A Note on One Secure Anti-Collusion Data Sharing Scheme for Dynamic Groups in the Cloud

Zhengjun Cao and Lihua Liu

Abstract. We remark that the scheme [IEEE TPDS, 27(1), 2016, 40-50] is flawed because the group manager cannot complete his computational task in the registration phase. Actually, they have misunderstood the concept of public key which is usually associated with an asymmetric encryption algorithm. Besides, the mechanism that the group manager has to re-encrypt all data stored in the cloud after a member is revoked, is somewhat infeasible because of its inefficiency.

Keywords. Cloud computing, key distribution, data sharing.

1 Introduction

Recently, Zhu and Jiang [1] have proposed a data sharing scheme for dynamic groups in the cloud. It claims that the proposed registration method is a secure way for key distribution without any secure communication channels, which enables the users can securely obtain their private keys from group manager. In this note, we remark that the group manager cannot complete his computational task in the registration phase. We stress that, from the practical point of view, the mechanism that the group manager has to re-encrypt all data after a member is revoked, is not generally acceptable because it is very inefficient.

2 The registration method in the Zhu-Jiang scheme

In the Zhu-Jiang scheme there are three entities, the cloud, a group manager and a number of users. The manager generates a bilinear mapping system $S = (q, G_1, G_2, e(\cdot,\cdot))$. He then picks $P, G \in G_1$, $\gamma \in Z_q^*$ and computes $W = \gamma \cdot P, Y = \gamma \cdot G$, $Z = e(G, P)$. He publishes $(S, P, W, Y, Z, f, f_1, Enc())$, where f is a hash function: $\{0,1\}^* \rightarrow Z_q^*$, f_1 is a hash function: $\{0,1\}^* \rightarrow G_1$, and $Enc()$ is a symmetric encryption system. The manager keeps (γ, G) as the secret master key.

• 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

Let ID_i be the identity of a user, pk be the public key of the user that needs to **be negotiated with** the manager, ac be the user's account for paying. The registration phase can be described as follows.

— The user picks $v_1 \in Z_q^*$ and sends (ID_i, pk, ac, v_1) to the manager.

— The manager picks $r \in Z_q^*$, computes $R = e(P,P)^r, U = (r + \gamma\, v_1\, f(pk\|ac\|ID_i)) \cdot P$ and sends U, R to the user.

— The user checks $R \cdot e(v_1\, f(pk\|ac\|ID_i) \cdot P, W) \overset{?}{=} e(U, P)$. If it holds, he picks $v_2 \in Z_q^*$ and sends $ID_i, v_2, AENC_{sk}(ID_i, v_1, ac)$ to the manager, where $AENC()$ is an asymmetric encryption system and sk is the private key corresponding to the public key pk.

— The manager compares the received ID_i with the identity ID_i computed by **decrypting** $AENC_{sk}(ID_i, v_1, ac)$. He also verifies if the decrypted number v_1 is equal to the random number v_1 in the first step. The other description in this step is omitted. We refer to the original for full details.

— The user decrypts $AENC_{pk}(KEY, v_2)$ to obtain his private key $KEY = (x_i, A_i, B_i)$.

In the phase the interactions between the user and the manager can be depicted as follows (see Fig. 3 in [1]).

$$\begin{array}{lcr}
\text{User} & & \text{Manager} \\
 & \xrightarrow{\quad ID_i, pk, ac, v_1 \quad} & \\
R \cdot e(v_1\, f(pk\|ac\|ID_i) \cdot P,\, W) \overset{?}{=} e(U, P) & \xleftarrow{\quad U, R \quad} & \\
 & \xrightarrow{ID_i, v_2, AENC_{sk}(ID_i, v_1, ac)} & \\
 & \xleftarrow{\quad AENC_{pk}(KEY, v_2) \quad} &
\end{array}$$

3 The scheme is flawed

1) *The manager cannot complete his computational task* in the registration phase because the manager cannot **decrypt** $AENC_{sk}(ID_i, v_1, ac)$. In order to ensure that the manager can decrypt a ciphertext, the user must use the manager's public key pk_M to encrypt data. That is to say, the user has to compute $AENC_{pk_M}(ID_i, v_1, ac)$ and send it to the manager. But we find the scheme does not assign the manager's public key pk_M at all.

2) The scheme stresses that the user's public key pk needs to be negotiated with the manager, and the user can securely obtain their private key from the group manager *without any Certificate Authorities*. The authors [1] have misunderstood the concept of public key which is associated with an asymmetric encryption algorithm. We here want to point out whether the negotiation is by online or offline interactions, the manager can certainly

assign the user's private key (x_i, A_i, B_i) as well as pk simultaneously. In such case, *the registration phase is totally unnecessary.*

Notice that in order to bind the identity of an entity to its public key, it is usual to introduce a trusted third party (TTP). The TTP is generally assumed to be honest and fair but it does not have access to the secret or private keys of users [2]. Before creating a public-key certificate for Alice, the TTP must take appropriate measures to verify the identity of Alice and that the public key to be certificated actually belongs to Alice. To this end, it is conventional that *Alice has to appear before the TTP with a passport as proof of identity*, and submit her public key along with evidence that she knows the corresponding private key. Explicitly, a user's public key has to satisfy: *creditability*–it should be authenticated by a certification authority; *accessibility*–it should be easily accessible to any user; *durability*–it should be repeatedly usable in the life duration because the cost to generate and distribute a user's public key is expensive.

3) The scheme adopts the mechanism that *the group manager has to re-encrypt all data stored in the cloud after a member is revoked* (see page 44 in [1]). The mechanism, from the practical point of view, is really infeasible because of its inefficiency.

4 Conclusion

We show that the Zhu-Jiang scheme is flawed. We would like to stress that the generation and authentication of a user's public key takes a lot of work. We specify that when Bob wants to encrypt a message and send it to Alice, Bob needs to invoke her public key, instead of his public key or secret key.

References

[1] Z.M. Zhu and R. Jiang, "A Secure Anti-Collusion Data Sharing Scheme for Dynamic Groups in the Cloud", IEEE Transactions on Parallel and Distributed Systems, 27(1), 40-50, 2016.

[2] A. Menezes, P. Oorschot and S. Vanstone, "Handbook of applied cryptography", CRC Press, 1996.