Threshold Privacy Preserving Keyword Searches

Peishun Wang[1], Huaxiong Wang[1,2], Josef Pieprzyk[1]

[1]Centre for Advanced Computing – Algorithms and Cryptography
Department of Computing, Macquarie University
Sydney, NSW 2109, Australia

[2]Division of Mathematical Sciences
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore

SOFSEM'08 – Jan. 2008

Outline

Applications

- **Scenario:** users of an organization wish to outsource storage of their sensitive information to a large database server. However, the server storing the data is untrusted, and other members of the organization alone cannot be trusted. Hence, all data have to be submitted in an encrypted form. Only the manager of the organization has the right to access all data, and any member of the organization must collaborate with others to search for the desired data.
- Examples: big intelligence or police organizations, such as CIA, FBI.

Applications

- **Scenario:** users of an organization wish to outsource storage of their sensitive information to a large database server. However, the server storing the data is untrusted, and other members of the organization alone cannot be trusted. Hence, all data have to be submitted in an encrypted form. Only the manager of the organization has the right to access all data, and any member of the organization must collaborate with others to search for the desired data.
- Examples: big intelligence or police organizations, such as CIA, FBI.

Existing Schemes

- Keyword search over encrypted data for a single-user;
- Keyword search over encrypted data for multi-users;
- Threshold Privacy Preserving Keyword Searches (TPPKS)**?** $\cdots$ No scheme!

Existing Schemes

- Keyword search over encrypted data for a single-user;
- Keyword search over encrypted data for multi-users;
- Threshold Privacy Preserving Keyword Searches (TPPKS)**?** $\cdots$ No scheme!

- Formal definitions for TPPKS:
  1. System Instantiation,
  2. Key Distribution,
  3. Data Encryption and Secure Index Generation,
  4. Trapdoor Generation and Data Search,
  5. Data Decryption
- Privacy requirement.
- A TPPKS scheme.

Three well-known hardness assumptions: Discrete Logarithm (DL), Decisional Diffie-Hellman (DDH) and Computational Diffie-Hellman (CDH).

Definition (DL Assumption)

Given a finite cyclic group $G = \langle g \rangle$ of prime order q with a generator g. For a given random number $x \in G$, the DL problem is to find an integer t ($0 \leq t < q$) such that $x = g^t$. An algorithm $\mathcal{A}$ is said to have an ϵ-advantage in solving the DL problem if

$$Pr[\mathcal{A}(g, g^t) = t] > \epsilon.$$

The DL assumption holds in G if no PPT algorithm has advantage at least ϵ in solving the DL problem in G.

Three well-known hardness assumptions: Discrete Logarithm (DL), Decisional Diffie-Hellman (DDH) and Computational Diffie-Hellman (CDH).

Definition (DL Assumption)

Given a finite cyclic group $G = \langle g \rangle$ of prime order q with a generator g. For a given random number $x \in G$, the DL problem is to find an integer t ($0 \leq t < q$) such that $x = g^t$. An algorithm $\mathcal{A}$ is said to have an ϵ-advantage in solving the DL problem if

$$Pr[\mathcal{A}(g, g^t) = t] > \epsilon.$$

The DL assumption holds in G if no PPT algorithm has advantage at least ϵ in solving the DL problem in G.

Definition (DDH Assumption)

Let $G = \langle g \rangle$ be a cyclic group of prime order q and g a generator of G. The DDH problem is to distinguish between triplets of the form (g^a, g^b, g^{ab}) and (g^a, g^b, g^c), where $a, b, c \xleftarrow{R} Z_q$. An algorithm $\mathcal{A}$ is said to have an ϵ-advantage in solving the DDH problem if

$$|Pr[\mathcal{A}(g^a, g^b, g^{ab}) = yes] - Pr[\mathcal{A}(g^a, g^b, g^c) = yes]| > \epsilon.$$

The DDH assumption holds in G if no PPT algorithm has advantage at least ϵ in solving the DDH problem in G.

Definition (CDH Assumption)

Let $G = \langle g \rangle$ be a cyclic group of prime order q and g a generator of G. The CDH problem is to compute g^{ab} for given $g, g^a, g^b \in G$, where $a, b \xleftarrow{R} Z_q$. An algorithm $\mathcal{A}$ is said to have an ϵ-advantage in solving the CDH problem if

$$Pr[\mathcal{A}(g, g^a, g^b) = g^{ab}] > \epsilon.$$

The CDH assumption holds in G if no PPT algorithm has advantage at least ϵ in solving the CDH problem in G.

Let G_1, G_2 be two cyclic groups of some large prime order q. A bilinear pairing is defined as a function $e : G_1 \times G_1 \rightarrow G_2$ with the following properties:

1. Bilinear: for all $P, Q \in G_1$ and $a, b \in Z_q$, $e(aP, bQ) = e(P, Q)^{ab}$.
2. Non-degenerate: there exist $P, Q \in G_1$ such that $e(P, Q) \neq 1$, where 1 is the identity of G_2.
3. Computable: for all $P, Q \in G_1$, $e(P, Q)$ is computable in polynomial time.

A Bilinear Pairing Parameter Generator is defined as a polynomial-time algorithm $\mathcal{BPPG}$, which takes as input a security parameter k and outputs a uniformly random tuple (e, G_1, G_2, q) of bilinear pairing parameters.

- The manager C runs a $\mathcal{BPPG}$ with a security parameter k to generate bilinear pairing parameters (q, G_1, G_2, e), where G_1 is an additive group of large prime order q with a generator P, $q' = \frac{q-1}{2}$ is also a prime, G_2 is a multiplicative group of order q and the DL and CDH assumptions hold in both G_1 and G_2.
- C chooses two cyclic groups: a multiplicative group G of prime order q with a generator g, in which the DDH assumption holds, and an additive group $G_0 = \langle P_0 \rangle$ of prime order q', in which the computation is based on the modulus q and the DL assumption holds.

- C chooses three cryptographic hash functions:

  $$H : \{0,1\}^* \to Z_q^*, H_1 : \{0,1\}^* \to G_1, \text{ and } H_2 : G_2 \to \{0,1\}^l,$$

  where $\{0,1\}^l$ is the plaintext space.
- C chooses $Q \xleftarrow{R} G_1$, and five different values $\lambda, \sigma, r, d, s \xleftarrow{R} Z_q^* \setminus \{1\}$ and computes $P' = \lambda P, Q' = (\lambda - \sigma)Q, g' = g^{\frac{r}{d}}, \tilde{g} = g'^s$ and $u = \frac{s}{d}$.
- C publishes system's public parameters $\{e, G, G_0, G_1, G_2, q, q', g, g', \tilde{g}, u, P_0, P, P', Q, Q', H, H_1, H_2\}$ and keeps $\{\lambda, \sigma, r, d, s\}$ secret.

- Every member M_i $(1 \leq i \leq n)$ has an unique identity number ID_i, and C computes $x_i = H(ID_i)$ $(i = 1, \ldots, n)$.
- C randomly generates three secret polynomials f_0, f_1, f_2 of degree $t-1$ of the form

$$\begin{aligned} f_0(x) &= r + a_1^{(0)}x + \cdots + a_{t-1}^{(0)}x^{t-1}, \\ f_1(x) &= d + a_1^{(1)}x + \cdots + a_{t-1}^{(1)}x^{t-1}, \\ f_2(x) &= \sigma + a_1^{(2)}x + \cdots + a_{t-1}^{(2)}x^{t-1}, \end{aligned}$$

where $\{a_j^{(i)}\}$ $(i = 0, 1, 2; j = 1, \cdots, t-1)$ are secretly random numbers in Z_q^*.

- For every member M_i, C lets $r_i = f_0(x_i), d_i = f_1(x_i), \sigma_i = f_2(x_i)$, and delivers the secret shares r_i, d_i, σ_i to M_i ($1 \leq i \leq n$) via a secure channel.
- C computes

$$\nu_i^{(r)} = r_i H_1(ID_i), \nu_i^{(d)} = d_i H_1(ID_i), \text{ and } \nu_i^{(\sigma)} = e(\sigma_i H_1(ID_i), P),$$

and publishes $(\nu_i^{(r)}, \nu_i^{(d)}, \nu_i^{(\sigma)})$ as the verification keys of M_i ($1 \leq i \leq n$).

- A user encrypts her data $\mathcal{M}$ as follows: chooses $\gamma \xleftarrow{R} Z_q^* \setminus \{1\}$, computes

$$X = \gamma P \text{ and, } Y = \mathcal{M} \oplus H_2(e(Q, P')^{\gamma}),$$

  and let $R = (X, Y)$ be the ciphertext of $\mathcal{M}$.
- The user chooses $\alpha \xleftarrow{R} Z_q^* \setminus \{1\}$ and computes $W' = g^{-\alpha}$ and $\bar{W} = g'^{\alpha}$. For each keyword w_j in $\mathcal{M}$, the user computes $W_j = \tilde{g}^{\alpha H(w_j) P_0}$.
- The user lets $I = \{W', \bar{W}, W_1, W_2, \ldots, W_m\}$ be the secure index of the data $\mathcal{M}$, and uploads $\{I, R\}$ to the server.

- The t members $\{M_{i_j}\}_{j=1,\cdots,t}$ with identities $\{ID_{i_j}\}_{j=1,\cdots,t}$ together compute

$$c_{i_j} = \prod_{m'=1, m' \neq j}^{t} \frac{H(ID_{i_{m'}})}{H(ID_{i_{m'}}) - H(ID_{i_j})},$$

choose $\beta \xleftarrow{R} Z_q^* \setminus \{1\}$, and compute $A^{(0)} = \beta P_0$ in G_0. For every queried keyword $w'_{m'}$ in the queried keyword list $L' = \{w'_{m'}\}_{m'=1,\ldots,l}$ ($l \leq m$), they compute $A^{(m')} = (uH(w'_{m'}) + \beta)P_0$ in G_0.

- Each member M_{i_j} computes in G_0

$$A_{i_j}^{(0)} = c_{i_j} r_{i_j} A^{(0)} \text{ and } A_{i_j}^{(m')} = c_{i_j} d_{i_j} A^{(m')} \ (m' = 1, \cdots, l),$$

and takes them as her share of the trapdoor of L'.

- The t members verify every member's shares by checking whether it holds for $j = 1, \cdots, t$ that

$$\begin{aligned} e(H_1(ID_{i_j}), A_{i_j}^{(0)}P) &= e(c_{i_j}\nu_{i_j}^{(r)}, A^{(0)}P) \text{ and} \\ e(H_1(ID_{i_j}), A_{i_j}^{(m')}P) &= e(c_{i_j}\nu_{i_j}^{(d)}, A^{(m')}P) \; (m' = 1, \cdots, l). \end{aligned}$$

If it holds for all $j = 1, \cdots, t$, this means that all search shares are valid, then they go to next step. If it does not hold for some j, this means that M_{i_j} provides a invalid search share, then they terminate the protocol.

- The t members compute

$$A_0 = \sum_{j=1}^{t} A_{i_j}^{(0)} \text{ and } A_{m'} = \sum_{j=1}^{t} A_{i_j}^{(m')} \ (m' = 1, \cdots, l),$$

  and sends $(A_0, \{A_{m'}\}_{m'=1}^{l})$ as the trapdoor of L' to the server.
- On receiving the trapdoor, the server tests on a secure index for every $m' \in [l]$ if there exists some $i \in [m]$ such that

$$W'^{A_0} \cdot \bar{W}^{A_{m'}} = W_i.$$

  If so, the server puts the data R in a collection. After all secure indices are checked, if the collection is not empty, the server returns the collection to the member; otherwise, returns No Data Matched to the t members.

When the t members receive a data $R = (X, Y)$, they do the following.

- Each member M_{i_j} $(j = 1, \cdots, t)$ chooses $y \xleftarrow{R} Z_q^* \setminus \{1\}$, computes

$$\begin{aligned} y_{i_j}^{(1)} &= e(Q, X)^y, \\ y_{i_j}^{(2)} &= e(H_1(ID_{i_j}), P)^y, \\ \tilde{p}_{i_j} &= H(y_{i_j}^{(1)} || y_{i_j}^{(2)}), \\ z_{i_j} &= \tilde{p}_{i_j}\sigma_{i_j} + y \text{ and} \\ v_{i_j} &= e(Q, X)^{\sigma_{i_j}}, \end{aligned}$$

  and provides $\{y_{i_j}^{(1)}, y_{i_j}^{(2)}, z_{i_j}, v_{i_j}\}$ as her decryption share.

- The t members compute $\tilde{p}_{i_j} = H(y_{i_j}^{(1)} \| y_{i_j}^{(2)})$ $(j = 1, \cdots, t)$ and check whether it holds that

$$e(Q, X)^{z_{i_j}} = v_{i_j}^{\tilde{p}_{i_j}} y_{i_j}^{(1)} \text{ and } e(H_1(ID_{i_j}), P)^{z_{i_j}} = (\nu_{i_j}^{(\sigma)})^{\tilde{p}_{i_j}} y_{i_j}^{(2)}.$$

  If it holds for all $j = 1, \cdots, t$, this means that all decryption shares are valid, then they go to next step. If it does not hold for some j, this means that M_{i_j} provides a invalid decryption share, then they terminate the protocol.
- Finally, the t members compute $D_{i_j} = v_{i_j}^{c_{i_j}}$ $(j = 1, \cdots, t)$, and then output the plaintext

$$\mathcal{M} = Y \oplus H_2\left(\prod_{j=1}^{t} D_{i_j} \cdot e(Q', X)\right).$$

- GA member has an above trapdoor $(A_0, \{A_{i_j}\}_{j=1}^{l})$ for a list of keywords $L = \{w_{i_j}\}_{j=1}^{l}$, where i_j is the position where the keyword w_{i_j} appears in the secure index, this means, $\{i_j\}_{j=1}^{l} \subset [m]$.
- The member computes $A_0^{(c)} = (A_0)^l, A_1^{(c)} = \sum_{j=1}^{l} A_{i_j}$ and sends $\{A_0^{(c)}, A_1^{(c)}, i_1, \cdots, i_l\}$ as the trapdoor of L to the server.
- The sever checks if it holds that $W'^{A_0^{(c)}} \cdot \bar{W}^{A_1^{(c)}} = \prod_{j=1}^{l} W_{i_j}$ to guess whether all the keywords $\{w_{i_j}\}_{j=1}^{l}$ are in the secure index or not.

Theorem

The secret key distribution process in the proposed TPPKS is secure against impersonation and a coalition of up to $(t-1)$ adversaries.

Theorem

The secret share verification algorithms used in the proposed TPPKS are secure under the DL and CDH assumptions.

Theorem

The data cryptosystem used in the proposed TPPKS is semantically secure.

Theorem

The search process in the proposed TPPKS is semantically secure under the DDH assumption according to the security game ICLR.

- Formal definition of TPPKS and security requirement.
- A TPPKS scheme based on Shamir secret sharing, Boneh and Franklin's ID-based cryptosystem and the group computation.
- Secure is based on the assumptions of DL, DDH and CDH.
- Attractive Properties:
  1. Shares are verified without leaking any information about them,
  2. Any invalid share fails the verification,
  3. There is no information disclosed about the shares after they have been used.
- Open Problem: designing the scheme for a dynamic group.

Thank You !!!