Some Security Comparisons of GOST R 34.10-2012 and ECDSA Signature Schemes

Trieu Quang Phong
Nguyen Quoc Toan

Institute of Cryptography Science and Technology
Gover. Info. Security Committee, Viet Nam

June 6, 2017

Content

Part I:
Introduction

Background

- ECDSA and GOST R 34.10-2012 are considered as the secure and popular signature schemes recently. These schemes are the elliptic curve versions of DSA and GOST R 34.10-94, respectively. However, there are not much research comparing the efficiency and security of these schemes.
- The common point between GOST R 34.10-2012 and ECDSA is that in these two schemes the value of the hash function only depends on the signed message. This implies that there is no security proof for these two schemes in the random oracle model.

Related Works

- In [3], E. Brickell, D. Pointcheval, S. Vaudenay, M. Yung provided two variants of DSA, and then proved the security for these variants in the random oracle model using *the forking lemma.*
- In [1], J. Malone-Lee and N.P. Smart described two variants ECDSA-II and ECDSA-III of ECDSA which are secure against the no-message attack in the random oracle model using *the Improved Forking Lemma*
- In [2], J. Stern, D. Pointcheval, J. Malone-Lee and N.P. Smart provided two flaws of ECDSA, namely *duplicate signature* and *malleability*.

Our Works

We will provide two comparisons between GOST R 34.10-2012 and ECDSA by:

- Applying the method of J. Malone-Lee and N.P. Smart in [1] for GOST R 34.10-2012, and then obtain two variants GOST-I and GOST-II.
- Showing that GOST R 34.10-2012 is able to resist the two flaws of ECDSA in [2].

Part II:
Description of GOST R 34.10-2012 and ECDSA

Notations

p	Prime number, $p > 3$.
$\mathbb{F}_p$	Finite prime field represented by a set of integers $\{0, 1, ..., p-1\}$.
$E(\mathbb{F}_p)$	Elliptic curve defined on $\mathbb{F}_p$.
$\lvert E(\mathbb{F}_p)\rvert$	The number of $\mathbb{F}_p$-rational points on $E(\mathbb{F}_p)$.
$\mathcal{O}$	Zero point of the elliptic curve $E(\mathbb{F}_p)$.
n	A prime divisor of $\lvert E(\mathbb{F}_p)\rvert$.
c	Cofactor, $c = \frac{\lvert E(\mathbb{F}_p)\rvert}{n}$.
P	Elliptic curve point of order n.
H, H_{GOST}	Hash function.

Notations

A	Signer.
$\mathcal{A}$	Attacker.
$\in_R$	Generate a random integer.
d	Integer number, the signature (private) key of signer A.
Q	Elliptic curve point, the verification (public) key of signer A.
k	Ephemeral secret value.
M	Signer's message.
(r, s)	digital signature for the message M.
x_R, y_R	Coordinates of elliptic point R.
$\log(x)$	Binary logarithm of x.

Description of GOST R 34.10-2012

Signing	**Verification**
1. $e = H_{GOST}(M)$	1. $e = H_{GOST}(M)$
2. $k \in_R [1, n-1]$	2. $u_1 = se^{-1} \ mod \ n$
3. $R = kP$	3. $u_2 = -re^{-1} \ mod \ n$
4. $r = x_R \ mod \ n$	4. $R = u_1 P + u_2 Q$
5. $s = ke + dr \ mod \ n$	5. $v = x_R \ mod \ n$
6. Output (r, s)	6. Accept iff $r = v$

Description of ECDSA

Signing	**Verification**
1. $k \in_R [1, n-1]$	1. $h = H(M)$
2. $R = kP$	2. $u_1 = hs^{-1} \ mod \ n$
3. $r = x_R \ mod \ n$	3. $u_2 = rs^{-1} \ mod \ n$
4. $h = H(M)$	4. $R = u_1 P + u_2 Q$
5. $s = k^{-1}(h + dr) \ mod \ n$	5. $v = x_R \ mod \ n$
6. Output (r, s)	6. Accept iff $r = v$

Part III: Comparison of GOST R34.10-2012 and ECDSA scheme via two flaws of ECDSA

Two flaws of ECDSA

The first flaw (Duplicate signature)

For any two distinct messages m_1 and m_2, we always can generate an ECDSA signature which is valid for both messages, if we have a possible control on the key generation.

For any $m_1 \neq m_2$, compute $h_1 = H(m_1)$ and $h_2 = H(m_2)$. Generate $k \in_R \{1, .., n-1\}$, compute $r = x_{kP}$, and set

$$d = -((h_1 + h_2))/2r \ mod \ n \text{ and } Q = dP. \quad (1)$$

Finally, compute $s = k^{-1}(h_1 + dr) \ mod \ n$. Hence, (r, s) is a valid ECDSA signature on m_1 with the public/ private key pair (Q, d). And, (r, s) is also a valid ECDSA signature on m_2, since

$$R = \frac{h_2}{s}P + \frac{r}{s}Q = \frac{(h_2 + rd)}{k^{-1}(h_1 + dr)}P = -kP. \quad (2)$$

Two flaws of ECDSA

The second flaw

From an ECDSA signature (r, s) of a message m, one can derive another valid ECDSA signature of m, namely $(r, -s)$.

If (r, s) is a valid ECDSA signature of m, then $(r, -s)$ is also a valid signature, since

$$r = x_{\frac{H(m)}{s}P+\frac{r}{s}Q} = x_{-(\frac{H(m)}{s}P+\frac{r}{s}Q)} = x_{\frac{H(m)}{-s}P+\frac{r}{-s}Q}. \quad (3)$$

GOST R 34.10-2012 resist two flaws of ECDSA

- The main cause of these flaws of ECDSA is the property: $x_R = x_{-R}, \forall R \in E(\mathbb{F}_p)$.
- Another cause:

  - For first flaw, the following equation system:

$$\begin{cases} s &= k^{-1}(H(m_1) + dr) \ mod \ n \\ s &= -k^{-1}(H(m_2) + dr) \ mod \ n. \end{cases} \quad (4)$$

    always has solution $d = -\frac{H(m_1)+H(m_2)}{2r} \ mod \ n$.
  - For second flaw, if (r, s) is an ECDSA signature on m, and $R = s^{-1}H(m)P + s^{-1}rQ$, it is easy to compute the elliptic point $-R$ from $(r, -s)$ and m.

GOST R 34.10-2012 resist two flaws of ECDSA

If the values of the hash function are uniformly distributed, GOST R 34.10-2012 is able to resist two flaws of ECDSA.

- For first flaw, the following equation system:

$$\begin{cases} s &= kH_{GOST}(m_1) + dr \mod n \\ s &= -kH_{GOST}(m_2) + dr \mod n. \end{cases} \tag{5}$$

  always has no solution with unknown d.

- For second flaw, if (r, s) is a GOST R 34.10-2012 signature on m, and $R = s^{-1}H_{GOST}(m)P + s^{-1}rQ$, it is not easy to find (r', s') such that $-R$ is computed from (r', s') and m.

Part IV:

Constructing two variant of GOST R34.10-2012 in the way of ECDSA-II and ECDSA-III construction

Elliptic Curve Trusted El Gamal Type Signature Scheme – ECTEGTSS

A signature scheme is an ECTEGTSS if it has the following properties:

i. $E(\mathbb{F}_p)$ satisfies $|E(\mathbb{F}_p)| = c \cdot n$, where n prime and c small. A point $P \in E(\mathbb{F}_p)$ of order n and the underlying group $\langle P \rangle$.

ii. It uses two function G and H, with ranges $\mathcal{G}$ and $\mathcal{H}$ respectively. H is modelled as a random oracle and G is (multi)-collision-resistance or (multi)-collision-freeness.

iii. There are three functions:

$$F_1(Z_n, Z_n, \mathcal{G}, \mathcal{H}) \to Z_n;\ F_2(Z_n, \mathcal{G}, \mathcal{H}) \to Z_n;\ F_3 : (Z_n, \mathcal{G}, \mathcal{H}) \to Z_n$$

satistfying for all $(k, d, r, h) \in (Z_n, Z_n, \mathcal{G}, \mathcal{H})$,

$$F_2(F_1(k, d, r, h), r, h) + dF_3(F_1(k, d, r, h), r, h) = k \ mod \ n. \quad (6)$$

Elliptic Curve Trusted El Gamal Type Signature Scheme – ECTEGTSS

iv. private key d, Q, public key $Q = dP$.

v. To sign a message m, the signer picks $k \in Z_n^*$, computes $R = kP$ and $r = G(R)$. Then gets $h = H(m||r)$ and computes $s = F_1(k, x, r, h)$. The signature on m is (s, r, h).

vi. To verify the signature (s, r, h) on a message m the verifier computes $e_P = F_2(s, r, h)$, $e_Q = F_3(s, r, h)$ and $W = e_P P + e_Q Q$. Then checks $r = G(W)$ and $h = H(m||r)$.

vii. The functions F_2 and F_3 must satisfy the following one-to-one condition: for given r, e_P and e_Q, there exists a unique pair (h, s) such that

$$e_P = F_2(s, r, h) \text{ and } e_Q = F_3(s, r, h).$$

The Improved Forking Lemma

The Improved Forking Lemma

Let us consider a probabilistic polynomial time Turing machine $\mathcal{A}$, called the attacker, and a probabilistic polynomial time simulator $\mathcal{B}$. If $\mathcal{A}$ can find with probability $\varepsilon > 4/p$ a verifiable tuple (M, R, S, T, U) with less than q queries to the hash function, for a new message M and for a U directly defined by H, then with a constant probability $1/96$, with $(1 + 24q\ell \log(2\ell))/\varepsilon$ replays of $\mathcal{A}$ and $\mathcal{B}$ with different random oracles, $\mathcal{A}$ will output $\ell + 1$ verifiable tuples $(M_i, R_i, S_i, T_i, U_i)_{i=1,..,\ell+1}$ such that the U_i are pairwise distinct, and all the R_i equal for TEGTSS-I schemes but all the (M_i, T_i) equal for TEGTSS-II schemes.

Note that, if a signature scheme is an ECTEGTSS is also, it is also a TEGTSS-II.

ECDSA-II

The first variant of ECDSA is called ECDSA-II, which replace $h = H(m)$ with $h = H(m||r)$.

Signing	**Verification**
1. $k \in_R [1, n-1]$	1. $h = H(M\|\|r)$
2. $R = kP$	2. $u_1 = hs^{-1} \ mod \ n$
3. $r = x_R \ mod \ n$	3. $u_2 = rs^{-1} \ mod \ n$
4. $h = H(M\|\|r)$	4. $R = u_1 P + u_2 Q$
5. $s = k^{-1}(h + dr) \ mod \ n$	5. $v = x_R \ mod \ n$
6. Output (r, s)	6. Accept iff $r = v$

- ECDSA-II can not resist the two flaws of ECDSA.
- In [1], ECDSA-II is proved secure against the no-message attack in the random oracle model.

ECDSA-III

ECDSA-III is identical to ECDSA-II, except that replace $r = x_R \ mod \ n$ with $r = x_R + y_R$.

Signing	**Verification**
1. $k \in_R [1, n-1]$	1. $h = H(M\|\|r)$
2. $R = kP$	2. $u_1 = hs^{-1} \ mod \ n$
3. $r = x_R + y_R$	3. $u_2 = rs^{-1} \ mod \ n$
4. $h = H(M\|\|r)$	4. $R = u_1 P + u_2 Q$
5. $s = k^{-1}(h + dr) \ mod \ n$	5. $v = x_R + y_R$
6. Output (r, s)	6. Accept iff $r = v$

- ECDSA-III can resist the two flaws of ECDSA.
- In [1], ECDSA-III is also proved secure against the no-message attack in the random oracle model.

Two variants of GOST R34.10-2012

We consider two variants of GOST R 34.10-2012, called **GOST-I** and **GOST-II**. We also assume that the parameters p and n for these variants satisfy:

- If $2^{254} < n < 2^{256}$ then $p < 2^{256}$.
- If $2^{508} < n < 2^{512}$ then $p < 2^{512}$.

GOST-I

In a similar way to gain ECDSA-II, we obtain GOST-I by replacing the hash function evaluation $e = H_{GOST}(M)$ in GOST R 34.10-2012 by $e = H_{GOST}(M||r)$.

Signing	**Verification**
1. $k \in_R [1, n-1]$	1. $e = H_{GOST}(M\|\|r)$
2. $R = kP$	2. $u_1 = se^{-1} \ mod \ n$
3. $r = x_R \ mod \ n$	3. $u_2 = -re^{-1} \ mod \ n$
4. $e = H_{GOST}(M\|\|r)$	4. $R = u_1P + u_2Q$
5. $s = kh + dr \ mod \ n$	5. $v = x_R \ mod \ n$
6. Output (r, s)	6. Accept iff $r = v$

The results on GOST-I

GOST-I is able to resist two flaws of ECDSA

The GOST-I signature scheme is an ECTEGTSS.

Suppose an adversary $\mathcal{A}$ against GOST-I exists which succeeds with probability $\varepsilon > 4/p$ after q queries to the random oracle H, then one can solve the discrete logarithm problem in $E(\mathbb{F}_p)$ using

$$\frac{1 + 768q \log 64}{\varepsilon} = \frac{1 + 4608q}{\varepsilon}$$

replays of $\mathcal{A}$ with probability greater than $1/100$.

GOST-II

GOST-II is identical to GOST-I, except that replace $r = x_R \ mod \ n$ with $r = x_R + y_R$.

Signing	**Verification**
1. $k \in_R [1, n-1]$	1. $e = H_{GOST}(M\|\|r)$
2. $R = kP$	2. $u_1 = se^{-1} \ mod \ n$
3. $r = x_R + y_R \ mod \ n$	3. $u_2 = -re^{-1} \ mod \ n$
4. $e = H_{GOST}(M\|\|r)$	4. $R = u_1 P + u_2 Q$
5. $s = kh + dr \ mod \ n$	5. $v = x_R + y_R \ mod \ n$
6. Output (r, s)	6. Accept iff $r = v$

The results on GOST-II

GOST-II is able to resist two flaws of ECDSA

The GOST-II signature scheme is an ECTEGTSS.

Suppose an adversary $\mathcal{A}$ against GOST-II exists which succeeds with probability $\varepsilon > 4/p$ after q queries to the random oracle H, then one can solve the discrete logarithm problem in $E(\mathbb{F}_p)$ using

$$\frac{1 + 72q \log 6}{\varepsilon}$$

replays of $\mathcal{A}$ with probability greater than $1/100$.

Conclusion

Feature	ECDSA-II	ECDSA-III	GOST-I	GOST-II	ECDSA	GOST R 34.10-2012
Resistance to two flaws of ECDSA	No	Yes	Yes	Yes	No	Yes
The security proof in the random oracle	Secure against the no-message attack	Secure against the no-message attack	Secure against the no-message attack	Secure against the no-message attack	No proof	No proof

References

1. J. Malone-Lee and N. P. Smart, "Modifications of ECDSA", International Workshop on Selected Areas in Cryptography Selected Areas in Cryptography, 2002, pages 1-12.
2. J. Stern, D. Pointcheval, J. Malone-Lee, Nigel P. Smart, "Flaws in Applying Proof Methodologies to Signature Schemes", Annual International Cryptology Conference, CRYPTO 2002: Advances in Cryptology CRYPTO 2002, pages 93-110.
3. E. Brickell, D. Pointcheval, S. Vaudenay, M. Yung, "Design Validations for Discrete Logarithm Based Signature Schemes", PKC 2000: Public Key Cryptography, pages 276-292.

Thanks for your listen!