Standardization of Attested TLS Protocols

Muhammad Usama Sardar[1,2]

[1]TU Dresden, Germany

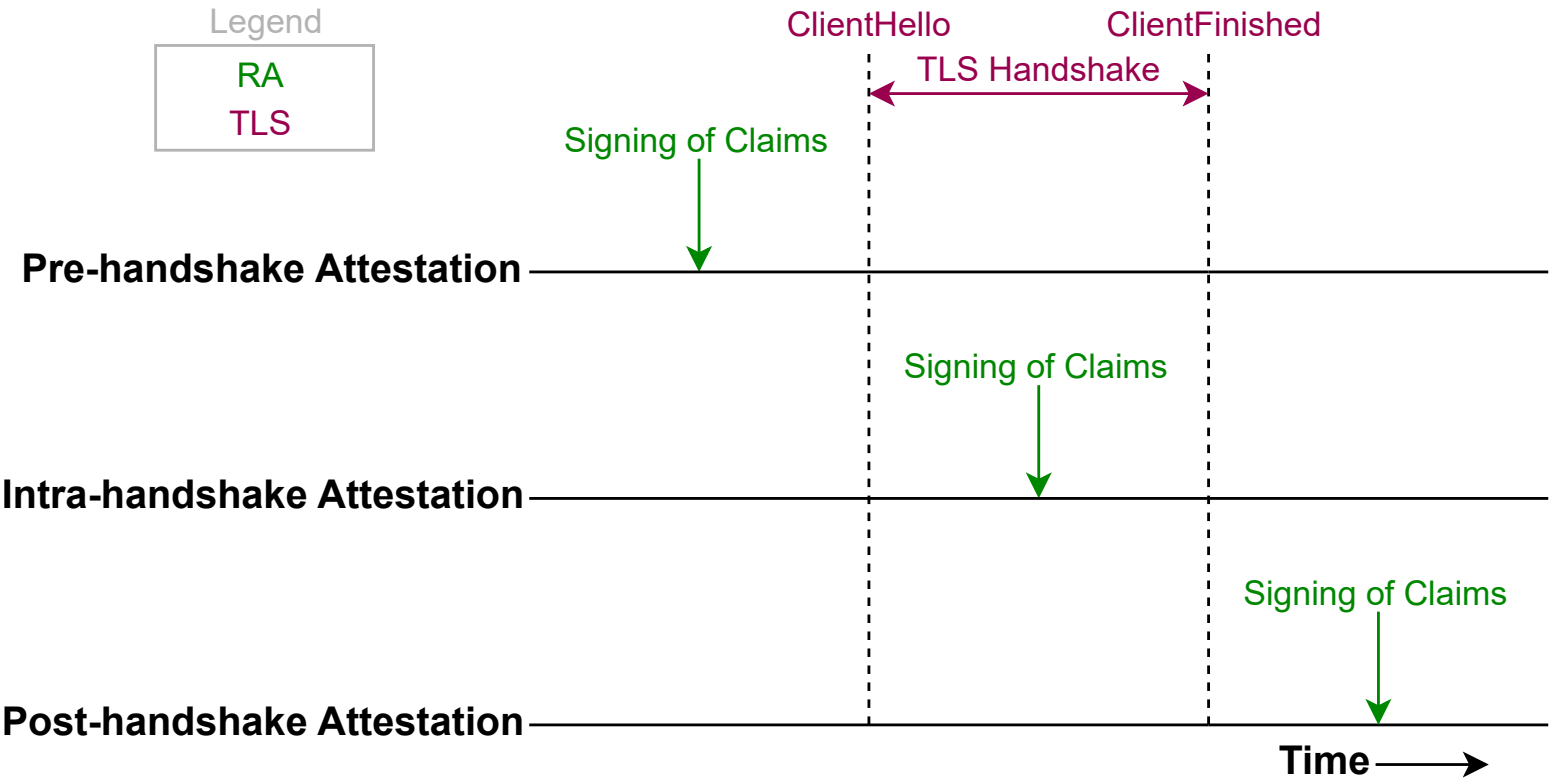
[2]Co-chair, Trusted Research Environment (TRE) Open Suite,
Global Alliance for Genomics and Health (GA4GH)

December 12, 2025

Outline

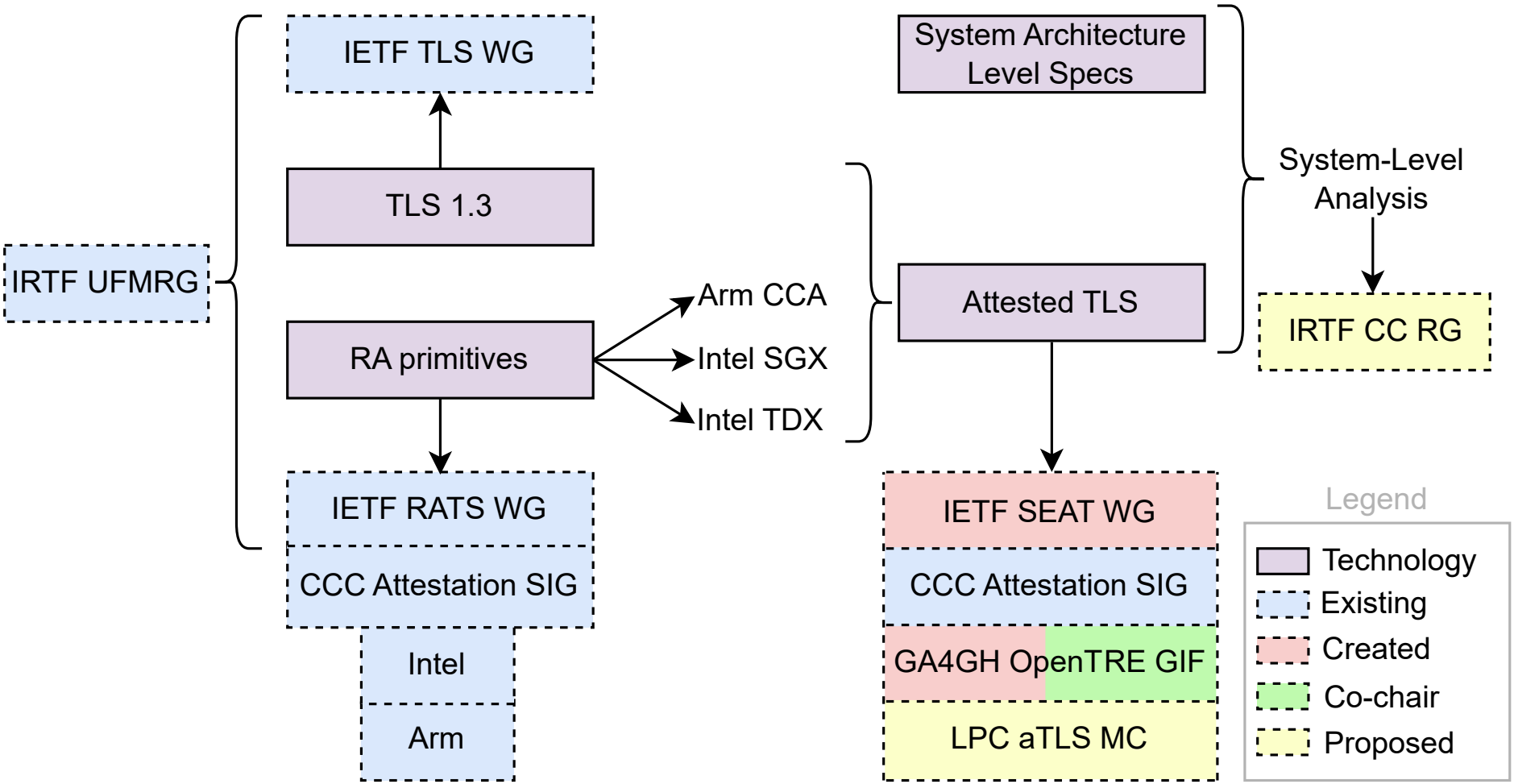
Quick Reminder from LPC'24

- Link to LPC'24 presentation

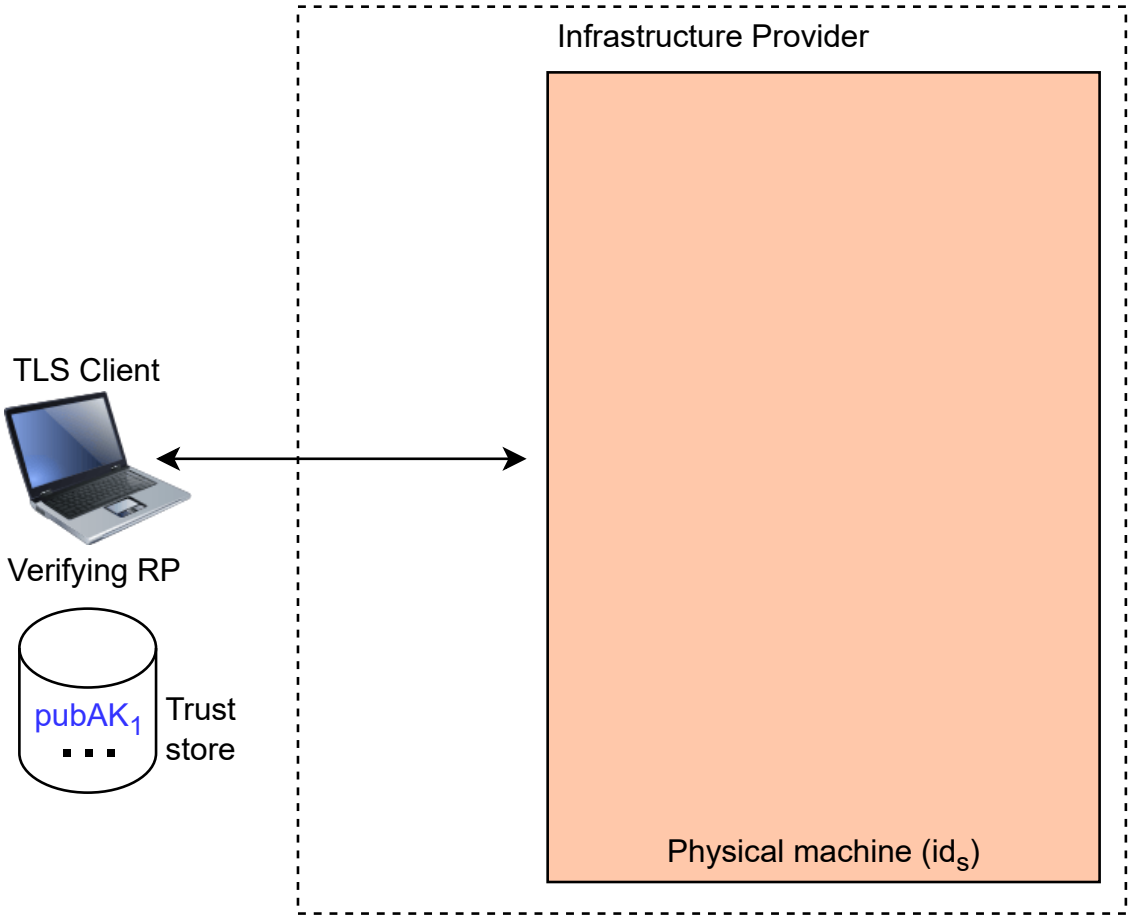


- Pre-handshake: Intel's RA-TLS/Interoperable RA-TLS (IRA-TLS)
- Intra-handshake: draft-fossati-tls-attestation (TLS-a)
- Post-handshake: draft-fossati-seat-expat

Big Picture

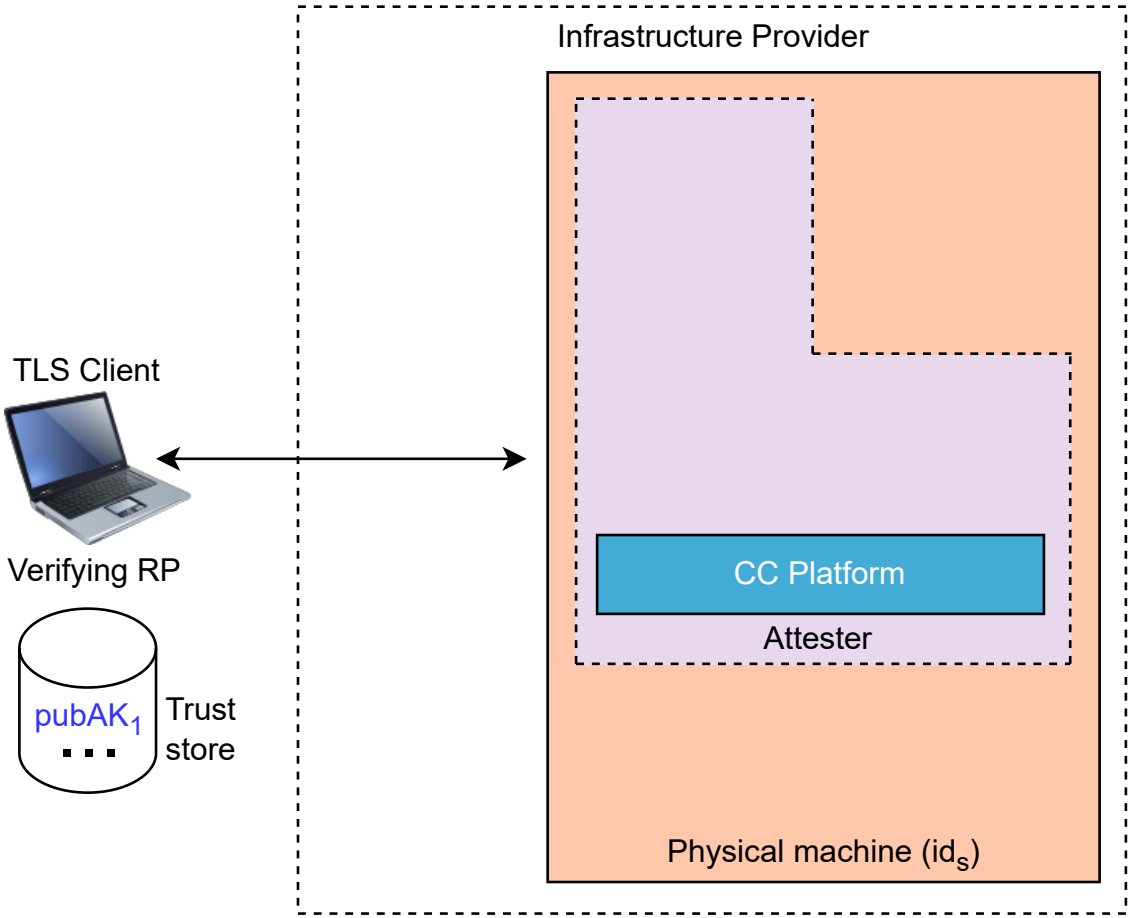

IETF TLS WG
TLS 1.3
IRTF UFMRG
RA primitives
Arm CCA
Intel SGX
Intel TDX
IETF RATS WG
CCC Attestation SIG
Intel
Arm
System Architecture Level Specs
System-Level Analysis
Attested TLS
IRTF CC RG
IETF SEAT WG
CCC Attestation SIG
GA4GH OpenTRE GIF
LPC aTLS MC
Legend
Technology
Existing
Created
Co-chair
Proposed

System Model (TLS Server as RATS Attester)



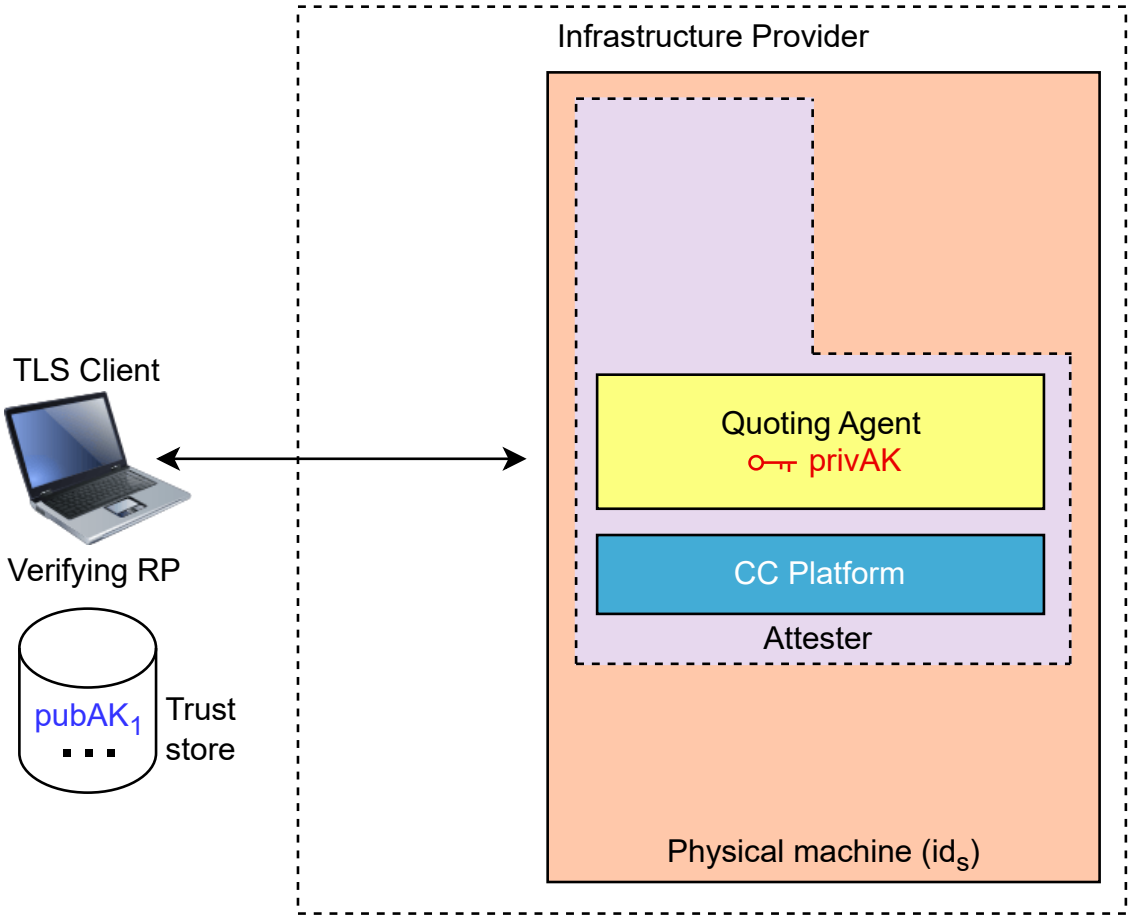
- AK = Attestation Key

System Model (TLS Server as RATS Attester)



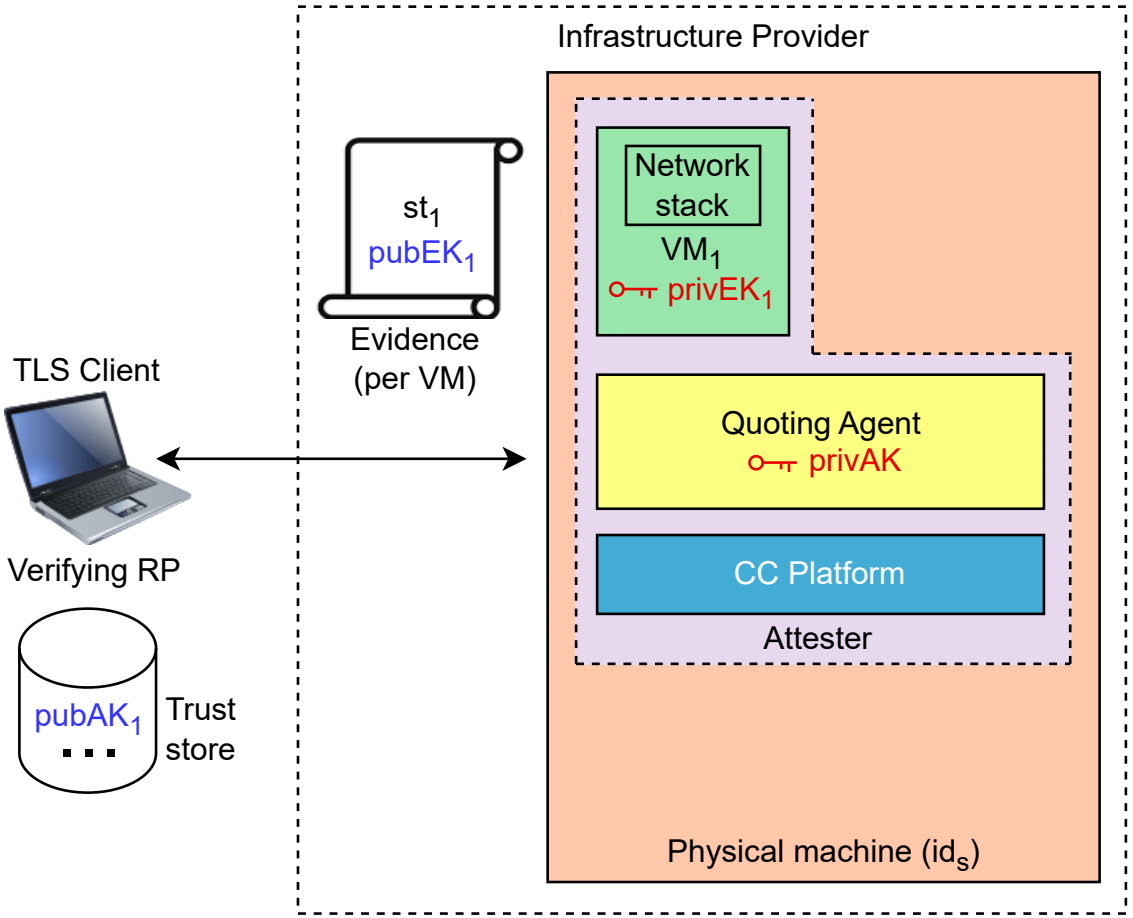
- AK = Attestation Key

System Model (TLS Server as RATS Attester)



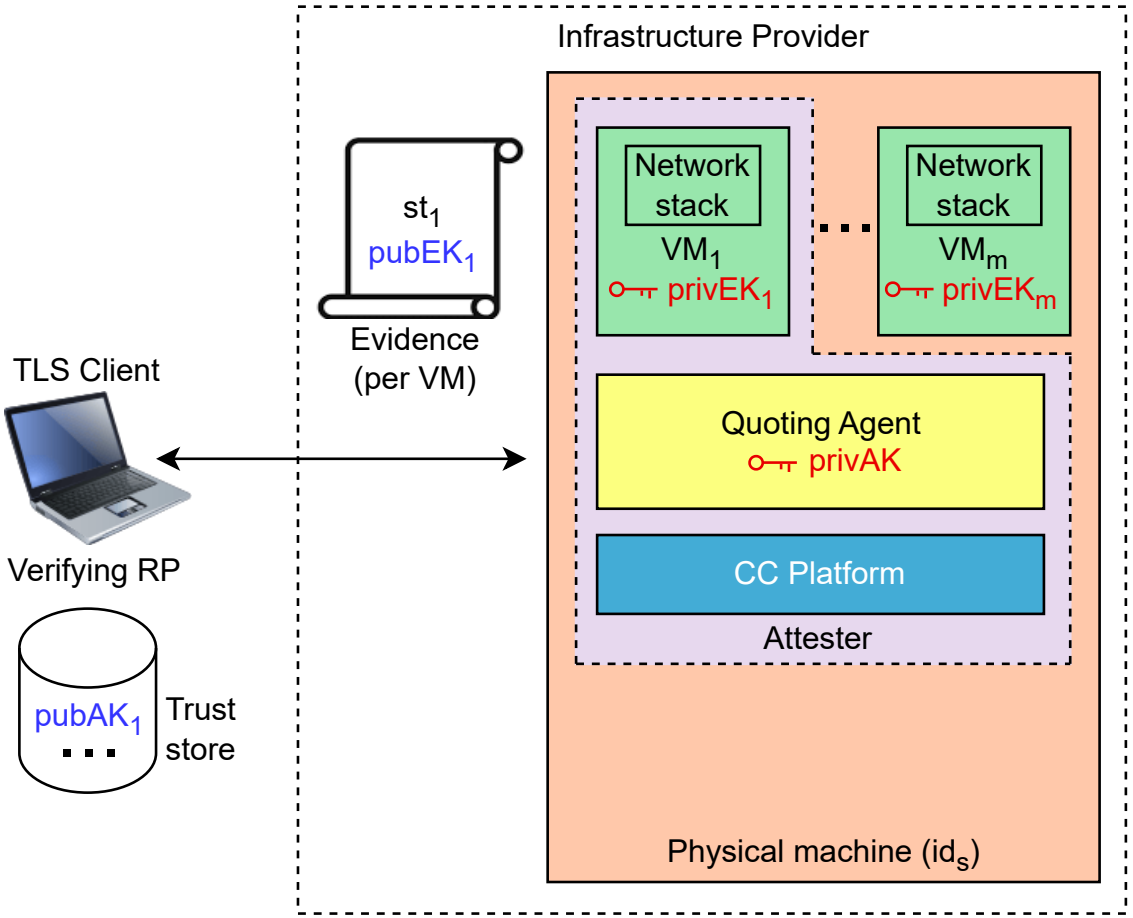
- AK = Attestation Key

System Model (TLS Server as RATS Attester)



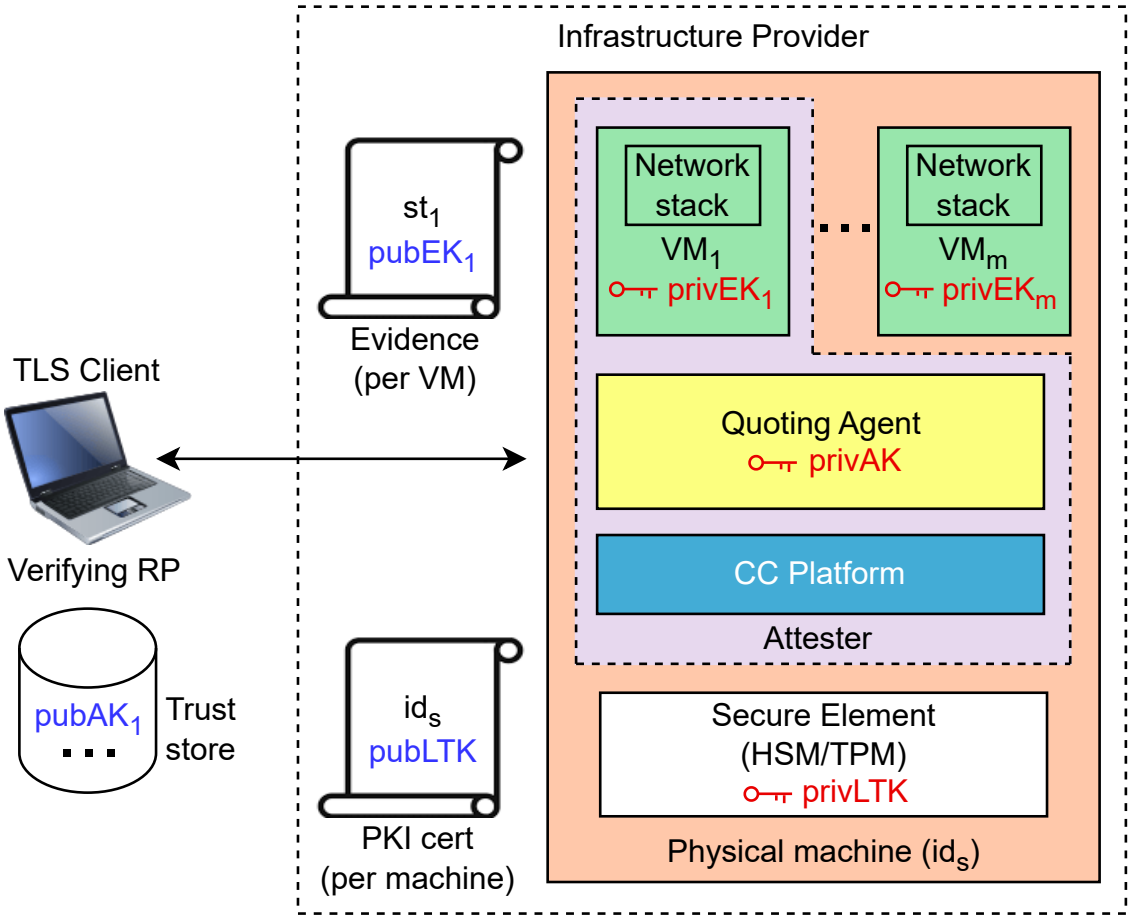
- AK = Attestation Key
- EK = Ephemeral Key

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

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

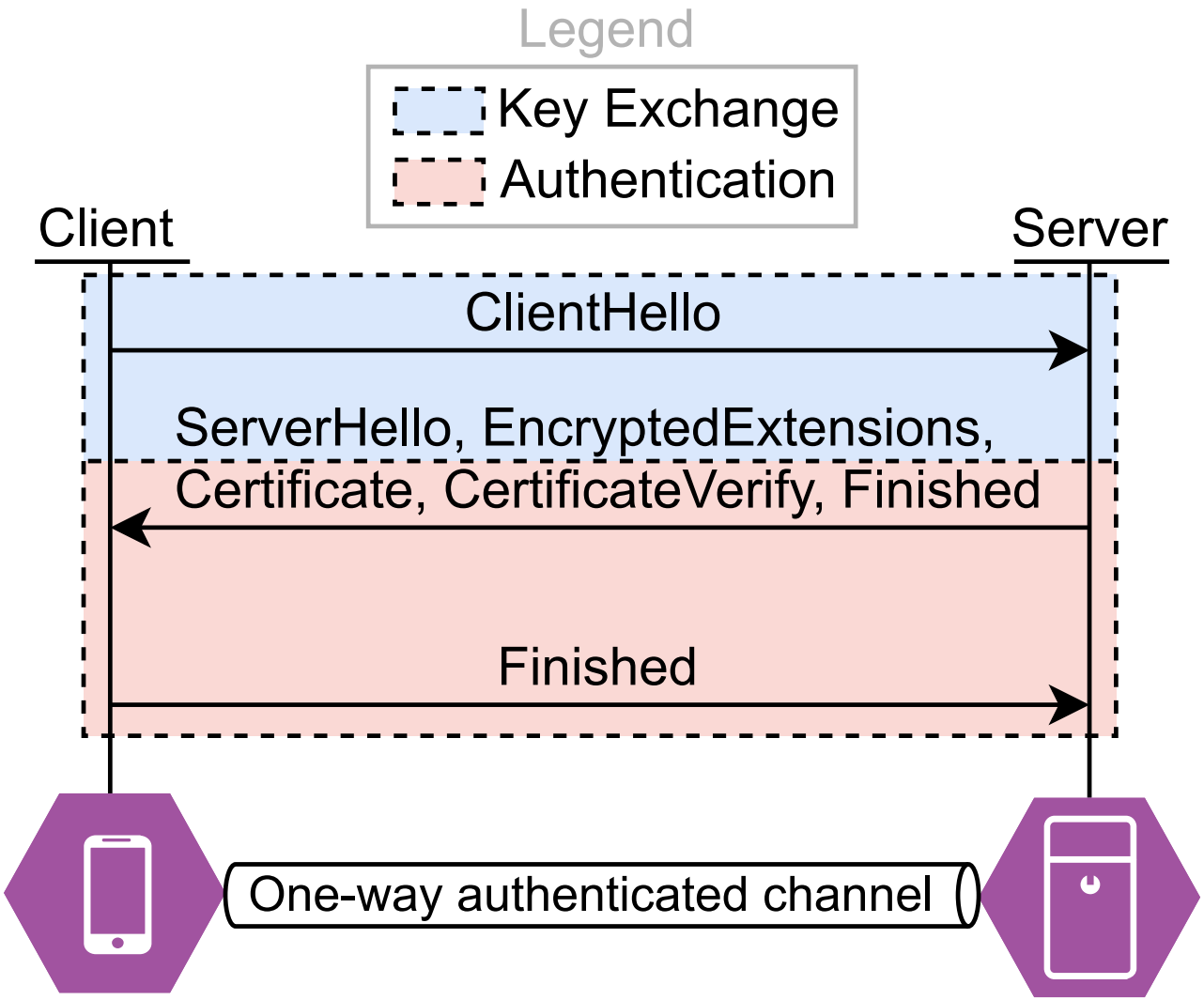
Informal Security Goals

- 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
    - Correlating Evidence to a specific TLS connection: g^{xy}, *htsc*, *atsc*
  - G-C2: Agreement of all Remote Attestation and TLS parameters

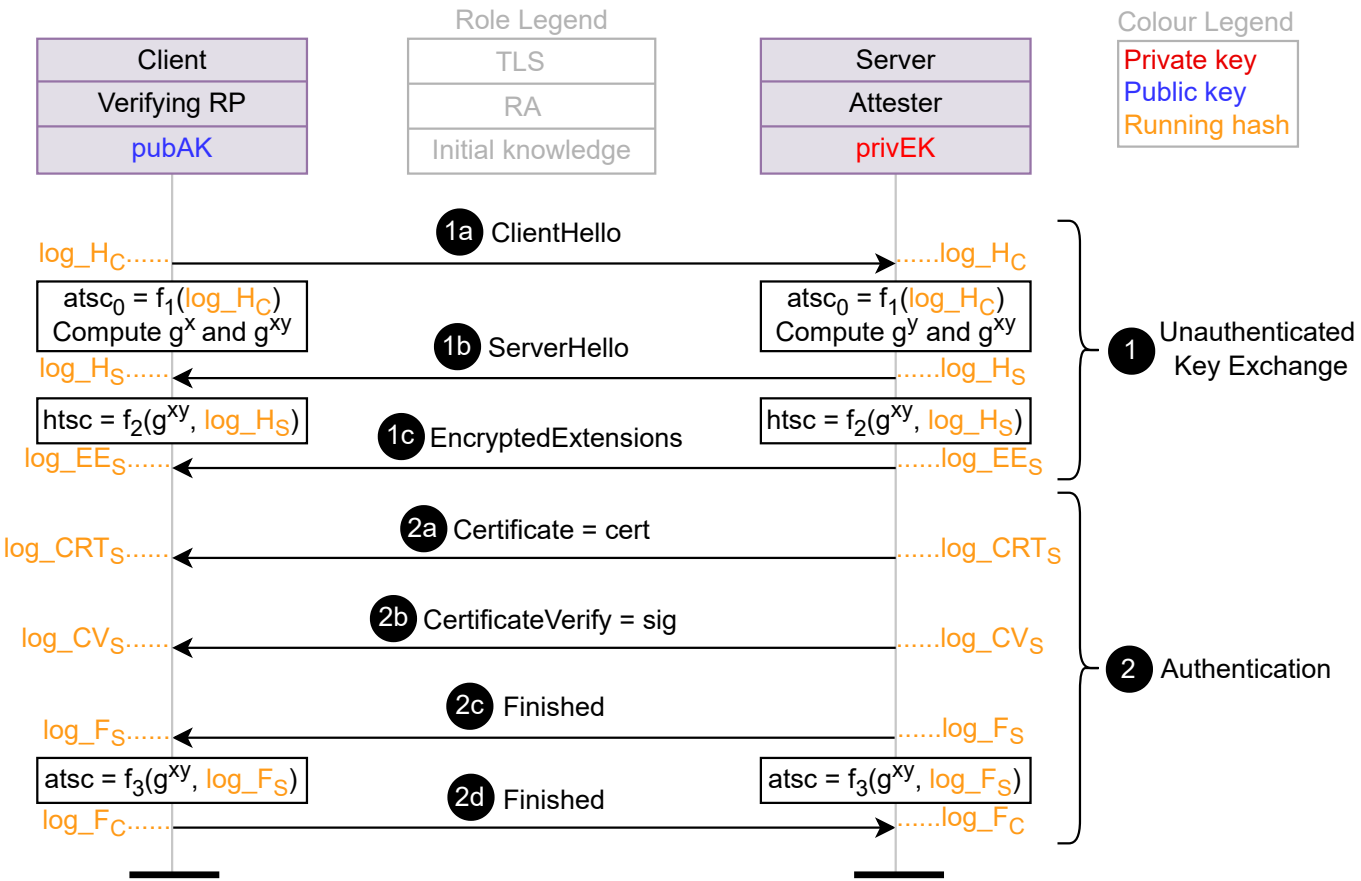
Informal Security Goals

- 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
    - Correlating Evidence to a specific TLS connection: g^{xy}, *htsc*, *atsc*
  - G-C2: Agreement of all Remote Attestation and TLS parameters
- Discussion: Any other (verifiable) security goals?

Standard TLS 1.3

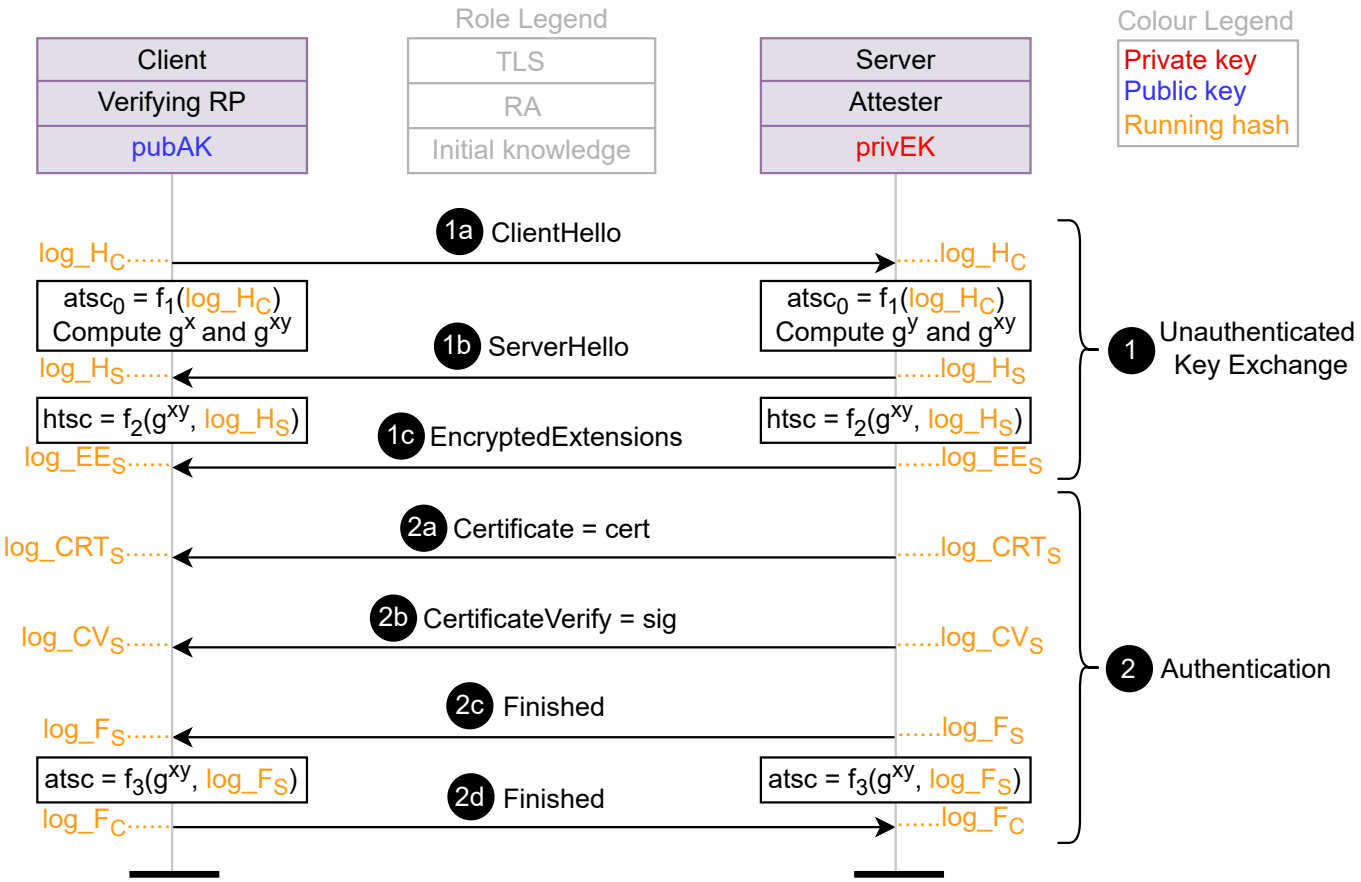


Strong Binding vs. Relay of Evidence (Abstracted)



- Discussion: Correlating Evidence to `htsc` vs. `atsc`
  - Running hash $\Longrightarrow$ `atsc` transitively includes all contributions in `htsc`
  - `atsc` provides stronger binding and avoids relay attacks.

Strong Binding vs. Relay of Evidence (Abstracted)



- `htsc`: used for encryption of clientFinished message (2d).
  - Irrelevant for security goals
- `atsc`: used for encryption of application data (client's secret, e.g., decryption key)
  - Relevant for security goals

Outline

Results for Pre- and Intra-handshake Attestation[1]

- 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: Correlation of Evidence to `gxy`	×	×
G-C1b: Correlation of Evidence to `htsc`	×	×
G-C1c: Correlation of Evidence to `atsc`	×	×
G-C2: Agreement of all parameters	×	×

✓ = satisfied; × = not satisfied.

[1] Details in `https://datatracker.ietf.org/doc/slides-124-ufmrg-formal-analysis-of-attested-tls-protocols`

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: Correlation of Evidence to `gxy`	×	×	×	×	✓
G-C1b: Correlation of Evidence to `htsc`	×	×	×	×	✓
G-C1c: Correlation of Evidence to `atsc`	×	×	×	×	×
G-C2: Agreement of all parameters	✓	✓	✓	✓	✓

Summary

- Pre- and intra-handshake attestation (draft-fossati-tls-attestation) are not suitable choices for standardization.

Summary

- Pre- and intra-handshake attestation (draft-fossati-tls-attestation) are not suitable choices for standardization.
- Key insights from formal analysis
  - Need infrastructure identity to prevent diversion attacks
  - Need cryptographic binding of RA and TLS to prevent relay attacks
  - Need VM identity to prevent replication attacks
  - One of the most challenging protocols of the IETF: formal analysis is critical to the success.

Summary

- Pre- and intra-handshake attestation (draft-fossati-tls-attestation) are not suitable choices for standardization.
- Key insights from formal analysis
  - Need infrastructure identity to prevent diversion attacks
  - Need cryptographic binding of RA and TLS to prevent relay attacks
  - Need VM identity to prevent replication attacks
  - One of the most challenging protocols of the IETF: formal analysis is critical to the success.
- Part of formal analysis accepted at AsiaCCS
- WiP: post-handshake attestation, i.e., draft-fossati-seat-expat
  - Formal analysis in ProVerif
  - Implementation in RustIs (Peg Jones)
  - Implementation in BoringSSL (Pavel Nikonorov)
- Questions for discussion:
  - Any other (verifiable) security goals?
  - `htsc` vs. `atsc`?
  - Any other solution?

Summary

- Pre- and intra-handshake attestation (draft-fossati-tls-attestation) are not suitable choices for standardization.
- Key insights from formal analysis
  - Need infrastructure identity to prevent diversion attacks
  - Need cryptographic binding of RA and TLS to prevent relay attacks
  - Need VM identity to prevent replication attacks
  - One of the most challenging protocols of the IETF: formal analysis is critical to the success.
- Part of formal analysis accepted at AsiaCCS
- WiP: post-handshake attestation, i.e., draft-fossati-seat-expat
  - Formal analysis in ProVerif
  - Implementation in RustIs (Peg Jones)
  - Implementation in BoringSSL (Pavel Nikonorov)
- Questions for discussion:
  - Any other (verifiable) security goals?
  - `htsc` vs. `atsc`?
  - Any other solution?
- BoF today at 15:45 https://lpc.events/event/19/contributions/2299/

Links to Resources

- Wiki page
- Formal proof of insecurity of pre- and intra-handshake attestation
- Post-handshake attestation draft
- Attestation in Arm CCA and Intel TDX
- Security considerations of remote attestation
- IETF SEAT WG
- Technical Concepts
- Validation of TLS 1.3 Key Schedule
- General Approach
- Weekly meetings

ACK

Co-authors

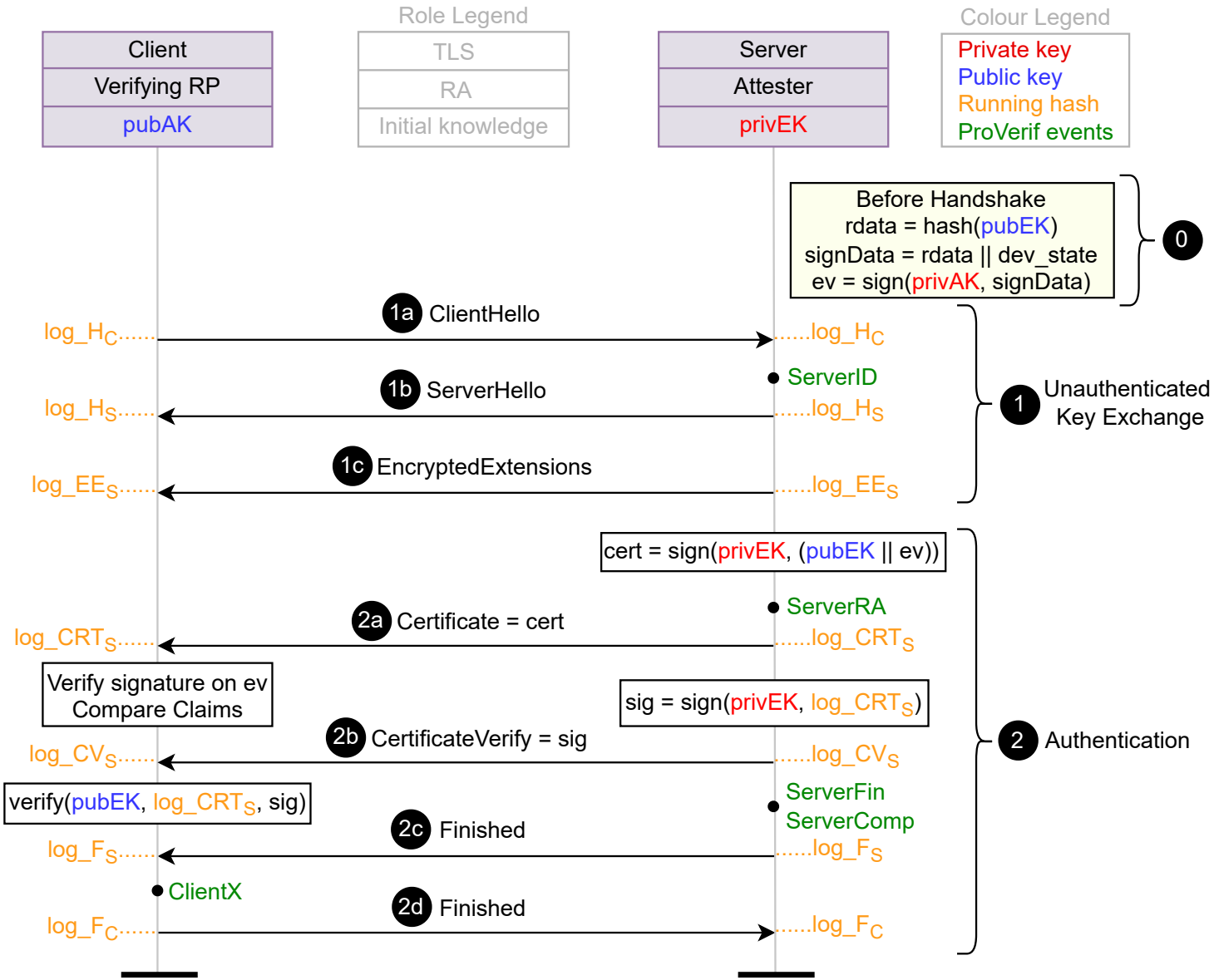
- Jean-Marie Jacquet (University of Namur)
- Ionut Mihalcea (Arm)
- Thomas Fossati (Linaro)
- Arto Niemi (Huawei)
- Hannes Tschofenig (University of Applied Sciences Bonn-Rhein-Sieg and Siemens)
- Simon Frost (Arm)
- Ned Smith (Intel)
- Carsten Weinhold (Barkhausen Institut)
- Michael Roitzsch (Barkhausen 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)
- Yuning Jiang (Huawei Partners)
- Meiling Chen (China Mobile)
- Peter Chunchi Liu (Huawei)

Contributors

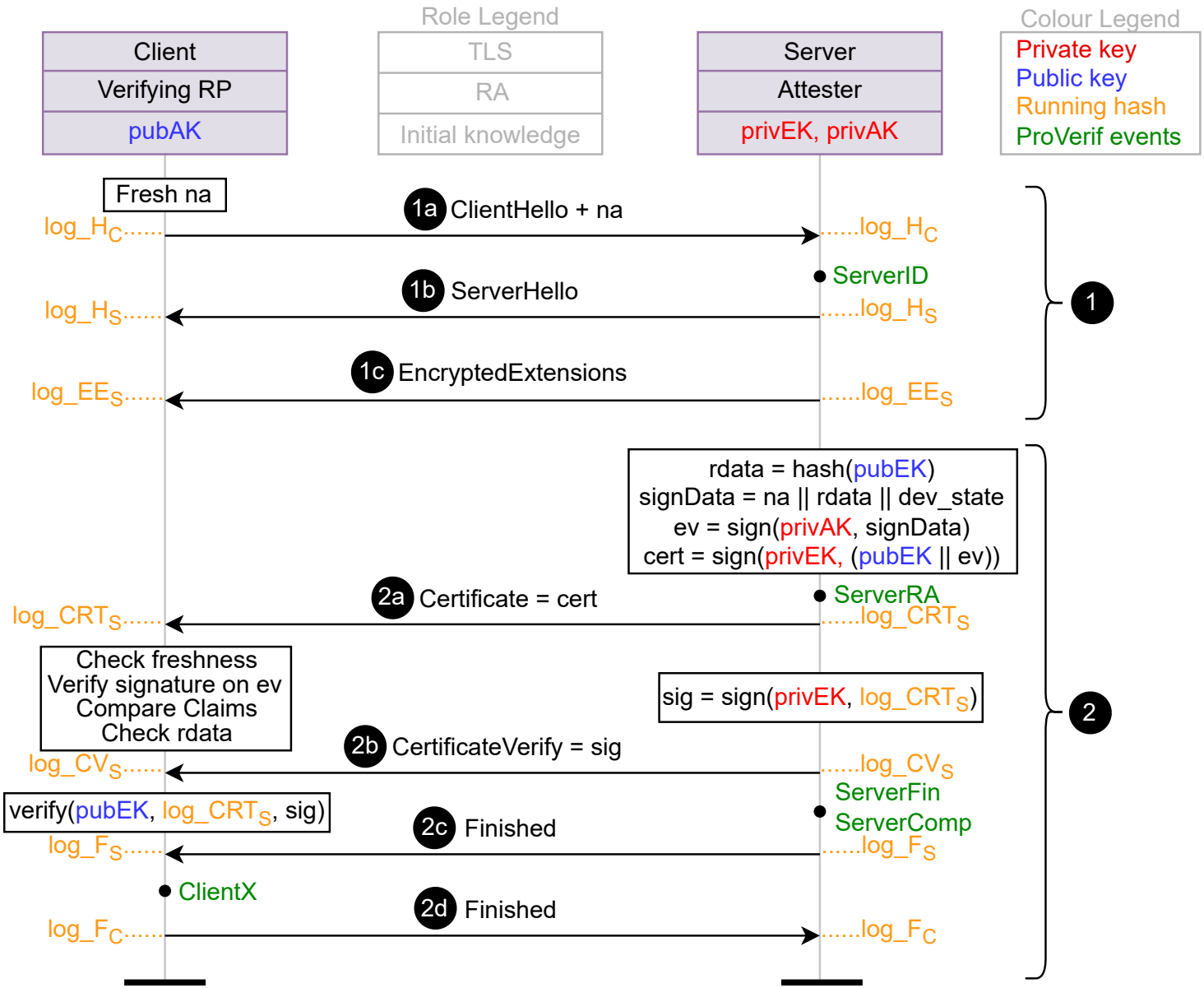
- Eric Rescorla (Independent)
- Laurence Lundblade (Security Theory LLC)
- Göran Selander (Ericsson AB)
- Marco Tiloca (RISE AB)
- Richard Barnes (Cloudflare)
- Giridhar Mandyam (AMD)
- Christopher Patton (Cloudflare)
- Pavel Nikonorov (GENXT)
- Dionna Amalie Glaze (Google)
- Bob Beck (Google)
- Mike 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)
- Paul Wouters (Aiven)
- Dennis Jackson (Mozilla)
- Peg Jones (Flashbots)

Outline

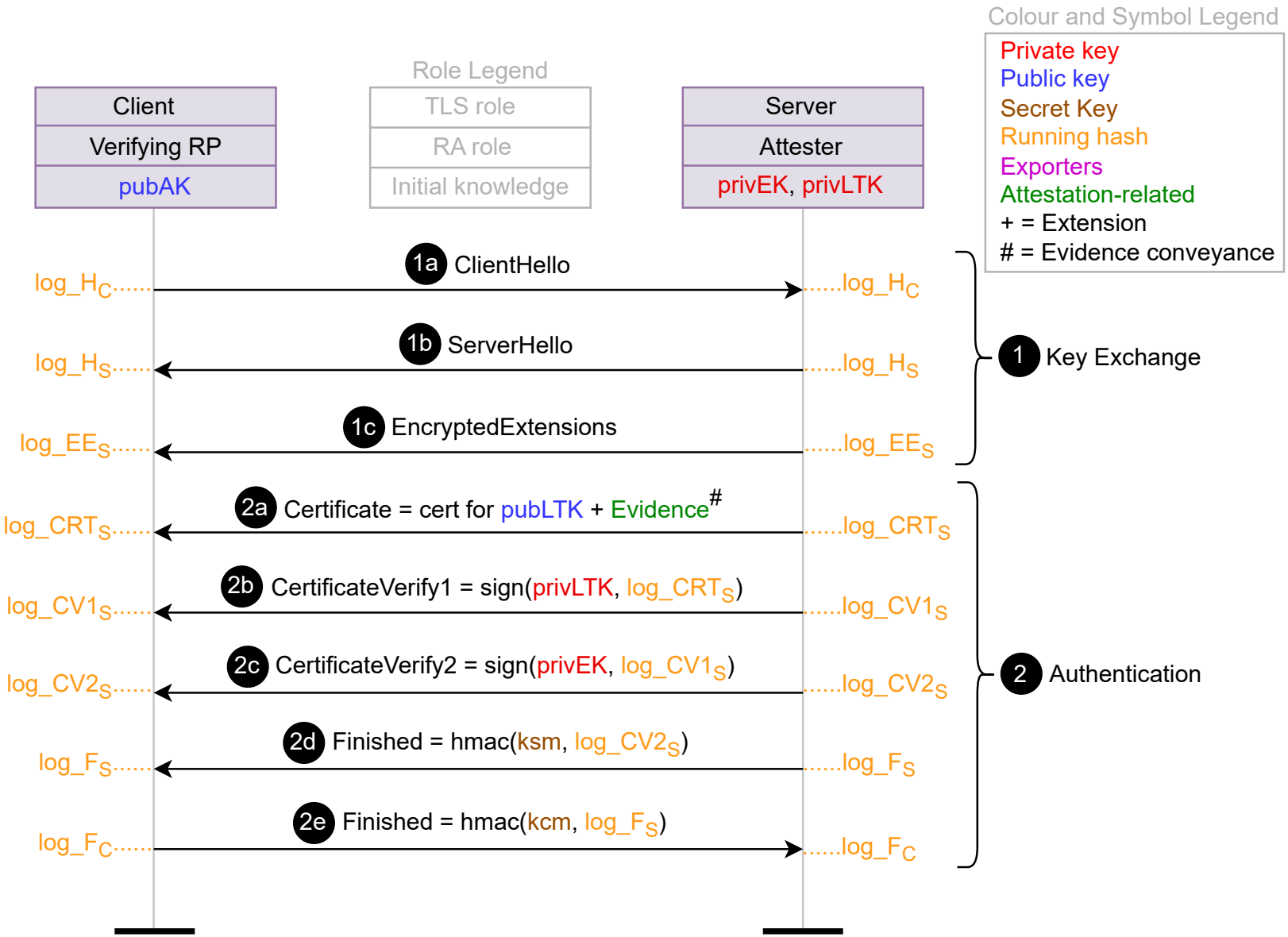
IRA-TLS


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

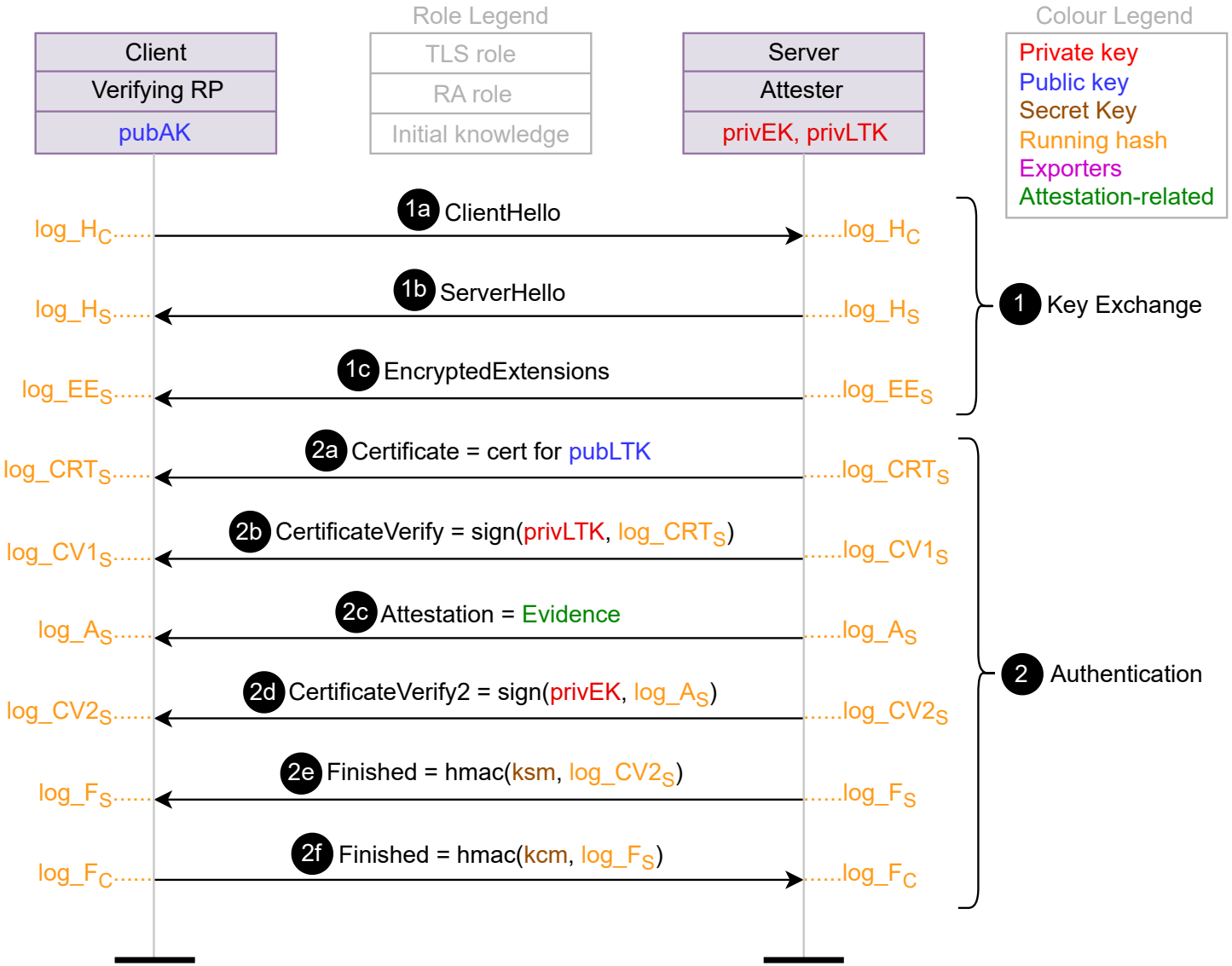
TLS-a


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
log_H_C
1a ClientHello + na
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

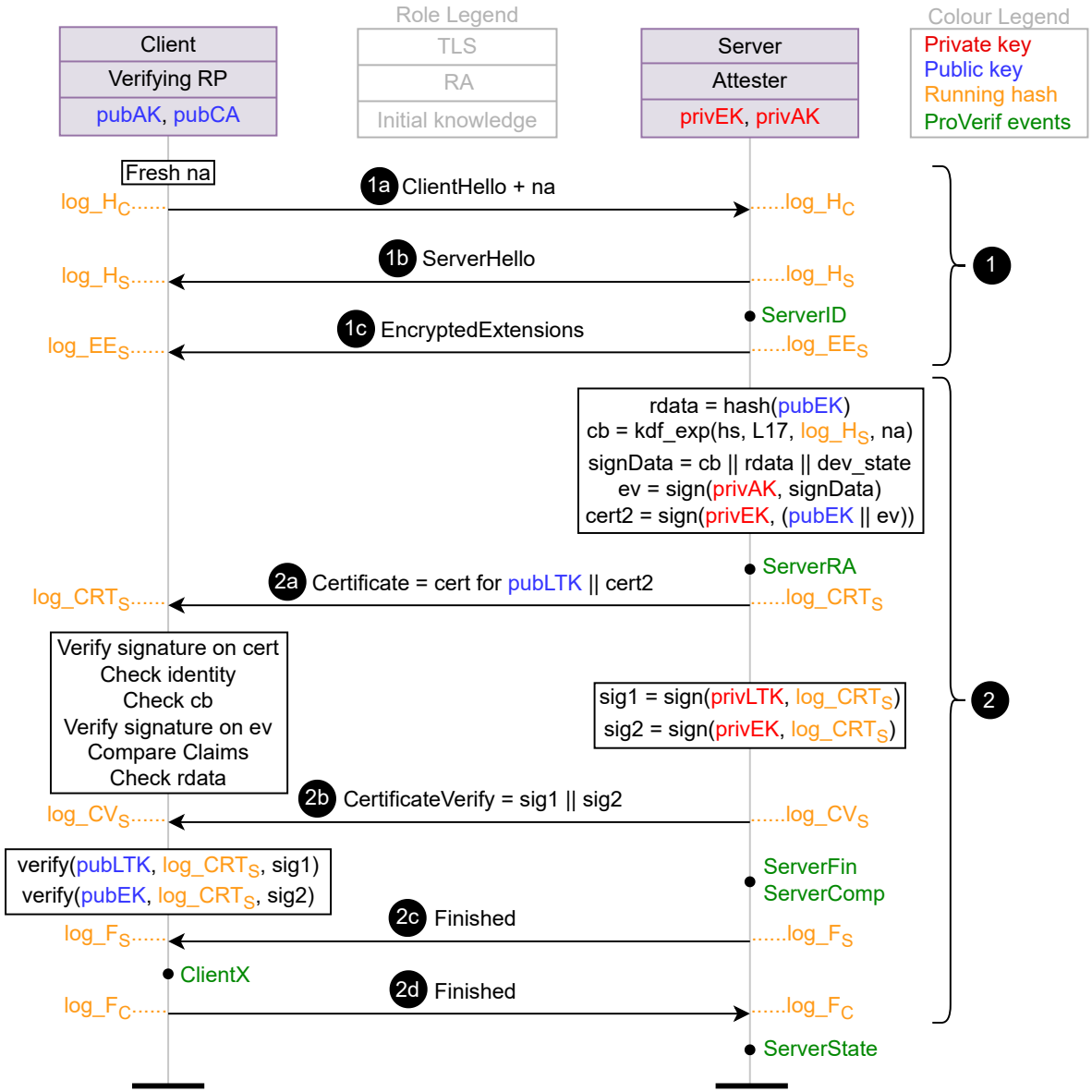
Sol. 2: Two CertificateVerify Messages



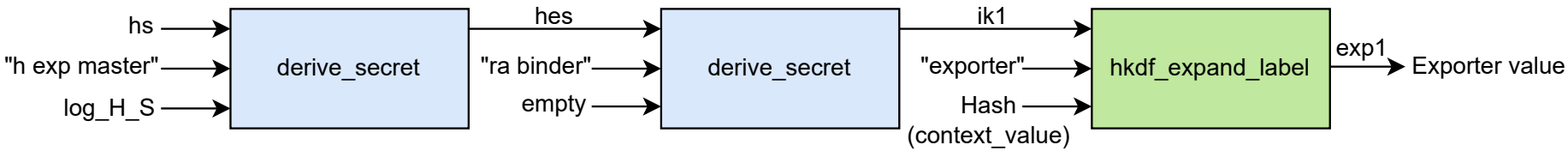
Sol. 4: New Attestation Message



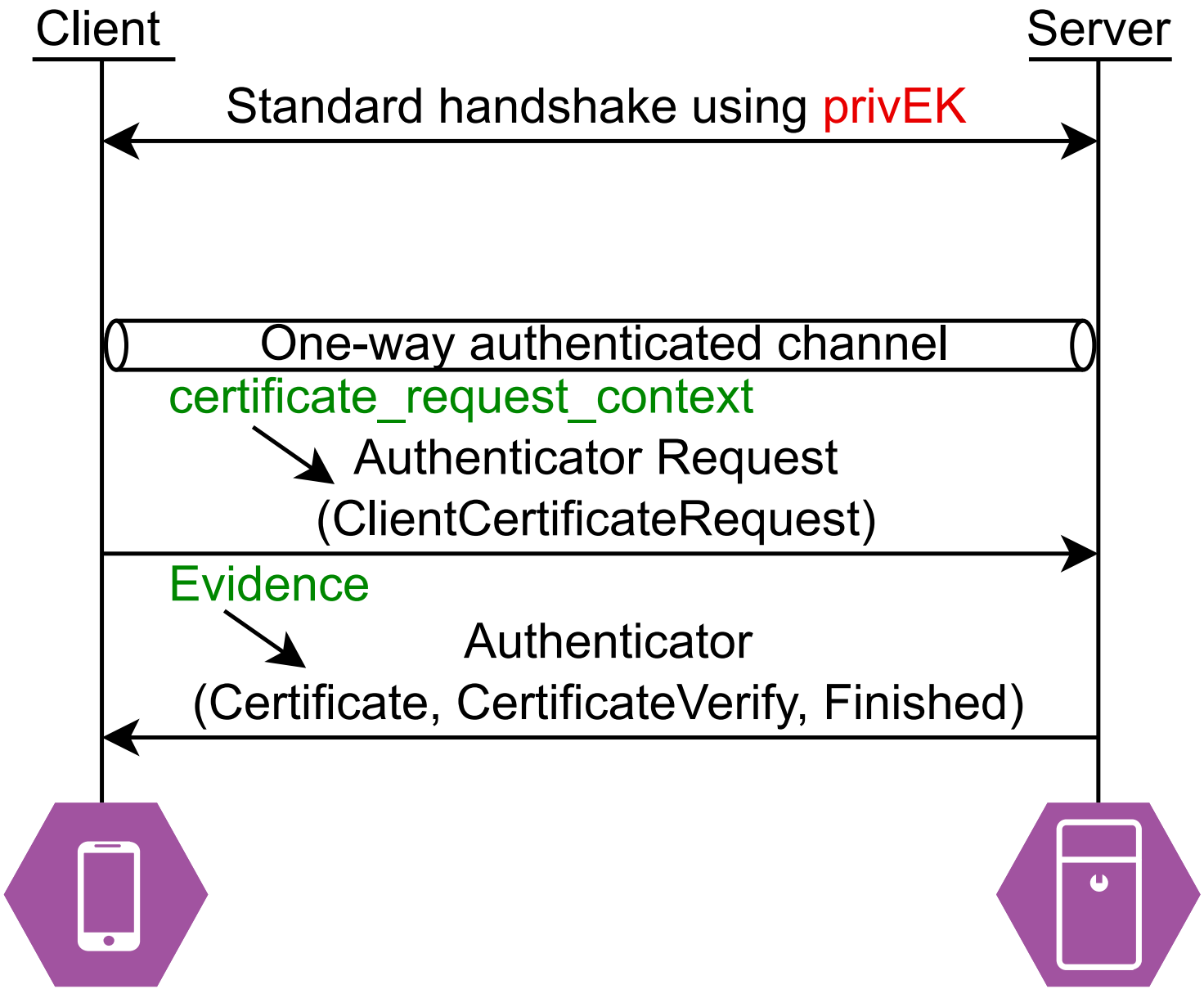
Sol. 5: Cryptographic Binding



Sol. 5 (cont.): Cryptographic Binding



Proposal for Post-handshake Attestation (RFC 9261)



Post-handshake Flow

1. Authenticator Request
   - Unique *certificate_request_context* within connection
2. Evidence based on this context and *Exported Keying Material (EKM)*
3. Authenticator
   - Certificate message extended with Evidence
   - CertificateVerify as in RFC 9261
   - Finished as in RFC 9261
4. Validation: additionally appraise Evidence