

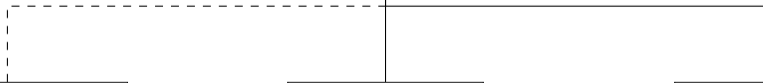
Grafting Trees

THALES

Laurent Castelnovi

Ange Martinelli

Thomas Prest



Outline of this talk

- 1 Introduction
- 2 Hash-based signatures
 - 1 One-time signatures (OTS)
 - 2 Merkle's construction
 - 3 Goldreich's construction
 - 5 The SPHINCS framework
- 3 Grafting trees
 - 1 Outline of the attack
 - 2 Faulting step
 - 3 Grafting step
 - 4 Specifics of each scheme
- 4 Conclusion



Introduction

Hash-based signatures:

- ⇒ Signatures based on the collision or preimage resistance of hash functions
- ⇒ Optimal from a security perspective [Rom90]
- ⇒ Post quantum: two proposals to NIST's CFP [AE17, BDE⁺17]

Obvious question: do they resist to fault attacks?

- ⇒ Short answer: No.
- ⇒ This talk: a fault attack against schemes of the SPHINCS family:
 - ⇒ The original SPHINCS [BHH⁺15]
 - ⇒ Gravity-SPHINCS [AE17]
 - ⇒ SPHINCS⁺ [BDE⁺17]



Introduction

Hash-based signatures:

- ⇒ Signatures based on the collision or preimage resistance of hash functions
- ⇒ Optimal from a security perspective [Rom90]
- ⇒ Post quantum: two proposals to NIST's CFP [AE17, BDE⁺17]

Obvious question: do they resist to fault attacks?

- ⇒ Short answer: No.
- ⇒ This talk: a fault attack against schemes of the SPHINCS family:
 - ⇒ The original SPHINCS [BHH⁺15]
 - ⇒ Gravity-SPHINCS [AE17]
 - ⇒ SPHINCS⁺ [BDE⁺17]

Let's fault stuff!



One-time signatures (OTS) from hash functions

A toy example:

⇒ $sk = (s_1, s_2) \in \{0, 1\}^{256 \times 2}$

⇒ $pk = (p_1, p_2) = (H^N(s_1), H^N(s_2))$

⇒ Signature of a message $m \in \{0, \dots, N\}$:

$$\text{sig}(m) = H^{m(s_1), H^{N-m}(s_2)}(1)$$

⇒ one signature ⇒ existentially unforgeable

⇒ two signatures ⇒ existential forgery for a proportion $\approx \frac{|m_1 - m_2|}{N}$ of the messages

One-time signatures (OTS) from hash functions

A toy example:

$$\Rightarrow sk = (s_1, s_2) \in \{0, 1\}^{256 \times 2}$$

$$\Rightarrow pk = (p_1, p_2) = (H^N(s_1), H^N(s_2))$$

\Rightarrow Signature of a message $m \in \{0, \dots, N\}$:

$$\text{sig}(m) = H^{m(s_1), H^{N-m}(s_2)}(1)$$

\Rightarrow one signature \Rightarrow existentially unforgeable

\Rightarrow two signatures \Rightarrow existential forgery for a proportion $\approx \frac{|m_1 - m_2|}{N}$ of the messages

For WOTS(+), the OTS used in schemes of the SPHINCS family:

\Rightarrow one signature \Rightarrow existentially unforgeable

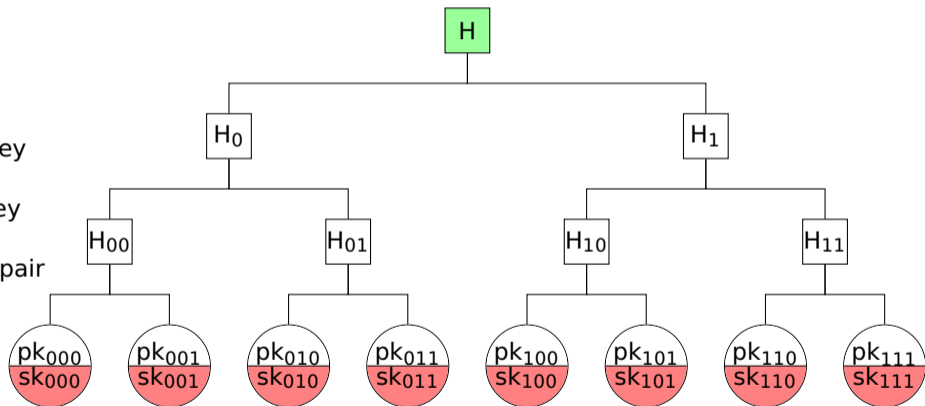
\Rightarrow two signatures \Rightarrow existential forgery for a proportion 2^{-34} of the messages

Merkle's construction [Mer90]

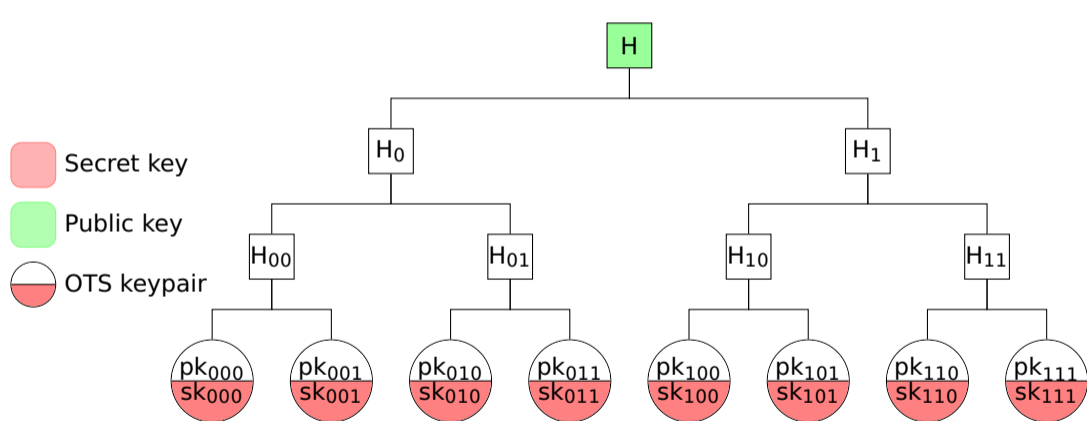
Secret key

Public key

OTS keypair

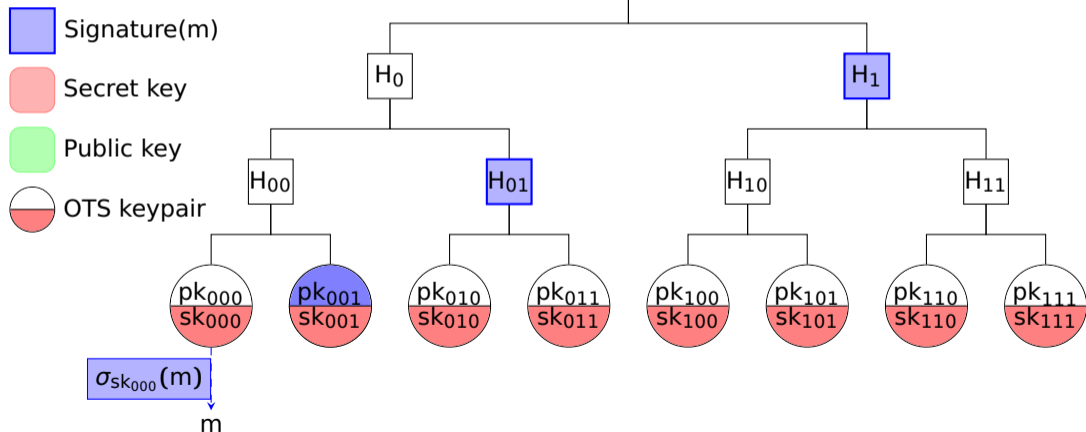


Merkle's construction [Mer90]

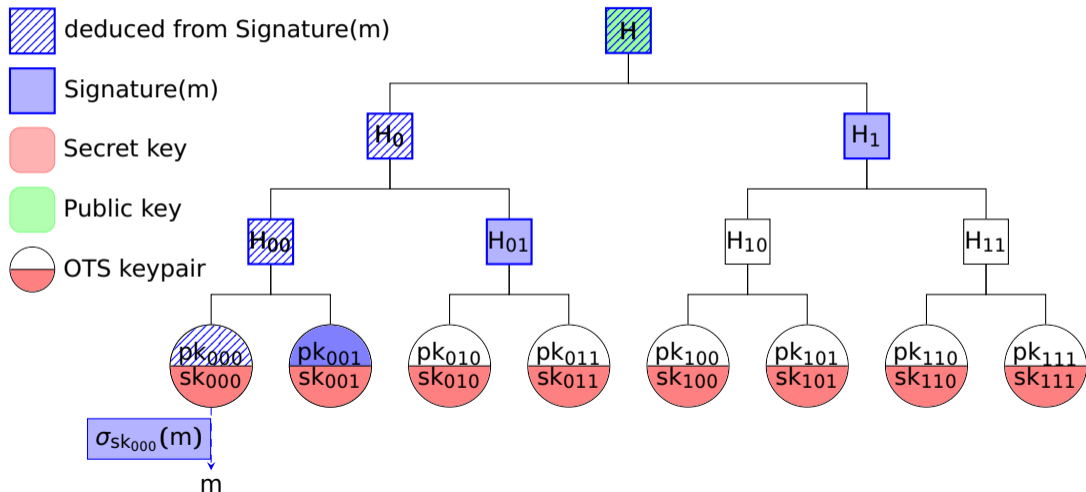


m

Merkle's construction [Mer90]



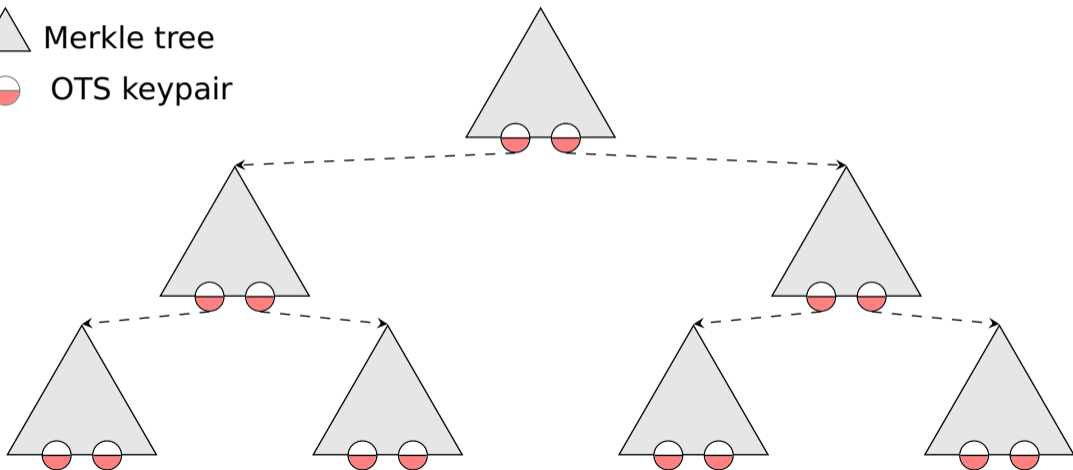
Merkle's construction [Mer90]



Goldreich's construction (abstract) [Gol86]

△ Merkle tree

○ OTS keypair



Goldreich's construction (detailed)



deduced from Signature(m)



Signature(m)



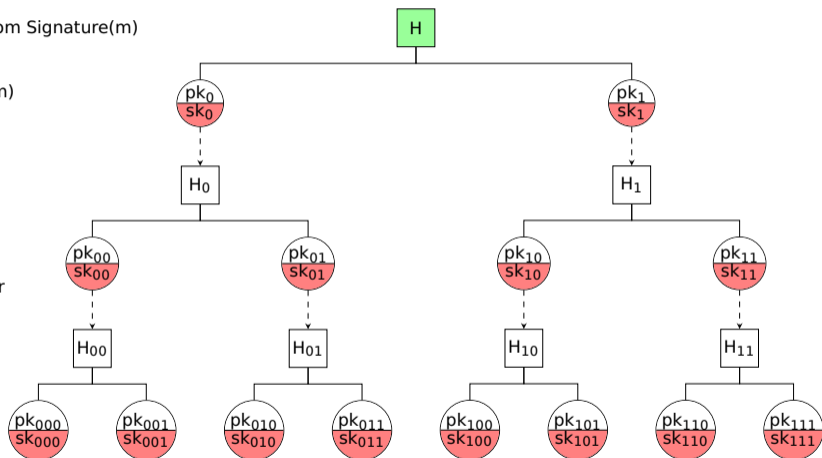
Secret key



Public key



OTS keypair



Goldreich's construction (detailed)



deduced from Signature(m)



Signature(m)



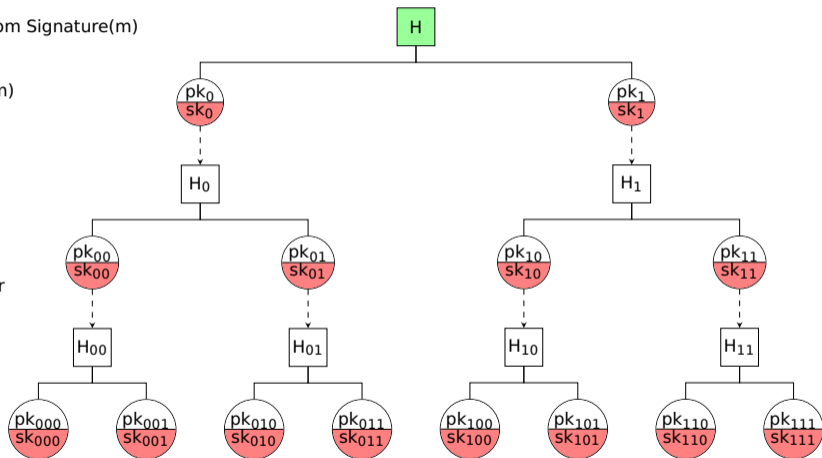
Secret key



Public key

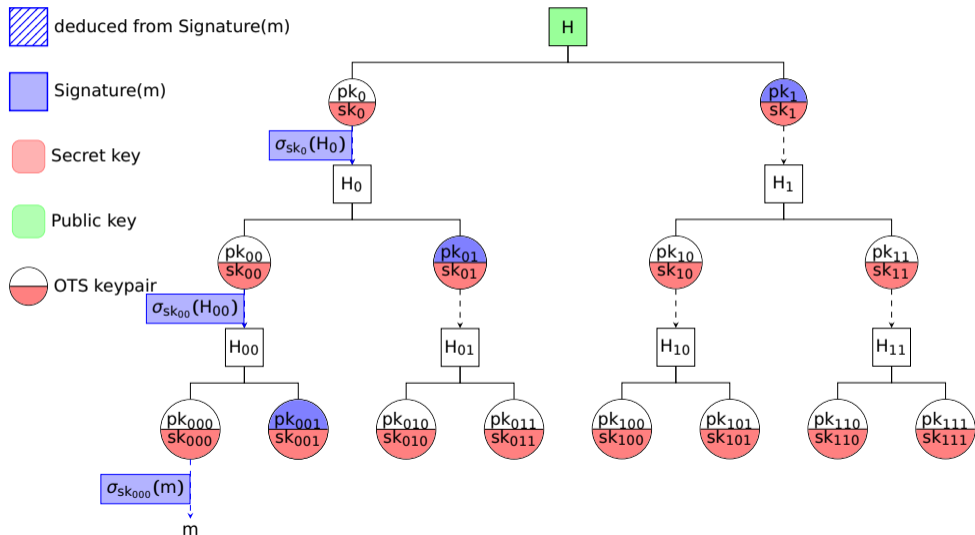


OTS keypair

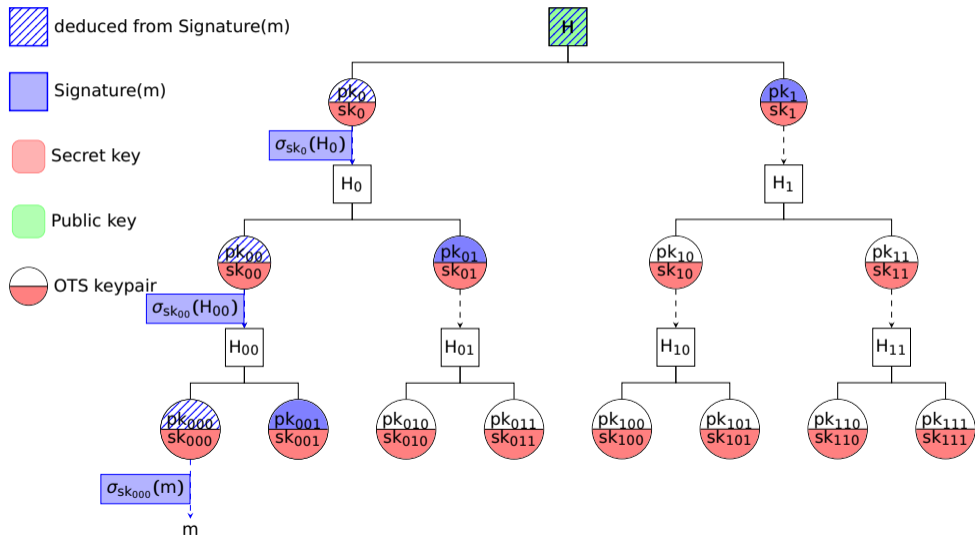


m

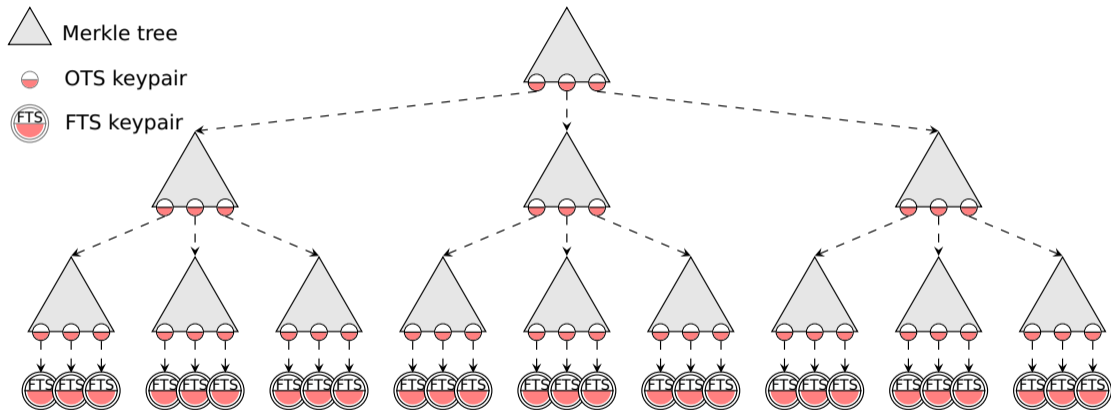
Goldreich's construction (detailed)



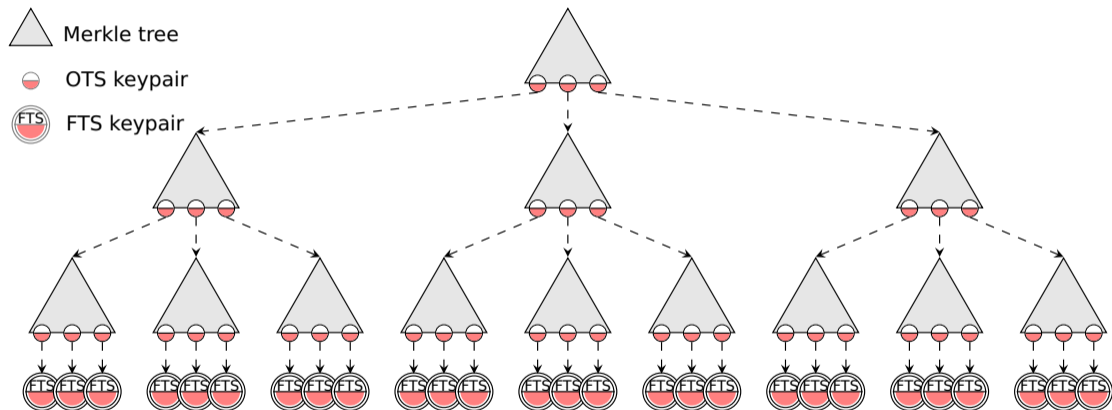
Goldreich's construction (detailed)



The SPHINCS framework (abstract)

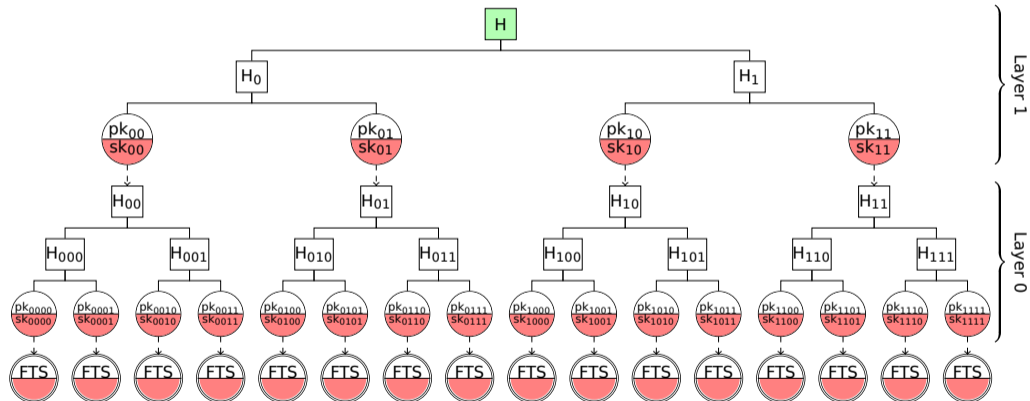


The SPHINCS framework (abstract)

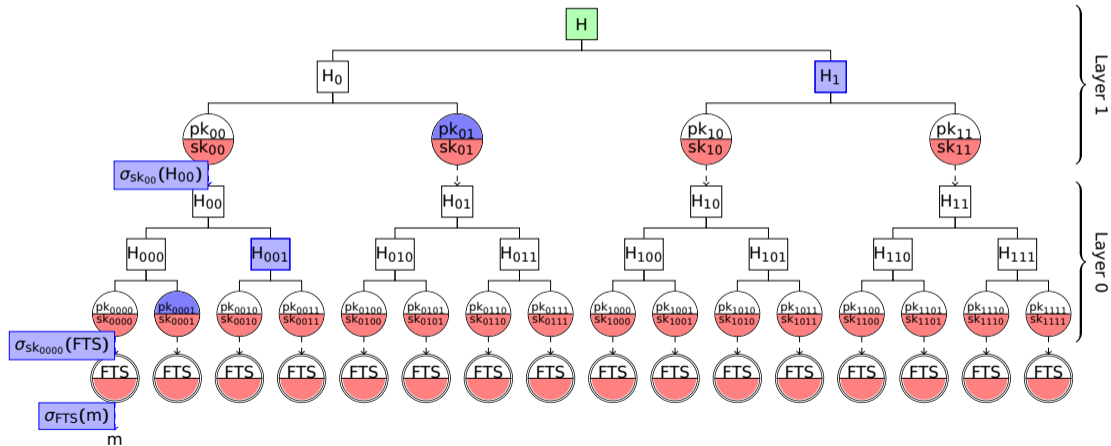


- ⇒ Common to SPHINCS [BHH⁺15], Gravity-SPHINCS [AE17] and SPHINCS⁺ [BDE⁺17]
- ⇒ Typical parameters: layers = 8, height of each Merkle tree = 8, total height = 64

The SPHINCS framework (detailed)



The SPHINCS framework (detailed)



Observations useful for our attack:

⇒ In all hash-based signatures:

[a valid signature $\sigma_{sk}(m)$] \Rightarrow [one can recover pk]

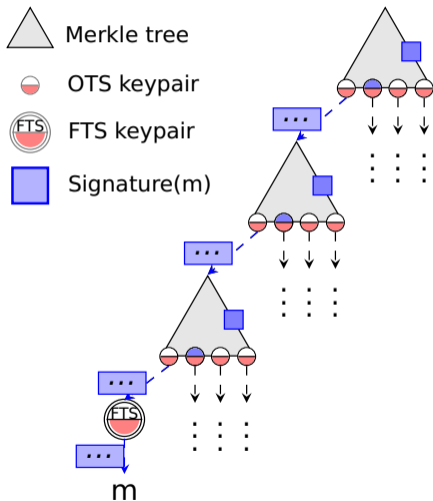
⇒ For the OTS used in SPHINCS:

[2 signatures] \Rightarrow [one can forge for 1 message over 2^{34}]

Outline of our attack:

- 1 *Faulting step.* We provoke a fault to make an OTS sign two different values
- 2 *Grafting step.* We use the compromised OTS to obtain an universal forgery

The faulting step



The faulting step:

- One normal sig(m), one faulted sig(m)
- Target the Merkle tree *just below the top*
- We may fault any computation "below" the authentication path

Regular vs faulted signature:

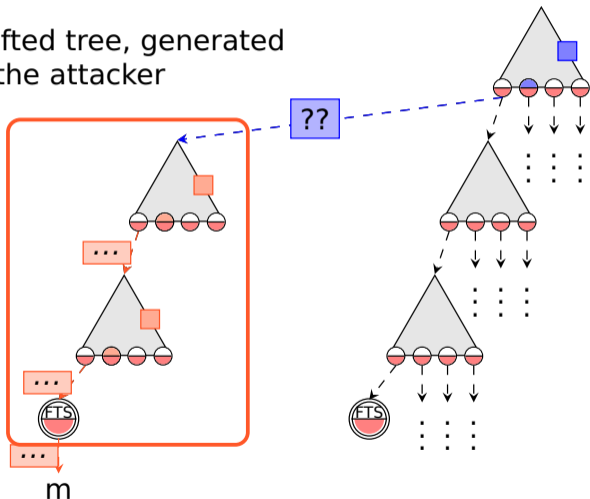
- Two \neq values are computed for the root of the faulted Merkle tree
- The top OTS signs two \neq values

Features of this fault:

- Stealthy
- One fault
- Little precision required

The grafting step

Grafted tree, generated by the attacker



Goal of the attacker:

- ⇒ Sign his own tree with the compromised OTS

Naïve approach:

- ⇒ Generate trees until a suitable one is found
- ⇒ Time: $2^{34} \times$ (generate a tree)

Adaptive approach:

- ⇒ Only modify the top of the grafted tree
- ⇒ Time: $2^{34} +$ (generate a tree)

Specifics of each scheme, impact on the attack

Index selection:

- 1 SPHINCS: $idx \leftarrow H(r, m)$, where r is private
⇒ very easy
- 2 Gravity-SPHINCS: $idx \leftarrow H(r, m)$, where $r \leftarrow H(sk, M)$
⇒ easy
- 3 SPHINCS⁺: $idx \leftarrow H(r, pk, m)$, where $r \leftarrow H(sk, \$, M)$
⇒ no control on the FTS index anymore, but still easy

Conclusion

Key takeaways:

- 1 A fault attack on schemes of the SPHINCS family
- 2 Universal forgery in one signature
- 3 Fault model is very weak:
 - 1 little to no control on the time of the fault
 - 2 little to no control on the precision of the fault
 - 3 independent of underlying hash function(s)
- 4 Stealthy
- 5 Specific countermeasures are ineffective (to our knowledge)

Related works:

- This work was based on Laurent Castelnovi's Master thesis [[Cas17](#)]
- Independently studied by Genêt [[Gen17](#)] and Kannwischer [[Kan17](#)]



Thanks!





Jean-Philippe Aumasson and Guillaume Endignoux.
Improving stateless hash-based signatures.
Cryptology ePrint Archive, Report 2017/933, 2017.
<https://eprint.iacr.org/2017/933>.



Daniel J. Bernstein, 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, and Peter Schwabe.
SPHINCS+, 2017.
<https://sphincs.org/>.



Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn.
SPHINCS: practical stateless hash-based signatures.
In *EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 368–397. Springer, 2015.



Laurent Castelnovi.
SÃcuritÃ physique de schÃmas cryptographiques post-quantique.
Master thesis, 2017.

Available at

<https://tprest.github.io/Publications/rapport-laurent-castelnovi.pdf>.



Aymeric Genêt.

Hardware attacks against hash-based cryptographic algorithms.

Master thesis, 2017.

Available at <https://infoscience.epfl.ch/record/253317>.



Oded Goldreich.

Two remarks concerning the Goldwasser-Micali-Rivest signature scheme.

In *CRYPTO '86*, volume 263 of *LNCS*, pages 104–110. Springer, 1986.



Matthias Kannwischer.

Physical attack vulnerability of hash-based signature schemes.

Master thesis, 2017.

Available at https://www.cdc.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_CDC/Documents/theses/Matthias_Kannwischer.master.pdf.



Ralph C. Merkle.

A certified digital signature.

In *CRYPTO' 89*, volume 435 of *LNCS*, pages 218–238. Springer, 1990.



John Rompel.

One-way functions are necessary and sufficient for secure signatures.

In *STOC*, pages 387–394. ACM, 1990.