State Management for Hash-Based Signatures

David McGrew, Panos Kampanakis, Scott Fluhrer,
Stefan-Lukas Gazdag, Denis Butin, Johannes Buchmann

{mcgrew,pkampana,sfluhrer}@cisco.com
stefan-lukas_gazdag@genua.eu
{dbutin,buchmann}@cdc.informatik.tu-darmstadt.de

SSR 2016

CISCO

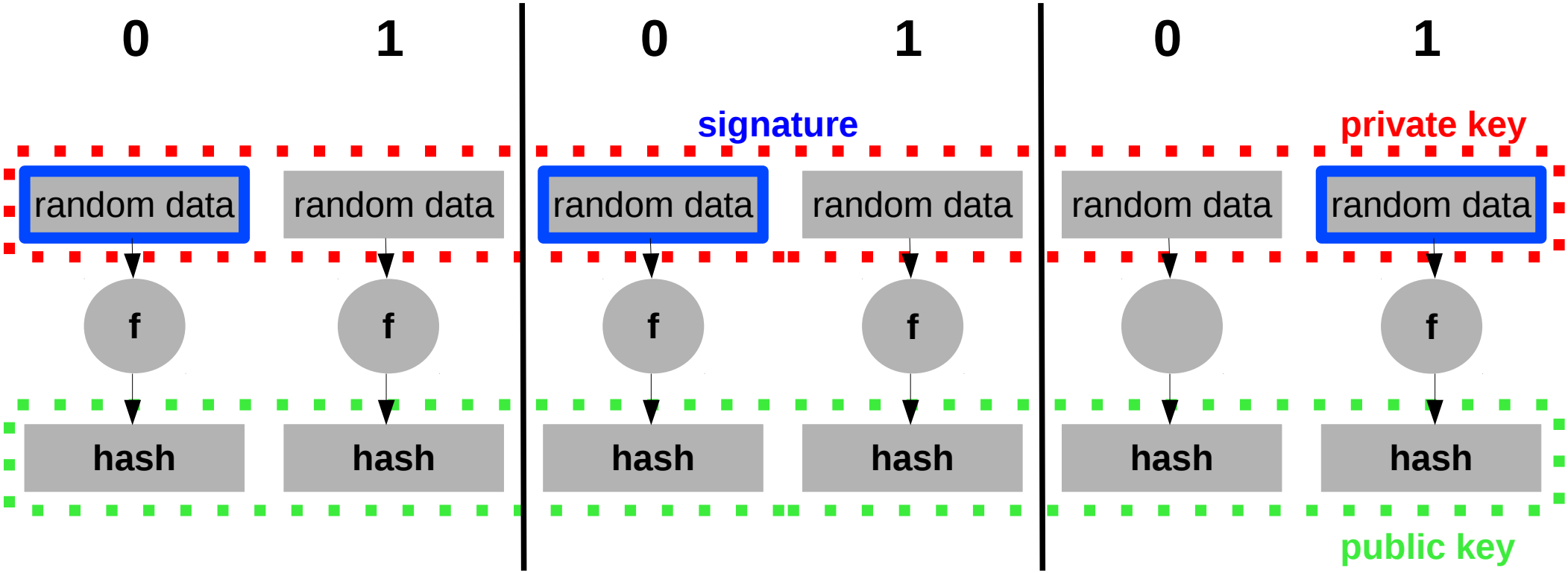
genua

TECHNISCHE UNIVERSITÄT DARMSTADT

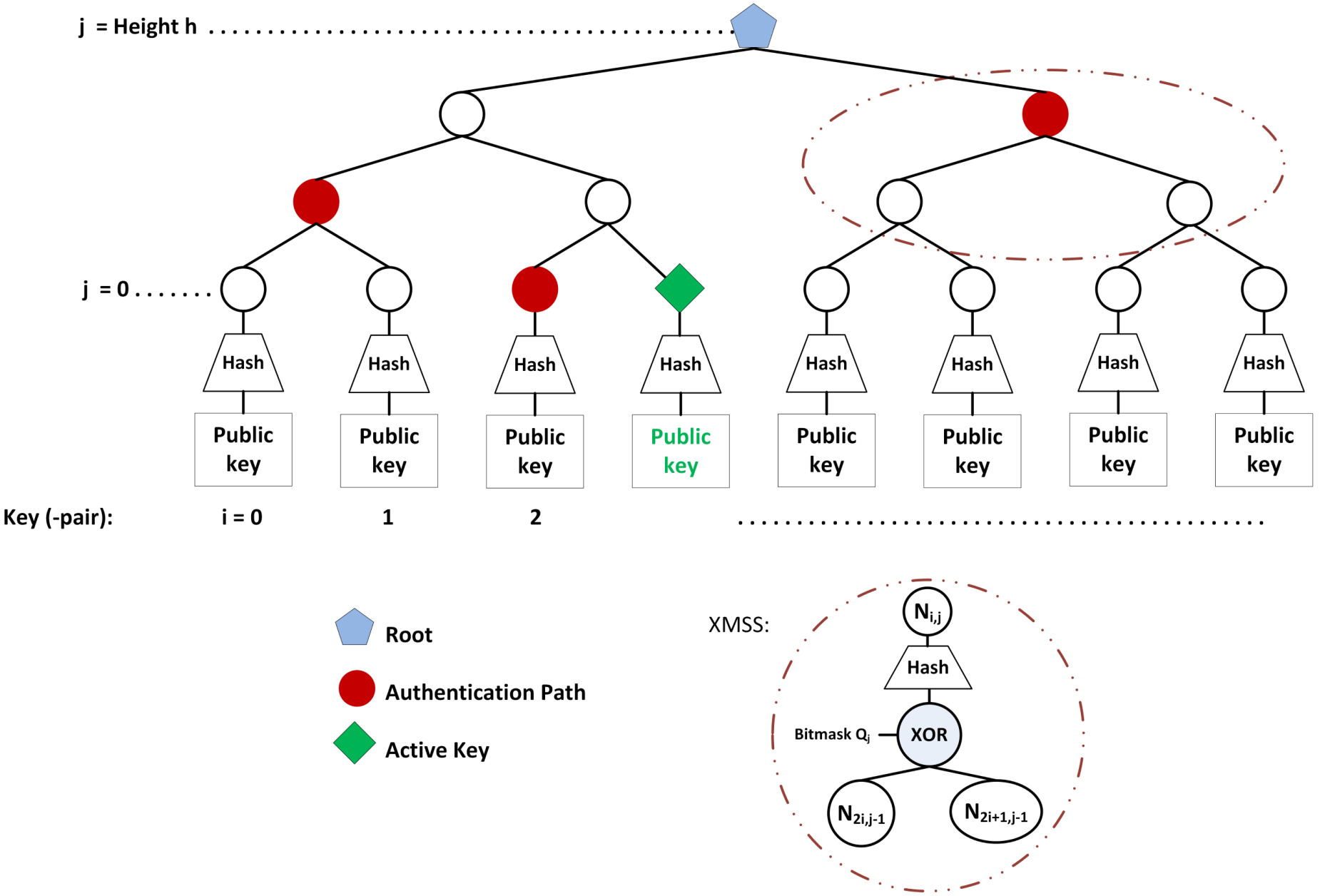
What's so great about HBS?

- Well understood
- Post-Quantum
- No further intractability assumptions other than cryptographic hash functions
- Minimal security requirements feasible
- Forward secure constructions possible

Intro: Hash-Based Signatures



Intro: Hash-Based Signatures



Statefulness

- Private key has to be updated
  - Any copy may reveal secrets
  - Interrupts may threaten consistency
  - Key is critical resource
  - Data to be updated differs by implementation decisions (Starting from single index to several nodes)

How about stateless schemes?

- SPHINCS (https://sphincs.cr.yp.to/)
  - Signatures size ~ 41 KB
  - Slower signing times

	Sig Size (B)	Pub Key Size (B)
LMS	2828	100
XMSS	2820	68
HSS	8688	112
XMSS^MT	8392	68
SPHINCS	41k	1056

Similar parameter sets,
total height of 30 for LMS and XMSS,
total height of 60 for HSS, XMSS^MT and SPHINCS.

How about stateless schemes?

- SPHINCS (https://sphincs.cr.yp.to/)
  - Signatures size ~ 41 KB
  - Slower signing times

Definitely working for some use cases!

But stateful schemes are sometimes still the better choice.

What's in line for standardization?

Crypto Forum Research Group
Internet-Draft
Intended status: Informational
Expires: May 4, 2017

D. McGrew
M. Curcio
S. Fluhrer
Cisco Systems

October 31, 2016

Hash-Based Signatures
draft-mcgrew-hash-sigs-05

Abstract

This note describes a digital signature system based on cryptographic hash functions, following the seminal work in this area of Lamport, Diffie, Winternitz, and Merkle, as adapted by Leighton and Micali in 1995. It specifies a one-time signature scheme and a general signature scheme. These systems provide asymmetric authentication without using large integer mathematics and can achieve a high security level. They are suitable for compact implementations, are relatively simple to implement, and naturally resist side-channel attacks. Unlike most other signature systems, hash-based signatures would still be secure even if it proves feasible for an attacker to build a quantum computer.

Crypto Forum Research Group
Internet-Draft
Intended status: Informational
Expires: April 22, 2017

A. Huelsing
TU Eindhoven
D. Butin
TU Darmstadt
S. Gazdag
genua GmbH
A. Mohaisen
SUNY Buffalo

October 19, 2016

XMSS: Extended Hash-Based Signatures
draft-irtf-cfrg-xmss-hash-based-signatures-07

Abstract

This note describes the eXtended Merkle Signature Scheme (XMSS), a hash-based digital signature system. It follows existing descriptions in scientific literature. The note specifies the WOTS+ one-time signature scheme, a single-tree (XMSS) and a multi-tree variant (XMSS^MT) of XMSS. Both variants use WOTS+ as a main building block. XMSS provides cryptographic digital signatures without relying on the conjectured hardness of mathematical problems. Instead, it is proven that it only relies on the properties of cryptographic hash functions. XMSS provides strong security guarantees and is even secure when the collision resistance of the underlying hash function is broken. It is suitable for compact implementations, relatively simple to implement, and naturally resists side-channel attacks. Unlike most other signature systems, hash-based signatures withstand attacks using quantum computers.

POST-QUANTUM CRYPTO STANDARDIZATION

Q: What are NIST's plans regarding stateful hash-based signatures?

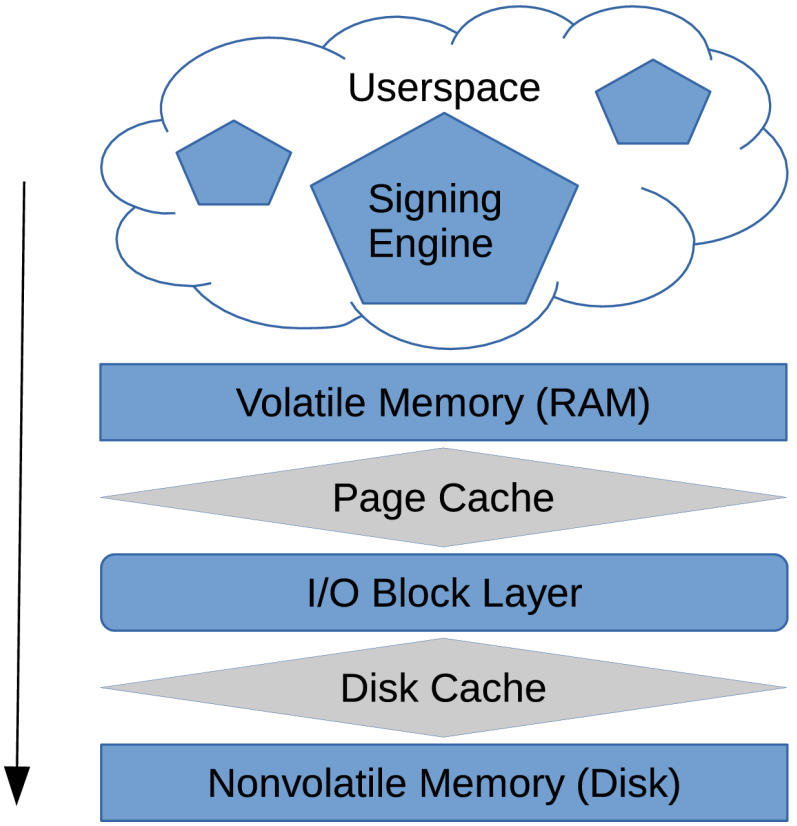
A: NIST plans to coordinate with other standards organizations, such as the IETF, to develop standards for stateful hash-based signatures. As stateful hash-based signatures do not meet the API requested for signatures, this standardization effort will be a separate process from the one outlined in the call for proposals. It is expected that NIST will only approve a stateful hash-based signature standard for use in a limited range of signature applications, such as code signing, where most implementations will be able to securely deal with the requirement to keep state.

How can we cope with statefulness?

State Synchronization

- Synchronization delay affects performance
- Synchronization failure may occur
- Several copies may exist

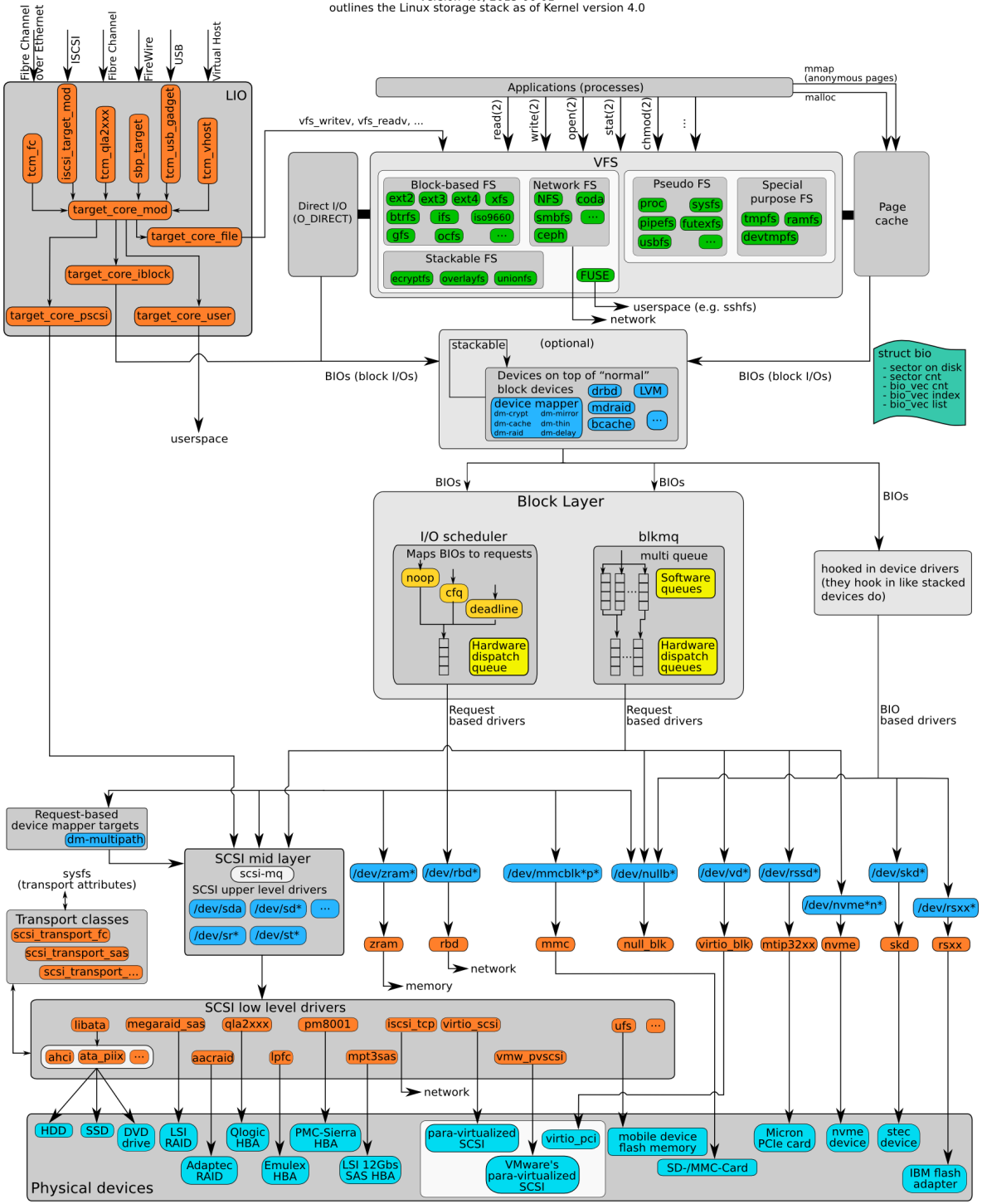
  => Special case of cloning



The Linux Storage Stack Diagram

version 4.0, 2015-06-01

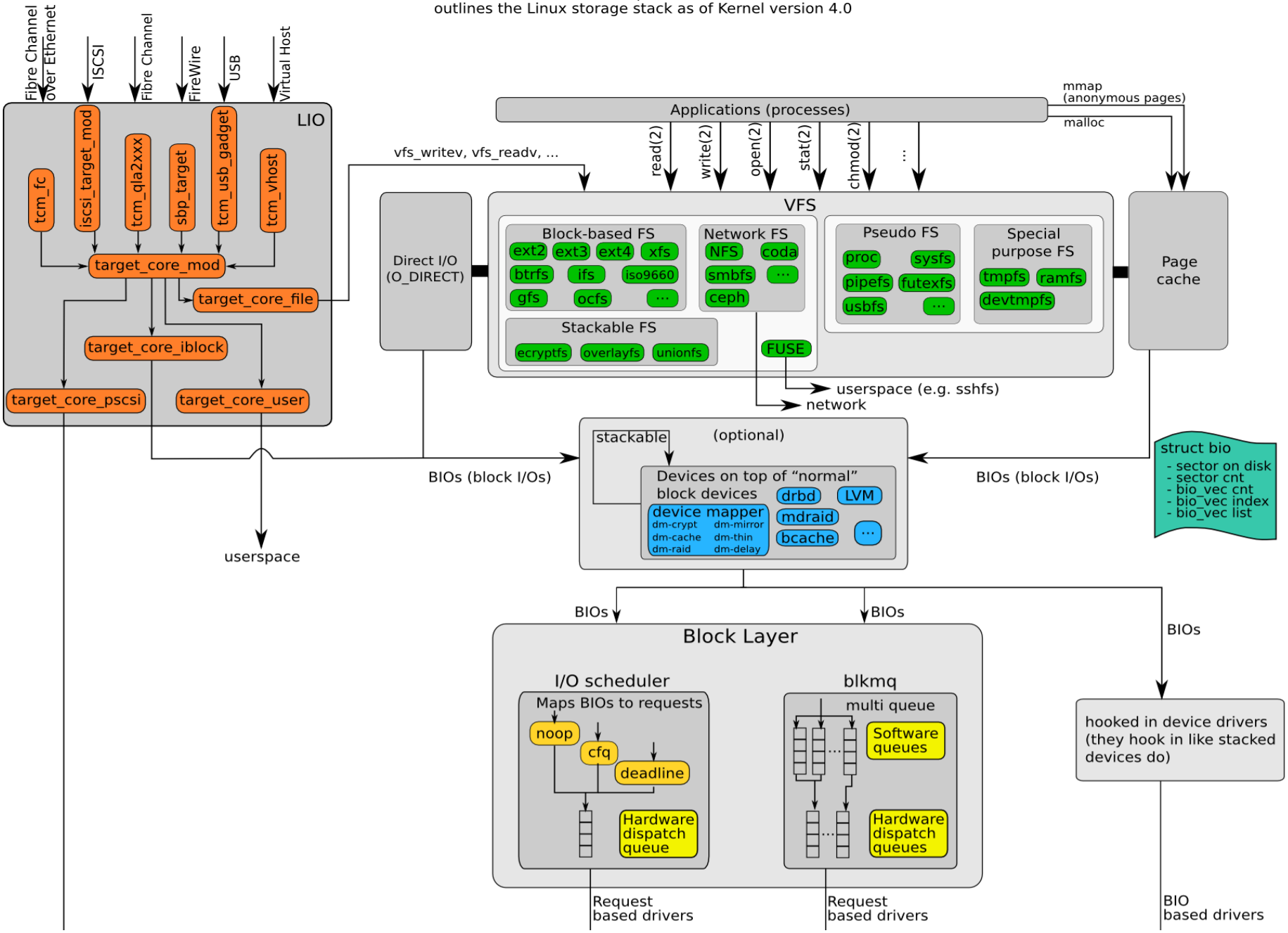
outlines the Linux storage stack as of Kernel version 4.0



The Linux Storage Stack Diagram

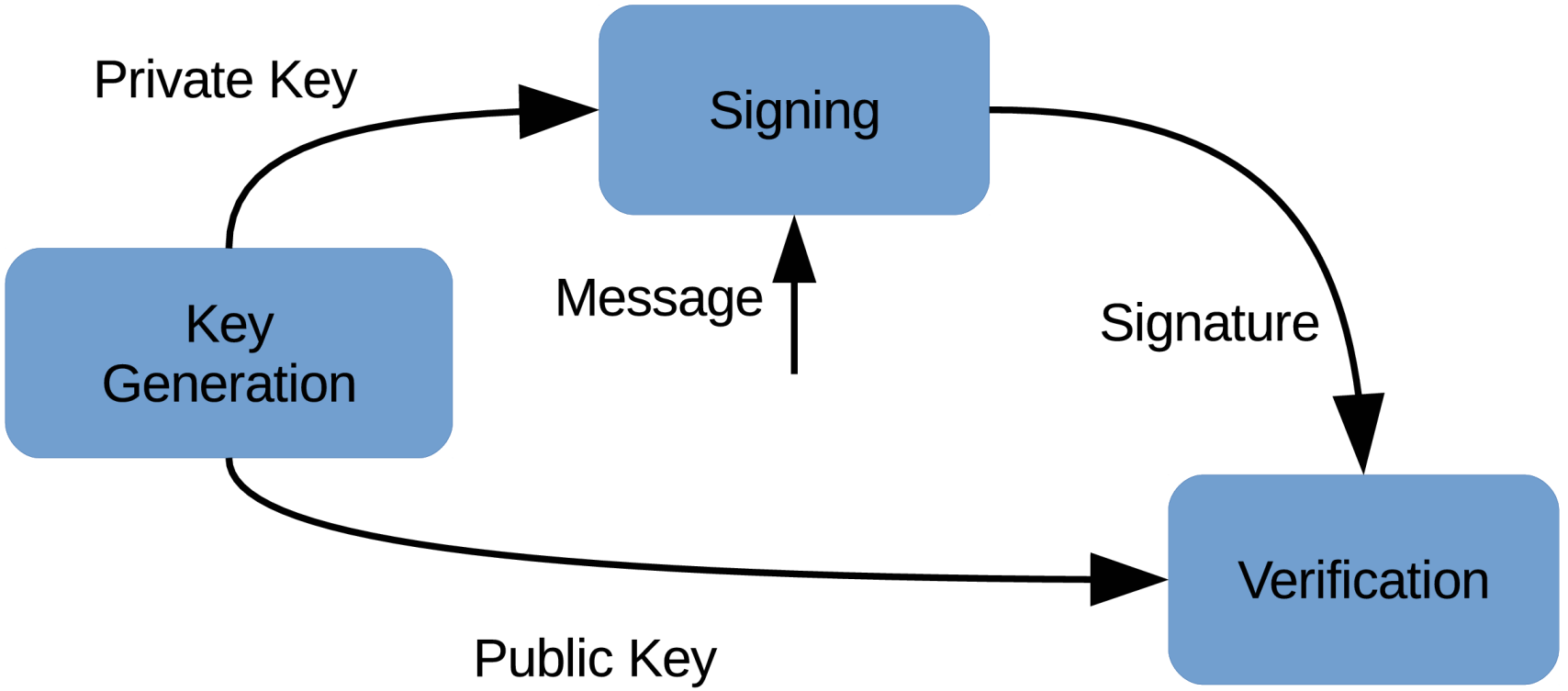
version 4.0, 2015-06-01

outlines the Linux storage stack as of Kernel version 4.0



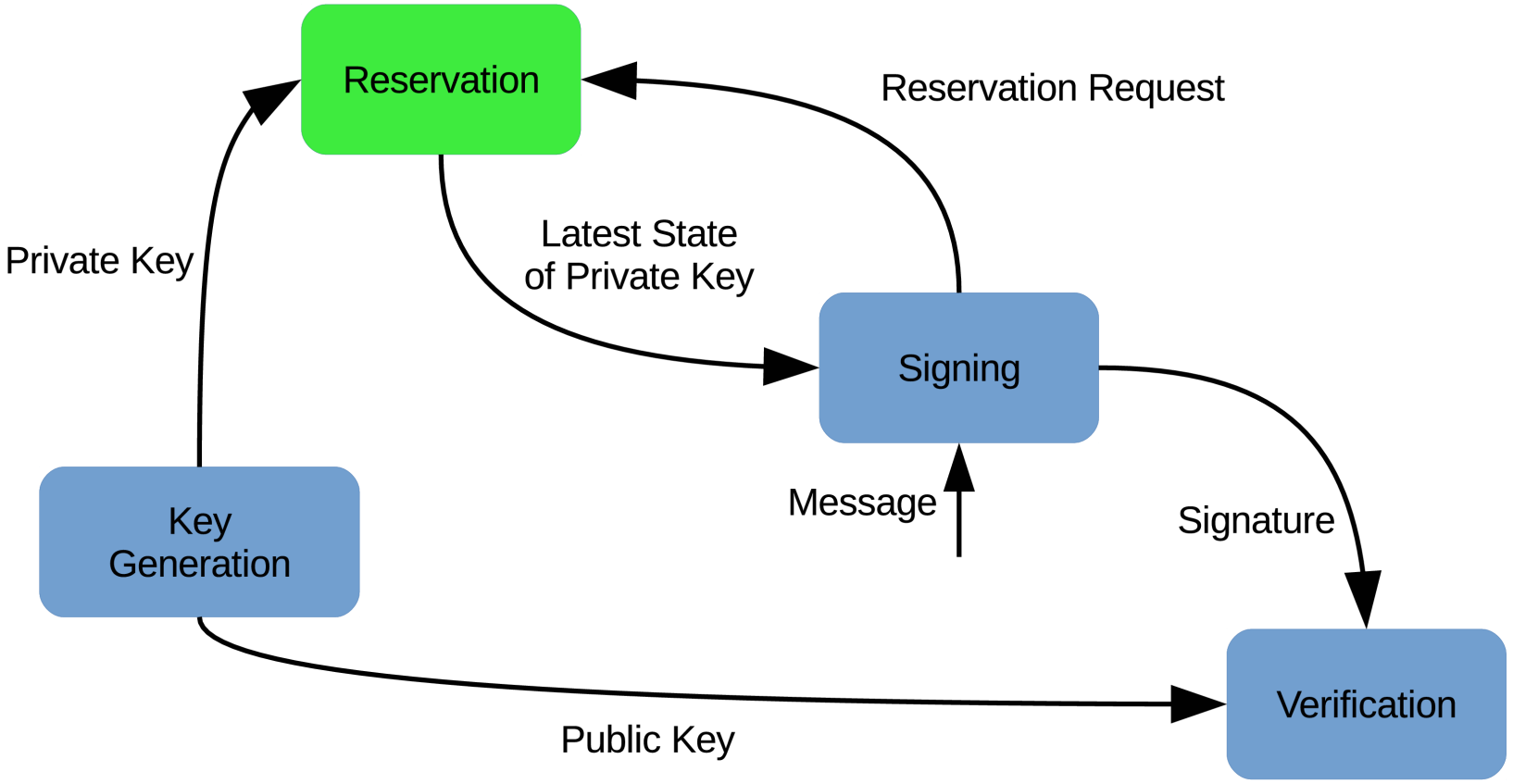
A classic digital signature

Scheme = (Key Generation, Signing, Verification)



A stateful digital signature

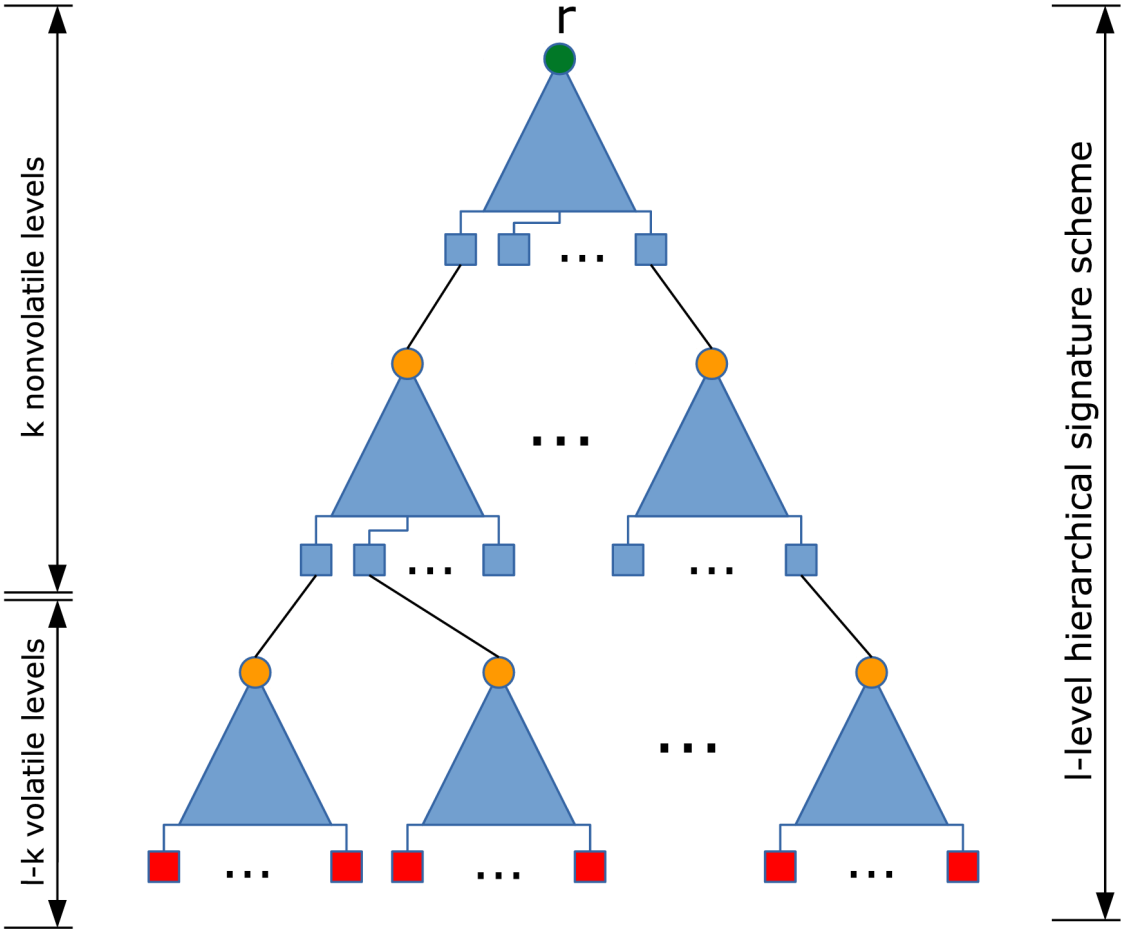
Scheme = (Key Generation, **Reservation**, Signing, Verification)



Reservation

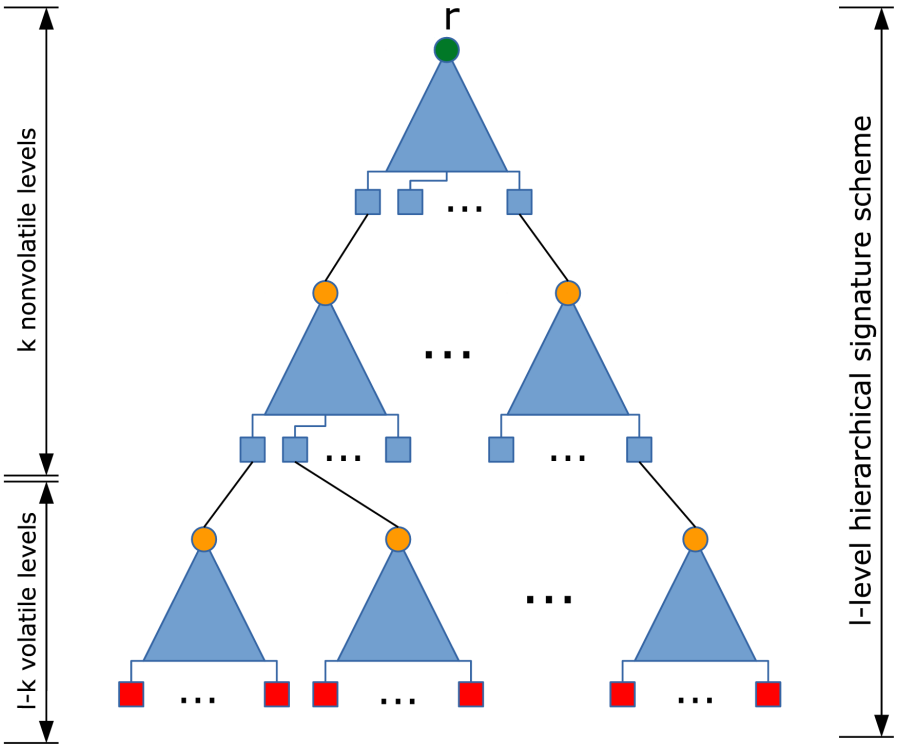
- Keys (pre-) generated in bulk
- Easy access management to critical resource
- Key synchronization and read/write operations alleviated
- Use case specific key pool feasible

Hierarchical Signatures / Key Reservation



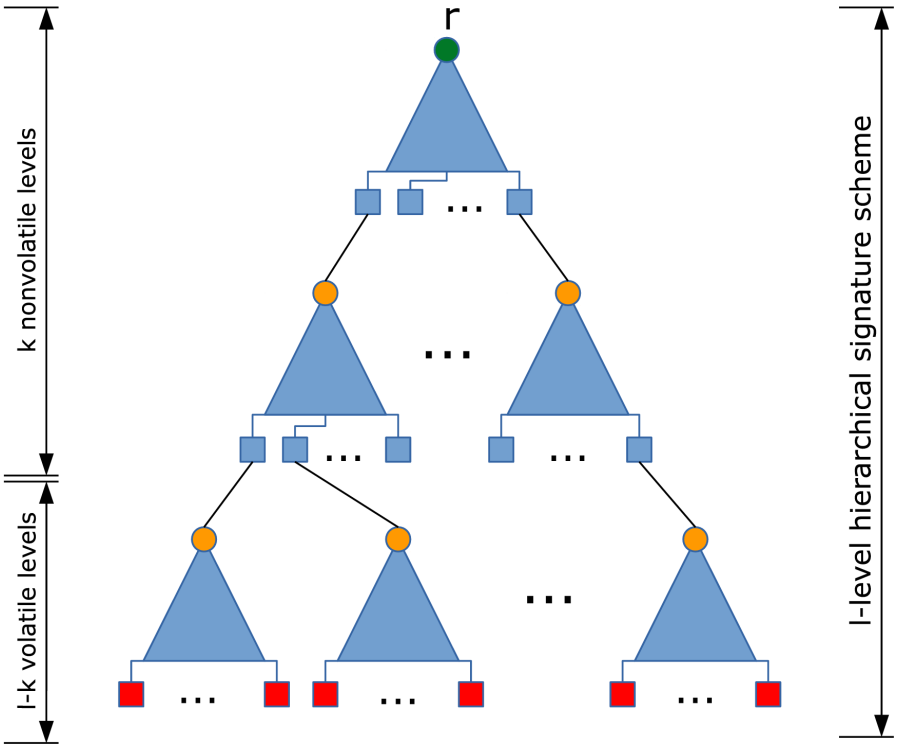
Hierarchical Signatures / Key Reservation

- Synchronization delay
- Synchronization failure
- Unintended cloning
  - Nonvolatile
  - Volatile

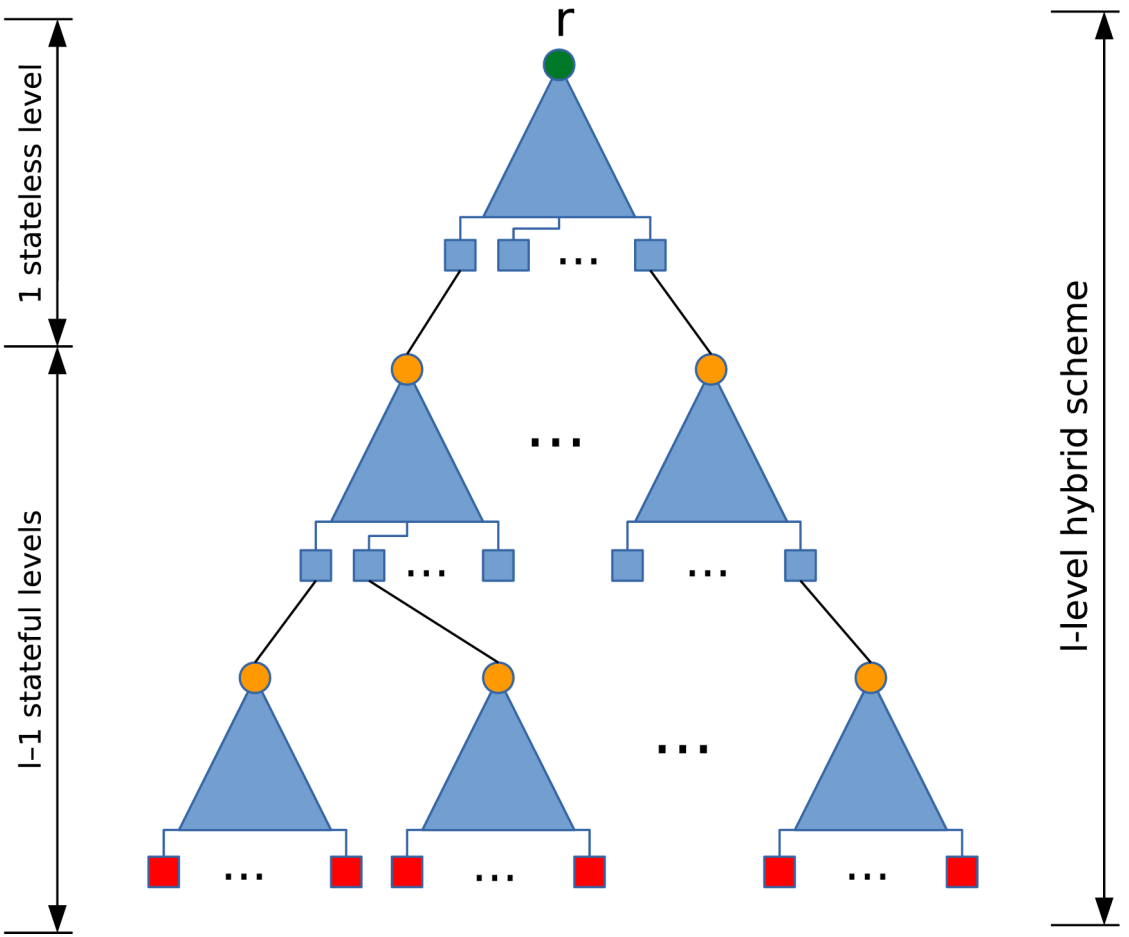


Hierarchical Signatures / Key Reservation

- Synchronization delay ✓
- Synchronization failure ✓
- Unintended cloning ✗
  - Nonvolatile
  - Volatile

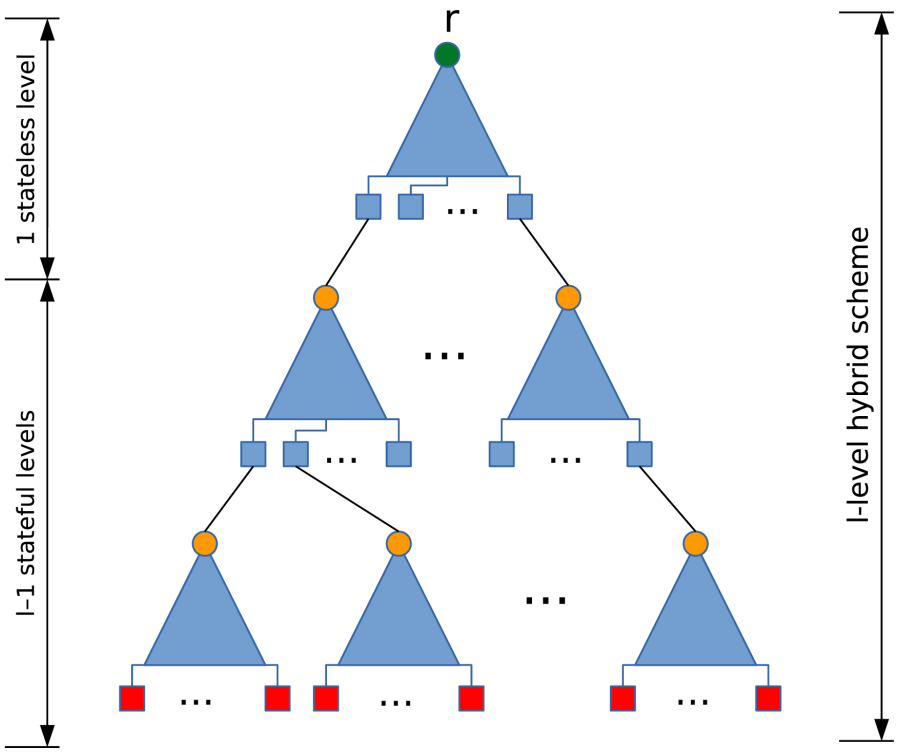


Hybrid Scheme and Reservation



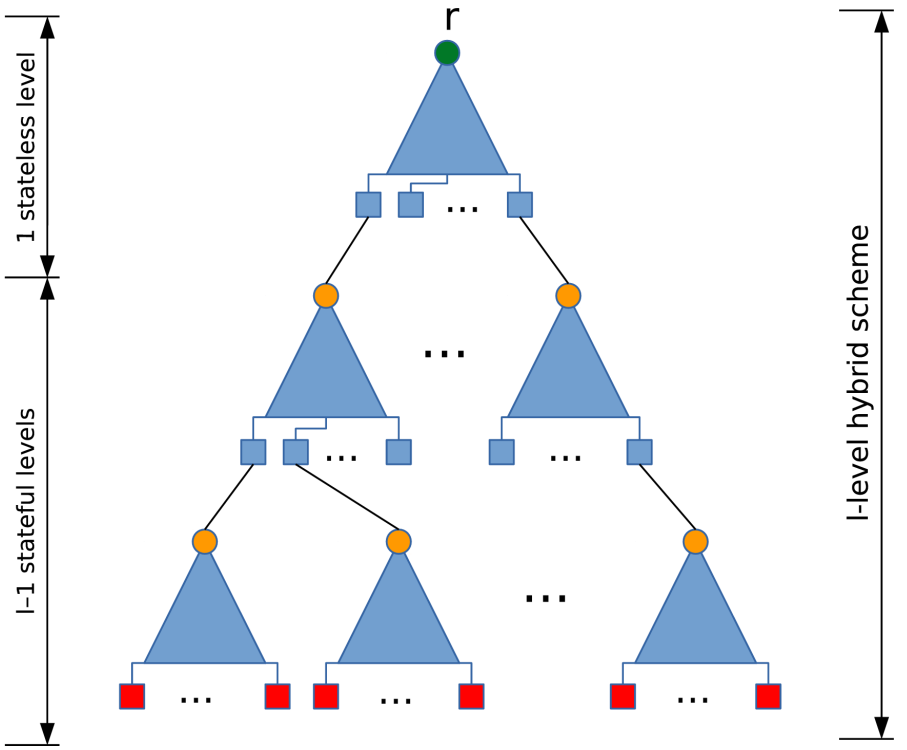
Hybrid Scheme and Reservation

- Synchronization delay
- Synchronization failure
- Unintended cloning
  - Nonvolatile
  - Volatile



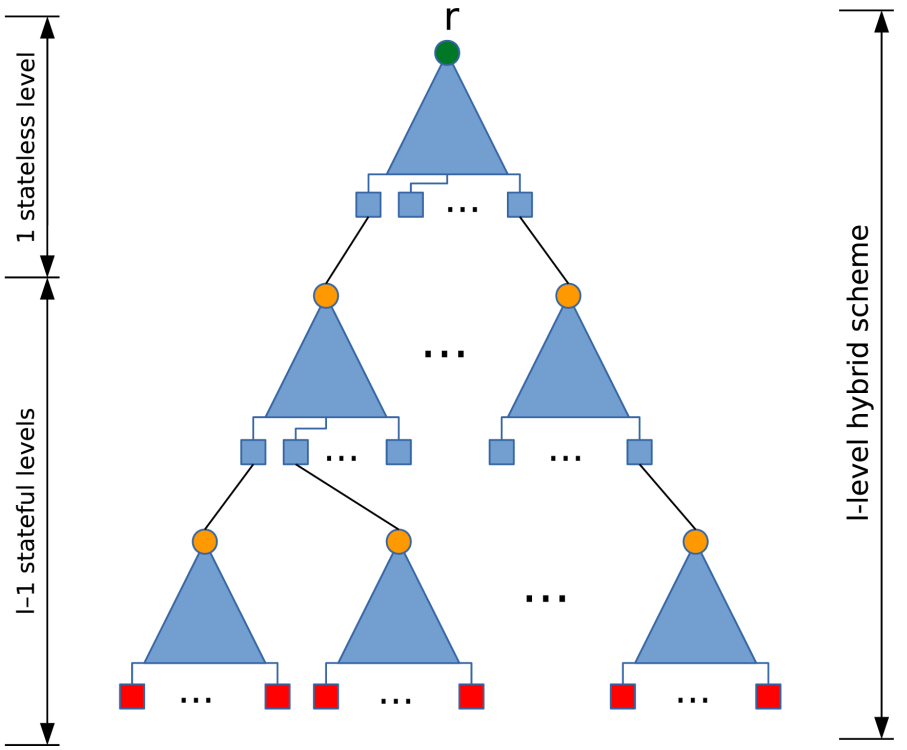
Hybrid Scheme and Reservation

- Synchronization delay ✓
- Synchronization failure ✓
- Unintended cloning
  - Nonvolatile ✓
  - Volatile ✗



Hybrid Scheme and Reservation

- Synchronization delay ✓
- Synchronization failure ✓
- Unintended cloning
  - Nonvolatile ✓
  - Volatile ✗?



Hybrid Scheme and Reservation

- Synchronization delay ✓
- Synchronization failure ✓
- Unintended cloning
  - Nonvolatile ✓
  - Volatile ✗

Breaks so much more:

- Entropy pools and PRNGs
- Deterministic IVs and Nonces
- Encryption counters
- Digital signature seeds
- One Time Passwords (OTP)
- TCP sequence numbers
- ...

Conclusion

- First official standards available soon
- Safe deployment / good performance feasible
- Future work:

  standardization document on HBS deployment

Any questions?

{mcgrew,pkampana,sfluhrer}@cisco.com
stefan-lukas_gazdag@genua.eu
{dbutin,buchmann}@cdc.informatik.tu-darmstadt.de