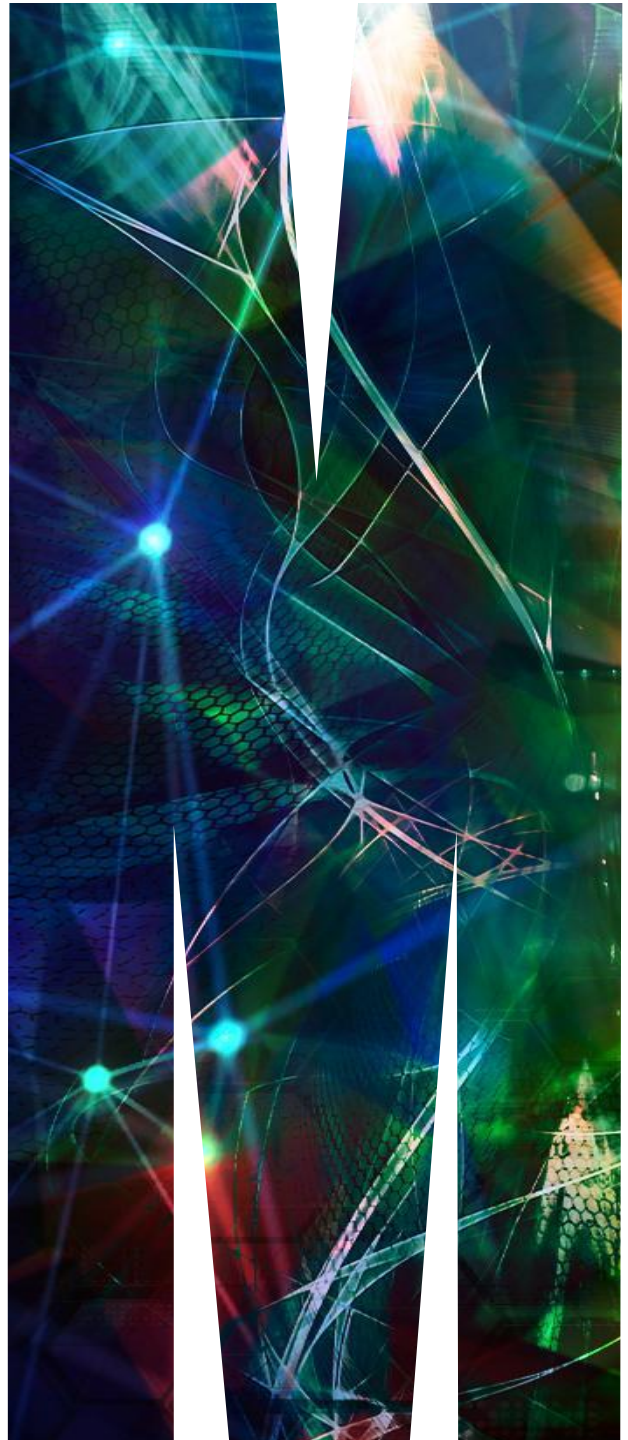
MONASH University

Post-Quantum Signatures from Symmetric Key Primitives



2024.07.03

Xinyu Zhang

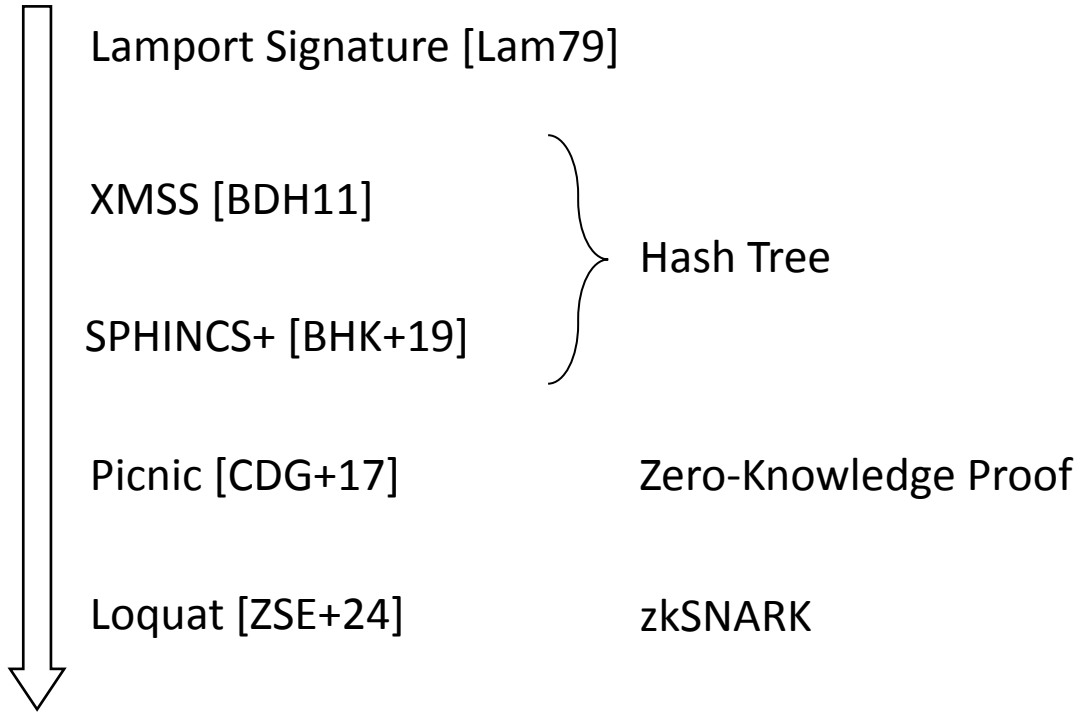
Department of Software System and Cybersecurity

Faculty of Information Technology

Monash University, Australia

Agenda

- Post-Quantum Signatures from Symmetric Key Primitives



OTS: Lamport Signature [Lam79]

- **One-Time Signature**: Each key pair can only be used to produce ***ONE*** signature.
  - Example: Lamport Signature, WOTS [Hül13]

Setup

Let λ be security parameter, and $F: K \to V$ be any one-way function

KeyGen

For $i \in [\lambda]$, j $\in \{0,1\}$ choose $k_{i,j} \in K$ and compute $v_{i,j} = F(k_{i,j})$. Output sk consists of 2λ values of $k_{i,j}$ and pk consists of 2λ values of $v_{i,j}$.

Sign

Let $m = (m_1, \dots, m_\lambda) \in \{0,1\}^\lambda$. For $i \in [\lambda]$:

- If $m_i = 0$, then $\sigma_i = k_{i,0}$.
- If $m_i = 1$, then $\sigma_i = k_{i,1}$.
- Output $\sigma = (\sigma_1, \dots, \sigma_\lambda)$.

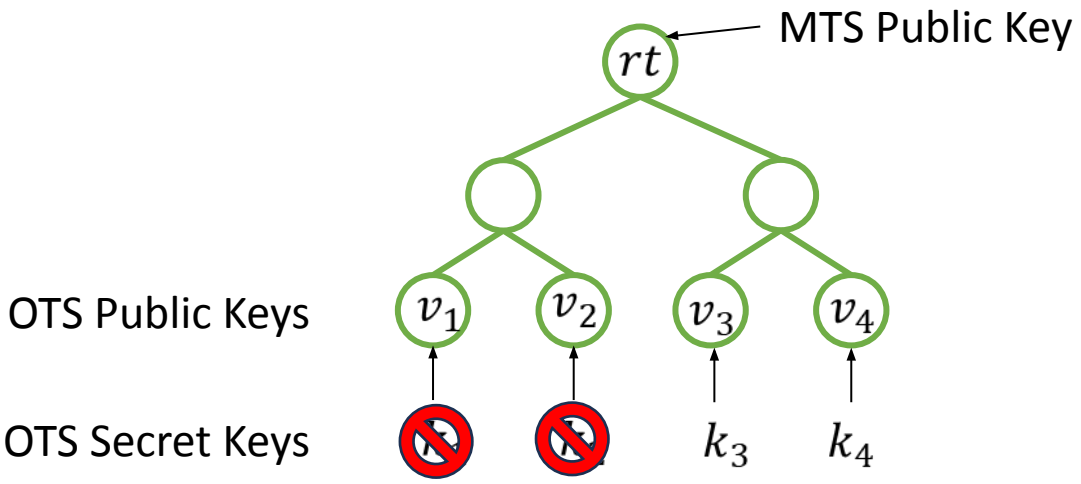
Verify

Let $m = (m_1, \dots, m_\lambda) \in \{0,1\}^\lambda$. For $i \in [\lambda]$:

- If $m_i = 0$, check $F(\sigma_i) = v_{i,0}$.
- If $m_i = 1$, check $F(\sigma_i) = v_{i,1}$.

MTS: XMSS [BDH11]

- **(Stateful) Many-Time Signature**: Each key pair can be used to sign many messages, but the signer must make sure that the secret key is not used twice.
  - Example: XMSS [BDH11]
- High-level Idea: Combine OTS with a Merkle tree



Signing Key: (k_1, k_2, k_3, k_4)

Verification Key: rt

Message: m_1

one-time signature generated with k_1

Signature: $(\sigma_1, \text{auth\_path}(v_1))$

Message: m_2

one-time signature generated with k_2

Signature: $\big(\sigma_2, \text{auth\_path}(v_2)\big)$

MTS: SPHINCS+ [BHK+19]

- **(Stateless) Many-Time Signature**: Each key pair can be used to sign any message as a normal signature scheme.
  - Example: SPHINCS and SPHINCS+ [BHK+19]
- **High-level Idea:**
  - Replace leaf OTS with FTS (Few Time Signature)
  - Use hyper trees (trees of Merkle tree) that are connected by OTS
  - See details in next slide…

MTS: SPHINCS+ [BHK+19]

FORS

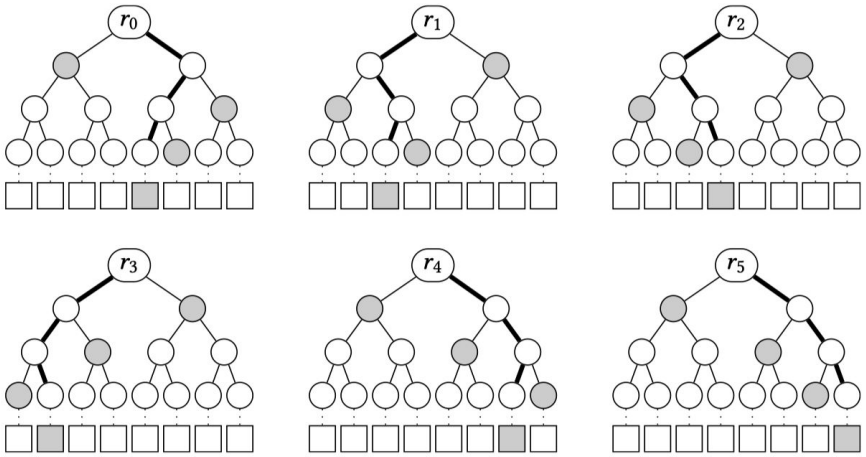


Figure 3: An illustration of a FORS signature with k = 6 and a = 3, for the message 100 010 011 001 110 111.

- FORS has k trees, each with height a
- Break n bits message to k blocks, each block contains a bits.
- Each block $i \in [k]$ defines the index of a tree
- Each a-bits string defines a unique leaf in tree i

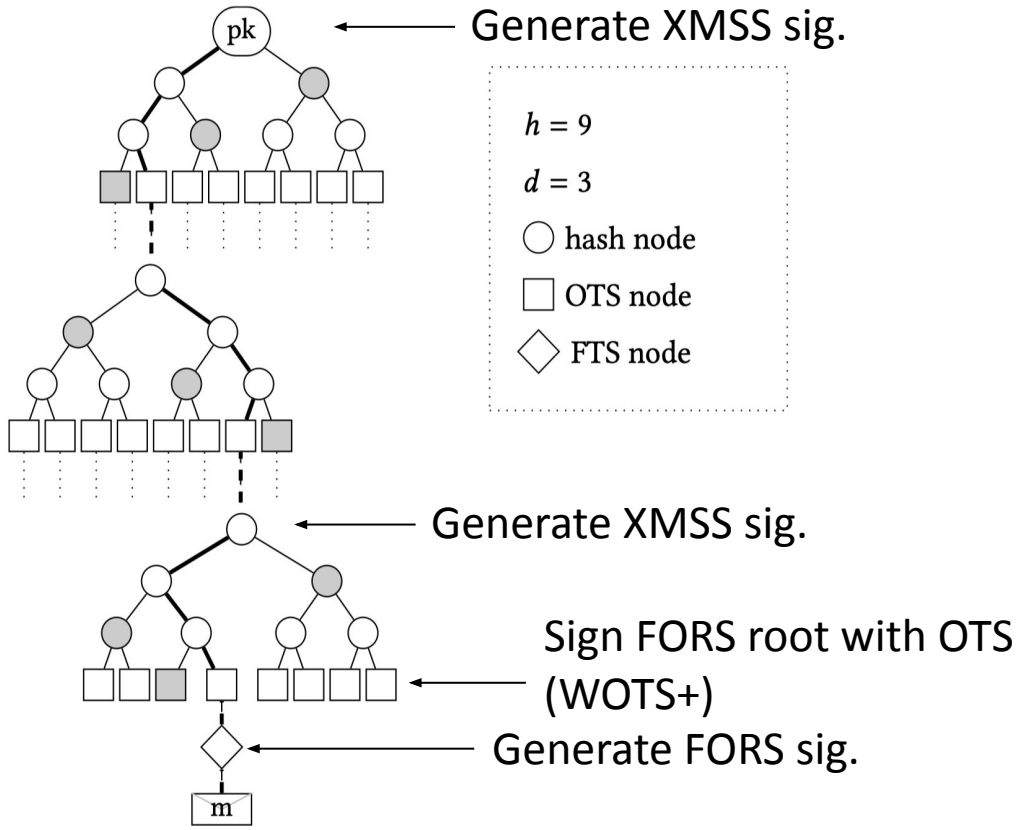


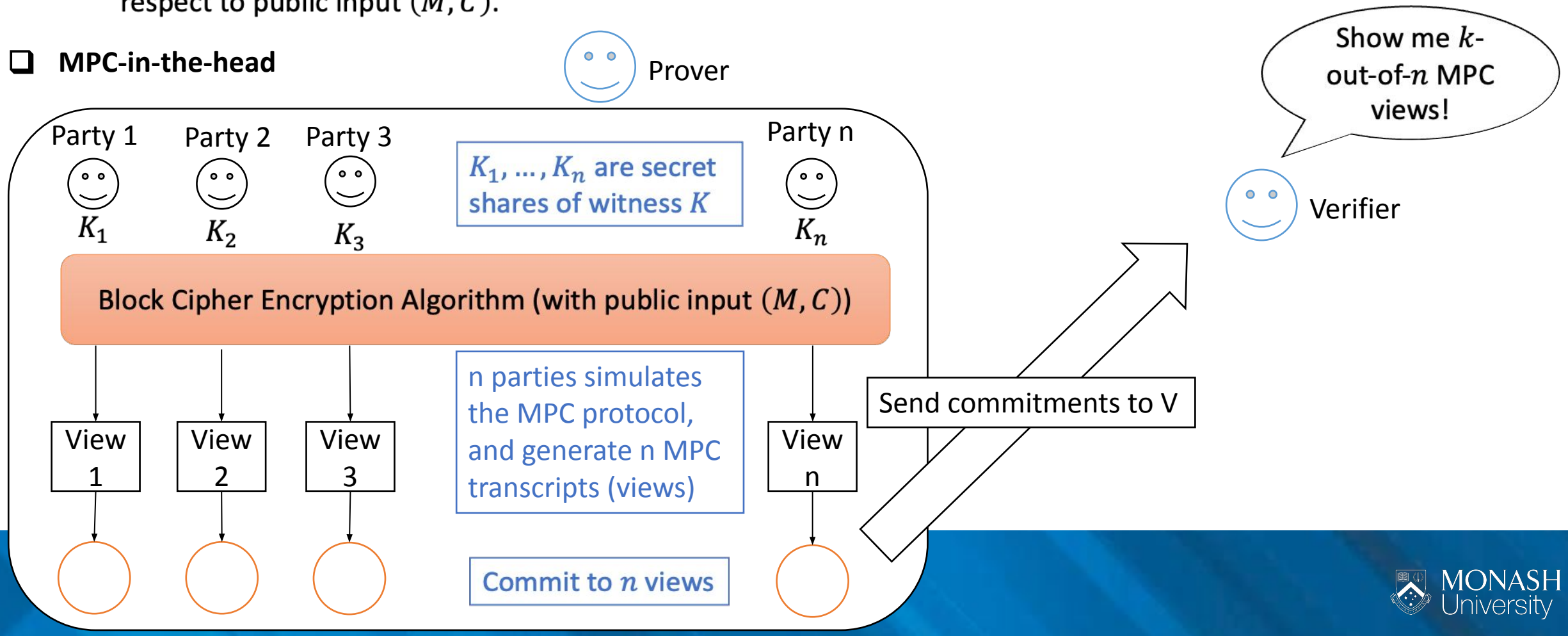
Figure 1: An illustration of a (small) SPHINCS structure.

Images from: [BHK+19]

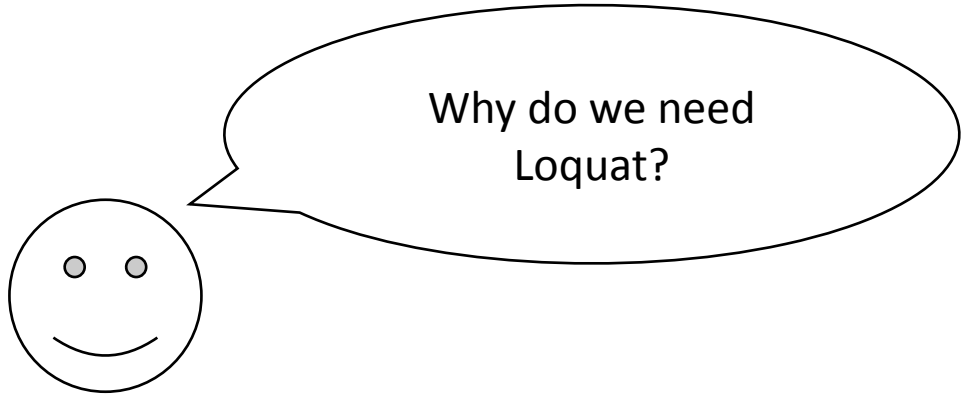
ZKP: Picnic [CDG+17]

- **High-level Idea:**
  - Given a block cipher encryption algorithm $\mathrm{Enc}(K, M) \rightarrow C$, where K is the key, M is the message and C is the ciphertext, using a **generic zero-knowledge proof** (MPC-in-the-head [IKOS07]) to prove the knowledge of K, with respect to public input (M, C).

- **MPC-in-the-head**



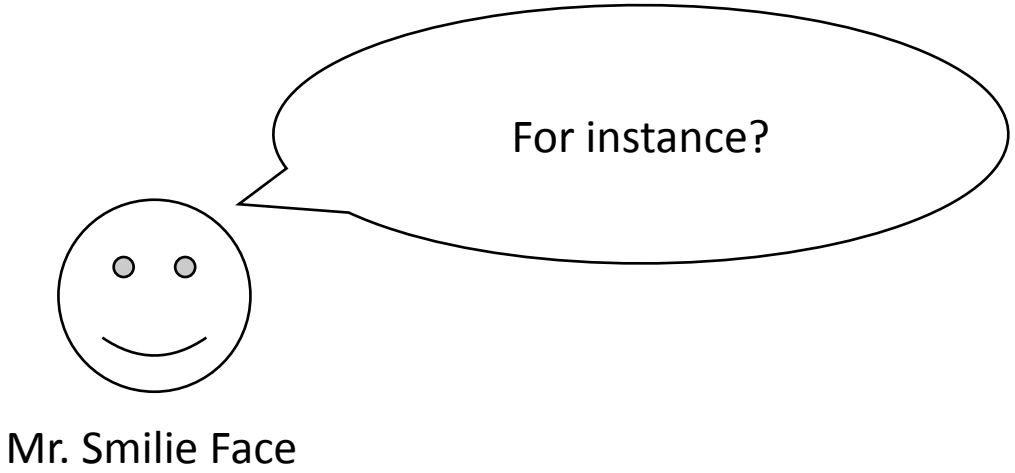
zkSNARK: Loquat [ZSE+24]

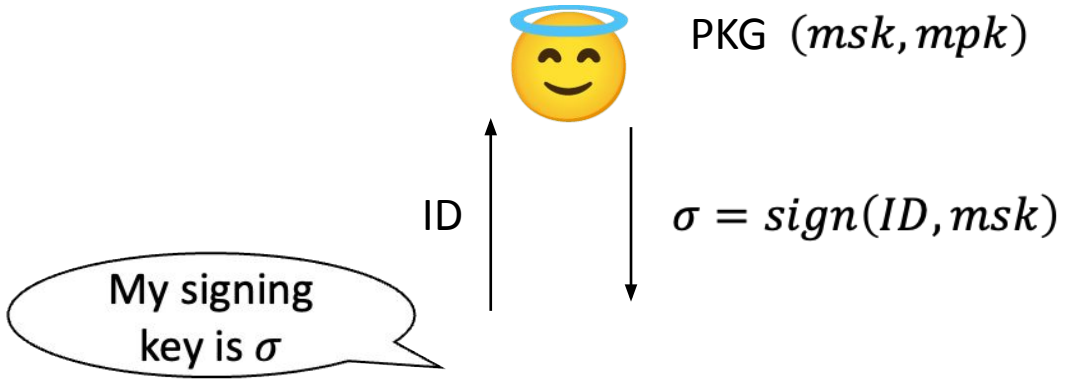


Mr. Smilie Face

In some scenario, we need to prove the validity of signatures using a zero-knowledge proof.

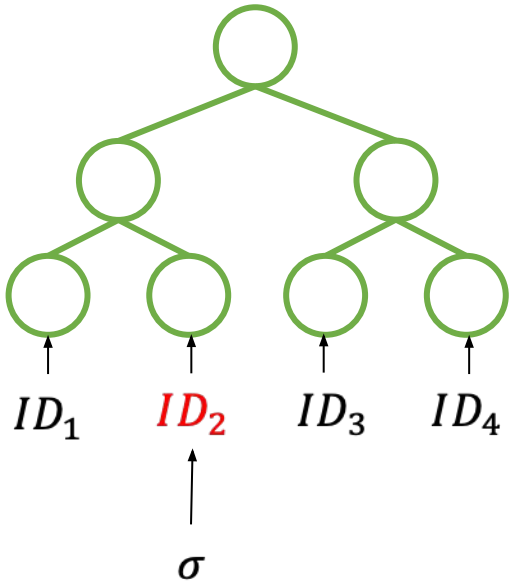
zkSNARK: Loquat [ZSE+24]





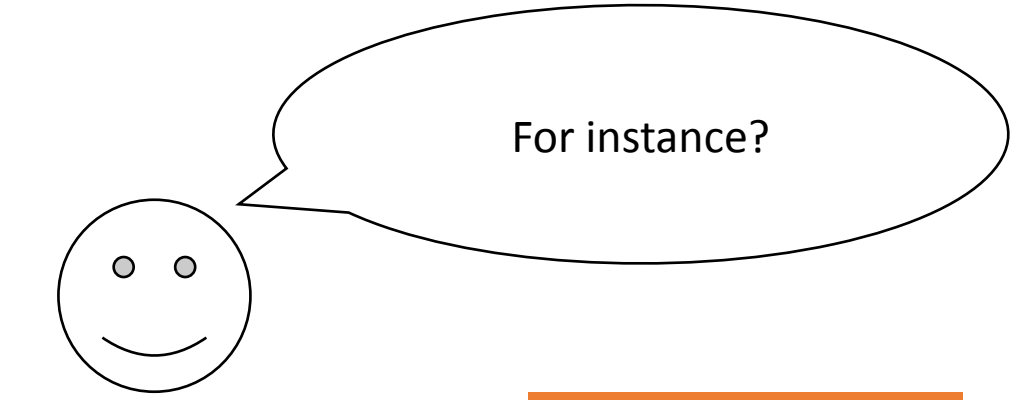
- More efficient **post-quantum ring signature** and aggregate signature from symmetric key primitives!

Ring signature is the zero-knowledge proof which shows that $\exists\ ID_?$ such that $Verify(\sigma, ID_?, mpk) = 1$

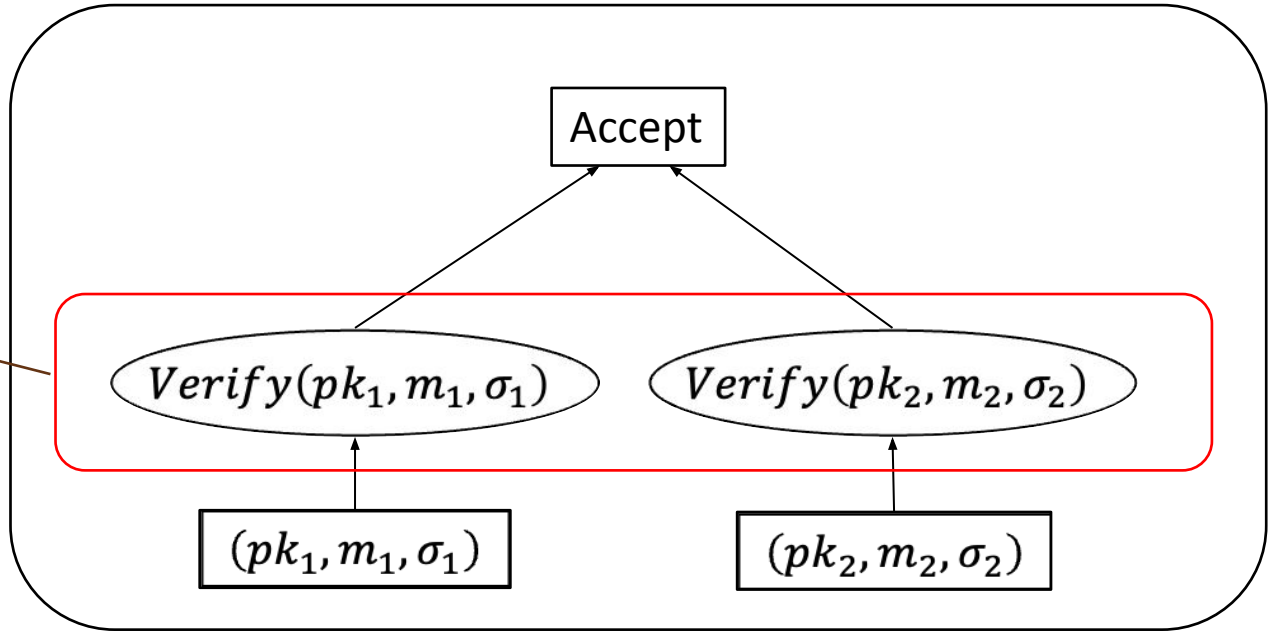


zkSNARK: Loquat [ZSE+24]

- More efficient **post-quantum** ring signature and **aggregate signature** from symmetric key primitives!



Translate to a circuit, then prove the circuit output is 1 using a generic SNARK.

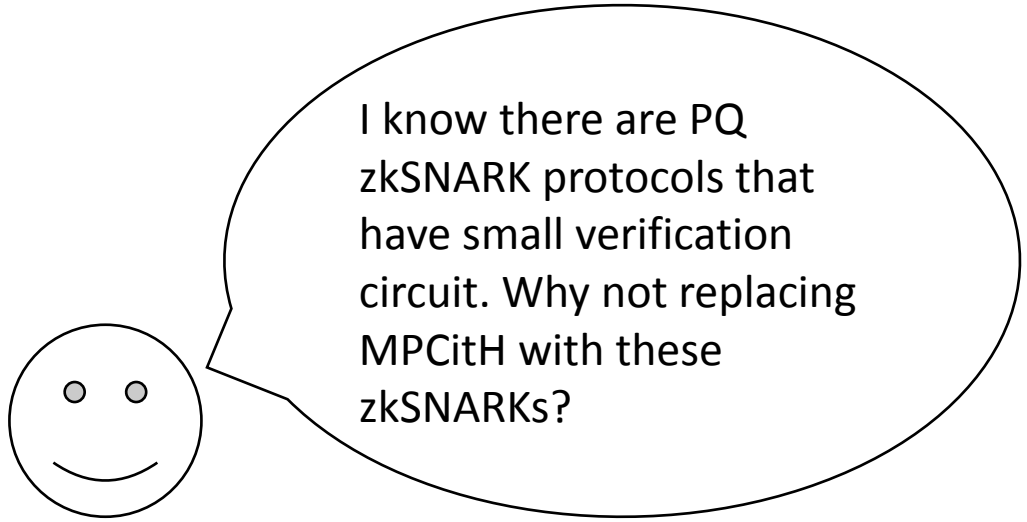


zkSNARK: Loquat [ZSE+24]

Mr. Smilie Face

- OTS and stateful MTS have limited use cases.
- MPCitH-based signatures have extremely large verification circuits.
- SPHINCS+ has a lot of hash functions in KeyGen and Sign.

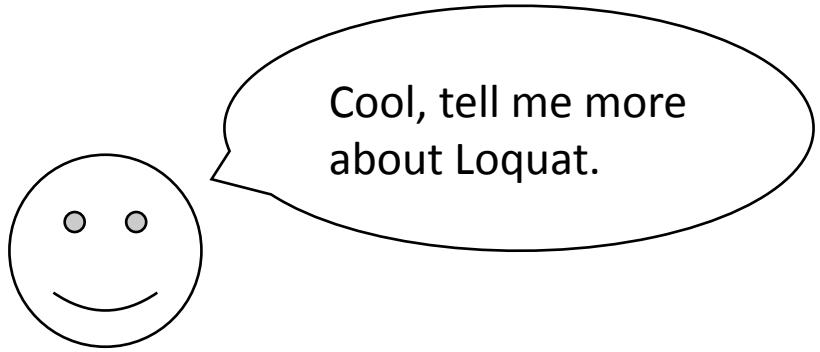
zkSNARK: Loquat [ZSE+24]



Mr. Smilie Face

- Generic PQ zk-SNARKs have very large proof size in comparison with MPCitH-based proofs.
- Generic zkSNARKs are quite slow as well.
- And their verification circuits are not small enough.

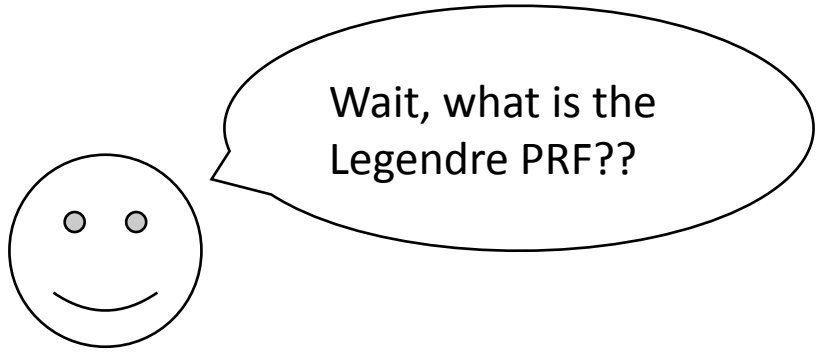
zkSNARK: Loquat [ZSE+24]



Mr. Smilie Face

- Loquat is a signature which let the signer prove its knowledge about the witness of the ***Legendre PRF***.
- A balanced choice between signature performance and SNARK friendliness

zkSNARK: Loquat [ZSE+24]



Mr. Smilie Face

- Legendre PRF is the PRF based on the pseudo-randomness of the Legendre symbol.

Legendre Symbol: $$\left(\frac{a}{p}\right) = \begin{cases} 1, & \text{if } a = b^2 \pmod{p} \text{ for some } b \in \mathbb{F}_p^* \\ 0, & \text{if } a = 0 \pmod{p} \\ -1, & \text{otherwise} \end{cases}$$

Legendre PRF: $$\mathcal{L}_0(a) = \left\lfloor \frac{1}{2}\left(1 - \left(\frac{a}{p}\right)\right) \right\rfloor \in \{0, 1\}$$

Keyed Legendre PRF: $$\mathcal{L} : \mathbb{F}_p \times \mathbb{F}_p \to \mathbb{Z}_2$$
$$\mathcal{L} : (K, a) \mapsto \mathcal{L}_0(K + a)$$

Legendre PRF based key pair:

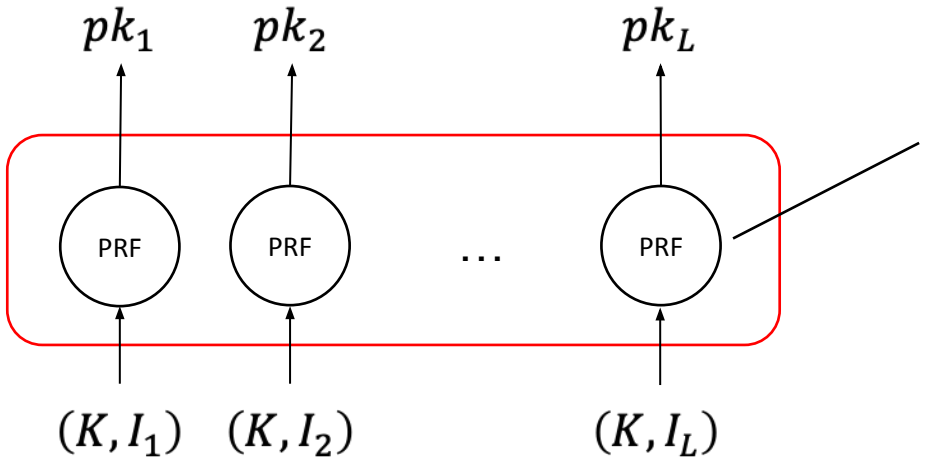
- $pk = (pk_1, \ldots, pk_L) = \mathcal{L}_K(I_1), \ldots, \mathcal{L}_K(I_L)$
- $sk = K$

zkSNARK: Loquat [ZSE+24]

Mr. Smilie Face

Naively

…



Prove using ZKP

zkSNARK: Loquat [ZSE+24]

Commit to $(K, r_1, \dots, r_B)$

$B \ll L$

$\mathcal{L}_0(o_b)$
$= \mathcal{L}_0((K + I_b) \times r_b)$
$= \mathcal{L}_0(K + I_b) + \mathcal{L}_0(r_b)$
$= pk_{I_b} + \mathcal{L}_0(r_b)$

Send the commitment and $\mathcal{L}_0(r_b)$ for all $b \in [B]$ to V

Randomly pick B elements from $(I_1, \dots, I_L)$ and send to P

Compute and send $o_b = (K + I_b)r_b$ for all $b \in [B]$

Mr. Smilie Face

Randomly pick B field elements $(\lambda_1, \dots, \lambda_B)$and send to P

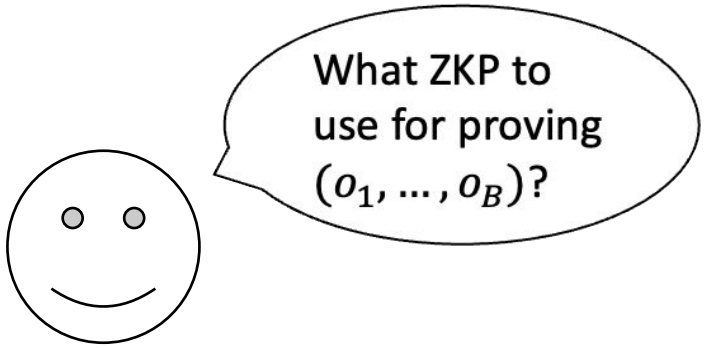
- Relax the computation to a random subset of symbols!
- exploit the multiplicative homomorphism of the PRF:

$$\mathcal{L}_0(a) + \mathcal{L}_0(b) = \mathcal{L}_0(a \times b)$$

- Prove the random linear computation instead of individual symbols

Prove $\sum_{b \in [B]} \lambda_b o_b = \lambda_b (K + I_b) r_b$ using ZKP

zkSNARK: Loquat [ZSE+24]

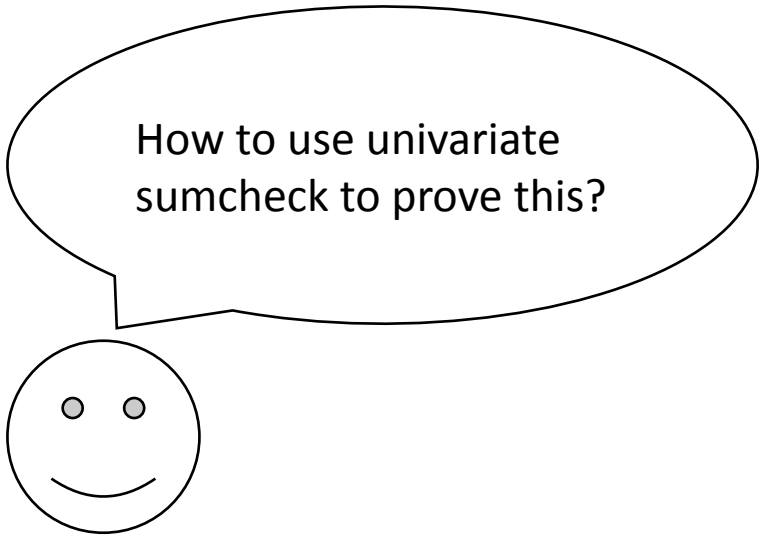


Mr. Smilie Face

Instead of MPCitH, we use univariate sumcheck to enable SNARK-friendliness!

- Univariate sumcheck is the core protocol used in Aurora [BCR+19].
- Let H be an additive or multiplicative coset of $\mathbb{F}$, given a polynomial $f(x) \in \mathbb{F}(X)$, it allows the prover to prove that $\mu = \sum_{a \in H} f(a)$.
- Verification complexity is polylog in the size of H.

zkSNARK: Loquat [ZSE+24]



Mr. Smilie Face

Our key observation is that proving $\sum_{b\in[B]} \boldsymbol{\lambda_b o_b} = \boldsymbol{\lambda_b}\,(\boldsymbol{K} + \boldsymbol{I_b})\boldsymbol{r_b}$ can be seen as evaluating a multivariate polynomial!

Define $f(x_1, \ldots, x_B, y_1, \ldots, y_B) = \sum_{b\in[B]} K r_b y_b + r_b x_b\, y_b$.

Commit to coefficients of $f(x_1, \ldots, x_B, y_1, \ldots, y_B)$ and send the commitment along with $\mathrm{T_b} = \mathcal{L}_0(r_b)$ for all $b \in [B]$ to V.

Randomly pick B elements from $(I_1, \ldots, I_L)$ and send to P

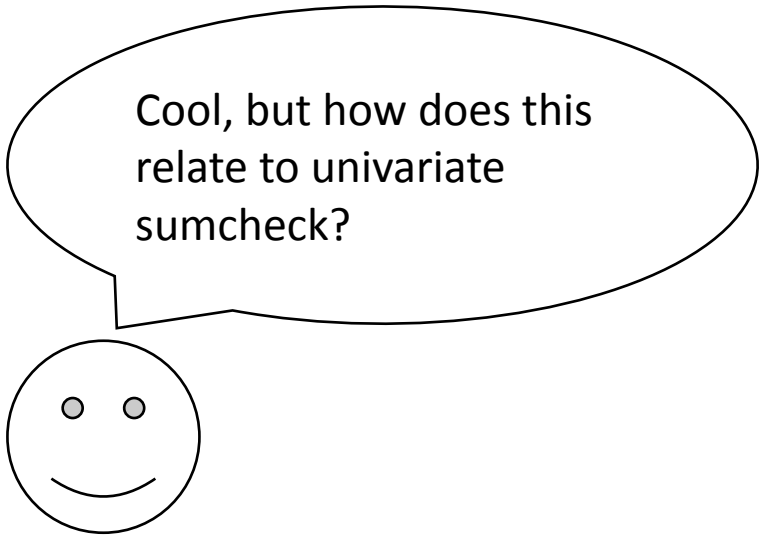
Compute and send $o_b = (K + I_b)r_b$ for all $b \in [B]$

Randomly pick B field elements $(\lambda_1, \ldots, \lambda_B)$and send to P

Send to V the polynomial evaluation $\mu = f(I_1, \ldots, I_B, \lambda_1, \ldots, \lambda_B)$ and the proof of the evaluation.

Check if $\mu = \sum_{b\in[B]} \lambda_b o_b$ and the polynomial commitment proof.

zkSNARK: Loquat [ZSE+24]

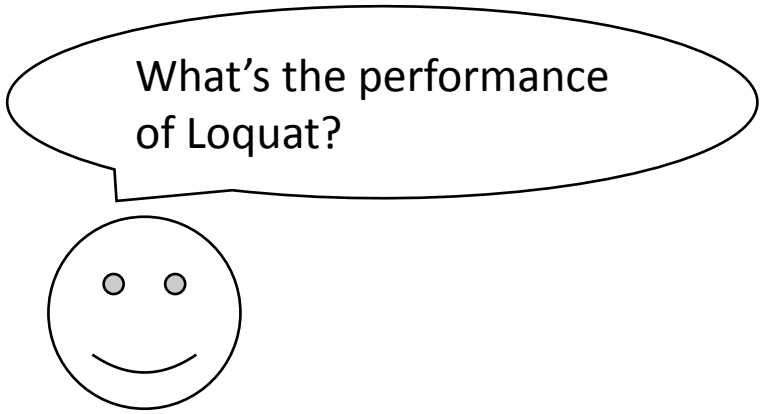


Mr. Smilie Face

Virgo [ZXZS20] shows how to use univariate sumcheck to construct polynomial commitment!

- $f(I_1, \dots, I_B, \lambda_1, \dots, \lambda_B) = \sum_{b\in[B]} Kr_b\lambda_b + r_b I_b\, \lambda_b$ can be seen as an inner-product between two vectors:
  - Coefficient vector: $\vec{c} = (c_1, \dots, c_{2B}) = (Kr_1, \dots, Kr_B, r_1, \dots, r_B)$;
  - Evaluation vector: $\vec{t} = (t_1, \dots, t_{2B}) = (\lambda_1, \dots, \lambda_B,\ \lambda_1 I_1, \dots, \lambda_B I_B,)$.
- We can interpolate two univariate polynomials from these two xsvectors:
  - Let us define a univariate polynomial $c(x)$ such that $c(h_i) = c_i$ for some predefined set H where $h_i \in H$ and $c_i \in \vec{c}$.
  - Similarly, define a univariate polynomial $t(x)$ such that $t(h_i) = t_i$ for predefined set H where $h_i \in H$ and $t_i \in \vec{t}$.
- Then, define a polynomial for univariate sumcheck $g(x) = c(x) \times t(x)$. Let the prover to show that $\mu = \sum_{b\in[B]} \lambda_b o_b = \sum_{h_i\in H} g(h_i) = \sum_{h_i\in H} c(h_i) \times t(h_i) = \sum_{i\in[2B]} c_i \times t_i$.

zkSNARK: Loquat [ZSE+24]

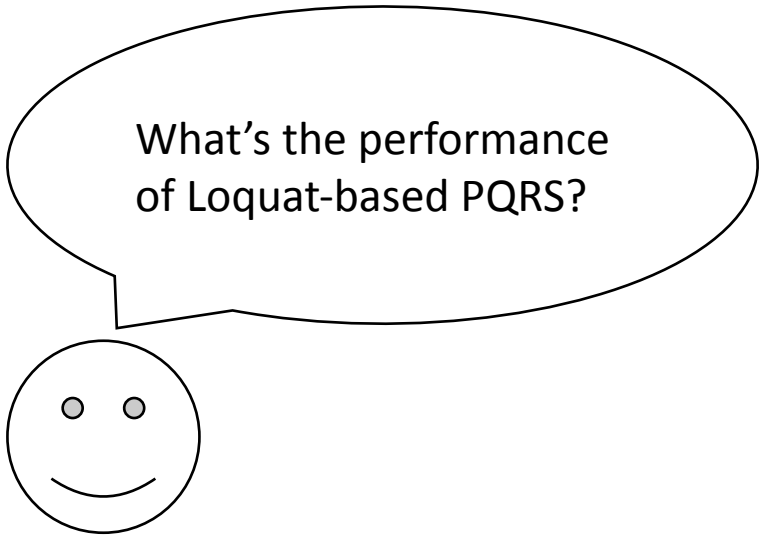


Mr. Smilie Face

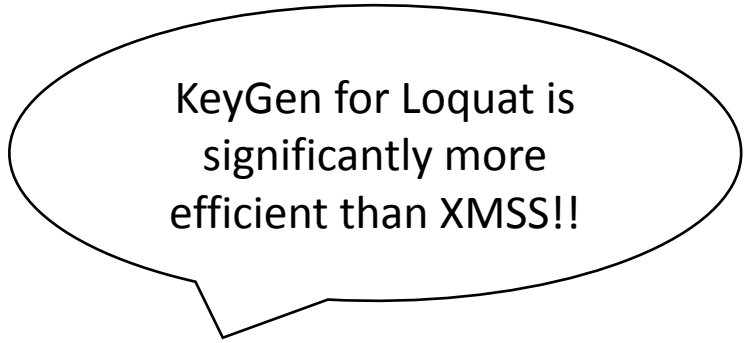
Scheme	$\|\sigma\|$ (KB)	# R1CS	# Hash (KG)	# Hash (Sign)
Picnic1 [25]	32	3,451,440	0	7,008
Picnic3 [64]	12	21,598,340	0	1,679,784
LegRoast [16]	16	1,143,351	0	10,152
Banquet [6]	12	11,804,560	0	4,147
Rainer [30]	8	26,130,303	0	3,337
SPHINCS+s [14]	7.8	458,920	305,663	2,397,129
SPHINCS+f [14]	16	1,390,209	4,775	111,353
LOQUAT-128	57	148,825	0	8,434

Table 1. Comparison of Post-Quantum Signature Schemes based on Symmetric-Key Primitives. $|\sigma|$ = signature size, # R1CS = number of R1CS constraints for verifying one signature, # Hash (KG)/# Hash (Sign) = the number of hash invocations in key generation/signing, respectively.

PQRS and PQAS based on Loquat



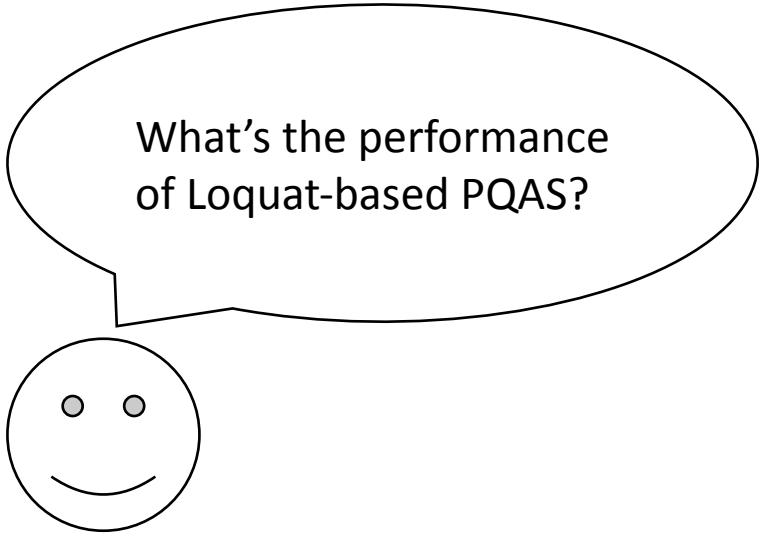
Mr. Smilie Face



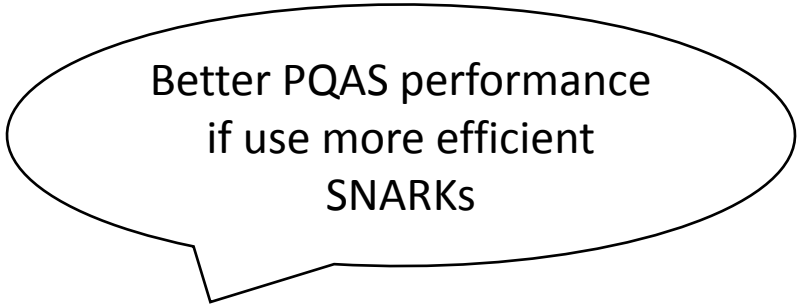
Sig	t_{KG}(s)	$\|SID\|$(KB)	R1CS	Max N	$\|\sigma\|$ (MB) (est.)		
					$N = 2^6$	$N = 2^{12}$	$N = 2^{20}$
Picnic [64]	≈ 0	167	6.88M	unli.	1.898	1.899	1.900
XMSS [22]	3355	4.899	431,480	2^{20}	0.885	0.889	0.893
Loquat-100	0.1	46	648,549	unli.	0.973	0.977	0.982

Table 4. Instantiate IDRS [23] with Various Signatures. t_{KG}= Master Key Generation Time, $|SID|$ = User Signing Key Size, N = ring size, $|\sigma|$ = ID-based Ring Signature Size.

PQRS and PQAS based on Loquat



Mr. Smilie Face



	Aurora					Fractal			Fractal-R			(est.)
	$n = 4$	8	16	32	64	4	8	16	4	8	16	32
t_I (s)	-	-	-	-	-	33	88	218	5K	13K	32K	32K
t_P (s)	14	33	67	160	553	119	251	556	3K	6K	14K	14K
t_V (s)	2.4	4.3	10	20	62	0.2	0.47	0.78	2.4	5.64	9.6	9.6
$\|\sigma\|$ (KB)	126	137	182	197	207	131	140	145	131	140	145	145

Table 5. Performance of Aggregating LOQUAT Using Aurora, Fractal, and Recursive Fractal (Fractal-R). t_I = indexer time, t_P = proving time, t_V = verification time, n = the number of signatures being aggregated.

Conclusion

- Post-quantum signatures based on symmetric key primitives.
- Loquat: SNARK-friendly post-quantum signature from the Legendre PRF
  - More efficient ID-based ring signature.
  - Post-quantum aggregate signature from symmetric key primitives.

THANK YOU!

MONASH University

References

[Lam79] Lamport, Leslie. "Constructing digital signatures from a one way function." (1979).

[BDH11] Buchmann, Johannes, Erik Dahmen, and Andreas Hülsing. "XMSS-a practical forward secure signature scheme based on minimal security assumptions." In *Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011*, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4, pp. 117-129. Springer Berlin Heidelberg, 2011.

[BHK+19] Bernstein, Daniel J., Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. "The SPHINCS+ signature framework." In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pp. 2129-2146. 2019.

[CDG+17] Chase, Melissa, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. "Post-quantum zero-knowledge and signatures from symmetric-key primitives." In *Proceedings of the 2017 acm sigsac conference on computer and communications security*, pp. 1825-1842. 2017.

[Hül13] Hülsing, Andreas. "W-OTS+–shorter signatures for hash-based signature schemes." In Progress in Cryptology–AFRICACRYPT 2013: 6th International Conference on Cryptology in Africa, Cairo, Egypt, June 22-24, 2013. Proceedings 6, pp. 173-188. Springer Berlin Heidelberg, 2013.

References

[IKOS] Ishai, Yuval, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. "Zero-knowledge from secure multiparty computation." In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pp. 21-30. 2007.

[ZSE+24] Zhang, Xinyu, Ron Steinfeld, Muhammed F. Esgin, Joseph K. Liu, Dongxi Liu, and Sushmita Ruj. "Loquat: A SNARK-Friendly Post-Quantum Signature based on the Legendre PRF with Applications in Ring and Aggregate Signatures." *Cryptology ePrint Archive* (2024).

[BCR+19] Ben-Sasson, E., Chiesa, A., Riabzev, M., Spooner, N., Virza, M., Ward, N.P.: Aurora: Transparent succinct arguments for r1cs. In: *Eurocrypt*. pp. 103–128. Springer (2019)

[ZXZS20] Zhang, J., Xie, T., Zhang, Y., Song, D.: Transparent polynomial delegation and its applications to zero knowledge proof. In: *2020 IEEE S&P.* pp. 859–876. IEEE (2020)

[BLSS22] Buser, M., Liu, J.K., Steinfeld, R., Sakzad, A.: Post-quantum id-based ring signatures from symmetric-key primitives. In: ACNS. pp. 892–912. Springer (2022)