Entities of Attestation Protocols

Verifier (V)

The trusted entity that validates integrity. Often the network's base station, though it can be any wireless sensor node with potentially more computational power than regular nodes.

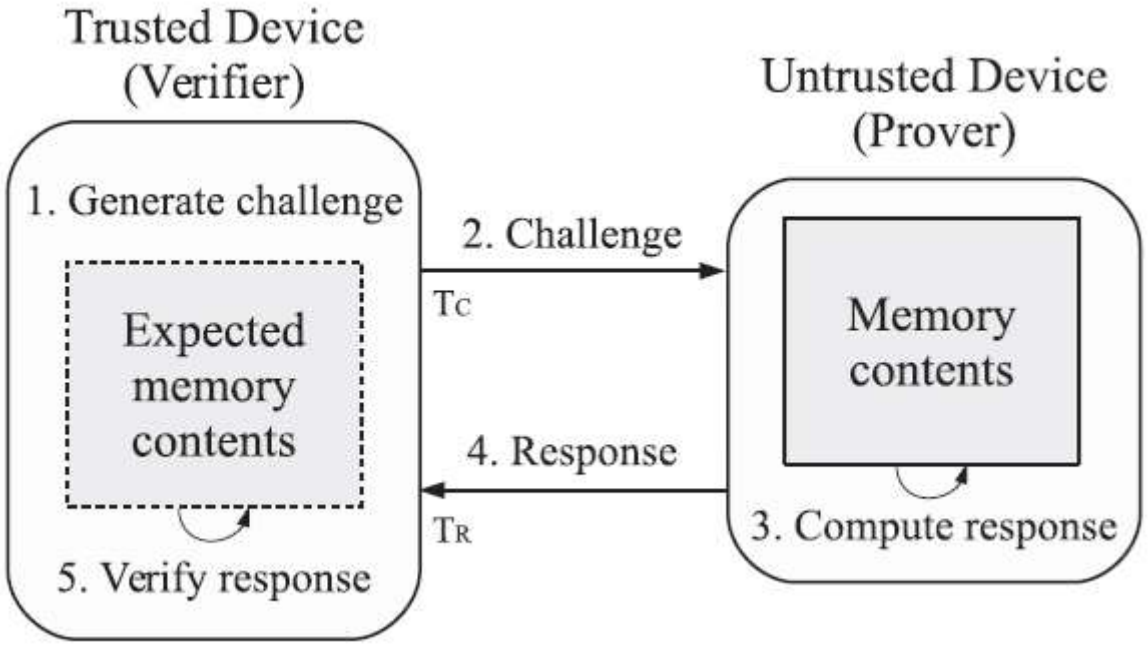
Prover (P)

The device being verified. Has an internal state reflecting its memory contents including program memory, data memory, registers, MMIO, and potentially external memories.

Adversary (A)

The attacker launching attacks remotely or using an already-compromised network node. Aims to compromise nodes without detection by the attestation procedure.

Attestation Overview



$$
\begin{aligned}
& V : c \xleftarrow{R} Challenge() \\
V \rightarrow P & : c \\
& V : T_C \leftarrow current\ time \\
& P : r \leftarrow Attest(S, c) \\
P \rightarrow V & : r \\
& V : T_R \leftarrow current\ time \\
& V : Verify(S, c, r, T_A, T_C, T_R)
\end{aligned}
$$

Key Assumptions

- About the Verifier
  - Cannot be compromised by the attacker
  - Knows the expected state of the Prover
  - Knows the hardware architecture of the Prover
- About the Adversary
  - Can reverse engineer Prover's software and hardware
  - Has full control of Prover's memory
  - Cannot modify Prover's hardware

Requirements

Authenticity

Allows the Verifier to confirm the source of the response

Atomicity

Guarantees uninterrupted execution, preventing memory modification or parallel computation

Unforgeability

Prevents adversaries from producing the same response faster than ATTEST

Dynamicity

Reflects the actual running system, not just static memory

Determinism

Enables the Verifier to reach the same result independently

Target

- Code integrity
  - The integrity of the code is verified
  - The computation result could be distorted by data manipulation
- Code + data integrity
  - The data regions are initialized to known values and verified
  - It is hard to verify the integrity during run-time
- Hardware-based attestation
  - The hardware measures the integrity of the code and data
  - The integrity metric is typically the hash value of the memory signed by the hardware
  - The run-time integrity is guaranteed by isolation

Common Attacks: Computational Exploits

1

Precomputation

Adversary precomputes operations not influenced by the challenge, gaining time for other operations. If challenges are predictable, valid responses can be precomputed entirely.

2

Forgery

Adversary attempts to generate valid responses despite memory modifications by executing modified attestation routines or altering memory so modifications neutralize each other during computation.

3

Return-Oriented Programming

Uses existing code without alteration to execute malicious operations by linking together small instruction sequences called "gadgets," circumventing defenses that assume code modification.

Common Attacks: Memory Manipulation

Memory Copy

1

If sufficient free space exists, adversary stores malicious data while keeping a copy of original contents, modifying the attestation routine to compute responses over the original copy location

Data Substitution

2

Special case when insufficient space exists for full copy—adversary modifies part of original contents and keeps copy only of overwritten data, redirecting memory reads during attestation

Compression

3

Adversary compresses original contents to gain space for malicious data, then decompresses on the fly during attestation to compute expected responses

Common Attacks: Network-Based Threats

Replay

Eavesdrop valid attestation responses from non-compromised nodes and retransmit when challenged. Only works if nodes execute the same program and receive identical challenges.

Collusion

Compromised nodes collaborate to compute valid responses, exchanging messages to recover original memory contents or dividing attestation operations across multiple devices.

Impersonation

Adversary takes multiple identities (Sybil attack), masquerading as genuine nodes during attestation or impersonating the base station to forward challenges.

Proxy

Special collusion case using a device with better computing capabilities that keeps copies of original memory contents and computes valid responses faster than common nodes.