SPHINCS$^+$

Jean-Philippe Aumasson, Daniel J. Bernstein, Ward Beullens,
Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer,
Stefan-Lukas Gazdag, **Andreas Hülsing**, Panos Kampanakis, Stefan Kölbl,
Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen,
Christian Rechberger, Joost Rijneveld, Peter Schwabe, Bas Westerbaan

Hash-based signatures
(Merkle ´89)

Boring crypto:

- Dates back to beginning of public key cryptography
- No fancy new mathematical assumption:
  Only requires a secure hash function
  („minimal security assumptions“)
- Stateful schemes already in standardization

Hash-based signatures

(Merkle ´89)

Boring crypto:

- Dates back to beginning of public key cryptography
- No fancy new mathematical assumption:
  Only requires a secure hash function
  („minimal security assumptions“)
- Stateful schemes already ~~in standardization~~ standardized ✓

SPHINCS (Eurocrypt 2015)

Joint work with Daniel J. Bernstein, Daira Hopwood, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn

Stateless hash-based signatures

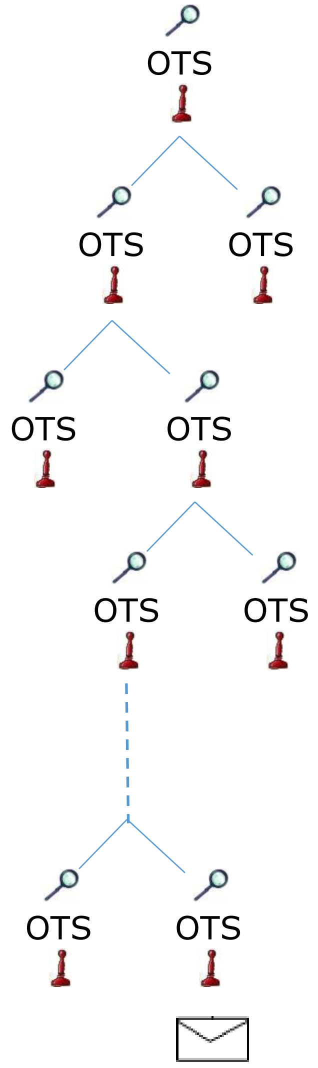
[NY89,Gol87,Gol04]

Goldreich's approach [Gol04]:

Security parameter $\lambda = 128$

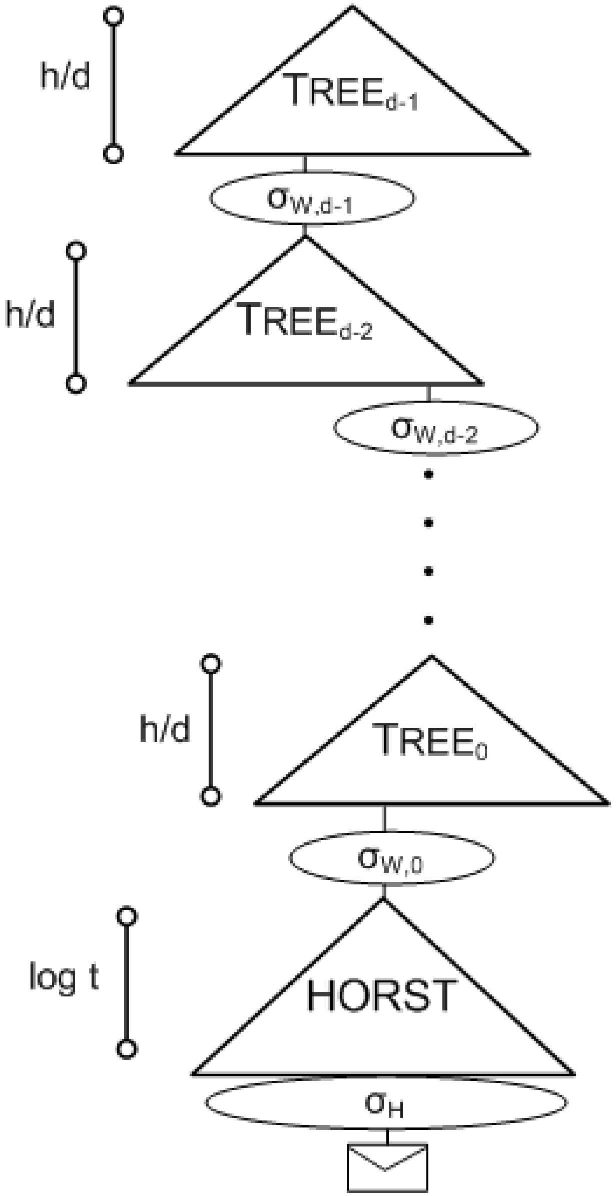
Use binary tree as in Merkle, but...

- ...for security
  - pick index i at random;
  - requires huge tree to avoid index collisions (e.g., height $\mathrm{h} = 2\lambda = 256$).
- ...for efficiency:
  - use binary certification tree of OTS key pairs (= Hypertree with $d = h$),
  - all OTS secret keys are generated pseudorandomly.



SPHINCS [BHH+15]

- Select index pseudorandomly
- Use a few-time signature key-pair on leaves to sign messages
  - Few index collisions allowed
  - Allows to reduce tree height
- Use hypertree: Use d << h.



SPHINCS^{+} vs SPHINCS

- Allow for 2^{64} instead of 2^{50} signatures per key pair
- Add multi-target attack mitigation (Tweakable hash functions)
- “Simple” and “Robust” parameters
- New few-time signature scheme FORS
- Verifiable index selection
- Optional non-deterministic signatures

SPHINCS$^+$ in 3rd Round

Joint work with Jean-Philippe Aumasson, Daniel J. Bernstein, Ward Beullens, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Bas Westerbaan

3rd Round changes

- Two new team members: Ward Beullens, Bas Westerbaan
- New parameter sets (more efficient at same security)
- (Discussed hierarchical PRG & constant sum WOTS but discarded both)

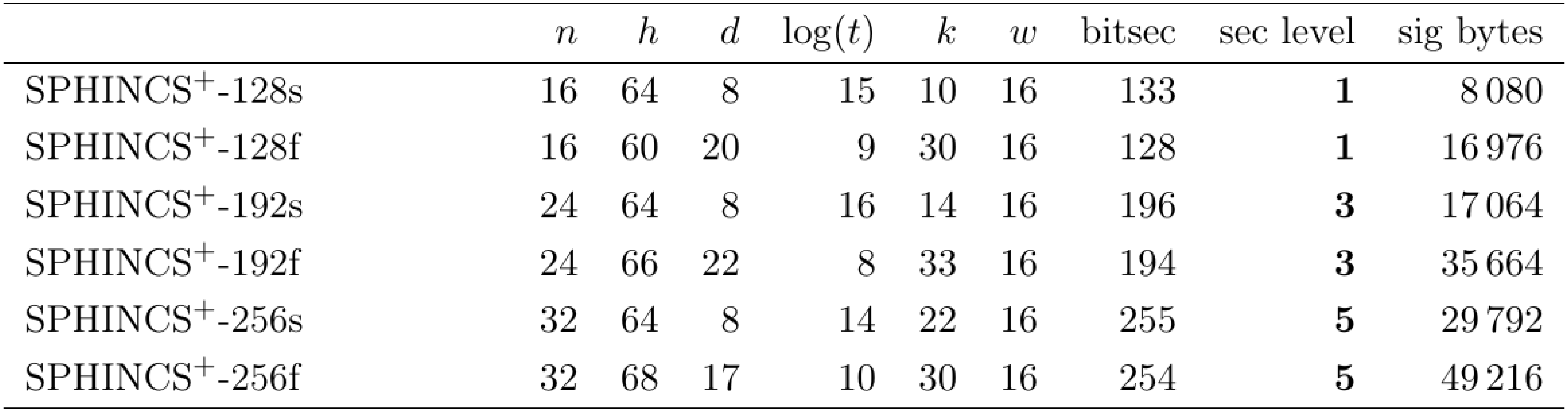
New parameter sets

Search criteria:

- Improvement in optimized metric (**f**ast / **s**mall)
- No significant penalty in other metric
- No worse verification speed
- No change to security assumptions / strength
- No increased complexity

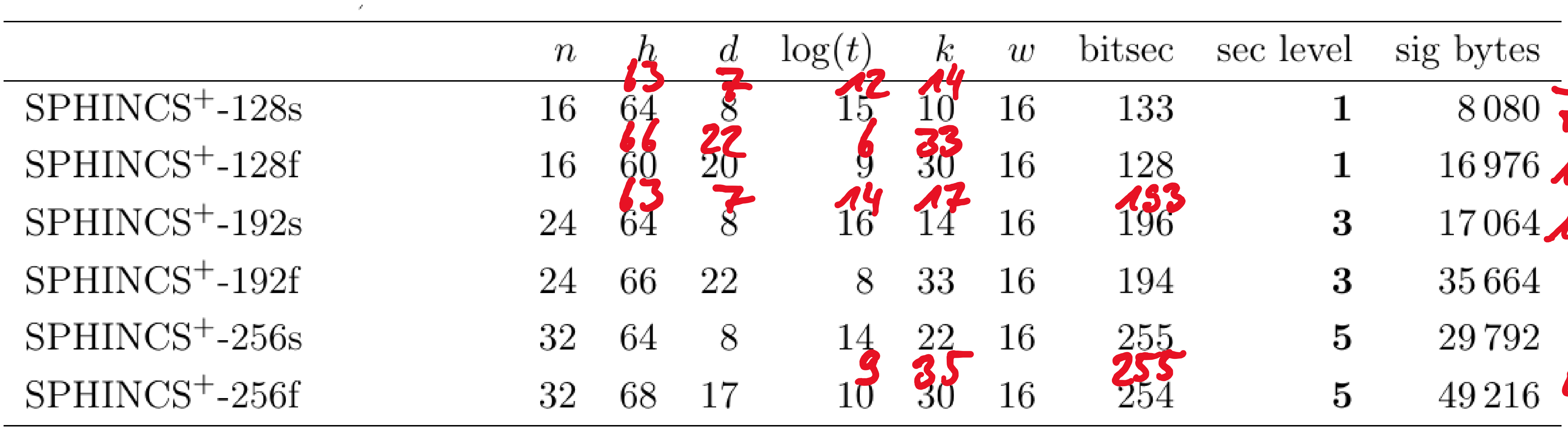
-> We only changed h, d, log(t) & k

New parameter sets



	n	h	d	$\log(t)$	k	w	bitsec	sec level	sig bytes
SPHINCS$^+$-128s	16	64	8	15	10	16	133	**1**	8 080
SPHINCS$^+$-128f	16	60	20	9	30	16	128	**1**	16 976
SPHINCS$^+$-192s	24	64	8	16	14	16	196	**3**	17 064
SPHINCS$^+$-192f	24	66	22	8	33	16	194	**3**	35 664
SPHINCS$^+$-256s	32	64	8	14	22	16	255	**5**	29 792
SPHINCS$^+$-256f	32	68	17	10	30	16	254	**5**	49 216

New parameter sets



	n	h	d	$\log(t)$	k	w	bitsec	sec level	sig bytes	
SPHINCS$^+$-128s	16	63 64	7 8	12 15	14 10	16	133	**1**	8 080	7 856
SPHINCS$^+$-128f	16	66 60	22 20	6 9	33 30	16	128	**1**	16 976	17 088
SPHINCS$^+$-192s	24	63 64	7 8	14 16	17 14	16	133 196	**3**	17 064	16 224
SPHINCS$^+$-192f	24	66	22	8	33	16	194	**3**	35 664	
SPHINCS$^+$-256s	32	64	8	14	22	16	255	**5**	29 792	
SPHINCS$^+$-256f	32	68	17	9 10	35 30	16	255 254	**5**	49 216	49 856

New parameter sets

	sign	verify	sig	sec
128s	± 0	- 8 %	- 2.77 %	± 0
128f	- 24 %	+ 10 %	+ 0.66 %	± 0
192s	- 20 %	- 10 %	- 4.92 %	-3 bit (still 193 > 192)
192f	± 0	± 0	± 0	± 0
256s	± 0	± 0	± 0	± 0
256f	- 13 %	± 0	+ 1.30 %	+1 bit

Changes in speed are averaged over robust / simple & SHA2, SHAKE & Haraka parameter sets. For more details see our change log and the latest specification.

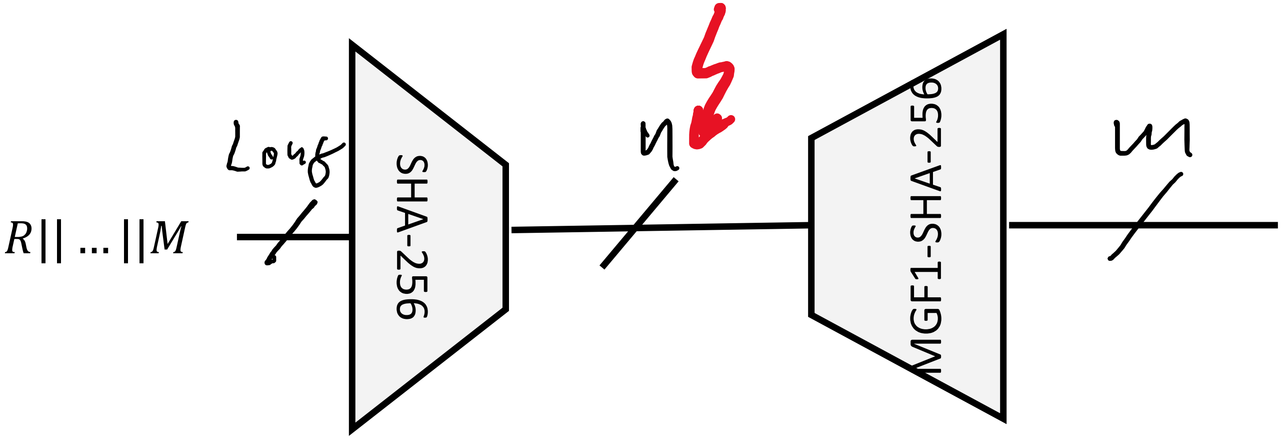
After round three updates

H_msg with SHA-256 #1

Feb 11: Mail by Morgan Stern

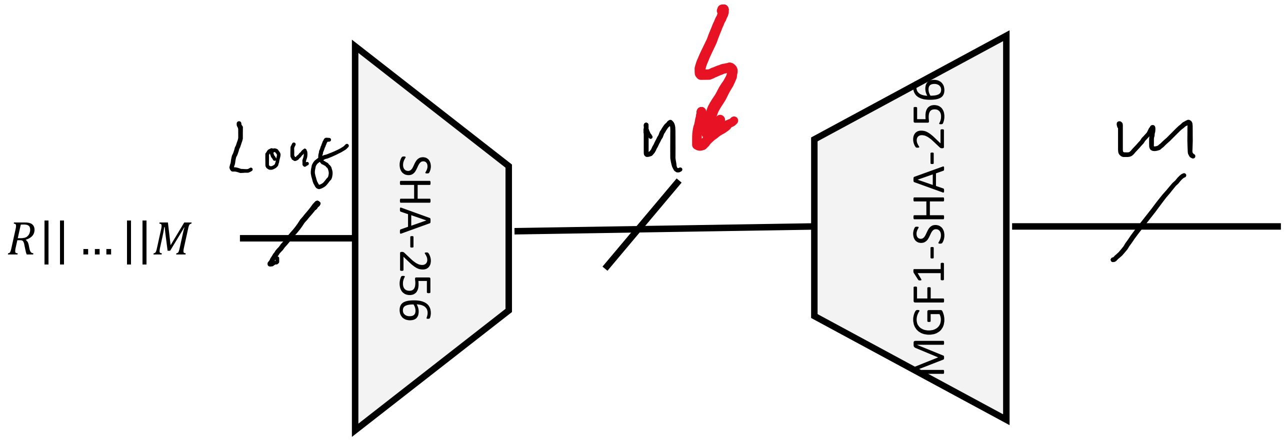
"In particular, in SPHINCS+-SHA-256 there is an issue with the definition of the H_msg function so that the security of the signature presently relies on the multi-target second pre-image resistance of the SHA-256 hash function."

$$H\_msg(R, PK.seed, PK.root, M)$$
$$= \text{MGF1}-\text{SHA}-256(\text{SHA}-256(R||PK.seed||PK.root||M), m).$$



H_msg with SHA2-256 #1

- The multi-target second preimage attack loses about 64 bit in security
- Security down to 192 bits (for all SHA-256 parameters)
- Violates L5
- Fix: Switch to SHA2-512 for H_msg (& H_PRF) at L5.



H_msg with SHA2-256 #2

Feb 16: Mail by John Kelsey

“I believe there’s also a long-message second preimage attack that applies here. (Ray Perlner pointed this out in a discussion.)”

Fix:

$$\text{H\_msg}\ : \\ = \text{MGF1}-\text{SHA}-\text{X}(R \,||\, PK.seed \,||\, \text{SHA}-\text{X}(R \,||\, PK.seed \,||\, PK.root \,||\, M\,), m)$$

(where X is 256 for L1 & L3, and 512 for L5)

H_msg with SHA2-256 #2

Fix:

$$\text{H\_msg} = \text{MGF1}-\text{SHA}-\text{X}(R \,||\, PK.seed \,||\, \text{SHA}-\text{X}(R \,||\, PK.seed \,||\, PK.root \,||\, M\,), m)$$

(where X is 256 for L1 & L3, and 512 for L5)

Attack:

1. Ask for q signatures on long messages (2^k message blocks)
2. Find expandable messages (takes time $\sim O(2^{n/2})$)
3. Find collision between expandable message and a message block in long message (takes times $\mathrm{O}(2^{n-k-\cancel{\log q}-1})$)
4. Expand expandable message sufficiently

H_msg with SHA2-256 #2

- Attack before fix takes time $\mathrm{O}(2^{n/2} + 2^{n-k-\log q-1})$
- Max values are $\mathrm{q} = 2^{64}, k = 55$ ⇒We lose 119 bit security.
- Recall: Honest user signs!
- Assume compression function call takes 2^{-22}seconds ($\approx 200ns$).
- Attack takes $2^{64} \cdot\ 2^{55} =\ 2^{119}$compression function calls.
- That is 2^{97}sec = 2^{72} years.
- Still $\mathbf{2^{52}}$ **years** if key continuously used on 1 million machines!

Conclusion

- Possible synergies with standardizing stateful hash-based signatures
- *The* most conservative submission in the competition.

Thank you!

Questions?