A Note on Signature Standards

Michael Braun and Anton Kargl

Siemens AG
Corporate Technology
Otto-Hahn-Ring 6
D-81739 Munich
{mic.braun|anton.kargl}@siemens.com

September 10, 2007

Abstract

A major security goal for signature schemes is to prevent an adversary from producing new valid signatures even though he can receive valid signatures of any messages from the legitimate signer. On the one hand the security of elliptic curve signature schemes, as ECDSA, ECGDSA, or ECKCDSA, is based on the elliptic curve discrete logarithm problem, respectively on the security of the used hash function. On the other hand some special cases for ephemeral keys and signature components also have to be excluded to guarantee the security of the signature scheme. In this paper we are going to investigate some exceptional cases, which are not covered by current signature generation algorithms, but leak information on the private signature key.

Keywords: Digital Signatures, ECDSA, ECGDSA, ECKCDSA

1 Introduction

A digital signature is a cryptographic primitive to provide data origin authentication, data integrity and non-repudiation. Goldwasser, Micali and Rivest define the notion of security of a signature scheme (see [2]): A secure signature scheme means that only legitimate signers can generate valid signatures of a message. No adversary should have the ability to produce signatures even if he knows arbitrary pairs of messages and valid signatures.

Elliptic curve signature algorithms are very attractive for security systems based on smart card solutions and embedded systems. The security of these algorithms is based on the complexity of the discrete logarithm problem of the corresponding elliptic curve (ECDLP), respectively the security of the hash function. Even though the underlying elliptic curve and the

used hash function are cryptographically secure, there are still some exceptional cases, concerning the ephemeral keys and the signature components, in which the private signature key can be recovered.

The digital signature generation standards usually comprise the cases when signature components equal zero, but in this paper we are going to investigate further exceptions to be considered for a secure signature generation.

2 Basic Idea

Let E be a cryptographically strong elliptic curve represented in a Weierstraß-equation defined over a finite field $\mathbb{F}$ of characteristic p with order n (see [1]). Let P be the generator of the large subgroup of E with order q. In addition, let H denote a cryptographically secure hash function whose output bitlength is not greater than that of q. According to the latest version of the standard if this condition cannot be satisfied due to the choice of the domain parameters of the elliptic curve, a hash function with larger output length can also be used. Then the output of H has to be truncated.

The Elliptic Curve Digital Signature Algorithm (ECDSA [3]) to perform a signature generation of the message m, works as follows:

ECDSA Signature Generation

Input:	domain parameters (E, P), private key d, message m
Output:	signature (r, s)

1. pick $0 < k < q$ randomly
2. $(x_R, y_R) \leftarrow kP$
3. $r \leftarrow x_R \mod q$
4. if $r = 0$ then goto 1
5. $k \leftarrow k^{-1} \mod q$
6. $e \leftarrow H(m)$
7. $s \leftarrow k(e + rd) \mod q$
8. if $s = 0$ then goto 1
9. return (r, s)

In this algorithm there are two cases which lead to a repeated choice of the ephemeral key k. In the first case, if $r = 0$ the signature component s would not depend on the private key d of the signer. In the second case, if $s = 0$, the inversion in the signature verification algorithm cannot be performed. Both cases appear very unlikely, but they have to be treated in the signature standard.

In order to verify the digital signature (r, s) of the message m the verifier computes the following steps ($Q = dP$ denotes the public key):

ECDSA Signature Verification

Input:	domain parameters (E, P), public key Q, message m, signature (r, s)
Output:	acceptance or rejection of the signature

1. verify that $0 < r, s < q$
2. $s' \leftarrow s^{-1} \mod q$
3. $e \leftarrow H(m)$
4. $h_1 \leftarrow s'e \mod q$
5. $h_2 \leftarrow s'r \mod q$
6. $R = (x_R, y_R) \leftarrow h_1 P + h_2 Q$
7. if $R = 0$ then reject
8. if $x_R \mod q = r$ then accept else reject

During the signature verification the results of the two scalar multiplications $h_1 P$ and $h_2 Q$ must be added. Since the addition formula of two distinct points is different from the formula to double a point, the verifier may detect if $h_1 P = h_2 Q$. This leads to the relation $h_1 = h_2 d$ with known terms h_1 and h_2. Hence, the secret signature key can be recovered with the cost of one multiplication and one inversion in $\mathbb{Z}/q\mathbb{Z}$. Of course, this special case is quite rare, since its probability is equal to the probability of the event $s = 0$ in the signature generation.

In order to avoid these doublings during the signature verification the signature components must not fulfill $h_1 P = h_2 Q$ which is equivalent to

$$h_1 = h_2 d \iff s^{-1} e = r s^{-1} d \iff e = rd$$

Thus, ephemeral keys k producing the condition $e = rd$ have to be discarded in the signature generation primitive:

Modified ECDSA Signature Generation

Input:	domain parameters (E, P), private key d, message m
Output:	signature (r, s)

1. pick $0 < k < q$ randomly
2. $(x_R, y_R) \leftarrow kP$
3. $r \leftarrow x_R \mod q$
4. if $r = 0$ then goto 1
5. $k \leftarrow k^{-1} \mod q$
6. $e \leftarrow H(m)$
7. **if $e = rd$ then goto 1**
8. $s \leftarrow k(e + rd) \mod q$
9. if $s = 0$ then goto 1
10. return (r, s)

3 Application to ECGDSA and ECKCDSA

In the previous section we have shown that occurring doublings in the signature verification can reveal information on the private signature key. In the following we investigate whether this situation can also be applied to similar elliptic curve signature algorithms, as ECGDSA and ECKCDSA.

In the German version of elliptic curve signature standard (ECGDSA [4]) the signer generates the private signature key d and derives the public signature key by $Q = d^{-1}P$. The inversion of the private signature key d results in a simplified signature generation algorithm, in particular, the ephemeral key k needn't be inverted anymore.

ECGDSA Signature Generation

Input:	domain parameters (E, P), private key d, message m
Output:	signature (r, s)

1. pick $0 < k < q$ randomly
2. $(x_R, y_R) \leftarrow kP$
3. $r \leftarrow x_r \mod q$
4. If $r = 0$ then goto 1
5. $e \leftarrow H(m)$
6. $s \leftarrow d(kr - e) \mod q$
7. If $s = 0$ then goto 1
8. return (r, s)

The corresponding signature verification works as follows:

ECGDSA Signature Verification

Input:	domain parameters (E, P), public key Q, message m, signature (r, s)
Output:	acceptance or rejection of the signature

1. verify that $0 < r, s < q$
2. $r' \leftarrow r^{-1} \mod q$
3. $e \leftarrow H(m)$
4. $h_1 \leftarrow r'e \mod q$
5. $h_2 \leftarrow r's \mod q$
6. $(x_R, y_R) \leftarrow h_1P + h_2Q$
7. If $x_R \mod q = r$ then accept else reject

If the verifier identifies the case $h_1P = h_2Q$, he is also able to recover the ephemeral key k immediately from $e = H(m)$ and r:

$$h_1 = h_2 d \iff e = d(kr - e)d^{-1} \iff 2e = kr \iff k = 2er^{-1}.$$

By obtaing the ephemeral key k the secret key d can be deduced from the second signature component s. To avoid the doubling in the ECGDSA signature verification, the ephemeral key k must not solve the equation $2e = kr$. Hence, we obtain the following extended version of the signature generation:

Modified ECGDSA Signature Generation

Input:	domain parameters (E, P), private key d, message m
Output:	signature (r, s)

1. pick $0 < k < q$ randomly
2. $(x_R, y_R) \leftarrow kP$
3. $r \leftarrow x_r \mod q$
4. If $r = 0$ goto 1
5. $e \leftarrow H(m)$
6. **If $2e = kr$ goto 1**
7. $s \leftarrow d(kr - e) \mod q$
8. If $s = 0$ goto 1
9. return (r, s)

Finally, we also study the ECKCDSA which is the elliptic curve analog of the Korean Certificate-Based Digital Signature Standard (KCDSA [5]). In this algorithm a cryptographically secure hash function H is used with output length of ℓ bit. It is recommended that the order q of the large subgroup is greater than 2^ℓ. Furthermore a hash value *hcert* of the signer's certificate is inlcuded.

ECKCDSA Signature Generation

Input:	domain parameters (E, P), private key d, hashed certificate *hcert*, message m
Output:	signature (r, s)

1. pick $0 < k < q$ randomly
2. $(x_R, y_R) \leftarrow kP$
3. $r \leftarrow H(x_R)$
4. $e \leftarrow H(m, hcert)$
5. $w \leftarrow r \oplus e \mod q$
6. $s \leftarrow d(k - w) \mod q$
7. if $s = 0$ then goto 1
8. return (r, s)

Similiar to the ECGDSA the public key Q is generated by $Q = d^{-1}P$, such that no modular inversion is neccessary neither in the signature generation nor in the verification primitive.

The corresponding signature verification works as follows:

ECKCDSA Signature Verification

Input:	domain parameters (E, P), public key Q, hashed certificate $hcert$, message m, signature (r, s)
Output:	acceptance or rejection of the signature

1. verify that $0 < s < q$
2. verify that $r < 2^{\ell}$
2. $e \leftarrow H(m, hcert)$
3. $w \leftarrow r \oplus e \mod q$
4. $(x_R, y_R) \leftarrow sQ + wP$
5. $v \leftarrow H(x_R)$
7. If $v = r$ then accept else reject

Detecting a doubling in the verfication scheme, the equation $sQ = wP$ leads to the ephemeral key $k = 2w$ and thus the private key d can be reconstructed. Finally, testing the condition $k = 2w$ prevents this doubling.

4 Conclusion

In this paper we have investigated some particular values of the ephemeral key, satisfying a certain condition, where the private signature key can be deduced only from the signature components. The occurrence of such an exceptional case can be identified during the signature verification by detecting a doubling of points. By extenting the signature generation primitive by one further equation which requires a modular multiplication these exceptions can be prevented.

This situation can also be applied to DSA signatures. In this case if a squaring can be detected during the signature verification the private signature key can be recovered.

References

[1] S. Vanstone D. Hankerson, A. Menezes. *Guide to Elliptic Curve Cryptography.* Springer-Verlag, 2004.

[2] S. Goldwasser, S. Micali, and R. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen Message Attacks. In *SIAM Journal of Computing*, volume 17, April 1988.

[3] American National Standards Institute. Public Key Cryptography for the Financial Services Industry — The Elliptic Curve Digital Signature Algorithm (ECDSA). ANSI X9.62 — 2005, 2005.

[4] ISO/IEC 15946-2. Information Technology — Security Techniques — Cryptographic Techniques Based on Elliptic Curves — Part 2: Digital Signatures, 2002.

[5] KCDSA Task Force Team. The Korean Certificate-Based Digital Signature Algorithm, August 1988. available at `http://grouper.ieee.org/groups/1363/P1363a/PSSigs.html`.