Exploring the power of Threshold BLS

Pratyay Mukherjee

SUPRA

MPTS 2023: NIST Workshop on Multi-Party Threshold Schemes 2023
Sep 26, 2023

3.2. Category 2 (Cat2)

The goal of Cat2 is to enable submissions that make a strong case for certain threshold-feasible primitives that are not standardized by NIST. While the scope is wide, Cat2-submissions should be justified on the basis of the primitives being thresholdized having/enabling useful differentiating features, such as having/being: (i) **t**hreshold-**f**riendly(ier) (TF); (ii) based on alternative cryptographic assumptions (e.g., pairings), possibly **q**uantum-**r**esistant (QR) (e.g., lattice-based); (iii) useful probabilistic properties (e.g., determinism versus non-determinism), (iv) more efficient in a relevant metric, or/and (v) advanced functional features (e.g., allowing homomorphic computation over encrypted data).

Cat2 has eight subcategories, including five "regular" (somewhat matching the subcategories of Cat1), and three others ("advanced", "ZKPoK" and "gadgets"), as listed in Table 2:

- **"Regular":**
  - C2.1, for signing (e.g., verifiably-deterministic succinct signatures, and/or TF-QR);

BLS (Asiacrypt'01, JoC'04)

Short Signatures from the Weil Pairing

Dan Boneh*, Ben Lynn, and Hovav Shacham

Computer Science Department, Stanford University
{dabo,blynn,hovav}@cs.stanford.edu

Abstract. We introduce a short signature scheme based on the Computational Diffie-Hellman assumption on certain elliptic and hyper-elliptic curves. The signature length is half the size of a DSA signature for a similar level of security. Our short signature scheme is designed for systems where signatures are typed in by a human or signatures are sent over a low-bandwidth channel.

Recall: BLS Signature

Pairing(bilinear map) $e : G_1 \times G_2 \rightarrow G_T$: $e(g_1^a, g_2^b) = e(g_1^b, g_2^a) = e(g_1, g_2)^{ab}$

H : MSG $\rightarrow G_1$ (Random Oracle)

KGen $\rightarrow$ (sk, vk):
- $sk \leftarrow \$ \mathbb{Z}_p$
- $vk := g_2^{sk}$

Sign(sk,m) $\rightarrow$ (σ):
- $\sigma := H(m)^{sk}$

Verify(vk, m, σ) $\rightarrow$ 1/0
- RET $(e(H(m), vk) = e(\sigma, g_2))$

Recall: BLS Signature

Recall

– C2.1, for signing (e.g., verifiably-deterministic succinct signatures, and/or TF-QR);

Main Distinctive Features:

- Verifiably deterministic (Unique) ✓
- Succinct ✓
- **Key-homomorphism**
  - Any linear combination in the exponent:
    - $\sigma_i = H(m)^{sk_i}$
    - KEY-HOM($\sigma_1, \sigma_2 \ldots, \sigma_t$; $e_1, \ldots, e_t$):
      - $\sigma = \prod \sigma_i^{e_i} = H(m)^{\Sigma sk_i e_i}$

→ Readily threshold-friendly: simple design

- Use linear secret sharing for sk and use $e_i = \lambda_i$
- Non-interactive threshold signing

- **Key-homomorphism also in the vk**
  - Any linear combination in the exponent:
    - $vk_i = g_2^{sk_i}$
    - KEY-HOM($vk_1, vk_2 \ldots vk_t$; $e_1, \ldots, e_t$):
      - $vk = \prod vk_i^{e_i} = g_2^{\Sigma sk_i e_i}$

→ Multi-sig friendly: simple design

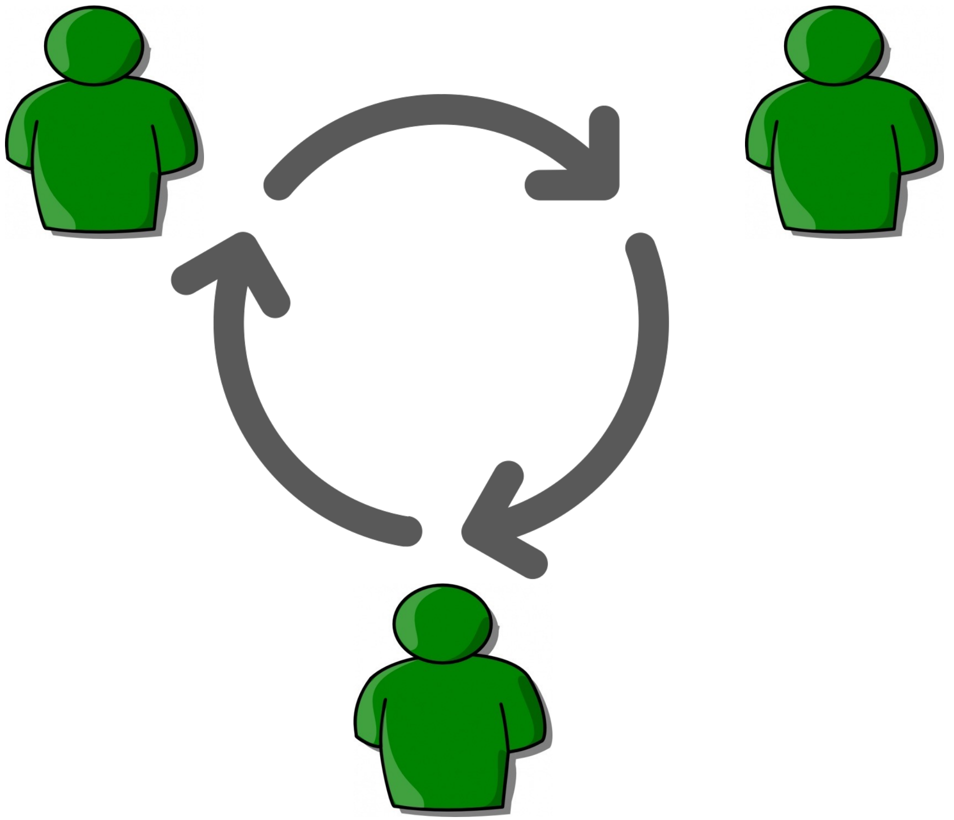
- Use linear secret sharing for sk and use $e_i = \lambda_i$
- Non-interactive, simple aggregation

Cons:

- Needs Bilinear Pairing
- Verification more expensive: Pairing
- Not PQ-secure ✗

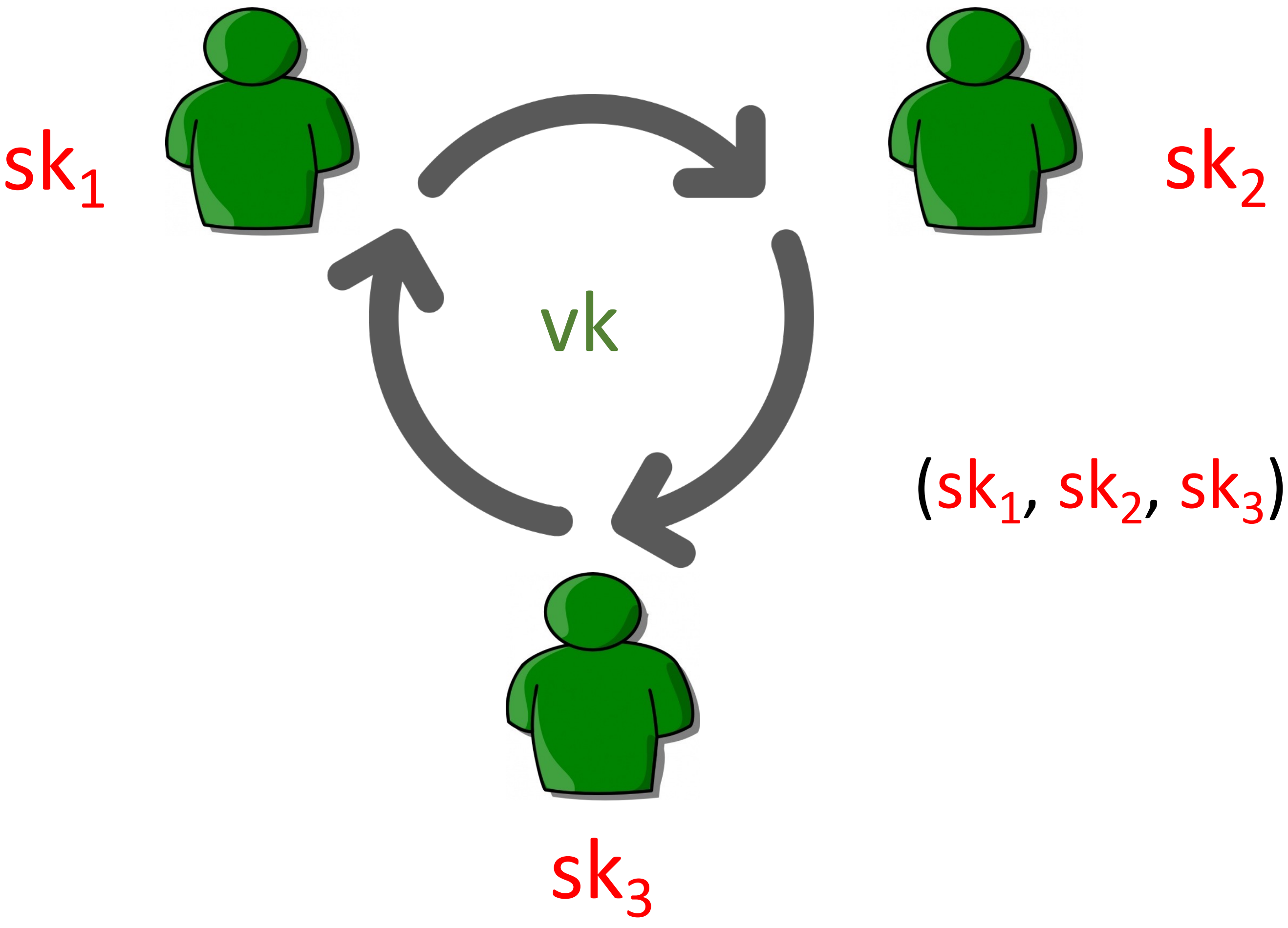
Recall: Threshold BLS (Example: N = 3, T = 2)

Dist-Kgen (similar to Schnorr/ECDSA)



Recall: Threshold BLS (Example: N = 3, T = 2)

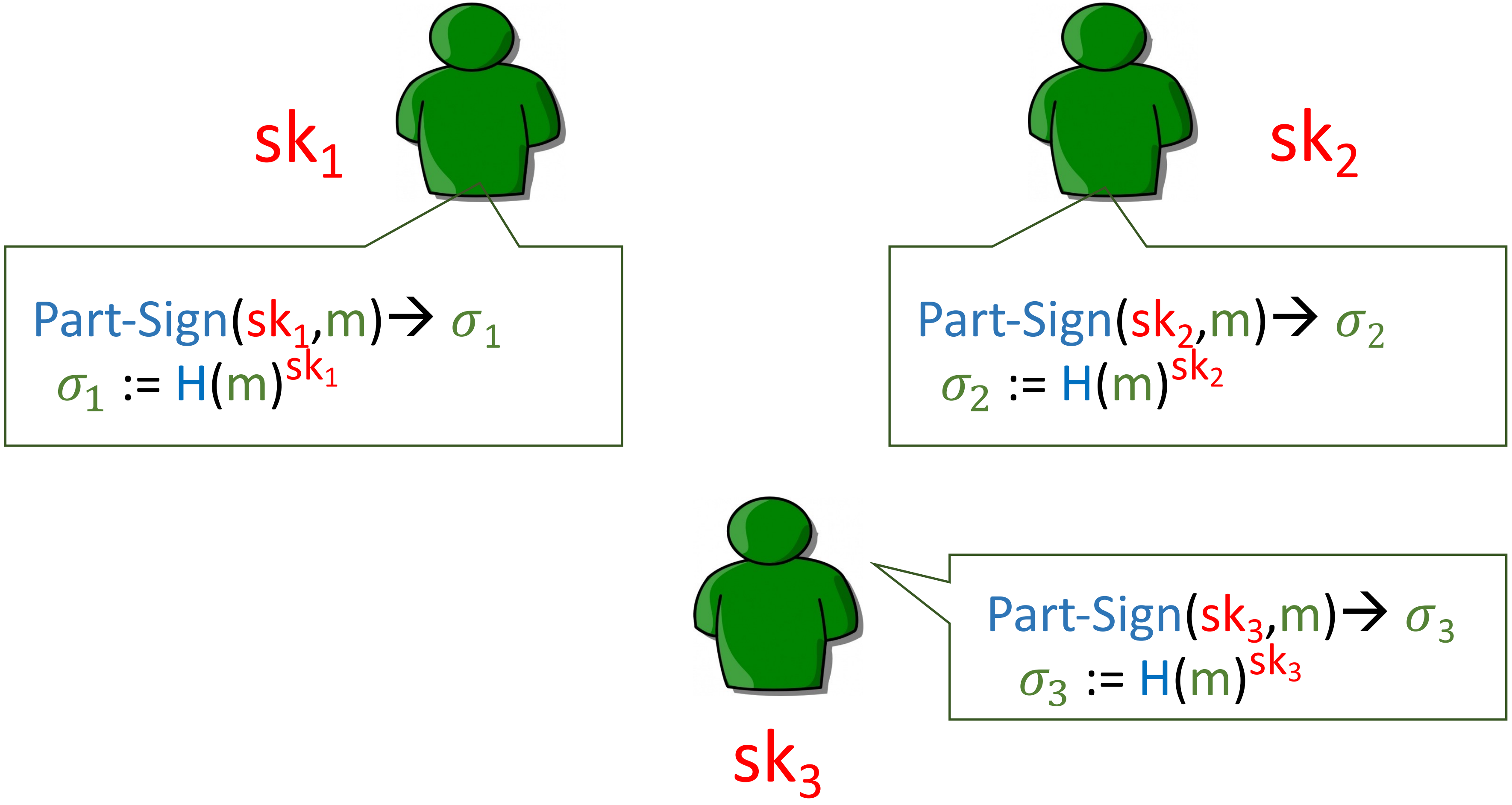
Dist-Kgen (similar to Schnorr/ECDSA)



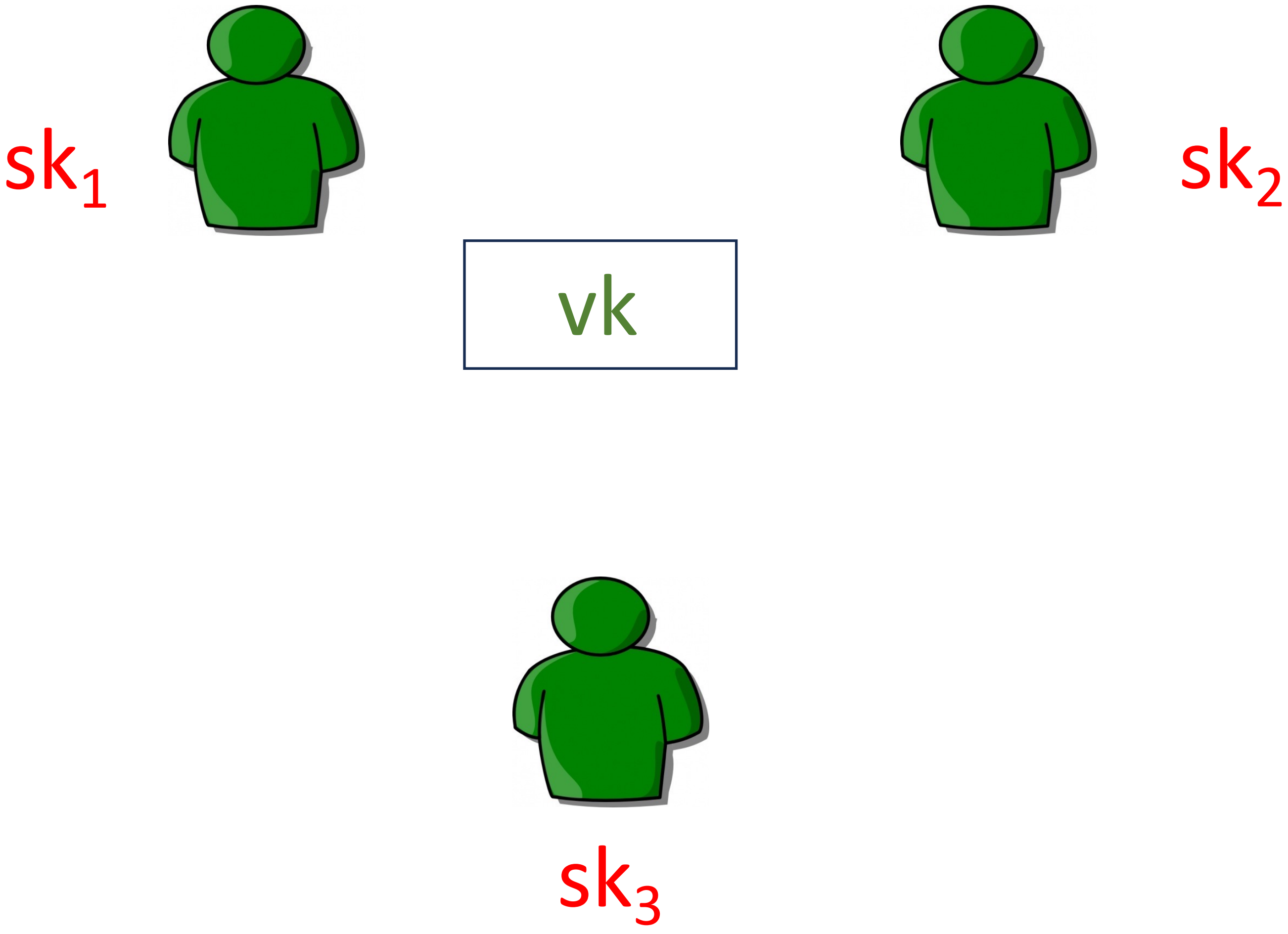
$(sk_1, sk_2, sk_3) \leftarrow \text{Shamir}_{2,3}(sk)$

Recall: Threshold BLS (Example: N = 3, T = 2)

Part-Sign (same as non-threshold BLS)



Recall: Threshold BLS (Example: N = 3, T = 2)



Aggregate $(\sigma_1, \sigma_3) \rightarrow \sigma$:

RET σ := KEY-HOM$(\sigma_1, \sigma_3, \lambda_1, \lambda_3)$

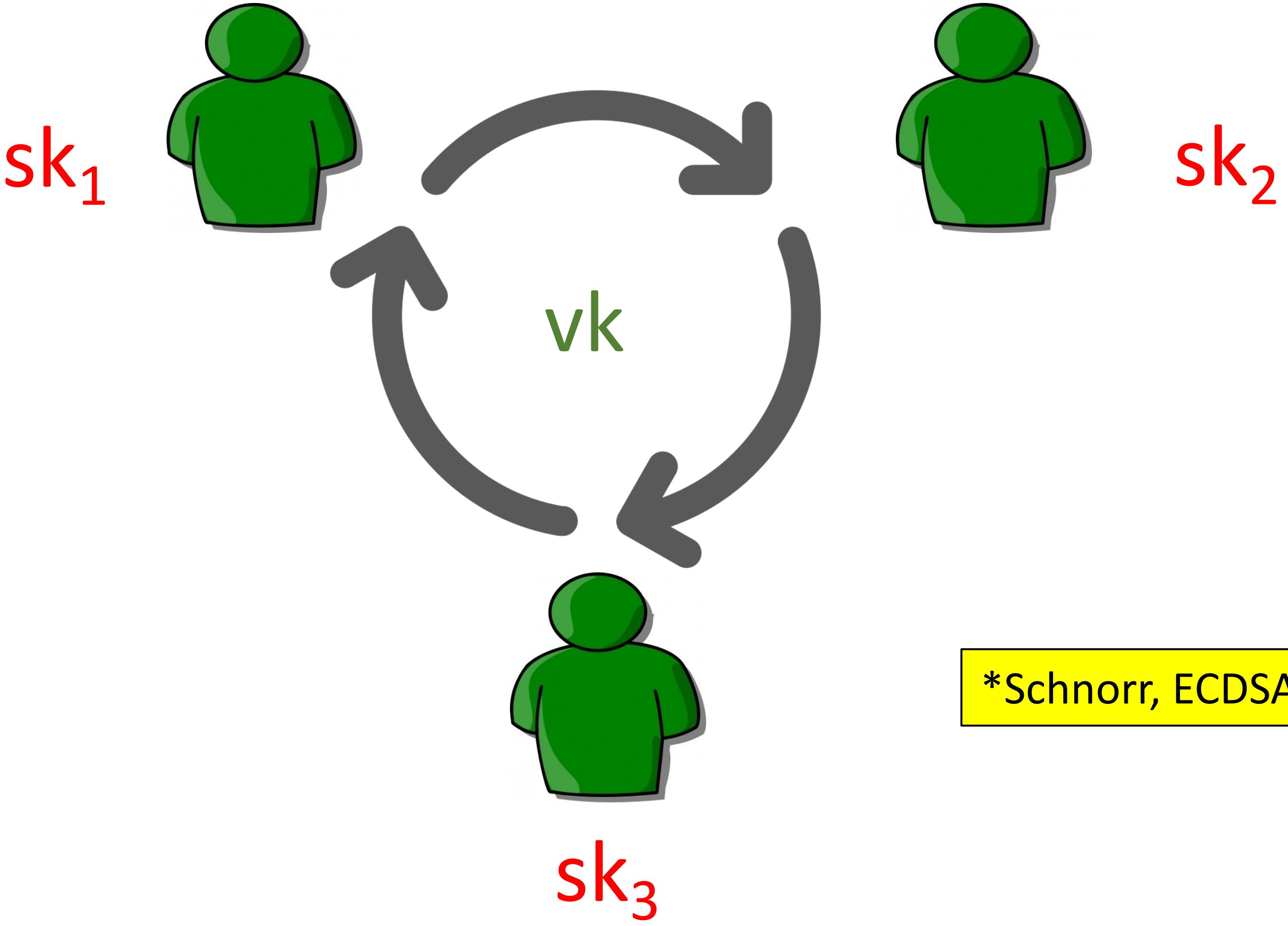
Interchangeability – same verification

Verify(vk, m, σ) → 1/0

- RET (e(H(m),vk) = e (σ, g_2))

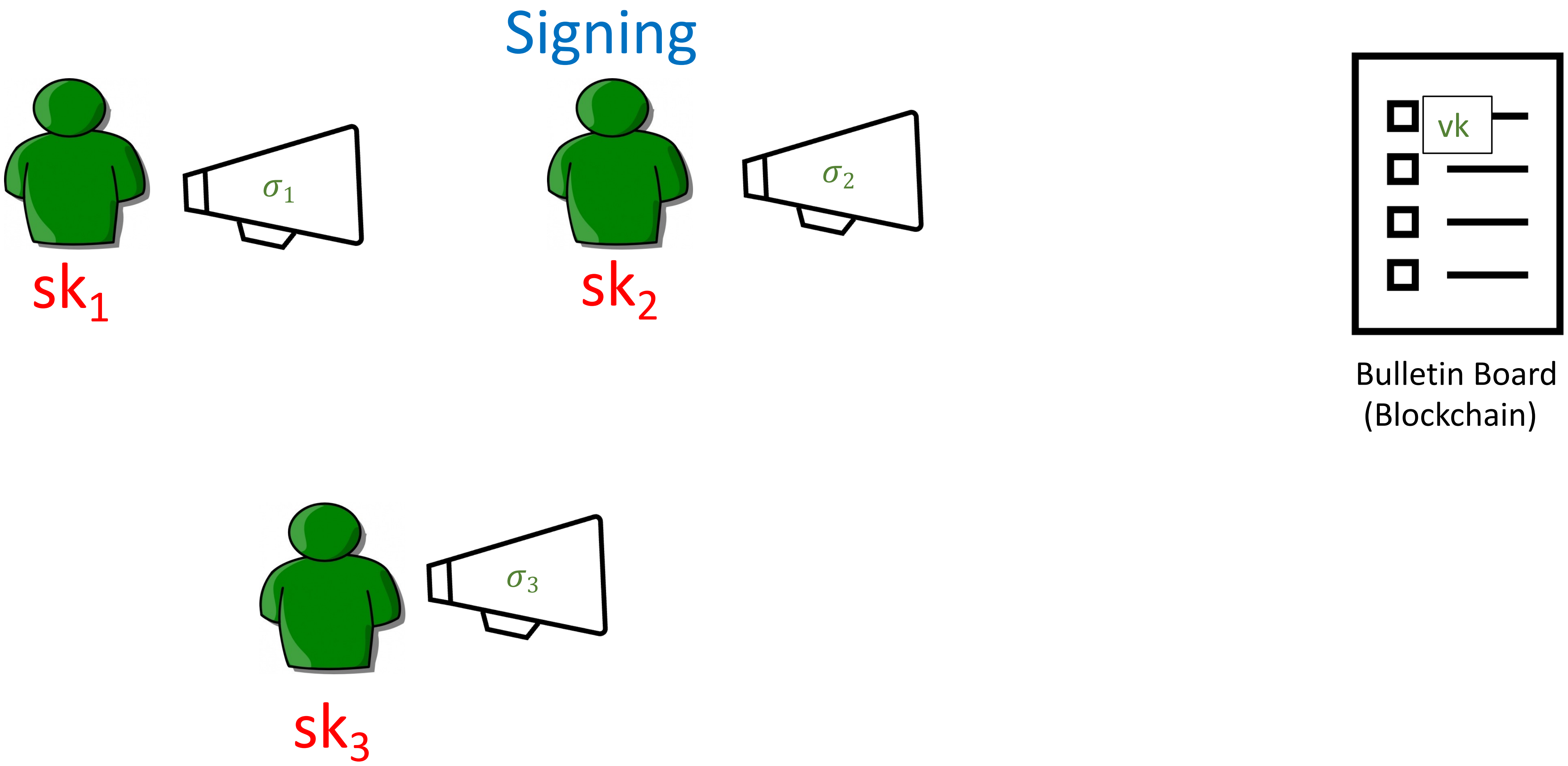
Threshold BLS: Non-interactive workflow

Dist-Kgen: Interactive, but reusable*

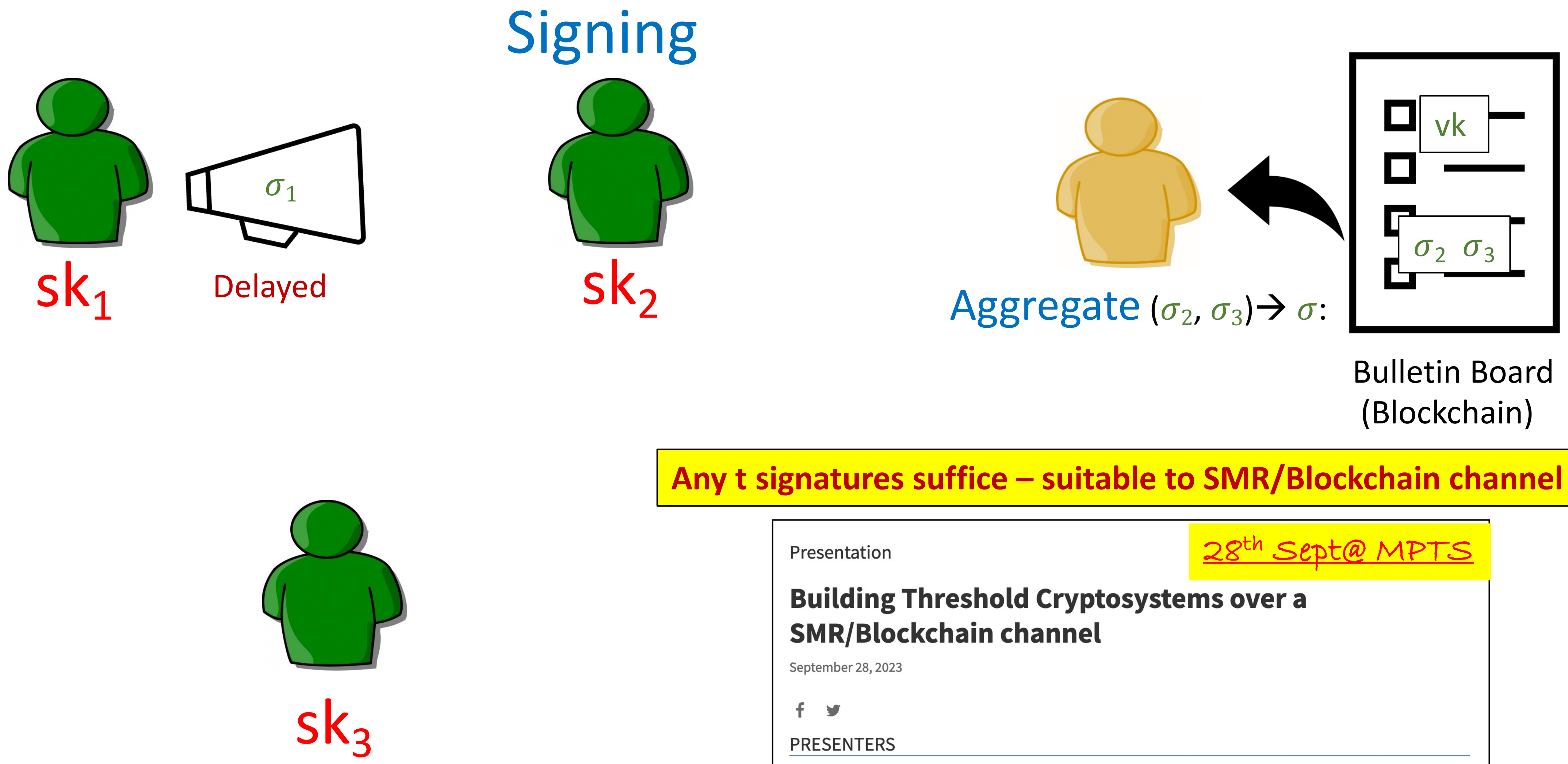


*Schnorr, ECDSA's pre-processing NOT reausable

Threshold BLS: Non-interactive workflow



Threshold BLS: Non-interactive workflow



Any t signatures suffice – suitable to SMR/Blockchain channel

Presentation

28th Sept@ MPTS

Building Threshold Cryptosystems over a SMR/Blockchain channel

September 28, 2023

PRESENTERS

Aniket Kate - Purdue University / Supra Research

Performance

- Signing: 1 exp per singer
- Aggregation: 1 t-multi-exp over G_1: $O(t/\log(t))$
- Verification: 2 pairing
- Signature size: 1 G_1

SCAN ME

Ethereum BLS implementation numbers

More features: (Distributed) VRF from BLS

Signature

Sign(sk,m)→ (σ):
- $\sigma := H(m)^{sk}$

Verify(vk, m, σ)→ 1/0
- RET (e(H(m),vk) = e (σ, g_2))

→

VRF

EVAL(sk,x)→ (y, π):
- $\pi := H(m)^{sk}$
- y = H'(π)

Verify(vk, x, y, π)→ 1/0
- RET (e(H(m),vk) = e (π, g_2))
  AND
  (H'(π) = y)

Readily distributed using key-homomorphism

Even more…

- New results:
    - Efficient Weighted (in fact, general access structure) Threshold Signature without DKG (compatibility with SNARK + Key-hom) [GJMSWZ'24, DCXNBR'23]:
    - Multiverse Threshold Signatures [BGJMSWZ'23] (Key-hom)
    - Adaptive security in AGM [BL'22]

High-level comparison w ECDSA and Schnorr

+

- Verifiably deterministic: only BLS
- Fully Non-interactive: only BLS
- Most threshold/multi-sig/aggregation friendly: BLS
- Most succinct: BLS
  - 2x smaller
- Signing time: Similar (?)

−

- Assumption:
  - BLS – pairing; ECDSA – heuristic; Schnorr – Dlog
- Verification time:
  - BLS about 5x costlier

Adaptation (in blockchain)

Ethereum 2.0 Validation (Multi-sig)

Dfinity Chain-key

Algorand Validation

Hashgraph State-proof

SUPRA

drand

DVRF based on Threshold BLS

Standardization of (non-threshold) BLS

The BLS Standard Draft has been Submitted to the IETF

By: Sergey Gorbunov

The BLS signature scheme was introduced by Boneh-Lynn-Shacham in 2001. The signature scheme relies on pairing-friendly curves and supports non-interactive aggregation properties. That is, given a collection of signatures (sigma_1, …, sigma_n), anyone can produce a short signature (sigma) that authenticates the entire collection. BLS signature scheme is simple, efficient and can be used in a variety of network protocols and systems to compress signatures or certificate chains.

CFRG
Internet-Draft
Expires: August 12, 2019

D. Boneh
Stanford University
S. Gorbunov
Algorand and University of Waterloo
H. Wee
Algorand and ENS, Paris
Z. Zhang
Algorand

February 8, 2019

BLS Signature Scheme
draft-boneh-bls-signature-00

Abstract

The BLS signature scheme was introduced by Boneh-Lynn-Shacham in 2001. The signature scheme relies on pairing-friendly curves and supports non-interactive aggregation properties. That is, given a collection of signatures (sigma_1, ..., sigma_n), anyone can produce a short signature (sigma) that authenticates the entire collection. BLS signature scheme is simple, efficient and can be used in a variety of network protocols and systems to compress signatures or certificate chains. This document specifies the BLS signature and the aggregation algorithms.

Summary

Threshold BLS is great! (if ok with bilinear pairing)

- Simple, non-interactive, deterministic, aggregatable….

- Active area of research:
  - Adaptive security (currently only in AGM + OMDL)
  - More efficient robustness
  - More efficient verification
  - More efficient weighted signatures

A Great Match

– C2.1, for signing (e.g., verifiably-deterministic succinct signatures, ~~and/or TF-QR~~);

Thank You!