Threshold EdDSA/Schnorr Signatures

Information Technology Laboratory

Multi-Party Threshold Cryptography Project, Cryptographic Technology Group, Computer Security Division

NIST National Institute of Standards and Technology U.S. Department of Commerce

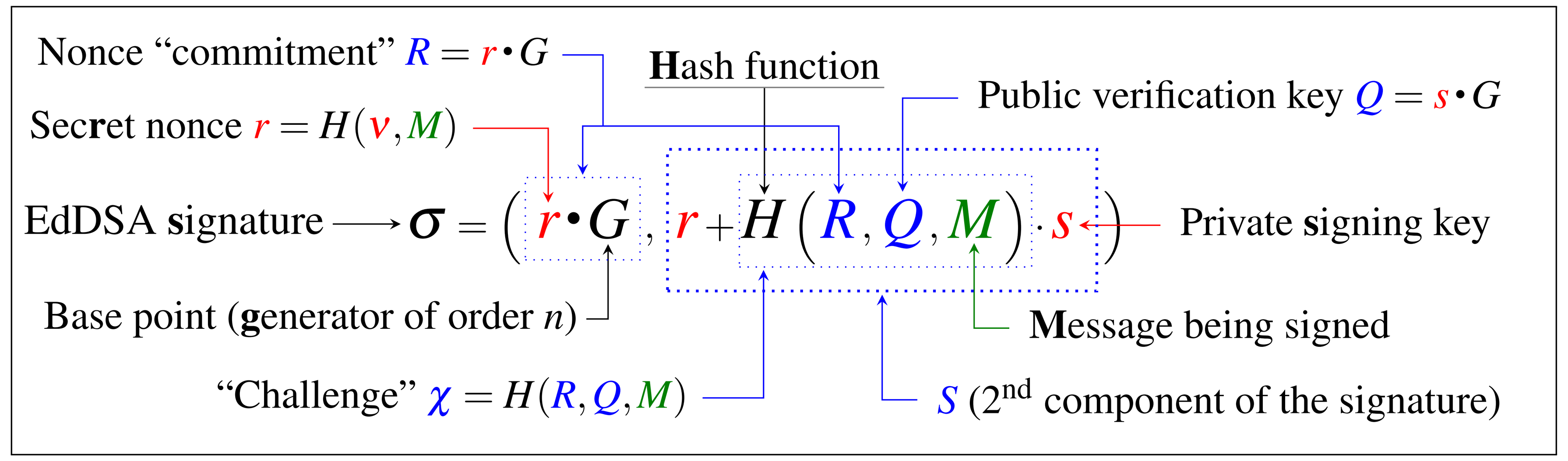
EdDSA / Schnorr Signatures

EdDSA = Edwards-Curve Digital Signature Algorithm

Known as a variant of the Schnorr signature scheme (1989)

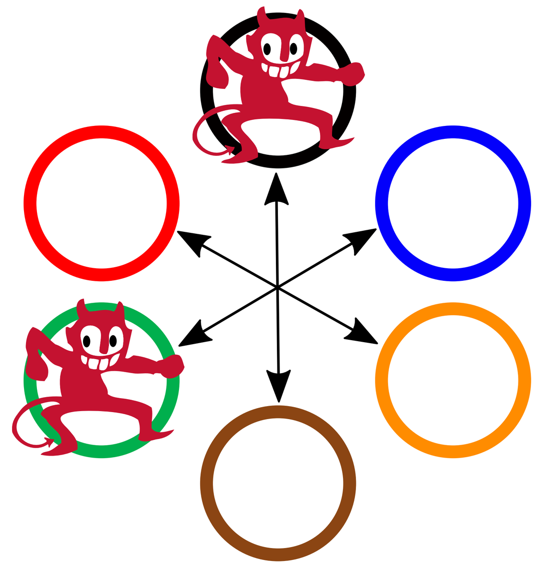
Has three operations: Keygen; **Sign**; Verify.

The EdDSA signature formula $\sigma = (R, S)$



The Threshold Paradigm

- The private key is split (via secret-sharing) across various parties.
- The signing goes through without the key being in any one place.
- It is secure even if a threshold number of parties is compromised.



Poster presented at the NIST-ITL Science Day 2022 (October 24th), by Luís Brandão (Foreign Guest Researcher at NIST, contractor from Strativia) and Michael Davidson. **Reference:** https://doi.org/10.6028/NIST.IR.8214B.ipd

Recent publication: IR 8214B

Notes on Threshold EdDSA/Schnorr Signatures

- Reviews security of conventional EdDSA
- Summarizes known threshold approaches
- Supports future call for threshold proposals

Some properties of conventional scheme:

- **Deterministic (non-verifiably):** The EdDSA standard asks for **deterministic** signatures (avoids problems with bad randomness), but malicious signer can undetectably randomize it.
- **Strong unforgeability (SUF):** Adversary (without private key) cannot by themself create a new signature (even for already signed messages).
- **Strong binding?** Standardized verification does not avoid the use of malformed keys. Malicious signer can find a different pair public key / message that is consistent with some signature.

"Threshold" considerations

- **Interchangeability:** threshold-produced EdDSA signatures must be verifiable with the conventional "Verify" algorithm. This allows probabilistic signatures.
- **Concurrency:** the set of "parties" must securely handle concurrent signature request (where the quorum may change).
- **Communication model:** Timing assumptions (e.g., synchrony) strongly affect the set of feasible protocols. Some primitives are often modularized, e.g., *broadcast*.
- **Good/Bad randomness:** Good randomness from a single party can be leveraged to improve the randomness used by other parties.

Takeaways:

- **Gained insights:** also useful for other schemes.
- **Intended followup:** Public call for threshold schemes; future guidance and recommendations.