Final Project: Conjunctive, Subset, and Range Queries on Encrypted Data

Julia Matsieva
Modern Cryptography – Professor Rogaway
March 17, 2012

Abstract

The 2006 paper *Conjunctive, Subset, and Range Queries on Encrypted Data* by Dan Boneh and Brent Waters addresses the problem of predicate evaluation on encrypted data in the public key setting – one that does not leak any additional information about the data. The authors provide both a security notion for such queries and an efficient construction based on the bilinear and 3-party composite Diffie-Hellman assumptions. The goal of this write-up is to present the security notion devised by Boneh and Waters and describe their construction at a high level.

Introduction

This paper is motivated by situations in which a non-malicious party may wish to learn information of interest about encrypted data. For example, one may want to encrypt student test results for privacy reasons; however, a statistical program may want to know some information the data. For example, it may want to count how many students scored over 90%, which is a fairly innocuous goal. Yet we would not want to allow such a program to decrypt all of the data and find out the identities of the students. Therefore, it is desirable to encode the "$x \geq 90\%$" predicate in a token TK_p to allow the program to compute this information. Furthermore, if the predicate contains a conjunction $P_1 \wedge P_2$, we do not want the encryption scheme to leak which predicate satisfied the expression.

Boneh and Waters describe the syntactic components of a searchable encryption scheme defined with respect to a set of predicates Φ; they do not provide a decryption algorithm, arguing that it can always be added by using a standard public key encryption system. The authors then define two similar notions of security for a searchable system, both of a find-then-guess flavor, that capture the intuition that an adversary should not be able to distinguish encrypted data values that agree on the predicate; then, they provide a trivial construction of a searchable encryption system based on standard pubic key encryption, showing it to produce very long ciphertexts. They then construct a Hidden Vector Encryption system based on the bilinear and 3-party composite Diffie-Hellman assumptions which can be used to encode conjunctions of range and subset queries, and demonstrate significant improvement in efficiency.

Searchable Encryption

Let Σ be the space containing data values and $\mathcal{M}$ be a space of messages that can be associated with the data. Given a set of predicates Φ, a Φ-searchable encryption scheme is a four-tuple of algorithms $(Setup, Encrypt, GenToken, Query)$ that behave as follows:

- $Setup(\lambda)$ is a probabilistic algorithm that generates keys PK, SK based on the security parameter λ.
- $Encrypt(PK, I, M)$ encrypts the pair (I, M) where $I \in \Sigma$ is a data value and $M \in \mathcal{M}$ is a safe-to-reveal message that provides some additional information about the data. In a scenario where we are only interested in whether a predicate is satisfied, $\mathcal{M}$ may be limited to just {true}.
- $GenToken(SK, \langle P \rangle)$ takes as input the secret key and a description of the predicate $P \in \Phi$ and generates the corresponding token TK_P.
- $Query(TK, C)$ describes how a token for predicate $P \in \Phi$ can be used to test the ciphertext C, outputting message $M \in \mathcal{M}$.

This description is subject to the **correctness condition**, which states that for all data-message pairs $(I, M) \in \Sigma \times \mathcal{M}$ and all predicates $P \in \Phi$, if we run the setup, encryption and token-generation algorithms

$$(PK, SK) \xleftarrow{\$} Setup(\lambda); C \xleftarrow{\$} Encrypt(PK, I, M); TK \xleftarrow{\$} GenToken(SK, \langle P \rangle)$$

then the query algorithm will behave as expected, revealing M if the predicate P holds for I and failing otherwise.

- If $P(I) = 1$ then $Query(TK, C) = M$
- If $P(I) = 0$ then $\Pr[Query(TK, C) = \perp] > 1 - \epsilon(\lambda)$, where ϵ is a negligible function.

Security

The paper defines the security of a Φ-searchable encryption system through the following game, which consists of several phases:

- **Setup** – The $Setup(\lambda)$ algorithm is run by the simulation environment and PK is passed to the adversary.
- **Query Phase I** – The adversary is allowed adaptively request tokens for the predicates $P_1, \dots P_{q'} \in \Phi$; these predicate queries are answered by running $GenToken(SK, \langle P_i \rangle)$.
- **Challenge** – The adversary chooses two data-message pairs (I_0, M_0) and (I_1, M_1) subject to the restrictions that
  - $Pi(I_0) = P_i(I_1)$ for all $i = 1, \dots, q'$ – the data values agree on all previously computed predicates.
  - If $M_0 \neq M_1$ then $P_i(I_0) = P_i(I_1) = 0$ for all $i = 1, \dots, q'$ – the tokens do not directly distinguish M_0 from M_1.
- **Query Phase II** – The adversary can request more tokens for predicates $P_{q'+1}, \dots, P_q \in \Phi$ as long as they adhere to the above restrictions.
- **Guess** – We flip a coin $\beta \in \{0, 1\}$ and give $C_* \xleftarrow{\$} Encrypt(PK, I_\beta, M_\beta)$ to the adversary, who returns a guess $\beta' \in \{0, 1\}$. The advantage of adversary A is then given by

$$\mathbf{Adv}_{\mathcal{E}}^{QU}(A) = \left| \Pr[\beta' = \beta] - \frac{1}{2} \right|$$

  and a Φ-searchable scheme $\mathcal{E}$ is considered **secure** if $\mathbf{Adv}_{\mathcal{E}}^{QU}$ is a negligible function of λ.

This is a familiar, find-then-guess type of game; the authors do not justify their preference of this style of game to a left-or-right construction or something else entirely. The authors also define **selective security** to be a slightly weaker version of the game above – in this game, everything is the same except the adversary commits to I_0, I_1 during the Setup phase.

Trivial Construction

The authors provide a trivial construction of a Φ-searchable security system $\mathcal{E}_{TR}$ with t predicates based on any existing public key system $\mathcal{E} = (Setup', Encrypt', Decrypt')$. The searchable keys PK, SK are generated simply by running $Setup'(\lambda)$ t times and the encryption algorithm $Encrypt(PK, I, M)$ computes ciphertext $C_1 C_2 \dots C_t$ where each C_j is computed by running $Encrypt'(M)$ if P_j holds for I and $Encrypt'(\perp)$ otherwise. The token TK_{P_j} is the index j of the predicate in Φ and the secret key SK_j corresponding to that predicate and $Query(TK, C)$ is trivially accomplished by decrypting C_j with SK_j. This scheme is shown to be secure as a searchable scheme assuming $\mathcal{E}$ is secure against chosen message attacks using a hybrid argument with a chain of $t + 1$ experiments, where the i^{th} experiment encrypts M_0 for all P_j that hold for I_0 if $j \geq i$ and M_1 for all P_j that hold for I_1 if $j < i$. If $\textsc{Exp}_{QU}^{i}$ is the probability that the adversary guesses $\beta' = 1$ in experiment i then its advantage is given by the difference in the outer experiments

$$\mathbf{Adv}_{\mathcal{E}_{TR}}^{QU} = |\textsc{Exp}_{QU}^{1}(A) - \textsc{Exp}_{QU}^{t+1}(A)| \leq \sum_{i=1}^{t} |\textsc{Exp}_{QU}^{i}(A) - \textsc{Exp}_{QU}^{i+1}(A)|$$

and we count on $|\textsc{Exp}_{QU}^{i}(A) - \textsc{Exp}_{QU}^{i+1}(A)|$ to be negligible since $\mathcal{E}$ is assumed to be semantically secure in the public key setting. Boneh and Waters then point out that, given data space $\Sigma = \{1, \dots n\}^w$ and the set $\Phi_{n,w}$ of all comparison predicates over $\Sigma = \{1, \dots, n\}^w$

$$P_{a_1, \dots a_w}(x_1, \dots, x_w) = x_j \geq a_j \quad \text{for all } j = 1, \dots, w$$

this system will produce ciphertexts of length $O(n^w)$; this motivates the construction of a scheme that is more efficient.

Hidden Vector Encryption

Boneh and waters create a general scheme for searchable encryption of vector-like data that can be used to easily encrypt equality or range predicates that test if a data value falls within a certain range. They also demonstrate that the scheme can be extended to encode more general subset predicates, which ask if a certain data value is a member of a subset. Their construction uses two flavors of the Diffie-Hellman assumption: the *bilinear Diffie-Hellman*

assumption states that, if $\mathcal{G}$ is a group generator which outputs tuple (p, q, G, H, m) where $p \neq q$ are distinct primes and G, H are two groups order $n = pq$ and m is a non-degenerate, bilinear map $m : G \times G \to H$, then given random element $g_p \in G_p$ where G_p is a subgroup of order p of G and the values of g_p^a, g_p^b, g_p^c, it is difficult to distinguish $m(g_p, g_p)^{abc}$ from an element randomly chosen from H where $a, b, c \in \mathbb{Z}$. The *composite 3-party Diffie-Hellman assumption* states that, given $g_p^a, g_p^b, g_p^{ab} \cdot R_1, g_p^{abc} \cdot R_2$, it is difficult to identify $g_p^c \cdot R_3$ where R_1, R_2, R_3 are randomly selected from G_q, a subgroup of order q of G; that is, that it is difficult to test for Diffie-Hellman tuples in the order-p subgroup if the elements have a random component in the order-q subgroup.

The authors give a construction for defining general equality predicates that can also be used for comparison and subset queries. The predicates are constructed over $\Sigma_* = \Sigma \cup \{*\}$, where Σ is now a set of predicate values and * is a wildcard symbol that indicates that we do not care about a value. For example, an equality predicate P_σ is defined over Σ_*^ℓ so that if $x \in \Sigma^\ell$ then $P_\sigma(x)$ is 1 on all non-* coordinates j of σ where $\sigma_j = x_j$ and zero otherwise. The construction of a secure HVE system views Σ as $\mathbb{Z}_m$ and the message space $\mathcal{M}$ as a small subset of the group H described above; specifically, it stipulates that $|\mathcal{M}| < |H|^{1/4}$. A high-level description of the algorithm follows:

- *Setup* generates a bilinear group with $\mathcal{G}$ and creates the secret key by choosing a random triple from G_p^3 for each of ℓ predicates; it then creates the public key by choosing $3\ell + 1$ random blinding factors in G_q and multiplying them with some of the previously chosen values. The algorithm also chooses two random elements g, v from G_p and publishes the result of the bilinear map $m(g, v)^\alpha$ as part of the public key where α is an exponent chosen from $\mathbb{Z}_p$.
- $Encrypt(PK, I, M)$ is performed by picking random elements in G_q by raising g_q to random exponents from $\mathbb{Z}_n$ and the predicate is encrypted by multiplying those values with components of the public key. The algorithm encrypts the message using the value of the bilinear map.
- $GenToken(SK, \langle P \rangle)$ generates a token for a predicate $P_\sigma \in \Sigma_*^\ell$ based on the product of the secret key at the indices at which $\sigma \neq *$, along with randomly chosen exponents.
- $Query(TK, C)$ is then able to compute the message through division, using values and indicies known from the token and exponentiation properties of the bilinear map.

Although the details are not shown, one of the elegant parts of this construction is that it gives part one of the correctness condition almost automatically, leveraging the property that $m(h_p, h_q) = 1$ if $h_p \in G_p$ and $h_q \in G_q$. The second part of the correctness condition is established from the stipulation that $|\mathcal{M}| < |H|^{1/4} < (pq)^{1/4}$. Boneh and Waters estabilish the security of HVE using a hybrid argument, employing the Diffie-Hellman assumptions outlined above by establishing games that differ in a certain index of the predicate vector.

Application

The Hidden Vector Encryption system can be used to construct predicate and range queries with relatively compact ciphertexts and tokens; compared to the results of the trivial construction, this system produces ciphertexts and tokens for the predicate family $\Phi_{n,w}$ of size $O(nw)$ and $O(w)$, respectively. The $Encrypt(PK, I, M)$ algorithm builds a vector $\sigma(i, j)$ for $I = (x_1, \ldots, x_w) \in \{1, \ldots, n\}^w$, setting $\sigma_{i,j} = j \geq x_i$ interpreted as a boolean value and run $Encrypt_{HVE}$ on this vector, which is now defined over $\{0, 1\}$. $GenToken(SK, \langle P_{\overline{a}} \rangle)$ where $\overline{a} = \{a_1, \ldots, a_w\} \in \{1, \ldots, n\}^w$ will define a predicate vector $\sigma_{*i,j} = 1$ if $x_i = j$ and * otherwise. Thus, we can see how this algorithm significantly compacts the values of the encryption while retaining the necessary information to compute the query. The authors show how this construction can be used to build conjunctions of range, as well as subset, queries.

Conclusion

This paper introduces and motivates the notion of searchable encryption schemes, pointing out scenarios in which searching encrypted data would be useful. The authors give a syntactic notion of a searchable encryption system, which should generate predicate tokens that could then be used to run queries on encrypted data. They provide a security notion built on the intuition that the knowledge that can be gained about the data from the encryption scheme should be limited to only that which is revealed by the evaluating the predicates. The authors give a trivial construction for the scheme to demonstrate existence and then construct a more efficient scheme using two variants of the Diffie-Hellman assumption while leveraging other group-theoretic properties.