EdDSA for more curves

Daniel J. Bernstein, University of Illinois at Chicago; TU/e
Simon Josefsson, Simon Josefsson Datakonsult
Tanja Lange, Technische Universiteit Eindhoven
Peter Schwabe, Radboud Universiteit
Bo-Yin Yang, Academia Sinica

CFRG, IETF 93, Prague

22 July 2015
Background

How ECC signatures fail:

- PlayStation 3 disaster.
Background

How ECC signatures fail:

- PlayStation 3 disaster.
- Hash-function collisions.
- Biased nonces leaking secret key.
- Timing leaks from, e.g., inversion mod group order.

1992 Rivest (on DSA):
"The poor user is given enough rope with which to hang himself—something a standard should not do."
Background

How ECC signatures fail:
- PlayStation 3 disaster.
- Hash-function collisions.
- Biased nonces leaking secret key.
- Timing leaks from, e.g., inversion mod group order.
- Being so complex that errors are bound to occur.
- Being so slow that protocol designer skips signatures.
- Being so slow that implementor turns them off.

1992 Rivest (on DSA):
"The poor user is given enough rope with which to hang himself—something a standard should not do."
Background

How ECC signatures fail:

- PlayStation 3 disaster.
- Hash-function collisions.
- Biased nonces leaking secret key.
- Timing leaks from, e.g., inversion mod group order.
- Being so complex that errors are bound to occur.
- Being so slow that protocol designer skips signatures.
- Being so slow that implementor turns them off.

1992 Rivest (on DSA):
“The poor user is given enough rope with which to hang himself—something a standard should not do.”
The Ed25519 signature system

2011 Bernstein–Duif–Lange–Schwabe–Yang
“High-speed high-security signatures”
ed25519.cr.yp.to:

Eliminate failures.
The Ed25519 signature system

2011 Bernstein–Duif–Lange–Schwabe–Yang
“High-speed high-security signatures”
ed25519.cr.yp.to:

Take advantage of crypto research:
- Curve25519.
- Edwards curves.
- Schnorr signatures, including collision resilience. (Schnorr patent expired 2008.)
- Conservative hash functions.
- Fast batch verification.
- Barwood–Wigley pseudorandom nonce generation.
Nicolai Brown is tracking applications and implementations:  
ianix.com/pub/ed25519-deployment.html

Examples of applications:

- OpenSSH.
- GnuPG.
- GNUnet.
- DNSCrypt.
- OpenBSD’s signify.

Many independent interoperable implementations.
A few examples of Ed25519 implementations

**Fast** constant-time implementation from 2015 Chou:
- 57164 cycles for keygen on Intel Sandy Bridge.
- 63526 cycles for sign.
- 205741 cycles for (non-batch) verify. Compare to 430000 cycles for OpenSSL 1.0.2 ecdsap256 verify.

**Small** constant-time implementations of Salsa20+Poly1305+X25519+SHA-512+Ed25519:
- 2013 Hutter–Schwabe “NaCl on 8-bit AVR microcontrollers”: 17366 bytes of object code.

---

EdDSA  
eprint.iacr.org/2015/677  
Speakers: Daniel J. Bernstein (UIC, TU/e) and Tanja Lange (TU/e)  
5
New: EdDSA for more curves

Ed25519 is an example of “EdDSA” defined in 2011 paper. 2015 Bernstein–Josefsson–Lange–Schwabe–Yang “EdDSA for more curves”:

- Easy extension of original EdDSA definition.
- Ed25519 is still an example!
New: EdDSA for more curves

Ed25519 is an example of “EdDSA” defined in 2011 paper.

“EdDSA for more curves”:

- Easy extension of original EdDSA definition.
- Ed25519 is still an example!
- Also allows Ed448-Goldilocks.
- Also allows Curve41417 and E-521.
New: EdDSA for more curves

Ed25519 is an example of “EdDSA” defined in 2011 paper. 2015 Bernstein–Josefsson–Lange–Schwabe–Yang “EdDSA for more curves”:

- Easy extension of original EdDSA definition.
- Ed25519 is still an example!
- Also allows Ed448-Goldilocks.
- Also allows Curve41417 and E-521.
- Also explicitly describes prehashing: e.g., GnuPG uses Ed25519-SHA-512 to sign \( \text{SHA-256}(m) \).

Note: Mixing SHA-256+SHA-512 is bad for code size!
[switch to browser showing merged Python implementation for comparing details of signature proposals]