Cryptanalysis of Key Issuing Protocols in ID-based Cryptosystems

Raju Gangishetti, M. Choudary Gorantla, Manik Lal Das, Ashutosh Saxena

Institute for Development and Research in Banking Technology
Castle Hills, Road #1, Masab Tank, Hyderabad 500057, AP, INDIA.
{graju, gmchoudary}@mtech.idrbt.ac.in, {mldas, asaxena}@idrbt.ac.in

February 1, 2008

Abstract

To remove key escrow problem and avoid the need of secure channel in ID based cryptosystem Lee et al.[1] proposed a secure key issuing protocol. However we show that it suffers from impersonation, insider attacks and incompetency of the key privacy authorities. We also cryptanalyze Sui et al.'s[2] separable and anonymous key issuing protocol.

1 Review of Lee et al.'s Protocol [1]

It includes five stages namely, System Setup, System Public Key Setup, Key Issuing, Key Securing and Key Retrieving.

1.1 System Setup

The KGC specifies two cyclic groups G_1, G_2 of prime order q where G_1 is additive and G_2 is multiplicative groups. It also defines a bilinear mapping as $e : G_1 \times G_1 \rightarrow G_2$ between G_1, G_2 and hash functions $H : \{0,1\}^* \rightarrow G_1$, $h : G_2 \rightarrow Z_q^*$. Let $P \in G_1$ be an arbitrary generator of G_1. The KGC selects a master key $s_0 \in Z_q^*$ at random and computes its pubic key $P_0 = s_0P$.

1.2 System Public Key Setup

The n KPAs establish their key pairs. KPA_i chooses his master key s_i and computes his public key $P_i = s_iP, \forall i = 1, ..., n$. Then all KPAs cooperate sequentially and computes $Y_i'{=}s_iY_{i-1}$ where $Y_0' = P_0 = s_0P$.

Finally, $Y = Y_n' = s_0s_1...s_nP$ is published as system public key. This sequential process can be verified by $e(Y_i', P) = e(Y_{i-1}', P_i)$.

1.3 Key Issuing

A user with ID chooses a random secret x, computes a blinding factor $X = xP$ and requests the KGC to issue a partial private key by sending X, ID. Then the KGC issues a blinded partial private key as follows.

1. Checks the identification and computes the public key of the user as $Q_{ID} = H(ID, KGC, KPA_1, ..., KPA_n)$.

2. Computes a blinded partial private key as $Q_0' = h(e(s_0X, P_0))s_0Q_{ID}$.

3. Computes KGC's signature on Q'_0 as $Sig_0(Q'_0) = s_0Q'_0$.
4. Sends Q'_0 and $Sig_0(Q'_0)$ to the user.

The user can unblind Q'_0 using his knowledge of x, since $h(e(s_0X, P_0))$=$h(e(s_0xP, P_0)) = h(e(P_0, P_0)^x)$.

1.4 Key Securing

The user requests $KPA_i(i = 1, ..., n)$ sequentially to provide key privacy service by sending ID, X, Q'_{i-1} and $Sig_{i-1}(Q'_{i-1})$. Then KPA_i performs following steps

1. Checks $e(Sig_{i-1}(Q'_{i-1}), P) = e(Q'_{i-1}, P_{i-1})$.
2. Computes $Q'_i = h(e(s_iX, P_i))s_iQ'_{i-1}$ and $Sig_i(Q'_i = s_iQ'_i)$.
3. Sends Q'_i and $Sig_i(Q'_i)$ to the user.

This process is carried out up to KPA_n. Finally user receives Q'_n.

1.5 Key Retrieving

The user retrieves his private key S_{ID} by unblinding Q'_n as follows.

$$S_{ID} \quad = \quad \frac{Q'_n}{h(e(P_0, P_0)^x)h(e(P_1, P_1)^x)...h(e(P_n, P_n)^x)} = s_0s_1...s_nQ_{ID}$$

The user can verify the correctness of his private key by $e(S_{ID}, P) = e(Q_{ID}, Y)$.

2 Cryptanalysis of Lee et al.'s Protocol

2.1 Impersonation Attack

In Key Issuing phase, user sends $X = xP$ and ID to the KGC. Any active adversary can modify the X as $X^* = x^*P$ and still it cannot be detected by KGC. Because there is no binding between the ID and X. Then KGC computes partial private key $Q^*_0 = h(e(s_0X^*, P_0))s_0Q_{ID}$, and sends to the user through public channel. Adversary can eavesdrop Q^*_0 and request the KPAs for key privacy service. At the end Adversary can extract the private key by unblinding Q^*_n.

2.2 Insider Attack

In Key Securing phase, user requests KPA_i to provide key privacy service by sending ID, X, Q'_{i-1}, $Sig_{i-1}(Q'_{i-1})$, where fourth parameter is a signature of KPA_{i-1} on third parameter.

If KPA_{i-1} wants a signature of KPA_i on m, he sends ID^*, $X^* = x^*P$, $Q^*_{i-1} = rH(m)$ and $Sig_{i-1}(Q^*_{i-1}) = rs_{i-1}H(m)$ to KPA_i where $r \in_R Z^*_q$. Then KPA_i performs the following steps

1. Checks $e(Sig_{i-1}(Q^*_{i-1}), P) = e(Q^*_{i-1}, P_{i-1})$.
2. Computes $Q^*_i = h(e(s_iX^*, P_i))s_iQ^*_{i-1}$ and $Sig_i(Q^*_i) = s_iQ^*_i$.
3. Sends Q^*_i and $Sig_i(Q^*_i)$ to the user(i.e. KPA_{i-1}).

Now, KPA_{i-1} has $Q^*_i = h(e(s_iX^*, P_i))s_irH(m)$ and he can extract the signature of KPA_i on m as $h(e(P_i, P_i)^{x^*})^{-1}r^{-1}Q^*_i = s_iH(m)$. At the same time KPA_i cannot get signature of the KPA_{i-1} (i.e. $s_{i-1}H(m)$), because KPA_{i-1} sends his signature in blinded manner. Thus, KPA_{i-1} can obtain KPA_i's signature on any message of his choice.

2.3 Incompetency of KPAs

In Key Securing Phase, the user requests $KPA_i (i = 1, 2, ..., n)$ sequentially to provide key privacy service by sending ID, X, Q'_{i-1}, and $Sig_{i-1}(Q'_{i-1})$. Then KPA_i validates the received parameters by checking the equality
$e(Sig_{i-1}(Q'_{i-1}), P)$=$e(Q'_{i-1}, P_{i-1})$.

Any active adversary can alter Q'_{i-1}, $Sig_{i-1}(Q'_{i-1})$ and replaces with the following $Q^*_{i-1} = r^* Q'_{i-1}$, $Sig_{i-1}(Q^*_{i-1}) = r^* Sig_{i-1}(Q'_{i-1})$. Then KPA_i performs

1. Checks $e(Sig_{i-1}(Q^*_{i-1}), P) = e(Q^*_{i-1}, P_{i-1})$
2. Computes $Q^*_i = h(e(s_i X, P_i)) s_i Q'_{i-1}$ and $Sig_i(Q^*_i) = s_i Q^*_i$
3. Sends Q^*_i, and $Sig_i(Q^*_i)$ to the user.

It may be noted that the user is not checking the correctness of the received parameters in intermediate stages. Therefore any modification by an Adversary during the communication between user and KPA_i will be undetected till the end of Key Securing Phase. This requires the user to execute this phase again from the beginning. Further, as the KGC and KPAs are not capable of checking the validity of the received parameters, they are signing them blindly.

The attack given in Section 2.1 can also be applied to [3].

3 Review of Sui et al. [2]

A one time password pwd can be established between the Local Registration Authority(LRA) and the user after the off-line authentication.
Setup(run by KGC): It takes the security parameter k and returns $params$ (System Parameters) and the master-key. Let G be a GDH group of prime order p. Public information is $I_{SAKI} = (G, p, H, P_{PKG})$. P is a generator of G and $H : 0, 1^* \rightarrow G$ is a oneway hash function and $Q_A = H(id_A)$. $P_{PKG} = sP$ is the system public key.
Key Generation: It takes inputs as $params$, master-key, and an arbitrary $ID \in \{0, 1\}^*$; and returns a private key S_{ID}. The password pwd is user's chosen password during off-line authentication and the tuple (ID, pwd) is stored in KGC's database of "pending private key".

1. A:selects a random number r, $A \rightarrow KGC : Q = rH(ID), T = r^{-1}H(pwd)$.
2. KGC: checks the validity of the request by checking whether $e(Q, T) = e(H(ID), H(pwd))$ holds for a certain tuple in KGC's database.
3. KGC: computes sQ, $KGC \rightarrow A : S = sQ$
4. A: verifies the blinded private key by checking $e(S, P) = e(Q, P_{PKG})$. If it holds, A unblinds the encrypted private key and obtains $sH(ID)$.

The user can delete pwd after obtaining the private key. The KGC can also remove the tuple (ID, pwd) from the database after the protocol.

4 Cryptanalysis of Sui et al. Protocol

4.1 Stolen Verifier Attack

In Sui et al. protocol, (ID, password) is stored in KGC's database. If an Adversary steals the database he can have genuine users' secrets on requesting the KGC on behalf of any registered user available in database. Though the KGC stores (ID, password) for a short-time till the corresponding secret key is issued, it affects the protocol entirely.

4.2 Insider Attack

In practice, it is likely that a user uses same password to access several systems and other purposes for his convenience. In the registration phase, the user gives his password *pwd* to LRA and the LRA stores the ID and corresponding password in the database. In the extended scheme given to remove the key escrow by single KGC, the database is accessible by multiple KGC's and LRA. Any one of the insider of the system could impersonate user's login on stealing password and can get access of the other systems.

4.3 Incompetency of KGCs

A user requests for private key as follows:

- Selects a random number r, and computes $Q = rH(ID), T = r^{-1}H(password)$ and sends to the KGC.
- KGC checks the validity of the request by checking the equality $e(Q, T) = e(H(ID), H(password))$.
- Computes blinded private key $S = sQ$ and sends to the user where s is the KGC's private key.
- Then user verifies S by checking the equality $e(S, P) = e(Q, P_{pub})$ where $P_{pub} = sP$ is KGC's public key.

Any Adversary can alter the parameters Q, T and replace with $Q^* = r^*Q$, $T^* = {r^*}^{-1}T$ and KGC verifies the equality $e(Q^*, T^*) = e(H(ID), H(password))$. Then the KGC computes $S^* = sQ^*$ and sends to the user. In this protocol the KGC cannot check the validity of the parameters received and thus blindly signs on it.

5 Conclusion

In this work we have cryptanalyzed two ID based key issuing protocols of [1, 2]. We showed that the Lee et al. [1] protocol suffers from impersonation, insider attacks and incompetency of the key privacy authorities. We also showed that the Sui et al.'s[2] separable and anonymous key issuing protocol suffers from stolen verifier, insider attacks and incompetency of key generation centers.

References

[1] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, S. Yoo, "Secure Key Issuing in ID-based Cryptography", *In proceedings of the Second Australian Information Security Workshop-AISW 2004*, ACSW Frontiers 2004, ACS Conferences in Research and Practice in Information Technology 32, 2004, pp. 69-74.

[2] A. Sui, S. S. M. Chow, L. C. K. Hui, S. M. Yiu, K. P. Chow, W. W. Tsang, C. F. Chong, K. H. Pun, H. W. Chan, " Seperable and Anonymous Identity-Based Key Issuing without Secure Channel", IACR eprint Archive, Available from http://eprint.iacr.org/2004/322,(refered on 18-May-2005).

[3] R. Gangishetti, M. C. Gorantla, M. L. Das, A. Saxena and V. P. Gulati, " An Efficient Secure Key Issuing Protocol in ID-Based Cryptosystems", *In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2005)*, Volume-1, IEEE Computer Society, 4-6 April, 2005, Las Vegas, USA, pp. 674-678.