

# COMSM0007 Assignment: A Summary of Public Key Encryption with keyword Search

Dominic Moylett - dm1905@my.bristol.ac.uk

April 10, 2015

## 1 Introduction

Public Key Encryption with keyword Search[3] is a paper by Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky and Giuseppe Persiano published at Eurocrypt 2004. This paper looks at the ability to search for items contained within encrypted data.

An example use is an email server: Alice can grant the server the ability to check if the email has the keyword 'Urgent'. When the server now receives an email, it checks if it has the keyword 'Urgent': if so the email is routed to Alice's pager, otherwise it is sent to her laptop. Non-interactive public key encryption with keyword search (sometimes abbreviated to 'searchable encryption') schemes mean that the server can check an encrypted email for keywords without learning what the keywords themselves are.

Due to the computational cost of public key encryption in practice, a searchable encryption scheme does not allow a computer to search the entire encrypted message for keywords. Instead, Alice gives the machine is given access to a list of keywords that they are able to search for. When Bob sends an email, he then selects a number of words to be keywords, and sends them using a *Public-Key Encryption with keyword Search* (PEKS) of each keyword. When the server receives the email, it checks if any of the keywords Alice has given it the ability to search for are within the list of keywords sent by Bob.

## 2 Definitions

**Definition 1.** *A Non-interactive public key encryption with keyword search scheme consists of the following polynomial time randomised algorithms:*

1. **KeyGen( $s$ ):** *Given a security parameter  $s$ , produce public and private keys  $A_{pub}, A_{priv}$  respectively.*

2.  $\text{PEKS}(A_{\text{pub}}, W)$ : Given a public key  $A_{\text{pub}}$  and keyword  $W$ , produce a searchable encryption of  $W$ .
3.  $\text{Trapdoor}(A_{\text{priv}}, W)$ : Given a private key  $A_{\text{priv}}$  and keyword  $W$ , produce a trapdoor  $T_W$ .
4.  $\text{Test}(A_{\text{pub}}, S, T_W)$ : Given a public key  $A_{\text{pub}}$ , searchable encryption  $S = \text{PEKS}(A_{\text{pub}}, W')$  and trapdoor  $T_W = \text{Trapdoor}(A_{\text{priv}}, W)$ , output 'yes' if  $W = W'$  and 'no' otherwise.

When Bob sends an email  $\text{msg}$  with keywords  $W_1, \dots, W_k$  to Alice, he computes the searchable encryptions for each keyword and sends the following:

$$[E_{A_{\text{pub}}}[\text{msg}], \text{PEKS}(A_{\text{pub}}, W_1), \dots, \text{PEKS}(A_{\text{pub}}, W_k)]$$

The email is received by Alice's server, which has trapdoors  $T_{W'_1}, \dots, T_{W'_l}$  where  $T_{W'} = \text{Trapdoor}(A_{\text{priv}}, W')$  for  $W'$  in keywords  $[W'_1, \dots, W'_l]$ . The server then uses *mathsf{Test}* with the trapdoors it has been given to see if the email contains one of the keywords specified by Alice.

Accompanying this definition is a security definition, specified as a PEKS Security game against an active attacker  $\mathcal{A}$ . The game is defined similarly to IND-CCA:

1. The challenger runs  $\text{KeyGen}(s)$  to generate  $A_{\text{pub}}$  and  $A_{\text{priv}}$ .  $A_{\text{pub}}$  is given to the attacker.
2. The attacker can adaptively ask for the trapdoor  $T_W$  of any keyword  $W \in \{0, 1\}^*$  of his choice.
3. At some point, the attacker  $\mathcal{A}$  sends two keywords  $W_0, W_1$ . The only restriction is that the attacker must have not previously asked for the trapdoors  $T_{W_0}$  or  $T_{W_1}$ . The challenger picks a random bit  $b \in \{0, 1\}$  and gives the attacker  $C = \text{PEKS}(A_{\text{pub}}, W_b)$ .  $C$  is the challenge PEKS.
4. The attacker can continue to ask for trapdoors  $T_W$  for any  $W$  as long as  $W \neq W_0, W_1$ .
5. The attacker eventually outputs a response bit  $b' \in \{0, 1\}$  and wins if  $b = b'$ .

The adversary's advantage for winning the game is defined as:

$$\text{Adv}_{\mathcal{A}}(s) = |\Pr[b' = b] - \frac{1}{2}|$$

**Definition 2.** A PEKS is semantically secure against an adaptive chosen keyword attack if for any polynomial time attacker  $\mathcal{A}$  we have that  $\text{Adv}_{\mathcal{A}}(s)$  is a negligible function.

### 3 Constructions

The remainder of the paper gives two constructions of public key searchable encryption schemes.

The first is based on bilinear maps. Bilinear maps are functions  $e : G_1 \times G_1 \rightarrow G_2$  for groups  $G_1, G_2$  of prime order  $p$  which are efficiently computable and satisfy the properties that for any integers  $x, y \in [1, p]$  it holds that  $e(g^x, g^y) = e(g, g)^{xy}$  and that if  $g$  is a generator of  $G_1$  then  $e(g, g)$  is a generator of  $G_2$ . Using  $e$  and two hash functions  $H_1 : \{0, 1\}^* \rightarrow G_1$  and  $H_2 : G_2 \rightarrow \{0, 1\}^{\log p}$ , a searchable encryption scheme can be devised.

The scheme is proven semantically secure against an adaptive chosen keyword attack in the random oracle model. This is proven via a reduction to the Bilinear Diffie-Hellman Problem: if an adversary  $\mathcal{A}$  can break the scheme with non-negligible advantage  $\epsilon$ , an adversary  $\mathcal{B}$  can be constructed which, given  $g, g^a, g^b, g^c \in G_1$  as input, can compute  $e(g, g)^{abc} \in G_2$  with advantage  $\frac{\epsilon}{eq_T q_{H_2}}$ , where  $q_T$  and  $q_{H_2}$  are the number of calls to `Trapdoor` and  $H_2$  respectively.

The second construction is based on any trapdoor permutation. A result from Bellare et al. states that given any trapdoor permutation, we can construct a source indistinguishable public key encryption scheme. A public key encryption scheme is source indistinguishable if, given a ciphertext, it is computationally difficult to determine the public key that encrypted it. The scheme works by  $A_{pub}$  being a set of public keys - one for each keyword - and  $A_{priv}$  being their corresponding private keys. The security of this scheme is shown by a reduction: If the keywords can be distinguished, the underlying encryption scheme is not source indistinguishable.

The advantage of this second scheme is that it does not rely on random oracles. But it has a cost in that the public and private keys grow linearly in size as the number of the keywords increase. This is improved by re-using source indistinguishable keys, which can still offer security via cover-free families. The construction uses parameters  $(d, t, k, q)$  where  $k$  is the number of words in the dictionary,  $t$  is an upper bound on the number of keyword trapdoors given to the server by Alice and  $d$  and  $q$  are two integers satisfying  $q = \lceil \frac{d}{4t} \rceil$  and  $d \leq 16t^2(1 + \frac{\log(k/2)}{\log 3})$  and is based on a lemma by Du and Hwang that there exists a deterministic algorithm that, for any  $(d, t, k, q)$  satisfying the above bounds, produces a  $q$ -uniform  $t$ -cover-free family. A reduction shows that if the PKES scheme has an adversary with advantage  $\epsilon$ , then there exists an algorithm that breaks source indistinguishable public key encryption scheme and aborts with probability  $\frac{1}{\text{poly}(t, q, d)}$ . Running this algorithm repeatedly until it doesn't abort yields an expected polynomial time adversary with advantage  $\frac{\epsilon}{q^2}$ .

## 4 Importance and Impact

Boneh et al. were not the first people to investigate the topic of searchable encryption[4], but they were the first to devise formal definitions for both the functionality and security behind searchable encryption schemes.

Work inspired by this paper has focused on two general themes. The first theme is improving the ideas proposed in the original paper. One example of this is Abdalla et al.[1], who looked at the concept of consistency within the schemes, where if  $\text{Test}(m, C) = \text{'yes'}$  and  $\text{Test}(m', C) = \text{'yes'}$  then  $m = m'$ . Baek et al.[2] improved the efficiency of the bilinear maps construction by removing the need for the trapdoors to be sent from Alice to the email server via a secure (encrypted and authenticated) channel. And both papers investigated extending schemes so that keywords and trapdoors are only temporary, and expire after a given time such as the end of the day. The benefit of public key encryption with temporary keyword search (PETKS) schemes is that when a server is given a trapdoor for a keyword, it is not able to learn if previous ciphertexts contain that keyword.

## References

- [1] Michel Abdalla et al. “Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions”. English. In: *Advances in Cryptology - CRYPTO 2005*. Ed. by Victor Shoup. Vol. 3621. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 205–222. ISBN: 978-3-540-28114-6. DOI: 10.1007/11535218\_13. URL: [http://dx.doi.org/10.1007/11535218\\_13](http://dx.doi.org/10.1007/11535218_13).
- [2] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. “Public Key Encryption with Keyword Search Revisited”. English. In: *Computational Science and Its Applications - ICCSA 2008*. Ed. by Osvaldo Gervasi et al. Vol. 5072. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 1249–1259. ISBN: 978-3-540-69838-8. DOI: 10.1007/978-3-540-69839-5\_96. URL: [http://dx.doi.org/10.1007/978-3-540-69839-5\\_96](http://dx.doi.org/10.1007/978-3-540-69839-5_96).
- [3] Dan Boneh et al. *Public Key Encryption with keyword Search*. Cryptology ePrint Archive, Report 2003/195. <http://eprint.iacr.org/>. 2003.
- [4] Dawn Xiaoding Song, D. Wagner, and A. Perrig. “Practical techniques for searches on encrypted data”. In: *Security and Privacy, 2000. S P 2000. Proceedings. 2000 IEEE Symposium on*. 2000, pp. 44–55. DOI: 10.1109/SECPRI.2000.848445.