A Note on One Privacy-Preserving Multi-Keyword Ranked Search Scheme over Encrypted Cloud Data

Zhengjun Cao and Lihua Liu

Abstract. We show that the scheme [IEEE TPDS, 25(1), 2014, 222-233] fails, because the introduced similarity scores do not represent the true similarities between the indexing vectors and the querying vector. The returned documents by the cloud server are not indeed related to the queried keywords.

Keywords. Cloud computing, privacy-preserving search, multi-keyword ranked search, indexing vector, querying vector.

1 Introduction

Recently, Cao et al. [1] have proposed a scheme for privacy-preserving multi-keyword ranked search over encrypted cloud data. In the scheme the client has the plaintext document set $\mathcal{F} = \{F_1, F_2, \cdots, F_m\}$. He first encrypts $\mathcal{F}$ as $\mathcal{C}$ using a symmetric key encryption system. Given the keyword set $\mathcal{W} = \{W_1, W_2, \cdots, W_n\}$, he generates a binary indexing vector D_i for F_i where each bit $D_i[j]$ represents whether the corresponding keyword W_j appears in F_i. He then masks (F_i, D_i) as $(C_i, I_i), i = 1, \cdots, m$. Finally, he uploads all (C_i, I_i) to the cloud. For the binary querying vector Q corresponding to the keyword set $\widetilde{\mathcal{W}}$ which is of interest, the client masks Q as $T_{\widetilde{\mathcal{W}}}$. Upon receiving $T_{\widetilde{\mathcal{W}}}$, the cloud server computes the similarity scores

$$s_i = I_i \cdot T_{\widetilde{\mathcal{W}}} = r(D_i \cdot Q + \varepsilon_i) + t, \quad i = 1, \cdots, m$$

where t is the number of keywords in the query, and r, ε_i are random numbers unknown to the cloud server.

In this note we would like to remark that in the Cao et al.'s scheme the cloud server cannot decided which mediate indexing vector I_i is more similar to the mediate querying vector $T_{\widetilde{\mathcal{W}}}$. Actually, the defined similarity score s_i does not represent the true similarity between the indexing vector D_i and the querying vector Q.

• Z. Cao is with the Department of Mathematics, Shanghai University, Shanghai, China.
• L. liu is with the Department of Mathematics, Shanghai Maritime University, China. liulh@shmtu.edu.cn

2 Review of the Cao et al.'s scheme

Table 1: Cao et al.'s scheme for privacy-preserving multi-keyword ranked search over encrypted cloud data

Client		Server
–*Setup*. Pick a $(n+2)$-bit vector S and two $(n+2)\times(n+2)$ invertible matrices M_1, M_2. Set (S, M_1, M_2) as the secret key. Select a symmetric key encryption system $(\mathcal{E}, \mathcal{D})$. –*BuildIndex*. For documents $\{F_1, F_2, \cdots, F_m\}$ and keywords $\{W_1, W_2, \cdots, W_n\}$, set a binary indexing vector D_i for F_i, where each bit $D_i[j]$ represents whether W_j appears in F_i. Pick a random number ε_i, set $\overrightarrow{D}_i = (D_i, \varepsilon_i, 1)$. Split $\overrightarrow{D}_i$ into $(\overrightarrow{D}'_i, \overrightarrow{D}''_i)$ according to S: if $S[j]=0$, $\overrightarrow{D}'_i[j] = \overrightarrow{D}''_i[j] = \overrightarrow{D}_i[j]$; if $S[j]=1$, $\overrightarrow{D}'_i[j] + \overrightarrow{D}''_i[j] = \overrightarrow{D}_i[j]$. Set $(C_i, I_i) = (\mathcal{E}(F_i), \{M_1^T\overrightarrow{D}'_i, M_2^T\overrightarrow{D}''_i\})$. $i = 1, \cdots, m$. Send all (C_i, I_i) to the cloud.	$\xrightarrow[i=1,\cdots,m]{(C_i,I_i)}$	Store all (C_i, I_i).
–*Input*. $\widetilde{\mathcal{W}} \subset \mathcal{W}$, which is of t keywords. –*Trapdoor*. Set the binary querying vector Q where each bit $Q[j]$ represents whether the keyword W_j appears in $\widetilde{\mathcal{W}}$. Set $\overrightarrow{Q} = (rQ, r, t)$, where r is a random number. Split $\overrightarrow{Q}$ into $(\overrightarrow{Q}', \overrightarrow{Q}'')$ according to S: if $S[j]=1$, $\overrightarrow{Q}'[j] = \overrightarrow{Q}''[j] = \overrightarrow{Q}[j]$; if $S[j]=0$, $\overrightarrow{Q}'[j] + \overrightarrow{Q}''[j] = \overrightarrow{Q}[j]$. Set $T_{\widetilde{\mathcal{W}}} = \{M_1^{-1}\overrightarrow{Q}', M_2^{-1}\overrightarrow{Q}''\}$.		
–*Query*. Send $T_{\widetilde{\mathcal{W}}}$ to the server.	$\xrightarrow{T_{\widetilde{\mathcal{W}}}}$	Compute all $s_i = I_i \cdot T_{\widetilde{\mathcal{W}}}$ and sort them.
–*Output*. Decrypt all documents in $\mathcal{C}_{\widetilde{\mathcal{W}}}$.	$\xleftarrow{\mathcal{C}_{\widetilde{\mathcal{W}}}}$	Return the top-k ranked id list $\mathcal{C}_{\widetilde{\mathcal{W}}}$.

3 Analysis of Cao et al.'s scheme

In the Cao et al.'s scheme, the cloud server has to compute the similarity scores

$$\begin{aligned} s_i &= I_i \cdot T_{\widetilde{W}} = \{M_1^T\overrightarrow{D}'_i, M_2^T\overrightarrow{D}''_i\} \cdot \{M_1^{-1}\overrightarrow{Q}', M_2^{-1}\overrightarrow{Q}''\} \\ &= \overrightarrow{D}'_i \cdot \overrightarrow{Q}' + \overrightarrow{D}''_i \cdot \overrightarrow{Q}'' = \overrightarrow{D}_i \cdot \overrightarrow{Q} = (D_i, \varepsilon_i, 1)\cdot(rQ, r, t) \\ &= r(D_i \cdot Q + \varepsilon_i) + t, \qquad i = 1, \cdots, m. \end{aligned}$$

The server then sorts them and returns the top-k ranked id list $\mathcal{C}_{\widetilde{W}}$.

We would like to point out that the mechanism fails because the score s_i cannot indicate the true similarity between the indexing vector D_i and the querying vector Q. In fact, given two scores $s_i, s_j, i \neq j$, we have

$$|s_i - s_j| = |r(D_i \cdot Q + \varepsilon_i) - r(D_j \cdot Q + \varepsilon_j)| = |r| \times |(D_i - D_j) \cdot Q + (\varepsilon_i - \varepsilon_j)|.$$

Whether or not $s_i > s_j$, one cannot claim that D_i is more similar to Q than D_j, because $\varepsilon_i, \varepsilon_j$ are randomly selected by the data owner during the phase of BulidIndex. The random term $(\varepsilon_i - \varepsilon_j)$ violates the scalar-product-preserving property of k-nearest neighbor (kNN) computation on an encrypted database [2]. Conventionally, given two n-dimension vectors X_1, X_2 and another n-dimension vector Y, to determine which $X_i, i = 1, 2$, is more similar to Y, it is usual to compute the distances

$$d(X_1, Y) = \|X_1 - Y\| = \sqrt{\|X_1\|^2 - 2X_1 \cdot Y + \|Y\|^2},$$
$$d(X_2, Y) = \|X_2 - Y\| = \sqrt{\|X_2\|^2 - 2X_2 \cdot Y + \|Y\|^2},$$

where $\|X\|$ represents the Euclidean norm of X, and compare the distances. If $d(X_1, Y) < d(X_2, Y)$, then we assert X_1 is more similar to Y. The routine of Distance-Comparison is broadly adopted whether we call the result as "X_1 is more similar to Y", "X_1 is nearer to Y", "X_1 is closer to Y", etc.

We here want to point out that the technique of so-called Secure Inner Product Computation (SIPC) introduced in the Cao et al.'s scheme [1] is derived from the Scalar-Product-Preserving Encryption (SPPE) developed in Wong et al.'s work [2]. But Cao et al. have not observed that the technique of SPPE should be integrated with the subsequent routine of Distance-Comparison. Otherwise, the isolated SPPE does not represent the true similarity between an indexing vector and a querying vector.

4 Conclusion

We show that the Cao et al.'s scheme fails. We would like to stress that the technique of Scalar-Product-Preserving Encryption should be integrated with the conventional routine of Distance-Comparison when it is used to compare the similarities between some vectors and a given vector.

References

[1] N. Cao, C. Wang, M. Li, K. Ren, and W.J. Lou, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data", IEEE Transactions on Parallel and Distributed Systems, 25(1), 222-233, 2014.

[2] W.K. Wong, D.W. Cheung, B. Kao, and N. Mamoulis, "Secure kNN Computation on Encrypted Databases", Proc. 35th ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), pp. 139-152, 2009.