Speed-ups of Elliptic Curve-Based Schemes

René Struik
Certicom Research
e-mail: rstruik@certicom.com

Outline

- ECDSA signature scheme
- Fast ECDSA signature scheme
- Speed-ups:
  - ECDSA fast verification
  - ECDSA certificate verification and ECC-based key agreement (ECDH, ECMQV)
  - Batch ECDSA verification
- How to get from ECDSA to Fast ECDSA

ECDSA signature scheme

System-wide parameters

Elliptic curve of prime order n with generator G. Hash function h.

Initial set-up

Signer A selects private key $d \in [1, n-1]$ and publishes its public key $Q = dG$.

Signature generation

INPUT: Message m, private key d.
OUTPUT: Signature (r, s).

ACTIONS:

1. Compute $e := h(m)$.
2. Select random $k \in [1, n-1]$.
3. Compute $R := kG$ and map R to r.
4. Compute $s := k^{-1}(e + d\,r) \bmod n$.
5. If $r, s \in [1, n-1]$, return (r, s); otherwise, go to Step 2.

Signature verification

INPUT: Message m, signature (r, s); Public signing key Q of Alice.
OUTPUT: <u>Accept</u> or <u>reject</u> signature.

ACTIONS:

1. Compute $e := h(m)$.
2. Check that $r, s \in [1, n-1]$. If verification fails, return 'reject'.
3. Compute $R' := s^{-1}(e\,G + r\,Q)$.
4. Check that R' maps to r. If verification succeeds, return '<u>accept</u>'; otherwise return '<u>reject</u>'.

<u>Non-repudiation:</u> Verifier knows the true identity of the signing party, since the public signing key Q is bound to signing party Alice.

Fast ECDSA signature scheme

System-wide parameters

Elliptic curve of prime order n with generator G. Hash function h.

Initial set-up

Signer A selects private key $d \in [1, n-1]$ and publishes its public key $Q = dG$.

Signature generation

INPUT: Message m, private key d.
OUTPUT: Signature (R, s).

ACTIONS:

1. Compute $e := h(m)$.
2. Select random $k \in [1, n-1]$.
3. Compute $R := kG$ and map R to r.
4. Compute $s := k^{-1}(e + d\, r) \bmod n$.
5. If $r, s \in [1, n-1]$, return (R, s); otherwise, go to Step 2.

Signature verification

INPUT: Message m, signature (R, s); Public signing key Q of Alice.
OUTPUT: Accept or reject signature.

ACTIONS:

1. Compute $e := h(m)$.
2. Map R to r.
3. Check that $r, s \in [1, n-1]$. If verification fails, return 'reject'.
4. Check that $R = s^{-1}(e\, G + r\, Q)$. If verification succeeds, return 'accept'; otherwise return 'reject'.

Non-repudiation: Verifier knows the true identity of the signing party, since the public signing key Q is bound to signing party Alice.

Fast ECDSA signature scheme

Computational aspects

Ordinary signature verification	Fast signature verification
ACTIONS: **...** **3. Compute** $R' := (e\, s^{-1})\, G + (r\, s^{-1})\, Q$. **4. Check that** R' **maps to** r. **...**	**ACTIONS:** **...** **2. Map** R **to** r. **4. Check that** $R = (e\, s^{-1})\, G + (r\, s^{-1})\, Q$. **...**

Ordinary signature verification

Compute expression $R' := (e\, s^{-1})\, G + (r\, s^{-1})\, Q$.

<u>Cost:</u> *full-size* linear combination of *known* point G and *unknown* point Q.

Fast signature verification

Evaluate expression $\Delta := s^{-1}\, (e\, G + r\, Q) - R$ and check that $\Delta = O$,
by verifying instead

$$\mu\, \Delta := (\mu\, e\, s\text{-}1)\, G + (\mu\, r\, s\text{-}1)\, Q - \mu\, R = O \text{ for suitable } \mu \in [1, n\text{-}1].$$

Speed-up: 40% for prime curves and binary non-Koblitz curves.

Fast ECDSA and speed-ups

Speed-ups for the following curves:
– NIST prime curves, 'Suite B' curves, Brainpool curves
– NIST random binary curves

Fast verification of ECDSA signatures ([2]):
40% speed-up compared to ordinary approach

ECDSA certificate verification + Static ECDH/ECMQV ([6]):
Speed-up incremental cost ECDSA verify compared to separate approach: -
2.4x speed-up (compared to ordinary ECDSA verify)
1.7x (compared to Fast ECDSA verify)
Simple side channel resistance virtually for free

Batch verification of ECDSA signatures ([3]):
Dependent on number of signatures involved.

ECDSA vs. Fast ECDSA

Security of Fast ECDSA

Both schemes are equally secure: ECDSA has signature (r, s) if and only if Fast ECDSA has signature (R, s) where R maps to r.

ECDSA signature verification

- Convert ECDSA signature (r, s) to Fast ECDSA signature (R, s)
- Verify Fast ECDSA signature (R, s)

Note:

- Conversion generally yields *pair* (R, $-R$) of *candidate points* that map to r.
- Verification involves trying out all those candidate points not discarded based on some side constraints (the so-called *admissible points*).

How to ensure only one admissible point:

- Generate ECDSA signature with k such that y-coordinate of $R:=kG$ can be prescribed. (If necessary, change the sign of k.)
- Use the fact that (r, s) is a valid ECDSA signature if and only if (r, $-s$) is.

Conversion of ECDSA to Fast Verify friendly format: via simple post-processing

Further reading

1. ANSI X9.63-2001, ‘Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography,’ American National Standard for Financial Services, American Bankers Association, November 20, 2001.
2. A. Antipa, D.R. Brown, R. Gallant, R. Lambert, R. Struik, S.A. Vanstone, ‘Accelerated Verification of ECDSA Signatures,’ in *Proceedings of Selected Areas in Cryptography – SAC2005*, B. Preneel, S. Tavares, Eds., Lecture Notes in Computer Science, Vol. 3897, pp. 307-318, New York: Springer, 2006.
3. M. Bellare, J.A. Garay, T. Rabin, ‘Fast Batch Verification for Modular Exponentiation and Digital Signatures,’ in *Proceedings of Advances in Cryptology – EUROCRYPT'98*, K. Nyberg, Ed., Lecture Notes in Computer Science, Vol. 1403, pp. 236-250, New York: Springer-Verlag, 1998.
4. FIPS Pub 186-2, *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication 186-2, US Department of Commerce/National Institute of Standards and Technology, Gaithersburg, Maryland, USA, January 27, 2000. (Includes change notice, October 5, 2001.)
5. NIST SP800-56a, *Recommendation for Pair-wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, March 8, 2007.
6. R. Struik, ‘Combined Verifications and Key Computations,’ draft.