

Standardization and Open-source Implementation of Attested TLS for Confidential Computing

Muhammad Usama Sardar^{1,2} and Peg Jones³

¹TU Dresden, Germany

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

³Flashbots, Germany

February 1, 2026

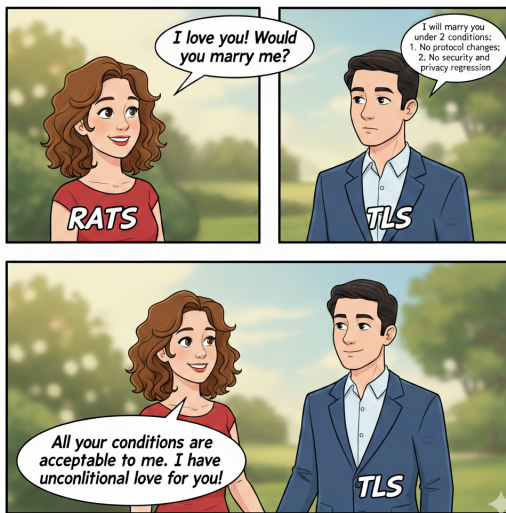
Usama thanks his travel sponsor!



Outline

- 1 Why are we here?
- 2 Technical Background
- 3 Open-source Implementation
- 4 TC;DU (Too Complicated; Didn't Understand)

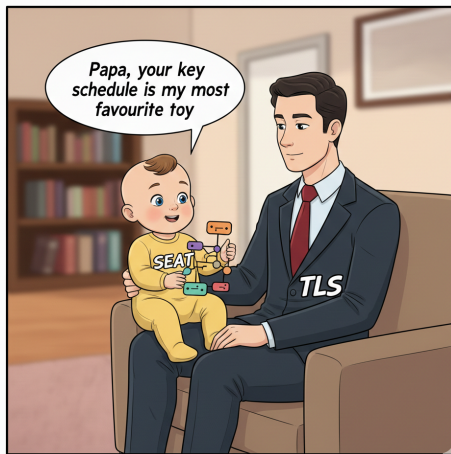
TLS weds RATS¹ (July 21, 2025)



(Image credits: Gemini)

¹<https://datatracker.ietf.org/doc/bofreq-fossati-tls-exported-attestation-expat/>

SEAT is born² (October 9, 2025) and soon after starts breaking it



(Image credits: Gemini)

²<https://mailarchive.ietf.org/arch/msg/seat/Jo7RomCQ9Is0-rf48RMJtas0s0Y/>

Why are we here? Relevance to FOSDEM

We provided Free matrimonial services!

We need help with Free babysitting services!

(Image credits: Gemini)

Why are we here? Relevance to FOSDEM

We are developing Open-Source impl. and formal verification!

We welcome further Open-Source impl. and formal verification!

(Image credits: Gemini)

Outline

- 1 Why are we here?
- 2 Technical Background**
- 3 Open-source Implementation
- 4 TC;DU (Too Complicated; Didn't Understand)

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

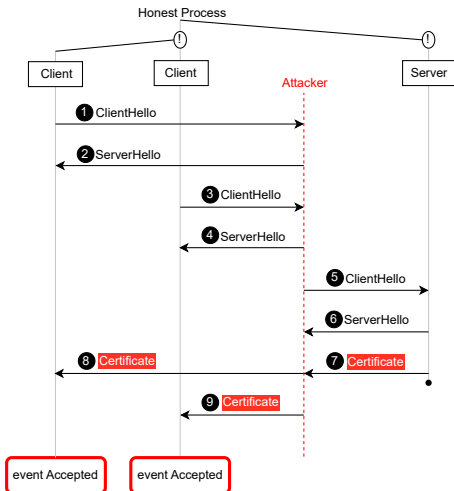
CA/TA = Certification/Trusted Authority

X = no change; = changes required

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

1 Pre-handshake Attestation: Replay Attacks³

Adversary can **replay** Evidence in Certificate



³Sardar, Niemi, Tschofenig, and Fossati, "Towards Validation of TLS 1.3 Formal Model and Vulnerabilities in Intel's RA-TLS Protocol", 2024.

2.1 Intra-handshake Attestation: Diversion Attacks⁴

No PKI cert ⇒ No identity authentication

Hostname not measured ⇒ Diversion to a different data center

⁴Sardar, Moustafa, and Aura, "Identity Crisis in Confidential Computing: Formal Analysis of Attested TLS", 2026.

2.2 Intra-handshake Attestation: Relay Attacks⁵

Adversary can relay nonce to a genuine Attester

Cocos AI is broken!

⁵Sardar, Perspicuity of Attestation Mechanisms in Confidential Computing, 2026.

2.3 Intra-handshake Attestation: Hard to Secure⁶

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)

Relevant for security goals

⁶Sardar, Perspicuity of Attestation Mechanisms in Confidential Computing, 2026.

Exported Authenticators⁷(RFC 9261)

ClientCertificateRequest: Same as CertificateRequest

Key for HMAC of Finished using **Exported Keying Material (EKM)**

⁷Sullivan, Exported Authenticators in TLS, 2022.

3 Post-handshake Attestation⁸

No change in TLS handshake protocol

Example: TLS Server as RATS Attester

⁸Sardar, Moustafa, and Aura, "Identity Crisis in Confidential Computing: Formal Analysis of Attested TLS", 2026.

Outline

- 1 Why are we here?
- 2 Technical Background
- 3 Open-source Implementation**
- 4 TC;DU (Too Complicated; Didn't Understand)

Overview of implementation

Based on Rustls

veraison/rust-cmw for RATS Conceptual Messages Wrapper⁹

Transport agnostic, with included implementation over QUIC

Supports remote peer and mutual attestation

Supported by Cypherpunk fellowship program. Thank you!



The screenshot shows a web browser displaying the Rust documentation for the `export_keying_material` method. The URL is `docs.rs/rustls/latest/rustls/enum.Connection.html#method.export_keying_material`. The page content includes a description of the method and its signature.

```
Processes any new packets read by a previous call to Connection::read_tls.  
See ConnectionCommon::process_new_packets() for more information.
```

```
pub fn export_keying_material<T: AsMut<[u8]>>(  
    &self,  
    output: T,  
    label: &[u8],  
    context: Option<&[u8]>,  
) -> Result<T, Error>
```

Derives key material from the agreed connection secrets.

⁹Birkholz, Smith, Fossati, Tschofenig, and Glaze, *RATS Conceptual Messages Wrapper (CMW)*, 2025.

Difficulties encountered

Only one known open-source reference implementation¹⁰ of exported authenticators (RFC9261)

Partial implementation

Implementing exported authenticators requires using TLS handshake messages

Rustls intentionally does not expose the handshake message types - so they had to be implemented from scratch.

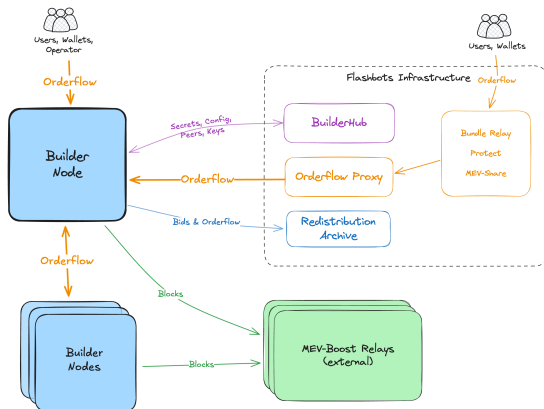
¹⁰<https://github.com/cloudflare/opaque-ea>

What it works well for

Server-to-server - where we fully control client and server

Proxies - where we can allow naive clients/server to communicate over an attested channel

My work at Flashbots (Ethereum block-building inside TEEs) where we are using a very similar protocol inspired by this



What current implementation does not support

Browser adoption will take some time. Browser APIs intentionally do not expose TLS internals needed for exporting key material.

Traversing proxies which terminate TLS. Many projects route traffic through eg: Cloudflare, often with TLS termination

Reliance on certificate authorities at the point of CVM boot

Many projects use 'nested' encryption to get around these issues

Why I think confidential computing is relevant to FOSS

Its about transparency, not only privacy.

Attestation provides a way to link open-source server-side code to running deployments.

Outline

- 1 Why are we here?
- 2 Technical Background
- 3 Open-source Implementation
- 4 TC;DU (Too Complicated; Didn't Understand)**

Take-aways

FOSS: Towards open source spec, implementation, and formal proof

Pre-handshake attestation results in **replay** and **diversion** attacks.






Intra-handshake attestation is vulnerable to **diversion** and **relay** attacks.

Post-handshake attestation is unavoidable, e.g., for re-attestation.

We believe this is a good toy to give to SEAT to live happily ever after!

Call to action: Requesting FOSS/CC community feedback

Key References

-  Birkholz, Henk, Ned Smith, Thomas Fossati, Hannes Tschofenig, and Dionna Glaze. *RATS Conceptual Messages Wrapper (CMW)*. Internet-Draft draft-ietf-rats-msg-wrap-23. Work in Progress. Internet Engineering Task Force, Dec. 2025. 41 pp. URL: <https://datatracker.ietf.org/doc/draft-ietf-rats-msg-wrap/23/>.
-  Sardar, Muhammad Usama. *Perspicuity of Attestation Mechanisms in Confidential Computing*. 2026.
-  Sardar, Muhammad Usama, Mariam Moustafa, and Tuomas Aura. “Identity Crisis in Confidential Computing: Formal Analysis of Attested TLS”. In: *Proceedings of the 21st ACM ASIA Conference on Computer and Communications Security (ACM ASIACCS 2026)*. New York, NY, USA: ACM, 2026. URL: https://www.researchgate.net/publication/398839141_Identity_Crisis_in_Confidential_Computing_Formal_Analysis_of_Attested_TLS.
-  Sardar, Muhammad Usama, Arto Niemi, Hannes Tschofenig, and Thomas Fossati. “Towards Validation of TLS 1.3 Formal Model and Vulnerabilities in Intel’s RA-TLS Protocol”. In: *IEEE Access* 12 (2024), pp. 173670–173685. DOI: 10.1109/ACCESS.2024.3497184.
-  Sullivan, Nick. *Exported Authenticators in TLS*. RFC 9261. July 2022. DOI: 10.17487/RFC9261. URL: <https://www.rfc-editor.org/info/rfc9261>.

Links to Resources

Implementation

- <https://github.com/tls-attestation/attestation-exported-authenticators>

Wiki page

- github.com/EuroProofNet/ProgramVerification/wiki/AttestedTLS

Formal proof of insecurity of pre- and intra-handshake attestation

- github.com/CCC-Attestation/formal-spec-id-crisis

Post-handshake attestation draft

- datatracker.ietf.org/doc/draft-fossati-seat-expat/

Attestation in Arm CCA and Intel TDX

- github.com/CCC-Attestation/formal-spec-TEE

Security considerations of remote attestation

- datatracker.ietf.org/doc/draft-sardar-rats-sec-cons/

IETF SEAT WG

- datatracker.ietf.org/wg/seat/about/

Technical Concepts

Validation of TLS 1.3 Key Schedule

General Approach

Weekly meetings: 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 (Scout24)
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)
and many others...