Hidden-Vector Encryption with Groups of Prime Order

Vincenzo Iovino[1] and Giuseppe Persiano[1]

Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84084 Fisciano (SA), Italy. {iovino,giuper}@dia.unisa.it.

Abstract. Predicate encryption schemes are encryption schemes in which each ciphertext Ct is associated with a binary attribute vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ and keys K are associated with predicates. A key K can decrypt a ciphertext Ct if and only if the attribute vector of the ciphertext satisfies the predicate of the key. Predicate encryption schemes can be used to implement fine-grained access control on encrypted data and to perform search on encrypted data.

Hidden vector encryption schemes [Boneh and Waters – TCC 2007] are encryption schemes in which each ciphertext Ct is associated with a binary vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ and each key K is associated with binary vector $\boldsymbol{y} = (y_1, \cdots, y_n)$ with "don't care" entries (denoted with $\star$). Key K can decrypt ciphertext Ct if and only if $\boldsymbol{x}$ and $\boldsymbol{y}$ agree for all i for which $y_i \neq \star$.

Hidden vector encryption schemes are an important type of predicate encryption schemes as they can be used to construct more sophisticated predicate encryption schemes (supporting for example range and subset queries).

We give a construction for hidden-vector encryption from standard complexity assumptions on bilinear groups of *prime order*. Previous constructions were in bilinear groups of *composite order* and thus resulted in less efficient schemes. Our construction is both payload-hiding and attribute-hiding meaning that also the privacy of the attribute vector, besides privacy of the cleartext, is guaranteed.

1 Introduction

Traditional public key encryption schemes are well tailored for point-to-point security in which a sender wishes to send private messages to the owner of the public key. Recently, there has been a trend for private user data to be stored over the Internet by a third party server. It is then expected that user will encrypt the data so to preserve the privacy of the data itself. If a traditional encryption scheme is employed then user will not be able to search its data. Indeed, the user has to download and the decrypt its data and then perform the search; which can be very inconvenient.

This problem has been first studied by Boneh et al. [BDOP04] that introduced the concept of an encryption scheme supporting test equality. Roughly speaking, in such an encryption scheme, the owner of the public key can compute, for any message M, a trapdoor information K_M that allows the server that physically holds the data to check whether a given ciphertext encrypts message M without obtaining any additional information. Boneh et al. [BDOP04] suggested to use this system for storing encrypted e-mail messages on a server so that the user could decide to download only the e-mail messages with a given subject without having to compromise his privacy (and without having to download and decrypt all the messages).

Recently along this line of research, Goyal et al. [GPSW06] have introduced the concept of an attribute-based encryption scheme (ABE scheme). In an ABE scheme, a cyphertext is labeled with a set of attributes and private keys are associated with a predicate. A private key can decrypt a ciphertext iff the attributes of the ciphertext satisfy the predicate associated with the key. An ABE schem can thus been seen as a special encryption scheme for which, given the key associated with a predicate P, one can test whether a given ciphertext Ct carries a message M that satisfies predicates P without having to decrypt and without getting any additiocal information. The construction of [GPSW06] is very general as it supports any predicate that can be expressed as a circuit with threshold gates. On the other hand the construction only achieved what is called *payload security* which consists in guaranteeing the security of the cleartext. Indeed, in the construction of [GPSW06], the attribute vector associated with a ciphertext appears in clear in the ciphertext.

In several applications instead one would like to be able to encrypt a cleartext and label the ciphertext with attributes so that both the cleartext and the attributes are secure. This extra property is called *attribute hiding.* Indeed, it is an important research problem to design encryption schemes for large predicate classes that enjoy both the payload and the attribute hiding property. In [BW07], Boneh and Waters give construction for encryption schemes for several families of predicates including conjuctions, and subset and range predicates. This has been recently extended to disjunctions, polynomial equations and inner products [KSW08]. Both constructions are based on hardness assumptions regarding bilinear groups on *composite order.* More efficient schemes for range queries over encrypted data have been presented in [SBC$^+$07].

Our results. In this paper we give a construction for *hidden vector encryption* schemes (HVE, in short). Roughly speaking, in a hidden vector encryption scheme ciphertexts are associated with binary vectors and private keys are associated with with binary vectors with "don't care" entries (denoted by $\star$). A private key can decipher a ciphertext if all entries of the key vector that are not $\star$ agree with the corresponding entries of the ciphertext vector (see Definition 1). The first construction for HVE has been given by [BW07] which also showed that HVE gives efficient encryption schemes supporting conjunctions of equality queries, range queries and subset queries. By applying the reductions of [BW07] to our construction we obtain encryption schemes supporting the same classe of predicates as [BW07].

Both the payload and the attribute security of our construction rely on standard computational assumptions on bilinear groups of *prime* order; namely, the Bilinear Decision Diffie-Hellman assumption and the Decision Linear assumption (used also in [BW06,GPSW06]). As already noted above, the security of the construction of [BW07] instead relies on the Composite Bilinear Decision Diffie-Hellman assumption and the Composite 3-Party Diffie-Hellman assumption. Both assumptions imply that the order of the group is difficult to factor and this results in larger group elements and thus more expensive operations.

2 The Symmetric Bilinear Setting

We have multiplicative groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p and a non-degenerate bilinear pairing function $\mathbf{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. That is, for all $g \in \mathbb{G}, g \neq 1$, we have $\mathbf{e}(g, g) \neq 1$ and $\mathbf{e}(g^a, g^b) = \mathbf{e}(g, g)^{ab}$. We denote by g and $\mathbf{e}(g, g)$ the generators of $\mathbb{G}$ and $\mathbb{G}_T$. We call a *symmetric bilinear* instance a tuple $\mathcal{I} = [p, \mathbb{G}, \mathbb{G}_T, g, \mathbf{e}]$ and assume that there exists an efficient generation procedure that, on input security parameter 1^k, outputs an instance with $|p| = \Theta(k)$.

In our constructions we make the following hardness assumptions.

Decision BDH. Given a tuple $[g, g^{z_1}, g^{z_2}, g^{z_3}, Z]$ for random exponents $z_1, z_2, z_3 \in \mathbb{Z}_p$ it is hard to distinguish between $Z = \mathbf{e}(g, g)^{z_1 z_2 z_3}$ and a random Z from $\mathbb{G}_T$. More specifically, for an algorithm $\mathcal{A}$ we define experiment $\mathsf{DBDHExp}_{\mathcal{A}}$ as follows.

```
$\mathsf{DBDHExp}^{\mathcal{A}}(1^k)$
Choose instance $\mathcal{I} = [p, \mathbb{G}, \mathbb{G}_T, g, \mathbf{e}]$ with security parameter $1^k$;
Choose $a, b, c \in \mathbb{Z}_p$ at random;
Choose $\eta \in \{0, 1\}$ at random;
if $\eta = 1$ then choose $z \in \mathbb{Z}_p$ at random
    else set $z = abc$;
set $A = g^a, B = g^b, C = g^c$ and $Z = \mathbf{e}(g, g)^z$;
let $\eta' = \mathcal{A}(\mathcal{I}, A, B, C, Z)$;
if $\eta = \eta'$ then return 0 else return 1;
```

Assumption 1 (Decision Bilinear Diffie-Hellman) *For all probabilistic polynomial-time algorithms $\mathcal{A}$,*

$$\left|\mathrm{Prob}[\mathsf{DBDHExp}^{\mathcal{A}}(1^k) = 1] - 1/2\right| = \nu(k)$$

for some negligible function ν.

Decision Linear. Given a tuple $[g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^u, Z]$ for random exponents $z_1, z_2, z_3, u \in \mathbb{Z}_p$ it is hard to distinguish between $Z = g^{z_2(u-z_3)}$ and a random Z from $\mathbb{G}$. More specifically, for an algorithm $\mathcal{A}$ we define experiment $\mathsf{DLExp}_{\mathcal{A}}$ as follows.

```
$\mathsf{DLExp}^{\mathcal{A}}(1^k)$
Choose instance $\mathcal{I} = [p, \mathbb{G}, \mathbb{G}_T, g, \mathbf{e}]$ with security parameter $1^k$;
Choose $z_1, z_2, z_3, u \in \mathbb{Z}_p$ at random;
Choose $\eta \in \{0, 1\}$ at random;
if $\eta = 1$ then choose $z \in \mathbb{Z}_p$ at random
    else set $z = z_2(u - z_3)$;
set $Z_1 = g^{z_1}, Z_2 = g^{z_2}, Z_{13} = g^{z_1 z_3}, U = g^u$, and $Z = g^z$;
let $\eta' = \mathcal{A}(\mathcal{I}, Z_1, Z_2, Z_{13}, U, Z)$;
if $\eta = \eta'$ then return 0 else return 1;
```

Assumption 2 (Decision Linear) *For all probabilistic polynomial-time algorithms $\mathcal{A}$,*

$$\left|\mathrm{Prob}[\mathsf{DLExp}^{\mathcal{A}}(1^k) = 1] - 1/2\right| = \nu(k)$$

for some negligible function ν.

Note that Decision Linear implies Decision Bilinear Diffie-Hellman and the Decision Linear assumption has been introduced in [BBS04] and used also in [BW06].

3 HVE schemes

Let $\boldsymbol{x}$ be a string over the alphabet $\{0, 1\}$ and $\boldsymbol{y}$ be a string over the alphabet $\{0, 1, \star\}$. Assume $\boldsymbol{x}$ and $\boldsymbol{y}$ have the same length n and define predicate $P_{\boldsymbol{x}}(\boldsymbol{y})$ to be true if and only if for each $1 \leq i \leq n$ we have $x_i = y_i$ or $y_i = \star$. In other words, for $P_{\boldsymbol{x}}(\boldsymbol{y})$ to be true, the two strings must match in positions i where $y_i \neq \star$ and, intuitively, $\star$ is the "don't care" symbol.

Definition 1 (HVE). *A* Hidden Vector Encryption Scheme *(a* HVE scheme*) is a quadruple of probabilistic polynomial-time algorithms* (Setup, Enc, KeyGeneration, Dec) *such that:*

1. Setup *takes as input the security parameter 1^k and the* attribute length $n = \mathsf{poly}(k)$ *and outputs the* master public key Pk *and the* master secret key Msk.
2. KeyGeneration *takes as input the master secret key* Msk *and string $\boldsymbol{y} \in \{0, 1, \star\}^n$ and outputs the decryption key $K_{\boldsymbol{y}}$ associated with $\boldsymbol{y}$.*
3. Enc *takes as input the public key* Pk*,* attribute string $\boldsymbol{x} \in \{0, 1\}^n$ *and message M from the associated message space and returns ciphertext* $\mathsf{Ct}_{\boldsymbol{x}}$.
4. Dec *takes as input a secret key $K_{\boldsymbol{y}}$ and a ciphertext* $\mathsf{Ct}_{\boldsymbol{x}}$ *and outputs a message M.*

We require that for all k and $n = \mathsf{poly}(k)$, and for all strings $\boldsymbol{x} \in \{0, 1\}^n$ and $\boldsymbol{y} \in \{0, 1, \star\}^n$ such that $P_{\boldsymbol{x}}(\boldsymbol{y}) = 1$, it holds that:

$$\mathrm{Prob}[(\mathsf{Pk}, \mathsf{Msk}) \leftarrow \mathsf{Setup}(1^k, n); \quad K_{\boldsymbol{y}} \leftarrow \mathsf{KeyGeneration}(\mathsf{Msk}, \boldsymbol{y}); \quad \mathsf{Ct}_{\boldsymbol{x}} \leftarrow \mathsf{Enc}(\mathsf{Pk}, \boldsymbol{x}, M) : \mathsf{Dec}(K_{\boldsymbol{y}}, \mathsf{Ct}) = M] = 1.$$

We define two notions of security for our HVE scheme: semantic security that captures the payload-hiding property and the attribute hiding property that guarantees security of the attribute string. Both notions are in the selective models in which the adversary committs to the attribute vector at the beginning of the game. We note that this is same notion of security used in [BW07,KSW08].

Definition 2 (Semantic Security). *An HVE scheme* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGeneration}, \mathsf{Dec})$ *is* semantically secure *if for all PPT adversaries* $\mathcal{A}$*,*

$$\left|\mathrm{Prob}[\mathsf{SemanticExp}_{\mathcal{A}}(1^k) = 1] - 1/2\right| = \nu(k)$$

for some negligible function ν*, where* $\mathsf{SemanticExp}_{\mathcal{A}}(1^k)$ *is the following experiment.*

Init. *The adversary* $\mathcal{A}$ *announces the vector* $\boldsymbol{x}$ *it wishes be challenged upon.*
Setup. *The public and the secret key* $(\mathsf{Msk}, \mathsf{Pk})$ *are generated using the* Setup *procedure and* $\mathcal{A}$ *receives* Pk*.*
Query Phase I. $\mathcal{A}$ *requests and gets private keys* $K_{\boldsymbol{y}}$ *relative to vectors* $\boldsymbol{y}$ *such that* $P_{\boldsymbol{x}}(\boldsymbol{y}) = 0$*. Key* $K_{\boldsymbol{y}}$ *is computed using the* $\mathsf{KeyGeneration}$ *procedure.*
Challenge. $\mathcal{A}$ *returns two different messages* M_0, M_1 *of the same length in the message space.* η *is chosen at random from* $\{0,1\}$*.* $\mathcal{A}$ *is given ciphertext* $\mathsf{Ct}_{\boldsymbol{x}} \leftarrow \mathsf{Enc}(\mathsf{Pk}, \boldsymbol{x}, M_\eta)$*.*
Query Phase II. *Identical to Query Phase I.*
Output. $\mathcal{A}$ *returns* η'*. If* $\eta = \eta'$ *then return* 1 *else return* 0*.*

We are now ready to define the notion of attribute hiding.

Definition 3. *An HVE scheme* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGeneration}, \mathsf{Dec})$ *is* attribute hiding *if for all PPT adversaries* $\mathcal{A}$*,*

$$\left|\mathrm{Prob}[\mathsf{AttributeHidingExp}_{\mathcal{A}}(1^k) = 1] - 1/2\right| = \nu(k)$$

for some negligible function ν*, where* $\mathsf{AttributeHidingExp}_{\mathcal{A}}(1^k)$ *is the following experiment.*

Init. *The adversary* $\mathcal{A}$ *announces two attribute strings* $\boldsymbol{x}_0 \neq \boldsymbol{x}_1$ *it wishes be challenged upon.*
Setup. *The public and the secret key* $(\mathsf{Msk}, \mathsf{Pk})$ *are generated using the* Setup *procedure and* $\mathcal{A}$ *receives* Pk*.*
Query Phase I. $\mathcal{A}$ *requests and gets private keys* $K_{\boldsymbol{y}}$ *relative to vectors* $\boldsymbol{y}$ *such that* $P_{\boldsymbol{x}_1}(\boldsymbol{y}) = P_{\boldsymbol{x}_2}(\boldsymbol{y}) = 0$*. Key* $K_{\boldsymbol{y}}$ *is computed using the* $\mathsf{KeyGeneration}$ *procedure.*
Challenge. $\mathcal{A}$ *returns two different messages* M_0, M_1 *of the same length.* η *is chosen at random from* $\{0,1\}$*.* $\mathcal{A}$ *is given ciphertext* $\mathsf{Ct}_{\boldsymbol{x}} \leftarrow \mathsf{Enc}(\mathsf{Pk}, \boldsymbol{x}_\eta, M_\eta)$*.*
Query Phase II. *Identical to Query Phase I.*
Output. $\mathcal{A}$ *returns* η'*. If* $\eta = \eta'$ *then return* 1 *else return* 0*.*

If in the previous experiment we let $\boldsymbol{x}_0 = \boldsymbol{x}_1$ we have the definition of Semantic Security.

4 Our construction

In this section we describe our construction for an HVE scheme.

Setup. Procedure Setup, on input security parameter 1^k and attribute length $n = \mathsf{poly}(k)$, computes the public key Pk and the master secret key Msk in the following way.

Choose a random instance $\mathcal{I} = [p, \mathbb{G}, \mathbb{G}_T, g, \mathbf{e}]$.

Choose y at random in $\mathbb{Z}_p$ and set $Y = \mathbf{e}(g, g)^y$.

For $1 \leq i \leq n$, choose t_i, v_i, r_i, m_i at random in $\mathbb{Z}_p$ and set $T_i = g^{t_i}, V_i = g^{v_i}$ and $R_i = g^{r_i}, M_i = g^{m_i}$.

Then, $\mathsf{Setup}(1^k, n)$ returns $[\mathsf{Pk}, \mathsf{Msk}]$ where

$$\mathsf{Pk} = [\mathcal{I}, Y, (T_i, V_i, R_i, M_i)_{i=1}^n] \text{ and } \mathsf{Msk} = [y, (t_i, v_i, r_i, m_i)_{i=1}^n].$$

Encryption. Procedure Enc takes as input cleartext $M \in \mathbb{G}_T$, attribute string $\boldsymbol{x}$ and public key Pk and computes ciphertext as follows.

Choose s at random in $\mathbb{Z}_p$, and, for $1 \leq i \leq n$, choose s_i at random in $\mathbb{Z}_p$ and compute ciphertext

$$\mathsf{Enc}(\mathsf{Pk}, \boldsymbol{x}, M) = [\Omega, C_0, (X_i, W_i)_{i=1}^n],$$

where $\Omega = M \cdot Y^{-s}$, $C_0 = g^s$ and

$$X_i = \begin{cases} T_i^{s-s_i}, & \text{if } x_i = 1; \\ R_i^{s-s_i}, & \text{if } x_i = 0. \end{cases} \quad \text{and} \quad W_i = \begin{cases} V_i^{s_i}, & \text{if } x_i = 1; \\ M_i^{s_i}, & \text{if } x_i = 0. \end{cases}$$

Key Generation. Procedure $\mathsf{KeyGeneration}$ on input Msk and $\boldsymbol{y} \in \{0, 1, \star\}^n$ derives private key $K_{\boldsymbol{y}}$ relative to attribute string $\boldsymbol{y}$ in the following way.

If $\boldsymbol{y} = (\star, \star, \ldots, \star)$ then $K_{\boldsymbol{y}} = g^y$. Else, denote by $S_{\boldsymbol{y}}^1$ and $S_{\boldsymbol{y}}^0$ the set of indices i for which $y_i = 1$ and $y_i = 0$, respectively and let $S_{\boldsymbol{y}} = S_{\boldsymbol{y}}^1 \cup S_{\boldsymbol{y}}^0$ be the set of indices for $y_i \neq \star$. Then, for $i \in S_{\boldsymbol{y}}$, choose a_i at random in $\mathbb{Z}_p$ under the constraint that $\sum_{i \in S_{\boldsymbol{y}}} a_i = y$ and let $K_{\boldsymbol{y}} = (Y_i, L_i)_{i=1}^n$, where

$$Y_i = \begin{cases} g^{\frac{a_i}{t_i}}, & \text{if } y_i = 1; \\ g^{\frac{a_i}{r_i}}, & \text{if } y_i = 0; \\ \emptyset, & \text{if } y_i = \star. \end{cases} \quad \text{and} \quad L_i = \begin{cases} g^{\frac{a_i}{v_i}}, & \text{if } y_i = 1; \\ g^{\frac{a_i}{m_i}}, & \text{if } y_i = 0; \\ \emptyset, & \text{if } y_i = \star. \end{cases}$$

Decryption. Procedure Dec decrypts cyphertext $\mathsf{Ct}_{\boldsymbol{x}}$ using secret key $K_{\boldsymbol{y}}$ such that $P_{\boldsymbol{x}}(\boldsymbol{y}) = 1$.

$$\mathsf{Dec}(\mathsf{Pk}, \mathsf{Ct}_{\boldsymbol{x}}, K_{\boldsymbol{y}}) = \Omega \cdot \prod_{i \in S_{\boldsymbol{y}}} \mathbf{e}(X_i, Y_i)\mathbf{e}(W_i, L_i)$$

where $S_{\boldsymbol{y}}$ is the set of indices i such that $y_i \neq \star$. If $S_{\boldsymbol{y}} = \emptyset$ then $K_{\boldsymbol{y}} = g^y$ and

$$\mathsf{Dec}(\mathsf{Pk}, \mathsf{Ct}_{\boldsymbol{x}}, K_{\boldsymbol{y}}) = \Omega \cdot \mathbf{e}(C_0, K_{\boldsymbol{y}}).$$

This ends the description of our construction. We remark that our construction can be extended to attribute vectors taken from a larger alphabet Σ (and not simply $\{0,1\}$) without increasing the length of the ciphertexts and of the secret keys but only the length of the public key Pk. We omit further details.

We next prove that the quadruple is indeed an HVE.

Theorem 1. *The quadruple of algorithms* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGeneration}, \mathsf{Dec})$ *specified above is an HVE.*

Proof. It is sufficient to verify that this procedure computes M correctly when $P_{\boldsymbol{x}}(\boldsymbol{y}) = 1$. The case in which $\boldsymbol{y} = (\star, \star, \cdots, \star)$ is obvious.

We remind the reader that $S^1_{\boldsymbol{y}}$ (respectively, $S^0_{\boldsymbol{y}}$) denotes the (possibly empty) set of indices i such that $y_i = 1$ (respectively, $y_i = 0$) and that $S_{\boldsymbol{y}} = S^1_{\boldsymbol{y}} \cup S^0_{\boldsymbol{y}}$.

Then we have

$$\begin{aligned}
\mathsf{Dec}(\mathsf{Pk}, \mathsf{Ct}_{\boldsymbol{x}}, K_{\boldsymbol{y}}) &= \Omega \prod_{i \in S_{\boldsymbol{y}}} \mathbf{e}(X_i, Y_i)\mathbf{e}(W_i, L_i) \\
&= M\mathbf{e}(g,g)^{-ys} \cdot \prod_{i \in S^1_{\boldsymbol{y}}} \mathbf{e}(g^{t_i(s-s_i)}, g^{\frac{a_i}{t_i}})\mathbf{e}(g^{w_i s_i}, g^{\frac{a_i}{w_i}}) \\
&\quad \cdot \prod_{i \in S^0_{\boldsymbol{y}}} \mathbf{e}(g^{r_i(s-s_i)}, g^{\frac{a_i}{r_i}})\mathbf{e}(g^{m_i s_i}, g^{\frac{a_i}{m_i}}) \\
&= M\mathbf{e}(g,g)^{-ys} \prod_{i \in S^1_{\boldsymbol{y}}} \mathbf{e}(g,g)^{(s-s_i)a_i}\mathbf{e}(g,g)^{s_i a_i} \prod_{i \in S^0_{\boldsymbol{y}}} \mathbf{e}(g,g)^{(s-s_i)a_i}\mathbf{e}(g,g)^{s_i a_i} \\
&= M\mathbf{e}(g,g)^{-ys} \prod_{i \in S_{\boldsymbol{y}}} \mathbf{e}(g,g)^{(s-s_i)a_i}\mathbf{e}(g,g)^{s_i a_i} \\
&= M\mathbf{e}(g,g)^{-ys} \prod_{i \in S_{\boldsymbol{y}}} \mathbf{e}(g,g)^{s a_i} \\
&= M\mathbf{e}(g,g)^{-ys}\mathbf{e}(g,g)^{ys} = M.
\end{aligned}$$

Efficiency. In our construction we have that, for an attribute string of length n, the ciphertext contains 1 element from $\mathbb{G}_T$ and $O(n)$ elements from $\mathbb{G}$. The secret key corresponding to vector $\boldsymbol{y}$ instead contains $O(\mathsf{weight}(\boldsymbol{y}))$ elements from $\mathbb{G}$, where $\mathsf{weight}(\boldsymbol{y})$ is the number of entries of $\boldsymbol{y}$ that are either 0 or 1. Thus our scheme has the same ciphertext and key-length as the constructions presented in [KSW08,BW07].

5 Proofs

In this section we prove that our construction is semantically secure and attribute hiding.

Theorem 2 (Semantic Security). *Assume BDDH holds. Then HVE scheme* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGeneration}, \mathsf{Dec})$ *described above is semantically secure.*

Proof. Suppose that there exists PPT adversary $\mathcal{A}$ which has success in experiment $\mathsf{SemanticExp}$ with probability non-negligibly larger than 1/2. We then construct an adversary $\mathcal{B}$ for the experiment $\mathsf{DBDHExp}$ that uses $\mathcal{A}$ as subroutine.

Input. $\mathcal{B}$ receives in input $[\mathcal{I}, A = g^a, B = g^b, C = g^c, Z]$, where Z is $\mathbf{e}(g,g)^{abc}$ or a random element of $\mathbb{G}_T$.

Init. $\mathcal{B}$ runs $\mathcal{A}$ and receives the attribute string $\boldsymbol{x}$ it wishes to be challenged upon.

Setup. Set $Y = \mathbf{e}(A,B)$. For every $1 \le i \le n$, $\mathcal{B}$ chooses $t'_i, v'_i, r'_i, m'_i \in \mathbb{Z}_p$ at random and set

$$T_i = \begin{cases} g^{t'_i}, & \text{if } x_i = 1; \\ B^{t'_i}, & \text{if } x_i = 0; \end{cases} \quad \text{and} \quad V_i = \begin{cases} g^{v'_i}, & \text{if } x_i = 1; \\ B^{v'_i}, & \text{if } x_i = 0; \end{cases}$$

$$R_i = \begin{cases} B^{r'_i}, & \text{if } x_i = 1; \\ g^{r'_i}, & \text{if } x_i = 0; \end{cases} \quad \text{and} \quad M_i = \begin{cases} B^{m'_i}, & \text{if } x_i = 1; \\ g^{m'_i}, & \text{if } x_i = 0; \end{cases}$$

$\mathcal{B}$ runs $\mathcal{A}$ on input $\mathsf{Pk} = [\mathcal{I}, Y, (T_i, V_i, R_i, M_i)_{i=1}^n]$.

Notice that Pk has the same distribution of a public key received by $\mathcal{A}$ in the Setup phase of $\mathsf{SemanticExp}$ with $y = a \cdot b$, and with $t_i = t'_i$, $v_i = v'_i$, $r_i = br'_i$, and $m_i = bm'_i$ for i with $x_i = 1$, and $t_i = bt'_i$, $v_i = bv'_i$, $r_i = r'_i$, and $m_i = m'_i$ for i with $x_i = 0$.

Query Phase I. $\mathcal{B}$ answers $\mathcal{A}$'s queries for $\boldsymbol{y}$ such that $P_{\boldsymbol{x}}(\boldsymbol{y}) = 0$ as follows. Let j be an index where $x_j \neq y_j$ and $y_j \neq \star$ (such an index always exists). For every $i \neq j$ such that $y_i \neq \star$, choose a'_i uniformly at random in $\mathbb{Z}_p$ and let $a' = \sum a'_i$.

Set Y_j and L_j as

$$Y_j = \begin{cases} A^{1/t'_j} g^{-a'/t'_j}, & \text{if } \ y_j = 1; \\ A^{1/r'_j} g^{-a'/r'_j}, & \text{if } \ y_j = 0. \end{cases} \quad \text{and} \quad L_j = \begin{cases} A^{1/v'_j} g^{-a'/v'_j}, & \text{if } \ y_j = 1; \\ A^{1/m'_j} g^{-a'/m'_j}, & \text{if } \ y_j = 0. \end{cases}$$

and, for $i \neq j$, set Y_i, L_i as follows

$$Y_i = \begin{cases} B^{a'_i/t'_i}; & \text{if } \ x_i = y_i = 1; \\ B^{a'_i/r'_i}; & \text{if } \ x_i = y_i = 0; \\ g^{a'_i/r'_i}; & \text{if } \ x_i = 1 \text{ and } \ y_i = 0; \\ g^{a'_i/t'_i}; & \text{if } \ x_i = 0 \text{ and } \ y_i = 1; \\ \emptyset; & \text{if } \ y_i = \star. \end{cases} \quad \text{and} \quad L_i = \begin{cases} B^{a'_i/v'_i}; & \text{if } \ x_i = y_i = 1; \\ B^{a'_i/m'_i}; & \text{if } \ x_i = y_i = 0; \\ g^{a'_i/m'_i}; & \text{if } \ x_i = 1 \text{ and } \ y_i = 0; \\ g^{a'_i/v'_i}; & \text{if } \ x_i = 0 \text{ and } \ y_i = 1; \\ \emptyset; & \text{if } \ y_i = \star. \end{cases}$$

Notice that $K_{\boldsymbol{y}}$ has the same distribution of the key returned by the $\mathsf{KeyGeneration}$ procedure. In fact, for $i \neq j$, set $a_i = ba'_i$ and set $a_j = b(a - a')$. Then we have that $\sum_{i \in S_{\boldsymbol{y}}} a_i = y$. Moreover, if $y_i = 1$ then $Y_i = g^{\frac{a_i}{t_i}}$ and $L_i = g^{\frac{a_i}{v_i}}$ and, if $y_i = 0$ then $Y_i = g^{\frac{a_i}{r_i}}$ and $L_i = g^{\frac{a_i}{m_i}}$.

Challenge. $\mathcal{A}$ returns two messages $M_0, M_1 \in \mathbb{G}_T$.

$\mathcal{B}$ chooses uniformly at random $\eta \in \{0,1\}$ and $s_i \in \mathbb{Z}_p$, for $i = 1, \cdots, n$. Then $\mathcal{B}$ constructs $\mathsf{Ct}_{\boldsymbol{x}} = (\varOmega, C, (X_i, W_i)_{i=1}^n)$, where $\varOmega = M_\eta Z^{-1}$, $C_0 = C$ and

$$X_i = \begin{cases} C^{t'_i} g^{-t'_i s_i}; & \text{if } x_i = 1; \\ C^{r'_i} g^{-r'_i s_i}; & \text{if } x_i = 0. \end{cases} \quad \text{and} \quad W_i = \begin{cases} g^{-v'_i s_i}; & \text{if } x_i = 1; \\ g^{-m'_i s_i}; & \text{if } x_i = 0. \end{cases}$$

Observe that if $Z = \mathbf{e}(g,g)^{abc}$ then $\mathsf{Ct}_{\boldsymbol{x}}$ is an encryption of M_η with $s = c$. If instead Z is random in $\mathbb{G}_T$ then $\mathsf{Ct}_{\boldsymbol{x}}$ is independent from η.

Query Phase II. Identical to Query Phase I.

Output. $\mathcal{A}$ outputs η'. $\mathcal{B}$ returns 0 iff $\eta' = \eta$.

To conclude the proof observe that, if $Z = \mathbf{e}(g,g)^{abc}$ then, since $\mathcal{A}$ is a successful adversary for semantic security, the probability that $\mathcal{B}$ returns 0 is at least $1/2 + 1/\mathsf{poly}(k)$. On the other hand if Z is random in $\mathbb{G}_T$ the probability that $\mathcal{B}$ returns 0 is at most $1/2$. This contradicts the BDDH assumption.

We now turn our attention at the attribute hiding property. We stress that a crucial tool in achieving this property is the "linear splitting" technique first used to construct anonymous hierarchical identity-based encryption in [BW06]. As an effect of employing this technique our ciphertexts and keys roughly double in sizes. If one does not require attribute hiding then our scheme can be modified so that, for attribute vectors of length n, the ciphertext has $n+2$ elements and keys at most n elements.

To prove that the HVE scheme presented is attribute hiding we show that for any attribute string $\boldsymbol{x}$ and for any message M, an encryption of M with respect to attribute string $\boldsymbol{x}$ is computationally indistinguishable from the uniform distribution on $\mathbb{G}_T \times \mathbb{G}^{2n+1}$ to an adversary that has access to the key generation procedure for $\boldsymbol{y}$ such that $P_{\boldsymbol{x}}(\boldsymbol{y}) = 0$.

Specifically, for $j = 0, 1, \ldots, n$, we denote by $\mathsf{Dist}_j(\boldsymbol{x}, M)$ the following distribution.

```
$\mathsf{Dist}_j(\boldsymbol{x}, M)$
 1. choose $\mathcal{I} = [p, \mathbb{G}, \mathbb{G}_T, g, \mathbf{e}]$ with security parameter $1^k$;
 2. compute $[\mathsf{Msk}, \mathsf{Pk}]$ by executing $\mathsf{Setup}(1^k, n)$;
 3. choose $R_0$ uniformly at random from $\mathbb{G}_T$ and $s$ uniformly at random from
    $\mathbb{Z}_p$; set $C_0 = g^s$;
 4. for $i = 1, \cdots, j$ choose $X_i, W_i$ uniformly at random from $\mathbb{G}$;
 5. for $i = j+1, \cdots, n$
    choose $s_i$ uniformly at random $\mathbb{Z}_p$ and set

      $X_i = \begin{cases} T_i^{s-s_i}, & \text{if } x_i = 1; \\ R_i^{s-s_i}, & \text{if } x_i = 0. \end{cases}$ and $W_i = \begin{cases} V_i^{s_i}, & \text{if } x_i = 1; \\ M_i^{s_i}, & \text{if } x_i = 0. \end{cases}$

 6. return: $(R_0, C_0, (X_i, W_i)_{i=1}^n)$;
```

From the proof of semantic security it follows, that under the BDDH, distribution $\mathsf{Dist}_0(\boldsymbol{x}, M)$ is indistinguishable from the distribution of the legal ciphertexts $\mathsf{Enc}(\mathsf{Pk}, \boldsymbol{x}, M)$ of M with attribute string $\boldsymbol{x}$. Moreover, for all j, $\mathsf{Dist}_j(\boldsymbol{x}, M)$

is independent from M and $\mathsf{Dist}_n(\boldsymbol{x}, M)$ is the uniform distribution on $\mathbb{G}_T \times \mathbb{G}^{2n+1}$ and thus is independent from $\boldsymbol{x}$. Next lemma shows that distributions $\mathsf{Dist}_{\ell-1}$ and Dist_ℓ are computational indistinguishable even to an adversary that has access to the key generation oracle. This concludes the proof of the attribute hiding property.

Lemma 1. *Under the DL assumption, for $\ell = 1, 2, \ldots, n$ and for any $\boldsymbol{x} \in \{0,1\}^n$, we have that distributions $\mathsf{Dist}_{\ell-1}(\boldsymbol{x})$ and $\mathsf{Dist}_\ell(\boldsymbol{x})$ are computationally indistinguishable to an adversary that has access to the key generation oracle.*

Proof. Suppose that there exists PPT adversary $\mathcal{A}$ which distinguishes $\mathsf{Dist}_{\ell-1}$ from Dist_ℓ. We then construct an adversary $\mathcal{B}$ for the experiment DLExp.

Input. $\mathcal{B}$ takes in input $[\mathcal{I}, Z_1 = g^{z_1}, Z_2 = g^{z_2}, Z_{13} = g^{z_1 z_3}, U = g^u, Z]$, where either $Z = g^{z_2(u-z_3)}$ or Z is a random element of $\mathbb{G}$.

Init. $\mathcal{B}$ receives from $\mathcal{A}$ the attribute string $\boldsymbol{x}$ it wishes to be challenged upon.

Setup. $\mathcal{B}$ sets $Y = \mathbf{e}(Z_1, Z_2)$ and, for $i = 1 \cdots n$, $\mathcal{B}$ chooses t'_i, v'_i, r'_i, m'_i uniformly at random from $\mathbb{Z}_p$ and sets

$$T_\ell = \begin{cases} Z_2^{t'_\ell}, & \text{if } x_\ell = 1; \\ Z_1^{t'_\ell}, & \text{if } x_\ell = 0; \end{cases} \quad \text{and} \quad V_\ell = \begin{cases} Z_1^{v'_\ell}, & \text{if } x_\ell = 1; \\ Z_1^{v'_\ell}, & \text{if } x_\ell = 0; \end{cases}$$

$$R_\ell = \begin{cases} Z_1^{r'_\ell}, & \text{if } x_\ell = 1; \\ Z_2^{r'_\ell}, & \text{if } x_\ell = 0; \end{cases} \quad \text{and} \quad M_\ell = \begin{cases} Z_1^{m'_\ell}, & \text{if } x_\ell = 1; \\ Z_1^{m'_\ell}, & \text{if } x_\ell = 0; \end{cases}$$

Moreover, for $i \neq \ell$, $\mathcal{B}$ sets

$$T_i = \begin{cases} g^{t'_i}, & \text{if } x_i = 1; \\ Z_1^{t'_i}, & \text{if } x_i = 0; \end{cases} \quad \text{and} \quad V_i = \begin{cases} g^{v'_i}, & \text{if } x_i = 1; \\ Z_1^{v'_i}, & \text{if } x_i = 0; \end{cases}$$

$$R_i = \begin{cases} Z_1^{r'_i}, & \text{if } x_i = 1; \\ g^{r'_i}, & \text{if } x_i = 0; \end{cases} \quad \text{and} \quad M_i = \begin{cases} Z_1^{m'_i}, & \text{if } x_i = 1; \\ g^{m'_i}, & \text{if } x_i = 0; \end{cases}$$

$\mathcal{B}$ runs $\mathcal{A}$ on input $\mathsf{Pk} = [\mathcal{I}, Y, (T_i, V_i, R_i, M_i)_{i=1}^n]$.

Notice that Pk has the same distribution of a public key computed using $\mathsf{KeyGeneration}$, with $y = z_1 \cdot z_2$, and $t_i = t'_i, v_i = v'_i, r_i = z_1 r'_i, m_i = z_1 m'_i$ for $i \neq \ell$ with $x_i = 1$, and $t_i = z_1 t'_i, v_i = z_1 v'_i, r_i = r'_i, m_i = m'_i$ for $i \neq \ell$ with $x_i = 0$; moreover, if $x_\ell = 1$ then we have $t_\ell = z_2 t'_\ell, v_\ell = z_1 v'_\ell, r_\ell = z_1 r'_\ell, m_\ell = z_1 m'_\ell$ whereas, if $x_\ell = 0$, we have $t_\ell = z_1 t'_\ell, v_\ell = z_1 v'_\ell, r_\ell = z_2 r'_\ell, m_\ell = z_1 m'_\ell$.

Query Phase I. $\mathcal{B}$ answers $\mathcal{A}$'s queries for $\boldsymbol{y}$ such that $P_{\boldsymbol{x}}(\boldsymbol{y}) = 0$ in the following way. We distinguish two cases.

Case 1: $x_\ell = y_\ell$ or $y_\ell = \star$. In this case there exists index $j \neq \ell$ such that $x_j \neq y_j$ and $y_j \neq \star$.

Then, for $i \neq j$ B chooses a'_i uniformly at random in $\mathbb{Z}_p$ and let us denote by a' the sum $a' = \sum_{i \neq j, \ell} a'_i$.

For $i \neq j$ and $i \neq \ell$, $\mathcal{B}$ sets

$$Y_i = \begin{cases} Z_1^{a'_i/t'_i}, & \text{if } \; x_i = y_i = 1; \\ Z_1^{a'_i/r'_i}, & \text{if } \; x_i = y_i = 0; \\ g^{a'_i/r'_i}, & \text{if } \; x_i = 1, y_i = 0; \\ g^{a'_i/t'_i}, & \text{if } \; x_i = 0, y_i = 1; \\ \emptyset, & \text{if } \; y_i = \star. \end{cases} \quad \text{and} \quad L_i = \begin{cases} Z_1^{a'_i/v'_i}, & \text{if } \; x_i = y_i = 1; \\ Z_1^{a'_i/m'_i}, & \text{if } \; x_i = y_i = 0; \\ g^{a'_i/m'_i}, & \text{if } \; x_i = 1, y_i = 0; \\ g^{a'_i/v'_i}, & \text{if } \; x_i = 0, y_i = 1; \\ \emptyset, & \text{if } \; y_i = \star. \end{cases}$$

Moreover, $\mathcal{B}$ sets

$$Y_\ell = \begin{cases} Z_1^{a'_\ell/t'_\ell}, & \text{if } \; y_\ell = 1; \\ Z_1^{a'_\ell/r'_\ell}, & \text{if } \; y_\ell = 0; \\ \emptyset, & \text{if } \; y_\ell = \star. \end{cases} \quad \text{and} \quad L_\ell = \begin{cases} Z_2^{a'_\ell/v'_\ell}, & \text{if } \; y_\ell = 1; \\ Z_2^{a'_\ell/m'_\ell}, & \text{if } \; y_\ell = 0; \\ \emptyset, & \text{if } \; y_\ell = \star. \end{cases}$$

Finally, $\mathcal{B}$ sets

$$Y_j = \begin{cases} Z_2^{(1-a'_\ell)/t'_j} g^{-a'/t'_j}, & \text{if } \; y_j = 1; \\ Z_2^{(1-a'_\ell)/r'_j} g^{-a'/r'_j}, & \text{if } \; y_j = 0. \end{cases} \quad \text{and} \quad L_j = \begin{cases} Z_2^{(1-a'_\ell)/v'_j} g^{-a'/v'_j}, & \text{if } \; y_j = 1; \\ Z_2^{(1-a'_\ell)/m'_j} g^{-a'/m'_j}, & \text{if } \; y_j = 0. \end{cases}$$

By the settings above we have that, for $i \neq j$ and $i \neq \ell$, $a_i = z_1 a'_i$, $a_\ell = z_1 z_2 a'_\ell$ and $a_j = z_1 z_2 - z_1 z_2 a'_\ell - z_1 a'$. Therefore, the a_i's are independently and randomly chosen in $\mathbb{Z}_p$ under the costraint that their sum is $z_1 z_2 = y$ and thus the key computed by $\mathcal{B}$ has the exact same distribution as the key computed by the KeyGeneration algorithm.

Case 2: $x_\ell \neq y_\ell$ and $y_\ell \neq \star$. In this case, for $i \neq \ell$, $\mathcal{B}$ chooses a'_i uniformly at random in $\mathbb{Z}_p$ and let us denote by a' the sum $a' = \sum_{i \neq \ell} a'_i$. Then for $i \neq \ell$, $\mathcal{B}$ sets Y_i and L_i exactly as in the previous case, whereas, $\mathcal{B}$ sets Y_ℓ and L_ℓ as follows

$$Y_\ell = \begin{cases} Z_2^{1/r'_\ell} g^{-a'/r'_\ell}, & \text{if } \; x_\ell = 1; \\ Z_2^{1/t'_\ell} g^{-a'/t'_\ell}, & \text{if } \; x_\ell = 0; \end{cases} \quad \text{and} \quad L_\ell = \begin{cases} Z_2^{1/m'_\ell} g^{-a'/m'_\ell}, & \text{if } \; x_\ell = 1; \\ Z_2^{1/v'_\ell} g^{-a'/v'_\ell}, & \text{if } \; x_\ell = 0; \end{cases}$$

By the settings above we have that $a_i = z_1 a'_i$ and $a_\ell = z_1 z_2 - z_1 a'$. Therefore, the a_i's are independently and randomly chosen in $\mathbb{Z}_p$ under the costraint that their sum is $z_1 z_2 = y$. Hence, also in this case, the key computed by $\mathcal{B}$ has the exact same distribution as the key returned by the KeyGeneration algorithm.

Challenge. $\mathcal{B}$ chooses R_0 uniformly at random $\mathbb{G}_T$ and, for $\ell \leq i \leq n$, chooses s'_i uniformly at random in $\mathbb{Z}_p$. $\mathcal{B}$ then constructs the tuple

$$D^* = (R_0, C_0, (X_i, W_i)_{i=1}^n)$$

where $C_0 = U$, and, for $i < \ell$, X_i and W_i are chosen uniformly from $\mathbb{G}$ whereas, for $i \geq \ell$, $\mathcal{B}$ computes

$$X_i = \begin{cases} Z^{t'_l}, & \text{if } \; i = \ell, x_i = 1; \\ Z^{r'_l}, & \text{if } \; i = \ell, x_i = 0; \\ U^{t'_i} g^{-t'_i s'_i}, & \text{if } \; i > \ell, x_i = 1; \\ U^{r'_i} g^{-r'_i s'_i}, & \text{if } \; i > \ell, x_i = 0; \end{cases} \quad \text{and} \quad W_i = \begin{cases} Z_{13}^{v'_l}, & \text{if } \; i = \ell, x_i = 1; \\ Z_{13}^{m'_l}, & \text{if } \; i = \ell, x_i = 0; \\ g^{v'_i s'_i}, & \text{if } \; i > \ell, x_i = 1; \\ g^{m'_i s'_i}, & \text{if } \; i > \ell, x_i = 0; \end{cases}$$

Now observe that if $Z = g^{z_2(u-z_3)}$ then $D^\star$ is distributed according to $\mathsf{Dist}_{\ell-1}(\boldsymbol{x})$, $s = u, s_\ell = z_3$, and $s_i = s'_i$ for $i > \ell$. On the other hand, if Z is random in $\mathbb{G}$, then $D^\star$ is distributed according to $\mathsf{Dist}_\ell(\boldsymbol{x})$ with $s = u$ and $s_i = s'_i$ for $i > \ell$.

Query Phase II. Identical to Query Phase I.

Output. $\mathcal{A}$ outputs η which represents a guess for the tuple in input ($\eta = 0$ for $D_{\ell-1}$ and $v = 1$ for D_ℓ). $\mathcal{B}$ forwards the same bit as its guess for the tuple of the experiment DLExp.

By the observation above, we observe that if $Z = g^{z_2(u-z_3)}$ then $\mathcal{A}$'s view is exactly the same as $\mathcal{A}$'s view (including the answers for queries for private keys) when receiving an input from $\mathsf{Dist}_{\ell-1}(\boldsymbol{x}, M)$; if Z is randomly and uniformly distributed in $\mathbb{G}$ then $\mathcal{A}$'s view (again this includes the replies obtained to the queries for private keys) is the same as when receiving an input from $\mathsf{Dist}_\ell(\boldsymbol{x}, M)$. Therefore, if $\mathcal{A}$ distinguishes between Dist_ℓ and $\mathsf{Dist}_{\ell-1}$ then the DL assumption is broken.

The above lemma implies the following theorem.

Theorem 3 (Attribute Hiding). *Assume DL holds. Then HVE scheme* ($\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGeneration}, \mathsf{Dec}$) *described above is attribute hiding.*

6 Applications

As we have discussed in the introduction HVE schemes are a special type of *predicate encryption schemes.*

Definition 4. *A* predicate encryption scheme *for a class $\mathcal{F}$ of predicates over n-bit strings is quadruple of probabilistic polynomial-time algorithms* ($\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGeneration}, \mathsf{Dec}$) *such that:*

1. Setup *takes as input the security parameter* 1^k *and attribute length* $n = \mathsf{poly}(k)$ *and outputs the* master public key Pk *and the* master secret key Msk.
2. $\mathsf{KeyGeneration}$ *takes as input the master secret key* Msk *and a predicate* $f \in \mathcal{F}$ *and outputs the decryption key* K_f *associated with* f.
3. Enc *takes as input the public key* Pk *and an* attribute string $\boldsymbol{x} \in \{0,1\}^n$ *and a message* M *in some associated message space and returns ciphertext* $\mathsf{Ct}_{\boldsymbol{x}}$.
4. Dec *takes as input a secret key* K_f *and a ciphertext* $\mathsf{Ct}_{\boldsymbol{x}}$ *and outputs a message* M.

We require that for all k and $n = \mathsf{poly}(k)$, and for all strings $\boldsymbol{x} \in \{0,1\}^n$ and predicates $f \in \mathcal{F}$ such that $f(\boldsymbol{x}) = 1$, it holds that:

$$\text{Prob}[(\mathsf{Pk}, \mathsf{Msk}) \leftarrow \mathsf{Setup}(1^k, n); \quad K_f \leftarrow \mathsf{KeyGeneration}(\mathsf{Msk}, f); \quad \mathsf{Ct}_{\boldsymbol{x}} \leftarrow \mathsf{Enc}(\mathsf{Pk}, \boldsymbol{x}, M) : \mathsf{Dec}(K_f, \mathsf{Ct}) = M] = 1.$$

The construction of searchable encryption of [BDOP04] can be seen an predicate encryption for the class $\mathcal{F}$ of predicates P_a defined as $P_a(x) = 1$ iff and only if $a = x$.

In [BW07], it is shown that HVE scheme can be used to construct predicate encryption for the class of *conjunctive comparison predicates* defined as follows $P_{a_1,\cdots,a_n}(x_1,\cdots,x_n) = 1$ if and only if $a_i \leq x_i$ for all i. Futhermore, in [BW07] it was shown how to construct predicate encryption schemes also for conjunctive range query predicates and subset query predicates starting from HVE. All reductions can be applied to our HVE thus yielding the following theorem.

Theorem 4. *Assume DL holds. Then there exist predicate encryption schemes for conjunctive comparison predicates, conjunctive range query predicates and subset query predicates that are semantically secure and attribute hiding.*

We expect there to be several other applications of HVE.

Acknowledgements. The work of the authors has been supported in part by the European Commission through Contract IST-2002-507932 ECRYPT and Contract FP6-1596 AEOLUS.

References

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, Santa Barbara, CA, USA, August 15–19, 2004. Springer-Verlag, Berlin, Germany.

[BDOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag, Berlin, Germany.

[BW06] Xavier Boyen and Brent Waters. Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307, Santa Barbara, CA, USA, August 20–24, 2006. Springer-Verlag, Berlin, Germany.

[BW07] Dan Boneh and Brent Waters. Conjunctive, subset and range queries on encrypted data. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554, Amsterdam, The Netherlands, February 21–24, 2007. Springer-Verlag, Berlin, Germany.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-Based Encryption for Fine-Grained Access Control for Encrypted Data. In *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 89–98, Alexandria, VA, USA, October 30 - November 3, 2006. ACM Press.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate Encryption Supporting Disjunction, Polynomial Equations, and Inner Products. In *Advances in Cryptology – EUROCRYPT 2008*, Lecture Notes in Computer Science, page to appear, Istanbul, Turkey, May 2008. Springer-Verlag, Berlin, Germany.

[SBC$^+$07] Elaine Shi, John Bethencourt, Hubert Chan, Dawn Song, and Adrian Perrig. Multi-Dimensional Range Query over Encrypted Data. In *2007 IEEE Symposium on Security and Privacy*, Oakland, CA, 2007. IEEE Computer Society Press.