Robustness for Dishonest Majority in Threshold ECDSA

Damian Straszak

Based on *Threshold ECDSA for Decentralized Asset Custody*

Joint work with: Adam Gągol, Jędrzej Kula and Michał Świętek
Threshold Signature schemes: BLS vs ECDSA

**BLS threshold signatures 2 of 4** [BLS’04, Boldyreva’03]

**Setup**

- Key generation

**Signing message m**

- Signature “shares” generated non-interactively.

- Verification of shares (uses public data)

- Signature(m) = Combine(s₁, s₄)
Threshold Signature schemes: BLS vs ECDSA

**ECDSA threshold signatures 2 of 4** [GG18, LN18, DKLS18, CCL+20, CMP20, GG20, …]

**Setup**

- Key generation

**Signing message m**

- Select a subset of 2 signers
- 2 of 2 signing protocol
  - Signature correct only if both participants honest!
Applications

**ECDSA** threshold signatures  
[GG18, LN18, DKLS18, CCL+20, CMP20, GG20, …]

**Great for**

distributing an ECDSA key over several devices.

**Not so great for**

holding a joint custody by a large number of nodes over a BTC account.

Many of the nodes can be dishonest, tricky to select “honest subset” of signers. BLS-style would work much better here.

Useful for blockchain “bridges”
New Threshold ECDSA Scheme

ECDSA threshold signatures 2 of 4 \([GKSS'20]\) based on \([LN'18]\)

**Setup**

- Key generation

**Signing message** \(m\)

Verification round 1

\[\text{Signature}(m) = \text{Combine}(s_1, s_4)\]

Verification round 2

Private Data (specific to each node)
Conclusion

● Robust threshold ECDSA scheme similar to “BLS style” (only little interaction required when signing)
● Useful when:
  ○ Large number of nodes
  ○ Nodes dishonest or prone to DDoS attacks
● Experiments: scales to ~100 nodes with <1 sec signing time

Future work:
● Setup not quite robust yet
● Protocol heavy on ZKPs