Secure access of multiple keywords over encrypted data in cloud environment using ECC-PKI and ECC ElGamal

Sourabh Prakash, Nitish Andola, S. Venkatesan

Department of Information Technology, IIIT- Allahabad
Data Outsourcing over Cloud
Searchable Encryption

client

search query: keyword

server
Background: Searchable Encryption

• The searching and responding process of searchable encryption takes place over the encrypted data set and indexes. Performing such process over encrypted data set is constantly complex and harder as compared to normal data set.

• Therefore, the need for effective and secure searchable encryption became a major research problem, and improvement in the existing searchable encryption scheme is still a growing task for the research community.

• Usually, a data user prefers to request for a set of keywords, which actually leads to a more precise closer relation to expected data. More keywords in query help to narrow down the results more accurate.
Background: Searchable Encryption.

- Despite remarkable development in the area of searchable encryption, few unsolved issues are noticeable when deployed to the cloud environment.
- First, keyword privacy and file content privacy both are equally important in searchable encryption.
- Second, honest but curious cloud server or such adversaries can still breach privacy in a scheme which does not have high robustness against the similarity relevance of terms and files.
- Third, traditional public key based schemes have its own computational complexities and limitations that require a serious reconsideration to use. We propose a scheme to overcome the above-mentioned problems.
Key functionaries

Cloud

Revocation

Data

Registration

Key distribution

Data Owner

Data User
Basic Consideration

• Design Goal
• Threat Model
• Scoring
Proposed System model
ECEG and Homomorphism

- Algorithms mentioned here are inspired by standard elliptic curve based ElGamal encryption and decryption algorithms

Algorithm 1 Algorithm for Message Encoding to the Point over Elliptic Curve E

Input: \((m, G)\)
- \(m\): Plain Text Message value
- \(G\): The Generator Point of curve \(E\)

Output: \((P_m)\)
- \(P_m\): Point representation of message \(m\) over curve \(E\)

Step 1: Initially get the message value \(m\) and generator point \(G\)

Step 2: Compute the multiplication of message value \(m\) with the generator point \(G\);
- \(P_m = mG\), Where \(m \in \mathbb{Z}_F(p)\)
- The resultant point \(P_m\); \(P_m \in E\) is corresponding message encoded over that curve \(E\).

Algorithm 2 Elliptic Curve Based ElGamal (ECEG) Encryption

Input: \((P_k, P_m)\)
- \(P_k\): Public Key \(P_k \in E\)
- \(P_m\): The Encoded Message

Output: \(P_c(C_1, C_2)\)
- \(P_c\) is a cipher text pair \(P_c \in E\)

Step 1: Initially selects a random integer \(r \in [1, n - 1]\), \(n\) is order of curve \(E\), \(n \in E\).

Step 2: Compute \(C_1 = rG\)

Step 3: Encrypt message point with the public key
- \(C_2 = P_m + rP_k\)

Step 4: Return \(P_c(C_1, C_2)\), is associated cipher-text pair.
System Scheme

Set of files $C = \{f_1, f_2, \ldots, f_n\}$

Encrypted $C'$: $C' = \{f'_1, f'_2, \ldots, f'_n\}$

Keyword Set $W = \{w_1, w_2, \ldots, w_m\}$

Index $I = \{I_1, I_2, \ldots, I_n\}$

Where,

$I_i = \{FID_i, tf-idf_{jw1}, tf-idf_{jw2}, \ldots, tf-idf_{jw1}\}$

Encrypted Encrypt $(Pk, I) = I'$

$S = I' . Q$

$S = \{(FID1, r1), (FID2, r2), \ldots, (FIDn, rn)\}$

Where, $r_i = r_{fiQ} = I_i . Q$

Keyword Set $W = \{w_1, w_2, \ldots, w_m\}$

Set of Requested Keywords;

$Q = \{Q_1, Q_2, \ldots, Q_m + 1\}$

If $Q[i] = 1$, the $i$th word is being requested;
and if $Q[i] = 0$, the $i$th word is not requested.
Proof of Additive Homomorphism

• Let's consider there are two (2) keywords requested, and for a file score of \( w_1 \) is encoded to point \( P_{m_1} \), and \( w_2 \) is to corresponding point \( P_{m_2} \).

• The encryption of \( P_{m_1} \) and \( P_{m_2} \) returns cipher-text \( P_{c_1} \) and \( P_{c_2} \).

Where:

\[
\begin{align*}
\text{Encrypt}(p_k, P_{m_1}) &= P_{c_1}(C_{1a}, C_{1b}) \\
\text{Encrypt}(p_k, P_{m_2}) &= P_{c_2}(C_{2a}, C_{2b})
\end{align*}
\]

Where:

\[
\begin{align*}
C_{1a} &= r_1 G \\
C_{1b} &= P_{m_1} + r_1 p_k \\
C_{2a} &= r_2 G \\
C_{2b} &= P_{m_2} + r_2 p_k
\end{align*}
\]

For additive homomorphism:

\[
\text{Decrypt}(P_{c_1} + P_{c_2}) = P_{m_1} + P_{m_2}
\]
Proof of Additive Homomorphism Contd..

Proof:

\[ P_{c_1} + P_{c_2} = (C_{1a}, C_{1b}) + (C_{2a}, C_{2b}) \]
\[ = (r_1 G, P_{m_1} + r_1 P_k) + (r_2 G + P_{m_2} + r_2 P_k) \]

Where, \( r_1 \) and \( r_2 \) are random number generated for corresponding message encryption, in Algorithm 2.

\[ = (r_1 G + r_2 G), (P_{m_1} + r_1 P_k + P_{m_2} + r_2 P_k) \]
\[ = (r_1 + r_2) G, (P_{m_1} + P_{m_2}) + (r_1 + r_2) P_k \]

\[ Decrypt(P_{c_1} + P_{c_2}): \]

Take \((r_1 + r_2) G\), multiply it with Private key \( k \)
\[ = (r_1 + r_2) G_k \]

Add inverse of \((r_1 + r_2) G_k\) with \((P_{m_1} + P_{m_2}) + (r_1 + r_2) P_k\)
\[ = -(r_1 + r_2) G_k + (P_{m_1} + P_{m_2}) + (r_1 + r_2) P_k \]

From KeyGen\((k, P_k)\) we know Public Key \( P_k = k_G \)
\[ = P_{m_1} + P_{m_2} \]

Here Proved:

\[ Decrypt(P_{c_1} + P_{c_2}) = P_{m_1} + P_{m_2} \]
Conclusion and Future Scope

• In this paper, we tried to solve the issues of multi-keyword search over encrypted data.

• We aim to develop a scheme which should provide the correctness of results while maintaining the query and data privacy.

• We attained it by using Elliptic curve based ECEG algorithm. This scheme offers benefits of additive homomorphism together with an ECC level of security.

• Because of ECC and binary query vector used in the scheme, this fits suitable for the devices with limited computational capacity.

• This scheme opens up scope for lots of future work, as ECEG is used the very first time for this purpose.
Thank You