An attack on the certificateless signature scheme from EUC Workshops 2006

Je Hong Park

National Security Research Institute
161 Gajeong-dong, Yuseong-Gu, Daejeon, South Korea

Abstract. In this letter, we will show that the certificateless signature scheme recently proposed by Yap, Heng and Goi at EUC Workshops 2006 is insecure against a key replacement attack. Our attack shows that anyone who replaces a signer's public key can forge valid signatures for that signer without knowledge of the signer's private key.
Keywords: Signature scheme, Certificateless PKC, Public Key Cryptography, Cryptography

1 Introduction

The certificateless cryptosystem introduced by Al-Riyami and Paterson [1] is designed to overcome the key escrow limitation which is inherent in identity-based cryptosystems. Each user has a unique identifier and the partial private key associated with that identifier is computed by a Key Generation Center (KGC), who knows some special master secret, and distributed to the user with that identifier. But the user's private key also contains a unique secret value generated by the user. That is, the user's private key is not generated by the KGC alone and so the KGC does not know the user's private key that implies the escrow freeness. Independent to the identifier, the user also publishes the public key, based on the secret value. Note that the user's public key does not need to be certified by any trusted authority as in conventional PKIs. But the structure of the certificateless scheme ensures that the key can be verified without a certificate. So some adversaries to attack a particular certificateless scheme may attempt to replace a user's public key with a value of their choice, and want to gain an advantage in breaking the scheme. It is called in general a *key replacement attack* [1, 5, 6] and successfully applied to some certificateless schemes [3, 4].

Recently, Yap, Heng and Goi proposed a certificateless signature scheme (called the YHG scheme here) and claimed that their scheme is efficient, comparison to previous schemes [5]. Their efficiency is caused by the elimination of a public key validation checking which requires pairing computations in the signature verification phase. But we will show that the YHG scheme is insecure against a key replacement attack as the result of [3]. Our attack is based on the fact that the user private key of the YHG scheme has the form of a BLS multisignature [2] generated by the KGC and the user. Due to the lack of a public key validity checking in the signature verification phase, a verifier cannot ensure that the signer knows the secret value. It implies that an adversary who replaces a signer's public key can forge signatures of that signer, without knowledge of the signer's private key.

2 Review of YHG certificateless signature scheme

Throughout this paper, $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, \cdot)$ denote two cyclic groups of prime order q. A *pairing* is an efficiently computable, non-degenerate function $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ with the bilinearity property that $e(P + Q, R) = e(P, R)e(Q, R)$ and $e(P, Q + R) = e(P, Q)e(P, R)$ for $P, Q, R \in \mathbb{G}_1$. Let $H_1 : \{0,1\}^* \to \mathbb{G}_1$ and $H_2 : \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{Z}_q^*$ be hash functions. These are used as a part of the system parameter generated by the KGC. The YHG certificateless signature scheme can be described as follows:

- Setup: Given a security parameter k, the KGC chooses an arbitrary generator $P \in \mathbb{G}_1$ and selects a random $s \in \mathbb{Z}_q^*$ and sets $P_0 = sP$. Then the system parameters are $\mathsf{params} = \langle \mathbb{G}_1, \mathbb{G}_2, e, q, P, P_0, H_1, H_2 \rangle$. The message space is $M = \{0,1\}^*$. The master secret key is $\mathsf{mk} = s$.
- Set Partial Private Key: Given params, mk and a user A's identifier ID_A, the KGC computes $Q_A = H_1(\mathsf{ID}_A) \in \mathbb{G}_1$ and outputs a partial private key $D_A = sQ_A$.
- Set Secret Value: Given params, the user A selects a random value $x_A \in \mathbb{Z}_q^*$ as a user secret value.
- Set Private Key: Given params and the partial private key D_A, the user A computes a user private key $S_A = x_A Q_A + D_A$.
- Set Public Key: Given params and the secret value x_A, the user A computes a user public key $P_A = x_A P \in \mathbb{G}_1$.
- Signature Generation: Given params, ID_A, a message m and the private key S_A, the user A randomly chooses $r \in \mathbb{Z}_q^*$ and sets $U = rQ_A \in \mathbb{G}_1$. Then computes a signature $\sigma = (U, V)$ for the message m where $V = (r + h)S_A$ and $h = H_2(m, U) \in \mathbb{Z}_q^*$.
- Signature Verification: Given a signature/message pair (σ, m), the signer's identifier ID_A and the signer's public key P_A, the verifier computes $h = H_2(m, U) \in \mathbb{Z}_q^*$ and checks whether $e(P, V) = e(P_0 + P_A, U + hQ_A)$. If not, then rejects the signature else accepts it.

The authors claim that this scheme is more efficient then previously proposed schemes because fewer bilinear pairing computations are required. As described above, this scheme requires only two pairing computations in the signature verification phase. This efficiency is induced from the elimination of a public key validation check. We will show that this elimination makes the YHG scheme vulnerable against a key replacement attack.

3 Security Analysis

Without loss of generality, the signer forwards his/her public key to the intended verifier(s) and announces his/her identifier. So an adversary who wants to forge a signature of the user A with the identifier ID_A runs as follows:

1. Randomly chooses $x \in \mathbb{Z}_q^*$ and computes a signature $\sigma = (U, V)$ for a message m as follows:

$$U = rQ_A, h = H_2(m, U) \text{ and } V = (r + h)xQ_A.$$

2. Sets $P'_A = xP - P_0$ as a public key of the user A.
3. Then sends the signature σ, the message m, the identifier ID_A and the public key P'_A to the verifier(s).

Then the verifier computes $h = H_2(m, U)$ and checks whether $\langle P, P_0 + P'_A, U + hQ_A, V\rangle$ is a valid Diffie-Hellman tuple. Since $e(P, V) = e(P, (r+h)xQ_A)$ and $e(P_0+P'_A, U+hQ_A) = e(xP, (r+h)Q_A)$, $e(P, V) = e(P_0 + P'_A, U + hQ_A)$ and so $\langle P, P_0 + P'_A, U + hQ_A, V\rangle$ is valid. Hence σ is verified as a valid signature for the message m generated by the user A.

Our attack is based on the algebraic structure of a user's private key in the YHG scheme. Since the signer A's private key S_A has the form of a BLS multisignature [2] generated by the KGC and A, we can apply a rogue attack using the key substitution trick. In general, a rogue attack for BLS multisignatures can be described as follows: Let $pk_A = x_A P$ and $pk_B = x_B P$ be two public keys of the user Alice and Bob, respectively. But Bob replaces pk_B by $pk_B - pk_A$. Then for a message m, $x_B H_1(m) = x_A H_1(m) + (x_B - x_A)H_1(m)$ can be regarded as a valid multisignature on m by both Alice and Bob. In the YHG scheme, the KGC plays the role of a honest user to generate a multisignature of ID_A for a user A and is prohibited to replace the user A's public key. But a third party can use this key substitution trick for BLS multisignatures to forge a YHG signature of the user A, based on two facts that a user A's public key is not certified and knowledge of the secret value corresponding to the signer's public key is not, even implicitly, checked in the signature verification phase.

To prevent this attack, therefore, the signature verification phase is required to demonstrate that the signer has knowledge of the secret value corresponding to the public key [3]. One instance to provide it is to modify the public key of a user A to include an additional value $x_A P_0$, where x_A is the secret value of A and then to add the public key validity checking equation

$$e(P_0, P_A) = e(P, x_A P_0) \tag{1}$$

to the signature verification phase. This equation basically ensures that the signer A's public key $\langle X, Y\rangle$ holds the relation $Y = sX$ where $Y = x_A P_0$ and $X = P_A$. Furthermore, it makes sure that the secret value x_A, chosen by the signer A, has been used correctly to obtain $S_A = x_A Q_A + D_A$ [4]. Though an adversary is able to replace P_A by P'_A, it is impossible to pass the equation (1) without knowledge of the discrete logarithm of P'_A. Unfortunately, this modification requires 4 pairing computations though only 2 are needed per signature if multiple signatures by the same signer are to be verified.

4 Conclusion

We showed that the YHG certificateless signature scheme is vulnerable to a key replacement attack. To prevent this attack, it is required to add the signer's public key validation checking in the signature verification phase.

References

1. S.S. Al-Riyami and K.G. Paterson, "Certificateless public key cryptography," Proc. of ASIACRYPT 2003, Lecture Notes in Comput. Sci., vol. 2894, pp. 452–473, 2003.
2. A. Boldyreva, "Efficient threshold signature, multisignature and blind signature schemes based on the gap-Diffie-Hellman-group signature scheme," Proc. of PKC 2003, Lecture Notes in Comput. Sci., vol.2567, pp.31–46, 2003.
3. X. Cao, K.G. Paterson and W. Kou, "An attack on a certificateless signature scheme," preprint, 2006.

4. B.C. Hu, D.S. Wong, Z. Zhang and X. Deng, "Key replacement attack against a generic construction of certificateless signature", Proc. of ACISP 2006, Lecture Notes in Comput. Sci., vol. 4058, pp. 235–246, 2006.
5. W.-S. Yap, S.-H. Heng and B.-M. Goi, "An efficient certificateless signature scheme," Proc. of EUC Workshops 2006, Lecture Notes in Comput. Sci., vol. 4097, pp. 322–331, 2006.
6. Z. Zhang, D.S. Wong, J. Xu and D. Feng, "Certificateless public-key signature: Security model and efficient construction", Proc. of ACNS 2006, Lecture Notes in Comput. Sci., vol. 3989, pp. 293–308, 2006.