Formal Analysis of Attested TLS

Muhammad Usama Sardar[1,2]

[1]TU Dresden, Germany

[2]Co-chair, Trusted Research Environment (TRE) Open Suite,
Global Alliance for Genomics and Health (GA4GH)

April 16, 2026

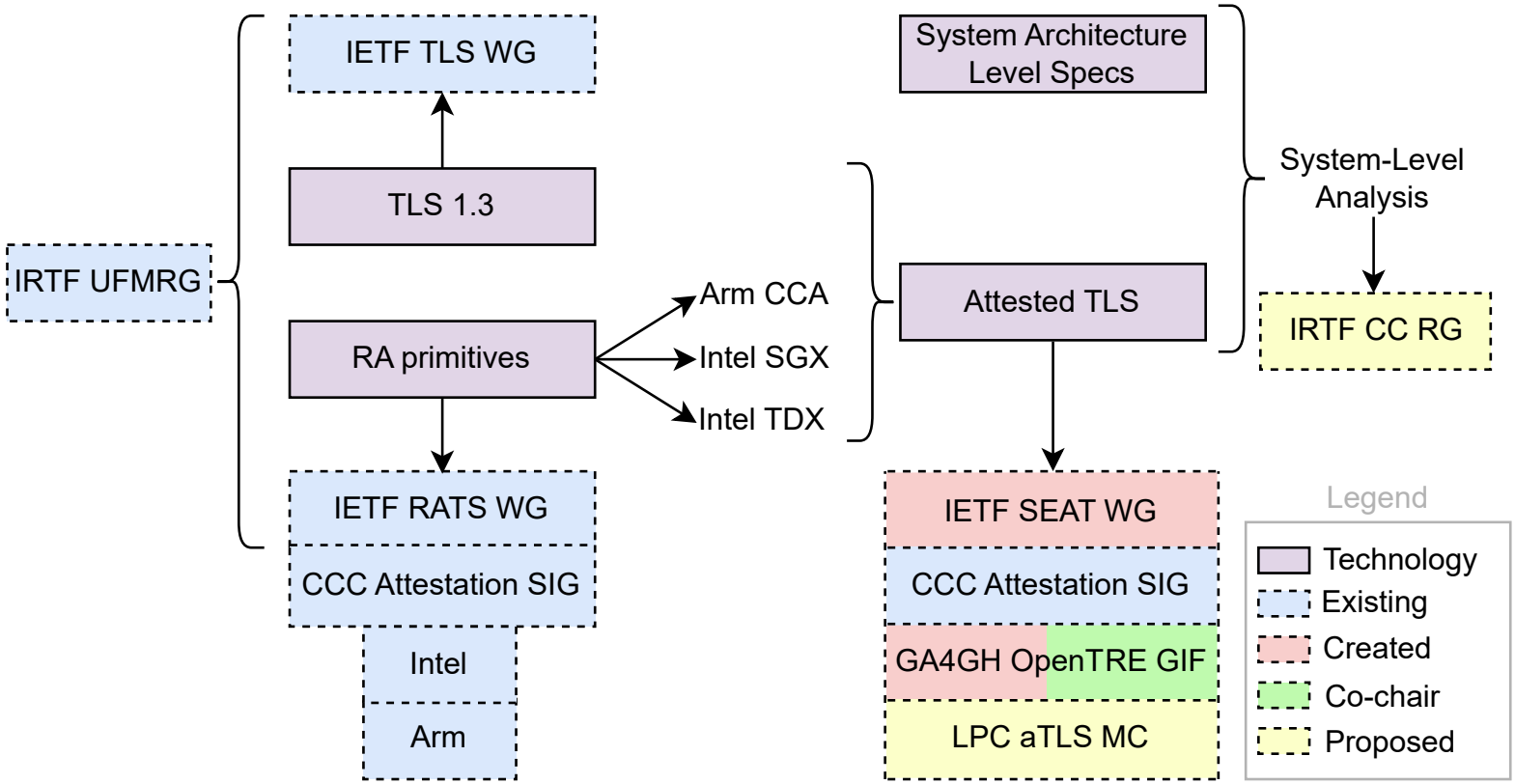
Background

- TLS protocol: one of the most widely used protocols
- Attestation: Establish trustworthiness of endpoint
  - Formal analysis of attestation presented at HotSoS'23 and '24
- Attested TLS = Attestation + TLS
- Internet Engineering Task Force (IETF): Standards Development Organization (SDO) for internet protocols
  - Attested TLS work started in IETF TLS WG
  - FATT process[1] in TLS WG
  - We found diversion attacks[2] (with Mariam and Tuomas at AsiaCCS'26)
  - Defended new WG (SEAT WG) as a proponent
  - First WG in the history of IETF to have formal analysis in its charter

[1] https://github.com/tlswg/tls-fatt

[2] https://www.researchgate.net/publication/398839141_Identity_Crisis_in_Confidential_Computing_Formal_Analysis_of_Attested_TLS

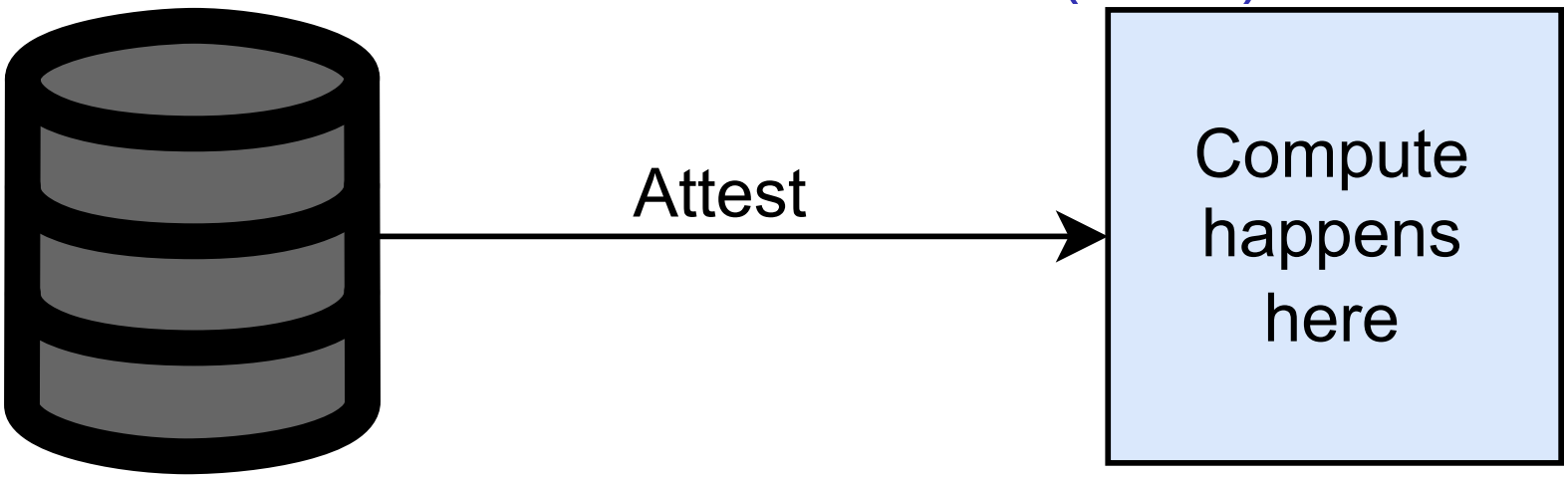
Big Picture



- IETF: Internet Engineering Task Force
- IRTF: Internet Research Task Force
- CCC: Confidential Computing Consortium
- GA4GH: Global Alliance for Genomics and Health
- RATS: Remote ATtestation ProcedureS
- TLS: Transport Layer Security
- SEAT: Secure Evidence and Attestation Transport

- UFMRG: Usable Formal Methods Research Group
- CC: Confidential Computing
- WG: Working Group
- RG: Research Group
- aTLS: Attested TLS
- SIG: Special Interest Group
- OpenTRE: Trusted Research Environment Open Suite

Recap: Trusted Research Environment (TRE) Scenario



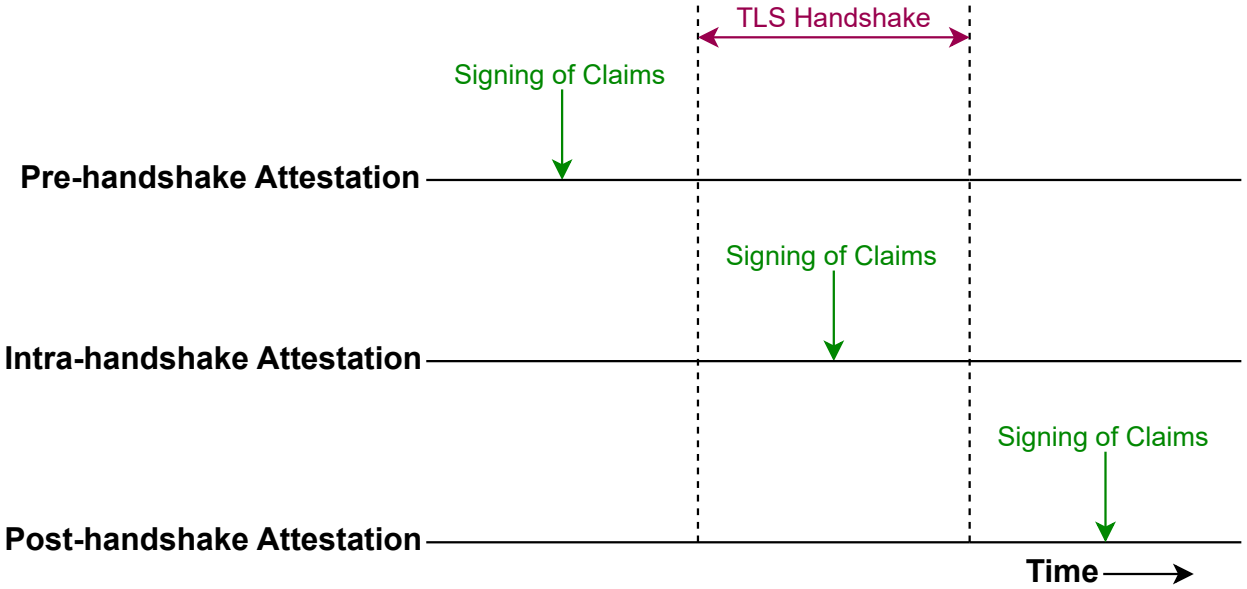
Data source
(Verifying Relying Party)
(TLS Client)

Infrastructure provider
(Attester)
(TLS Server)

- TLS + Attestation = Attested TLS
- We discovered vulnerability in all intra-handshake attestation implementations
  - Even after extensive security audit by **Trail of Bits**
- CVE-2026-33697 published: Severity = HIGH (7.5/10)

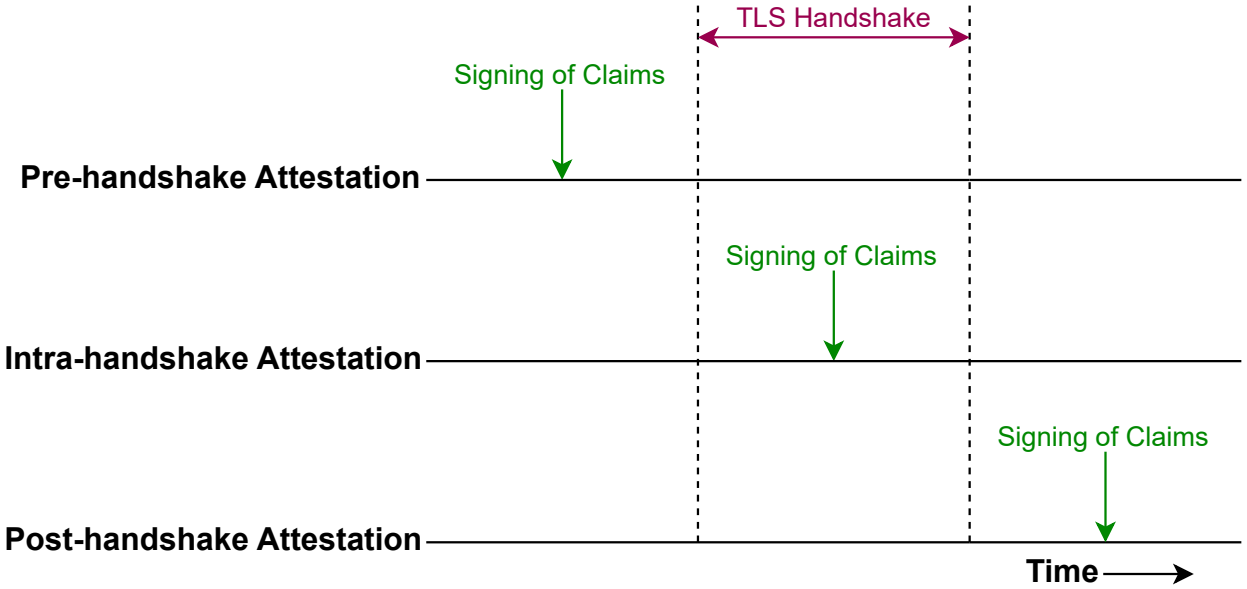
Categories of Attested TLS

- Temporal ordering of RA and TLS at Attester side



Categories of Attested TLS

- Temporal ordering of RA and TLS at Attester side



- Typical changes in each category
  - CA/TA = Certification/Trusted Authority
  - ✓ = no change; × = changes required

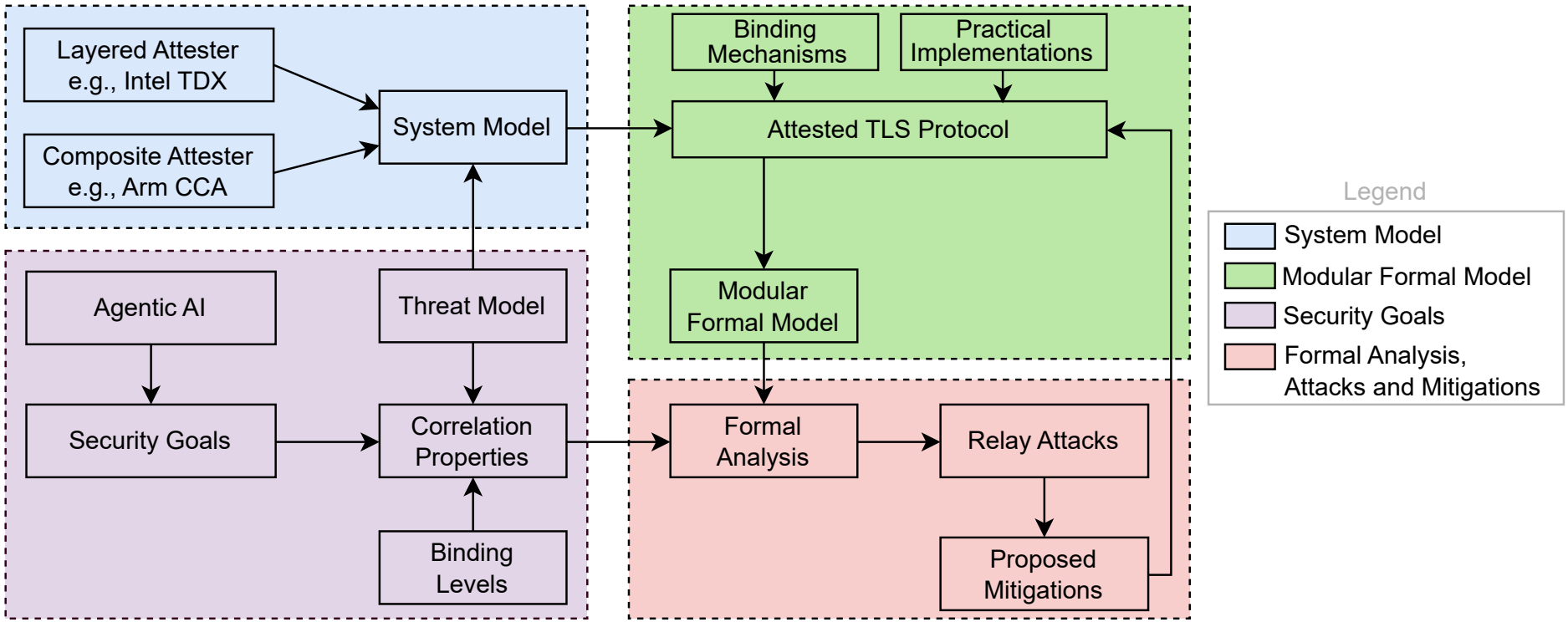
	Protocol	Deployment	Higher layer
Pre-handshake attestation	✓	× (CA/TA)	×
Intra-handshake attestation	× (Invasive)	✓	×
Post-handshake attestation	✓	✓	×

Adopted project of CCC
Attestation SIG

IETF SEAT WG AD's instruction
to exhaustively explore
intra-handshake attestation

Approach[3]

- TLS 1.3 as the transport protocol
- Agentic AI as example use case
- Formalization tool: ProVerif



[3] https://mailarchive.ietf.org/arch/msg/seat/x3eQxFjQFJLceae6l4_NgXnmsDY/

Potential Mechanisms for Binding (TLS Server as Attester)

S. No.	Binding material	Value of rdata field
1	Client's TLS nonce	nc
2	Client's attestation nonce	na
3	Early exporter	exp_0
4	Server's public key	pubEK
5	Client's attestation nonce and early exporter	na $\|$ exp_0
6	Client's attestation nonce and server's public key	na $\|$ pubEK
7	Nonce, server's public key, and early exporter	na $\|$ pubEK $\|$ exp_0

- Discussion: Any other useful binding material/combination?
- Example implementations: all are vulnerable
  - #1: Meta's AI (even after extensive security review by *Trail of Bits* without formal methods)
    - No Evidence freshness
  - #5: draft-fossati-tls-attestation-06
  - #6: Edgeless Systems Contrast, Cocos AI and CCC PoC[4]

[4] https://github.com/CCC-Attestation/attested-tls-poc

Binding Levels: Is Level 2 Sufficient?

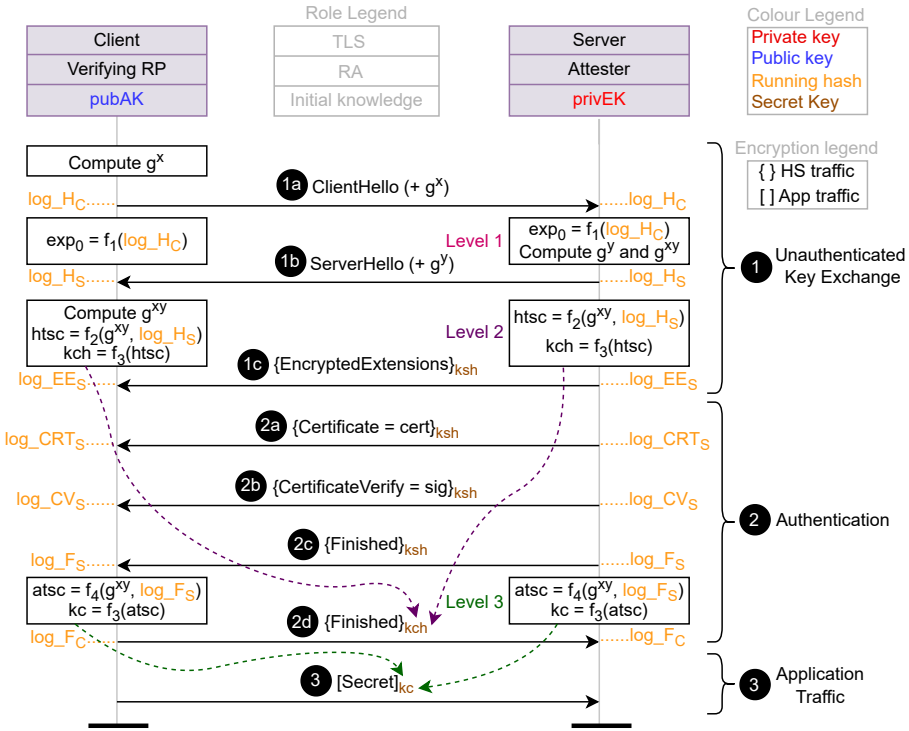
$$\forall\ ev, x, y.\ action(ClientX(ev, x)) \wedge action(ServerX(ev, y)) \rightarrow x = y \tag{1}$$

Level	Secret	Handshake state
1	DH shared secret (g^{xy})	g^x from ClientHello and g^y from Server
2	Client's handshake traffic key (*htsc*)	complete ClientHello and ServerHello
3	Client's application traffic key (*atsc*)	ClientHello up to ServerFinished

$$G_3 \Rightarrow G_2 \Rightarrow G_1 \tag{2}$$

Is Binding to client handshake traffic secret Sufficient?

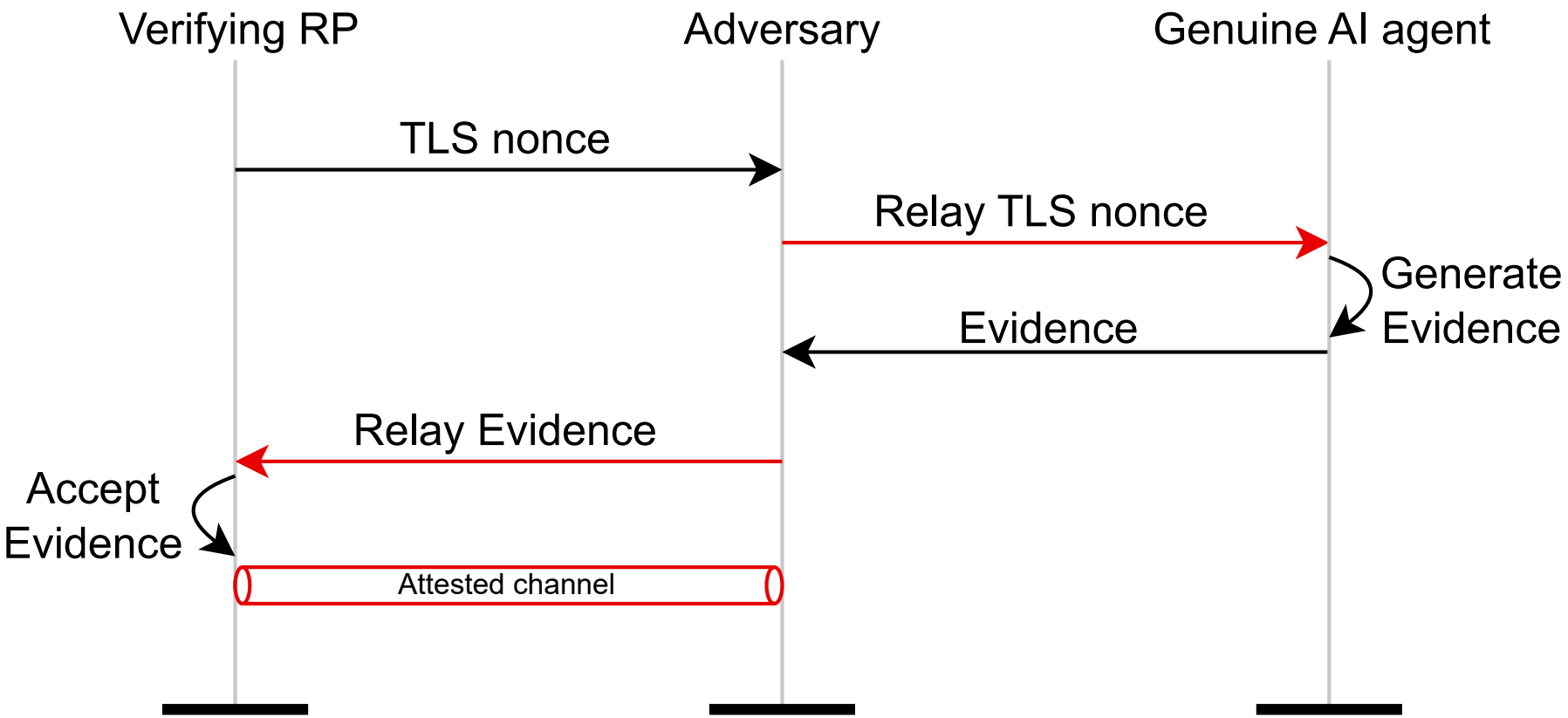
Devil is in the Details!



- `htsc`: used for encryption of clientFinished message (2d).
  - Irrelevant for security goals
  - Server not yet authenticated at this point
- `atsc`: used for encryption of application data (client's secret, e.g., decryption key)
  - Relevant for security goals

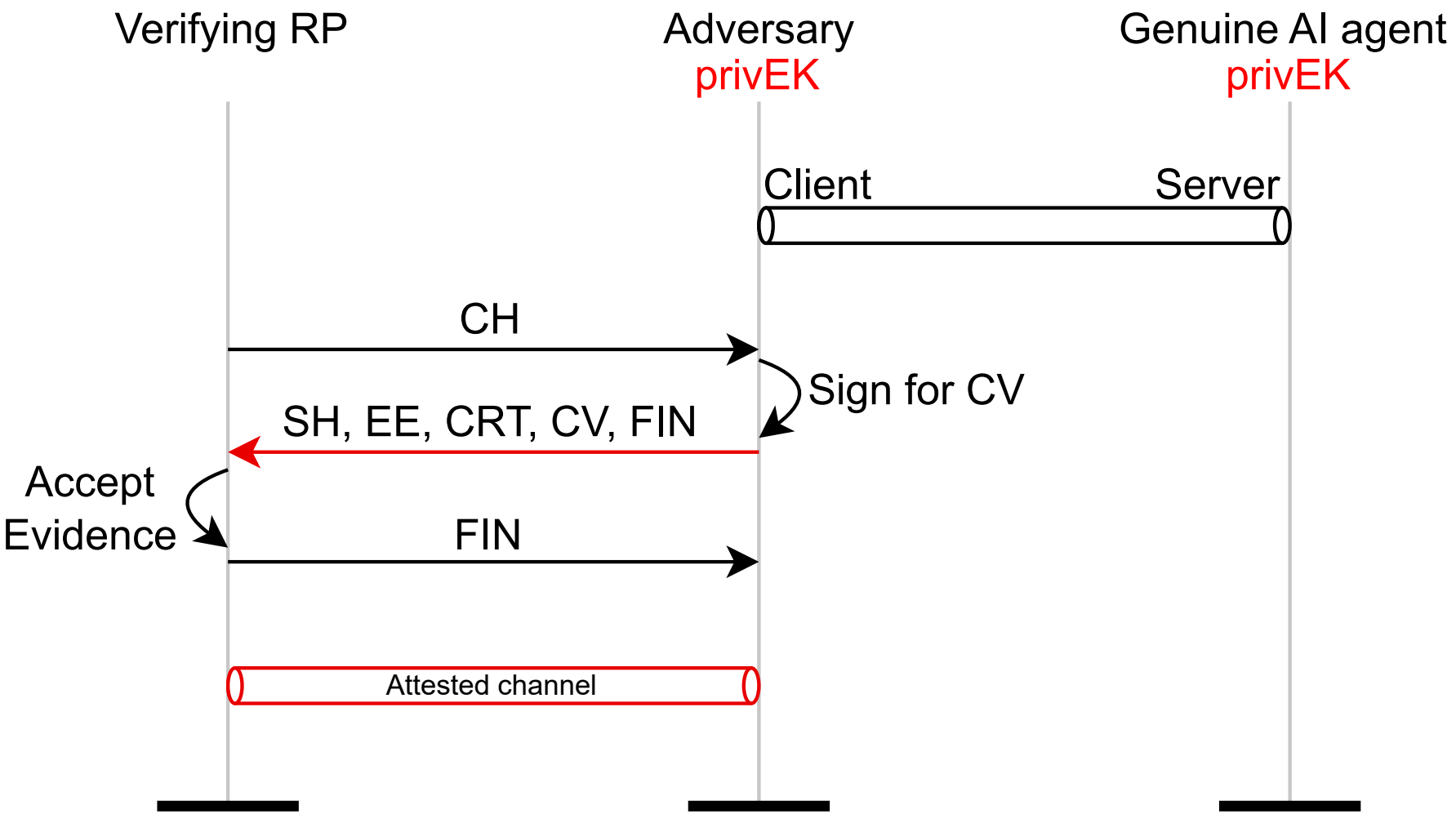
Relay Attack 1 (Abstracted): Mechanism # 1 (nc)

- Adversary can relay nonce to a genuine Attester



Relay Attack 2 (Abstracted): Mechanism # 4 (pubEK)

- No protection if privEK is accessible to some entity other than an AI agent (e.g., provisioning at runtime) or leaked via vulnerability in code
- Adversary gets signed Evidence from genuine AI agent



Relay attacks acknowledged by Cocos AI[5]

- Cocos AI uses "Attestation nonce ‖ Server's public key"

Limitations and the Relay Attack

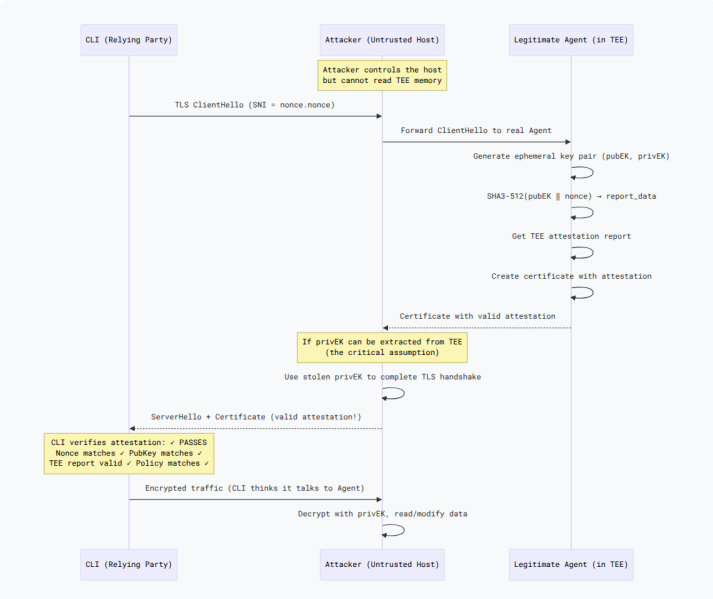
The Formal Analysis

Sardar et al. analyzed all known intra-handshake attestation implementations using ProVerif, a symbolic security analysis tool. Their key findings, presented at IETF SEAT meeting 124 and documented in draft-usama-seat-intra-vs-post:

1. **All** analyzed binding mechanisms fail to achieve even Level 1 binding (correlation of Evidence to the shared DH secret).
2. Any binding that involves the server's public key requires the **additional assumption** that the server's private key does not leak.
3. The extension of TLS with attestation in these implementations does not bring the security benefit one might expect from a purely theoretical perspective.

The Relay Attack Scenario

Here is the concrete attack that the formal analysis proves is possible:



[5] https://web.archive.org/web/20260227160554/https://www.ultraviolet.rs/blog/tee-tls-privacy/

High-severity (7.8/10) CVE issued by Cocos AI[6]

CoCoS attested TLS is vulnerable to relay attacks via extracted ephemeral TLS keys

High **dborovcanin** published **GHSA-vfgg-mvxx-mgg7** last week

Package	Affected versions	Patched versions
github.com/ultravioletrs/cocos (Go)	v0.4.0	v0.9.0

Description

Impact

The current implementation of attested TLS (aTLS) in CoCoS is vulnerable to a relay attack affecting all versions from v0.4.0 through v0.8.2. This vulnerability is present in both the AMD SEV-SNP and Intel TDX deployment targets supported by CoCoS.
In the affected design, an attacker may be able to extract the ephemeral TLS private key used during the intra-handshake attestation. Because the attestation evidence is bound to the ephemeral key but not to the TLS channel, possession of that key is sufficient to do a relay attack. A client will accept the connection under false assumptions about the endpoint it is communicating with — the attestation report cannot distinguish the genuine attested service from the attacker's relay.

This undermines the intended authentication guarantees of attested TLS. A successful attack may allow an attacker to impersonate an attested CoCoS service and access data or operations that the client intended to send only to the genuine attested endpoint.

Exploitation requires the attacker to first extract the ephemeral TLS private key, which is possible through physical access to the server hardware, transient execution attacks, or side-channel attacks.

Note: The aTLS implementation was fully redesigned in v0.7.0, but the redesign does not address this vulnerability. The relay attack weakness is architectural and affects all releases in the v0.4.0–v0.8.2 range.

This vulnerability class was formally analyzed and demonstrated across multiple attested TLS implementations, including CoCoS, by researchers whose findings were disclosed to the IETF SEAT Working Group. Formal verification was conducted using ProVerif.

Severity

High 7.8 / 10

CVSS v3 base metrics

Attack vector	Local
Attack complexity	High
Privileges required	None
User interaction	None
Scope	Changed
Confidentiality	High
Integrity	High
Availability	None

Learn more about base metrics

CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:N

CVE ID

CVE-2026-33697

Weaknesses

- CWE-322
- CWE-346

[6] https://github.com/ultravioletrs/cocos/security/advisories/GHSA-vfgg-mvxx-mgg7

Formal Security Analysis Using ProVerif Tool

CWE 2 Total

Learn more

- CWE-322: CWE-322: Key Exchange without Entity Authentication
- CWE-346: CWE-346: Origin Validation Error

CVSS 1 Total

Learn more

Score	Severity	Version	Vector String
7.5	HIGH	3.1	CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:C/C:H/I:H/A:N

- draft-fossati-tls-attestation[7]: vulnerable[8] (CVE-2026-33697)
  - Responsible disclosure[9] followed

[7] https://datatracker.ietf.org/doc/draft-fossati-tls-attestation/

[8] https://www.cve.org/CVERecord?id=CVE-2026-33697

[9] https://github.com/ultravioletrs/cocos/security/advisories/GHSA-vfgg-mvxx-mgg7

Binding Evidence to
unauthenticated peer (Server)

Need for Implementation Guidance

- Meta's AI: TLS nonce (hence, no Evidence freshness)
- Conveyance of attestation nonce
  - Edgeless Systems Contrast (previously) and Cocos AI abuse SNI (intended for hostname)
  - Edgeless Systems Contrast (now) abuse ALPN (intended for protocol names)

Summary of Security Analysis So Far

- Pre-handshake attestation: replay and diversion attacks
- Intra-handshake attestation: diversion and relay attacks
  - draft-fossati-tls-attestation[10]
  - draft-fossati-seat-early-attestation[11]
- Post-handshake attestation: no known attacks
  - draft-fossati-seat-expat[12]

- Intra-handshake attestation can achieve at most level 2 binding

- Unless one re-establishes connection each time attestation is required, post-handshake attestation is required, and hence, intra-handshake only adds unnecessary complexity.

- PQ attested TLS (hybrid vs. pure debates ongoing)

[10] https://datatracker.ietf.org/doc/draft-fossati-tls-attestation/

[11] https://datatracker.ietf.org/doc/draft-fossati-seat-early-attestation/

[12] https://datatracker.ietf.org/doc/draft-fossati-seat-expat/

Links to Resources

- Paper on identity crisis
  - https://www.researchgate.net/publication/398839141_Identity_Crisis_in_Confidential_Computing_Formal_Analysis_of_Attested_TLS
- Wiki page
  - https://github.com/EuroProofNet/ProgramVerification/wiki/AttestedTLS
- Formal proof of insecurity of pre- and intra-handshake attestation
  - https://github.com/CCC-Attestation/formal-spec-id-crisis
- Post-handshake attestation draft
  - https://datatracker.ietf.org/doc/draft-fossati-seat-expat/
- Attestation in Arm CCA and Intel TDX
  - https://github.com/CCC-Attestation/formal-spec-TEE
- Security considerations of remote attestation
  - https://datatracker.ietf.org/doc/draft-sardar-rats-sec-cons/
- IETF SEAT WG
  - https://datatracker.ietf.org/wg/seat/about/
- Technical Concepts
  - https://www.researchgate.net/publication/396199290_Perspicuity_of_Attestation_Mechanisms_in_Confidential_Computing_Technical_Concepts
- Validation of TLS 1.3 Key Schedule
  - https://www.researchgate.net/publication/396245726_Perspicuity_of_Attestation_Mechanisms_in_Confidential_Computing_Validation_of_TLS_13_Key_Schedule
- General Approach
  - https://www.researchgate.net/publication/396593308_Perspicuity_of_Attestation_Mechanisms_in_Confidential_Computing_General_Approach
- Weekly meetings for protocol design
  - https://github.com/tls-attestation#meetings

ACK: Co-authors (in papers/IETF drafts)

- Jean-Marie Jacquet (University of Namur)
- Ionut Mihalcea (Arm)
- Thomas Fossati (Linaro)
- Arto Niemi (Huawei)
- Hannes Tschofenig (University of Applied Sciences Bonn-Rhein-Sieg and Siemens)
- Simon Frost (Arm)
- Ned Smith (Intel)
- Carsten Weinhold (Barkhausen Institut)
- Michael Roitzsch (Barkhausen Institut)
- Yogesh Deshpande (Arm)
- Yaron Sheffer (Intuit)
- Tirumaleswar Reddy K. (Nokia)
- Henk Birkholz (Fraunhofer SIT)
- Mariam Moustafa (Aalto University)
- Tuomas Aura (Aalto University)
- Liang Xia (Huawei)
- Weiyu Jiang (Huawei)
- Jun Zhang (Huawei)
- Houda Labiod (Huawei)
- Yuning Jiang (Huawei International)
- Meiling Chen (China Mobile)
- Peter Chunchi Liu (Huawei Technologies)
- Minghui Xu (Shandong University)
- Pavel Nikonorov (GENXT)
- Viacheslav Dubeyko (IBM)

ACK: Contributors

- Eric Rescorla (Independent)
- Laurence Lundblade (Security Theory LLC)
- Göran Selander (Ericsson AB)
- Marco Tiloca (RISE AB)
- Richard Barnes (Cloudflare)
- Giridhar Mandyam (AMD)
- Christopher Patton (Cloudflare)
- Dionna Amalie Glaze (Google)
- Bob Beck (Google)
- Mike Ounsworth (Cryptic Forest Software)
- John Preuß Mattsson (Ericsson Research)
- Cedric Fournet (Microsoft)
- Thore Sommer (TU Munich)
- Nikolaus Thümmel (Scontain)
- Jonathan Hoyland (Cloudflare)
- Jo Van Bulck (KU Leuven)
- Martin Thomson (Mozilla)
- Britta Hale (Naval Postgraduate School)
- Werner Staub (CORE Association)
- Christian Simmen (DENIC)
- Dennis Jackson (Mozilla)
- Paul Wouters (Aiven)
- Matthias Wählisch (TU Dresden)
- Andrey Ruzhanskiy (Telekom MMS)
- Muuhh Ikede (Cybertrust)
- Mike Bursell (CCC)
- Ravi Sahita (Rivos)
- Samuel Ortiz (Rivos)
- Mathieu Poirier (Linaro)
- Hannes Reinecke (SUSE)
- Alexander Graf (AWS)
- Elena Reshetova (Intel)
- Jon Lange (Microsoft)
- Daniel Kiper
- David Woodhouse (AWS)
- David Kaplan (AMD)
- Tiziano Santoro (Google)
- Juho Forsén
- Ira McDonald
- Markus Rudy (Edgeless Systems)
- Ayoub Benaissa (Zama)
- Greg Kostal (Microsoft)
- Mike Stunes (Microsoft)
- David Altobelli (Microsoft)
- Venkat Malladi (Quest Diagnostics)
- and many others...