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

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

April 23, 2015

1 Introduction

Public Key Encryption with keyword Search[5] 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 an email has the keyword 'Urgent'. When the server receives an email, it checks if it has the keyword: 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 (or 'searchable encryption') schemes mean that the server can check a ciphertext for keywords without learning what the other keywords are, or any information about the plaintext besides whether or not it contains those keywords.

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 access to a list of keywords which can be searched for. When Bob sends an email, he selects a number of 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 in its collection are in 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}.*

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 $T_{W'} = \mathsf{Trapdoor}(A_{priv}, W')$. The server then uses Test to see if the email contains one of the keywords in its database.

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, where the goal is for an adversary to distinguish between one of two keywords without the trapdoors of either keyword:

1. The challenger runs $\mathsf{KeyGen}(s)$ to generate A_{pub} and A_{priv}. A_{pub} is given to the attacker.

2. The attacker $\mathcal{A}$ sends two keywords W_0, W_1 to the challenger. The challenger picks a random bit $b \in 0, 1$ and returns $C = \mathsf{PEKS}(A_{pub}, W_b)$.

3. The attacker eventually outputs a response bit $b' \in \{0, 1\}$ and wins if $b = b'$.

Throughout the whole game, the adversary has access to a $\mathsf{Trapdoor}$ oracle $\mathcal{O}_{\mathsf{Trapdoor}}$. $\mathcal{A}$ is allowed to adaptively send words W to $\mathcal{O}_{\mathsf{Trapdoor}}$ at any point in the game and the oracle will return $T_W = \mathsf{Trapdoor}(A_{priv}, W)$. The only caveat is that $W \notin \{W_0, W_1\}$, otherwise $\mathcal{A}$ could simply request the trapdoor T_{W_0} and return 0 if $\mathsf{Test}(A_{pub}, C, T_{W_0})$ returns true and 1 otherwise.

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.*

Note that since the adversary has A_{pub}, $\mathcal{A}$ can adaptively call PKES a polynomial number of times. Thus in order for $\mathcal{A}$ to lose this game, PKES must be a random function that cannot be exhaustively searched in polynomial time.

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 these properties:

- For integers $x, y \in \{1, ..., p\}$, $e(g^x, g^y) = e(g, g)^{xy}$.
- 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 in polynomial time with non-negligible advantage ϵ, 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.[3] states that given a 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 its proof of security 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. Instead of using a single source indistinguishable public key for each keyword, we use a set of keys. These keys are combined to form a cover-free family, which allows us to re-use the keys. As long as Alice only creates t trapdoors for the server, a cover-free family can be created for each keyword using q keys, where q is a parameter defined by[8].

A reduction shows that if the PKES scheme has an adversary with advantage ϵ, then there exists an algorithm that breaks source indistinguishable public key encryption scheme and aborts with probability $\frac{1}{poly(t,q,d)}$, where q and d are parameters from[8]. Running this algorithm repeatedly until it doesn't abort yields an expected polynomial time adversary with advantage $\frac{\epsilon}{q^2}$. This reduces the key size, but the scheme is now limited as Alice can only create t trapdoors.

4 Importance and Impact

Boneh et al. were not the first people to investigate the topic of searchable encryption, which was instead done by Song, Wagner and Perrig[12]. But Song, Wagner and Perrig's research looked into private key searchable encryption, as opposed to public key. Boneh et al. were also the first to provide definitions of syntax and security for searchable encryption, whereas Song, Wagner and Perrig instead argued that their scheme was secure by proving it was computationally indistinguishable from random text.

Boneh et al. were also not the only people working on this subject at the time. A similar concept was developed by Goh[9] and submitted to the Cryptology ePrint Archive a matter of weeks later. The authors acknowledged each other's work in later versions of both papers.

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 $\mathsf{Test}(m, C) =$ 'yes' and $\mathsf{Test}(m', C) =$ 'yes' then $m = m'$. Baek, Safavi-Naini and Susilo[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.

The second theme is investigating what other predicates can be performed on encrypted data. The above papers by Song et al. and Boneh et al. provided methods for testing equality predicates on encrypted data. Boneh and Waters[4] expanded this by developing encryption schemes catering for three more predicates: Comparison, subset and conjunctive queries. Katz, Sahai and Waters[10] developed an encryption scheme that allowed the evaluation of inner products over $\mathbb{Z}_{\mathbb{N}}$ for some large integer N, which in turn allows the evaluation of polynomial equations and disjunctions.

It is worth noting that while Public Key Encryption with keyword Search has had a large impact in the theory world, it has seen little utilisation in

practice. In particular, I have been unable to find a single email system which implements a PEKS scheme, or even a paper with an empirical analysis of such a scheme. A possible reason for this might be because of the practicality of the constructions for this problem; we are unlikely to see data large enough for this to be beneficial sent via email.

But not all is lost; since 2004 when the paper was first published, a new paradigm has appeared. Boneh et al.'s work has given rise to investigations around the topic of keyword searches on encrypted data which is stored on a remote server[7]. In particular, keyword searches on data stored in the cloud is a growing trend, with research expaning to problems such as approximate keyword searching[11] and searching files for multiple keywords[6]. This area has seen experimental analysis, due to the larger amounts of data that can be stored in the cloud.

5 Conclusion

With over 1100 citations on Google Scholar, Public Key Encryption with keyword Search has had a large impact on the theory of cryptography. While it was not the first paper to investigate to topic of searchable encryption, it provided a framework for searchable encryption schemes, along with a means of defining security. This paper–along with its symmetric predecessor–has led to a large body of research in granting permission to make certain computations on encrypted data.

Although the original proposed application of an email server has seen little support, subsequent schemes have seen benefits in practice on other problems. Most notably, the growth of cloud computing has given rise to papers investigating computation on encrypted data that is stored remotely. As more and more companies switch to the cloud for their computation needs, we are likely to see even more research in this area.

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.

[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.

[3] Mihir Bellare et al. "Key-Privacy in Public-Key Encryption". English. In: *Advances in Cryptology–ASIACRYPT 2001*. Ed. by Colin Boyd. Vol. 2248. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, pp. 566–582. ISBN: 978-3-540-42987-6. DOI: 10.1007/3-540-45682-1_33. URL: http://dx.doi.org/10.1007/3-540-45682-1_33.

[4] Dan Boneh and Brent Waters. "Conjunctive, Subset, and Range Queries on Encrypted Data". English. In: *Theory of Cryptography*. Ed. by SalilP. Vadhan. Vol. 4392. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, pp. 535–554. ISBN: 978-3-540-70935-0. DOI: 10.1007/978-3-540-70936-7_29. URL: http://dx.doi.org/10.1007/978-3-540-70936-7_29.

[5] Dan Boneh et al. *Public Key Encryption with keyword Search*. Cryptology ePrint Archive, Report 2003/195. http://eprint.iacr.org/. 2003.

[6] Ning Cao et al. "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data". In: *Parallel and Distributed Systems, IEEE Transactions on* 25.1 (Jan. 2014), pp. 222–233. ISSN: 1045-9219. DOI: 10.1109/TPDS.2013.45.

[7] Yan-Cheng Chang and Michael Mitzenmacher. "Privacy Preserving Keyword Searches on Remote Encrypted Data". English. In: *Applied Cryptography and Network Security*. Ed. by John Ioannidis, Angelos Keromytis, and Moti Yung. Vol. 3531. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 442–455. ISBN: 978-3-540-26223-7. DOI: 10.1007/11496137_30. URL: http://dx.doi.org/10.1007/11496137_30.

[8] Ding-Zhu Du and Frank K. Hwang. *Combinatorial Group Testing and Its Applications*. World Scientific, 1993.

[9] Eu-Jin Goh. *Secure Indexes*. Cryptology ePrint Archive, Report 2003/216. http://eprint.iacr.org/. 2003.

[10] Jonathan Katz, Amit Sahai, and Brent Waters. "Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products". English. In: *Journal of Cryptology* 26.2 (2013), pp. 191–224. ISSN: 0933-2790. DOI: 10.1007/s00145-012-9119-4. URL: http://dx.doi.org/10.1007/s00145-012-9119-4.

[11] Jin Li et al. "Fuzzy Keyword Search over Encrypted Data in Cloud Computing". In: *INFOCOM, 2010 Proceedings IEEE*. Mar. 2010, pp. 1–5. DOI: 10.1109/INFCOM.2010.5462196.

[12] 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.