ESORICS 2014

Efficient Hidden Vector Encryption with Constant-Size Ciphertext

Tran Viet Xuan Phuong, Guomin Yang, Willy Susilo

University of Wollongong

Content

- Hidden Vector Encryption
- Motivation
- Background
- Ciphertext Policy Hidden Vector Encryption
- Proposed schemes
- Security Proof
- Comparison and Conclusion

Hidden Vector Encryption (HVE)

Predicate Encryption: $P{\downarrow}Y\ (X)$

-Suppose two attribute vector X, Y with length L:

X	1	1	1	…	8
Y	*	1	*	…	8

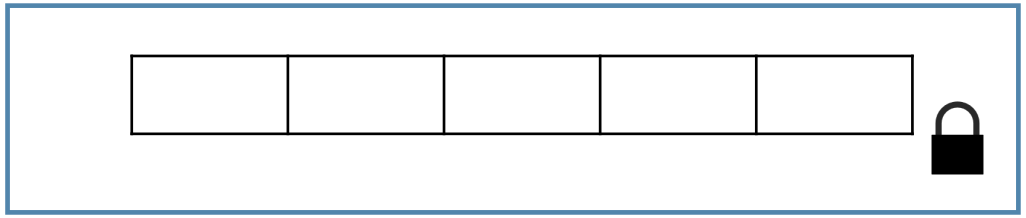
Specific value

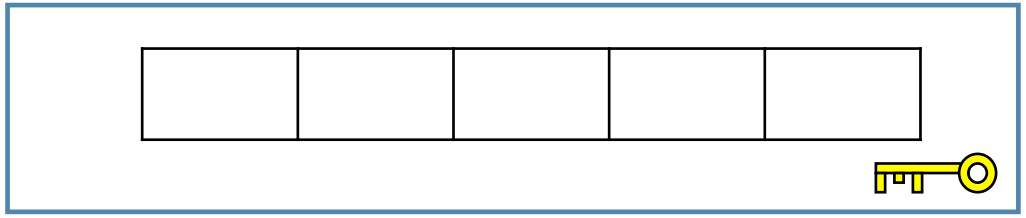
'*': don't care

Hidden Vector Encryption (HVE)

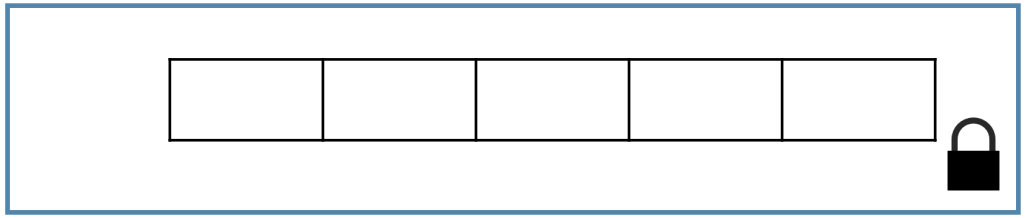
-There are two flavors :

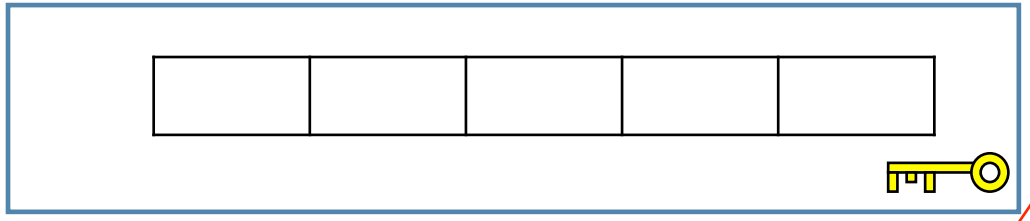
Key Policy (KP-) HVE





Ciphertext Policy (CP-) HVE





Motivation

A table of comparion on Ciphertext Size and Key Size among HVE schemes

Scheme	Type	Constant CT Size	Constant Key Size
Katz et al.	Key Policy	No	No
Shi, Waters	Key Policy	No	No
Ivovino and Persiono	Key Policy	No	No
Sedghi et all	Key Policy	No	Yes
Lee and Dong	Key Policy	No	Yes
Park	Key Policy	No	Yes
Hattori et al	Ciphertext Policy	No	No
Ours	Ciphertext Policy	**Yes**	No

Background of our scheme

Bilinear map on Composite Order Groups

$\boldsymbol{n=pq}$: p,q large prime numbers

$\boldsymbol{e}$:$\mathbb{G}\times\mathbb{G}\to\mathbb{G}_{\boldsymbol{T}}$: bilinear map on composite order groups of cyclic groups $\mathbb{G}$, $\mathbb{G}_T$

If:

1. Bilinearity: $\boldsymbol{e(u^a,u^b)=e(u^b,u^a)=e(u,v)^{ab}}$, $\forall u,v\in\mathbb{G}$ and $a,b\in\mathbb{Z}_p$.
2. Non-degeneracy: $\boldsymbol{e(g,g)\neq 1}$

$\mathbb{G}_{\boldsymbol{p}}$, $\mathbb{G}_{\boldsymbol{q}}$: two subgroups of $\mathbb{G}$ order p,q

Then $\mathbb{G}=\mathbb{G}_p\times\mathbb{G}_q$, $\mathbb{G}_T=\mathbb{G}_{T,p}\times\mathbb{G}_{T,q}$; g_p,g_q generators of $\mathbb{G}_p$, $\mathbb{G}_q$

$\boldsymbol{e(h_p,h_q)=e(g_p^a,g_q^b)=e(g^{pa},g^{qb})=e(g,g)^{pqab}=1}$

$\forall h_p\in\mathbb{G}_p$, $h_q\in\mathbb{G}_q$, $g\in\mathbb{G}$.

Background of our scheme

Viète's formulas

$w=(w\downarrow 1, w\downarrow 2, *, .., *, w\downarrow L)$

$z=(z\downarrow 1, z\downarrow 2, z\downarrow 3 \ldots z\downarrow L-1, z\downarrow L)$

$J=\{j\downarrow 1, j\downarrow 2, .., j\downarrow n\} \subset \{1,\ldots,L\}$

positions wildcard in w

$w\downarrow i = z\downarrow i \vee w\downarrow i = *$ for $i=1,\ldots,L$ $\longrightarrow \sum i=1, i\notin J\uparrow L \square w\downarrow i \prod j\in J\uparrow \square (i-j) = \sum i=1\uparrow L \square z\downarrow i \prod j\in J\uparrow \square (i-j)$

$\prod j\in J\uparrow \square (i-j) = \sum k=0\uparrow n \square a\downarrow k i\uparrow k$ coefficients of J $\longrightarrow \sum i=1, i\notin J\uparrow L \square w\downarrow i \prod j\in J\uparrow \square (i-j) = \sum k=0\uparrow n \square a\downarrow k \sum i=1\uparrow L \square z\downarrow i\, i\uparrow k$

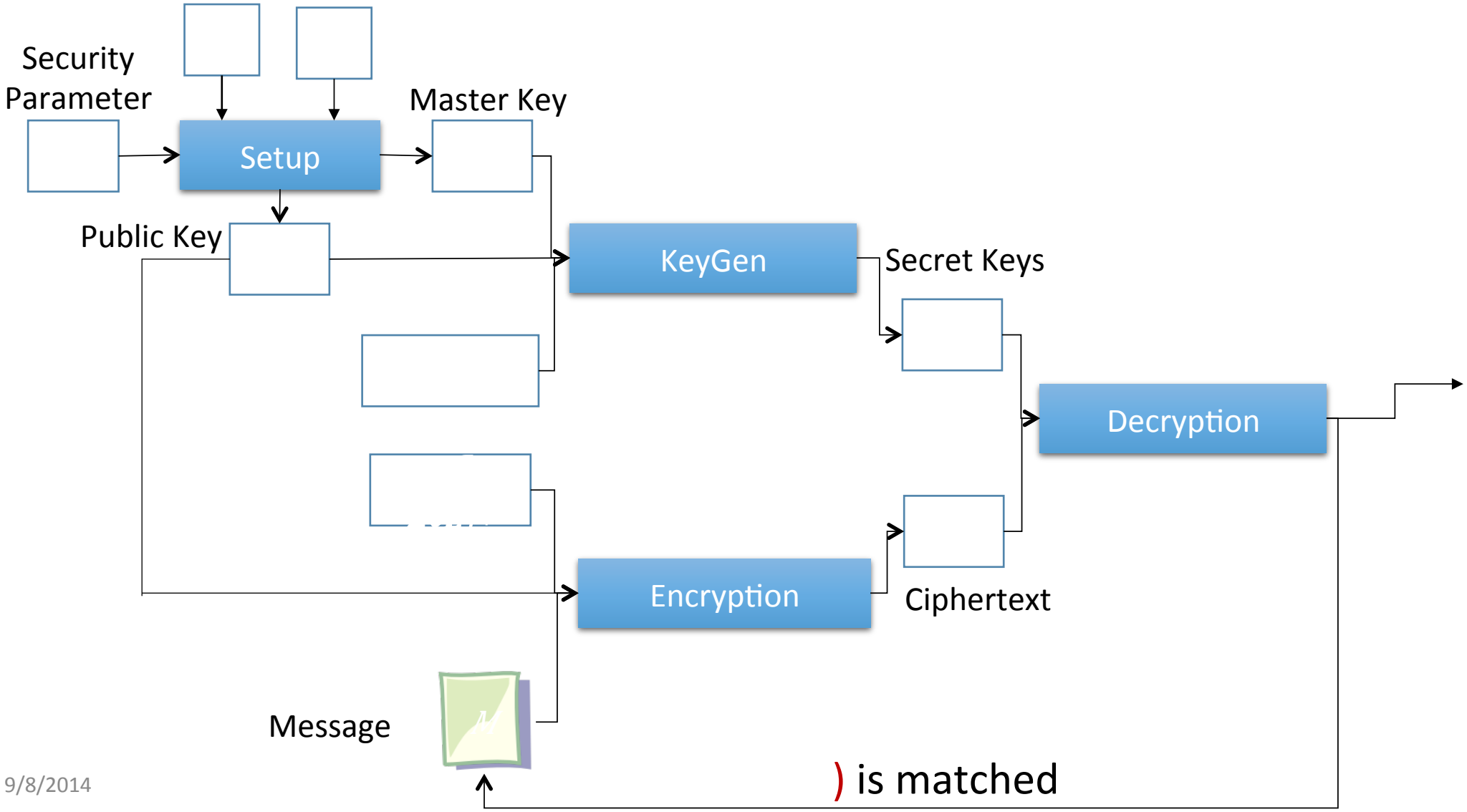
Hiding computation, $H\downarrow i \in\downarrow R\, G$ $\longrightarrow \prod i=1, i\notin J\uparrow L \square H\downarrow i\uparrow w\downarrow i \prod j\in J\uparrow \square (i-j) = \prod k=0\uparrow n \square (\prod i=1, \uparrow L \square H\downarrow$

Combine all elements in
=> Constant Ciphertext Size

Using Viète formulas, construct $a\downarrow k$: $a\downarrow k = (-1)\uparrow k \sum 1\le i\downarrow 1 < i\downarrow 2 < \ldots < i\downarrow k \le n\uparrow \square j\downarrow i\downarrow 1\ j\downarrow i\downarrow 2\ \ldots j\downarrow i\downarrow k\ , 0\le k\le n$

Ciphertext Policy Hidden Vector Encryption (CP-HVE)

Model of CP-HVE



Ciphertext Policy Hidden Vector Encryption (CP-HVE)

Security Model of CP-HVE

Game: an adversary $\mathcal{A}$<=>$\mathcal{B}$ a challenger

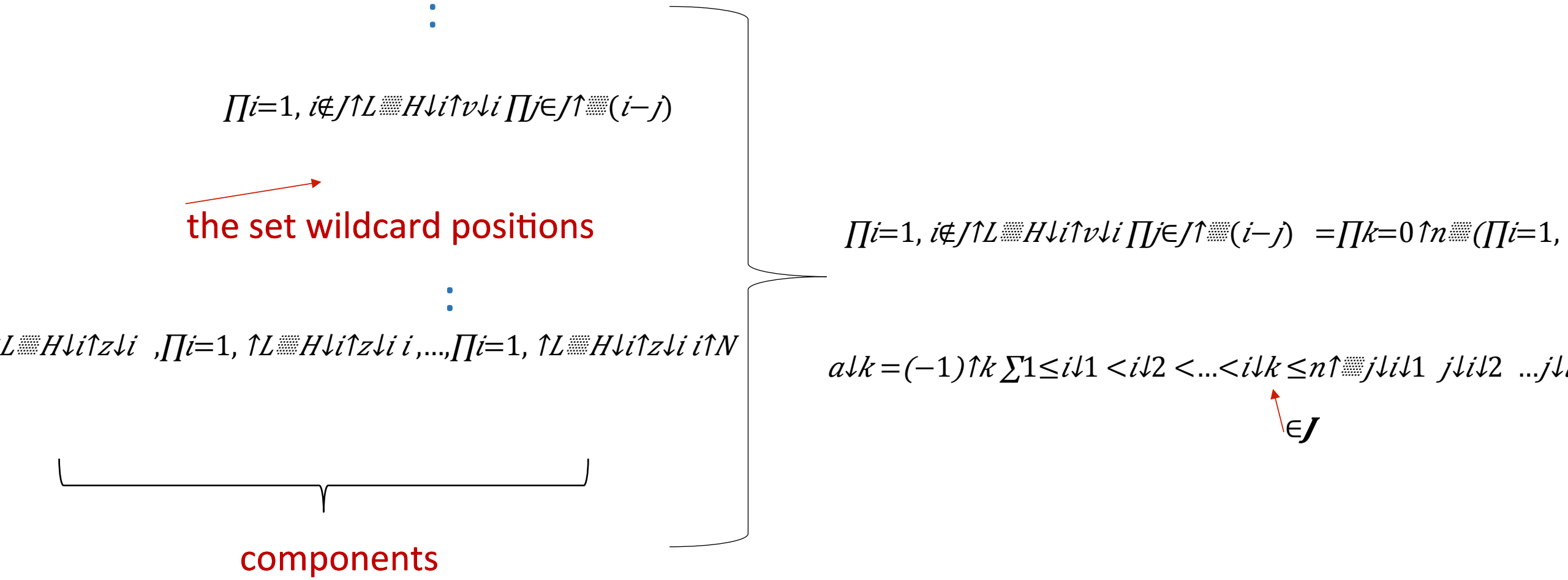
- **Init**: $\mathcal{A}$ chooses two target patterns: $v\downarrow 0\uparrow * =(v\downarrow 0,1 ,v\downarrow 0,2 ,...,v\downarrow 0,L)$ and $v\downarrow 1\uparrow * =(v\downarrow 1,1 ,v\downarrow 1,2 ,...,v\downarrow 1,L)$
- **Setup**: $\mathcal{B}$ generate PK, MSK, then pass PK to $\mathcal{A}$.
- **Phase 1**: $\mathcal{A}$ adaptively issues key queries $\sigma =(\sigma\downarrow 1 ,...,\sigma\downarrow L)\in\Sigma\downarrow L$ (σ does not match $v\downarrow o\uparrow * ,v\downarrow 1\uparrow *$)

(there exist $i,j\in\{1,...,L\}$ as $v\downarrow 0,i\uparrow * \neq * \wedge v\downarrow 0,i\uparrow * \neq \sigma\downarrow i$ and $v\downarrow 1,j\uparrow * \neq * \wedge v\downarrow 1,j\uparrow * \neq \sigma\downarrow j$), and returns the corresponding decryption key to $\mathcal{A}$.

- **Challenge**: $\mathcal{A}$ outputs $M\downarrow 0\uparrow * , M\downarrow 1\uparrow *$. $\mathcal{B}$ picks $\beta\leftarrow\{0,1\}$ and runs $C\uparrow * =Encrypt(PK,v\downarrow\beta\uparrow * ,M\downarrow\beta\uparrow *)$, then passes $C\uparrow *$ to $\mathcal{A}$.
- **Phase 2** : same as learning Phase 1
- **Output** : $\mathcal{A}$ outputs a bit $\beta\uparrow '$ as her guess for β

Our scheme : Main Idea

Assump : the maximum number wildcards



Our scheme : Setup

- Setup $(1\uparrow k, \Sigma, L)$:
  - Chooses $N \ll L$ be the maximum number of wildcards
  - Picks p, q large primes, generates bilinear groups $\mathbb{G}, \mathbb{G}\downarrow T$ of composite order $n = pq$
  - Selects $g\downarrow p \in \mathbb{G}\downarrow p$, $g\downarrow q \in \mathbb{G}\downarrow q$. Then it selects random elements:

$$g, f, v, v\uparrow' , h\downarrow 1 , \ldots, h\downarrow l , h\downarrow 1\uparrow' , \ldots, h\downarrow L\uparrow' , w \in \mathbb{G}\downarrow p$$
$$R\downarrow g . R\downarrow f , R\downarrow v , R\downarrow h\downarrow 1 , \ldots, R\downarrow h\downarrow L , R\downarrow h\downarrow 1\uparrow' , \ldots, R\downarrow h\downarrow L\uparrow' \in \mathbb{G}\downarrow q$$

computes:

$$G = gR\downarrow g , F = fR\downarrow f , V = vR\downarrow v , V\uparrow' = v\uparrow' R\downarrow v\uparrow' ,$$
$$H\downarrow 1 = h\downarrow 1 R\downarrow h\downarrow 1 , \ldots, H\downarrow L = h\downarrow L R\downarrow h ,$$
$$H'\downarrow 1 = h'\downarrow 1 R\downarrow h'\downarrow 1 , \ldots, H'\downarrow L = h'\downarrow L R\downarrow h'\downarrow L , E = (g, w)$$

  - Then it creates :

$$PK = (g\downarrow p , g\downarrow q , G, F, V, V\uparrow' , H\downarrow 1 , \ldots, H\downarrow L , H\uparrow' \downarrow 1 , \ldots, H\downarrow L\uparrow' , E),$$
$$MSK = (p, q, g, f, v, v\uparrow' , h\downarrow 1 , \ldots, h\downarrow L , h\downarrow 1\uparrow' , \ldots, h\downarrow L\uparrow' , w).$$

Our scheme : Encryption

- Encryption($PK, M, v = (v\downarrow 1, \ldots, v\downarrow L) \in \Sigma\downarrow L\uparrow *$):
  - Suppose v contains $\tau \leq N$ wildcards which occur at positions $J = \{j\downarrow 1, \ldots, j\downarrow \tau\}$.
  - Chooses $s \in\downarrow R \mathbb{Z}\downarrow n$ and $Z\downarrow 1, Z\downarrow 2, Z\downarrow 3, Z\downarrow 4 \in\downarrow R \mathbb{G}\downarrow q$.
  - Using Viéte formulas to compute $a\downarrow k$ for $k = 1,2,\ldots,\tau$, and $t = a\downarrow 0$.

$$C\downarrow 0 = M \cdot E\uparrow s, \; C\downarrow 1 = G\uparrow s/t \; Z\downarrow 1, C\downarrow 2 = F\uparrow s \; Z\downarrow 2$$

$$C\downarrow 3 = ((\prod i=1\uparrow L \blacksquare VH\downarrow i\uparrow v\downarrow i \;)\uparrow \prod k=1\uparrow \tau \blacksquare (i - j\downarrow k)\;)\uparrow s/t \cdot Z\downarrow 3, C\downarrow 4 = ((\prod i=1\uparrow L \blacksquare V' H'\downarrow i\uparrow v\downarrow i \;)\uparrow \prod k=1\uparrow \tau \blacksquare (i - j\downarrow k$$

  - Then ciphertext:

$$CT = (C\downarrow 0, C\downarrow 1, C\downarrow 2, C\downarrow 3, C\downarrow 4, J)$$

Our scheme : Key Generation

- Key Generation($MSK, z=(z_1,\ldots,z_L)\in\Sigma^L$):
  - Chooses $r_1, r_1', r_2 \in_R \mathbb{Z}_n$, and computes:

$$K_1 = g^{r_1},\ K_2 = g^{r_1'},\ K_3 = g^{r_2},$$

$$\left(K_{4,0} = w\left(\prod_{i=1}^{L} h_i^{z_i}\right)^{r_1}\left(\prod_{i=1}^{L} h_i'^{z_i}\right)^{r_2},\ K_{4,1} = w\left(\prod_{i=1}^{L} h_i^{z_i}\right)^{r_1}\left(\prod_{i=1}^{L} h_i'^{z_i}\right)^{r_2'},\ \ldots K_{4,N} = w\left(\prod_{i=1}^{L} h_i^{z_i}\right)^{N r_1}\left(\prod_{i=1}^{L} h_i'^{z_i}\right)^{N r_2'}\right)$$

  - The secret key :

$$SK = (K_1, K_2, K_3, K_{4,0}, \ldots, K_{4,N})$$

Our scheme : Decryption

- Decryption(CT, SK):

$$CT=(C\downarrow 0, C\downarrow 1, C\downarrow 2, C\downarrow 3, C\downarrow 4, J)$$

$$SK=(K\downarrow 1, K\downarrow 2, K\downarrow 3, K\downarrow 4,0, \ldots, K\downarrow 4,N)$$

  - Apply Viète formulas to compute:

$$a\downarrow k=(-1)\uparrow k \sum 1\le i\downarrow 1<i\downarrow 2<\ldots<i\downarrow k\le \tau\uparrow \square j\downarrow i\downarrow 1\ j\downarrow i\downarrow 2\ \ldots j\downarrow i\downarrow k\ , 0\le k\le \tau$$

  - Outputs:

$$M=e(K\downarrow 1, C\downarrow 3)\cdot e(K\downarrow 2, C\downarrow 4)\cdot e(K\downarrow 3, C\downarrow 3)/e(\prod k=0\uparrow \tau \square K\downarrow 4,k\uparrow a\downarrow k, C\downarrow 1)\cdot C\downarrow 0$$

Three Complexity Assumptions

1. **The Decisional L-cBDHE assumption:**

$$Let\ g\downarrow p, h \leftarrow_R \mathbb{G}\downarrow p,\ g\downarrow q \leftarrow_R \mathbb{G}\downarrow q,\ \alpha \leftarrow_R \mathbb{Z}\downarrow n,$$

$$Z=(g\downarrow p, g\downarrow q, h, g\downarrow p\uparrow\alpha, \ldots, g\downarrow p\uparrow\alpha\uparrow L,\ g\downarrow p\uparrow\alpha\uparrow L+2, \ldots, g\downarrow p\uparrow\alpha\uparrow 2L),$$

$$T=e(g\downarrow p, h)\uparrow\alpha\uparrow L+1,\ and\ R\leftarrow \mathbb{G}\downarrow T, p$$

We say that decisional L-cBDHE assumption holds if any PPT algorithm A:

$$|\Pr[A(Z,T)=1]-\Pr[A(Z,R)=1]|\leq \epsilon(k)$$: negligible function of k

2. **The $L-$cDDH assumption:**

$$Let\ g\downarrow p \leftarrow_R \mathbb{G}\downarrow p,\ g\downarrow q, R\downarrow 1, R\downarrow 2, R\downarrow 3 \leftarrow_R \mathbb{G}\downarrow q,\ \alpha, \beta \leftarrow_R \mathbb{Z}\downarrow n,$$

$$Z=(g\downarrow p, g\downarrow q, g\downarrow p\uparrow\alpha, \ldots, g\downarrow p\uparrow\alpha\uparrow L,\ g\downarrow p\uparrow\alpha\uparrow L+1\ R\downarrow 1, g\downarrow p\uparrow\alpha\uparrow L+1\ \beta\ R\downarrow 2),$$

$$T=g\downarrow p\uparrow\beta\ R\downarrow 3,\ and\ R\leftarrow \mathbb{G}$$

We say that decisional L-cDDH assumption holds if any PPT algorithm A:

$$|\Pr[A(Z,T)=1]-\Pr[A(Z,R)=1]|\leq \epsilon(k)$$: negligible function of k

Three Complexity Assumptions

3. **The BSD assumption:**

$Let\ g\downarrow p \leftarrow_R \mathbb{G}\downarrow p\ ,\ g\downarrow q \leftarrow_R \mathbb{G}\downarrow q\ ,$

$Z=(g\downarrow p\ ,g\downarrow q\ ),$

$T=\mathbb{G}\downarrow T,p\ ,\ and\ R\leftarrow\mathbb{G}\downarrow T,p$

We say that decisional BSD assumption holds if any PPT algorithm A:

$|\Pr[A(Z,T)=1]-\Pr[A(Z,R)=1]|\leq\epsilon(k)$: negligible function of k

Security Proof

- Theorem 1 : Our scheme is secure if three complexity assumptions hold
  - We prove by the following sequence of games:

$$\begin{aligned}
Game_1 &: [C_0 \cdot R_p, C_1, C_2, C_3, C_4] \\
Game_2 &: [R_0, C_1, C_2, C_3, C_4] \\
Game_3 &: [R_0, C_1, C_2, R_3, C_4] \\
Game_4 &: [R_0, C_1, C_2, R_3, R_4]
\end{aligned}$$

Where $R_p \in_R \mathbb{G}_T, p$, $R_0 \leftarrow_U \mathbb{G}_T$, and $R_0, R_3, R_4 \leftarrow_U \mathbb{G}$.

In $Game_4$ the challenge CT is **independent of the message and the encryption vector**, means the adversary **no advantage** in winning the game over random guess.

Our scheme based on prime order

Bilinear Map

: groups with order

A bilinear map

Bilinearity

For all ,

Non-degeneracy

Decision L-Bilinear Diffie Hellman Exponent Assumption (L-BDHE)

where , , two generators of ,

Definition 4: *We say that the L-BDHE assumption holds in $\mathbb{G}$ if for any probabilistic polynomial time algorithm $\mathcal{A}$*

$$|\Pr[\mathcal{A}(g, h, y_{\downarrow g, \alpha, L}, e(g_{\downarrow L+1}, h)) = 1] - [\mathcal{A}(g, h, y_{\downarrow g, \alpha, L}, T) = 1]| \leq \epsilon(k)$$

Our scheme based on prime order

- Setup($1\uparrow k, \Sigma, L$):

  - Chooses $N \ll L$ be the maximum number of wildcards
  - Generates bilinear groups $\mathbb{G}$, $\mathbb{G}\downarrow T$ with order p,
  - Selects $L+1$ random generators V, $H\downarrow 1$,…,$H\downarrow L \in \mathbb{G}$.
  - Randomly choose $g, w, f \in \mathbb{G}$,
  - Set $Y = e(g, w)$.
  - Then it creates :

PK = $(Y, V, H\downarrow 1, \ldots, H\downarrow L, g, f, p, \mathbb{G}, \mathbb{G}\downarrow T, e)$,

MSK= w

Our scheme based on prime order

- Encryption($PK, M, v = (v\downarrow 1, \ldots, v\downarrow L \in \Sigma\downarrow L\uparrow *)$):
  - Suppose v contains $\tau \leq N$ wildcards which occur at positions $J = \{j\downarrow 1, \ldots, j\downarrow \tau\}$.
  - Chooses $s \in\downarrow R \mathbb{Z}\downarrow n$.
  - Using Viéte formula to compute $a\downarrow k$ for $k = 1,2,\ldots,\tau$, and $t = a\downarrow 0$.

$$C\downarrow 0 = M Y\uparrow s,\ C\downarrow 1 = g\uparrow s/t,\ C\downarrow 2 = f\uparrow s,\qquad C\downarrow 3 = ((\prod i=1\uparrow L \blacksquare VH\downarrow i\uparrow v\downarrow i)\uparrow \prod k=1\uparrow \tau \blacksquare (i - j\downarrow k))\uparrow s/t \cdot$$

  - Then ciphertext:

$$CT = (C\downarrow 0, C\downarrow 1, C\downarrow 2, C\downarrow 3, J)$$

Our scheme based on prime order

- Key Generation($MSK, z=(z\downarrow 1 ,...,z\downarrow L)\in \Sigma\downarrow L$):
  - Chooses $r,r\downarrow 1 \in\downarrow R \mathbb{Z}\downarrow n$, and creates the secret key as:

$$K\downarrow 1 =g\uparrow r ,K\downarrow 2 =g\uparrow r\downarrow 1 \ ,$$

$$(\blacksquare K\downarrow 3,0 =w(\prod i=1\uparrow L\blacksquare H\downarrow i\uparrow z\downarrow i \ V)\uparrow r \ f\uparrow r\downarrow 1 \ ¦$$
$$K\downarrow 3,1 =(\prod i=1\uparrow L\blacksquare H\downarrow i\uparrow z\downarrow i \ V)\uparrow ir \ ...K\downarrow 3,N =$$
$$(\prod i=1\uparrow L\blacksquare H\downarrow i\uparrow z\downarrow i \ V)\uparrow i\uparrow N \ r \ )\downarrow .$$

  - The secret key :

$$SK=(K\downarrow 1 , K\downarrow 2 ,K\downarrow 3,0 ,...,K\downarrow 3,N))$$

Our scheme based on prime order

- Decryption(CT, SK):
  - Apply Viète formulas to compute:

$$a_k = (-1)^k \sum_{1 \le i_1 < i_2 < \ldots < i_k \le \tau} j_{i_1} j_{i_2} \ldots j_{i_k}, \quad 0 \le k \le \tau$$

  - Outputs:

$$M = e(K_1, C_3) \cdot e(K_2, C_2) / e(\prod_{k=0}^{\tau} K_{3,k}^{a_k}, C_1) \cdot C_0$$

Comparison

Scheme	Group Order	Ciphertext Size*	Decryption Cost	Assumption
Katz et al. [6]	pqr	$(2L+1)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$(2L+1)$p	c3DH
Shi–Waters [20]	pqr	$(L+3)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$(L+3)$p	c3DH
Ivovino–Persiano[21]	p	$(2L+1)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$(2L+1)$p	DBDH + DLIN
Sedghi et al. [8]	p	$(N+3)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	3p	DLIN
Lee–Dong [25]	pqr	$(L+2)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	4p	cBDH BSD c3DH
Park [23]	p	$(2L+3)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	5p	DBDH+DLIN
Hattori et al. [9]	pq	$(2L+3)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	3p	$L-w$DBDHI BSD $L-c$DDH
Our scheme 1	pq	$4\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$4p$	L-cBDHE, BSD, L-cDDH
Our scheme 2	p	$3\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$3p$	L-BDHE

***we do not count the wildcard positions when measuring the ciphertext size.**

Conclusion

- We proposed two efficient ciphertext policy Hidden Vector Encryption scheme.
- Our encryption schemes can achieve constant ciphertext size.
- We proved the security of our schemes in a selective security model.
- Future work : achieve adaptive security

Thank you for your attention