sigma_{j \in S} \widetilde{\ } y_j z_1)} = g^{\tilde{r}_5}.$$

Note that $\mathcal{B}$ cannot generate a normal token for vectors in Case II, which is because $\mathcal{B}$ is forced to set the exponent $\tilde{r}_4$ to include z_1. From this starting point, $\mathcal{B}$ eventually has to use the additional terms for semi-functional token[6] in generating the component K_2.

Challenge Ciphertext $\mathcal{A}$ outputs two vectors $\vec{x}_0^*, \vec{x}_1^*$ and two messages M_0, M_1. $\mathcal{B}$ picks a random bit $b \in \{0,1\}$. $\mathcal{B}$ selects random $s_1, x \in \mathbb{Z}_p$, random $R_1, \ldots, R_{k-1} \in \mathbb{G}$, and random $R_T \in \mathbb{G}_T$, and it sets $\mathrm{tag}_{c,i}^* = A_i$ for $i = 1, \ldots, \ell$. $\mathcal{B}$ implicitly sets

$$\tilde{s}_2 = z_4, \quad \tilde{s}_3 = z_3, \quad \tilde{x} = x + f_3 z_4/\delta_1.$$

$\mathcal{B}$ computes a semi-functional ciphertext $\mathrm{CT}^{\mathrm{sf}}$ for $((r_1, \ldots, r_{k-1}, r_k \text{ or } x_{b,k}^*, x_{b,k+1}^*, \ldots, x_{b,\ell}^*), M_b)$ as follows:

$$C_1 = (g^{z_1})^{s_1} g^{w_1 s_1} (g^{z_4})^{f_1} g^{\delta_2 x} (g^{z_4})^{f_3 \delta_2/\delta_1} = g^{(z_1 + w_1)s_1} g^{f_1 z_4} \cdot g^{\delta_2 (x + f_3 z_4/\delta_1)} = W_1^{s_1} F_1^{\tilde{s}_2} \cdot g^{\delta_2 \tilde{x}},$$

$$C_2 = g^{w_2 s_1} (g^{z_4})^{f_2} g^{-\delta_1 x} (g^{z_4})^{-\delta_1 f_3/\delta_1} = g^{w_2 s_1} g^{f_2 z_4} g^{-\delta_1 (x + f_3 z_4/\delta_1)} = W_2^{s_1} F_2^{\tilde{s}_2} \cdot g^{-\delta_1 \tilde{x}},$$

$$C_3 = g^{\alpha s_1} = g_2^{s_1},$$

$$\begin{aligned} C_4 &= (g^{z_1})^{(\phi_1 + w_2)s_1} g^{(w_1\phi_1 + w_2\phi_2)s_1} (g^{z_1})^{-\delta_1 x} g^{(\delta_2\phi_1 - \delta_1\phi_2)x} (g^{z_4})^{(\delta_2\phi_1 - \delta_1\phi_2) f_3/\delta_1} \\ &= \left((g^{z_1})^{\phi_1 + w_2} g^{w_1\phi_1 + w_2\phi_2} \right)^{s_1} (g^{f_3 z_1})^{z_4} \cdot g^{[\delta_2\phi_1 - \delta_1 (z_1 + \phi_2)](x + f_3 z_4/\delta_1)} = g_3^{s_1} F_3^{\tilde{s}_2} \cdot g^{(\delta_2\phi_1 - \delta_1\tilde{\phi}_2)\tilde{x}}, \end{aligned}$$

$$C_5 = g^{z_4} = g^{\tilde{s}_2},$$

$$C_{6,i} = R_i \quad (i = 1, \ldots, k-1), \quad C_{6,k} = Z^{y_k} \cdot (g^{z_4})^{\mu_k + t_k x_{b,k}^* + \gamma_2 \ \mathrm{tag}_{c,k}^*},$$

$$C_{6,i} = (g^{z_4})^{\mu_i + t_i x_{b,i}^* + \gamma_2 \mathrm{tag}_{c,i}^*} (g^{z_1 z_3})^{y_i} = (g^{\mu_i + t_i x_{b,i}^* + \gamma_2 \mathrm{tag}_{c,i}^*})^{z_4} (g^{y_i z_1})^{z_3} = (u_i h_i^{x_{b,i}^*} v^{\mathrm{tag}_{c,i}^*})^{\tilde{s}_2} Y_i^{\tilde{s}_3} \quad (i = k+1, \ldots, \ell),$$

$$C_{7,i} = (g^{z_4})^{(\zeta_i + \gamma_3 \mathrm{tag}_{c,i}^*)} = \left((g^{z_2})^{-A_i} g^{\zeta_i} (g^{z_2} g^{\gamma_3})^{\mathrm{tag}_{c,i}^*} \right)^{z_4} = (\tau_i \varphi^{\mathrm{tag}_{c,i}^*})^{\tilde{s}_2} \quad (i = 1, \ldots, \ell),$$

$$C_8 = g^{z_1 z_3} = g_4^{\tilde{s}_3}, \quad C_9 = R_T.$$

[6] This fact will be used in proving Claim 2.

$\mathcal{B}$ cannot compute a normal ciphertext as a challenge, because it is required to use the additional term $g^{(\delta_2\phi_1-\delta_1\tilde{\phi}_2)\vec{x}}$ for semi-functional ciphertext in order to remove $g^{z_1z_4}$ derived from $F_3^{\tilde{s}_2}=(g^{z_1z_4})^{f_3}$. At first glance, the term $g^{z_1z_4}$ can also be canceled out by making the exponent s_1 include z_4, but in that case, the component $C_1=W^{s_1}F_1^{s_2}=g^{(z_1+w_1)s_1}g^{f_2s_2}$ has the term $g^{z_1z_4}$. As a result, $\mathcal{B}$ has to generate the challenge ciphertext in a semi-functional form. Notice that $C_{6,i}$ for $i=1,\ldots,k-1$ can also be generated in the right form[7] as in $C_{6,i}$ for $i=k+1,\ldots,\ell$, but by the hybrid argument these elements are replaced with random group elements in $\mathbb{G}$. In case of C_9, $\mathcal{B}$ can generate the element in the right form[8], but it can also be replaced with a random element in $\mathbb{G}_T$.

If $Z=g^{z_2(z_3+z_4)}$, then $C_{6,k}$ is computed as follows:

$$C_{6,k}=(g^{z_2(z_3+z_4)})^{y_k}\cdot(g^{z_4})^{\mu_k+t_kx^*_{b,k}+\gamma_2\mathrm{tag}^*_{c,k}}=\left(g^{y_kz_2+\mu_k}g^{t_kx^*_{b,k}}g^{\gamma_2\mathrm{tag}^*_{c,k}}\right)^{z_4}(g^{y_kz_2})^{z_3}=(u_kh_k^{x^*_{b,k}}v^{\mathrm{tag}^*_{c,k}})^{\tilde{s}_2}Y_k^{\tilde{s}_3}.$$

In this case, $\mathcal{B}$ generates the semi-functional ciphertext for the vector $\left(r_1,\ldots,r_{k-1},x^*_{b,k},\ldots,x^*_{b,\ell}\right)$, so $\mathcal{B}$ plays $\mathsf{Game}_{Final^{\vec{x},k-1}}$ with $\mathcal{A}$. On the other hand, if $Z=g^{z_2(z_3+z_4)}g^{\pi}$ for some (non-zero) $\pi\in\mathbb{Z}_p$, then $\mathcal{B}$ has that

$$C_{6,k}=(g^{z_2(z_3+z_4)}g^{\pi})^{y_k}\cdot(g^{z_4})^{\mu_k+t_kx^*_{b,k}+\gamma_2\mathrm{tag}^*_{c,k}}=(u_kh_k^{x^*_{b,k}}v^{\mathrm{tag}^*_{c,k}})^{\tilde{s}_2}Y_k^{\tilde{s}_3}\cdot g^{\pi y_k}.$$

In this case, $\mathcal{B}$ generates the semi-functional ciphertext for the vector $(r_1,\ldots,r_{k-1},x^*_{b,k}+r'_k,x^*_{b,k+1},\ldots,x^*_{b,\ell})$ where r'_k is a discrete logarithm satisfying $g^{\pi y_k}=g^{t_kz_4r'_k}=(h_k^{r'_k})^{\tilde{s}_2}$. Since π is uniformly distributed at random, so is $x^*_{b,k}+r'_k$. This means that $\mathcal{B}$ plays $\mathsf{Game}_{Final^{\vec{x},k}}$ with $\mathcal{A}$.

Guess $\mathcal{B}$ receives a bit $b'\in\{0,1\}$ and outputs 0 if $b'=b$.

Analysis As mentioned above, if $Z=g^{z_2(z_3+z_4)}\mathcal{B}$ is in $\mathsf{Game}_{Final^{\vec{x},k-1}}$, whereas if $Z=g^{z_2(z_3+z_4)}g^{\pi}\mathcal{B}$ is in $\mathsf{Game}_{Final^{\vec{x},k}}$. It follows that under the Decision Linear assumption, these two games are indistinguishable. □

By combining the results of Lemmas 1–4, we obtain the following security result:

Theorem 1 (*Case 1*). *Assume the DLIN and DBDH assumptions hold in* $\mathbb{G}$. *Then, our HVE scheme is (attribute-hiding) secure in Case 1.*

Case 2: (Proof idea) We give a key idea behind the security proof in Case 2. When given the challenge ciphertext, the adversary aims to decide which one of the two pairs $(\vec{x}^*_0,M)$ and $(\vec{x}^*_1,M)$ is associated with the challenge ciphertext. As in Case 1, the basic step is to change the challenge ciphertexts into semi-functional one, but the difference is that tokens are changed from normal to semi-functional ones *or vice versa* during the security game. The main obstacle comes from the fact that the adversary can query matching tokens for both the challenge vectors $\vec{x}^*_0$ and $\vec{x}^*_1$ simultaneously. This means that the adversary can use the matching tokens to decrypt the challenge ciphertext that would be an encryption of $\vec{x}^*_b$ for a random bit $b\in\{0,1\}$. However, under the fair rule of the security game, we know that the matching tokens are associated with the vector components such that $x^*_{0,i}=x^*_{1,i}$ for $i\in\{1,\ldots,\ell\}$. Also, for the other pairwise-distinct vector components, tokens cannot be matching for $\vec{x}^*_0$ and $\vec{x}^*_1$ simultaneously, and thus they are not helpful to the adversary when trying to decrypt the challenge ciphertext. This observation shows that when tokens are generated with at least one pairwise-distinct component, we can generate the resulting tokens in a semi-functional form.

At each position $k\in\{1,\ldots,\ell\}$, we consider two cases depending on $x^*_{0,k}=x^*_{1,k}$ or not. In any case, all tokens including any kth component can be semi-functional and the other tokens not including any kth component are normal until the challenge phase. In the case where $x^*_{0,k}\neq x^*_{1,k}$, there should be no matching token including kth component as we observed above. Also, even if the challenge ciphertext is semi-functional, other matching tokens not involving kth component (if possible) will decrypt the challenge ciphertext correctly. Thus, we can change the k-th component $x^*_{b,k}$ in the challenge ciphertext into a random one by the similar simulation to the one of Case 1, and then we move onto the next position $k+1$. In the other case where $x^*_{0,k}=x^*_{1,k}$, we will perform the same process in generating tokens until the challenge phase. However, when the adversary outputs $(\vec{x}^*_0,M)$ and $(\vec{x}^*_1,M)$ where $x^*_{0,k}=x^*_{1,k}$, then we cannot construct the semi-functional challenge ciphertext since the adversary can be *already given semi-functional tokens* associated with the k-th component $x^*_{0,k}=x^*_{1,k}$. Fortunately, in that case, we can move onto the next position $k+1$ without generating the challenge ciphertext at position k, which is because the challenge ciphertext at position k has exactly the same distribution as that in the previous position $k-1$ (or further previous position). Before moving onto the next position $k+1$, we return all semi-functional tokens into normal ones to prepare for generating tokens in the next position. In this way, we can proceed to the last vector component.

The simulator considers a sequence of hybrid games as follows:

$\underline{\mathsf{Game}^2_{Real}}$: This is the actual HVE security game in Case 2. All tokens will be normal and the challenge ciphertext will be a normal challenge ciphertext on a pair $(\vec{x}^*_b,M)$, where $b\in\{0,1\}$ is a random bit and $M=M_0=M_1$.

$\underline{\mathsf{Game}^2_0}$: All tokens will be normal, but the challenge ciphertext will be a semi-functional ciphertext on a pair $(\vec{x}^*_b,M)$.

[7] This fact will be used in proving Claim 2.

[8] This fact will be used in proving Claim 2.

$\underline{\mathsf{Game}^2_1}$: All tokens will be normal, and the challenge ciphertext will be a semi-functional ciphertext on a pair $\left(\left(r_1, x^*_{b,2}, \ldots, x^*_{b,\ell}\right), M\right)$, where r_1 is a random element from Σ if $x^*_{0,1} \neq x^*_{1,1}$ and otherwise $r_1 = x^*_{b,1}$.

$\vdots \qquad \vdots$

$\underline{\mathsf{Game}^2_\ell}$: All tokens will be normal, and the challenge ciphertext will be a semi-functional ciphertext on a pair $((r_1, r_2, \ldots, r_\ell), M)$, where for $i = 1, \ldots, \ell$, r_i is a random element from Σ if $x^*_{0,i} \neq x^*_{1,i}$ and otherwise $r_i = x^*_{b,i}$.

In Game^2_{Real}, the normal challenge ciphertext on a pair $(\vec{x}^*_b, M)$ is given to the adversary. On the other hand, in Game^2_ℓ, the challenge ciphertext given to the adversary is a semi-functional ciphertext corresponding to $((r_1, \ldots, r_\ell), M)$ that leaks no information about $\vec{x}^*_b$. Note that r_i is random for each i such that $x^*_{0,i} \neq x^*_{1,i}$, so that the final vector $(r_1, r_2, \ldots, r_\ell)$ does not help the adversary to tell between $\vec{x}^*_0$ and $\vec{x}^*_1$. We will show that no polynomial time adversary is able to distinguish between Game^2_{Real} and Game^2_ℓ by proving that the transitions between the sequence of games above are all computationally indistinguishable under the DLIN assumption.

Lemma 5. *Suppose that the DLIN assumption holds. Then no polynomial time adversary $\mathcal{A}$ can distinguish between Game^2_{Real} and Game^2_0 with non-negligible advantage.*

Proof. The proof of Lemma 5 is identical to that of Lemma 1. □

Next, in order to prove that distinguishing between two games Game^2_{k-1} and Game^2_k for $k = 1, \ldots, \ell$ is computationally hard, we consider two cases (for each k) depending on whether $x^*_{0,k} \neq x^*_{1,k}$ or $x^*_{0,k} = x^*_{1,k}$ when $\mathcal{A}$ outputs two challenge vectors $\vec{x}^*_0 = \left(x^*_{0,1}, \ldots, x^*_{0,\ell}\right)$ and $\vec{x}^*_1 = \left(x^*_{1,1}, \ldots, x^*_{1,\ell}\right)$. In the first case, $\mathcal{A}$ should not issue token queries for $\vec{\sigma}_i = (\sigma_{i,1}, \ldots, \sigma_{i,k}, \ldots, \sigma_{i,\ell})$ such that $k \in S(\vec{\sigma}_i)$ and $P_\ell(\vec{\sigma}_i, \vec{x}^*_0) = P_\ell(\vec{\sigma}_i, \vec{x}^*_1) = 1$. This means that tokens including k-th component $\sigma_{i,k}$ as non-wildcard should not be matching queries on either $\vec{x}^*_0$ or $\vec{x}^*_1$. Thus, the tokens including $\sigma_{i,k}$ do not help $\mathcal{A}$ decrypt the challenge ciphertext correctly. The impossibility of decryption allows the simulator to change the tokens from normal to semi-functional even if the challenge ciphertext will also be semi-functional. However, other tokens remain normal so that $\mathcal{A}$ is able to use the other normal tokens to decrypt the challenge ciphertext successfully. Once the tokens including the k-th component $\sigma_{i,k}$ and the challenge ciphertext will be semi-functional, the k-th component $x^*_{b,k}$ for a random bit $b \in \{0,1\}$ is replaced with a random element. This is performed in a game defined (below) as $\mathsf{Game}^2_{k-1,F}$, and then the tokens return back to the normal type necessary for the next intermediate game Game^2_k. On the other hand, the second case when $x^*_{0,k} = x^*_{1,k}$ makes two intermediate games equal, i.e., $\mathsf{Game}^2_{k-1} = \mathsf{Game}^2_k$. Thus, we can naturally move onto the next intermediate games in Case 2.

Now, it remains to show how the simulator behaves in the case where $x^*_{0,k} \neq x^*_{1,k}$. Let q_k be the number of token queries for vectors $\{\vec{\sigma}_i\}_{i=1}^{q_k}$ such that $k \in S(\vec{\sigma}_i)$. Then the simulator considers a sequence of hybrid games as follows:

$\underline{\mathsf{Game}^2_{k-1}}$: All tokens will be normal, and the challenge ciphertext will be a semi-functional ciphertext on a pair $\left(\left(r_1, \ldots, r_{k-1}, x^*_{b,k}, \ldots, x^*_{b,\ell}\right), M\right)$, where the elements $\{r_i\}$ for $i = 1, \ldots, k-1$ are random from Σ if $x^*_{0,i} \neq x^*_{1,i}$ and otherwise $r_i = x^*_{b,i}$.

$\vdots \qquad \vdots$

$\underline{\mathsf{Game}^2_{k-1,j}}$: The first j tokens in $\{\vec{\sigma}_i\}_{i=1}^{q_k}$ will be semi-functional and the other tokens will be normal, and the challenge ciphertext will be a semi-functional ciphertext on a pair $\left(\left(r_1, \ldots, r_{k-1}, x^*_{b,k}, \ldots, x^*_{b,\ell}\right), M\right)$.

$\vdots \qquad \vdots$

$\underline{\mathsf{Game}^2_{k-1,q_k}}$: All tokens in $\{\vec{\sigma}_i\}_{i=1}^{q_k}$ will be semi-functional and the other tokens will be normal, and the challenge ciphertext will be a semi-functional ciphertext on a pair $\left(\left(r_1, \ldots, r_{k-1}, x^*_{b,k}, \ldots, x^*_{b,\ell}\right), M\right)$.

$\underline{\mathsf{Game}^2_{k-1,F}}$: All tokens in $\{\vec{\sigma}_i\}_{i=1}^{q_k}$ will be semi-functional and the other tokens will be normal, and the challenge ciphertext will be a semi-functional ciphertext on a pair $\left(\left(r_1, \ldots, r_{k-1}, r_k, x^*_{b,k+1}, \ldots, x^*_{b,\ell}\right), M\right)$, where r_k is random.

$\underline{\mathsf{Game}^2_{k-1,\widehat{q_k-1}}}$: The first $q_k - 1$ tokens in $\{\vec{\sigma}_i\}_{i=1}^{q_k}$ will be semi-functional and the other tokens will be normal, and the challenge ciphertext will be a semi-functional ciphertext on a pair $\left(\left(r_1, \ldots, r_k, x^*_{b,k+1}, \ldots, x^*_{b,\ell}\right), M\right)$.

$\vdots \qquad \vdots$

$\underline{\mathsf{Game}^2_{k-1,\widehat{j}}}$: The first j tokens in $\{\vec{\sigma}_i\}_{i=1}^{q_k}$ will be semi-functional and the other tokens will be normal, and the challenge ciphertext will be a semi-functional ciphertext on a pair $\left(\left(r_1,\ldots,r_k,x^*_{b,k+1},\ldots,x^*_{b,\ell}\right),M\right)$.

$\vdots$ $\vdots$

$\underline{\mathsf{Game}^2_k}$: All tokens will be normal, and the challenge ciphertext will be a semi-functional ciphertext on a pair $\left(\left(r_1,\ldots,r_k,x^*_{b,k+1},\ldots,x^*_{b,\ell}\right),M\right)$.

The following claims show that all these hybrid games are indistinguishable under the DLIN assumption, so that distinguishing between Game^2_{k-1} and Game^2_k is computationally infeasible.

Let $j = 1,\ldots,q_k$ and $\mathsf{Game}^2_{k-1} = \mathsf{Game}^2_{k-1,0}$.

Claim 1. *Suppose that the DLIN assumption holds. Then no polynomial time adversary $\mathcal{A}$ can distinguish between $\mathsf{Game}^2_{k-1,j-1}$ and $\mathsf{Game}^2_{k-1,j}$ with non-negligible advantage.*

Proof. The proof of Claim 1 is almost identical to that of Lemma 2. The difference is that the first $j-1$ tokens in $\{\vec{\sigma}_i\}_{i=1}^{q_k}$ are generated as semi-functional ones, and the j-th token is generated using the target element Z of a DLIN problem, and the other tokens are generated as normal ones. □

Claim 2. *Suppose that the DLIN assumption holds. Then no polynomial time adversary $\mathcal{A}$ can distinguish between $\mathsf{Game}^2_{k-1,q_k}$ and $\mathsf{Game}_{k-1,F}$ with non-negligible advantage.*

Proof. The proof of Claim 2 is almost identical to that of Lemma 4. The difference is that all tokens in $\{\vec{\sigma}_i\}_{i=1}^{q_k}$ are generated as semi-functional ones and the other tokens are generated as normal ones. In constructing the challenge ciphertext, the elements $\{C_{6,i}\}_{i=1}^{k-1}$ are generated in the right form if $x^*_{0,i} = x^*_{1,i}$ for $i = 1,\ldots,k-1$ and C_9 is also generated in the right form. □

Let $j = q_k,\ldots,1$, $\mathsf{Game}^2_{k-1,F} = \mathsf{Game}^2_{k-1,\widehat{q_k}}$, and $\mathsf{Game}^2_k = \mathsf{Game}^2_{k-1,\widehat{0}}$.

Claim 3. *Suppose that the DLIN assumption holds. Then no polynomial time adversary $\mathcal{A}$ can distinguish between $\mathsf{Game}^2_{k-1,\widehat{j}}$ and $\mathsf{Game}^2_{k-1,\widehat{j-1}}$ with non-negligible advantage.*

Proof. The proof of Claim 3 is identical to that of Claim 1 (Lemma 2), except that the k-th component $x^*_{b,k}$ is replaced with a random element in generating the challenge ciphertext. □

By putting the results of claims all together, we obtain the security result of Lemma 6. Let $k = 1,\ldots,\ell$.

Lemma 6. *Suppose that the DLIN assumption holds. Then no polynomial time adversary $\mathcal{A}$ can distinguish between Game^2_{k-1} and Game^2_k with non-negligible advantage.*

By combining the results of Lemmas 5 and 6, we obtain the following security result of Case 2:

Theorem 2. *(Case 2) Assume the DLIN assumption holds in $\mathbb{G}$. Then, our HVE scheme is (attribute-hiding) secure in Case 2.*

5. Comparison to other HVE schemes

Table 1 gives a comparison of the different features in previous HVE schemes and ours when encrypting ℓ-dimensional vectors. Any IPE scheme can be straightforwardly transformed into an HVE scheme by expanding the dimension of vectors from ℓ to 2ℓ [28], so we consider the previous IPE schemes [28,36,30,38,35,34] handling 2ℓ-dimensional vectors as HVE schemes handling ℓ-dimensional vectors. We point out that [34] is an independent work that has recently suggested a fully secure IPE scheme. In terms of achieving full security, [30,18,35] are weakly attribute-hiding in the sense that an adversary can make token queries such that $P_\ell(\vec{x}^*_0,\vec{\sigma}_i) = P_\ell(\vec{x}^*_1,\vec{\sigma}_i) = 0$ for all queried vectors $\{\vec{\sigma}_i\}$, where $\vec{x}^*_0$ and $\vec{x}^*_1$ are vectors challenged by the adversary. [18] suggested another HVE scheme which is secure in the opposite sense of security modelling where $P_\ell(\vec{x}^*_0,\vec{\sigma}_i) = P_\ell(\vec{x}^*_1,\vec{\sigma}_i) = 1$ for all queried vectors $\{\vec{\sigma}_i\}$. In any case, our HVE scheme (as well as the independent work [34]) is the first one that is fully secure in the security model where both matching and non-matching token queries are validly considered in a single security game.

Regarding efficiency, the schemes in [40,29,34] and ours have the property that both token size and the number of pairing computations (necessary for decryption) do not depend on the dimension ℓ of attribute vectors. As mentioned before, these schemes are desirable in applications where ℓ increases to deal with more expressive access control. Table 1 simply shows the comparison of ℓ conjunctive equality predicates, but when we consider access control along with conjunctive

combination of comparison and subset predicates, the efficiency impact is stronger. For instance, if a subset predicate is defined over a set of n elements, one subset predicate leads to a token of size $O(n)$ and pairing computations of size $O(n)$. Thus, if an access control is a conjunctive combination that consists of ℓ_1 equality, ℓ_2 comparison, and ℓ_3 subset predicates, the token size and pairing computations for such an access control increase with $O(\ell_1 + \ell_2 + n\ell_3)$ in other HVE schemes. However, in [40,29,34] and ours, these two factors remain $O(1)$ regardless of the numbers of conjunctions.

6. Conclusion

We presented the first HVE scheme that is fully secure under the DBDH and DLIN assumptions. Our HVE scheme required $O(1)$-sized private keys and $O(1)$ paring computations for decryption, regardless of the dimension of vectors. These advantages are attractive to the query server as the dimension increases to support more expressed access control. This was achieved by first constructing a novel type of (tag-based) dual system encryption. New techniques were then applied to both conceal vector components from ciphertexts and compress tag values into one. Our HVE scheme also yielded an anonymous IBE scheme that is fully secure under the standard assumptions.

It was difficult to extend our HVE (and anonymous IBE) scheme to support a hierarchical delegation mechanism. Thus, it is still an open problem to construct an HVE scheme supporting delegation, while preserving full security under standard assumptions. Another interesting open problem is to create an IPE scheme that is fully secure under standard assumptions in a way that both matching and non-matching key queries are allowed. It would also be interesting to show how to reduce the number of pairing computations to $O(1)$ in an IPE scheme, which has seemed difficult to achieve.

Acknowledgements

The authors thank the reviewers for their helpful comments and suggestions for this paper. Jong Hwan Park and Dong Hoon Lee were supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea government (MEST) (No. 2012-0008697). Willy Susilo was supported by ARC Future Fellowship FT0991397.

References

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, H. Shi, Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extentions, in: CRYPTO 2005, LNCS, vol. 3621, 2005, pp. 205–222.
[2] S. Agrawal, D. Boneh, X. Boyen, Efficient lattice (h)ibe in the standard model, in: EUROCRYPT 2010, LNCS, vol. 6110, 2010, pp. 553–572.
[3] N. Attrapadung, B. Libert, Functional encryption for inner product: achieving constant-size ciphertexts with adaptive security or support for negation, in: PKC 2010, LNCS, vol. 6056, 2010, pp. 384–402.
[4] N. Attrapadung, B. Libert, E.D. Panafieu, Expressive key-policy attribute-based encryption with constant-size ciphertexts, in: PKC 2011, LNCS, vol. 6571, 2011. pp. 90–108.
[5] J. Bethancourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: 28th IEEE Symposium on Security and Privacy, 2007, pp. 321-334.
[6] D. Boneh, X. Boyen, Efficient selective-ID secure identity based encryption without random oracles, in: EUROCRYPT 2004, LNCS, vol. 3027, 2004, pp. 223–238.
[7] D. Boneh, X. Boyen, E. Goh, Hierarchical identity based encryption with constant size ciphertext, in: EUROCRYPT 2005, LNCS, vol. 3493, 2005, pp. 440–456.
[8] D. Boneh, X. Boyen, H. Shacham, Short group signatures, in: CRYPTO 2004, LNCS, vol. 3152, 2004, pp. 41–55.
[9] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, in: CRYPTO 2001, LNCS, vol. 2139, 2001, pp. 213–229.
[10] D. Boneh, C. Gentry, B. Waters, Collusion resistant broadcast encryption with short ciphertexts and private keys, in: CRYPTO 2005, LNCS, vol. 3621, 2005, pp. 258–275.
[11] D. Boneh, A. Sahai, B. Waters, Functional encryption: definition and challenges, in: TCC 2011, LNCS, vol. 6597, 2011, pp. 253–273.
[12] D. Boyen, B. Waters, Anonymous hierarchical identity-based encryption (without random oracles), in: CRYPTO 2006, LNCS, vol. 4117, 2006, pp. 290–307.
[13] D. Boneh, B. Waters, Conjunctive, subset, and range queries on encrypted data, in: TCC 2007, LNCS, vol. 4392, 2007, pp. 535–554.
[14] C. Cocks, An identity based encryption scheme based on quadratic residues, in: 8th IMA International Conference on Cryptography and Coding 2001, 2001.
[15] C. Canetti, S. Halevi, J. Katz, A forward-secure public-key encryption scheme, in: EUROCRYPT 2003, LNCS, vol. 2656, 2003, pp. 255–271.
[16] D. Cash, D. Hofheinz, E. Kiltz, C. Peikert, Bonsai trees, or how to delegate a lattice basis, in: EUROCRYPT 2010, LNCS, vol. 6110, 2010, pp. 523–552.
[17] A. De Caro, V. Iovino, G. Persiano, Fully secure anonymous HIBE and secret-key anonymous IBE with short ciphertexts, in: Pairing 2010, LNCS, vol. 6487, 2010, pp. 347–366.
[18] A. De Caro, V. Iovino, G. Persiano, Efficient fully secure (hierarchical) predicate encryption for conjunctions, disjunctions and k-CNF/DNF formulae. Cryptology ePrint Archive, Report 2010/492, 2010. http://eprint.iacr.org/2010/492.
[19] A. Fiat, M. Naor, Broadcast encryption, in: CRYPTO 1993, LNCS, vol. 773, 1993, pp. 480–491.
[20] C. Gentry, Practical identity-based encryption without random oracles, in: EUROCRYPT 2006, LNCS, vol. 4004, 2006, pp. 445–464.
[21] C. Gentry, S. Halevi, Hierarchical identity based encryption with polynomially many levels, in: TCC 2009, LNCS, vol. 5444, 2009, pp. 437–456.
[22] C. Gentry, C. Peikert, V. Vaikuntanathan, Trapdoors for hard lattices and new cryptographic constructions, in: STOC 2008, ACM Press, 2008, pp. 197–206.
[23] C. Gentry, A. Silverberg, Hierarchical id-based cryptography, in: ASIACRYPT 2002, LNCS, vol. 2501, 2002, pp. 548–566.
[24] V. Goyal, A. Jain, O. Pandey, A. Sahai, Bounded ciphertext policy attribute based encryption, in: ICALP 2008 Part II, LNCS, vol. 5126, 2008, pp. 579–591.
[25] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: ACM-CCS 2006, ACM Press, 2006, pp. 89–98.
[26] J. Horwitz, B. Lynn, Toward hierarchical identity-based encryption, in: EUROCRYPT 2002, LNCS, vol. 2332, 2002, pp. 466–481.
[27] V. Iovino, G. Persiano, Hidden-vector encryption with groups of prime order, in: Pairing 2008, LNCS, vol. 5209, 2008, pp. 75–88.
[28] J. Katz, A. Sahai, B. Waters, Predicate encryption supporting disjunctions, polynomial equations, and inner products, in: EUROCRYPT 2008, LNCS, vol. 4965, 2008, pp. 146–162.
[29] K. Lee, D.H. Lee, Improved hidden vector encryption with short ciphertexts and tokens. Des. Codes Cryptogr. 58 (3) (2011) 297–319.

[30] A.B. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters, Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption, in: EUROCRYPT 2010, LNCS, vol. 6110, 2010, pp. 62–91.
[31] A. Lewko, B. Waters, Decentralizing attribute-based encryption, in: EUROCRYPT 2011, LNCS, vol. 6632, 2011, pp. 568–588.
[32] A. Lewko, B. Waters, New techniques for dual system encryption and fully secure HIBE with short ciphertexts, in: TCC 2010, LNCS, vol. 5978, 2010, pp. 455–479.
[33] A. Lewko, B. Waters, Unbounded HIBE and attribute-based encryption, in: EUROCRYPT 2011, LNCS, vol. 6632, 2011, pp. 547–567.
[34] T. Okamoto, K. Takashima, Adaptively attribute-hiding (hierarchical) inner product encryption, in: EUROCRYPT 2012, LNCS, vol. 7237, 2012, pp. 591–608.
[35] T. Okamoto, K. Takashima, Fully secure functional encryption with general relations from the decisional linear assumption, in: CRYPTO 2010, LNCS, vol. 6223, 2010, pp. 191–208.
[36] T. Okamoto, K. Takashima, Hierarchical predicate encryption for inner-products, in: ASIACRYPT 2009, LNCS, vol. 5912, 2009, pp. 214–231.
[37] R. Ostrovsky, A. Sahai, B. Waters, Attribute-based encryption with non-monotonic access structures, in: ACM-CCS 2007, ACM Press, 2007, pp. 195–203.
[38] J.H. Park, Inner-product encryption under standard assumptions, Des. Codes Cryptogr. 58 (3) (2011) 235–257.
[39] J.H. Park, H.J. Kim, M, H. Sung, D.H. Lee, Public key broadcast encryption schemes with shorter transmissions, IEEE Trans. Broadcasting, 54(3) (2008).
[40] J.H. Park, D.H. Lee, A hidden-vector encryption with constant-size tokens and pairing computations, IEICE Trans. Fundamentals E93-A (9) (2010) 1620–1631.
[41] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: EUROCRYPT 2005, LNCS, vol. 3494, 2005, pp. 457–473.
[42] J.H. Seo, T. Kobayashi, M. Oukubo, K. Suzuki, Anonymous hierarchical identity-based enryption with constant size ciphertexts, in: PKC 2009, LNCS, vol. 5443, 2009, pp. 215–234.
[43] A. Shamir, Identity-based cryptosystems and signature schemes, in: CRYPTO 1984, LNCS, vol. 196, 1984, pp. 47–53.
[44] E. Shi and B. Waters, Delegating capabilities in predicate encryption systems, in: ICALP 2008, LNCS, vol. 5126, 2008, pp. 560–578.
[45] B. Waters, Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization, in: PKC 2011, LNCS, vol. 6571, 2011, pp. 53–70.
[46] B. Waters, Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions, in: CRYPTO 2009, LNCS, vol. 5677, 2009, pp. 619–636.
[47] B. Waters, Efficient identity-based encryption without random oracles, in: EUROCRYPT 2005, LNCS, vol. 3494, 2005, pp. 114–127.