Practical and Secure Integrated PKE+PEKS with Keyword Privacy

Francesco Buccafurri[1], Gianluca Lax[1], Rajeev Anand Sahu[2] and Vishal Saraswat[2]
[1]*DIIES Department, University of Reggio Calabria, Reggio Calabria, Italy*
[2]*C.R.Rao Advanced Institute of Mathematics Statistics and Computer Science, Hyderabad, India*

Keywords: PKE+PEKS, Searchable Encryption, Perfect Keyword Privacy, Asymmetric Pairings (Type 3), Provable Security, Standard Model, SXDH, E-governance, Privacy in the Cloud.

Abstract: Public-key encryption with keyword search (PEKS) schemes are useful to delegate searching capabilities on encrypted data to a third party, who does not hold the entire secret key, but only an appropriate token which allows searching operations but preserves data privacy. We propose an efficient and practical integrated public-key encryption (PKE) and public-key encryption with keyword search (PEKS) scheme (PKE+PEKS) which we prove to be secure in the strongest security notion for PKE+PEKS schemes. In particular, we provide a unified security proof of its joint CCA-security in standard model. The security of our scheme relies on Symmetric eXternal Diffie-Hellman (SXDH) assumption which is a much simpler and more standard hardness assumption than the ones used in most of the comparable schemes. Ours is the first construction to use asymmetric pairings which enable an extremely fast implementation useful for practical applications. Finally we compare our scheme with other proposed integrated PKE+PEKS schemes and provide a relative analysis of its efficiency.

1 INTRODUCTION

(Boneh et al., 2004) gave the first formal construction of "searchable encryption" in public key setting and called it *public key encryption with keyword search* (PEKS), popularly known as BDOP scheme. The basic advantage of this primitive is that it allows one to delegate to a third party the capability of "searching on public key encrypted data" without impacting privacy. PEKS basically enables a search function to a public key encryption (PKE) scheme and hence a PEKS is directly related to an underlying PKE scheme and both are used together. We call such a combination integrated PKE and PEKS and denote it as PKE+PEKS. For a PKE+PEKS scheme the privacy must be simultaneously achieved for both the message (that is, data) and the keyword, that is, IND-PKE-CCA and IND-PEKS-CCA. But achieving the security based on two independent CCA-secure schemes is not trivial. So a unified security model for the joint CCA-security of PKE+PEKS is desired.

1.1 Related Work

(Abdalla et al., 2005) defined computational, statistical and perfect variations of consistency for a PEKS scheme and proposed a transform of an anonymous identity-based encryption (IBE) scheme to a PEKS scheme. (Baek et al., 2006) formally defined a combined scheme for PKE and PEKS based on the BDOP-PEKS and a variation of ElGamal encryption scheme with the randomness reuse technique (Kurosawa, 2002). (Crescenzo and Saraswat, 2007) constructed a PEKS scheme which, unlike all other schemes, is not based on bilinear forms. Various groups (Boyen and Waters, 2006; Fuhr and Paillier, 2007; Zhang and Imai, 2007; Baek et al., 2008; Abdalla et al., 2010) have studied the design and efficiency of the PEKS schemes while (Lu et al., 2009; Shmueli et al., 2010; Ibraimi et al., 2011) have studied the application aspects of PEKS.

(Boneh et al., 2004) formalized the security precisely for the PEKS scheme with IND-PEKS-CPA notion. (Baek et al., 2008) combined PKE and PEKS with a joint security notion but their notion covered only data privacy and not the keyword privacy. (Zhang and Imai, 2007) first extended the security notion to achieve both data privacy and keyword privacy. The security notion for data security is IND-PKE-CCA, which is achieved in their scheme using a tag-based CCA-secure PKE scheme, and for keyword privacy the notion is IND-PEKS-CPA, which they

Buccafurri F., Lax G., Sahu R. and Saraswat V..
Practical and Secure Integrated PKE+PEKS with Keyword Privacy.
DOI: 10.5220/0005577104480453
In *Proceedings of the 12th International Conference on Security and Cryptography* (SECRYPT-2015), pages 448-453
ISBN: 978-989-758-117-5

have achieved using a CPA-secure PEKS scheme. But their construction suffers from double key size, which increases key-maintenance overheads unnecessarily during the practical implementations. But none of the above works prove joint security of a PKE+PEKS scheme in strongest notion, that is, 'IND-PKE+PEKS-CCA security'. (Abdalla et al., 2010) introduced a new combined CCA-security notion on the standard model with a privilege to the adversary to access both decryption oracle and test oracle. Following the idea of (Dodis and Katz, 2005), they constructed a PKE+PEKS scheme by combining two schemes, a tag-based CCA-secure PKE scheme and a tag-based CCA-secure PEKS scheme. But their construction suffers from double key size and an increase in the computational overhead of the resulting PKE+PEKS. Recently, (Chen et al., 2014) gave a generic construction of a PKE+PEKS scheme from an anonymous IBE and one-time signatures using a single key for both PKE and PEKS operations.

1.2 Our Contribution

As discussed above that a PEKS scheme joins PKE to provide it a searchable functionality, hence combining both the schemes in a secure manner is of great interest. Attempts for the security of PEKS were started with the IND-PEKS-CPA notion of security on random oracle (Boneh et al., 2004), where the 'keyword' is considered as a plaintext. Though there have been lot of research on searchable encryption, the only fully secure schemes (Abdalla et al., 2010; Chen et al., 2014) are too inefficient to be practical enough to be used in implementation. We propose a state of art efficient, computationally and bandwidth-wise, fully secure practical scheme which, we believe, can be used in real applications.

At the heart of our scheme are asymmetric pairings (Type 3) which enables our scheme to have very short ciphertexts and extremely fast implementation. Typically, compared with symmetric pairings, for Type 3 pairings, the estimated bit sizes of group elements, over which most of the computation is done and are communicated, are four times smaller for 256-bit (AES) security (Chen et al., 2012).

We also give a relatively unified model for the joint security of PKE+PEKS scheme. We note that a scheme with a security proof in the random oracle model implies no security in the real world (Canetti et al., 2004), security of our scheme is proved in the standard model and hence our scheme achieves practical security. We will provide a full security analysis and will envisage a few suitable applications in the full version of this paper.

2 PRELIMINARIES

We denote by $y \leftarrow A(x)$ the operation of running a randomized or deterministic algorithm $A(x)$ and storing the output to the variable y. If X is a set, then $v \xleftarrow{\$} X$ denotes the operation of choosing an element v of X according to the uniform random distribution on X. We say that a given function $f : N \to [0,1]$ is *negligible in n* if $f(n) < 1/p(n)$ for any polynomial p for sufficiently large n. For a group G and $g \in G$, we write $G = \langle g \rangle$ if g is a generator of G. We let G_1, G_2 and G_T be multiplicative cyclic groups of the same prime order q and $e : G_1 \times G_2 \to G_T$ be a pairing defined as follows.

Definition 1. A map $e : G_1 \times G_2 \to G_T$ is called a *cryptographic bilinear map* or a *pairing* if it satisfies the following properties:

1. *Bilinearity*: For all $(g_1, g_2) \in G_1 \times G_2$ and for all $a, b \in \mathbb{Z}_q$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$,
2. *Non-Degeneracy*: There exists $(g_1, g_2) \in G_1 \times G_2$ such that $e(g_1, g_2) \neq 1$, the identity of G_T.
3. *Computability*: There exists an efficient algorithm to compute $e(g_1, g_2) \in G_T$, for all $(g_1, g_2) \in G_1 \times G_2$.

A pairing $e : G_1 \times G_2 \to G_T$ is called a *symmetric* or a *Type-1* pairing if $G_1 = G_2$ otherwise it is called *asymmetric*. Asymmetric pairings are further categorized into Type 2 and Type 3 pairings. If there exists an efficiently computable isomorphism between G_1 and G_2, the pairing is referred to as *Type 2*, whereas if there is no efficiently computable isomorphism between G_1 and G_2, the pairing is referred to as *Type-3*.

2.1 Dual Pairing Vector Space

Let $g \in G$ and $(g_1, g_2) \in G_1 \times G_2$. For a vector $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_q^n$ we define a vector of group elements as $g^{\mathbf{v}} := (g^{v_1}, \ldots, g^{v_n})$. For vectors $\mathbf{v} = (v_1, \ldots, v_n), \mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{Z}_q^n$ and for any $a \in \mathbb{Z}_q$, we further define the following properties on such a vector space,

$$(g^a)^{\mathbf{v}} := g^{a\mathbf{v}} = (g^{a\mathbf{v}_1}, \ldots, g^{a\mathbf{v}_n});$$
$$g^{\mathbf{v}} g^{\mathbf{w}} := g^{\mathbf{v}+\mathbf{w}} = (g^{\mathbf{v}_1+\mathbf{w}_1}, \ldots, g^{\mathbf{v}_n+\mathbf{w}_n})$$
$$e(g_1^{\mathbf{v}}, g_2^{\mathbf{w}}) := \prod_{i=1}^{n} e(g_1^{v_i}, g_2^{w_i}) = e(g_1, g_2)^{\mathbf{v}\cdot\mathbf{w}}.$$

Definition 2. Two bases $\mathcal{D} := \{d_1, d_2, d_3, d_4\}$ and $\mathcal{D}^* = \{d_1^*, d_2^*, d_3^*, d_4^*\}$ of $\mathbb{Z}_q^4$ are dual orthonormal if for all $1 \leq j \neq k \leq 4$,

$$d_j \cdot d_k^* = 0 \pmod{q} \quad \text{and} \quad d_j \cdot d_j^* = \psi \pmod{q}$$

where $\psi \xleftarrow{\$} \mathbb{Z}_q$.

For such dual bases, for all $(g_1, g_2) \in G_1 \times G_2$,

$$e(g_1^{d_j}, g_2^{d_k^*}) = 1 \quad \text{whenever } j \neq k.$$

Theorem 1 ((Okamoto and Takashima, 2010)). *We can efficiently select two random dual orthonormal bases $\mathcal{D} := \{d_1, d_2, d_3, d_4\}$ and $\mathcal{D}^* = \{d_1^*, d_2^*, d_3^*, d_4^*\}$ of $\mathbb{Z}_q^4$.*

2.2 Symmetric eXternal Diffie-Hellman (SXDH) Assumption

Let G_1 and G_2 be multiplicative cyclic groups as stated above, with $g_1 \in G_1$ and $g_2 \in G_2$.

Definition 3. Let G be a multiplicative cyclic group and g a generator. Let $a, b, c \in \mathbb{Z}_q^*$ be randomly chosen and kept secret. Given $g, g^a, g^b, g^c \in G$, the *decisional Diffie-Hellman problem* (DDHP) in the group G is to decide if $g^{ab} = g^c$.

Definition 4. The *DDH assumption* holds in a group G if there is no efficient algorithm which can solve DDHP in G.

Definition 5. Given two cyclic groups G_1 and G_2, we say the *Symmetric eXternal Diffie-Hellman* (SXDH) assumption holds if DDH assumption is true in both the groups G_1 and G_2.

3 INTEGRATED PKE AND PEKS SCHEME (PKE+PEKS)

Here we give a formal definition of an integrated public-key encryption (PKE) and public-key encryption with keyword search (PEKS) scheme (PKE+PEKS) based on the works of (Boneh et al., 2004; Baek et al., 2008; Chen et al., 2014).

In PEKS, three parties called *sender*, *receiver* and *server* are involved. The sender is a party that creates and sends encrypted keywords, which we call *PEKS ciphertexts*. The server is a party that receives PEKS ciphertexts and performs search upon receiving trapdoors from the receiver. The receiver is a party that creates trapdoors and sends them to the server to find the data that it wants.

3.1 Formal Definition of PKE+PEKS

A PKE+PEKS scheme comprises of six algorithms: *Setup*, *KeyGen*, *Encrypt*, *Decrypt*, *TokenGen* and *Test*.

$G \leftarrow$ ***Setup***(1^k)**:** This is the system initialization algorithm run by the receiver which takes as input a security parameter 1^k and outputs public parameters *Params* $G := (q, G_1, G_2, G_T, g_1, g_2, e)$ where $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$ and G_T are cyclic groups of prime order q, and $e : G_1 \times G_2 \to G_T$ is a Type 3 pairing.

$(pk_X, sk_X) \leftarrow$ ***KeyGen***(G)**:** This is the key generation algorithm run by a user X which takes as input the public parameters G and outputs a key pair (pk_X, sk_X). For the receiver $X = R$, the key pair is its (public key, private key) pair (pk_R, sk_R) and for a sender $X = S$, the key pair is its (verification key, signing key) pair (vk_S, sk_S).

$\mathcal{U} \leftarrow$ ***Encrypt***(G, pk_R, sk_S, m, w)**:** This is a randomized algorithm run by the sender and takes input *Params* G, the receiver's public key pk_R, the sender's signing key sk_S, a message m and keyword w and outputs the joint PKE+PEKS ciphertext $\mathcal{U}$.

$m \leftarrow$ ***Decrypt***$(G, pk_R, sk_R, \mathcal{U})$ This is a deterministic algorithm run by the receiver and takes input *Params* G, the receiver's public key pk_R and the secret key sk_R and a ciphertext $\mathcal{U}$ and outputs a message m or $\perp$.

$t_w \leftarrow$ ***TokenGen***(G, pk_R, sk_R, w) This is a randomized algorithm run by the receiver and takes input *Params* G, the receiver's public key pk_R and the secret key sk_R and a keyword w and outputs a token t_w which it gives to the server.

$b \leftarrow$ ***Test***$(G, pk_R, t_w, \mathcal{U})$ This is a deterministic algorithm run by the server and takes input *Params* G, the receiver's public key pk_R, a token t_w and a ciphertext $\mathcal{U}$ and outputs a bit $b \in \{0, 1\}$ or $\perp$.

Where the context is clear, the inputs *Params* and the keys will be assumed to be implicit and we will not write them explicitly in the algorithms.

3.2 Security Model for PKE+PEKS

Joint Data and Keyword Privacy for PKE+PEKS schemes is defined via the following experiment.

Setup: On input a security parameter 1^k, the challenger $\mathcal{C}$ runs $KeyGen(1^k)$ to generate the public parameter *Params* G and the system key pair (pk, sk) and gives the adversary $\mathcal{A}$ the public key pk.

Phase 1: $\mathcal{A}$ can adaptively make three types of queries:

- Decryption query $\langle u \rangle$: $\mathcal{C}$ responds with $m \leftarrow Decrypt(sk, u)$.
- Token query $\langle w \rangle$: $\mathcal{C}$ responds with $t_w \leftarrow TokenGen(sk, w)$.
- Test query $\langle u, w \rangle$: $\mathcal{C}$ responds with $Test(u, t_w)$ where $t_w \leftarrow TokenGen(sk, w)$.

Challenge: $\mathcal{A}$ outputs two messages m_0^* and m_1^* and two keywords w_0^* and w_1^*. $\mathcal{C}$ picks two random bits a, b and sends $u^* \leftarrow Encrypt(pk, m_a^*, w_b^*)$ to $\mathcal{A}$ as the challenge ciphertext.

Phase 2: $\mathcal{A}$ can adaptively make more queries as in Phase 1 subject to the restrictions that it is not al-

lowed to make
- Decryption query $\langle u^* \rangle$,
- Token queries $\langle w_0^* \rangle$ and $\langle w_1^* \rangle$, and
- Test queries $\langle u^*, w_0^* \rangle$ and $\langle u^*, w_1^* \rangle$.

$\mathcal{C}$ responds the same way as in Phase 1.

Guess: $\mathcal{A}$ outputs its guess (a_0, b_0) for (a, b).

Definition 6 (Data Privacy)**.** The adversary succeeds in breaking the data privacy if $a_0 = a$ and we say that a PKE+PEKS scheme has data privacy if there is no PPT adversary with a non-negligible advantage in 1^k in the above experiment.

Definition 7 (Keyword Privacy)**.** The adversary succeeds in breaking the keyword privacy if $b_0 = b$ and we say that a PKE+PEKS scheme has keyword privacy if there is no PPT adversary with a non-negligible advantage in 1^k in the above experiment.

Definition 8. We say a PKE+PEKS scheme is jointly CCA-secure if it has keyword privacy and data privacy simultaneously.

Remark 1. The joint CCA-security notion defined by us embodies both IND-PKE-CCA security and IND-PEKS-CCA security for PKE+PEKS in the joint sense and is stronger than previous ones considered in (Baek et al., 2006; Zhang and Imai, 2007), both of which are insecure in our joint CCA-security notion as analyzed in Section 1.

4 PROPOSED SCHEME

We present here our efficient and CCA secure integrated PKE+PEKS scheme motivated by the short IBE and IBS schemes of (Chen et al., 2012). As described in Section 3, our scheme consists of the following algorithms: *Setup*, *KeyGen*, *Encrypt*, *Decrypt*, *TokenGen*, and *Test*.

***Setup*:** A receiver R wishing to receive joint PKE+PEKS messages generates $G := (q, G_1, G_2, G_T, g_1, g_2, e)$ where $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$ and G_T are cyclic groups of prime order q and $e : G_1 \times G_2 \to G_T$ is a Type 3 pairing. The receiver then chooses two cryptographic hash functions $H : \{0,1\}^* \to \mathbb{Z}_q$ and $J : \{0,1\}^* \to G_T$. Finally, R publishes G, H and J as the public parameters of the system. (These may be considered as part R's public key, but for sake of clarity we keep these separate.)

***KeyGen*:** To generate keys for the system, the receiver does the following:
- samples random dual orthonormal bases $\mathcal{D}$ and $\mathcal{D}^*$ where $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$ and $\mathcal{D}^* = \{d_1^*, d_2^*, d_3^*, d_4^*\}$ so that for all $1 \le j \le 4$, $d_j \cdot d_j^* = \psi$;
- picks $\alpha \xleftarrow{\$} \mathbb{Z}_q$;
- computes $g_T^\alpha := e(g_1, g_2)^{\alpha d_1 \cdot d_1^*}$;
- sets the public key $pk_R = \{g_T^\alpha, g_1^{d_1}, g_1^{d_2}\}$; and
- sets the master secret $sk_R = \{\alpha, g_2^{d_1^*}, g_2^{d_2^*}\}$.

***Encrypt*:** To encrypt a message $m \in G_T$ with a keyword $w \in \{0,1\}^*$ for the receiver R, a sender S does the following:
- samples random dual orthonormal bases $\mathcal{E}$ and $\mathcal{E}^*$ where $\mathcal{E} = \{e_1, e_2, e_3, e_4\}$ and $\mathcal{E}^* = \{e_1^*, e_2^*, e_3^*, e_4^*\}$ so that for all $1 \le j \le 4$, $e_j \cdot e_j^* = \phi$;
- picks $\beta \xleftarrow{\$} \mathbb{Z}_q$;
- computes $g_T^\beta := e(g_1, g_2)^{\beta e_1 \cdot e_1^*}$;
- sets the signing key $sk_S = (\beta, g_2^{e_1^*}, g_2^{e_2^*})$; and
- sets the verification key $vk_S = (g_T^\beta, g_1^{e_1}, g_1^{e_2})$.
- sets $vk = J(vk_S)$, picks $x, y \xleftarrow{\$} \mathbb{Z}_q$ and computes
- $ID_{vk} = H(0 \| vk)$ and $ID_w = H(1 \| w)$
- $C_m = (C_{m_1} := m \cdot (g_T^\alpha)^x, C_{m_2} := g_1^{x(d_1 + ID_{vk} d_2)})$
- $C_w = (C_{w_1} := vk \cdot (g_T^\alpha)^x, C_{w_2} := g_1^{x(d_1 + ID_w d_2)})$
- $t = H(C_m \| C_w)$
- $\sigma = g_1^{(\beta + yt) e_1^* - y e_2^*}$

and finally declares the ciphertext $\mathcal{U} = (vk_S, C_m, C_w, \sigma)$.

***Decrypt*:** To decrypt the ciphertext $\mathcal{U} = (u_1, u_2, u_3, u_4)$, the receiver does the following:
- obtains h_1, h_2 and h_T from u_1;
- computes $t = H(u_2 \| u_3)$; and
- checks whether $e(u_4, g_2^{e_1 + te_2}) = g_T^\beta$.
- If the above equality does not hold then outputs $\perp$. Otherwise it obtains C_1, C_2 from u_2 and
- computes $vk = J(u_1)$ and sets $ID_{vk} = H(0 \| vk)$;
- computes the corresponding decryption key $SK_{ID_{vk}} = g_2^{\alpha d_1^* + (ID_{vk} \cdot d_1^* - d_2^*)}$;
- Finally it outputs $m \leftarrow \frac{C_1}{e(C_2, SK_{ID_{vk}})}$.

***TokenGen*:** To generate a token t_w for the keyword w, the receiver
- computes $ID_w = H(1 \| w)$;
- picks $r \xleftarrow{\$} \mathbb{Z}_q$ and computes $t_w = g_2^{\alpha d_1^* + r(ID_w \cdot d_1^* - d_2^*)}$

and gives to the server t_w.

***Test*:** To test whether the ciphertext $\mathcal{U} = (u_1, u_2, u_3, u_4)$ includes the keyword w or not using the token t_w, the server does the following:
- obtains h_1, h_2 and h_T from u_1;
- computes $t = H(u_2 \| u_3)$; and
- checks whether $e(u_4, g_2^{e_1 + te_2}) = g_T^\beta$.

- If the above equality does not hold then outputs 0.
- Otherwise it obtains C_3, C_4 from u_3 and checks whether
$$\frac{C_3}{e(C_4, t_w)} = J(u_1)$$
- If yes then outputs 1, else outputs 0.

4.1 Correctness of the Proposed Scheme

Theorem 2. *The presented scheme is correct.*

Proof. The decryption is correct since

$$e(C_2, SK_{ID_{vk}}) = e(g_1^{x(d_1+ID_{vk}d_2)}, g_2^{\alpha d_1^* + r(ID_{vk}d_1^* - d_2^*)})$$
$$= e(g_1, g_2)^{\alpha x d_1 \cdot d_1^*} e(g_1, g_2)^{xrID_{vk}(d_1 \cdot d_1^* - d_2 \cdot d_2^*)}$$
$$= g_T^{\alpha x}$$

and

$$\frac{C_1}{e(C_2, SK_{ID_{vk}})} = \frac{m \cdot (g_T^{\alpha})^x}{g_T^{\alpha x}} = m.$$

The testing is correct since

$$e(C_4, t_w) = e(g_1^{x(d_1+ID_w d_2)}, g_2^{\alpha d_1^* + r(ID_w \cdot d_1^* - d_2^*)})$$
$$= e(g_1, g_2)^{x(d_1+ID_w d_2)(\alpha d_1^* + r(ID_w \cdot d_1^* - d_2^*))}$$
$$= e(g_1, g_2)^{x(\alpha + rID_w)(d_1 \cdot d_1^* + ID_w d_2 \cdot d_1^*) - xID_w(d_1 \cdot d_2^* - d_2 \cdot d_2^*)}$$
$$= e(g_1, g_2)^{x(\alpha + rID_w)\psi - xID_w\psi}$$
$$= e(g_1, g_2)^{x\alpha\psi} = g_T^{x\alpha}$$

and

$$\frac{C_3}{e(C_4, t_w)} = \frac{vk \cdot (g_T^{\alpha})^x}{g_T^{x\alpha}} = vk = J(u_1).$$

5 SECURITY ANALYSIS

In this section, we analyze the security of our scheme. We here give a proof sketch of the security of our scheme. Note that the generic proof is quite complicated and causes security degradation and in the full version of this paper we will give a direct proof which is conceptually simpler and provides tight bounds.

Note that the basic IBE scheme that we use in our scheme was proved to be ANO-IBE-CCA anonymous (Chen et al., 2012) based on the SXDH assumption. The signature scheme that we use is a modification of the signature scheme in (Chen et al., 2012) which was obtained using a Naor-transform of the IBE in (Chen et al., 2012) and is proved to be sEUF-CMA secure based on the SXDH assumption. Straight-forward modifications of proof in (Chen et al., 2012) will also show the security of our signature scheme under the same assumption. Thus proceeding as in (Chen et al., 2014), we can also prove that our scheme is jointly CCA-secure, that is, our scheme is both IND-PKE-CCA and IND-PEKS-CCA and has keyword privacy and data privacy simultaneously.

6 EFFICIENCY ANALYSIS

We compare the efficiency of our Integrated PKE+PEKS scheme with the existing PEKS schemes (Baek et al., 2006; Zhang and Imai, 2007; Chen et al., 2014) in Table 1, and show that our scheme is more efficient than these schemes. In each of the four phases: Encryption, Decryption, Token Gen. and Test, we compare the total number of bilinear pairings (P), exponentiations and inverse in Z_q denoted as $E(Z_q)$ and $I(Z_q)$, exponentiations and multiplications in G_1 denoted as $E(G_1)$ and $M(G_1)$, exponentiations and multiplications in G_2 denoted as $E(G_2)$ and $M(G_2)$ and exponentiations and multiplications in G_T denoted as $E(G_T)$ and $M(G_T)$. Our scheme uses asymmetric pairings so we have considered operations in all the three different groups, G_1, G_2 and G_T, while for schemes using symmetric pairings, we have counted operations in groups G_1 and G_2, considering $|G_2| \approx |G_T|$.

7 CONCLUSION AND FUTURE WORK

We have proposed an integrated public-key encryption (PKE) and public-key encryption with keyword search (PEKS) scheme (PKE+PEKS) which is efficient, computationally and bandwidth-wise, and secure in the strongest sense for PKE+PEKS schemes. In the full version of this paper, we will provide a full security analysis of our scheme in a relatively unified model for the joint security of PKE+PEKS scheme in standard model. Further, we will provide a more detailed efficiency analysis and comparison with existing similar schemes. Finally we will envisage a few suitable applications for practical implementation of our scheme.

ACKNOWLEDGEMENT

This work has been partially supported by the TENACE PRIN Project (n. 20103P34XC) funded by the Italian Ministry of Education, University and Research, and by the Program “Programma Operativo

Table 1: Efficiency Comparison.

Operation	Scheme	P	E(Z_q)	I(Z_q)	E(G_1)	M(G_1)	E(G_2)	M(G_2)	E(G_T)	M(G_T)
Encryption	(Baek et al., 2006)	1	0	0	2	0	-	-	1	0
	(Zhang and Imai, 2007)	2	0	0	2	1	-	-	6	1
	(Chen et al., 2014)	5	0	1	7	2	-	-	5	2
	Our scheme	0	0	0	2	0	1	0	1	2
Decryption	(Baek et al., 2006)	0	0	0	1	0	-	-	0	0
	(Zhang and Imai, 2007)	0	0	0	0	0	-	-	2	1
	(Chen et al., 2014)	3	0	1	3	2	-	-	0	1
	Our scheme	2	0	0	1	0	0	0	0	1
TokenGen	(Baek et al., 2006)	0	0	0	1	0	-	-	0	0
	(Zhang and Imai, 2007)	0	0	0	1	1	-	-	0	0
	(Chen et al., 2014)	0	0	1	1	1	-	-	0	0
	Our scheme	0	0	0	0	0	0	0	0	0
Test	(Baek et al., 2006)	1	0	0	0	0	-	-	0	0
	(Zhang and Imai, 2007)	1	0	0	0	0	-	-	1	1
	(Chen et al., 2014)	4	1	0	2	2	0	0	2	3
	Our scheme	2	0	0	1	0	0	0	1	0
Overall comparison	(Baek et al., 2006)	2	0	0	4	1	-	-	1	0
	(Zhang and Imai, 2007)	3	0	0	3	2	-	-	9	3
	(Chen et al., 2014)	12	1	3	13	7	-	-	7	6
	Our scheme	4	0	0	4	0	1	0	2	3

Nazionale Ricerca e Competitività" 2007-2013, Distretto Tecnologico CyberSecurity funded by the Italian Ministry of Education, University and Research.

REFERENCES

Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., and Shi, H. (2005). Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. In *CRYPTO*, volume 3621 of *LNCS*, pages 205–222.

Abdalla, M., Bellare, M., and Neven, G. (2010). Robust encryption. In *TCC*, volume 5978 of *LNCS*, pages 480–497.

Baek, J., Safavi-Naini, R., and Susilo, W. (2006). On the integration of public key data encryption and public key encryption with keyword search. In *ISC*, volume 4176 of *LNCS*, pages 217–232.

Baek, J., Safavi-Naini, R., and Susilo, W. (2008). Public key encryption with keyword search revisited. In *ICCSA*, volume 5072 of *LNCS*, pages 1249–1259.

Boneh, D., Di Crescenzo, G., Ostrovsky, R., and Persiano, G. (2004). Public key encryption with keyword search. In *EuroCrypt*, volume 3027 of *LNCS*, pages 506–522.

Boyen, X. and Waters, B. (2006). Anonymous hierarchical identity-based encryption (without random oracles). In *CRYPTO*, volume 4117 of *LNCS*, pages 290–307.

Canetti, R., Goldreich, O., and Halevi, S. (2004). The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4):557–594.

Chen, J., Lim, H., Ling, S., Wang, H., and Wee, H. (2012). Shorter ibe and signatures via asymmetric pairings. In *Pairing*, volume 7708 of *LNCS*, pages 122–140.

Chen, Y., Zhang, J., Lin, D., and Zhang, Z. (2014). Generic constructions of integrated pke and peks. *Designs, Codes and Cryptography*, pages 1–34.

Crescenzo, G. D. and Saraswat, V. (2007). Public key encryption with searchable keywords based on jacobi symbols. In *IndoCrypt*, volume 4859 of *LNCS*, pages 282–296.

Dodis, Y. and Katz, J. (2005). Chosen-ciphertext security of multiple encryption. In *TCC*, volume 3378 of *LNCS*, pages 188–209.

Fuhr, T. and Paillier, P. (2007). Decryptable searchable encryption. In *ProvSec*, pages 228–236.

Ibraimi, L., Nikova, S., Hartel, P., and Jonker, W. (2011). Public-key encryption with delegated search. In *ACNS*, pages 532–549.

Kurosawa, K. (2002). Multi-recipient public-key encryption with shortened ciphertext. In *PKC*, pages 48–63.

Lu, W., Swaminathan, A., Varna, A. L., and Wu, M. (2009). Enabling search over encrypted multimedia databases. In *IS&T/SPIE Electronic Imaging*, pages 725418–725418. International Society for Optics and Photonics.

Okamoto, T. and Takashima, K. (2010). Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, volume 6223 of *LNCS*, pages 191–208.

Shmueli, E., Vaisenberg, R., Elovici, Y., and Glezer, C. (2010). Database encryption: an overview of contemporary challenges and design considerations. *ACM SIGMOD Record*, 38(3):29–34.

Zhang, R. and Imai, H. (2007). Generic combination of public key encryption with keyword search and public key encryption. In *CANS*, volume 4856 of *LNCS*, pages 159–174.