Formal Analysis of Attested TLS Protocols

Muhammad Usama Sardar

TU Dresden, Germany

November 4, 2025

Thanks to my travel sponsors!

Peg Jones

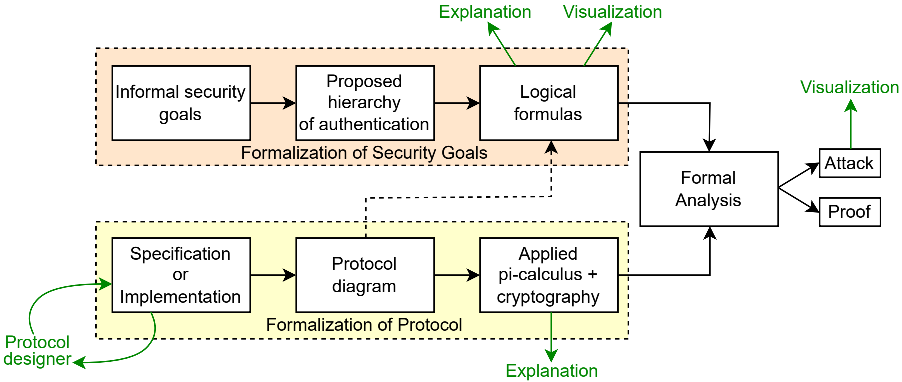
GENXT

CONFIDENTIAL COMPUTING CONSORTIUM

Teleport

Outline

Approach



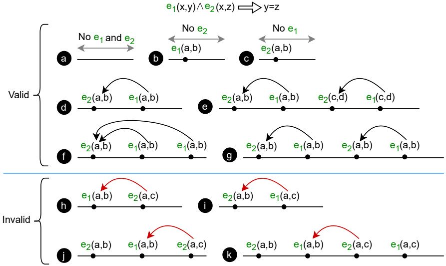
Explanation
Visualization
Informal security goals
Proposed hierarchy of authentication
Logical formulas
Formalization of Security Goals
Visualization
Formal Analysis
Attack
Proof
Specification or Implementation
Protocol diagram
Applied pi-calculus + cryptography
Formalization of Protocol
Protocol designer
Explanation

Correlation Property

- Explanation: If the client and server agree on the first parameter, then they also agree on the second parameter.

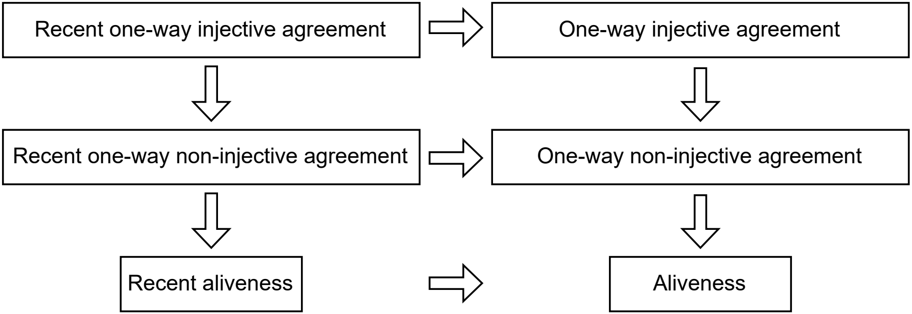
$$
\begin{aligned}
&\forall\, x, y, z. \\
&action(e_1(x, y)) \wedge \\
&action(e_2(x, z)) \rightarrow \\
&y = z
\end{aligned}
\tag{1}
$$

- Visualization



General Hierarchy of One-way Authentication

Level	Property	Agreement		
		Server's identity	Protocol parameters	One-to-one correspondence
1	Aliveness	✓	×	×
2	One-way non-injective agreement	✓	✓	×
3	One-way injective agreement	✓	✓	✓



Comparison with Lowe's Hierarchy

Definition

Weak agreement: We say that a protocol guarantees to an initiator A weak agreement with another agent B if, whenever A (acting as initiator) completes a run of the protocol, apparently with responder B, then B has previously been running the protocol, apparently with A.

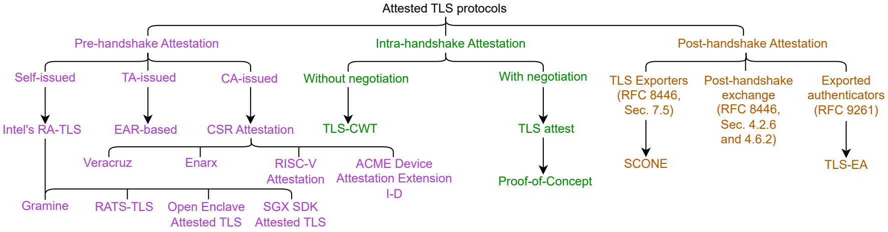
Discussion: What does the RG think?

1. Weak agreement is stronger than a one-way agreement as it requires the responder to agree on the initiator's identity
2. No placement of actions e_1 and e_2 that can satisfy Formula 2, with e_1 still representing an action for a client role and e_2 still representing an action for a server role.

$$\forall\ id_s, id_c.\ action(e_1(id_s, id_c)) \rightarrow action(e_2(id_s, id_c)) \tag{2}$$

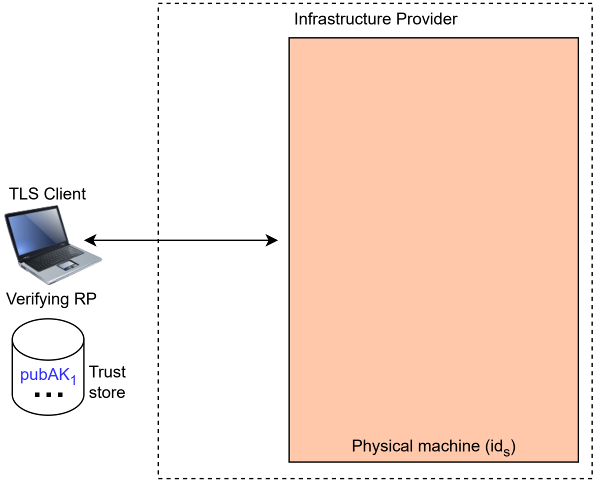
Outline

Scope



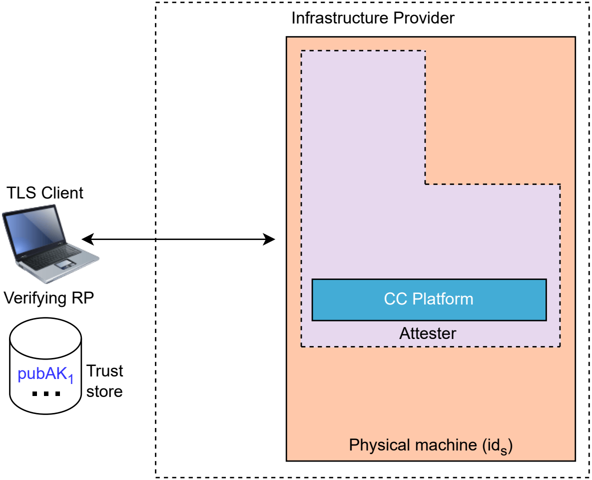
- Confidential Computing
- TLS Server as RATS Attester
- Pre- and intra-handshake attestation
- Post-handshake attestation will have the same goals (it is WiP)

System Model (TLS Server as RATS Attester)



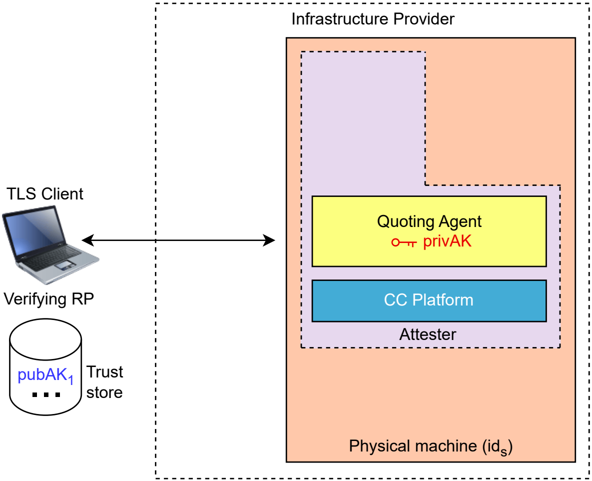
- AK = Attestation Key

System Model (TLS Server as RATS Attester)



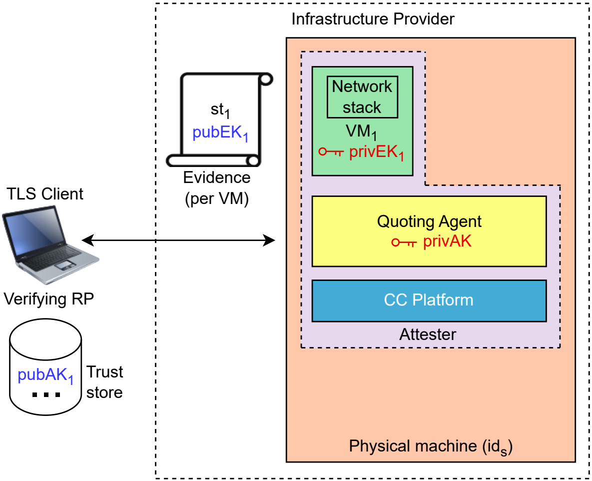
- AK = Attestation Key

System Model (TLS Server as RATS Attester)



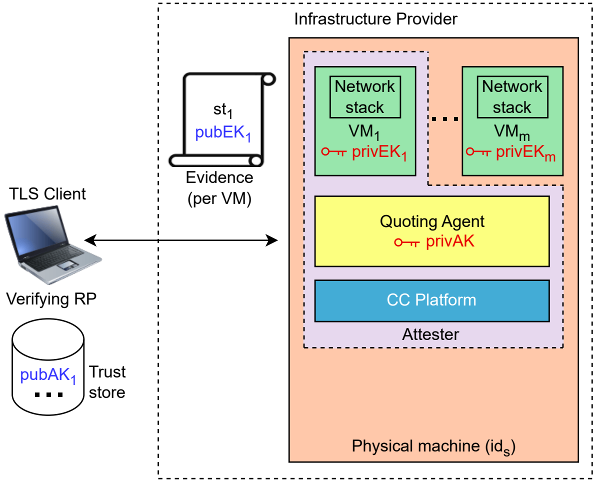
- AK = Attestation Key

System Model (TLS Server as RATS Attester)



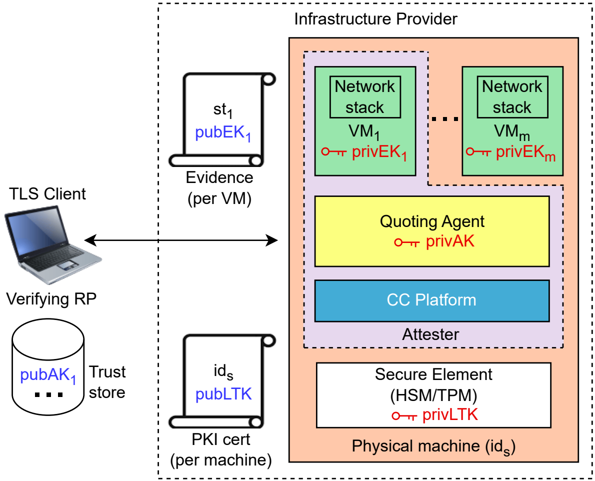
- AK = Attestation Key
- EK = Ephemeral Key

System Model (TLS Server as RATS Attester)



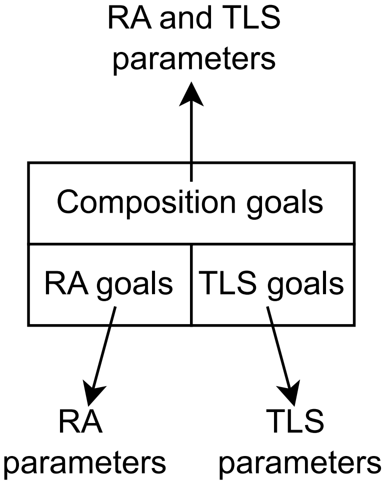
- AK = Attestation Key
- EK = Ephemeral Key

System Model (TLS Server as RATS Attester)



- AK = Attestation Key
- EK = Ephemeral Key
- LTK = Long-Term Key

General Approach for Security Goals



Informal Security Goals

- Remote Attestation
  - G-RA1: Integrity of Evidence
  - G-RA2: Freshness of Evidence
    - Binding Evidence to a specific RA interaction
    - Recentness of Evidence generation
  - G-RA3: Establishment of connection with privAK known to adversary
  - G-RA4: Establishment of connection with privEK known to adversary

Informal Security Goals

- Remote Attestation
  - G-RA1: Integrity of Evidence
  - G-RA2: Freshness of Evidence
    - Binding Evidence to a specific RA interaction
    - Recentness of Evidence generation
  - G-RA3: Establishment of connection with privAK known to adversary
  - G-RA4: Establishment of connection with privEK known to adversary

- Standard TLS properties
  - G-TLS1: Establishment of connection with `client_write_key` known to adversary
  - G-TLS2: Server authentication

Informal Security Goals

- Remote Attestation
  - G-RA1: Integrity of Evidence
  - G-RA2: Freshness of Evidence
    - Binding Evidence to a specific RA interaction
    - Recentness of Evidence generation
  - G-RA3: Establishment of connection with privAK known to adversary
  - G-RA4: Establishment of connection with privEK known to adversary

- Standard TLS properties
  - G-TLS1: Establishment of connection with `client_write_key` known to adversary
  - G-TLS2: Server authentication

- Composition goals
  - G-C1: Evidence is generated by the same server that is authenticated
    - Binding Evidence to a specific TLS connection: $g^{xy}, htsc, atsc$
  - G-C2: Agreement of all Remote Attestation and TLS parameters

Threat Model

- Standard TLS threat model (weak DH, weak hash etc.)
- Long-term Key (privLTK) may be leaked
- Ephemeral Key (privEK) may be leaked
- Attestation Key (privAK) may be leaked (e.g., via Foreshadow[1])

[1] Van Bulck, Minkin, Weisse, Genkin, Kasikci, Piessens, Silberstein, Wenisch, Yarom, and Strackx, "Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution", 2018.

Threat Model

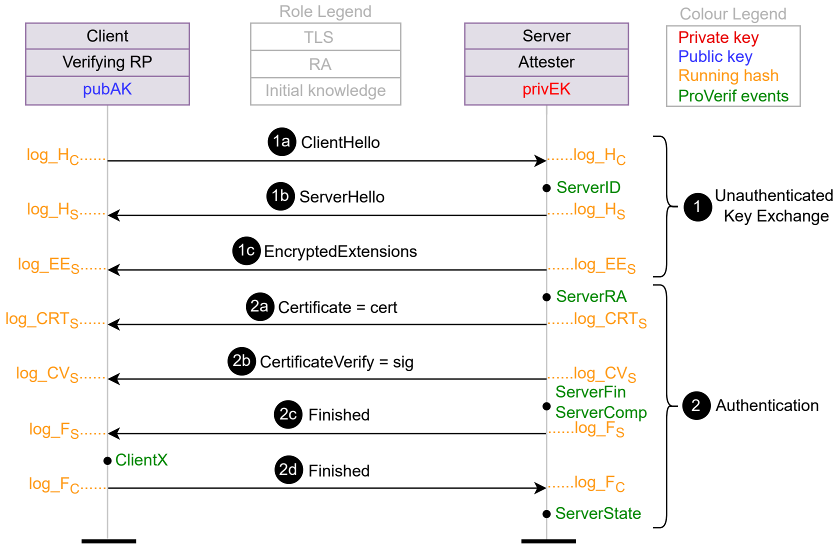
- Standard TLS threat model (weak DH, weak hash etc.)
- Long-term Key (privLTK) may be leaked
- Ephemeral Key (privEK) may be leaked
- Attestation Key (privAK) may be leaked (e.g., via Foreshadow[1])
- Assumptions (complementary to protocol design)
  - No weaknesses in the generation of Claims
  - Verifying RP provisioned with Reference Values
  - Verifying RP provisioned with pubAK
  - Verifying RP provisioned with pubCA
  - TOCTOU attacks out of the scope

[1] Van Bulck, Minkin, Weisse, Genkin, Kasikci, Piessens, Silberstein, Wenisch, Yarom, and Strackx, "Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution", 2018.

Security Goals and Properties

Security goal	Security property
G-RA1 holds unless `LAK` or `WH`	One-way non-injective agreement
G-RA2 holds unless `LAK` or `WH`	Recent one-way injective agreement
G-RA3 holds unless `LAK`	One-way non-injective agreement
G-RA4 holds unless `LAK`, `LEK`, `WDH` or `WH`	One-way non-injective agreement
G-TLS1 holds unless `LAK`, `LEK`, `WDH`, `WH` or `WE`	One-way non-injective agreement
G-TLS2 holds unless `LAK`, `LEK`, `WDH`, `WH` or `WE`	Recent one-way non-injective agreement
G-C1a should hold	Correlation property
G-C1b should hold	Correlation property
G-C1c should hold	Correlation property
G-C2 holds unless `LAK`, `LEK`, `WDH`, `WH` or `WE`	Recent one-way injective agreement

Events for Formalization of Security Goals



Abstract Security Properties (without assumptions) in ProVerif

Security goal	e_1	e_2	General Arguments
G-RA1	`ClientRA`	`ServerRA`	`RAparams`
G-RA2	`ClientRA`	`ServerRA`	`RAparams,na`
G-RA3	`ClientFin & attacker`	-	`pubAK`
G-RA4	`ClientFin & attacker`	-	`pubEK`
G-TLS1	`ClientFin & attacker`	-	`kc`
G-TLS2a	`ClientID & ServerID`	-	`ID,gxy`
G-TLS2b	`ClientFin`	`ServerFin`	`TLSparams`
G-C1a	`ClientState & ServerState`	-	`ev,gxy`
G-C1b	`ClientState & ServerState`	-	`ev,kch`
G-C1c	`ClientState & ServerState`	-	`ev,kc`
G-C2	`ClientComp`	`ServerComp`	`RAparams,TLSparams`

Results for Pre- and Intra-handshake Attestation

- Pre-handshake attestation: Interoperable RA-TLS (IRA-TLS)
- Intra-handshake attestation: draft-fossati-tls-attestation (TLS-a)

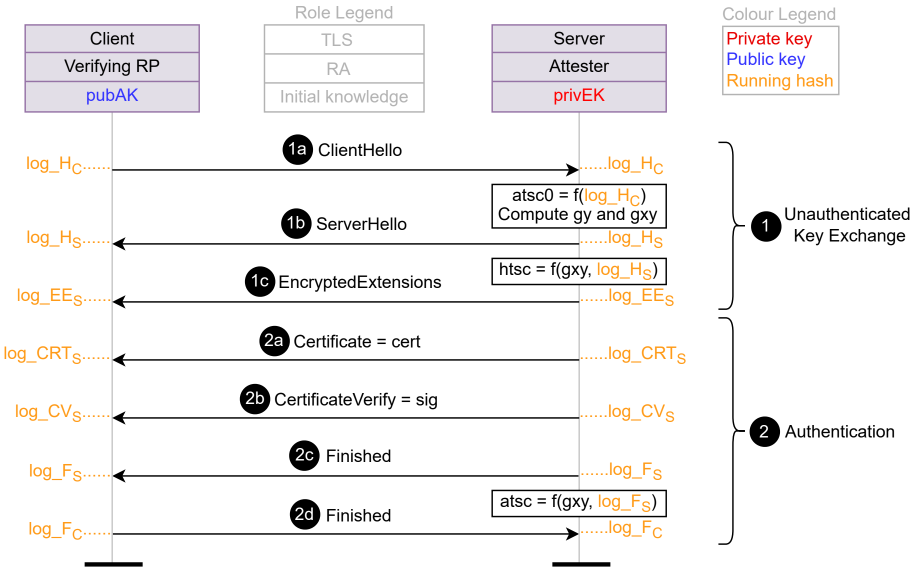
Security goal	IRA-TLS	TLS-a
G-RA1: Integrity of Evidence	✓	✓
G-RA2: Freshness of Evidence	×	✓
G-RA3: Protection of Attestation Keys	✓	✓
G-RA4: Protection of Ephemeral Keys	✓	✓
G-TLS1: Protection of Client's Write Key	✓	✓
G-TLS2.1: Server Authentication	×	×
G-TLS2.2: Server Authentication	×	×
G-C1a: Binding Evidence to `gxy`	×	×
G-C1b: Binding Evidence to `htsc`	×	×
G-C1c: Binding Evidence to `atsc`	×	×
G-C2: Agreement of all parameters	×	×

Results for Proposed Solutions in Intra-handshake Attestation

- Sol.1: Modify CertificateVerify message
- Sol.2: Two CertificateVerify messages
- Sol.3: New signature algorithm
- Sol.4: New Attestation message
- Sol.5: Modify CertificateVerify message + define new exporter

Security goal	Sol.1	Sol.2	Sol.3	Sol.4	Sol.5
G-RA1: Integrity of Evidence	✓	✓	✓	✓	✓
G-RA2: Freshness of Evidence	✓	✓	✓	✓	✓
G-RA3: Protection of Attestation Keys	✓	✓	✓	✓	✓
G-RA4: Protection of Ephemeral Keys	✓	✓	✓	✓	✓
G-TLS1: Protection of Client's Write Key	✓	✓	✓	✓	✓
G-TLS2.1: Server Authentication	✓	✓	✓	✓	✓
G-TLS2.2: Server Authentication	✓	✓	✓	✓	✓
G-C1a: Binding Evidence to `gxy`	×	×	×	×	✓
G-C1b: Binding Evidence to `htsc`	×	×	×	×	✓
G-C1c: Binding Evidence to `atsc`	×	×	×	×	×
G-C2: Agreement of all parameters	✓	✓	✓	✓	✓

Strong Binding vs. Relay of Evidence (Abstracted)



Summary

- Protocol-designer-in-the-loop approach with explanations and visualizations
- Identified subtle problems in Lowe's hierarchy and proposed general hierarchy of one-way authentication
- To my knowledge, SEAT WG is the first WG with formal analysis explicitly mentioned in the charter.
- Key insights
  - Need infrastructure identity to prevent diversion attacks
  - Cryptographic binding of RA and TLS is critical to prevent relay attacks
  - Pre- and intra-handshake attestation are not suitable choices for standardization.
  - One of the most challenging protocols of the IETF: formal analysis is critical to the success of WG.
- Requesting feedback on shared pre-prints
- WiP: post-handshake attestation, i.e., draft-fossati-seat-expat
- WiP: draft-ietf-tls-extended-key-update: formal analysis work is blocked. The authors are not following good practices.

ACK

Co-authors

- 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)

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)
- Pavel Nikonorov (GENXT)
- 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)

Peg Jones

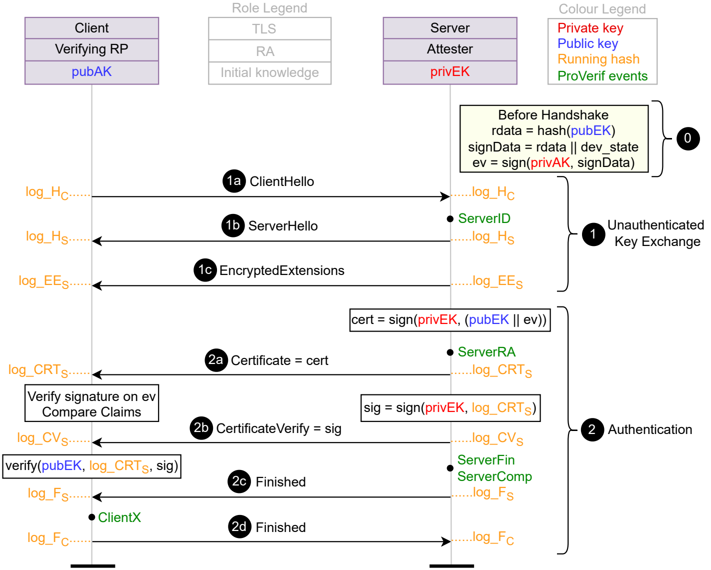
GENXT

CONFIDENTIAL COMPUTING CONSORTIUM

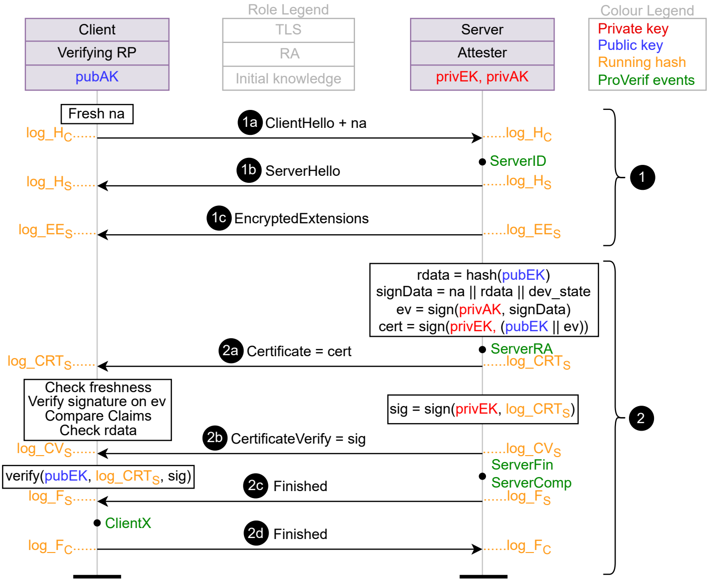
Teleport

Outline

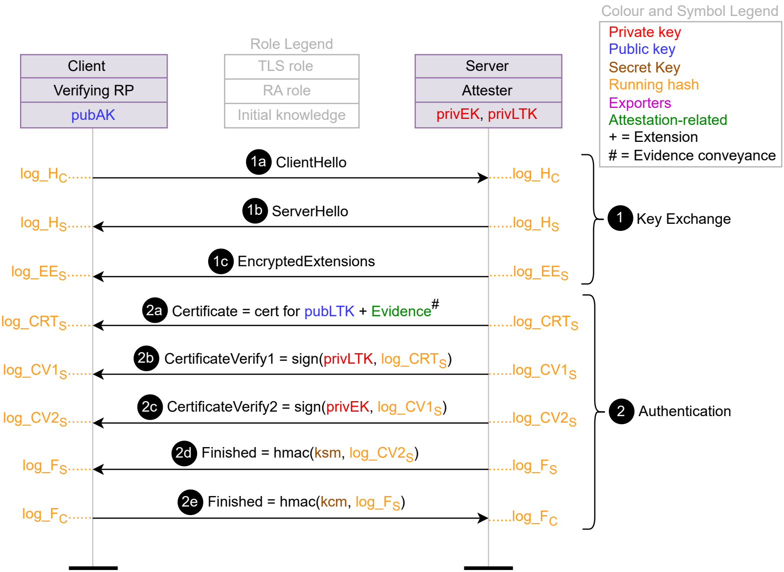
IRA-TLS


Role Legend
TLS
RA
Initial knowledge
Client
Verifying RP
pubAK
Server
Attester
privEK
Colour Legend
Private key
Public key
Running hash
ProVerif events
Before Handshake
rdata = hash(pubEK)
signData = rdata || dev_state
ev = sign(privAK, signData)
0
1a ClientHello
log_HC
1b ServerHello
ServerID
log_HS
1c EncryptedExtensions
log_EES
1 Unauthenticated Key Exchange
cert = sign(privEK, (pubEK || ev))
2a Certificate = cert
ServerRA
log_CRTS
Verify signature on ev
Compare Claims
sig = sign(privEK, log_CRTS)
2b CertificateVerify = sig
log_CVS
verify(pubEK, log_CRTS, sig)
ServerFin
ServerComp
2c Finished
log_FS
ClientX
2d Finished
log_FC
2 Authentication

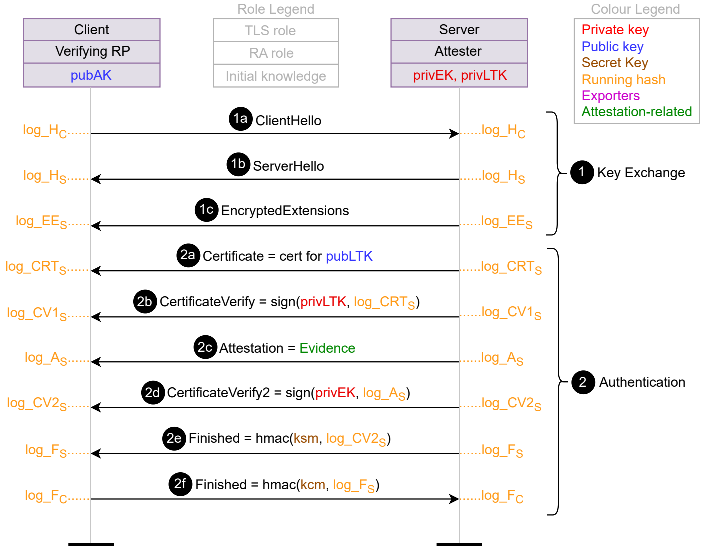
TLS-a


Role Legend
TLS
RA
Initial knowledge
Colour Legend
Private key
Public key
Running hash
ProVerif events
Client
Verifying RP
pubAK
Server
Attester
privEK, privAK
Fresh na
1a ClientHello + na
log_H$_C$......
......log_H$_C$
ServerID
1b ServerHello
log_H$_S$......
......log_H$_S$
1c EncryptedExtensions
log_EE$_S$......
......log_EE$_S$
1
rdata = hash(pubEK)
signData = na || rdata || dev_state
ev = sign(privAK, signData)
cert = sign(privEK, (pubEK || ev))
2a Certificate = cert
ServerRA
log_CRT$_S$......
......log_CRT$_S$
Check freshness
Verify signature on ev
Compare Claims
Check rdata
sig = sign(privEK, log_CRT$_S$)
2b CertificateVerify = sig
log_CV$_S$......
......log_CV$_S$
ServerFin
ServerComp
verify(pubEK, log_CRT$_S$, sig)
2c Finished
log_F$_S$......
......log_F$_S$
ClientX
2d Finished
log_F$_C$......
......log_F$_C$
2

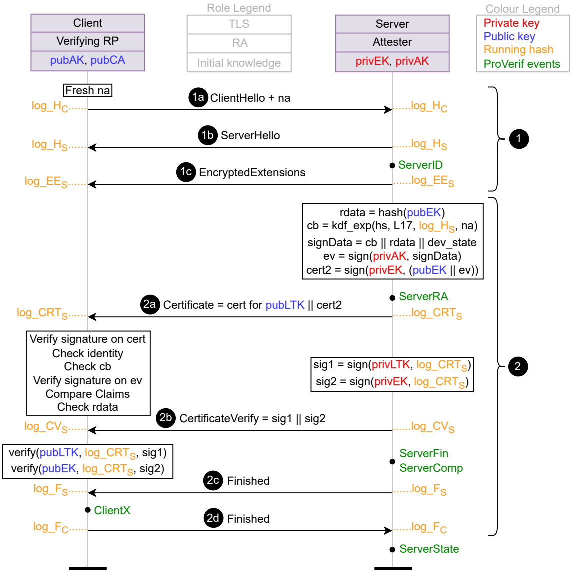
Sol. 2: Two CertificateVerify Messages



Sol. 4: New Attestation Message



Sol. 5: Cryptographic Binding



Sol. 5 (cont.): Cryptographic Binding