Property-Based Attestation without a Trusted Third Party

Liqun Chen (HP Labs, Bristol),

Hans Löhr (Ruhr-Uni Bochum),

Mark Manulis (Crypto Group UC Louvain),

Ahmad-Reza Sadeghi (Ruhr-Uni Bochum)

Chair for System Security
Horst Görtz Institute for IT Security
Ruhr-Universität Bochum, Germany

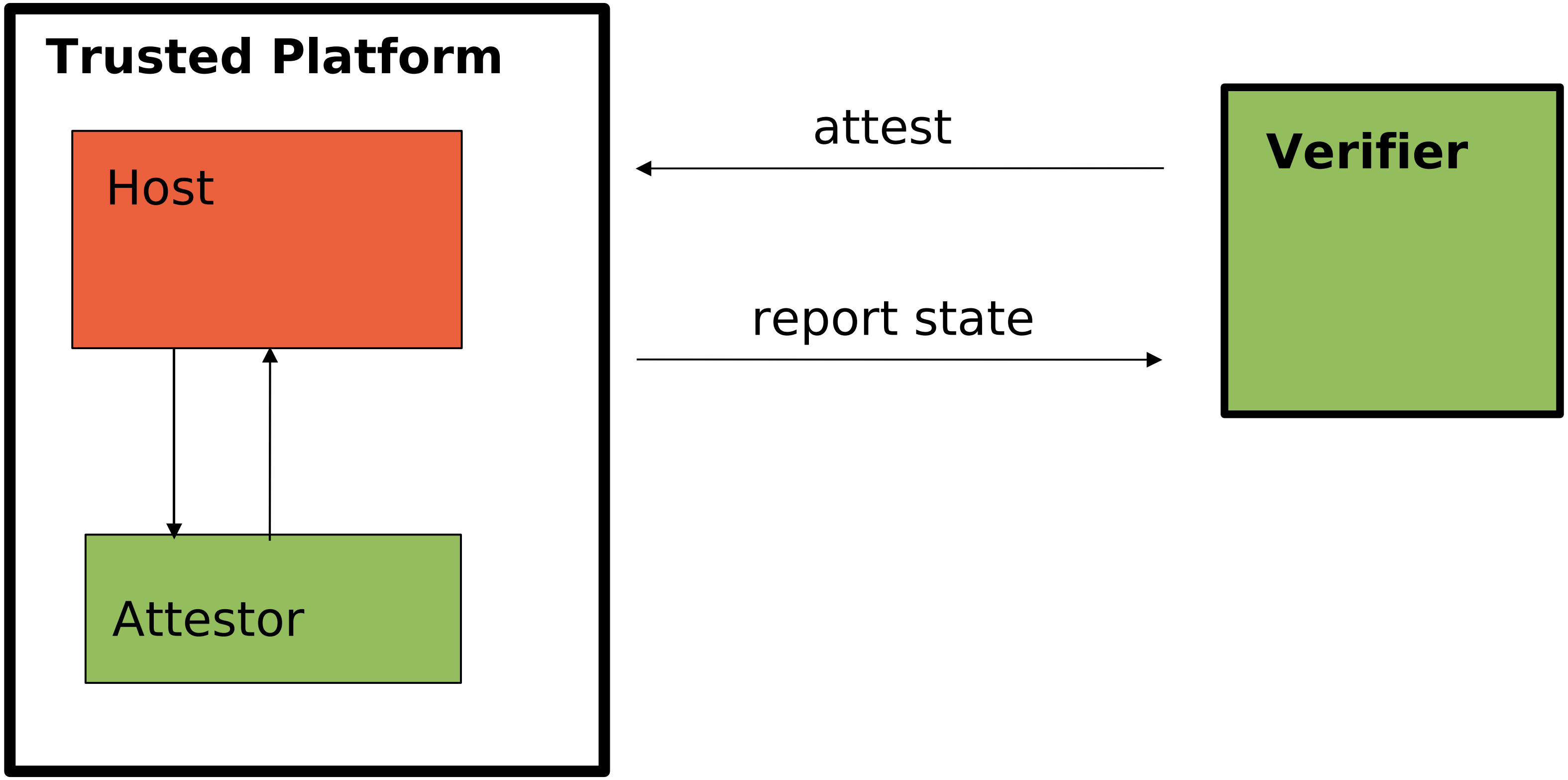
Information Security Conference (ISC) 2008

September 16, 2008, Taipei, Taiwan

Outline

- Introduction (Property-Based) Attestation
- Our Approach
- **(**In the paper: Formalization / Proof**)**
- Conclusions

Attestation (Overview)



Trusted Platform Module (TPM)

- **Trusted Computing Group (TCG)**
  - Industrial consortium, publishes specifications
- **Trusted Platform Module (TPM)**
  - Hardware security module (completely trusted)
  - Functionality:
    - Digital dignatures, en-/decryption
    - Random number generation, key generation
    - Cryptographic hash function (currently SHA-1)
    - Non-volatile memory, key storage, registers
    - ...

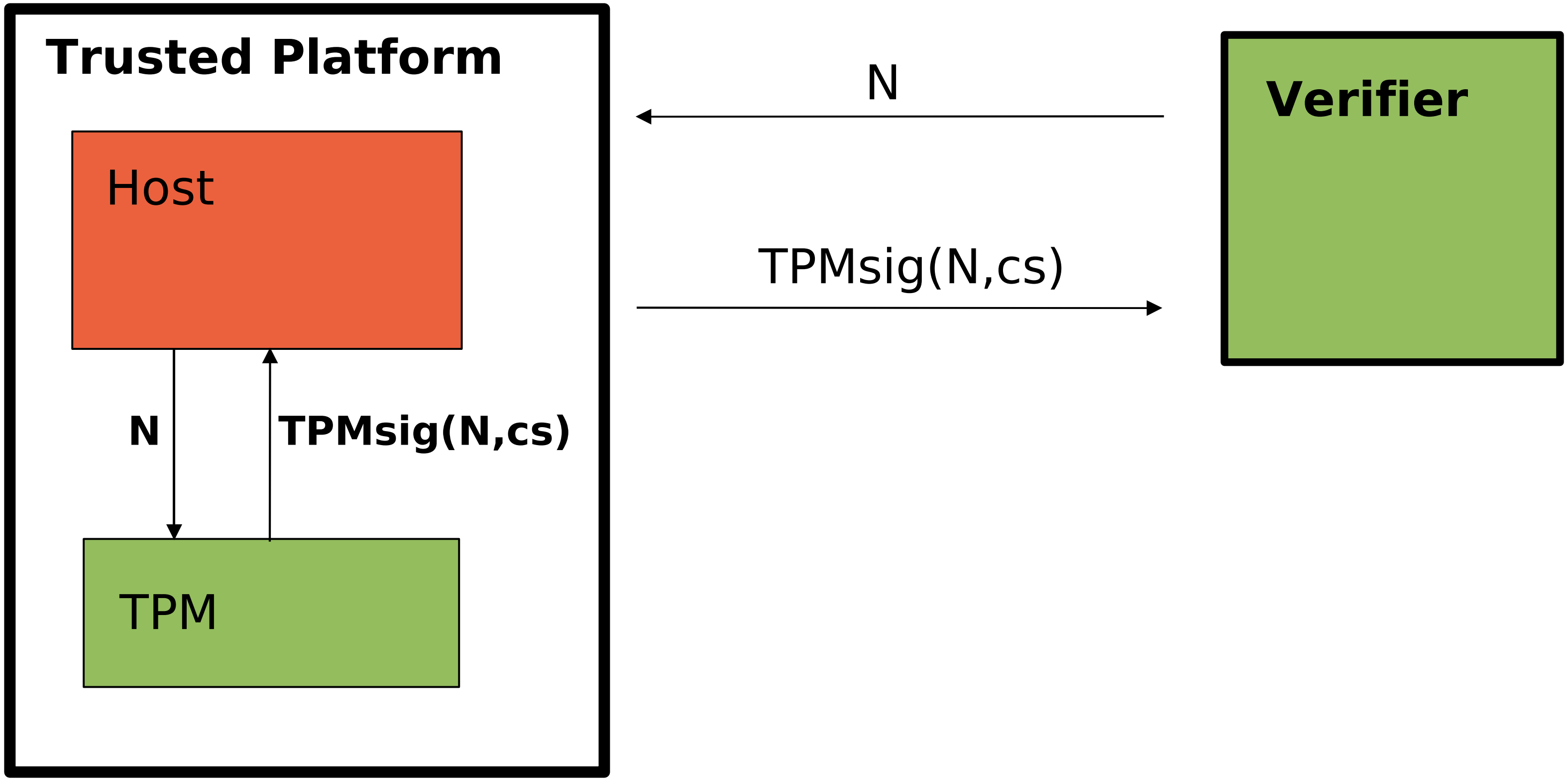
Authenticated Boot (simplified)

- **Goal:**
  - “Platform configuration” stored inside the TPM, in *platform configuration registers (PCRs)*
- **Boot process:**
  - Hash value of all components is written to PCR
  - PCRs can only be “extended”: $(PCR_0 := 0)$

$$PCR_{t+1} := \text{hash}(\ PCR_t \mid \text{hash(component)}\ )$$

  - Each component hashes next one that is started
- **Result (after boot):**
  - PCR contains accumulated hash of system components: *configuration*

TCG Attestation (simplified)



Nonce N: anti-replay value

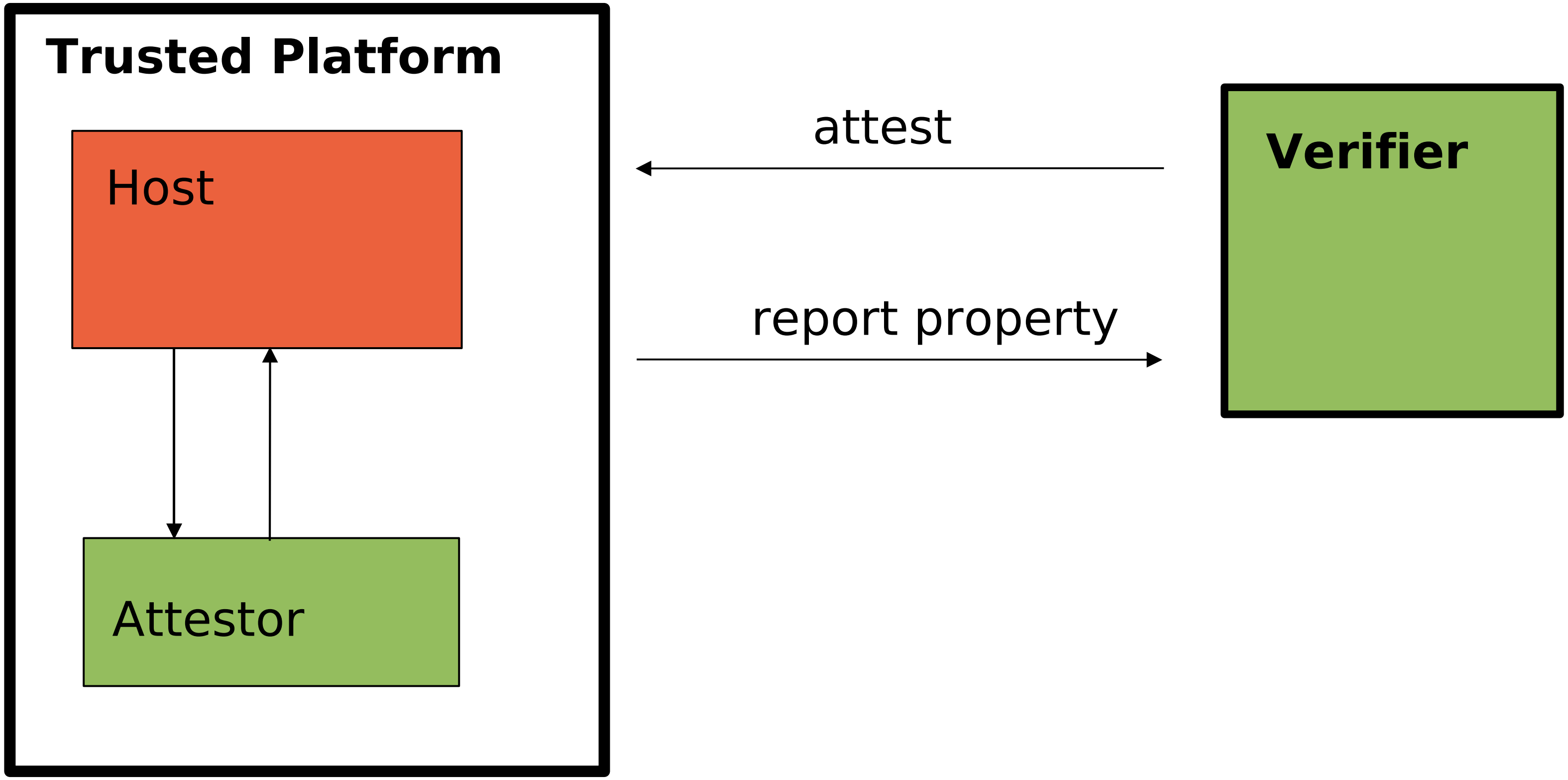
Configuration specification cs: hash value

Deficiencies (TCG Attestation)

- Privacy
  - Potential price discrimination
  - Disclosure of vulnerabilities
- Scalability
  - Binary hash values hard to manage
  - Minor change leads to different hash

Verifier is interested in properties (not exact configurations)

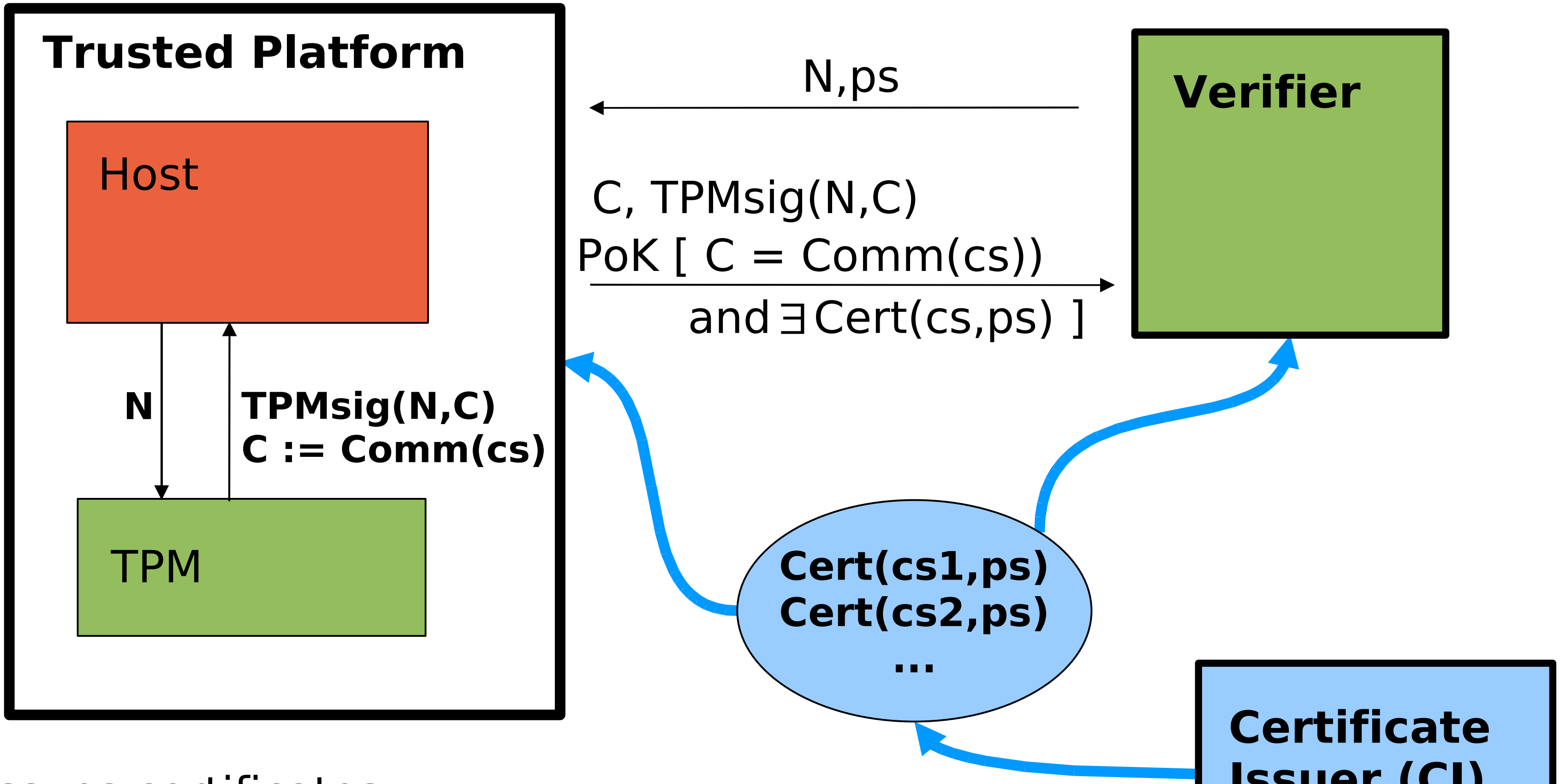
Property-Based Attestation (PBA)



Security Requirements

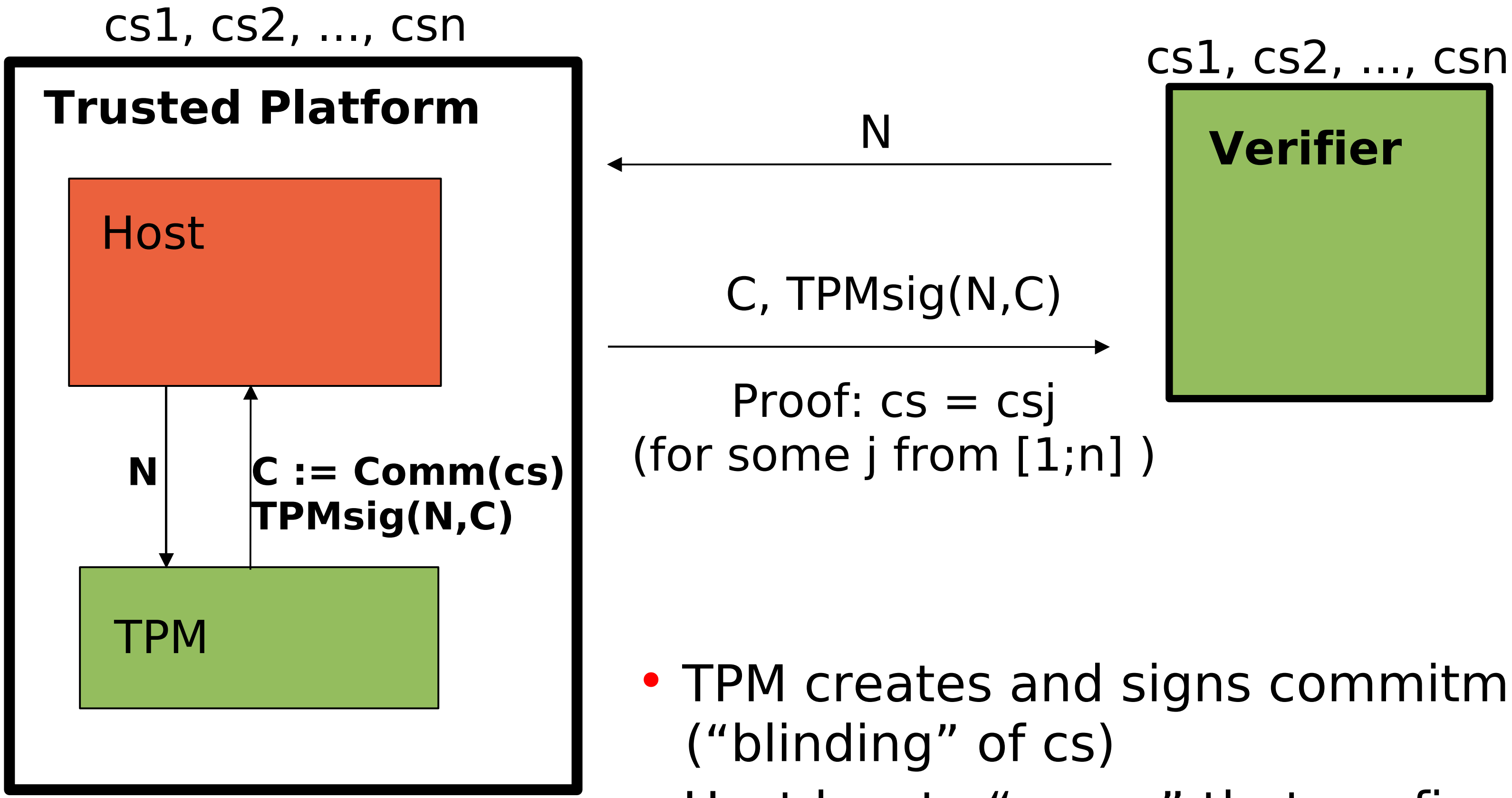
- **Evidence Authentication (informal):**
  - **Adversary (prover) must not be able to “forge” attestation (report wrong property)**
- **Configuration Privacy (informal):**
  - **Adversary (verifier) must not be able to determine configuration (probability not better than guessing)**
- **“Games” to formalize requirements**
  - **For cryptographic proofs (see paper)**

Delegation-Based PBA



- CI issues certificates:
  configs cs1, cs2, ... fulfill property ps
- TPM creates and signs commitment C (“blinding” of cs)
- Host performs zero-knowledge Proof of Knowledge (PoK)

PBA without Trusted Third Party



- TPM creates and signs commitment C (“blinding” of cs)
- Host has to “prove” that config “inside” C is from the list cs1, …, csn
- Index j is kept private
- How is the list cs1, …, csn negociated?

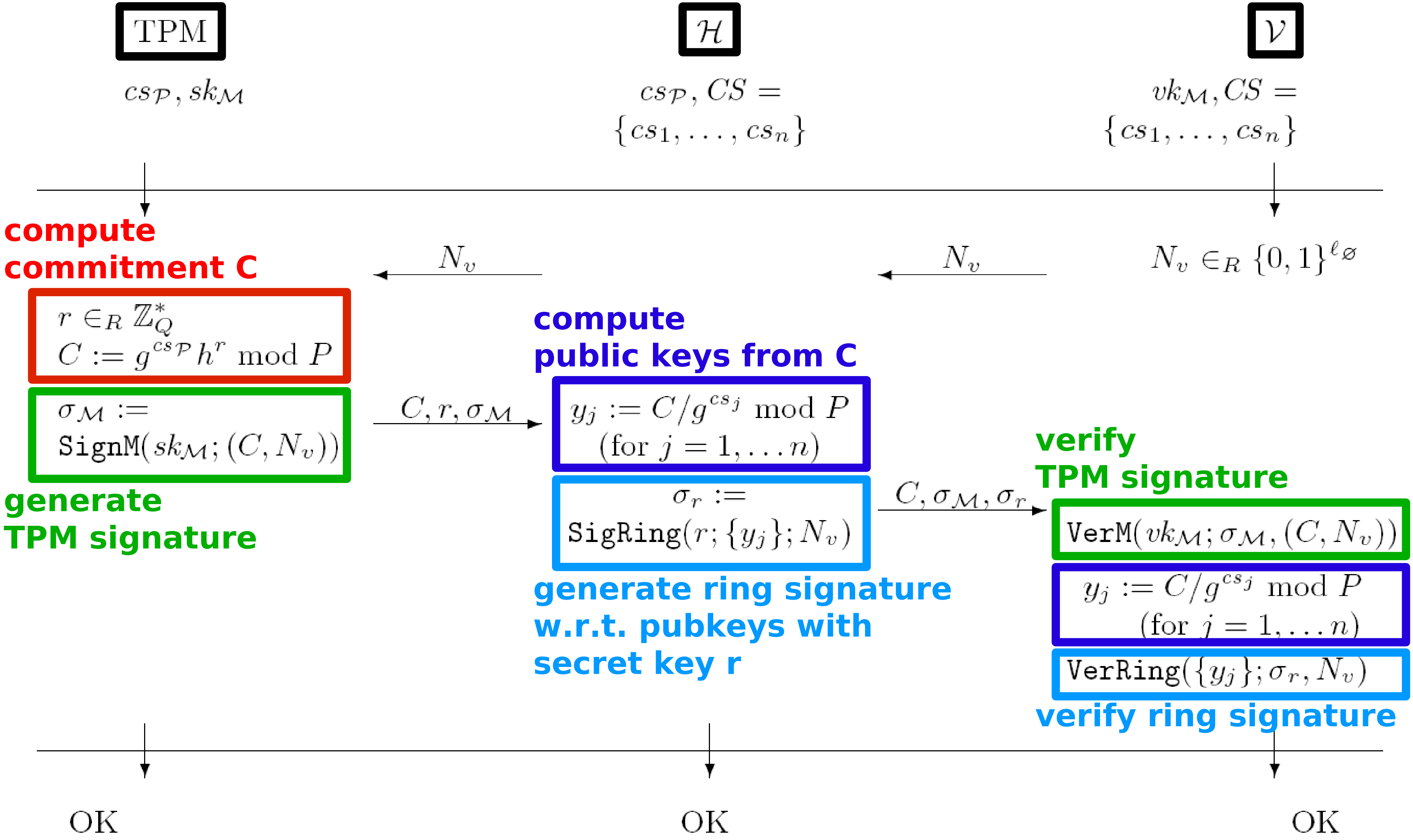
Realization with Ring Signatures

Idea: realize “proof cs=csj” with ring signature

- Ring signatures (abstract / simplified):
  - Public keys: PK1, ..., PKn
  - Signer who knows SKi (for PKi) can sign m:
    ringsig(m, (PK1,...,PKn), SKi)
  - Verifier can verify that signer knows one secret key matching one of the public keys, but not which one.
- Re-use existing ring sig scheme [AOS02]

Common input: g, h, P, Q

Full Protocol



TPM

$cs_{\mathcal{P}}, sk_{\mathcal{M}}$

$\mathcal{H}$

$cs_{\mathcal{P}}, CS = \{cs_1, \ldots, cs_n\}$

$\mathcal{V}$

$vk_{\mathcal{M}}, CS = \{cs_1, \ldots, cs_n\}$

$N_v \in_R \{0,1\}^{\ell_\varnothing}$

N_v

N_v

compute commitment C

$r \in_R \mathbb{Z}_Q^*$

$C := g^{cs_{\mathcal{P}}} h^r \mod P$

$\sigma_{\mathcal{M}} := \texttt{SignM}(sk_{\mathcal{M}}; (C, N_v))$

generate TPM signature

$C, r, \sigma_{\mathcal{M}}$

compute public keys from C

$y_j := C / g^{cs_j} \mod P$ (for $j = 1, \ldots n$)

$\sigma_r := \texttt{SigRing}(r; \{y_j\}; N_v)$

generate ring signature w.r.t. pubkeys with secret key r

$C, \sigma_{\mathcal{M}}, \sigma_r$

verify TPM signature

$\texttt{VerM}(vk_{\mathcal{M}}; \sigma_{\mathcal{M}}, (C, N_v))$

$y_j := C / g^{cs_j} \mod P$ (for $j = 1, \ldots n$)

$\texttt{VerRing}(\{y_j\}; \sigma_r, N_v)$

verify ring signature

OK

OK

OK

Security

Rough overview:

o **Evidence Authentication:**

- **Security of TPM sig. and commitment**
- **Security of ring signature**
- **=> Reduce to discrete log**

o **Configuration Privacy:**

- **Anonymity of ring signature**
- **Hiding property of commitment**
- **=> A's success probability not better than guessing**

Conclusions / Open Questions

- **New property-based attestation protocol, without a Trusted Third Party**
  - Generalizes existing protocols
  - Formalization of security requirements
  - Provably secure
- Not directly implementable on current TPMs
  - TPM supports all necessary operations
  - No command for “signed commitment”
- What are meaningful properties?
- How can such properties be “extracted”?

Some Related Work

- **[SS04]: Concept of PBA, classification, high-level solutions**
- **[PSVW04]: PBA with ”verification proxy”**
- **[HCF04]: ”Semantic remote attestation” (based on trusted VMs)**
- **[CLL+06]: Crypto protocol for delegation-based PBA**
- **[KSS07]: PBA (+ sealing) by hashing public keys of property certificates**