

Let Live and Let Die: Handling the State of Hash-based Signatures

Stefan-Lukas Gazdag, Denis Butin
& Johannes Buchmann

04/02/2014 - PQ Workshop - NIST 2015

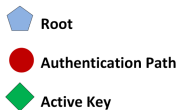
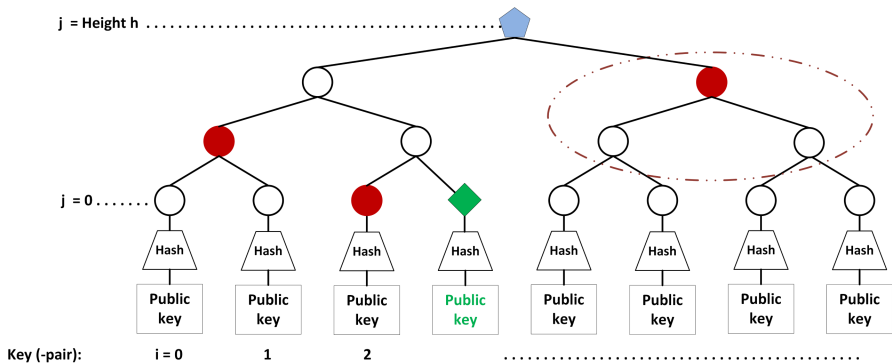


TECHNISCHE
UNIVERSITÄT
DARMSTADT

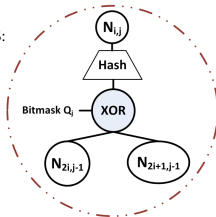
Presentation

- ▶ Introduction
- ▶ Statefulness
- ▶ Handling the state
- ▶ Protocol Integration and other considerations

Introduction: Merkle / XMSS tree



XMSS:



Statefulness

What's so bad about the state?

- Security leaks possible

- Software does not consider keys being stateful

- Missing infrastructure

Statefulness

Why bother coping with the state?

- Hash-based signatures well understood and post-quantum

- Current stateful methods faster than stateless ones

- Currently smaller signatures

- Forward secure constructions

Considerations

What we want:

Secure usage of secret key

What we need:

Practicability

The secret key

Considerations for the key:

- Any copies may reveal secrets

- Interrupts may threaten consistency

- Key is critical resource

Handling the state

How to cope with the state

- Index handling

- Error / consistency checking

- Storing

Index handling

Single state

Several two state solutions

Delegation of subtrees

Errors and Consistency

Does the index fit the actual state?

Is the state consistent itself?

Storing the secret key

Who's able to access the storage?

Has the key actually been written to storage?

⇒ Doesn't fit current libraries that well

Lots of use cases without tight restrictions:

- Update signing

- Email signing

But even with stricter timing:

- 200 ms maximum for SSH signature procedure

Key Management

Key provider concept

⇒ external management of key

Offers API to receive and write SK, PK,
authentication path information

Delegation of subsets of SK

INDEX $I = 3$

Index i_0	Index i_1	Index i_2	Index i_3	Index i_4
Auth 0	Auth 1	Auth 2	Auth 3	Auth 4
Sub tree 0	Sub tree 1	Sub tree 2	Sub tree 3	Sub tree 4

Protocol integration

Keys still fit most communication protocols

Need a PQ key exchange

Need PQ signatures (hash-based) for that

As seen in Andreas Hülsing's talk before

Internet-Draft available
[draft-huelsing-cfrg-hash-sig-xmss-00](#)

SPHINCS

See Daniel J. Bernstein's talk and SPHINCS paper
<http://sphincs.cr.yp.to/>

Conclusion

State can be managed a feasible way

But:

Trade-off: security $\langle \rangle$ performance

TBD: Exact comparison of those trade-offs

Thank you!

Questions?

www.pqsignatures.org