TRUSTWORTHY WORKLOAD IDENTITY (TWI) WITH RATS

MARK NOVAK https://www.linkedin.com/in/markfishelnovak/, CHAIR, CCC TWI SIG

WITH YOGESH DESHPANDE, HENK BIRKHOLZ

CONTENT BY THE TRUSTWORTHY WORKLOAD IDENTITY SPECIAL INTEREST GROUP AT THE CONFIDENTIAL COMPUTING CONSORTIUM

OPINIONS PRESENTED ARE MY OWN
NOT OF MY EMPLOYER

KEY POINTS

- This talk is about radically expanding the total addressable market of Confidential Computing
- Data protection is a whole-of-enterprise concern (at rest, in transit, in use)
- Data exchanges overwhelmingly happen between pairs of workloads and only few – between people and workloads
- Confidential and non-confidential workload identities must be harmonized
- Only manageable and deployable solutions will succeed at scale
- Understanding of what constitutes “manageable and deployable” starts at the Relying Party, and extends to the whole of enterprise

RELYING PARTY EMPATHY

- Relying Party authorization policies must remain stable for extended periods of time (years to decades)
  - … while accommodating software and hardware upgrades that happen over that period of time
  - … as well as rollbacks
- Relying Party authorization policy is normally stated in terms of business concepts (“Payroll Application”, “Cleared for Highly Classified Information”, “Can access PII”, “Physically Located in Germany”)
- Most workloads are also Relying Parties when it comes to Workload Identity

RATS LACKS RELYING PARTY EMPATHY

RATS Charter

To improve the confidence in a system component's trustworthiness, a relying party may require evidence about:

* system component identity,
* composition of system components, including nested components,
* roots of trust,
* an assertion/claim origination or provenance,
* manufacturing origin,
* system component integrity,
* system component configuration,
* operational state and measurements of steps which led to the operational state, or
* other factors that could influence trust decisions.

AR4SI Draft

There are several types of Attester identities defined in this document. This list is extensible:

* chip-vendor: the vendor of the hardware chip used for the Attesting Environment (e.g., a primary Endorsement Key from a TPM)
* chip-hardware: specific hardware with specific firmware from an 'chip-vendor'
* target-environment: a unique instance of a software build running in an Attester (e.g., MRENCLAVE [SGX], an Identity Block [SEV-SNP], a Realm Initial Measurement (RIM) [ARM-CCA], or a hash which represents a set of software loaded since boot (e.g., TPM based integrity verification.))
* target-developer: the organizational unit responsible for a particular 'target-environment' (e.g., MRSIGNER [SGX])
* instance: a unique instantiated instance of an Attesting Environment running on 'chip-hardware' (e.g., an LDevID [IEEE802.1AR], an Instance ID [RFC9783] [ARM-CCA])

IMBUING RATS WITH RELYING PARTY EMPATHY

Possible terminology confusion: "Relying Party" is the recipient of the Workload Credential, which may or may not be the same as the RATS notion of Relying Party as the recipient of "Attestation Results".

1. New Architectural Building Block: Claims Mapper
   - Receives as inputs either Evidence or Attestation Results about a Workload
   - Emits as outputs Workload Identity (identifier + additional stable claims)
2. Integration of Claims Mapper into Workload Credential issuance flows
   - At some point during credentials issuance, low-level information (Evidence or Attestation Results) about the Workload must be mapped to business-centric claims

TRUSTWORTHY WORKLOAD IDENTITY: TWO PARTS

1. Credential Key (asymmetric, signing)
   - Secret
   - Only available to authorized workloads
   - Can be self-generated or retrieved from a Key Store
2. Credential (x.509 certificate, WIMSE “Workload Identity Token”)
   - Public
   - Contains public portion of Credential Key
   - Always generated and signed by a Credential Authority
   - Requires Proof-of-Possession of the corresponding Credential Key to use

SOLUTION PIVOTS

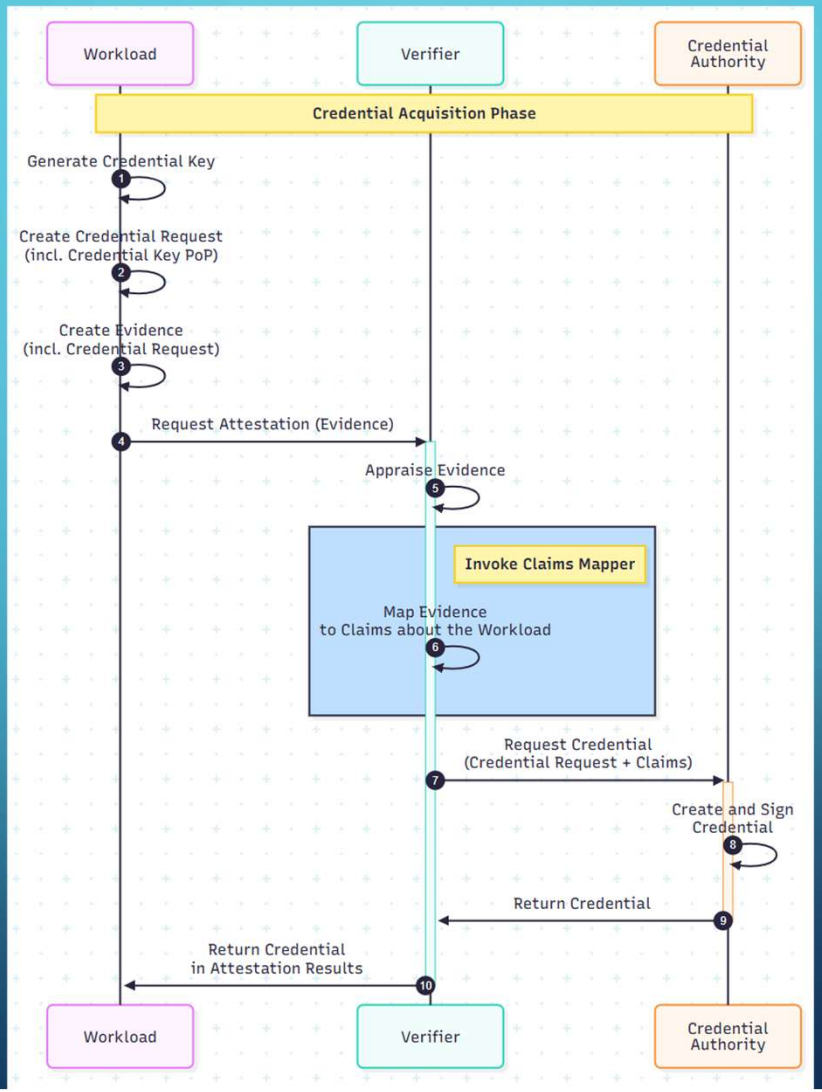
1. Key Source: Where the Workload gets its Credential Key from
   i. Self-generated by the Workload instance, certified by Remote Attestation
   ii. Retrieved from a Key Store, contingent on successful Remote Attestation
2. Credential Source: Where the Workload gets it Credential from
   i. The Verifier
   ii. The Credential Authority (e.g., a Certificate Authority, a Security Token Service, etc.)
   iii. The Workload Owner (the “control plane”)

This results in a set of Credential Acquisition Mechanisms

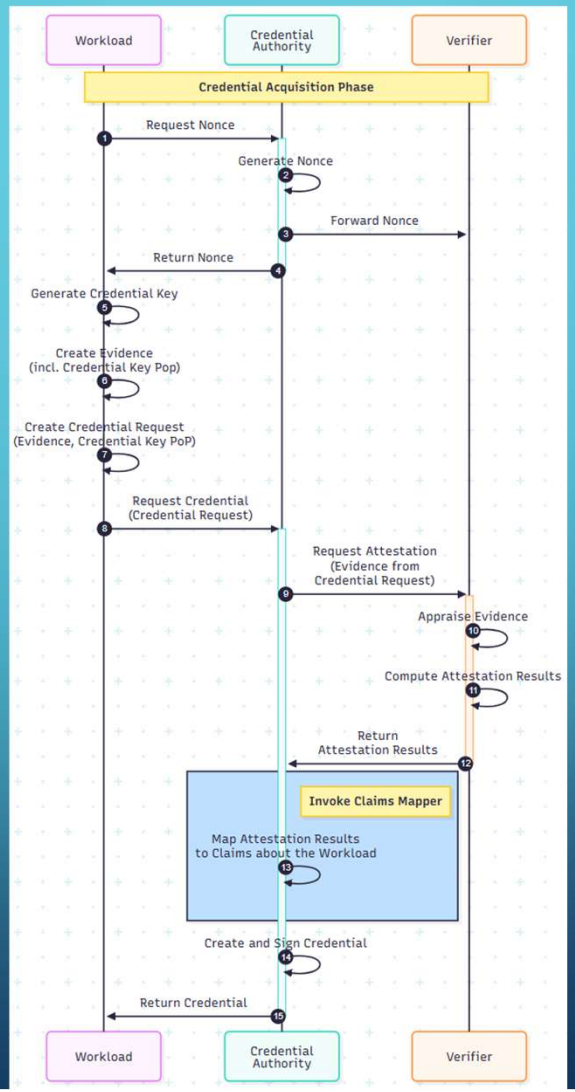
CREDENTIAL ACQUISITION MECHANISM MATRIX

Credential Source → / Key Source ↓	Verifier	Credential Authority	Workload Owner
Workload Instance	W-V-CA	W-CA-V W-V–W-CA	N/A
Key Store	N/A	N/A	W-V–W-KS

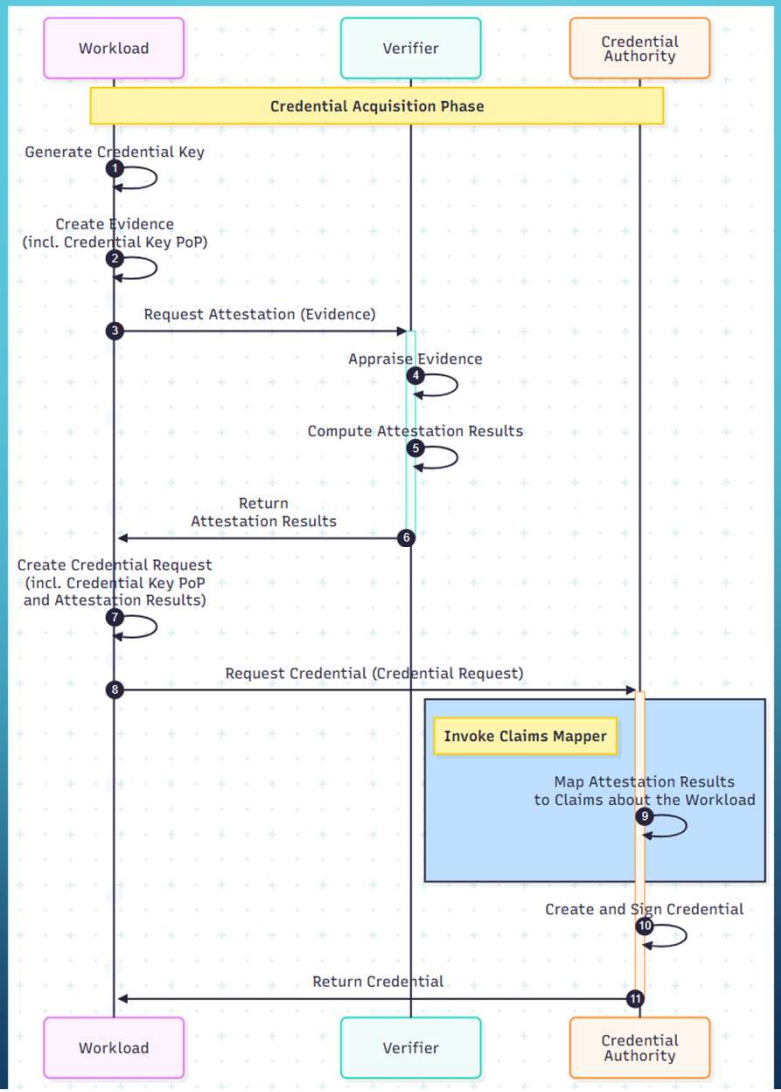
CREDENTIAL ACQUISITION MECHANISM: W-V-CA



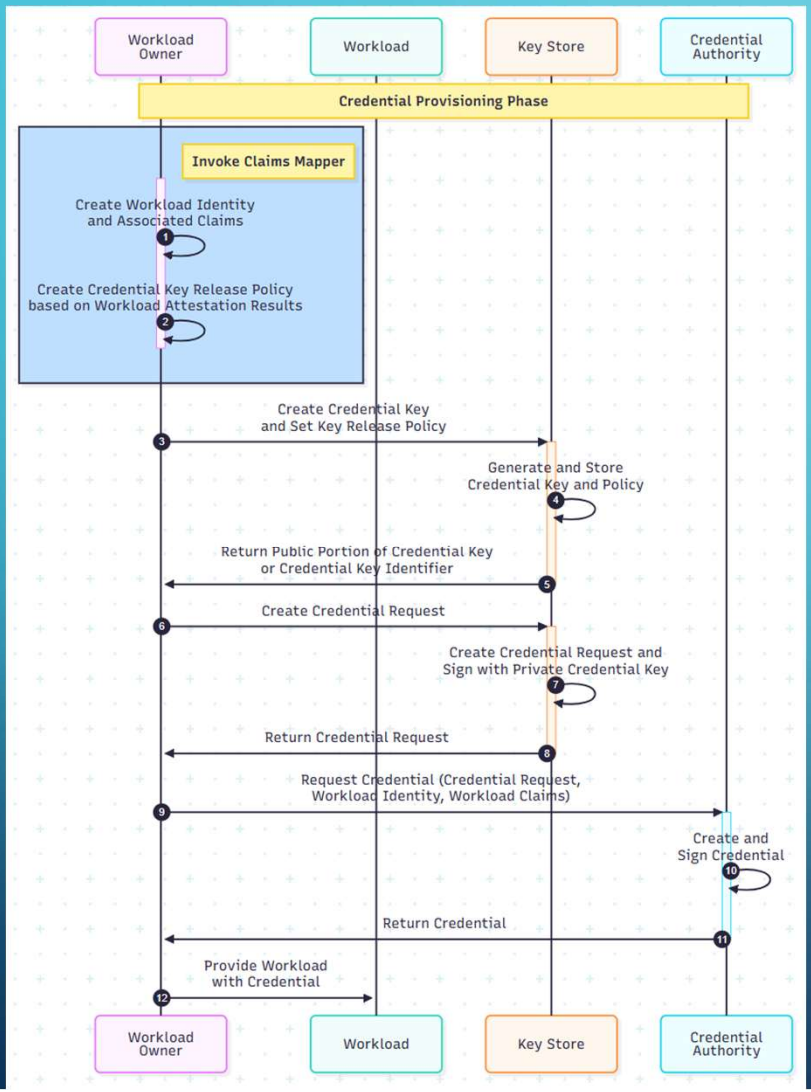
CREDENTIAL ACQUISITION MECHANISM: W-CA-V

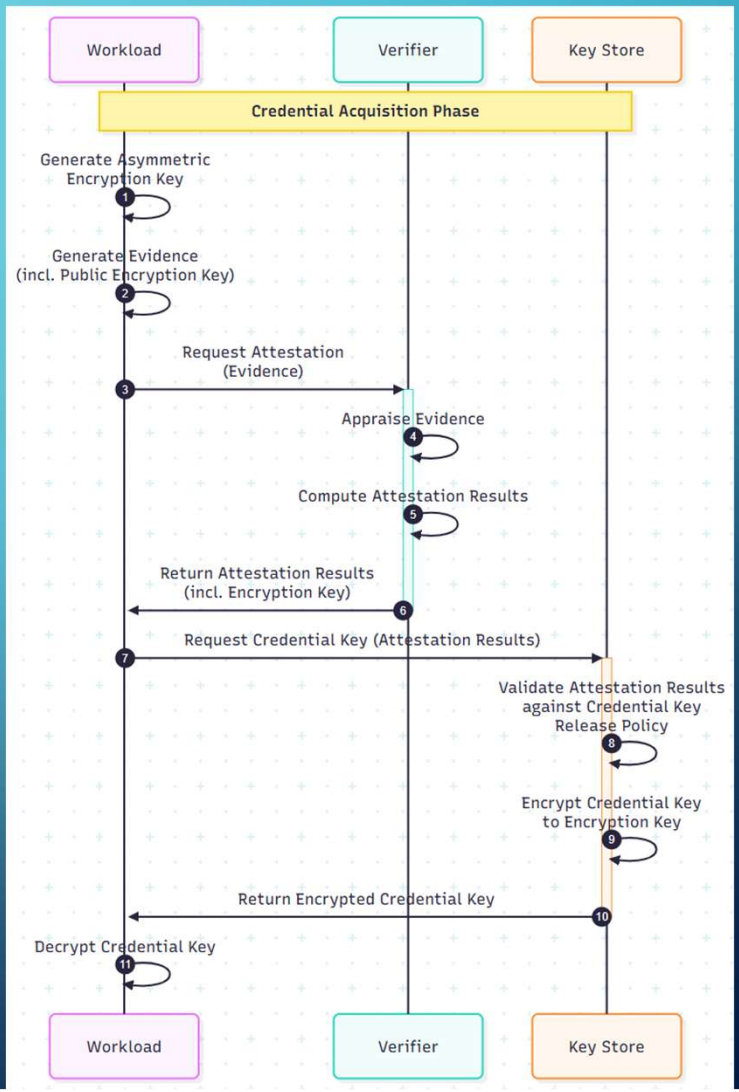


CREDENTIAL ACQUISITION MECHANISM: W-V–W-CA



CREDENTIAL ACQUISITION MECHANISM: W-V–W-KS





SECURITY ISSUES AND CONSIDERATIONS

- Verifier, Credential Authority, Key Store are fully trusted
- TLS should be used to secure all traffic between Workload, Verifier, Credential Authority and Key Store
- Live migration of a Workload could invalidate some of its claims (e.g. Germany-France migration may invalidate geolocation claims)
- Additional security analysis needed (relay, replay, redirect, …)

ADDITIONAL READING

TWI SIG Charter

https://github.com/confidential-computing/governance/blob/main/SIGs/TWI/TWI_Charter.md

TWI Definitions

https://github.com/confidential-computing/twi/blob/main/TWI_Definitions.md

TWI Requirements

https://github.com/confidential-computing/twi/blob/main/TWI_Requirements.md

TWI Reference Architecture (early draft)

https://github.com/confidential-computing/twi/blob/main/TWI_Reference_Architecture.md

This Draft

https://github.com/confidential-computing/twi-rats

ADDITIONAL INFORMATION

WHY TRUSTWORTHY WORKLOAD IDENTITY?

- Key architectural shortcomings of workload identity solutions today:
  1. Lack of workload isolation
  2. Lack of strong binding between a credential and a workload instance
  3. Lack of provenance
- A workload identity is considered *trustworthy* iff it addresses all three of these problems

WORKLOAD LIFETIMES

- Several orders of magnitude difference in lifetimes between classes of workloads:
  - Very long lived (years-decades): payroll database
  - Medium term (hours-days): payroll processing for May
  - Short term (minutes): container/pod
  - Very short term (seconds): serverless, step function, etc.
- Cannot always invoke an external service to get a credential, must use alternative approaches
- TWI Reference Architecture recommends separating the three stages of a credential's lifecycle: 1) provisioning, 2) acquisition and 3) usage

PROXIES

- Processing is multi-leg, potentially hiding server identity from client
- Multiple gateways frequently involved
  - Reverse proxies
  - Web Application Firewalls
  - Data Leakage Prevention Gateways
  - Etc.
- Workload identity must integrate with all of these and protect data in use throughout