Hedged ECDSA and EdDSA Signatures

draft-irtf-cfrg-det-sigs-with-noise-03

John Preuß Mattsson
Ericsson

Erik Thormarker
Ericsson

Sini Ruohomaa
Ericsson

Changes between -00 and -03

Changes between -00 and -03:

- Changed name to "Hedged ECDSA and EdDSA Signatures".
- **EdDSA and ECDSA:** Added a second 000... padding that separates the context from the hash of the private key (EdDSA), aligning with BSI recommendations (suggested by several people).
- **EdDSA:** Removed incorrect statement that context fits in first block (issue raised by Ilari Liusvaara).
- **EdDSA:** Changed Hedged EdDSA order to "0x00 || Z || dom2(F, C)" instead of "dom2(F, C) || Z". This avoids collisions with RFC 8032 and aligns with Bernstein's recommendation to put Z before the context.
- **ECDSA:** "Clarified" that Z is different in step d and f and then changed "back" to a single Z. Different Zd and Zf are not compatible with HMAC_DRBG [NIST SP 800-90A] (issue raised by Danny Niu).
- **ECDSA:** Changed KMAC output length recommendations to avoid multiple invocations (issue raised by Danny Niu).
- Added more description about the construction and its security properties.
- Added empty test vector section as TODO.

External specifications:

- NIST has published FIPS 186-5 which includes EdDSA and Deterministic ECDSA.
- NIST has published Draft FIPS 204 specifying the quantum-resistant ML-DSA signature scheme.
  A change between Dilithium and ML-DSA is that ML-DSA uses hedged signing by default.

EdDSA Constructions in -03

Ed25519ph, Ed25519ctx, and Ed25519:

Compute SHA-512(0x00 || Z || dom2(F, C) || 000... || prefix || 000... || PH(M)), where M is the message to be signed, Z is 32 octets of random data, the number of zeroes 000... is chosen so that the lengths of (0x00 || Z || dom2(F, C) || 000...) and (prefix || 000...) are multiples of 128 octets. Interpret the 64-octet digest as a little-endian integer r.

Ed448ph and Ed448:

Compute SHAKE256(0x00 || Z || dom4(F, C) || 000... || prefix || 000... || PH(M), 114), where M is the message to be signed, and Z is 57 octets of random data, the number of zeroes 000... is chosen so that the length of (0x00 || Z || dom4(F, C) || 000...) and (prefix || 000...) are multiples of 136 octets. F is 1 for Ed448ph, 0 for Ed448, and C is the context to use. Interpret the 114-octet digest as a little-endian integer r.

ECDSA Constructions in -03

"Deterministic" ECDSA:

d. Set:

K = HMAC_K(V || 0x00 || Z || 000... || int2octets(x) || 000... || bits2octets(h1))

The number of zeroes 000... is chosen so that the length of (V || 0x00 || Z || 000...) and (int2octets(x) || 000...) are multiples of the block size of the hash function.

f. Set:

K = HMAC_K(V || 0x01 || Z || 000... || int2octets(x) || 000... || bits2octets(h1)

Note that the "internal octet" is 0x01 this time. The string(Z || 000... || int2octets(x) || 000.. || bits2octets(h1)), called provided_data in HMAC_DRBG, is the same as in step (d).

Open issues

https://github.com/cfrg/draft-irtf-cfrg-det-sigs-with-noise/issues

#3 - Allow switching order of Z and "prefix"

— Daniel Bernstein suggests allowing switching of Z and "prefix".

— Also, two useful rules of thumb regarding input concatenations (for motivating attacks see, e.g., Crypto 1995 Preneel--van Oorschot): 1. Never put a variable-length input anywhere but last. Previous positions can have fixed-length *hashes* of other variable-length inputs. 2. Put whatever is least likely to be attacker-predictable first. Both Z and "prefix" are less likely to be attacker-predictable than C; scenarios where RNG failures are the top threat should put "prefix" first, while scenarios where side channels are the top threat (which seems to be the scenario under discussion) should put Z first.

```
SHA-512(0x00 || Z || dom2(F, C) || 000... || prefix || 000... || PH(M))
```

#10 – Deterministic interface

— Daniel Bernstein suggests a deterministic interface where Z is an explicit input.

Open issues

https://github.com/cfrg/draft-irtf-cfrg-det-sigs-with-noise/issues

#6 - FIPS 186 compliant mode where message-dependent values are used as 'additional input'

— Cisco suggested a FIPS 186 compliant mode where message-dependent pseudorandom values are used as 'additional input' in the random number generation for randomized ECDSA. Such a mode could compliment the current hedged construction based on Deterministic ECDSA.

#7 - Algorithm naming

— Simon Josefsson suggests using the name 'R* for the variants, e.g., 'REd25519ph', 'REd25519ctx', 'REd25519 etc.

#8 - RECOMMENDED or MUST

— Rene Stuik suggests MUST instead of RECOMMENDED.

Next steps

When we agree on the constructions:

— Add test vectors including the random value Z (Request from Taylor R Campbell). Danny Niu has kindly offered to provide test vectors.

— People have suggested that they would like to assess the effect of adding random bytes in the hash computations on the pseudorandomness of the generated nonces and see proofs for the hedged constructions.

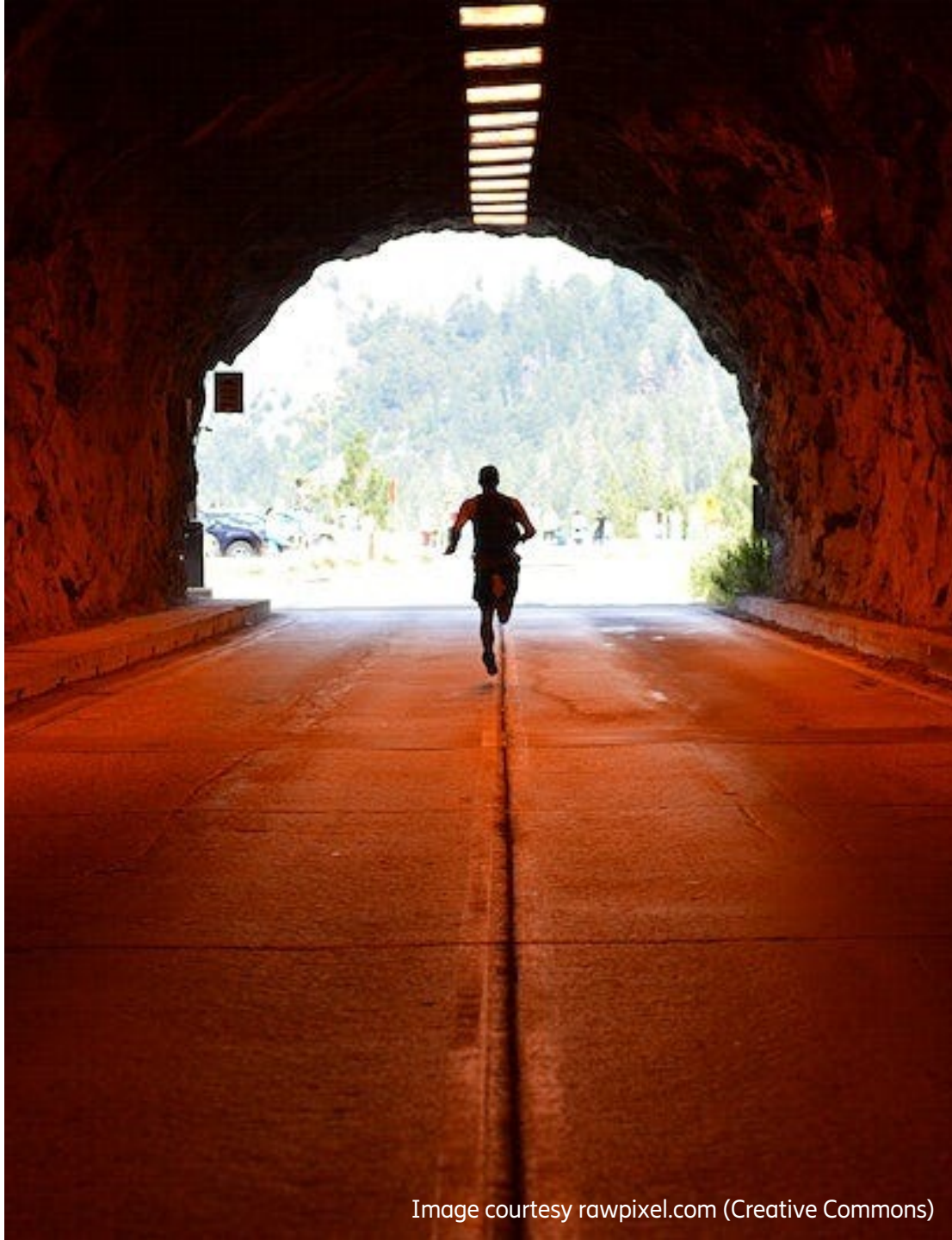

Image courtesy rawpixel.com (Creative Commons)