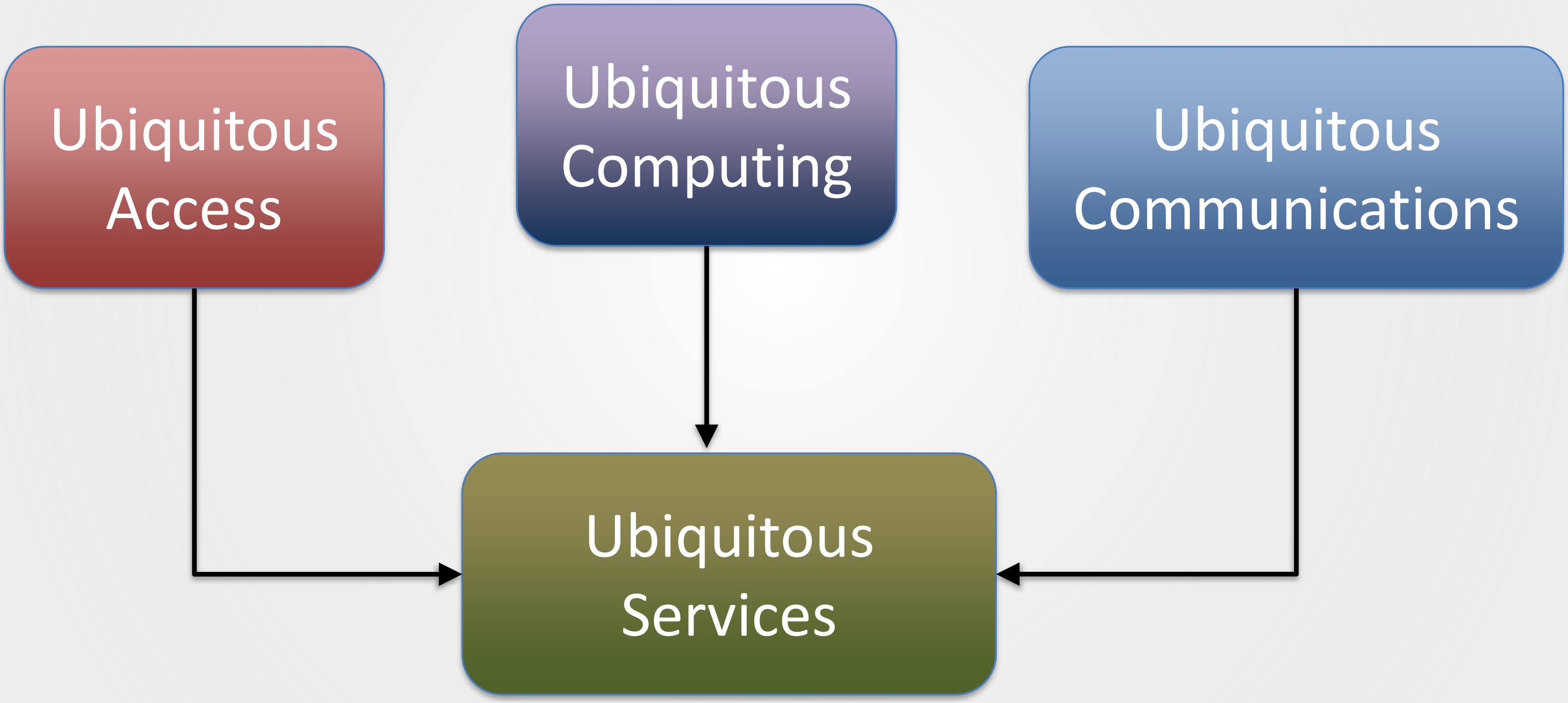
Privacy-preserving deep attestation

Cristina ONETE (maria-cristina.onete@unilim.fr)

Joint work with : G. Arfaoui, T. Jacques, M. Lacoste, A. Nedelcu, P.-A.-Fouque, P. Lafourcade, L. Robert

Anytime, anywhere

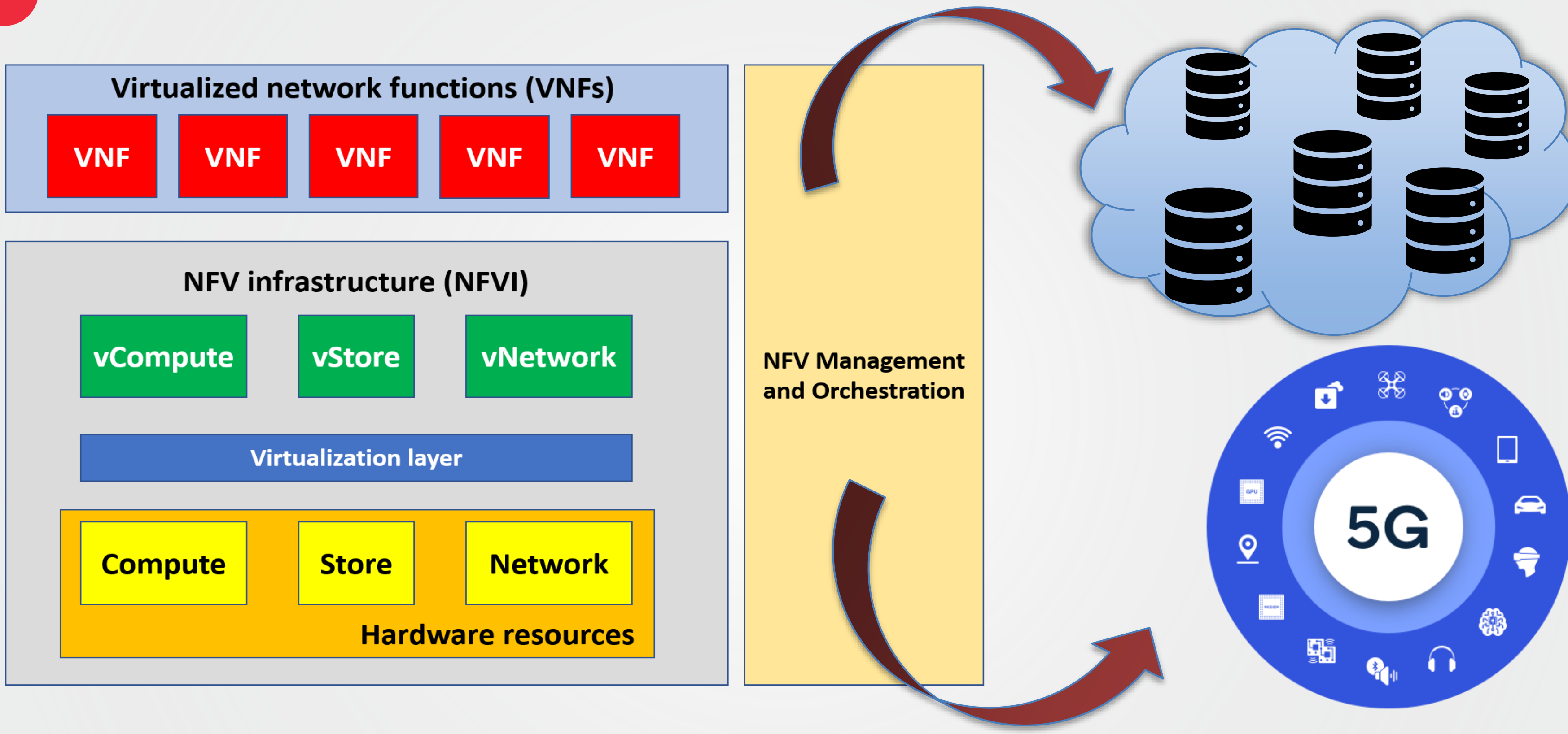


Ubiquitous architectures

- Ubiquity relies on :
  - Plentiful, (potentially-shared) resources
  - Repetition
  - Delegation
  - (Remote) reconfigurability
  - Orchestration of resources

- Ubiquity often provided as a service
  - … by a potentially semi-trusted provider
  - … or even a plurality of such providers

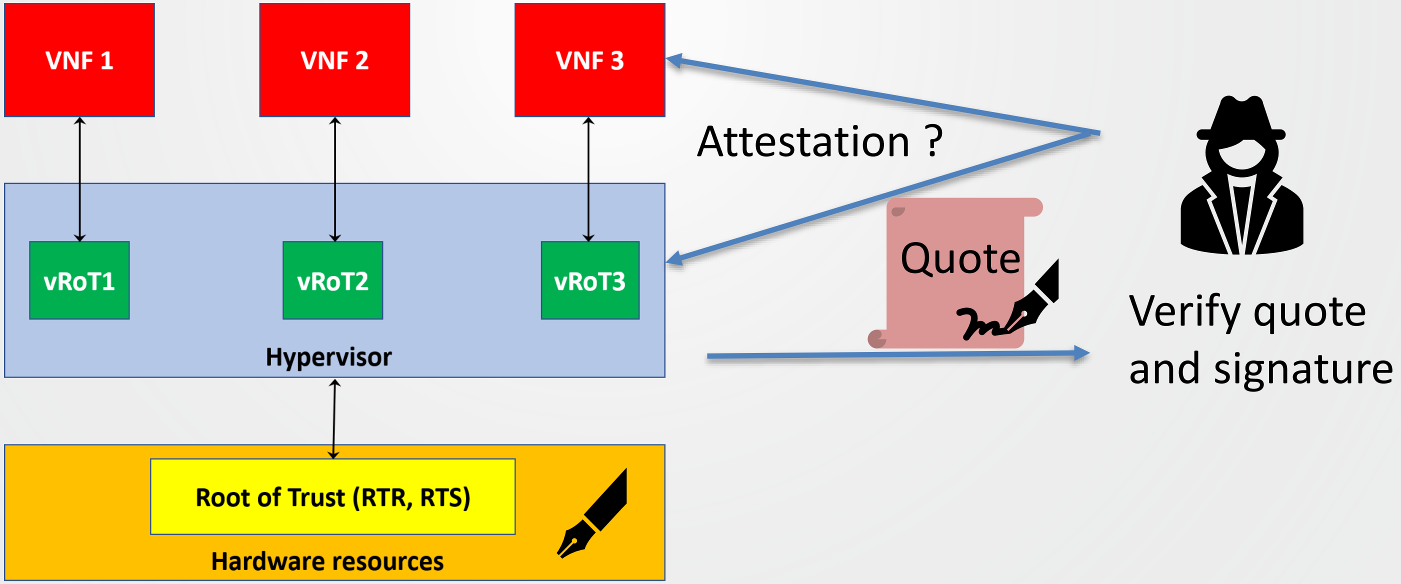
Virtualization



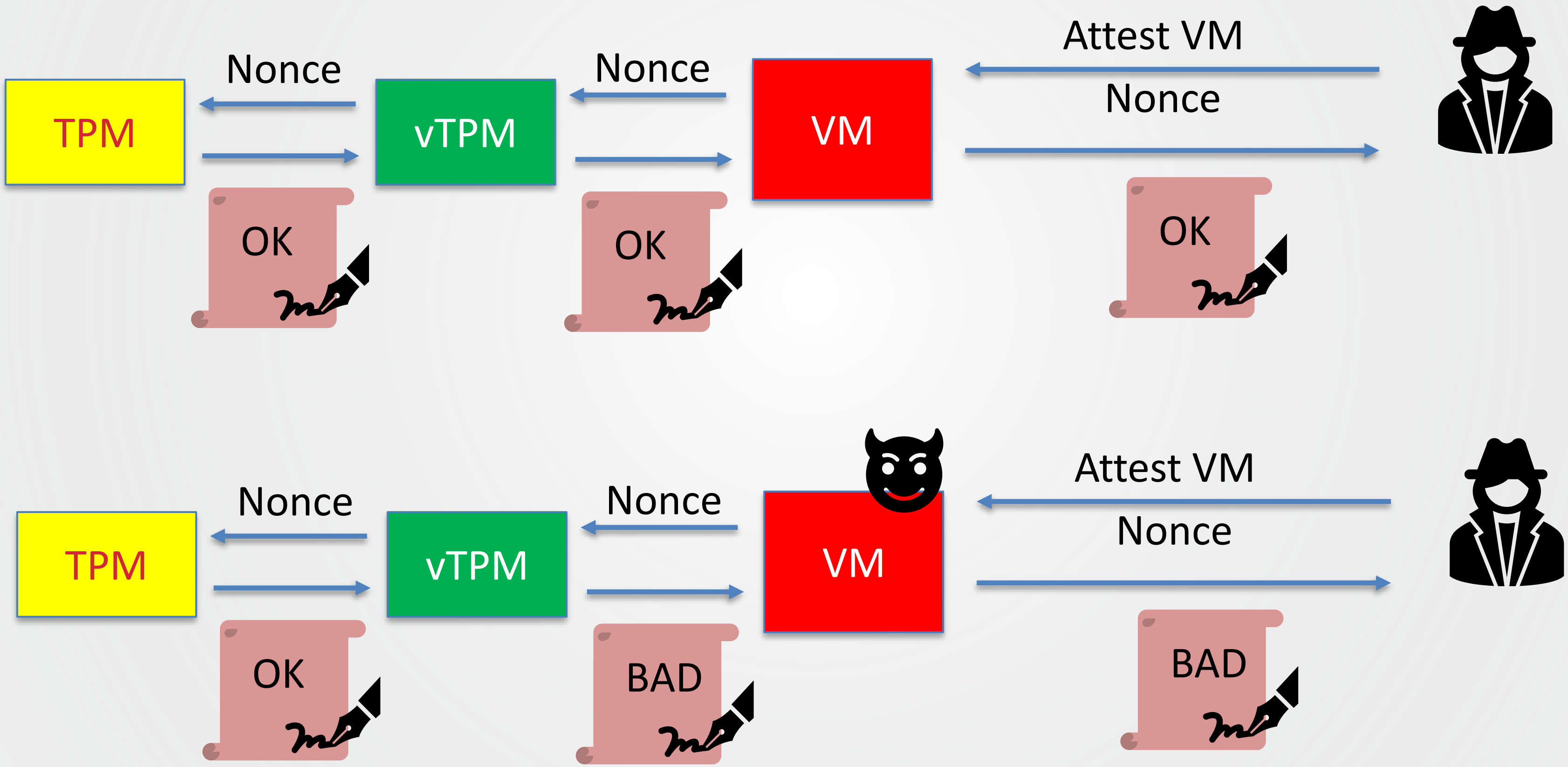
Efficient, practical, ubiquitous. Do we trust it ?

Deep attestation

- External verifier assesses boot state of virtual elements
  - VMs, hypervisors
  - Requires a “Root of Trust” and “Root of Storage”
  - Hash over values of some registers signed by TPM for fresh nonce



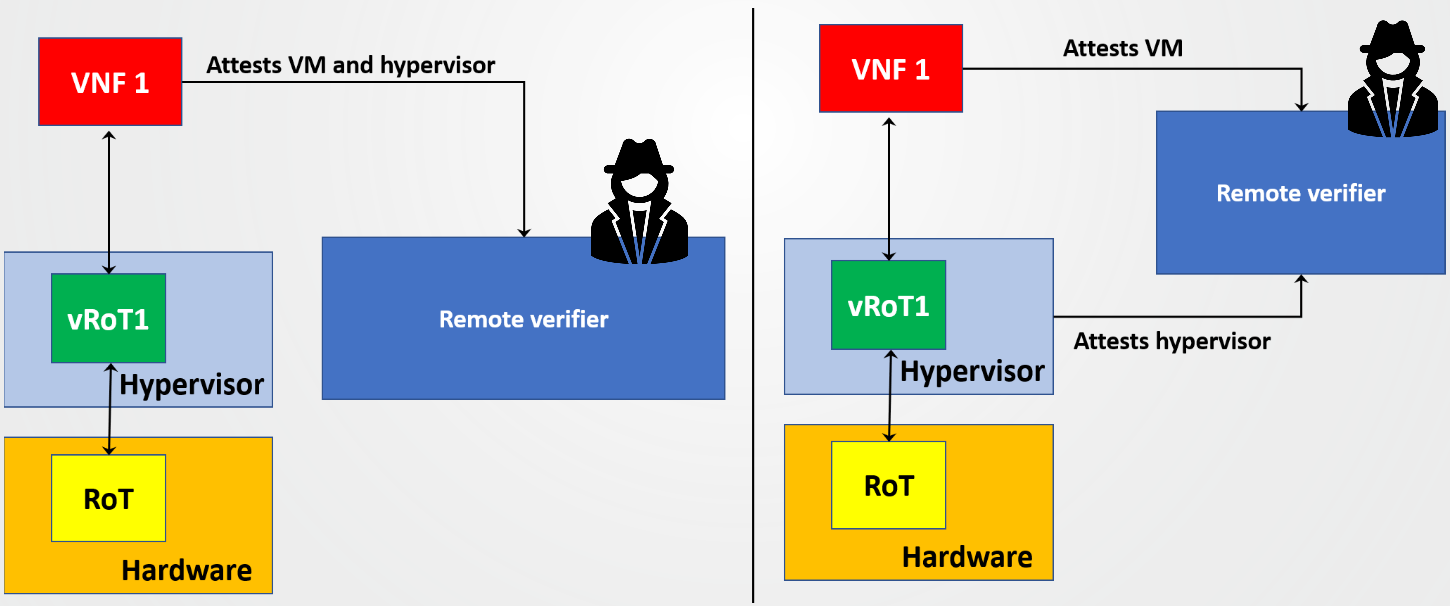
Intuition


TPM
Nonce
vTPM
Nonce
VM
Attest VM
Nonce
OK
OK
OK
TPM
Nonce
vTPM
Nonce
VM
Attest VM
Nonce
OK
BAD
BAD

Single/multiple channel DA

- ETSI : two ways to do deep attestation :

Single channel DA | **Multiple channel DA**



Single/multiple channel DA

Single channel DA	Multiple channel DA
1 hypervisor attestation for each VM attestation	1 hypervisor attestation
All attestations generated by TPM (slow)	VM attestations go through vTPM (fast)
Layer linking : this VM is managed by this hypervisor	No layer linking : independent attestations
Verification requires knowledge of hypervisor configuration	

Trust, Privacy, Ubiquity

- **Challenge 1**: efficiency vs. trust

  Efficiency | Trust | Single-owner

- **Challenge 2** : efficiency, trust, privacy, multitenancy

  Efficiency | Trust | Privacy | Multitenant, single-owner

- **Challenge 3** : efficiency, trust, ubiquity

  Efficiency | Trust | Privacy | Multitenant, multi-owner

This talk

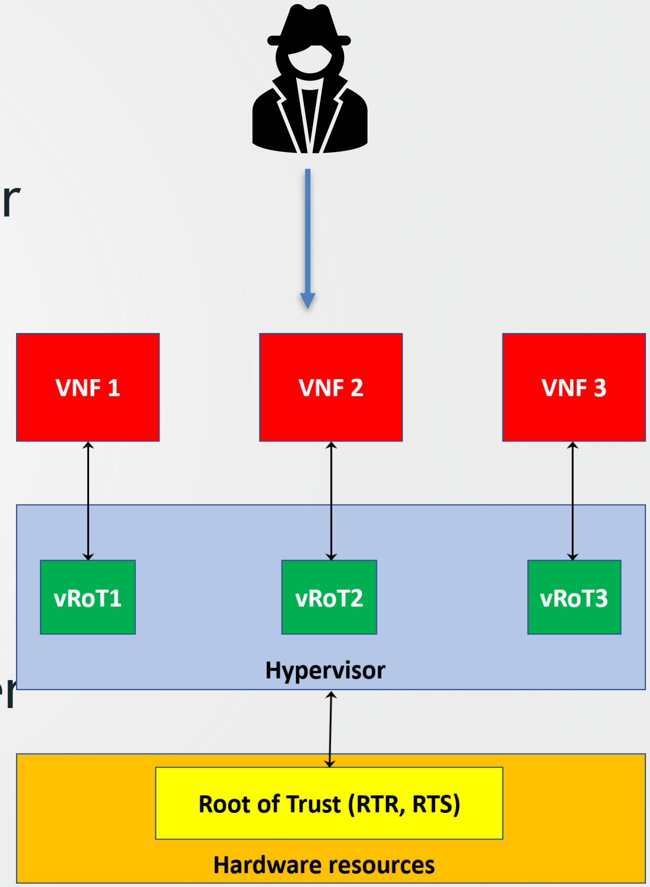
- Challenge 1: efficiency vs. trust
  - Layer-linking : our approach
  - Concrete construction
  - Provable security
- Challenge 2: trust in multitenant architectures
  - Privacy concerns
  - Our approach
- Performance
- Conclusion and Future Challenges

Challenge 1 : trust vs. efficiency

Context

- Simple virtualized architecture:
  - Single owner, single tenant
  - Potentially-external authorized verifier
  - No migration or multiple hypervisors

- Goals :
  - **Trust**: VM & hypervisor attestation
  - **Layer-linking**: hypervisor and VMs
  - **Authorization**: only authorized verifier can see attestation data
  - **Universal**: No modifications to TPM



Towards layer linking

- Single-channel attestation has layer-linking
  - Binding of VM and hypervisor quotes in single response
  - Freshness: attestation nonce
  - Trust: TPM generates quote and signature

Can we achieve binding in an efficient, scalable way?

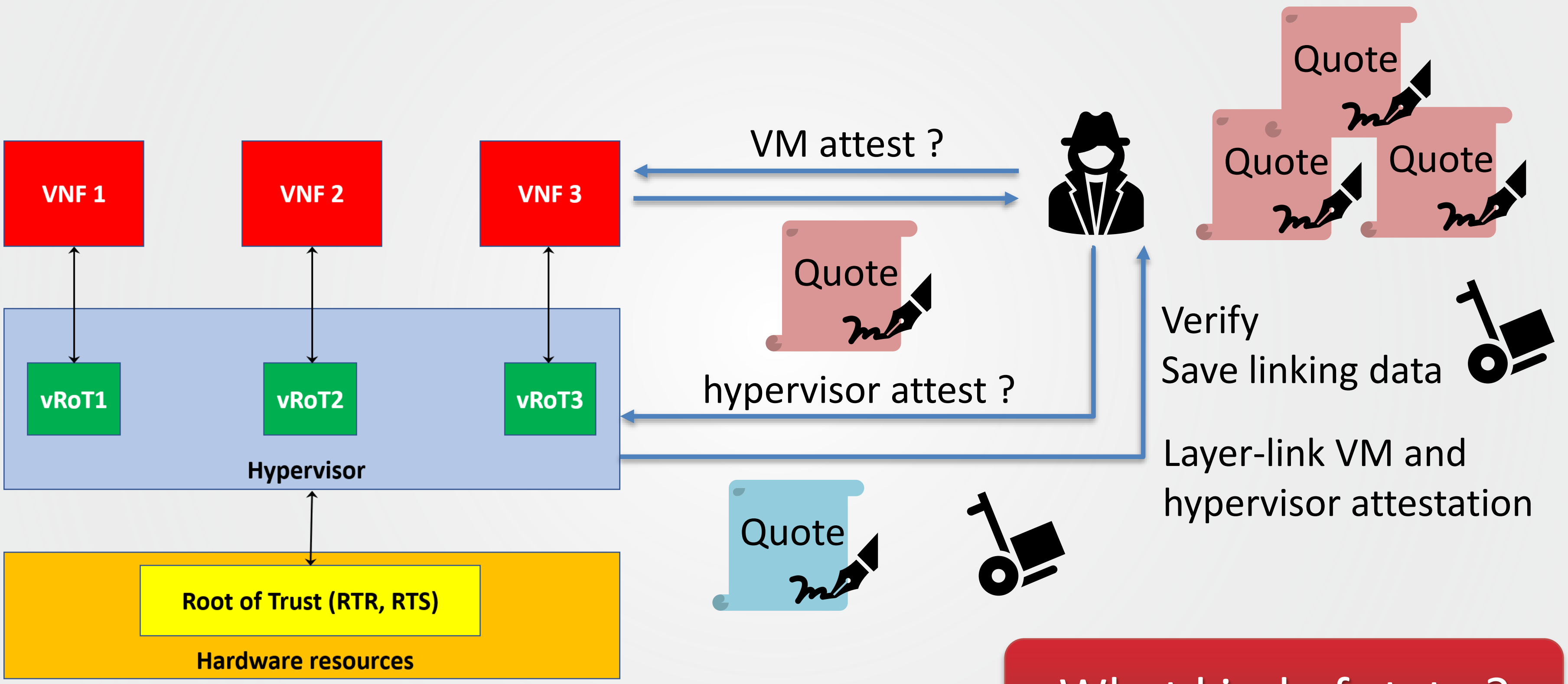
- Idea : use auxiliary information as binding state !

Arfaoui, Fouque, Jacques, Lafourcade, Nedelcu, Onete, Robert :
"A Cryptographic View of Deep-Attestation, or How to Do Provably-Secure Layer-Linking" [ACNS '22]

A non-trivial task

- Linking is a powerful tool :
  - Binding of hypervisor and VM to physical TPM
  - Confirmation of security settings
- Non-trivial to achieve :
  - Infrastructure owner might want to migrate VMs
  - Corruption/compromise of hypervisor is possible
  - Only trusted element is TPM … which is heavily standardized
- Verifier should only trust data authenticated by TPM

Layer-linking with state

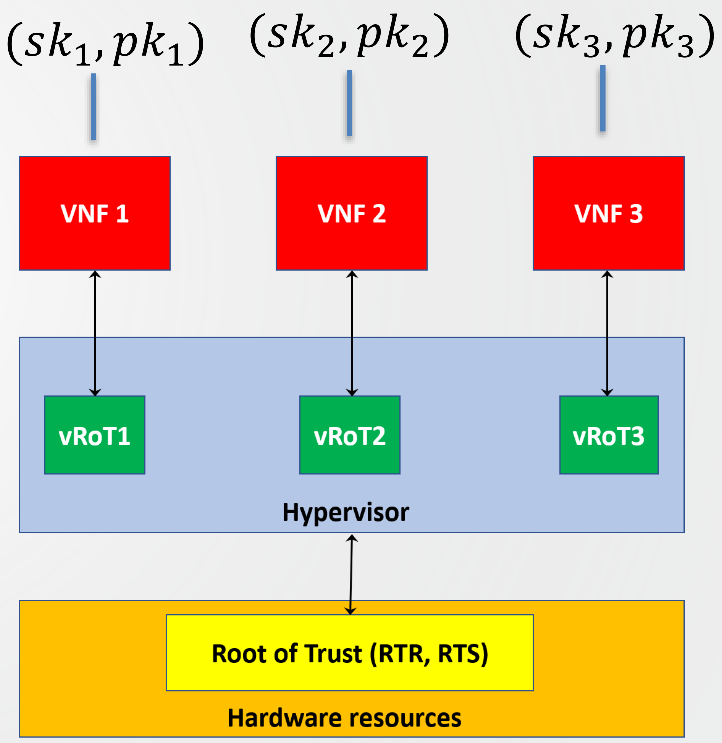


What kind of state ?

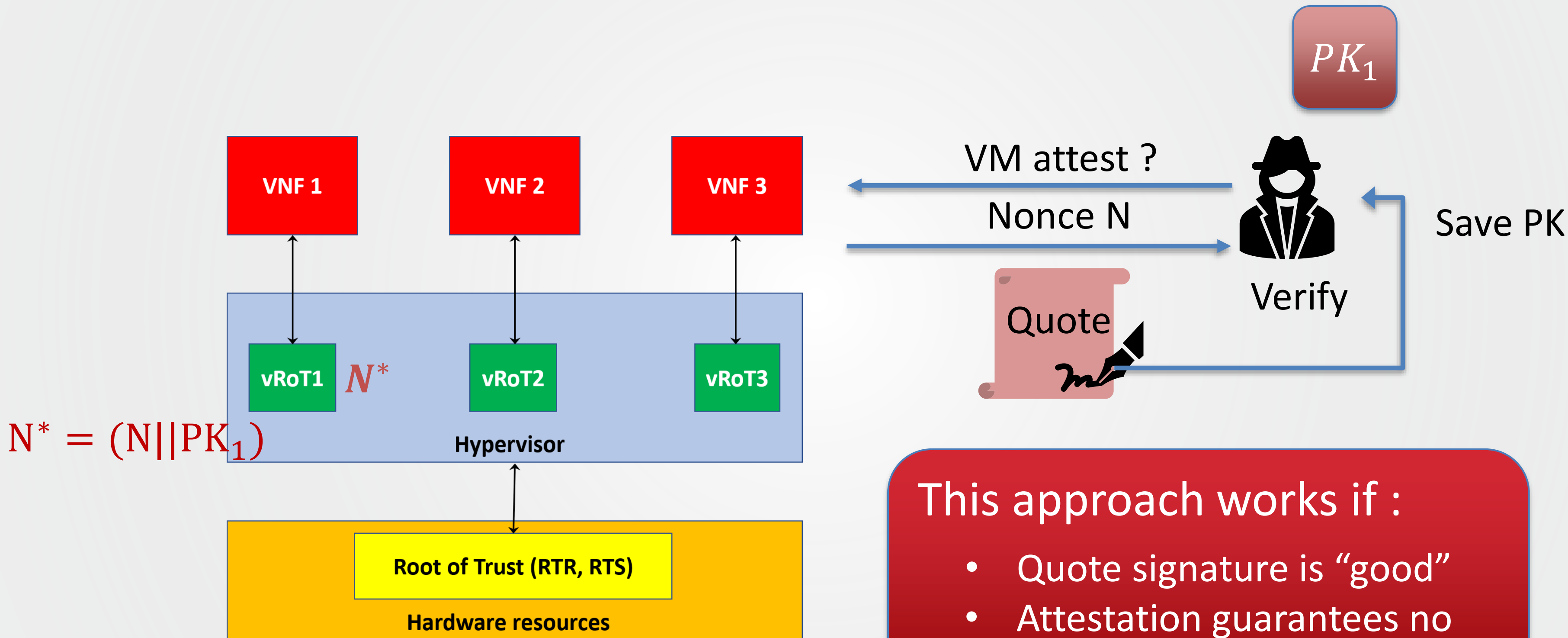
Attestation linking information

- VMs associated with vTPM-stored keys
- Associate VM quotes with keys … then have hypervisor list keys of managed VMs in TPM-signed quote
  - ❑ Oops: hypervisor is corruptible !
  - ❑ TPM could sign data, but not with the right key … or if modified
  - ❑ Can we do it without modifications ?

Glad you asked. YES !



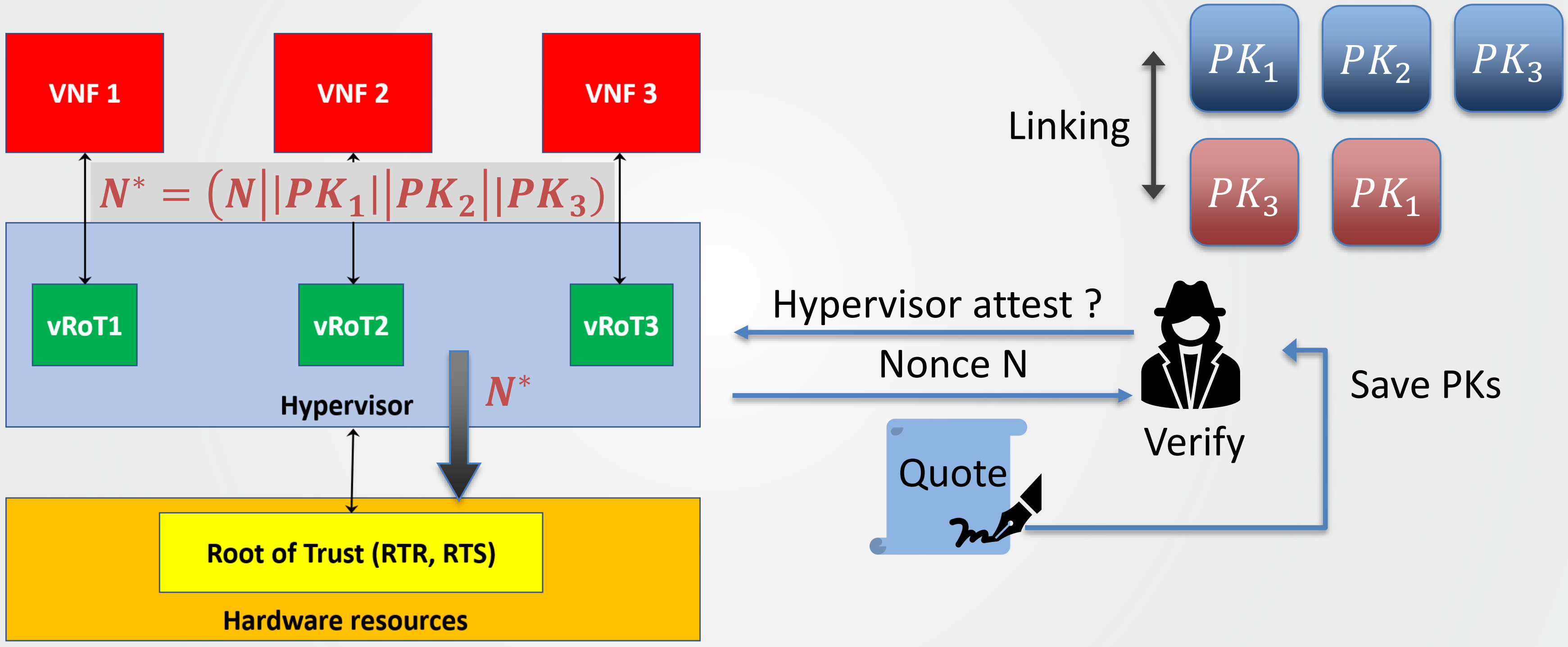
When nonces become heroes



This approach works if :

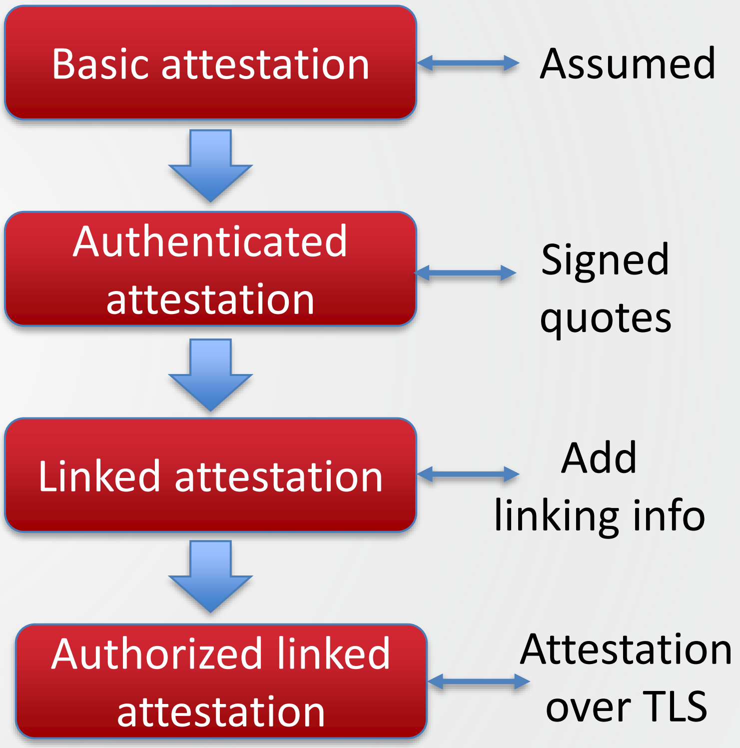
- Quote signature is “good”
- Attestation guarantees no compromise

When nonces become heroes



Provable security

- Powerful tool : mathematical security proof
- Construct scheme gradually
- Properties :
  - **Attestation** : assume attestation can flag compromise infallibly
  - **Authentication** : quote is sure to come from TPM
  - **Linking** : attestations are only linkable for co-hosted components
  - **Authorization** : confidentiality of quote w.r.t. non-authorized parties

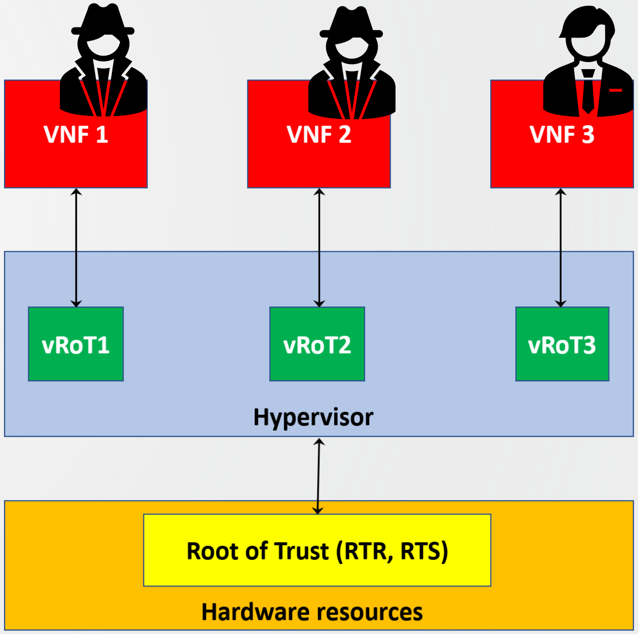


Challenge 1 unlocked !

Challenge 2:
trust in multitenant architectures

Context

- Multitenant architectures :
  - Single owner of infrastructure
  - Tenants register VMs and can check status of VMs, hypervisor, and their link



- Goals :
  - **Linkable Trust**: linkable attestations
  - **Inter-tenant privacy**: tenant only allowed to know about its own VMs
  - **Configuration-hiding**: Hide precise hypervisor configuration

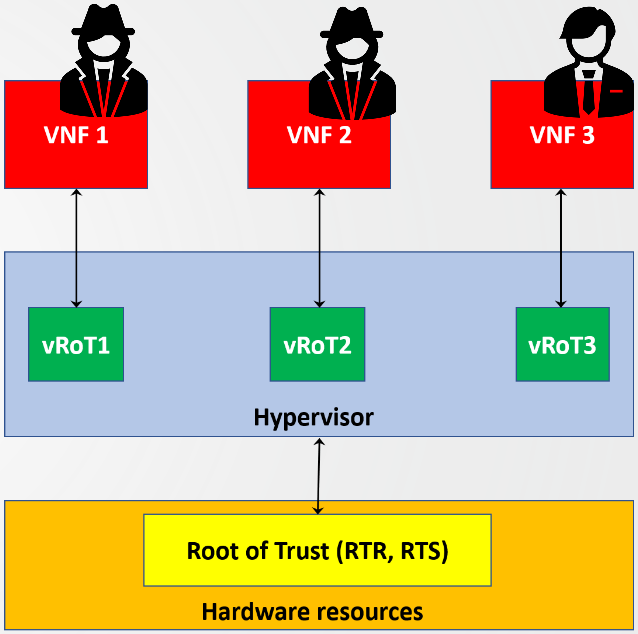
Privacy: what and why

- Multitenancy : each tenant owns only some VMs
  - Tenants can check the state of the infrastructure hosting VM
- Inter-tenant privacy: tenants know nothing about other tenants' VMs
  - In fact, tenants will not even know whether some other VMs are co-hosted with their VMs on same infrastructure
- Hypervisor configuration hiding:
  - Configuration can include sensitive details: versions of given software, presence/absence of given software…

No trivial task either

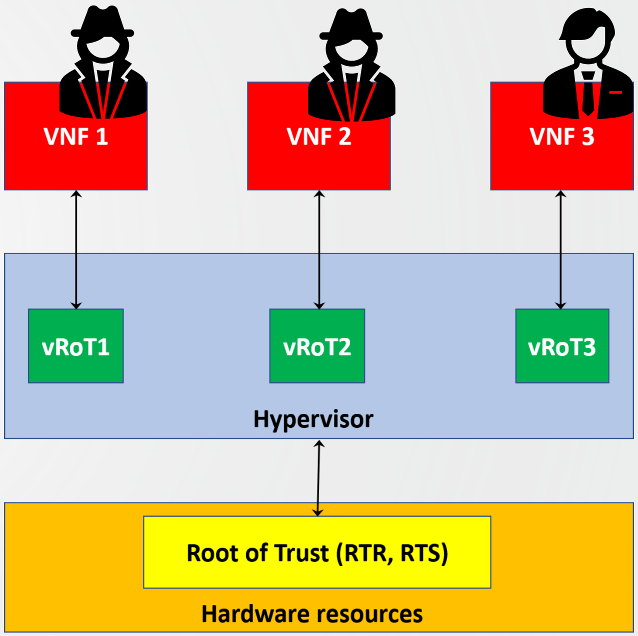
- Single-channel DA :
  - ❑ Linkable, but inefficient
  - ❑ Not configuration-hiding

- Solution from [ACNS22] :
  - ❑ Linkable, efficient
  - ❑ Hypervisor attestation breaks inter-tenant privacy
  - ❑ Not configuration-hiding



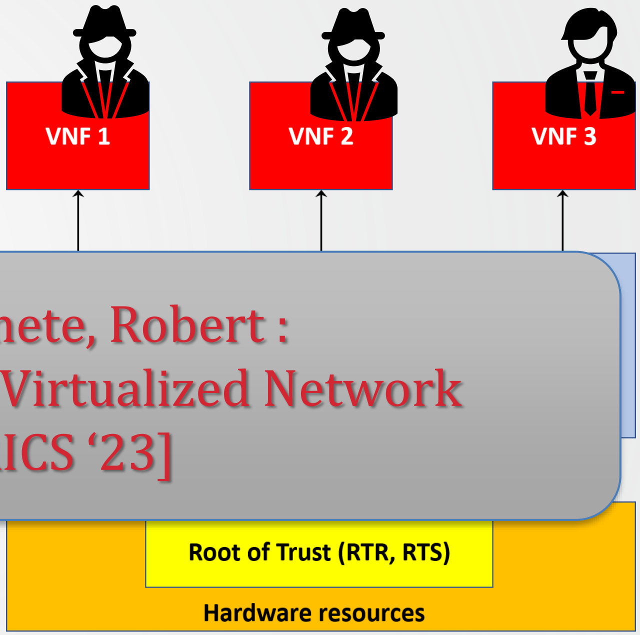
Some bad ideas

- Drop layer-linking entirely:
  - Layer-linking can ensure some conditions are fulfilled !
  - Hypervisor configuration revealed
- Make TPM a TTP for state + linking
  - Inefficient
  - Requires TPM modifications
- Reveal hypervisor configuration
  - Potentially sensitive information



Some good ideas

- How to modify linking:
  - Keep linking via keys included in hypervisor attestation…
  - …but make sure right tenant gets



Arfaoui, Jacques, Lacoste, Onete, Robert :
"Privacy-preserving Attestation for Virtualized Network Infrastructures" [ESORICS '23]

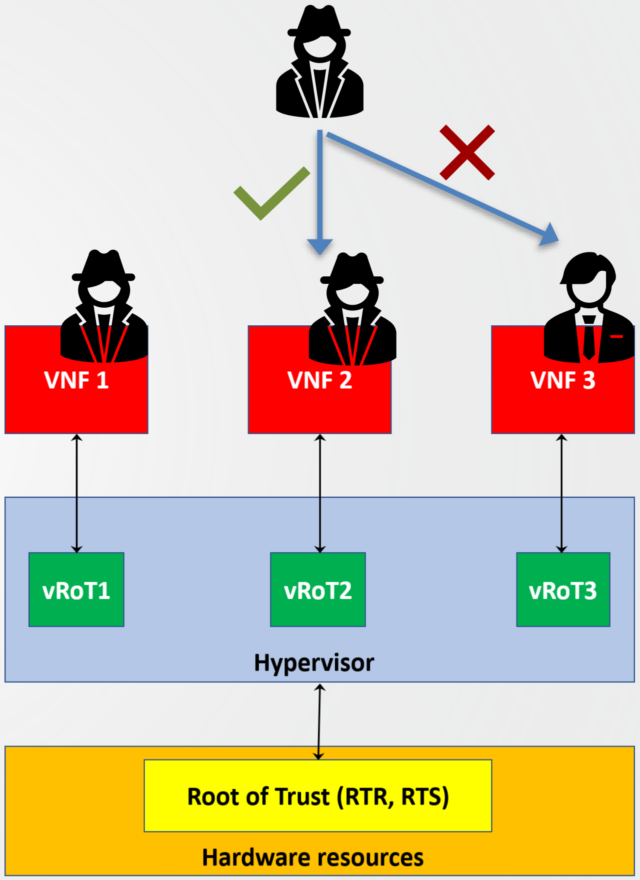
- A… hypervisor configuration
  - Hide real configuration in a set of possible configurations…
  - …without TPM modifications

Our approach: VM attestations

- Authenticate attestation demand:
  - VM only responds to tenant
  - VM plays dead otherwise

New property: responder-hiding AKE

- VM attestation demands leak no information about other tenants
- Linking information as [ACNS22]



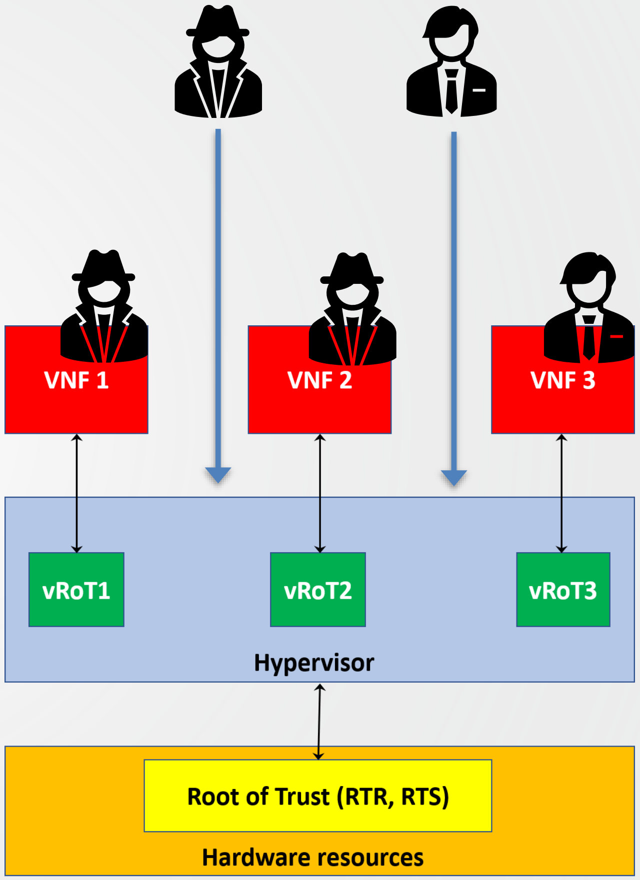
Our approach: hypervisor

- Hypervisor attestation: 1 for all:
  - Hypervisor batches attestation requests together
  - A single linkable attestation for all current requests
  - Different linking information/tenant

Use of vector commitments

  - Attestation proves configuration in set of possible configurations

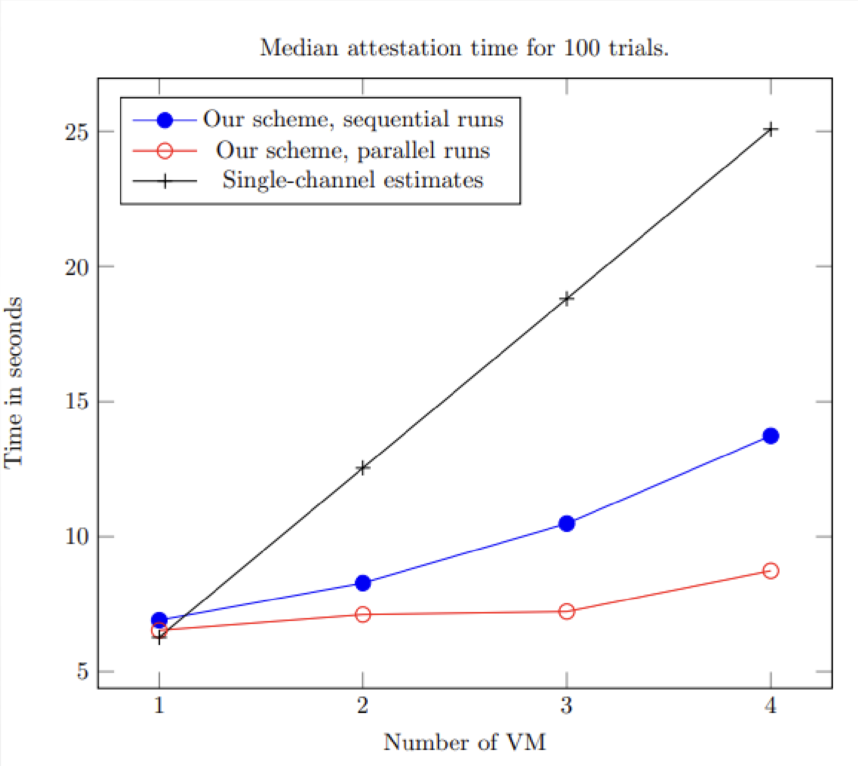
Use of ZK SNARKs



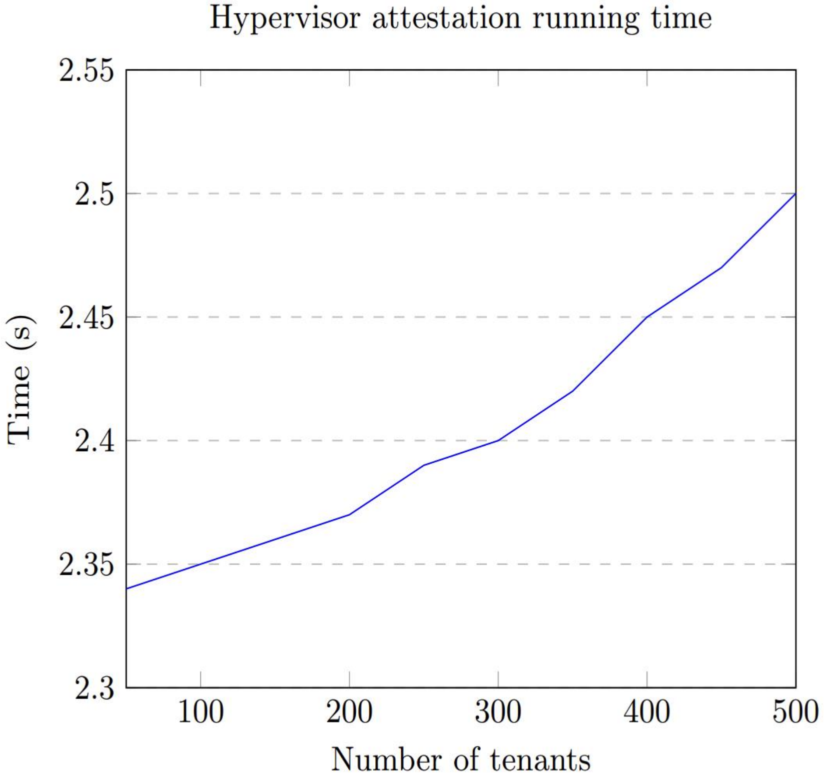
Performance

Linkable deep attestation

	min	median	mean	max
Hypervisor	3.22	5.30	5.68	11.55
VM	0.66	0.97	1.03	1.41



Privacy-preserving attestation



(a) Scaling (configuration set of size 128).

Attestation

	Mean	Median
Traditional (s)	0.94	0.94
Hypervisor (s)	2.40	2.40
SNARK (s)	1.46	1.46
Commitment (ms)	9.06	8.98

Verification

	Mean	Median
Traditional (ms)	2.42	2.36
Hypervisor (ms)	25.06	25.05
SNARK (ms)	25.02	24.99
Commitment (ms)	0.043	0.063

(b) Time to perform attestation

Conclusion and Future Work

Our results so far

- Challenge 1 : layer-linking deep-attestation
  - Layer-linking : include keys in nonces, as linking information
  - Efficiency of multi-channel DA, trust of single-channel DA
  - Properties : Attestation, Authentication (of quotes), Authorization

First provable security treatment of DA

- Challenge 2 : privacy-preserving multitenant DA
  - Strong privacy properties :
    - ✓ Inter-tenant privacy: tenants learn nothing about other tenants
    - ✓ Hypervisor configuration-hiding: hypervisor's configuration is private
  - Batching => efficiency, our ZK-SNARK => no TPM modification

Formal model and proofs of privacy properties

Some limitations

- Limited context:
  - No VM migration or cloning
- Attestation at boot time only:
  - Detects static compromise
  - Does not detect compromise during runtime
- Achieved properties:
  - Assumption of infallible attestation (detects all compromise)
  - Privacy assumes physical separation of resources

3 Challenges, 2 Results

- **Challenge 1**: efficiency vs. trust
  - ✓ ACNS 2022
- **Challenge 2** : efficiency, trust, privacy, multitenancy
  - ✓ ESORICS 2023
- **Challenge 3** : efficiency, trust, ubiquity
  - Work in progress…

Merci beaucoup !