Shorter, Tighter, FAESTer

Optimizations and Improved (QROM) Analysis for VOLE-in-the-Head Signatures

Carsten Baum[1] Ward Beullens[2] Lennart Braun[3] Cyprien Delpech de Saint Guilhem[4]
Michael Klooß[5] Christian Majenz[1] Shibam Mukherjee[6,7] Emmanuela Orsini[8]
Sebastian Ramacher[9] Christian Rechberger[6] Lawrence Roy[10] Peter Scholl[10]

[1]Technical University of Denmark
[2]IBM Research Europe
[3]Université Paris Cité, CNRS, IRIF
[4]3MI Labs
[5]Karlsruhe Institute of Technology
[6]Graz University of Technology
[7]Know Center
[8]Bocconi University
[9]Austrian Institute of Technology
[10]Aarhus University

team@faest.info

FAEST Overview

$SK = (x, k)$ $PK = (x, y)$

$y = AES_k(x)$

π = ZK Proof of Knowledge of k.

FAEST Overview

$SK = (x, k)$ $PK = (x, y)$

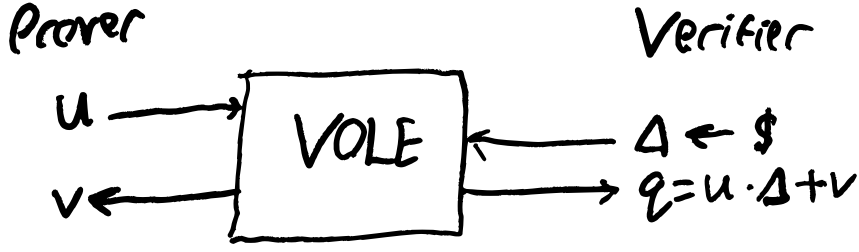
$y = AES_k(x)$

π = ZK proof of knowledge of k.

Why AES?

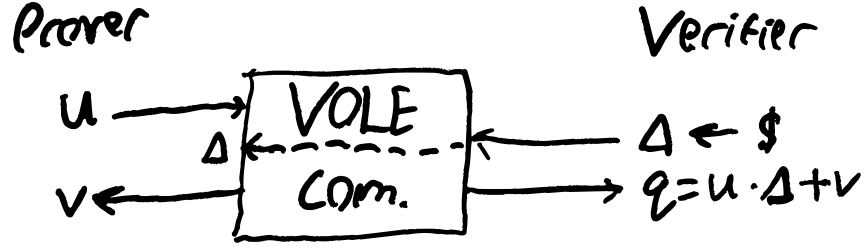
- withstood decades of analysis
- Algebraic SBOX: $x \to x^{-1}$
  (followed by a F_2-linear map)

VOLE

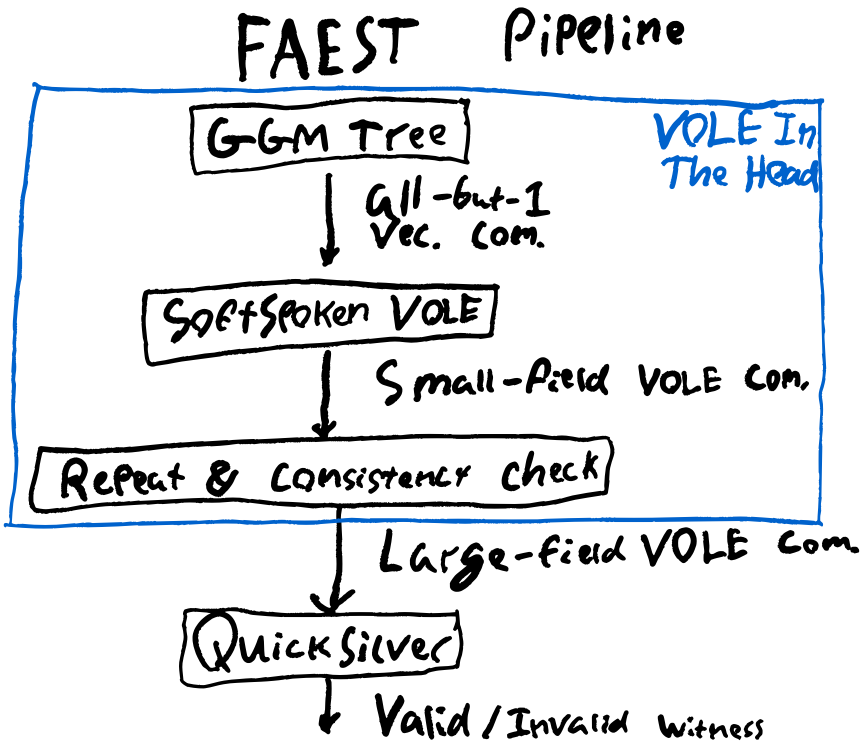


Lying about u requires guessing Δ. Used in VOLE-based ZKPs.

VOLE Commitments
(AKA VOLE-in-the-head)



Lying about u before Δ is chosen still requires guessing Δ. Sufficient for building public-coin VOLE-based ZKP.


FAEST Pipeline
GGM Tree
All-but-1 Vec. com.
SoftSpoken VOLE
Small-field VOLE com.
Repeat & consistency check
VOLE In The Head
Large-field VOLE com.
QuickSilver
Valid / Invalid witness

FAEST

+ Only uses symmetric key primitives

+ Smaller than SPHINCS+

+ Faster Pcover than SPHINCS+

+ Very small Public keys

FAEST

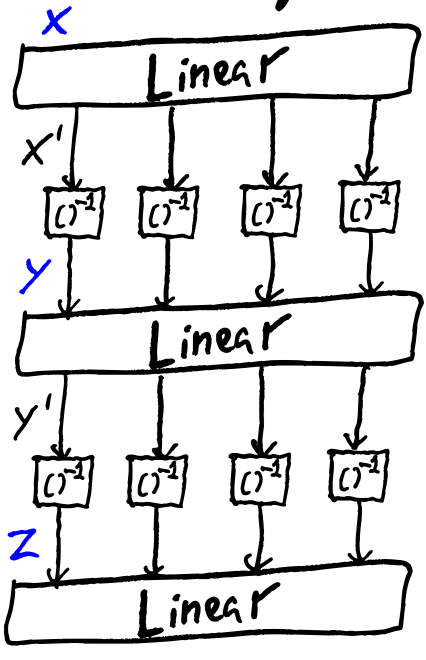
+ Only uses symmetric key primitives
+ Smaller than SPHINCS+
+ Faster keygen than SPHINCS+
+ Very small Public keys

- No complete proof in QROM.
- Bigger Signatures than Dilithium
- Much slower than Dilithium.

Improvements

- Reduce witness size by 25% by encoding degree 3 constraints.
- Replace Keccak with AES in GCM tree leaves, making this step 2.4 – 15x faster.
- Tight analysis in both ROM and QROM

Checking SNARKs



Witness:

$x \in F_{256}$

$y \in F_{256}$

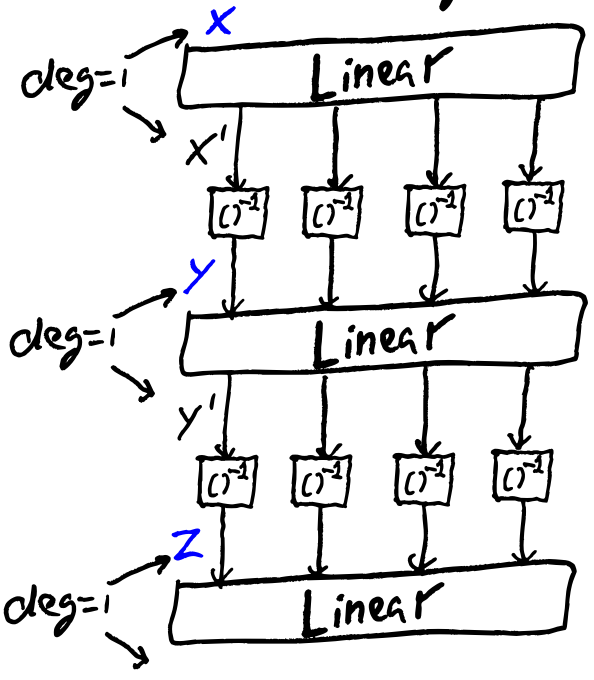
$z \in F_{256}$

Constraints:

$x' \cdot y = 1$

$y' \cdot z = 1$

Checking S-boxes



Witness:

$x \in F_{2^{96}}$

$y \in F_{2^{96}}$

$z \in F_{2^{96}}$

Constraints:

$x' \cdot y = 1$

$y' \cdot z = 1$

all deg = 2.

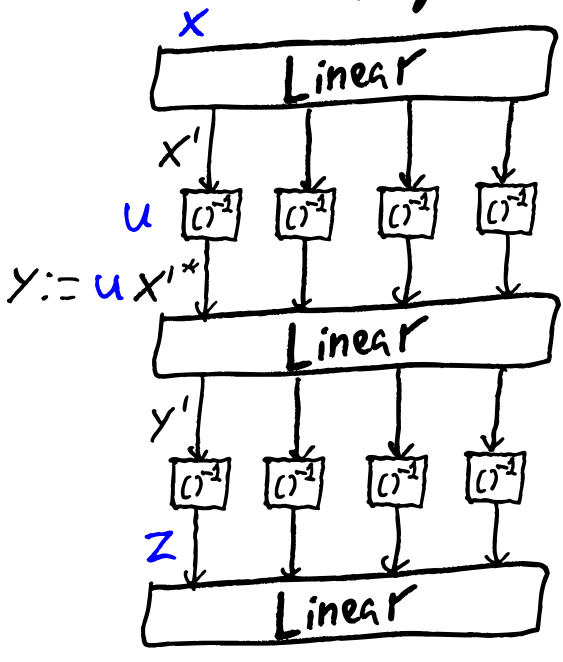
Norm trick:

For $x \in F_{256}$
Conjugation: $x \Rightarrow x^* = x^{16}$
(F_2-linear)

$$xx^* = x^{17} \in F_{16}$$

$\Rightarrow$ to compute x^{-1},
find $u = (xx^*)^{-1} \in F_{16}$.
Then $x^{-1} = u \cdot x^* = \frac{x^*}{x \cdot x^*}$

Checking SIOXs



Witness:

$x \in F_{256}$

$u \in F_{16}$

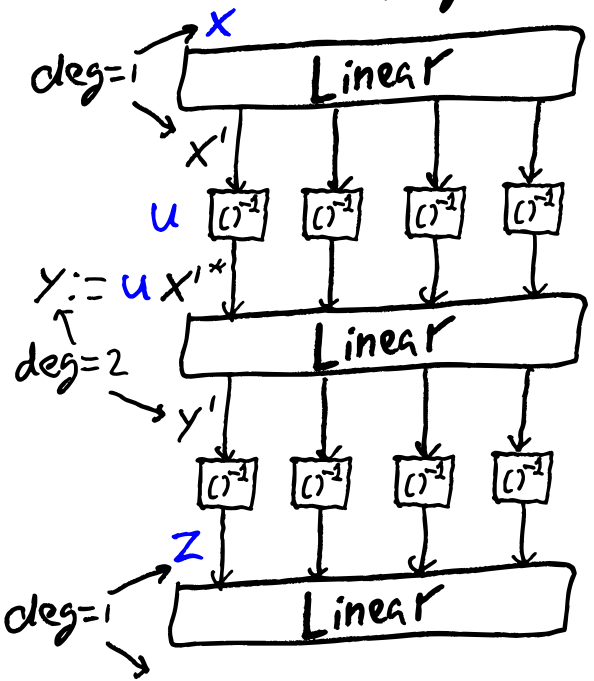
$z \in F_{256}$

Constraints:

$u x' x'^{*} = 1$

$y' z = 1$

Checking SNOXs



Witness:

$x \in F_{2^{96}}$

$u \in F_{16}$

$z \in F_{2^{96}}$

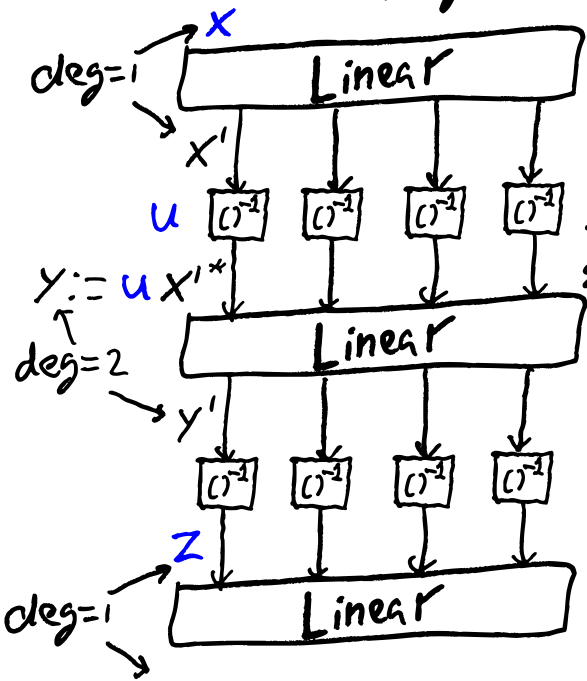
Constraints:

$u x' x'^{*} = 1$

$y' z = 1$

all deg = 3.

Checking SPOXs



25% smaller witness

Witness:

$x \in F_{256}$

$u \in F_{16}$

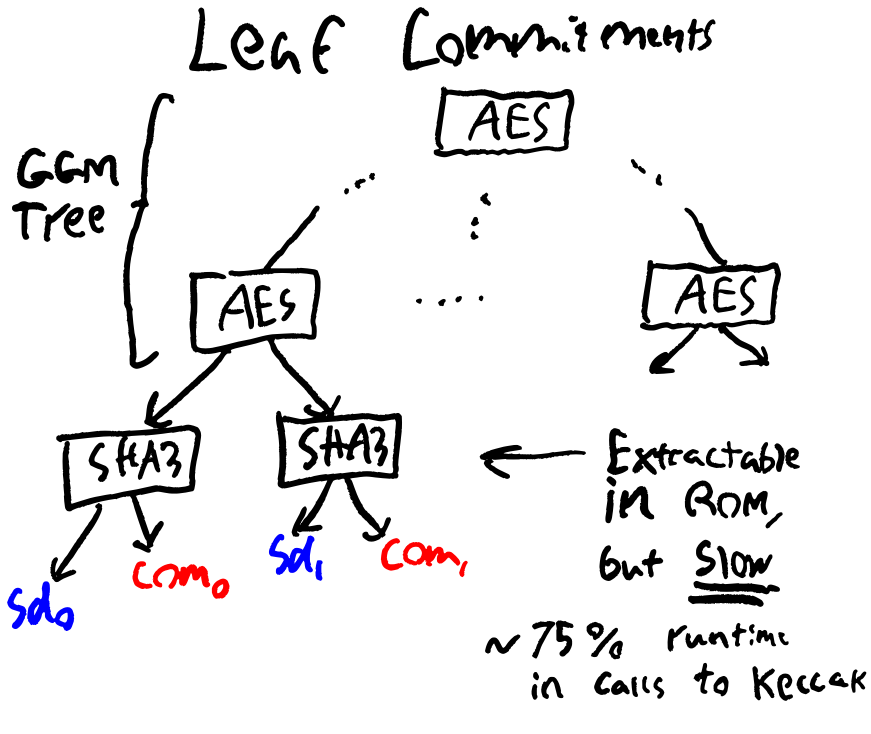
$z \in F_{256}$

Constraints:

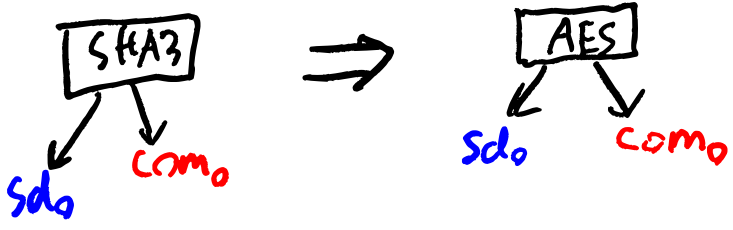
$u x' x'^* = 1$

$y' z = 1$

all deg = 3.


Leaf Commitments
GGM Tree
AES
AES
AES
SHA3
SHA3
Extractable in ROM, but Slow
sd0
com0
sd1
com1
~75% runtime in calls to Keccak

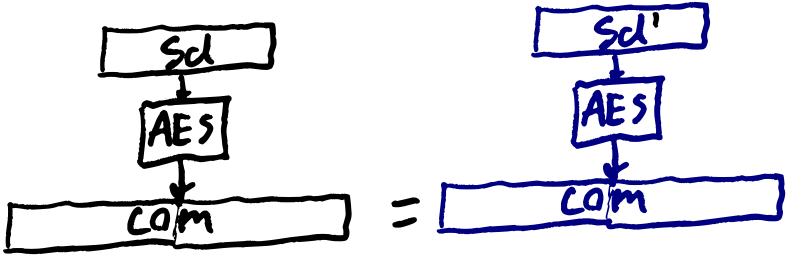
Replace SHA with AES?



Don't want to assume AES is an Ideal Cipher

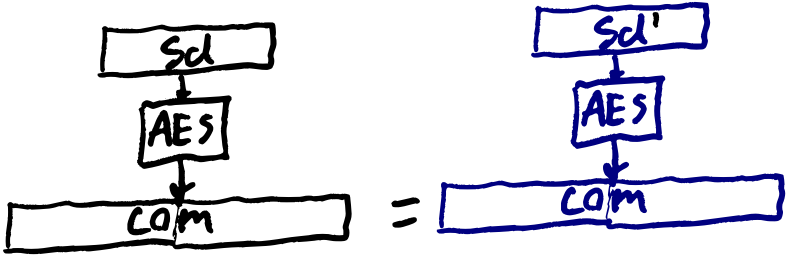
Almost-Injective PRGs

Hard to find sd such that $\exists$ sd' $\neq$ sd:



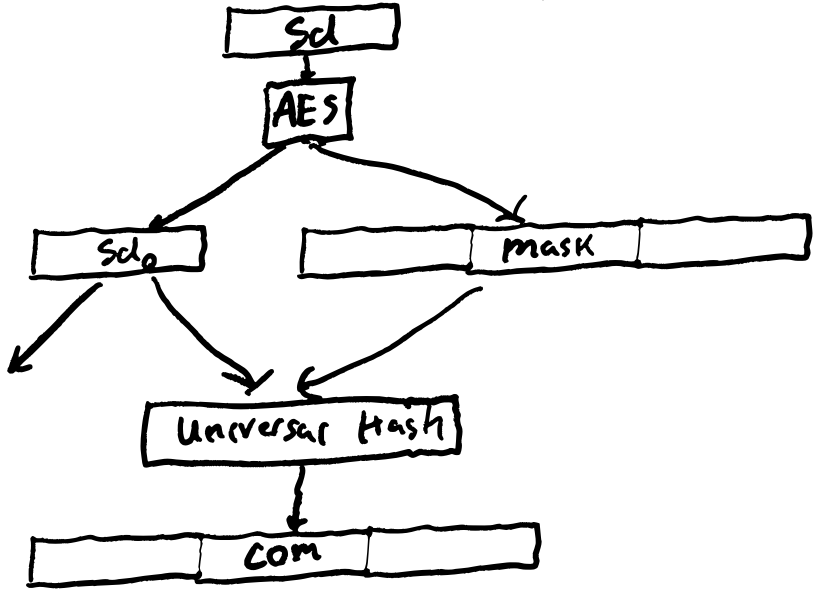
Almost-Injective PRGs

Hard to find sd such that $\exists$ sd' $\neq$ sd:

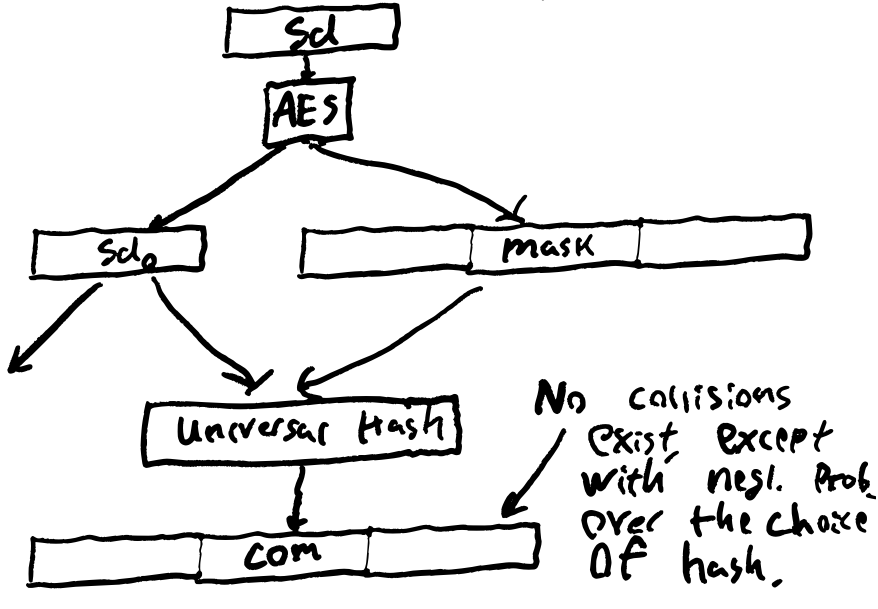


Like collision-resistance except the game finds the 2nd preimage for you.

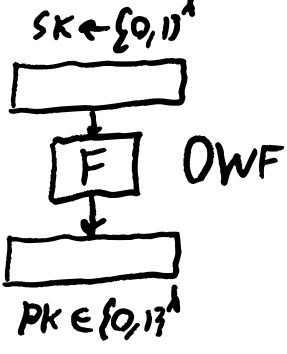
Statistically Binding Com.



Statistically Binding Com.

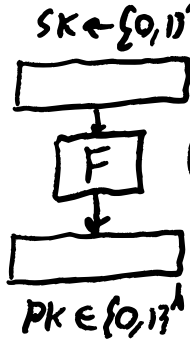


Extraction?



Needs a Proof of Knowlege

Extraction?

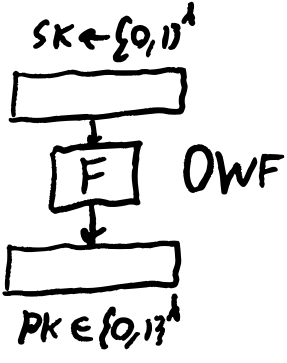


Needs a Proof of Knowlege

Requires either:

- Ideal cipher, to extract from AES-based Com.
- Rewinding. (In QROM!) Not tight.

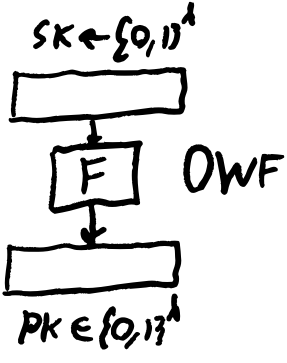
Soundness - Only
(Ineffient Extraction)



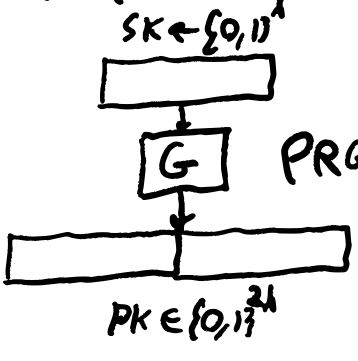
Needs a Proof
of Knowlege

Soundness-Only

Katz-Wang:

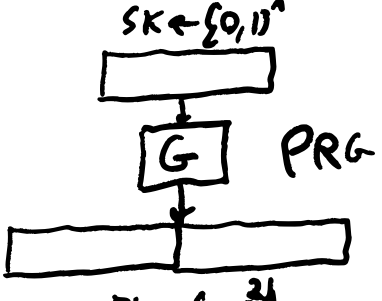


Needs a Proof of Knowlege



Soundness is sufficient.

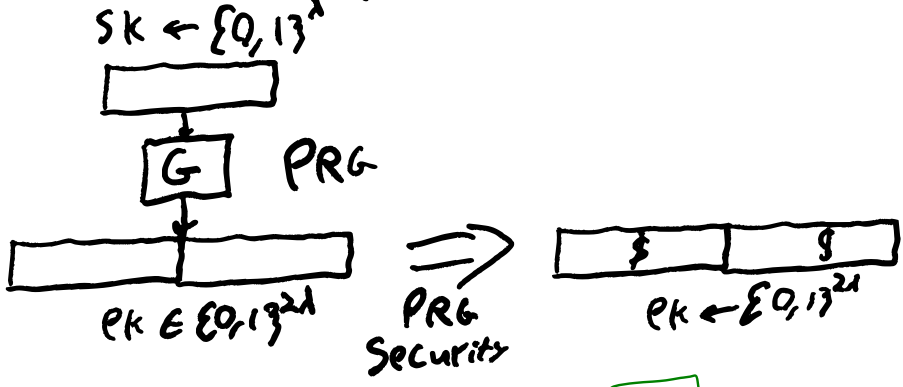
Soundness-Only Katz-Wang:



Soundness is easy to prove in either ROM or QROM, as the interactive proof is statistically Round-by-round sound. (As the com. is stat. binding)

Soundness is sufficient.

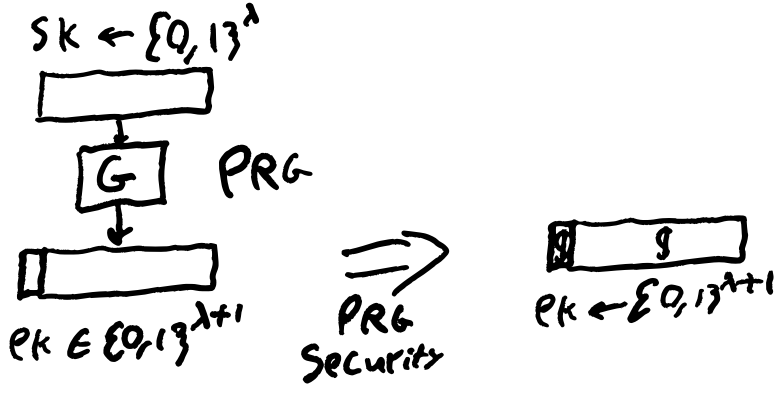
Katz-Wang



$$Pr[pk \in Range(G)] \leq \frac{2^\lambda}{2^{2\lambda}}$$

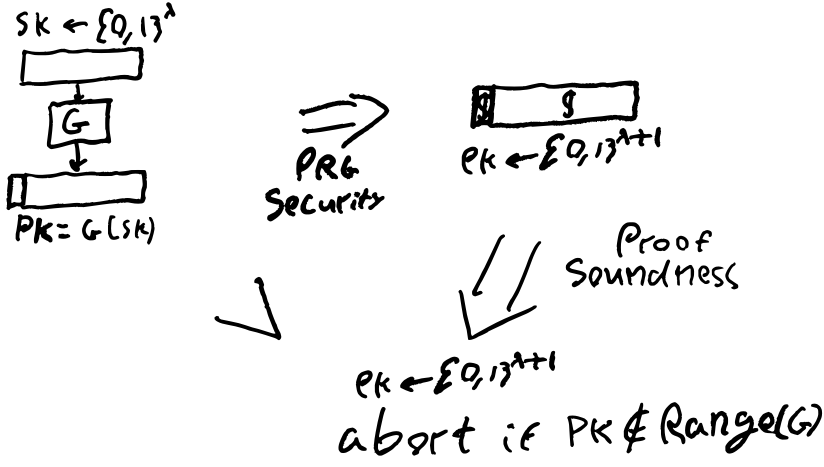
negl.

Improved Katz-Wang

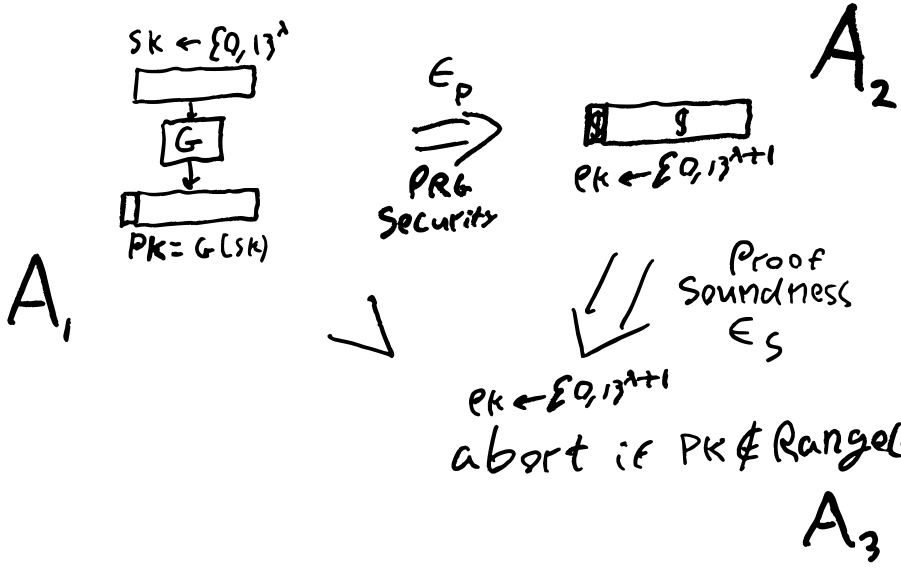


$$Pr[pk \in Range(G)] \leq \frac{2^\lambda}{2^{\lambda+1}} = \boxed{\frac{1}{2}}$$

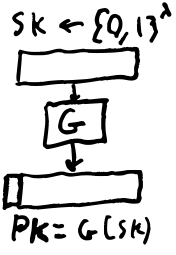
Improved Katz-Wang



Improved Katz-Wang

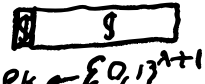


Improved Katz-Wang



ϵ_p

PRG Security



A_2

A_1

Proof Soundness ϵ_s

$pk \leftarrow \{0,1\}^{\lambda+1}$

abort if $pk \notin Range(G)$

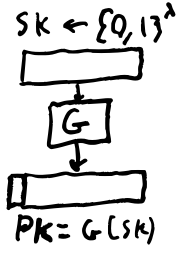
A_3

$A_1 \leq \epsilon_p + A_2$

$A_2 \leq \epsilon_s + A_3$

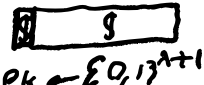
$A_3 \leq \frac{1}{2} A_1$

Improved Katz-Wang



ϵ_p

PRG Security



A_2

A_1

Proof Soundness ϵ_s

$pk \leftarrow \{0,1\}^{\lambda+1}$

abort if $pk \notin Range(G)$

A_3

$A_1 \leq \epsilon_p + A_2$

$A_2 \leq \epsilon_s + A_3$

$A_3 \leq \frac{1}{2} A_1$

$A_1 \leq \epsilon_p + \epsilon_s + \frac{A_1}{2}$

$A_1 \leq 2\epsilon_p + 2\epsilon_s$

Leaf Commitment Benchmarks

Table 1: Runtimes in ns per commitment for the different LeafCommit options.

LeafCommit	Size	$\lambda = 128$		$\lambda = 192$		$\lambda = 256$	
SHAKE	2λ	164.07		165.56		166.64	
AES-UniHash	3λ	22.89	(13.95%)	65.93	(39.82%)	73.21	(43.93%)
AES-Simple	2λ	10.44	(6.36%)	41.73	(25.21%)	42.75	(25.65%)

Signature Benchmarks: FAEST

Table 2: Key and signature sizes in B, and runtimes in µs for FAEST variants.

	Variant	$\lvert sk \rvert$	$\lvert pk \rvert$	**F**ast			**S**mall		
				$\lvert \sigma \rvert$	Sign	Verify	$\lvert \sigma \rvert$	Sign	Verify
128	[AC:BBMORR24]	32	32	6 052	878	802	4 594	6 485	5 790
	this	32	32	5 924	524	432	4 506	3 878	3 041
192	[AC:BBMORR24]	56	64	16 100	2 149	1 996	12 028	18 545	14 598
	this	40	48	14 948	2 030	1 759	11 260	15 038	11 475
256	[AC:BBMORR24]	64	64	28 084	3 321	3 226	21 752	25 169	25 364
	this	48	48	26 548	3 067	2 901	20 696	24 716	24 726

Signature Benchmarks: FAEST-EM

Table 3: Key and signature sizes in B, and runtimes in µs for FAEST-EM variants.

	Variant	|sk|	|pk|	**F**ast			**S**mall		
				$\|\sigma\|$	Sign	Verify	$\|\sigma\|$	Sign	Verify
EM-128	[**AC:BBMORR24**]	32	32	5 444	852	786	4 170	6 389	6 077
	this	32	32	5 060	455	337	3 906	2 809	2 253
EM-192	[**AC:BBMORR24**]	48	48	13 532	1 944	1 899	10 108	17 423	16 800
	this	48	48	12 380	1 467	1 333	9 340	11 401	10 576
EM-256	[**AC:BBMORR24**]	64	64	26 036	3 260	3 119	19 744	25 382	24 553
	this	64	64	23 476	2 615	2 431	17 984	17 537	17 004

Summary

- Optimized the signature size by switching to degree 3 constraints. Total signature plus public key size in the smallest setting nearly matches Dilithium.
- Optimized the runtime by using AES for the leaf commitments.
- Restructured the scheme to be based on just soundness of the underlying proof, allowing for a simpler and tighter analysis in both ROM and QROM.
  Our bounds seems to be tight to within a small multiplicative constant < 10.