A Note on Secure Key Issuing in ID-based Cryptography

XU Chunxiang ZHOU Junhui QIN Zhiguang
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, 610054, China
chxxu@uestc.edu.cn

Abstract: Most recently, Lee B. et al proposed a key issuing protocol for ID-based cryptography to solve the key escrow problem. However in this letter, we show that a malicious key generation center (KGC) can successfully attack the protocol to obtain users' private keys. This means that in the protocol, the key escrow problem isn't really removed.

Keywords: ID-based cryptography, Key escrow problem, Secure key issuing, Key generation center, Key privacy authority.

1. Introduction

ID-based cryptography allows for a user's identity information such as his telephone number, email address, and ID card number to serve as his public key. Such a public key is clearly bound to the user, and doesn't need a certificate to indicate the legitimate owning relation between the key and the user. Hence compared with the traditional certificate-based cryptography, the main advantage of ID-based cryptography is to reduce largely the amount of computation and memory requirements for certificate management. However, the key escrow problem in ID-based cryptography limits its application scope. This problem results from the fact that the key generation center (KGC) computes private keys for users. The KGC inevitably has users' private keys. This leads ID-based cryptography only to be applicable to small close environments, for example, small companies.

In recent years, researchers have been trying to solve the key escrow problem to allow for ID-based cryptography to be used in open environments. Some solutions are proposed [1-4]. Most recently, Lee B. et al. [5] presented a secure key issuing protocol for ID-based cryptography. This key issuing protocol sets multiple key privacy authorities (KPAs) in addition to the KGC to protect the privacy of users' private keys. The KGC and the KPAs share the original role of the KGC. They cooperatively compute user's private keys. Lee B. et al's protocol is claimed to be another approach to the key escrow problem. However, in this letter, we show that the KGC can obtain users' private keys if he is malicious. In other words, the key escrow problem remains in the protocol. Actually the protocol doesn't provide any mechanism for the KPAs to verify the correspondence between a user's identity and the partial private key issued by the KGC for the user. A malicious KGC can make use of this weakness to cheat the KPAs and successfully attack the protocol. The KPAs don't effectively limit the KGC's power and help to protect the privacy of users' private keys.

In the following section, we review Lee B. et al's key issuing algorithm. In Section 3, we analyze this protocol. And Section 4 concludes the letter.

2. Review of Lee B. et al's key issuing protocol

Lee B. et al's key issuing protocol can be described in two parts: system and public key setup, and key issuing.

Part 1. System and public key setup

The KGC initializes the system as follows:

● Generate two groups of prime order q: $(G_1,+)$, $(G_2,\bullet)$, and a bilinear map $e: G_1 \times G_1 \to G_2$;

● Choose a random number s_0 as its master key $s_0 \in Z_q^*$, and compute its public key $P_0 = s_0 P$, where $P \in G_1$ is a generator;

● Choose two hash functions H_1, H_2;

● Make s_0 private, and the parameters (G_1, G_2, e, H_1, H_2, P_0) and their description public.

The system public key is generated as follows:

Each KPA_i (i=1, 2, 3, … , n) chooses randomly a number s_i as his master key $s_i \in Z_q^*$ and computer his public key $P_i = s_i P$. The system public key is $Y = s_0 s_1 s_2 \cdots s_n P = s_1 s_2 \cdots s_n P_0$. Let $Y_0^{'} = P_0$ and $Y_i^{'} = s_i Y_{i-1}^{'}$. The KPAs sequentially compute $Y_1^{'}, Y_2^{'}, ..., Y_n^{'}$. Clearly $Y = Y_n^{'}$.

Part 2. Key issuing

Let U denote a user whose identity is ID. U chooses a random number x, computes $X = xP$, and sends X to the KGC. The KGC verifies U's identity. Then the KGC computes U's public key $Q_{ID} = H_1(ID, KGC, KPA_1, ...KPA_n)$, a blinded partial private key $Q_0^{'} = H_2(e(s_0 X, P_0))s_0 Q_{ID}$, and the KGC's signature on the partial private key $Sig_0(Q_0^{'}) = s_0 Q_0^{'}$. The KGC sends $(Q_0^{'}, Sig_0(Q_0^{'}))$ to U.

U sequentially sends $(ID, X, Q_{i-1}^{'}, Sig_{i-1}(Q_{i-1}^{'}))$ to KPA_i (i=1,2,…,n). KPA_i (i=1,2,…,n) checks if the equation $e(Sig_{i-1}(Q_{i-1}^{'}), P) = e(Q_{i-1}^{'}, P_{i-1})$ holds, computes $Q_i^{'} = H_2(e(s_i X, P_i))s_i Q_{i-1}^{'}$ and $Sig_i(Q_i^{'}) = s_i Q_i^{'}$, and send $(Q_i^{'}, Sig_i(Q_i^{'}))$ to U.

Finally, U computes his private key as:

$$D_{ID} = \frac{Q_n^{'}}{H_2(e(P_0, P_0)^x) \cdots H_2(e(P_n, P_n)^x)}.$$

Note $H_2(e(s_iX, P_i)) = H_2(e(s_ixP, P_i)) = H_2(e(P_i, P_i)^x)$. So we have $D_{ID} = \frac{Q_n^{'}}{H_2(e(P_0, P_0)^x) \cdots H_2(e(P_n, P_n)^x)} = s_0 s_1 \cdots s_n Q_{ID}$. *U* can verify his private key by checking if the equation $e(D_{ID}, P) = e(Q_{ID}, Y)$ holds.

3. Analysis of Lee B. et al's key issuing protocol

Seemingly, as Lee B. et al pointed out about their protocol, since KGC and the *n* KPAs cooperatively compute the private key of a user, the private key will keep in privacy if any one of the authorities (the KGC and the KPAs) is honest. This means that only if all the authorities collude they can obtain a user's private key. But in the following we will show that if the KGC is malicious, he can successfully attack the protocol and obtain users' private keys without colluding with any KPA.

In the protocol, the KGC is responsible for verifying a user's identity and issues a blinded partial private key to the user. Then the user sends a 4-tuple $(ID, X, Q_0^{'}, Sig_0(Q_0^{'}))$ to KPA_1. KPA_1 verifies if $Sig_0(Q_0^{'})$ is the KGC's signature on $Q_0^{'}$ and computes $Q_1^{'}$. Note that KPA_1 doesn't verify if $Q_0^{'}$ is corresponding to *ID*. In fact, there is no mechanism for such verification in the protocol. The parameter *ID* in the 4-tuple is not involved in computing $Q_1^{'}$, and KPA_1 doesn't verify the user's identity with *ID*. Actually, even if KPA_1 verifies the user's identity, he cannot verify if $Q_0^{'}$ is computed with this user's identity. This leaves a backdoor for a malicious KGC to attack the protocol.

A malicious KGC and an assistant can make a conspiracy attack to the protocol and illegally obtain the private keys of users. Assume that the KGC wants to obtain the private key of a user *A*. He uses the identity of *A* (denoted by ID_A) and a random number *x* (for computing $X = xP$) to start the private key computation protocol. His assistant pretends to be *A*. The KGC computes $(Q_0^{'}, Sig_0(Q_0^{'}))$ using ID_A. The assistant uses $(ID_A, X, Q_0^{'}, Sig_0(Q_0^{'}))$ to let the KPAs perform the computation required. Since the KPAs don't check the assistant's identity, the assistant will success. The KGC and his assistant can finally compute the user *A*'s private key. As discussed previously, even if the KPAs check the assistant's identity, since they cannot verify the correspondence between $Q_0^{'}$ and *ID*, the assistant replaces ID_A with his own identity and he will be successful as well.

Actually a malicious KGC can independently attack the protocol. This needs the KGC to do what the assistant does in some way.

The above analysis illustrated that the key escrow problem has not been really solved in the protocol. Additionally, the key escrow problem is hidden. Thus it becomes more dangerous.

We should point out that in Lee B. et al's protocol, the relation structure of the authorities is essentially serial with the KGC at the most upstream, and the KPAn at the most downstream, and this enable a malicious KGC to make use of flaws in the protocol to cheat the KPAs and attack the protocol. In other multiple authorities based protocols [3-4], the relation structure of the authorities is parallel, no authority occupies a predominant position. The attack from a malicious authority is difficult to be realized.

4. Conclusion

The key escrow problem in ID-based cryptography motivates researchers to seek various solutions. Lee B. et al's proposal doesn't really solve the key escrow problem, nevertheless the idea of setting multiple key privacy authorities to protect users' private keys is interesting. New secure protocols may be established based on the idea.

References

[1] Sattam S., Al-Riyami S. and Paterson K., Certificateless public key cryptography, Advances in Cryptology-Asiacrypt'2003, Springer-Verlag, pp.452-472.

[2] Gentry C., Certificate-based encryption and the certificate revocation problem, Advances in Cryptology-EUROCRYPT 2003, Springer-Verlag, pp.272-293.

[3] Chen L., Harrison K., Smart N. P. and Soldera D., Application of multiple trust authorities in pairing based cryptosystems, InfraSec 2002, Springer-Verlag, pp.260-275.

[4] Boneh D. and Franklin F., Identity-based encryption from the Wail pairing, Advances in Cryptology-Crypt'2001, Springer-Verlag, pp.213-229.

[5] Lee B., Boyd E., Daeson E., Kim K., Yang J. and Yoo S., Secure key issuing in ID-based cryptography, In proceedings of the Second Australian Information Security Workshop-AISW 2004, pp.69-74.