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

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

April 3, 2015

1 Introduction

Public Key Encryption with keyword Search 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 any more information about the email or other keywords contained within.

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 consist 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. $\mathsf{PEKS}(A_{pub}, W)$*: Given a public key* A_{pub} *and keyword* W*, produce a searchable encryption of* W*.*

3. $\mathsf{Trapdoor}(A_{priv}, W)$*: Given a private key* A_{priv} *and keyword* W*, produce a trapdoor* T_W*.*

4. $\mathsf{Test}(A_{pub}, S, T_W)$*: Given a public key* A_{pub}*, searchable encryption* $S = \mathsf{PEKS}(A_{pub}, W')$ *and trapdoor* $T_W = \mathsf{Trapdoor}(A_{priv}, W)$*, output 'yes' if* $W = W'$ *and 'no' otherwise.*

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

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

The email is received by Alice's server, which has trapdoors $T_{W_1'}, ..., T_{W_l'}$ where $T_{W'} = \mathsf{Trapdoor}(A_{priv}, W')$ for W' in keywords $[W_1', ..., W_l']$. The server then uses $mathsfTest$ 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 $\mathcal{A}$:

1. The challenger runs $\mathsf{KeyGen}(s)$ to generate A_{pub} and A_{priv}. A_{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 $C = \mathsf{PEKS}(A_{pub}, W_b)$. 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:

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

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