Efficient Predicate Encryption Supporting Construction of Fine-Grained Searchable Encryption

Minqing Zhang[1,2], Xu An Wang[2], Xiaoyuan Yang[2] and Weiyi Cai[2]
[1]School of Computer Science,
Northwestern Polytechnical University, Xi'an, 710072, P. R. China
[2]Key Laboratory of Information and Network Security
Engineering University of Chinese Armed Police Force, 710086, P. R. China
wangxazjd@163.com

***Abstract*— Predicate Encryption (PE) is a new encryption paradigm which provides more sophisticated and flexible functionality. PE is sufficient for some new applications, such as fine-grained control over access to encrypted data or search on encrypted data. We present an efficient construction of predicate encryption which is IND-AH-CPA secure by employing the dual system encryption without random oracle. We clarify the relations between PE and Searchable Encryption in detail. The new notion of Public-Key Encryption with Fine-grained Keyword Search (PEFKS) is proposed. We prove that a IND-AH-CPA secure PE scheme implies the existence of a IND-PEFKS-CPA secure PEFKS scheme. We develop the transformation of PE-2-PEFKS and use the transformation to construct an efficient PEFKS scheme from our new PE scheme. We believe our results will be useful to guide a final good result.**

***Index Terms*— Predicate Encryption, Public-Key Encryption with Fine-grained Keyword Search, IND-AH-CPA, IND-PEFKS-CPA, transformation.**

I. Introduction

In traditional public key encryption, most of the systems focus on the one to one secure communication. Data is encrypted to be decrypted by a particular one who has already established a public key. The ciphertext is decrypted to learn the entire plaintext or nothing. This characteristic is insufficient for new emerging applications, such as cloud computing etc.

Recently, a new innovative class of encryption system, Predicate Encryption (PE), was proposed by Katz, Sahai and Waters [14]. PE enables one to evaluate more sophisticated and flexible functionality $F : Key_f \times CT_I \rightarrow \{0,1\}^*$ given the ciphertext CT_I and secret key Key_f. In a Predicate Encryption system, a key corresponds to a predicate and a ciphertext is associated with an attribute vector. The secret key sk_f corresponding to a predicate f can be used to decrypt a ciphertext using key associated with attribute vector I if and only if $f(I) = 1$. PE implies several recent works aimed at constructing different types of fine-grained encryption schemes, such as Identity-Based Encryption(IBE)[20], [11], Attribute-Based Encryption (ABE)[19], [13], [9], [8], Hidden-Vector Encryption (HVE)[4]. They also introduced attribute-hiding (AH) which is a stronger security notion than payload-hiding. Attribute-hiding requires that a ciphertext conceal the associated attribute as well as the plaintext, while payload-hiding only requires that a ciphertext conceal the plaintext. In some applications which require the privacy of encryption key, payload-hiding is unacceptable. The notion of attribute-hiding addresses the limitation. These functional encryption schemes have broad application in cloud computation [6], [7].

The dual system encryption which was introduced by Waters. [22] is a useful technique to obtain fully secure PE. In a dual encryption system, keys and ciphertexts can take on one of two forms: normal and semi-functional. The semi-functional keys and semi-functional ciphertexts are not used in the real system, only in the proof of security. The proof employs a sequence of security games which are shown to be indistinguishable. The first is the real security game in which both keys and ciphertext are normal. In the second game, the ciphertext is semi-functional and the keys remain normal. In subsequent games, the keys requested by the attacker are changed to be semi-functional one by one. By the final game, none of the keys given out are actually useful for decrypting a semi-functional ciphertext, and proving security becomes relatively easy.

The Public-Key Encryption with Keyword Search (PEKS) scheme was proposed by Boneh et al.[3] for some interesting applications. An email user may want the server to deliver his/her emails according to some

978-0-7695-4988-0/13 $26.00 © 2013 IEEE
DOI 10.1109/INCoS.2013.82

keywords attached on the emails. The user generates some trapdoors for the keywords and sends them to the server. The server may test whether there are these keywords in the emails. If the test outputs true, the mail will be sent to the user according to the rule. A practical PEKS must meet two conditions, consistency and security [1]. The consistency is that the decryption will not work unless the trapdoor and the ciphertext are matched. The security is that the ciphertext does not reveal any information about the keywords unless given the trapdoor.

There are many similar properties between the anonymous IBE and PEKS. In [3], Boneh et.al found that PEKS implying IBE. In [1], Abdalla et.al proved that an anonymous IBE could be transformed to a secure and consistent PEKS.

A. Our Contribution

In this paper, we do further work for PE and PEKS. The results are as follows.

(1) We present a Predicate Encryption system for the class of inner-product predicates that is fully secure without random oracles. There are several advantages over previous systems. We adopt dual system encryption to prove the security of our construction based on simple assumptions. The cost of our scheme is nearly a half of the existed scheme [14]. There is only one group element for each attribute in the ciphertext and user's key. It only requires one pairing operation for each attribute in the decrytion algorithm.

(2) In previous searchable encryption, the server only can test whether one keyword was present in the ciphertext. We extend the notion of PEKS to Public-Key Encryption with Fine-grained Keyword Search(PEFKS). PEFKS not only can test whether mutiple kewords were present in the ciphertext, but aslo can evaluate the relations of the keywords, such as equal, disjunction/conjunction. These complicated relations can't be formulated only from single keyword search by adding some relations of keywords, since it leaks unnecessary information to the server [12]. We discuss the consistency via an experiment involving adversary and define the security of PEFKS through the game between the challenge and the adversary.

(3) We prove that IND-AH-CPA secure PE implies the existence of IND-PEFKS-CPA secure PEFKS and develop a transformation of PE to PEFKS, PE-2-PEFKS. The transformation is efficient. We also use it to construct a PEFKS scheme from our PE.

B. Organization

In Section 2, we give the definition for the rest of this paper. We present our construction of PE and prove its security in section 3. In section 4, PEFKS is presented and the PE-2-PEFKS transformation is described. In Section 5 we make our conclusion.

II. Preliminaries

In this section we introduce the notion of Predicate Encryption for the class of inner-product predicates and PEFKS. We also give the necessary background on composite order bilinear groups and our complexity assumptions.

A. Public-key Encryption with Fine-grained Keywords Search

In practice, one may need to append multiple keywords to one message and describe the relations between them, e.g. "urgent and business", "family or company". A Public-key Encryption with Fine-grained Keywords Search (PEFKS) is sufficient for this request. PEFKS allows a user to define the relations of keywords which makes it more appropriate in practice. PEFKS consists of the following algorithms.

1) KG$(1^k) \rightarrow (pk, sk)$. the key generation algorithm, which takes in security parameter λ and outputs a secret key sk and a public key pk;
2) Td$(sk, \bar{w})$, the trapdoor generation algorithm, which outputs t_w for keywords vector $\bar{w}$;
3) PEFKS$(pk, \bar{x}) \rightarrow \delta)$, the encryption algorithm, which outputs ciphertext δ for keywords vector $\bar{x}$;
4) Test $(t_w, \delta) \rightarrow \{0, 1\}$, the verification algorithm, which outputs 1 if $\bar{w}\bar{x} = 0$, otherwise outputs 0.

We will discuss the security of the PEFKS.

Security Model for PEFKS.

1) Setup. The challenger runs the KG algorithm to get (pk, sk) and gives pk to the adversary;
2) Phase 1. The adversary is allowed to adaptively issue queries for trapdoors $t_{\bar{w}}$ for many keywords vector $\bar{w}$;
3) Challenge. The adversary submits two keywords vectors $\bar{x_0}, \bar{x_1}$ where $\bar{x_0} \cdot \bar{w} \neq 0$ and $\bar{x_1} \cdot \bar{w} \neq 0$ for all keys queried in Phase 1. The challenger flips a random coin b and gives the adversary $\delta^* \leftarrow PEFKS(pk, \bar{x}_b)$;
4) Phase 2. The adversary may continue to issue adaptively queries like Phase1, except that $\bar{x_0} \cdot \bar{w} = 0$ and $\bar{x_1} \cdot \bar{w} = 0$;
Guess The adversary outputs a guess b' of b.
The advantage of an IND-PEFKS-CPA adversary in this game is defined as $|Pr[b' = b] - 1/2|$.

Definition 1: A PEFKS scheme is IND-PEFKS-CPA secure if all polynomial time adversaries have at most a negligible advantage in the above security game.

B. Assumption

Our systems are constructed in composite order bilinear groups. Composite order bilinear groups was first introduced by Boneh, Goh, and Nissim [4]. The only known instantiations of composite order bilinear groups use elliptic curves over finite fields. Since the elliptic curve group order N must be infeasible to factor, it must be at least (say) 1024 bits [10]. We define a group generator G, an algorithm which takes in a security parameter 1^λ and output $(N = p_1p_2p_3, G, G_T, e)$, where p_1, p_2, p_3 are distinct primes, G and G_T are cyclic groups of order N, and $e : G \times G \to G_T$ is a map that:

1) Bilinear: for all $u, v \in G$ and $a, b \in Z_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2) Non-degeneracy: their exist g, s.t. $e(g, g)$ has order N in G_T.

We say that G is a bilinear group if the group operation in G and the bilinear map $e : G \times G \to G_T$ are both efficiently computable. Let $G_{p_1}, G_{p_2}, G_{p_3}$ denote the subgroups of G. We can see that they have the orthogonality property [16], namely, when $h_i \in G_{p_i} h_j \in G_{p_j} i \neq j$, $e(h_i, h_j)$ is the identity element in G_T. We will implement this property in our construction.

We now state the complexity assumptions that we will rely on to prove security of our systems. These assumptions are subgroup assumptions which are extension of [16].

Assumption 1: Given a group generator G, we define the following distribution:

$$\begin{aligned} &G = (N = p_1p_2p_3, G, G_T, e) \\ &g, h \leftarrow G_{p_1}, X_3 \leftarrow G_{p_3} \\ &D = (G, g, h, X_3) \\ &T_0 \leftarrow G_{p_1}, T_1 \leftarrow G_{p_1p_2} \end{aligned}$$

We define the advantage of an algorithm A in breaking Assumption 1 to be: $Adv1_{G,A}(\lambda) := |Pr[A(D, T_0) = 0] - Pr[A(D, T_1) = 0]|$. Assumption 2: Given a group generator G, we define the following distribution:

$$\begin{aligned} &G = (N = p_1p_2p_3, G, G_T, e) \\ &g, h, X_1 \leftarrow G_{p_1}, X_2, Y_2 \leftarrow G_{p_2}, X_3, Y_3 \leftarrow G_{p_3}, \\ &D = (G, g, h, X_3, X_1X_2, Y_2Y_3) \\ &T_0 \leftarrow G_{p_1p_3}, T_1 \leftarrow G \end{aligned}$$

We define the advantage of an algorithm A in breaking Assumption 2 to be: $Adv2_{G,A}(\lambda) = |Pr[A(D, T_0) = 0] - Pr[A(D, T_1) = 0]|$.

Assumption 3: Given a group generator G, we define the following distribution:

$$\begin{aligned} &G = (N = p_1p_2p_3, G, G_T, e), r \in Z_N \\ &g, h \leftarrow G_{p_1}, X_2, Y_2, Z_2 \leftarrow G_{p_2}, X_3 \leftarrow G_{p_3}, \\ &D = (G, g, X_3, Z_2, g^r X_2, hY_2) \\ &T_0 = e(g, h)^r, T_1 \leftarrow G_T \end{aligned}$$

We define the advantage of an algorithm A in breaking Assumption 3 to be: $Adv3_{G,A}(\lambda) = |Pr[A(D, T_0) = 0] - Pr[A(D, T_1) = 0]|$

III. Efficient Predicate Encryption

In this section, we present an IND-AH-CPA secure Predicate Encryption system that support inner-product predicates. The class of predicates is $F = \{f_{\bar{v}} | \bar{v} \in Z_p^n \backslash \{\bar{0}\}\}$, with $f_{\bar{v}}(\bar{x}) = 1$ if $\bar{x}\bar{v} = 0 \bmod p_1$. In our construction, subgroup G_{p_1} will be used for encryption and decryption; G_{p_3} will be used for key randomizing; G_{p_2} will be used for semi-functional keys and semi-functional ciphertext, which is not used in real encryption system. The attributes of the ciphertext and predicate of the user are expressed as a vector. We define that each element of the vector must not be 0and $\sum_{i=1,\cdots,n} x_i \neq 0$.

1) Setup(1^λ). The KGC first runs $\mathbb{G}(1^\lambda)$ to get $G = (N = p_1p_2p_3, G, G_T, e)$. It then choose random generators $g, h \in G_{p_1}$, $X_3 \in G_{p_3}$ and random $a \in Z_N$, $t_i \in Z_N$, $i = 1, \cdots, n$, where $a \neq t_i$. The public parameters and master key are given as
$$\begin{aligned} &PK = \{g, h, g_1 = g^a, \{T_i = g^{t_i}\}_{i=1,\cdots,n}\}, \\ &MK = \{a, \{t_i\}_{i=1,\cdots,n}\} \end{aligned}$$
2) KeyGen($MK, \bar{v}$). The KGC runs this algorithm to generate a user key for user who is qualified with predicate vector $\bar{v}$. First, it choose a random value $s \in Z_N$, and $W_i \in G_{p_3}, i = 1, \cdots, n$. Let $\bar{v} = \{v_1, \cdots, v_n\}$, It creates the private key as
$$sk_v = \{\{d_i = (hg^{sv_i} W_i)^{1/(a-t_i)}\}_{i=1,\cdots,n}\}$$
3) Encrypt($PK, \bar{x}, m$). To encrypt $m \in M$ with attribute vector $\bar{x}$ the sum of which must not be 0, the sender chooses random $r \in Z_N$ then it sets
$$\begin{aligned} c = \{&c_0 = me(g, h)^{-r \sum_{i=1,\cdots,n} x_i}, \\ &\{c_i = (g_1 T_i^{-1})^{rx_i}\}_{i=1,\cdots,n}\} \end{aligned}$$
4) Decrypt($sk_{\bar{v}}, c$). The receiver downloads the ciphertext. It computes
$$c_0 \cdot \prod_{i=1,\cdots,n} e(c_i, d_i)$$

Correctness. To see that correctness holds, we assume the ciphertext is well-formed:

$$\begin{aligned}
&c_0 \cdot \prod_{i=1,\cdots,n} e(c_i, d_i) \\
&= me(g,h)^{-r\sum_{i=1,\cdot,n} x_i} \\
&\cdot \prod_{i=1,\cdots,n} e((g_1^r T_i^{-r})^x, (hg^{sv_i} W_i)^{1/(a-t_i)}) \\
&= me(g,h)^{-r\sum_{i=1,\cdot,n} x_i} e(g,h)^{r\sum_{i=1,\cdot,n} x_i} \\
&\cdot e(g,h)^{sr\sum_{i=1,\cdot,n} x_i v_i} \\
&= me(g,g)^{sr\sum_{i=1,\cdot,n} x_i v_i}
\end{aligned}$$

If $\bar{x}\bar{v} \neq 0 \bmod p_1$, then the decryption algorithm evaluates to a random element in the group of G_T. If $\bar{x}\bar{v} = 0 \bmod p_1$, namely $f_{\bar{v}}(\bar{x}) = 1$, the receiver can get the message.

We can conclude that, comparing with other typical PE scheme, our construction is also more efficient.

A. Security

To prove the security, we will adopt the dual system encryption methodology which was used in [16], [22]. We define two additional structures: semi-functional ciphertexts and keys. These will not be used in the real system, but will be needed in our proof.

Semi-functional Ciphertext Let g_2 denote a generator of G_{p_2}. $c \in Z_N$, and $\{z_i \in Z_N\}_{i=1,\cdots,n}$ are random values. A semi-functional ciphertext is formed as follows.

$$\{c_i = (g^{r_1 x_i} g_2^{cz_i})^{a-t_i}\}_{i=1,\cdots,n}$$

Semi-functional Key Let g_2 denote a generator of G_{p_2}. $d \in Z_N$ and $\{y_i \in Z_N\}_{i=1,\cdots,n}$ are random values. A semi-functional key is formed as follows.

$$\{d_i = (hg^{sv_i} W_i g_2^{dy_i})^{1/(a-t_i)}\}_{i=1,\cdots,n}$$

A normal key can decrypt both normal and semi-functional ciphertexts, while a normal ciphertexts can be decrypted by both normal and semi-functional keys. When we use a semi-functional key to decrypt a semi-functional ciphertext, we are left with additional term $e(g_2, g_2)^{cd\sum_{i=1,\cdots,n} y_i z_i}$. Notice that if a semi-functional key which is satisfy that $\sum_{i=1,\cdots,n} y_i z_i = 0$ is used to decrypt a semi-functional ciphertext, decryption will still work.

Based on the assumptions, we will prove the security of our system using a sequence of games. $Game_{real}$ is the real security game in which both keys and ciphertext are normal. In the second game, $Game_0$, the ciphertext is semi-functional and all keys are normal. In game $Game_k$, the first k key queries are semi-functional and the rest are normal. By the final game, $Game_{final}$, all of the key queries are semi-functional and the challenge ciphertext is a semi-functional encryption of a random message. We will prove these games are indistinguishable in the following lemmas.

Lemma 1: Assume there is an a polynomial time adversary $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{Game_{real}} - Adv_{\mathcal{A}}^{Game_0} = \epsilon$. Then we can construct a polynomial time simulator $\mathcal{B}$ with advantage ϵ in breaking Assumption 1.

Lemma 2: Assume there is a polynomial time adversary $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{Game_{k-1}} - Adv_{\mathcal{A}}^{Game_k} = \epsilon$. Then we can construct a polynomial time simulator $\mathcal{B}$ with advantage ϵ in breaking Assumption 2.

Lemma 3: Assume there is a polynomial time adversary $\mathcal{A}$ such that $Adv_{\mathcal{A}}^{Game_q} - Adv_{\mathcal{A}}^{Game_{final}} = \epsilon$. Then we can construct a polynomial time simulator $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.

Theorem 1: If Assumptions 1, 2, and 3 hold, then our PE system is IND-AH-CPA secure.

Proof: If Assumptions 1, 2, and 3 hold, the real security game is indistinguishable from $Game_{final}$ according to the previous lemmas. In $Game_{final}$, the challenge ciphertext will give no information about b. Therefore, $\mathcal{A}$ only can attain negligible advantage in breaking our construction. This is clear that the PE system is IND-AH-CPA secure. ■

IV. PE-2-PEFKS Transformation

In [3], Boneh et.al proved that an IND-ID-CCA secure IBE could rise from a secure PEKS, but they claimed that it was hard to construct a PEKS from a secure IBE. In [1], Abdalla et.al found that IND-ANO-CPA secure IBE implied the existence of IND-PEKS-CPA secure PEKS. They also proposed a general way to transform any IND-ANO-CPA secure IBE into an IND-PEKS-CPA secure and computationally consistent PEKS. But this kind of PEKS only can test whether the keyword in the ciphertext is match to that in the trapdoor. According to the definition of PEFKS in section 2.2, we will propose a general way to transform IND-AH-CPA secure PE into a PEFKS. The PE-PEFKS transformation consists of the following steps:

1) Setup(1^λ) can be used as $KG(1^\lambda)$ to generate (pk, sk),
2) KeyGen algorithm can be used as $Td(sk, \bar{w})$ to get $t_{\bar{w}}$ which will be delivered to server;
3) Choosing a random element R, Encrypt($PK, \bar{x}, R$) $\rightarrow$ c can be used as $PEFKS(pk, \bar{x})$ to encrypt keywords $\bar{x}$, and set $\delta = (R, c)$;
4) If Decrypt($sk_{\bar{v}}, c$) $\rightarrow$ R, Test $(t_{\bar{w}}, \delta) \rightarrow 1$. Otherwise Test $(t_{\bar{w}}, \delta) \rightarrow 0$.

The consistency and security of our scheme may be reduced to the security of PE. If an adversary can ruin the consistency and security of PEFKS, we can construct an algorithm to break the PE scheme. In theorem 2, we give the formal result and proof.

Theorem 2: If PE is IND-AH-CPA secure, then PEFKS is computational consistency and IND-PEFKS-CPA secure.

A. Our PEFKS

Based on the efficient PE and theorem 2, we can construct an IND-PEFKS-CPA secure PEFKS. The PEFKS works as follows:

1) KG(1^λ) KG(1^λ)=Setup(1^λ)

$$\begin{aligned} pk &= PK \\ &= \{g, h, g_1 = g^a, \{T_i = g^{t_i}\}_{i=1,\cdots,n}\}, \\ sk &= MK = \{a, \{t_i\}_{i=1,\cdots,n}\} \end{aligned}$$

2) Td($sk, \bar{w}$) Td($sk, \bar{w}$) = KeyGen($MK, \bar{w}$)

$$t_w = \{\{d_i = (hg^{sw_i}W_i)^{1/(a-t_i)}\}_{i=1,\cdots,n}\}$$

3) PEFKS($\bar{pk}, \bar{x}$) To encrypt keyword vector $\bar{x}$, the sender first Chooses a random element R, then runs the Encrypt($PK, \bar{x}, R$) to get c

$$\begin{aligned} c = \{c_0 = Re(g,h)^{-r\sum_{i=1,\cdots,n} x_i}, \\ \{c_i = (g_1 T_i^{-1})^{rx_i}\}_{i=1,\cdots,n}\} \end{aligned}$$

The ciphertext is set as $\delta = (R, c)$;

4) Test($t_{\bar{w}}, \delta$) The server computes $c_0 \prod_{i=1,\cdots,n} e(c_i, d_i)$. If it values to R, namely $\bar{w}\bar{x} = 0$, the server sets Test($t_{\bar{w}}, \delta$)=1. Otherwise it sets Test($t_{\bar{w}}, \delta$)=0.

V. Conclusion

We present an Inner-product Predicate Encryption system that is practical based on composite order bilinear groups. The security of our construction is proven IND-AH-CPA secure by adopting the dual system encryption, which is sufficient for PE-2-PEFKS transformation. PEFKS is proposed in this paper. The new notion will be more useful for applications.

There are still some interesting directions. One is to design more sophisticated and flexible functionality $F : Key \times CT \to \{0,1\}^*$ which will be more expressive than inner-product. Another is the possibility of transformation of PEFKS to PE.

VI. Acknowledgements

The second author is the corresponding author. This work is supported by the National Natural Science Foundation of China under contract no. 61103230, 61272492, 61103231.

References

[1] Abdalla M., Bellare M., Catalano D., Kiltz E., Kohno T., Lange T., Malone-Lee J., Neven G., Paillier P. and Shi H.. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. In *Crypto 2005*, LNCS 3621, Springer-Verlag, 2005, pp. 205-222.

[2] Baek J., Safiavi-Naini R., Susilo W.. Public Key Encryption with Keyword Search Revisited. Cryptology ePrint Archive, Report 2005/119.

[3] Boneh D., Crescenzo G. D., Ostrovsky R., and Persiano G.. Public Key Encryption with Keyword Search. In *Eurocrypt 2004*, LNCS 3027, Springer-Verlag, 2004, pp. 506-522.

[4] Boneh D., Goh E., Nissim K.. Evaluating 2-dnf formulas on ciphertexts. In *TCC 2005*, LNCS , pp. 325-342.

[5] Boneh D., Waters B.. Conjunctive, Subset, and Range Queries on Encrypted Data. In *TCC 2007*, LNCS 4392, pp. 535-554.

[6] X. Chen, J. Li, W. Susilo. Efficient fair conditional payments for outsourcing computations. In *IEEE Transactions on Information Forensics and Security*, 7(6), pages 1687–1694, 2012.

[7] X. Chen, J. Li, J. Ma, Q. Tang, W. Lou. New algorithms for secure outsourcing of modular exponentiations. In *ESORICS 2012*, volume 7459 of LNCS, pages 541–556, 2012.

[8] Chase M.. Multi-Authority Attribute Based Encryption. In *TCC 2007*, LNCS , pp. 515-534.

[9] Cheung L., Newport C.. Provably Secure Ciphertext Policy ABE. In *CCS 2007*, pp. 456-465.

[10] Freeman D. M.. Converting Pairing-Based Cryptosystems from Composite-Order Groups to Prime-Order Groups. In *EUROCRYPT 2010*, LNCS 6110, pp. 44-61.

[11] Gentry C.. Practical Identity-Based Encryption Without Random Oracles. In *EUROCRYPT 2006*, LNCS 4004, pp. 445-464.

[12] Golle P., Staddon J., and Waters B.. Secure Conjunctive Search over Encrypted Data. In *ACNS 2004*, LNCS 3089, pp. 31-45.

[13] Goyal V., Pandey O., Sahai A., Waters B.. Attribute-based encryption for finegrained access control of encrypted data. In *CCS 2006*, pp.89-98.

[14] Katz J., Sahai A., Waters B.. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In *EUROCRYPT 2008*. LNCS 4965, pp. 146-162.

[15] Lewko A., Okamoto T., Sahai A., Takashima K., Waters B.. Fully Secure Functional Encryption: Attribute-based Encryption and (Hierarchical) Inner Product Encryption. In *EUROCRYPT 2010*, LNCS 6110, pp. 62-91.

[16] Lewko A., Waters B.. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC 2010*, LNCS 5978, pp. 455-479.

[17] Okamoto T., Takashima K.. Hierarchical Predicate Encryption for Inner-Products. In *ASIACRYPT 2009*, LNCS 5912, pp. 214-231.

[18] Okamoto T., Takashima K.. Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption. In *CRYPTO 2010*, LNCS 6223, pp. 191-208.

[19] Sahai A., Waters B.. Fuzzy Identity Based Encryption. In *EUROCRYPT 2005*, LNCS 3494, pp. 457-473.

[20] Shamir A.. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO 1984*, LNCS 196, pp. 47-53.

[21] Shi E., Waters B.. Delegating Capabilities in Predicate Encryption Systems. In *ICALP 2008*, LNCS 5126, pp. 560-578.

[22] Waters B.. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *CRYPTO 2009*, LNCS 5677, pp. 619-636.