BLS signature

Ethereum Sharding Research

Jeongho Jeon <maczniak@gmail.com>

September 6, 2018

Whitepaper Foundation, Nonce
(for internal discussion purposes only)

Whitepaper
Foundation

DISCLAIMER

This talk is based on *Compact Multi-Signatures for Smaller Blockchains* (2018). It refers to *Simple Schnorr Multi-signatures with Applications to Bitcoin* (2018).

BLS signature

Dan **B**oneh, Ben **L**ynn, Hovav **S**hacham, *Short Signatures from the Weil Pairing* (2004)

- multi-signature
- aggregate signature
- threshold signature

multi-signature 1/2

- not BIP 0011
  x sig$_1$ …sig$_m$ m[1] pubkey$_1$ …pubkey$_n$ n[2] `OP_CHECKMULTISIG`
- make the constant size signature out of many signatures of the same document (transaction)
- we can shorten the above script!
- there are multi-signature schemes that are based on various techniques: RSA, discrete logarithms, pairings, and lattices.
- BLS signature scheme is based on a pairing.

[1] 1-16

[2] 1-16

multi-signature 2/2

- vs Schnorr signature scheme (2006)
- Schnorr - aggregate only when signing, require multi-round protocol between signers
- BLS - can aggregate at later time, aggregate by a simple multiplication (see also “public key aggregation”), allow off-line signers

pairing 1/2

- $\hat{e}(R+S,T)=\hat{e}(R,T)\hat{e}(S,T)$, $\hat{e}(R,S+T)=\hat{e}(R,S)\hat{e}(R,T)$
- bilinear: $\hat{e}(aS,bT)=\hat{e}(S,T)^{ab}$
- key generation: $pk \leftarrow g_2^{sk}$
- sign: $\sigma \leftarrow H(m)^{sk}$
- verify: $\hat{e}(\sigma,g_2)=\hat{e}(H(m),pk)$
- $\hat{e}(H(m)^{sk},g_2)=\hat{e}(H(m),g_2^{sk})$

pairing 2/2

- $\sigma \leftarrow \sigma_1 \cdots \sigma_n$
- $\hat{e}(\sigma, g_2) = \hat{e}(H(m_1), pk_1) \cdots \hat{e}(H(m_n), pk_n)$
- $\hat{e}(\sigma, g_2) = \hat{e}(H(m), pk_1 \cdots pk_n)$

aggregate signature (AS)

Dan Boneh, Craig Gentry, Ben Lynn, Hovav Shacham, *Aggregate and Verifiably Encrypted Signatures from Bilinear Maps* (2003)

- aggregate signatures of many parties that sign each their own message (transaction) into a single short signature
- we can compress all signatures in a block!
- vs Γ-signature (without bilinear maps, 2013; Bitcoin application, 2018)

threshold signature

Alexandra Boldyreva, *Threshold signatures, multisignatures and blind signatures based on the gapDiffie-Hellman-group signature scheme* (2002)

- t-of-n signatures
- accountable-subgroup multi-signature (ASM)

BLS in cryptocurrencies

1. Hot-Stuff and CodeChain use the threshold signature for combining validators' signatures in a block header. It reduces communication complexity, too.
2. Chia (Script) has `OP_BLSAGGREGATE`.
3. DFINITY makes the random beacon by using the threshold version of BLS. (vs VDF)