Proceedings of CECNet 2022
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA220511

Attack and Improvement of a Hidden Vector Encryption Scheme

Ke WANG [a,b], Zhikun WANG [c], Song LUO [c] and Zhi GUAN [d,1]

[a] *Key Laboratory of High Confidence Software Technologies (Peking University), MoE, Beijing, China*

[b] *Department of Computer Science and Technology, EECS, Peking University, Beijing, China*

[c] *School of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China*

[d] *National Engineering Research Center for Software Engineering, Peking University, Beijing, China*

Abstract. Hidden Vector Encryption (HVE) is a new kind of attribute-based encryption in which a vector is hidden in the ciphertext or linked with the secret key. In ESORICS 2014, Phuong et al. proposed an HVE scheme with constant-size ciphertext which is constructed in the prime order setting. In this paper, we show that Phuong et al.'s scheme is not vector-hiding due to public parameters in their scheme leak some information about vectors. Furthermore, an improved HVE scheme is proposed in the prime order setting and its security is proven in the security model. Comparison shows our scheme has more efficient in decryption than current other HVE schemes.

Keywords. Hidden vector encryption, constant-size ciphertext, prime order setting, bilinear group, security

1. Introduction

Hidden Vector Encryption (HVE) [1] is a new kind of attribute-based encryption [2,3] in which the message is encrypted to a hidden vector while a user holds a secret key linked with a vector. Wildcard can be used in either secret key or ciphertext, the former is called key policy HVE and the latter is called ciphertext policy HVE [4]. When both vectors match, the ciphertext can be decrypted. For example, in a ciphertext policy HVE scheme, two secret keys linked with $(1,2,3)$ and $(1,2,5)$ respectively can decrypt a ciphertext associated with $(1,2,*)$. Vector-hiding in HVE means the decryptor cannot know the concrete target vector except his vector matches the target vector. HVE can be used to do some operations on encrypted data such as comparison, range queries, conjunctions and subset queries, so it is very favorable in many applications requiring privacy protection such as cloud computing.

In ESORICS 2014, Phuong et al. [5] proposed two efficient ciphertext policy HVE schemes. They used composite order bilinear groups to construct the first HVE scheme.

[1]Corresponding Author: guan@pku.edu.cn

Their scheme has constant-size ciphertext and is proven selective security in the standard model. They then transformed the first scheme to get the second prime-order scheme. However, their prime-order construction is not secure.

Our Contribution. In this paper, we give an attack on Phuong et al.'s prime order HVE scheme (PYS-HVE in short) and show their scheme is not secure. We construct a special ciphertext and prove that PYS-HVE scheme doesn't have vector-hidden property by testing the ciphertext. Furthermore, we construct a new HVE scheme on the prime order bilinear groups. We also prove its selective security in the standard model. Experiment shows our scheme has better performance than current HVE schemes.

Related Works. Boneh and Waters [1] first introduced the notion of HVE and they gave a construction in composite order groups. Katz et al.'s study [6] found that inner-product encryption implies HVE so we can naturally derive fully secure HVE schemes from fully secure inner-product encryption schemes [7]. Hattori et al. [4] proposed the first ciphertext policy HVE scheme which was based on the anonymous HIBE [8] and the wildcarded IBE [9]. The ciphertext size in Hattori et al.'s CP-HVE scheme is linear to vector length and Phuong et al. [5] proposed the first HVE scheme with constant-size ciphertext. Liao et al. [10] presented a ciphertext policy HVE scheme supporting multiuser keyword search. Lee [11] presented a conversion method which can transform composite-order setting HVE schemes into prime-order setting schemes. Bartusek et al. [12] proposed a new function-private predicate encryption scheme in the public key setting which supports point functions, conjunctions, d-disjunctions with read-once conjunctions and d-CNFs with a constant d. Recently, HVE is extended to ABE with hidden policy. Murad et al. [13] proposed a new kind of CP-ABE with in which access structures for AND or OR gates with wildcards are partially hidden. In fact, an access structure using partially hidden AND-gates with wildcards equals to a hidden vector.

Organization. The rest of this paper is organized as follows. We provide some necessary background knowledge in Section 2. We analyze the PYS-HVE scheme in Section 3 and propose our improved construction with security proof in Section 4 respectively. Next a brief comparison is given in Section 5. Finally the paper is concluded with future work in Section 6.

2. Preliminaries

Definition 2.1. *Let p be a prime and $\mathbb{G}$, $\mathbb{G}_T$ be two multiplicative groups of order p. Let g be a generator of $\mathbb{G}$. $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map which satisfies the following two properties:*

(i) Bilinearity: $\forall x, y \in \mathbb{Z}_p$, $e(g^x, g^y) = e(g,g)^{xy}$.

(ii) Non-degeneracy: $e(g,g) \neq 1$.

We call $\mathbb{G}$ a bilinear group if the group operation in $\mathbb{G}$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ can be efficiently computed.

Definition 2.2. *Let g be a random generator of $\mathbb{G}$. Let h and Z are chosen randomly from $\mathbb{G}$ and $\mathbb{G}_T$ respectively. Let $\overrightarrow{g}_{g,\alpha,d}$ be $g_1, \cdots, g_d, g_{d+2}, \cdots, g_{2d} \in \mathbb{G}^{2d-1}$ where $g_i = g^{\alpha^i}$ and $\alpha \in \mathbb{Z}_p^*$ is unknown.*

We define the advantage for an algorithm $\mathscr{A}$ to break the decision d-BDHE assumption as

$$\left|\Pr[\mathscr{A}(g,h,\overrightarrow{g}_{g,\alpha,d},e(g_{d+1},h))=1]-\Pr[\mathscr{A}(g,h,\overrightarrow{g}_{g,\alpha,d},Z)=1]\right|.$$

If no probabilistic polynomial-time algorithm has non-negligible advantage to break the decision d-BDHE assumption, we say the decision d-BDHE assumption holds.

An HVE scheme consists of the following four algorithms: **Setup** algorithm for system setup , **Key Generation** algorithm for secret key generation, **Encrypt** algorithm for message encryption, and **Decrypt** algorithm for ciphertext decryption. The security model used for our HVE is called selective security model with six stages: **Init**, **Setup**, **Query Phase 1**, **Challenge**, **Query Phase 2** and **Guess**. The adversary should submit two challenging vectors at the **Init** stage and all queried identities in **Query Phase 1, 2** cannot match these two challenging vectors.

3. Attack on PYS-HVE Scheme

We first review the public parameters and ciphertext of PYS-HVE scheme. Suppose the maximum number of wildcards that are allowed in an encryption vector be N and the vector length is L. The public parameters include $L+1$ random elements $V,H_1,\cdots,H_L\in G$, three random generators $g,f,w\in\mathbb{G}$, a paring $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$ and $Y=e(g,w)$.

Let $\overrightarrow{v}=(v_1,\cdots,v_L)\in\sum_L^*$ be a vector with $\tau\le N$ wildcards. To encrypt a message M with $\overrightarrow{v}$, the **Encrypt** algorithm chooses a random $s\in\mathbb{Z}_p$ and sets

$$C_0=MY^s,C_1=g^{\frac{s}{t}},C_2=f^s,C_3=\prod_{i=1,i\notin J}^{L}(H_i^{v_i}V)^{\frac{\prod_{j\in J}(i-j)s}{t}} \tag{1}$$

where $J=\{j_1,j_2,\cdots,j_\tau\}$ is the set containing the indexes of wildcards in $\overrightarrow{v}$ and $t=(-1)^\tau j_1j_2\cdots j_\tau$. The ciphertext is $CT=(C_0,C_1,C_2,C_3,J)$.

In PYS-HVE scheme, the elements linked with vectors, i.e., $V,H_1,\cdots,H_L\in_R G$ are both used in encryption and decryption. This allows us to create elements similar to secret key. These elements cannot be used to decrypt but can be used to test the target vector. In fact, given the public parameters and a ciphertext, we can easily check whether a vector $\overrightarrow{z}=(z_1,\cdots,z_L)$ is used to encrypt the message. We first construct $K=\prod_{i=1,i\notin J}^{L}(H_i^{z_i}V)^{\prod_{j\in J}(i-j)}$ and check whether the equation

$$e(C_1,K)=e(C_3,g) \tag{2}$$

holds. If the equation holds, we can conclude the encryption vector is $\overrightarrow{z}$. Hence, the vector-hiding property in PYS-HVE scheme is broken.

4. Our Improved Scheme

4.1. Description

- **Setup**$(1^k, \sum, L)$: Assume that at most $N (N \ll L)$ wildcards are allowed in a vector for encryption. Then the algorithm generates a paring $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, randomly chooses $L+1$ elements $V, H_1, \cdots, H_L \in G$, two generators $g, w \in G$ and four integers $t_1, t_2, t_3, t_4 \in \mathbb{Z}_p$. Then it sets $U_1 = g^{t_1}$, $U_2 = g^{t_2}$, $U_3 = g^{t_3}$, $U_4 = g^{t_4}$ and $Y = e(g,w)^{t_1 t_2}$. The algorithm sets the public key $\mathrm{PK} = (\mathrm{PP}, V, (H_1, \cdots, H_L), U_1, U_2, U_3, U_4, Y)$ and the master secret key $\mathrm{MSK} = (w, t_1, t_2, t_3, t_4)$ where $\mathrm{PP} = \{g, p, \mathbb{G}, \mathbb{G}_T, e\}$.
- **Encrypt**$(\mathrm{PK}, M, \vec{v} = (v_1, ..., v_L) \in \sum_L^*)$: Assume that $\vec{v} = (v_1, ..., v_L)$ contains $\tau \leq N$ wildcards and $W = \{j_1, j_2, \cdots, j_\tau\}$ is the set of the positions of wildcards in $\vec{v}$. The algorithm randomly chooses three integers $s, s_1, s_2 \in \mathbb{Z}_p$. It then computes $C_0 = M \cdot Y^s, C_1 = \prod_{i=1, i \notin W}^{L} (H_i^{v_i} V)^{\prod_{j \in J}(i-j)s}, C_2 = U_1^{s-s_1}, C_3 = U_2^{s_1}, C_4 = U_3^{s-s_2}, C_5 = U_4^{s_2}$. The ciphertext CT is set as $(C_0, C_1, C_2, C_3, C_4, C_5, J)$.
- **Key Generation**$(\mathrm{MSK}, \vec{z} = (z_1, \cdots, z_L) \in \sum_L)$: Given a vector $\vec{z} = (z_1, ..., z_L)$ for key generation, the algorithm randomly chooses $r_1, r_2 \in \mathbb{Z}_p$, then it computes $K_1 = g^{r_1 t_1 t_2 + r_2 t_3 t_4}$,

$$\begin{pmatrix} K_{2,0} = w^{t_2} \prod_{i=1}^{L} (H_i^{z_i} V)^{r_1 t_2} \\ K_{2,1} = \prod_{i=1}^{L} (H_i^{z_i} V)^{i r_1 t_2} \\ \cdots \\ K_{2,N} = \prod_{i=1}^{L} (H_i^{z_i} V)^{i^N r_1 t_2} \end{pmatrix}, \begin{pmatrix} K_{3,0} = w^{t_1} \prod_{i=1}^{L} (H_i^{z_i} V)^{r_1 t_1} \\ K_{3,1} = \prod_{i=1}^{L} (H_i^{z_i} V)^{i r_1 t_1} \\ \cdots \\ K_{3,N} = \prod_{i=1}^{L} (H_i^{z_i} V)^{i^N r_1 t_1} \end{pmatrix},$$

$$\begin{pmatrix} K_{4,0} = \prod_{i=1}^{L} (H_i^{z_i} V)^{r_2 t_4} \\ K_{4,1} = \prod_{i=1}^{L} (H_i^{z_i} V)^{i r_2 t_4} \\ \cdots \\ K_{4,N} = \prod_{i=1}^{L} (H_i^{z_i} V)^{i^N r_2 t_4} \end{pmatrix}, \begin{pmatrix} K_{5,0} = \prod_{i=1}^{L} (H_i^{z_i} V)^{r_2 t_3} \\ K_{5,1} = \prod_{i=1}^{L} (H_i^{z_i} V)^{i r_2 t_3} \\ \cdots \\ K_{5,N} = \prod_{i=1}^{L} (H_i^{z_i} V)^{i^N r_2 t_3} \end{pmatrix}$$

  The corresponding key is $\mathrm{SK} = (K_1, K_{2,t}, K_{3,t}, K_{4,t}, K_{5,t}, t \in \{0, \cdots, N\})$.
- **Decrypt**$(\mathrm{CT}, \mathrm{SK})$: Suppose that CT is encrypted to $\vec{v}$ and SK is associated with $\vec{z}$ respectively. If $v_i = z_i$ for $i \in \{1, \cdots, L\} \setminus J$, the decryption algorithm decrypts the ciphertext as follows. It first applies the Viete formulas on $J = \{j_1, \cdots, j_\tau\}$ and computes $a_{\tau-k} = (-1)^k \sum_{i \leq i_1 < i_2 < \ldots < i_k \leq \tau} j_{i_1} j_{i_2} \cdots j_{i_k}$, for $0 \leq k \leq \tau$. Next it computes

$$K_2 = \prod_{t=0}^{\tau} K_{2,t}^{a_t},\ K_3 = \prod_{t=0}^{\tau} K_{3,t}^{a_t},\ K_4 = \prod_{t=0}^{\tau} K_{4,t}^{a_t},\ K_5 = \prod_{t=0}^{\tau} K_{5,t}^{a_t},$$

and then outputs

$$M = \left(\frac{e(C_1, K_1)}{e(C_2, K_2)e(C_3, K_3)e(C_4, K_4)e(C_5, K_5)} \right)^{a_0^{-1}} \cdot C_0 \tag{3}$$

4.2. Security

Theorem 4.1. *Assume the decision L-BDHE assumption hold in $\mathbb{G}$, then our improved scheme is secure.*

We prove Theorem 4.1 through a series of experiments similar to that of [14]. We define the following games based on the security model with different challenge ciphertexts:

- G_1: The challenge ciphertext is normal, i.e., $\mathrm{CT} = (C_0, C_1, C_2, C_3, C_4, C_5)$.
- G_2: This game is similar to G_1 but C_0 is replaced with a random element Z in $\mathbb{G}_T$, i.e., $\mathrm{CT} = (Z, C_1, C_2, C_3, C_4, C_5)$
- G_3: This game is similar to G_2 but C_2 is replaced with a random element Z_1 in $\mathbb{G}$, i.e., $\mathrm{CT} = (Z, C_1, R_1, C_3, C_4, C_5)$
- G_4: This game is similar to G_3 but C_4 is replaced with a random element Z_2 in $\mathbb{G}$, i.e., $\mathrm{CT} = (Z, C_1, Z_1, C_3, Z_2, C_5)$

In G_4, the elements of the challenge ciphertext are all random, so it will leak no information about the message or the vector. Therefore, if these four games are indistinguishable, the security of our HVE scheme is proven.

Lemma 4.1. *Under the decision L-BDHE assumption,* G_1 *and* G_2 *are indistinguishable.*

Proof. Suppose that the advantage of the adversary $\mathscr{A}$ for distinguishing between G_1 and G_2 is ε which is non-negligible. Then the decision L-BDHE problem can solved by an algorithm $\mathscr{B}$ based on $\mathscr{A}$. Given an L-BDHE challenge $(g, \overrightarrow{y}_{g,\alpha,L} = (g_1, g_2, \cdots, g_L, g_{L+2}, \cdots, g_{2L}), h, Z)$, where $g_i = g^{\alpha^i}$ and $\alpha \in \mathbb{Z}_p^*$ is unknown. $\mathscr{B}$ should determine whether $Z = e(g_{L+1}, h)$ or not.
Let $W(\overrightarrow{v})$ be $\{1 \leq i \leq L \mid v_i = *\}$ and $\overline{W}(\overrightarrow{v})$ be $\{1 \leq i \leq L \mid v_i \neq *\}$, and $W(\overrightarrow{v}\mid_j^k)$ be $\{i \in W(\overrightarrow{v} \mid j \leq i \leq k\}$. $\mathscr{B}$ executes with $\mathscr{A}$ as follows:

- **Init**: $\mathscr{A}$ sends two challenge vectors $\overrightarrow{v_0^*} \in \Sigma_L^*$ and $\overrightarrow{v_1^*} \in \Sigma_L^*$ where $W(\overrightarrow{v_0^*}) = W(\overrightarrow{v_1^*})$. $\mathscr{B}$ randomly chooses $\mu \in \{0, 1\}$. Let $\overrightarrow{v_\mu^*}$ be $(\overrightarrow{v_1^*}, \overrightarrow{v_2^*}, \ldots, \overrightarrow{v_L^*})$ for simplicity.
- **Setup**: $\mathscr{B}$ randomly chooses integers $\gamma, y, t_1, t_2, t_3, t_4, u_1, \ldots, u_L \in \mathbb{Z}_p$, then it sets

$$Y = e(g^\alpha, g^{\alpha^L} g^\gamma)^{t_1 t_2}, U_1 = g^{t_1}, U_2 = g^{t_2}, U_3 = g^{t_3}, U_4 = g^{t_4},$$
$$V = g^y \prod_{i \in \overline{W}(\overrightarrow{v_\mu^*})} g^{\alpha^{L+1-i} v_{\mu,i}^*}, \{H_i = g^{u_i - \alpha^{L+1-i}}\}_{i \in \overline{W}(\overrightarrow{v_\mu^*})}, \{H_i = g^{u_i}\}_{i \in W(\overrightarrow{v_\mu^*})}$$

The element w in public parameters is $g^{\alpha^{L+1} + \alpha\gamma}$. Since $g^{\alpha^{L+1}}$ is unknown to $\mathscr{B}$, w cannot computed by $\mathscr{B}$ directly.

- **Query Phase 1**: $\mathscr{A}$ sends a vector $\overrightarrow{\sigma_u} = (\sigma_1, \sigma_2, \ldots, \sigma_u)$ without matching the challenge vectors for key query. Let $k \in \overline{W}(\overrightarrow{v_\mu^*})$ which is the smallest integer for $\sigma_k \neq v_{\mu,k}^*$. $\mathscr{B}$ generates the corresponding key as follows. We start from $K_{2,i}$.

$$K_{2,0} = w^{t_2}(\prod_{i=1}^{L} H_i^{\sigma_i} V)^{r_1 t_2}$$

$$= (g^{\alpha^{L+1}+\alpha\gamma})^{t_2}(\prod_{\overline{W}(\overrightarrow{v_\mu^*})|_1^k} g^{u_i-\alpha^{L+1-i}} \prod_{W(\overrightarrow{v_\mu^*})|_1^k} (g^{u_i}))^{\sigma_i} \cdot g^{y+\sum_{\overline{W}(\overrightarrow{v_\mu^*})} \alpha^{L+1-i} v_{\mu,i}^*})^{r_1 t_2}$$

$$\stackrel{\text{def}}{=} (g^{\alpha^{L+1}+\alpha\gamma})^{t_2}(g^X)^{r_1 t_2}$$

where $X = \sum_{\overline{W}(\overrightarrow{v_\mu^*})} \alpha^{L+1-i} v_{\mu,i}^* + y + \sum_{\overline{W}(\overrightarrow{v_\mu^*})|_1^k}(u_i - \alpha^{L+1-i})\sigma_i + \sum_{W(\overrightarrow{v_\mu^*})|_1^k} u_i\sigma_i$ Since $\sum_{\overline{W}(\overrightarrow{v_\mu^*})|_1^k}(u_i - \alpha^{L+1-i})\sigma_i + \sum_{\overline{W}(\overrightarrow{v_\mu^*})|_1^k} u_i\sigma_i = \sum_{\overline{W}(\overrightarrow{v_\mu^*})|_1^k}(-\alpha^{L+1-i}\sigma_i) + \sum_{i=1}^k u_i\sigma_i$ and recall $\sigma_i = v_{\mu,i}^*$ for $i \in \overline{W}(\overrightarrow{v_\mu^*})|_1^{k-1}$ and $\sigma_k \neq v_{\mu,k}^*$. Hence, we have

$$X = \alpha^{L+1-k}\Delta_k + \sum_{\overline{W}(\overrightarrow{v_\mu^*})|_{k+1}^L} \alpha^{L+1-i} v_{\mu,i}^* + \sum_{i=1}^k x_i\sigma_i + y$$

where $\delta_k = v_{\mu,k}^* - \sigma_k$. Then we choose $\hat{r}_1$ randomly in $\mathbb{Z}_p$, and implicitly set $r_1 = \frac{-\alpha^k}{\delta_k} + \hat{r}_1$. $K_{2,0}$ can be represented as

$$\left[g^{\alpha^{L+1}+\alpha\gamma} \cdot g^{-\alpha^{L+1}} \cdot g^{\sum_{i\in W(v_\mu^*)|_{k+1}^L} \frac{-\alpha^{L+1-i+k} v_{\mu,i}^*}{\Delta_k}} \cdot g^{a^k(-\frac{\sum_{i=1}^k x_i\sigma_i + y}{\Delta_k})} \cdot (V\prod_{i=1}^{K} h_i^{\sigma_i})^{\hat{r}_1} \right]^{t_2}$$

$$= \left[g^{\alpha\gamma} \cdot g^{\sum_{i\in W(v_\mu^*)|_{k+1}^L} \frac{-\alpha^{L+1-i+k} v_{\mu,i}^*}{\Delta_k}} \cdot g^{a^k(-\frac{\sum_{i=1}^k x_i\sigma_i + y}{\Delta_k})} \cdot (V\prod_{i=1}^{K} h_i^{\sigma_i})^{\hat{r}_1} \right]^{t_2}$$

For $\hat{k} = 1$ to N, we compute $K_{2,\hat{k}}$ as

$$\left[g^{y+\sum_{\overline{W}(\overrightarrow{v_\mu^*})} \alpha^{L+1-i} v_{\mu,i}^*} \cdot (\prod_{\overline{W}(\overrightarrow{v_\mu^*})|_1^{k-1}} g^{u_i-\alpha^{L+1-i}} \cdot \prod_{W(v_\mu^*)|_1^{k-1}} (g^{u_i})^{\sigma_i})^{\frac{-\alpha^k i^{\hat{k}}}{\Delta_k} + \hat{r}_1 i^{\hat{k}}} \right]^{t_2}.$$

Note that $K_{3,i} = K_{2,i}^{\frac{t_1}{t_2}}$, so we can compute $K_{3,i}$ easily from $K_{2,i}$. Next we choose random $r_2 \in \mathbb{Z}_p$ and compute $K_{4,k} = \prod_{i=1}^{L}(H_i^{z_i}V)^{i^k r_2 t_4}$ and $K_{5,k} = \prod_{i=1}^{L}(H_i^{z_i}V)^{i^k r_2 t_3}$ for $k = 0, \cdots, N$ since $V, H_0, \cdots, H_L$ are known.
At last we can simulate the first element in the key:

$$K_1 = g^{r_1 t_1 t_2 + r_2 t_3 t_4} = (g^{\alpha_k})^{-t_1 t_2/\Delta_k} \cdot g^{\hat{r}_1 t_1 t_2 + r_2 t_3 t_4}$$

- **Challenge**: Two message M_0, M_1 are submitted to $\mathscr{B}$ by $\mathscr{A}$. $\mathscr{B}$ randomly chooses $s_1, s_2 \in \mathbb{Z}_p$ and computes:

$$C_0 = M_\mu \cdot Z^{t_1 t_2} \cdot e(g^\alpha, h)^{t_1 t_2 \gamma}, C_1 = (h^{y+\sum_{i=1}^{L} u_i v^*_{\mu,i}})^{\prod_{k=1}^{\tau}(i-j_k)},$$
$$C_2 = h^{t_1} U_1^{s_1}, C_3 = U_2^{s_1}, C_4 = h^{t_3} U_3^{-s_2}, C_5 = U_4^{s_2}.$$

  Here we implicitly set $g^s = h$. If $Z = e(g,h)^{\alpha^{L+1}}$, it is a valid ciphertext encrypted to M_b. Otherwise, if T is a random element of $\mathbb{G}_T$, the challenge ciphertext is an encryption to a random message.
- **Query Phase 2**: Query Phase 1 is repeated.
- **Guess**: $\mathscr{A}$ outputs $\mu' \in \{0,1\}$. $\mathscr{B}$ outputs 1 when $\mu' = \mu$ then, otherwise it outputs 0.

If $\mu' = \mu$, then the simulation equals to the real game. Therefore, the probability of $\mathscr{A}$ to guess μ correctly is $\frac{1}{2} + \varepsilon$. If $\mathscr{B}$ outputs 1, then Z is random in $\mathbb{G}_T$, then the probability of $\mathscr{A}$ to guess b correctly is $\frac{1}{2}$. Therefore, the advantage of $\mathscr{B}$ to solve the decision L-BDHE assumption is exactly ε. □ □

Lemma 4.2. *Under the decision linear assumption,* G$_2$*,* G$_3$ *and* G$_4$ *are indistinguishable.*

The proof of Lemma 4.2 will be provided in the full version of this paper due to space limitation.
Proof of Theorem 4.1. It is straightforward from Lemma 4.1 and Lemma 4.2. □

5. Comparison

We give a brief comparison for efficiency in the following Table 1. We compare our HVE scheme with some current ciphertext policy HVE schemes, including Hattori et al.'s scheme [4], Liao et al.'s scheme [10], Phuong et al.'s scheme [5][2] and Murad et al.'s scheme(restricted to AND-gate policy). All the schemes are implemented in Intel Core i5-8250U 1.60GHz, 8G RAM and Ubuntu 16.04. We consider the times of Setup, Key Generation, Encryption and Decryption in these schemes. We can see that decryption in our scheme is much quicker than other schemes. The weakness in our scheme(also in Phuong et al.'s scheme) is that we need a long time to generate a key. Since many applications need instant decryption, our scheme may have great advantage in those instant applications.

6. Conclusion

Hidden Vector Encryption can hide the information of vector used to encrypt the message. Phuong et al. proposed two HVE schemes with constant ciphertext size in composite order and prime order groups respectively. We give an analysis on Phuong et al.'s

[2]We only compare Phuong et al.'s first scheme in composite order groups because the second scheme is not secure as we show in Section 3.

Table 1. Efficiency Comparison (ms)

Scheme	Group	Setup	Key Generation	Encryption	Decryption
Hattori et al.[2011] [4]	Composite	1,033	17,710	13,405	332,629
Phuong et al.[2014] [5]	Composite	622	499,032	805	22,669
Liao et al.[2015] [10]	Prime	307	2,415	462	13,586
Murad et al.[2019] [13]	Prime	243	1,635	305	1,721
Our scheme	Prime	151	50,387	93	947

Note: we assume that the length of a vector is 1000 and the number of wildcard is 100.

prime order HVE scheme and show their scheme doesn't satisfy the vector-hiding property. Furthermore, we propose an improved construction which also has constant ciphertext size. The security of proposed scheme is proven under the L-BDHE assumption. Future work may be finding more efficient or secure HVE schemes under simple assumptions.

Acknowledgements

This work is supported by the National Key Research and Development Program of China No. 2020YFB1005404, Ministry of Education-China Mobile Scientific Research Fund MCM20200104, and National Natural Science Foundation of China (Grant No. 61872051).

References

[1] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In S.P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 535–554. Springer-Verlag, 2007.

[2] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98, New York, NY, USA, 2006. ACM.

[3] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *SP '07: IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[4] Mitsuhiro Hattori, Takato Hirano, Takashi Ito, Nori Matsuda, Takumi Mori, Yusuke Sakai, and Kazuo Ohta. Ciphertext-policy delegatable hidden vector encryption and its application to searchable encryption in multi-user setting. In Liqun Chen, editor, *Cryptography and Coding - 13th IMA International Conference, IMACC 2011, Oxford, UK, December 12-15, 2011. Proceedings*, volume 7089 of *Lecture Notes in Computer Science*, pages 190–209. Springer, 2011.

[5] Tran Viet Xuan Phuong, Guomin Yang, and Willy Susilo. Efficient hidden vector encryption with constant-size ciphertext. In Miroslaw Kutylowski and Jaideep Vaidya, editors, *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part I*, volume 8712 of *Lecture Notes in Computer Science*, pages 472–487. Springer, 2014.

[6] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In N. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer-Verlag, 2008.

[7] Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *LNCS*, pages 191–208. Springer-Verlag, 2010.

[8] Jae Hong Seo, Tetsutaro Kobayashi, Miyako Ohkubo, and Koutarou Suzuki. Anonymous hierarchical identity-based encryption with constant size ciphertexts. In S. Jarecki and G. Tsudik, editors, *Public Key Cryptography - PKC 2009*, volume 5443 of *LNCS*, pages 215–234. Springer-Verlag, 2009.

[9] Michel Abdalla, Dario Catalano, Alexander W. Dent, John Malone-Lee, Gregory Neven, and Nigel P. Smart. Identity-based encryption gone wild. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006*, volume 4052 of *LNCS*, pages 300–311. Springer-Verlag, 2006.

[10] Liao Zhenhua, Wang Jinmiao, and Lang Bo. A ciphertext-policy hidden vector encryption scheme supporting multiuser keyword search. *Secur. Commun. Networks*, 8(6):879–887, 2015.

[11] Kwangsu Lee. Transforming hidden vector encryption schemes from composite to prime order groups. In Seokhie Hong and Jong Hwan Park, editors, *Information Security and Cryptology - ICISC 2016 - 19th International Conference, Seoul, South Korea, November 30 - December 2, 2016, Revised Selected Papers*, volume 10157 of *Lecture Notes in Computer Science*, pages 101–125, 2016.

[12] James Bartusek, Brent Carmer, Abhishek Jain, Zhengzhong Jin, Tancrède Lepoint, Fermi Ma, Tal Malkin, Alex J. Malozemoff, and Mariana Raykova. Public-key function-private hidden vector encryption (and more). In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 489–519. Springer, 2019.

[13] Miada Murad, Yuan Tian, and Mznah A. Rodhaan. Computationally efficient fine-grain cube cp-abe scheme with partially hidden access structure. In Yuan Tian, Tinghuai Ma, and Muhammad Khurram Khan, editors, *Big Data and Security*, pages 135–156, Singapore, 2020. Springer Singapore.

[14] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In C. Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *LNCS*, pages 290–307. Springer-Verlag, 2006.