Towards Formal Analysis of Attested TLS

Muhammad Usama Sardar[1], Arto Niemi[2], Hannes Tschofenig[3] and Thomas Fossati[4]

[1]TU Dresden, Germany

[2]Huawei Technologies, Helsinki, Finland

[3]University of Applied Sciences Bonn-Rhein-Sieg and Siemens, Germany

[4]Linaro, Lausanne, Switzerland

July 24, 2024

Agenda

TLS[1]

- Good for network security

[1] https://datatracker.ietf.org/doc/html/rfc8446

TLS[1]

- Good for network security
- Not good for endpoint security

[1] https://datatracker.ietf.org/doc/html/rfc8446

TLS[1]

- Good for network security
- Not good for endpoint security
  - Keys

[1] https://datatracker.ietf.org/doc/html/rfc8446

TLS[1]

- Good for network security
- Not good for endpoint security
  - Keys
  - Software

[1] https://datatracker.ietf.org/doc/html/rfc8446

TLS[1]

- Good for network security
- Not good for endpoint security
  - Keys
  - Software
  - Platform

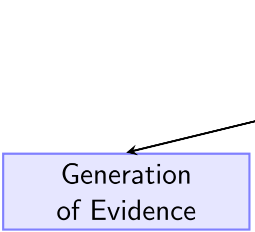
[1] https://datatracker.ietf.org/doc/html/rfc8446

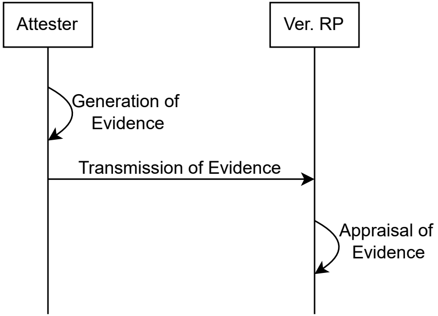
TLS[1]

- Good for network security
- Not good for endpoint security
  - Keys
  - Software
  - Platform
- Use case: Confidential Computing

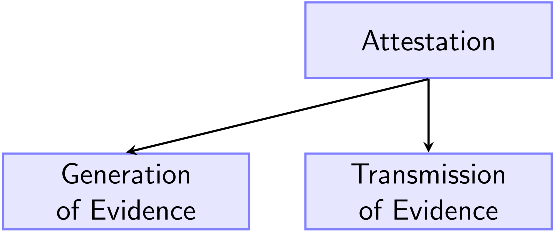
[1] https://datatracker.ietf.org/doc/html/rfc8446

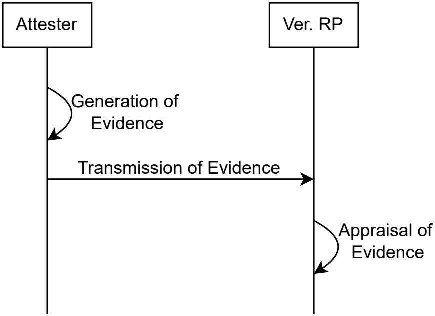
Remote Attestation (Key idea)



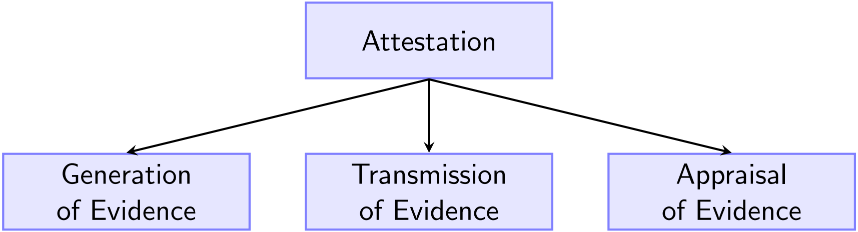


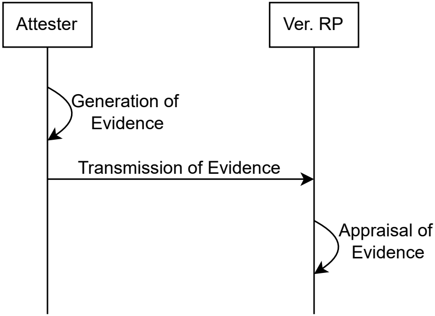
Remote Attestation (Key idea)



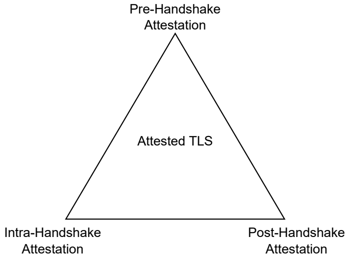


Remote Attestation (Key idea)

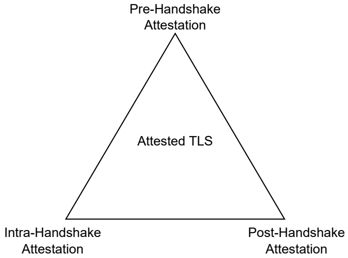


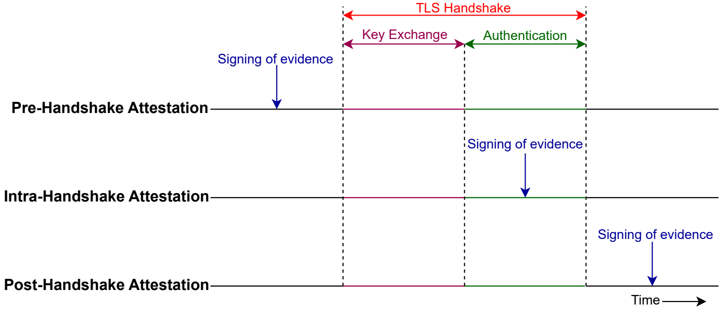


Design Options for Attested TLS



Design Options for Attested TLS





Pre-HS: Intel's RA-TLS[6]

- Widely used pre-HS attestation protocol, e.g., in
  - Gramine[2]
  - RATS-TLS[3]
  - Open Enclave Attested TLS[4]
  - SGX SDK Attested TLS[5]

[2] `https://github.com/gramineproject/gramine/tree/master/CI-Examples/ra-tls-mbedtls`

[3] `https://github.com/inclavare-containers/rats-tls`

[4] `https://github.com/openenclave/openenclave/tree/master/samples/attestedtls`

[5] `https://github.com/intel/linux-sgx/tree/master/SampleCode/SampleAttestedTLS`

[6] Knauth, Steiner, Chakrabarti, Lei, Xing, and Vij, *Integrating Remote Attestation with Transport Layer Security*, 2018.

Outline

Goal

- Formally analyze the security of Intel's RA-TLS

Outline

Analysis Approach and Tool

- Approach: Symbolic[7]

[7]Barbosa, Barthe, Karthik Bhargavan, Blanchet, Cremers, Liao, and Parno, "SoK : Computer-Aided Cryptography", 2021.

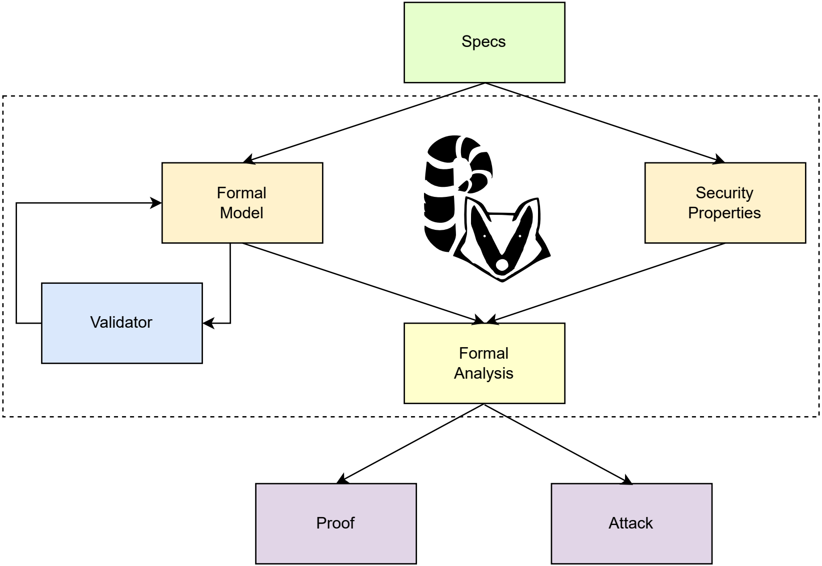
[8]Blanchet, Cheval, and Cortier, "ProVerif with lemmas, induction, fast subsumption, and much more", 2022.

Analysis Approach and Tool

- Approach: Symbolic[7]
- Tool used: ProVerif[8]

[7] Barbosa, Barthe, Karthik Bhargavan, Blanchet, Cremers, Liao, and Parno, "SoK : Computer-Aided Cryptography", 2021.
[8] Blanchet, Cheval, and Cortier, "ProVerif with lemmas, induction, fast subsumption, and much more", 2022.

Approach - Simplified



Challenge in Specification of Intel's RA-TLS[9]

- Incomplete and outdated specs for RA-TLS
  - Specs based on TLS 1.2 (TLS 1.3 is RFC since Aug 2018)
  - Fix: Used implementation and community input for formal model

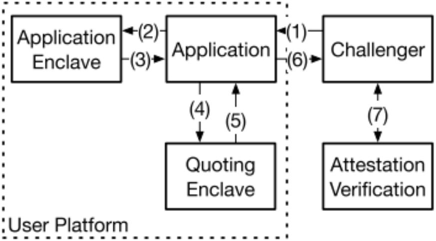


Figure 1: Remote Attestation Example. The challenger is off-platform with respect to the attester.

[9] Knauth, Steiner, Chakrabarti, Lei, Xing, and Vij, *Integrating Remote Attestation with Transport Layer Security*, 2018.

Challenge in Specification of Intel's RA-TLS[9]

- Incomplete and outdated specs for RA-TLS
  - Specs based on TLS 1.2 (TLS 1.3 is RFC since Aug 2018)
  - Fix: Used implementation and community input for formal model

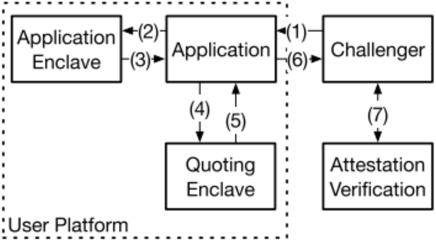


Figure 1: Remote Attestation Example. The challenger is off-platform with respect to the attester.

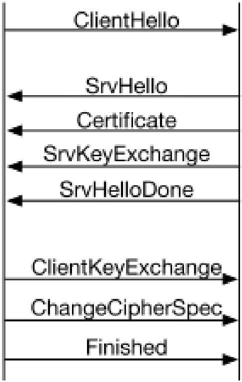


Figure 2: TLS 1.2 Handshake Messages.

[9] Knauth, Steiner, Chakrabarti, Lei, Xing, and Vij, *Integrating Remote Attestation with Transport Layer Security*, 2018.

Challenges from Formal Perspective

- Very few comments in Inria's TLS formal model[10]

[10] https://github.com/Inria-Prosecco/reftls/tree/master/pv

[11] https://mailarchive.ietf.org/arch/msg/tls/-nFk9Eu7n-YFsFfGUe9X4JnrxX8/

Challenges from Formal Perspective

- Very few comments in Inria's TLS formal model[10]
  - Literally no comments at all in main processes (such as Client12, Server12, Client13, Server13, appData, channelBindingQuery and secrecyQuery)!

[10] https://github.com/Inria-Prosecco/reftls/tree/master/pv

[11] https://mailarchive.ietf.org/arch/msg/tls/-nFk9Eu7n-YFsFfGUe9X4JnrxX8/

Challenges from Formal Perspective

- Very few comments in Inria's TLS formal model[10]
  - Literally no comments at all in main processes (such as Client12, Server12, Client13, Server13, appData, channelBindingQuery and secrecyQuery)!
  - Fix: Added extensive comments for future extensions

[10] https://github.com/Inria-Prosecco/reftls/tree/master/pv

[11] https://mailarchive.ietf.org/arch/msg/tls/-nFk9Eu7n-YFsFfGUe9X4JnrxX8/

Challenges from Formal Perspective

- Very few comments in Inria's TLS formal model[10]
  - Literally no comments at all in main processes (such as Client12, Server12, Client13, Server13, appData, channelBindingQuery and secrecyQuery)!
  - Fix: Added extensive comments for future extensions
  - Proposal: ACM-like badges (e.g., reusable)

[10] https://github.com/Inria-Prosecco/reftls/tree/master/pv

[11] https://mailarchive.ietf.org/arch/msg/tls/-nFk9Eu7n-YFsFfGUe9X4JnrxX8/

Challenges from Formal Perspective

- Very few comments in Inria's TLS formal model[10]
  - Literally no comments at all in main processes (such as Client12, Server12, Client13, Server13, appData, channelBindingQuery and secrecyQuery)!
  - Fix: Added extensive comments for future extensions
  - Proposal: ACM-like badges (e.g., reusable)
- Incomplete validation of draft 20 artifacts[11]

[10] https://github.com/Inria-Prosecco/reftls/tree/master/pv

[11] https://mailarchive.ietf.org/arch/msg/tls/-nFk9Eu7n-YFsFfGUe9X4JnrxX8/

Challenges from Formal Perspective

- Very few comments in Inria's TLS formal model[10]
  - Literally no comments at all in main processes (such as Client12, Server12, Client13, Server13, appData, channelBindingQuery and secrecyQuery)!
  - Fix: Added extensive comments for future extensions
  - Proposal: ACM-like badges (e.g., reusable)
- Incomplete validation of draft 20 artifacts[11]
  - Fix: Designed an automated validation framework for key schedule

[10] https://github.com/Inria-Prosecco/reftls/tree/master/pv

[11] https://mailarchive.ietf.org/arch/msg/tls/-nFk9Eu7n-YFsFfGUe9X4JnrxX8/

Challenges from Formal Perspective

- Very few comments in Inria's TLS formal model[10]
  - Literally no comments at all in main processes (such as Client12, Server12, Client13, Server13, appData, channelBindingQuery and secrecyQuery)!
  - Fix: Added extensive comments for future extensions
  - Proposal: ACM-like badges (e.g., reusable)
- Incomplete validation of draft 20 artifacts[11]
  - Fix: Designed an automated validation framework for key schedule
- A simple extension made the artifacts running for 1 month on high-end server (icelake)

[10] https://github.com/Inria-Prosecco/reftls/tree/master/pv

[11] https://mailarchive.ietf.org/arch/msg/tls/-nFk9Eu7n-YFsFfGUe9X4JnrxX8/

Challenges from Formal Perspective

- Very few comments in Inria's TLS formal model[10]
  - Literally no comments at all in main processes (such as Client12, Server12, Client13, Server13, appData, channelBindingQuery and secrecyQuery)!
  - Fix: Added extensive comments for future extensions
  - Proposal: ACM-like badges (e.g., reusable)
- Incomplete validation of draft 20 artifacts[11]
  - Fix: Designed an automated validation framework for key schedule
- A simple extension made the artifacts running for 1 month on high-end server (icelake)
  - Submitted to ProVerif developers for analysis

[10] https://github.com/Inria-Prosecco/reftls/tree/master/pv

[11] https://mailarchive.ietf.org/arch/msg/tls/-nFk9Eu7n-YFsFfGUe9X4JnrxX8/

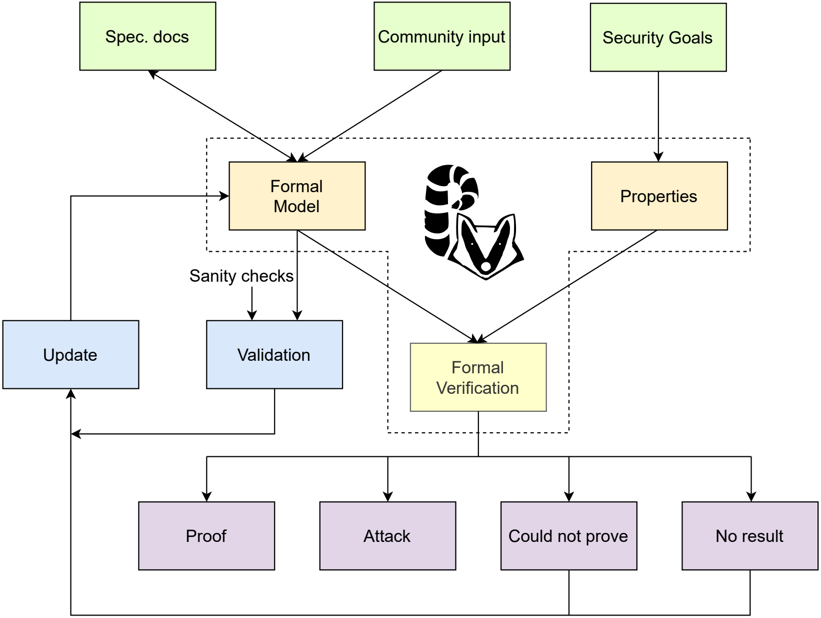
Challenges from Formal Perspective

- Very few comments in Inria's TLS formal model[10]
  - Literally no comments at all in main processes (such as Client12, Server12, Client13, Server13, appData, channelBindingQuery and secrecyQuery)!
  - Fix: Added extensive comments for future extensions
  - Proposal: ACM-like badges (e.g., reusable)
- Incomplete validation of draft 20 artifacts[11]
  - Fix: Designed an automated validation framework for key schedule
- A simple extension made the artifacts running for 1 month on high-end server (icelake)
  - Submitted to ProVerif developers for analysis
  - Fix: Formal model from scratch

[10] https://github.com/Inria-Prosecco/reftls/tree/master/pv

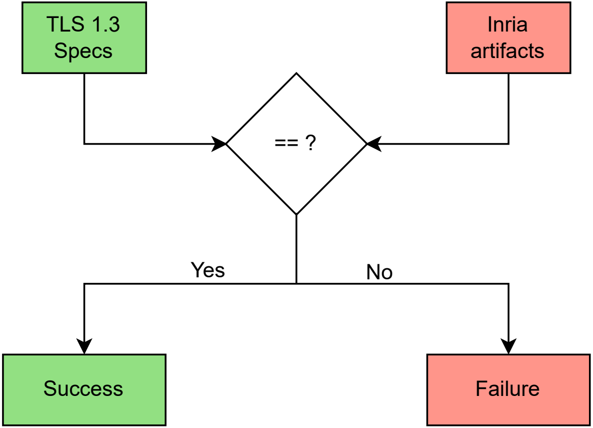
[11] https://mailarchive.ietf.org/arch/msg/tls/-nFk9Eu7n-YFsFfGUe9X4JnrxX8/

Approach

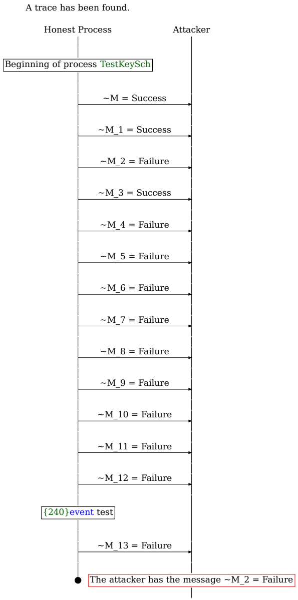

Spec. docs
Community input
Security Goals
Formal Model
Properties
Sanity checks
Update
Validation
Formal Verification
Proof
Attack
Could not prove
No result

Outline

Validation Framework



Validation Result



Example issue: Master Secret[12]

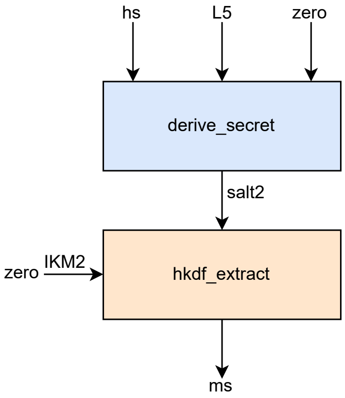


Figure: TLS 1.3 Specs

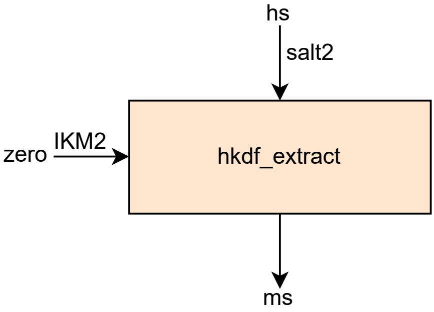


Figure: Inria artifacts

[12] https://github.com/Inria-Prosecco/reftls/issues/6

Ruling out Abstractions

- Ubuntu 20.04 LTS on an Intel Core i7-11800H processor with 64 GB of RAM

Code	ProVerif 2.04	ProVerif 2.05
Original	6 min 06.634 s	6 min 02.256 s
With issue 1 fixed	5 min 51.682 s	6 min 03.335 s
With issue 2 fixed	7 min 04.472 s	6 min 14.954 s
With issue 3 fixed	7 min 11.434 s	6 min 41.872 s
With all 3 issues fixed	6 min 40.010 s	6 min 31.887 s

A "Tale" of Community input

- Paper authors[13]

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, "Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate", 2017.

[14] https://github.com/lurk-t/proverif

[15] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/

[16] https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf

[17] https://wiki.ietf.org/meeting/119/hackathon

[18] https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00

A "Tale" of Community input

- Paper authors[13]
  - Bruno Blanchet

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, "Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate", 2017.

[14] https://github.com/lurk-t/proverif

[15] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/

[16] https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf

[17] https://wiki.ietf.org/meeting/119/hackathon

[18] https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00

A “Tale” of Community input

- Paper authors[13]
  - Bruno Blanchet
  - Karthikeyan Bhargavan

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, “Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate”, 2017.

[14] `https://github.com/lurk-t/proverif`

[15] `https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/`

[16] `https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf`

[17] `https://wiki.ietf.org/meeting/119/hackathon`

[18] `https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00`

A “Tale” of Community input

- Paper authors[13]
  - Bruno Blanchet
  - Karthikeyan Bhargavan
  - Nadim Kobeissi

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, “Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate”, 2017.

[14] https://github.com/lurk-t/proverif

[15] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/

[16] https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf

[17] https://wiki.ietf.org/meeting/119/hackathon

[18] https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00

A "Tale" of Community input

- Paper authors[13]
  - Bruno Blanchet
  - Karthikeyan Bhargavan
  - Nadim Kobeissi
- LURK authors[14]

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, "Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate", 2017.

[14] https://github.com/lurk-t/proverif

[15] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/

[16] https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf

[17] https://wiki.ietf.org/meeting/119/hackathon

[18] https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00

A “Tale” of Community input

- Paper authors[13]
  - Bruno Blanchet
  - Karthikeyan Bhargavan
  - Nadim Kobeissi
- LURK authors[14]
- IETF TLS WG[15]

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, “Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate”, 2017.

[14] https://github.com/lurk-t/proverif

[15] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/

[16] https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf

[17] https://wiki.ietf.org/meeting/119/hackathon

[18] https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00

A “Tale” of Community input

- Paper authors[13]
  - Bruno Blanchet
  - Karthikeyan Bhargavan
  - Nadim Kobeissi
- LURK authors[14]
- IETF TLS WG[15]
- IRTF UFMRG chairs

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, “Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate”, 2017.

[14] https://github.com/lurk-t/proverif

[15] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/

[16] https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf

[17] https://wiki.ietf.org/meeting/119/hackathon

[18] https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00

A “Tale” of Community input

- Paper authors[13]
  - Bruno Blanchet
  - Karthikeyan Bhargavan
  - Nadim Kobeissi
- LURK authors[14]
- IETF TLS WG[15]
- IRTF UFMRG chairs
- CCC attestation SIG[16]

---

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, “Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate”, 2017.

[14] https://github.com/lurk-t/proverif

[15] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/

[16] https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf

[17] https://wiki.ietf.org/meeting/119/hackathon

[18] https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00

A "Tale" of Community input

- Paper authors[13]
  - Bruno Blanchet
  - Karthikeyan Bhargavan
  - Nadim Kobeissi
- LURK authors[14]
- IETF TLS WG[15]
- IRTF UFMRG chairs
- CCC attestation SIG[16]
- ...

---

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, "Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate", 2017.

[14] https://github.com/lurk-t/proverif

[15] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/

[16] https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf

[17] https://wiki.ietf.org/meeting/119/hackathon

[18] https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00

A "Tale" of Community input

- Paper authors[13]
  - Bruno Blanchet
  - Karthikeyan Bhargavan
  - Nadim Kobeissi
- LURK authors[14]
- IETF TLS WG[15]
- IRTF UFMRG chairs
- CCC attestation SIG[16]
- ...
- IETF 119 Hackathon[17]

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, "Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate", 2017.

[14] https://github.com/lurk-t/proverif

[15] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/

[16] https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf

[17] https://wiki.ietf.org/meeting/119/hackathon

[18] https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00

A “Tale” of Community input

- Paper authors[13]
  - Bruno Blanchet
  - Karthikeyan Bhargavan
  - Nadim Kobeissi
- LURK authors[14]
- IETF TLS WG[15]
- IRTF UFMRG chairs
- CCC attestation SIG[16]
- ...
- IETF 119 Hackathon[17]
- IRTF Crypto Forum RG @ IETF 119[18]

---

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, “Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate”, 2017.

[14] https://github.com/lurk-t/proverif

[15] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/

[16] https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf

[17] https://wiki.ietf.org/meeting/119/hackathon

[18] https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00

A "Tale" of Community input

- Paper authors[13]
    - Bruno Blanchet
    - Karthikeyan Bhargavan
    - Nadim Kobeissi
- LURK authors[14]
- IETF TLS WG[15]
- IRTF UFMRG chairs
- CCC attestation SIG[16]
- ...
- IETF 119 Hackathon[17]
- IRTF Crypto Forum RG @ IETF 119[18]
- Tool session @ GT MFS'24

[13] Karthikeyan Bhargavan, Blanchet, and Kobeissi, "Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate", 2017.

[14] https://github.com/lurk-t/proverif

[15] https://mailarchive.ietf.org/arch/msg/tls/ZGmyHwTYh2iPwPrirj_rkSTYhDo/

[16] https://github.com/CCC-Attestation/meetings/blob/main/materials/MuhammadUsamaSardar_Formal_RA-TLS.pdf

[17] https://wiki.ietf.org/meeting/119/hackathon

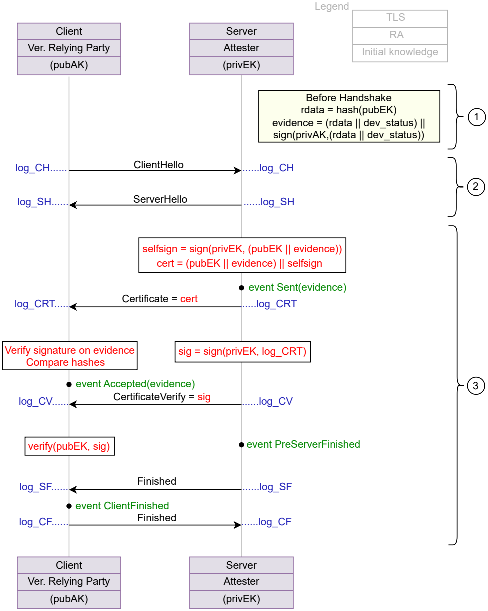
[18] https://datatracker.ietf.org/meeting/119/materials/slides-119-cfrg-formal-analysis-of-ra-tls-00

Outline

Agenda

5. Formal Analysis of Attested TLS
   - Flow
   - Threat Model
   - Properties
   - Proposed Mitigations/Fix

RA-TLS


Legend
TLS
RA
Initial knowledge
Client
Ver. Relying Party
(pubAK)
Server
Attester
(privEK)
Before Handshake
rdata = hash(pubEK)
evidence = (rdata || dev_status) ||
sign(privAK,(rdata || dev_status))
1
log_CH......
ClientHello
......log_CH
log_SH......
ServerHello
......log_SH
2
selfsign = sign(privEK, (pubEK || evidence))
cert = (pubEK || evidence) || selfsign
event Sent(evidence)
log_CRT......
Certificate = cert
......log_CRT
Verify signature on evidence
Compare hashes
sig = sign(privEK, log_CRT)
event Accepted(evidence)
log_CV......
CertificateVerify = sig
......log_CV
verify(pubEK, sig)
event PreServerFinished
log_SF......
Finished
......log_SF
event ClientFinished
log_CF......
Finished
......log_CF
3
Client
Ver. Relying Party
(pubAK)
Server
Attester
(privEK)

Agenda

5. Formal Analysis of Attested TLS
   - Flow
   - Threat Model
   - Properties
   - Proposed Mitigations/Fix

Threat Model

- Scope: TLS mode = non-PSK handshake

Threat Model

- Scope: TLS mode = non-PSK handshake
- Weak hash, e.g., SLOTH (represented by `WeakHash`)

Threat Model

- Scope: TLS mode = non-PSK handshake
- Weak hash, e.g., SLOTH (represented by `WeakHash`)
- Weak DH groups, e.g., Logjam (represented by `WeakDH`)

Threat Model

- Scope: TLS mode = non-PSK handshake
- Weak hash, e.g., SLOTH (represented by `WeakHash`)
- Weak DH groups, e.g., Logjam (represented by `WeakDH`)
- Bad elements within strong DH groups (rep. by `SentBadElement`)

Threat Model

- Scope: TLS mode = non-PSK handshake
- Weak hash, e.g., SLOTH (represented by `WeakHash`)
- Weak DH groups, e.g., Logjam (represented by `WeakDH`)
- Bad elements within strong DH groups (rep. by `SentBadElement`)
- With and without weak (or compromised) ephemeral key `privEK`

Threat Model

- Scope: TLS mode = non-PSK handshake
- Weak hash, e.g., SLOTH (represented by `WeakHash`)
- Weak DH groups, e.g., Logjam (represented by `WeakDH`)
- Bad elements within strong DH groups (rep. by `SentBadElement`)
- With and without weak (or compromised) ephemeral key `privEK`
- Without weak (or compromised) attestation key `privAK`

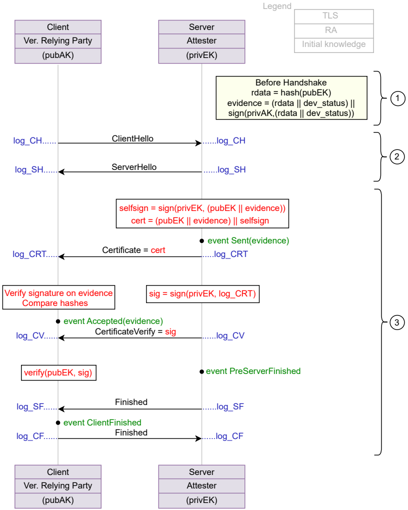
Agenda

5. Formal Analysis of Attested TLS
   - Flow
   - Threat Model
   - Properties
   - Proposed Mitigations/Fix

RA layer: Freshness of Evidence

$\forall\ ev.$

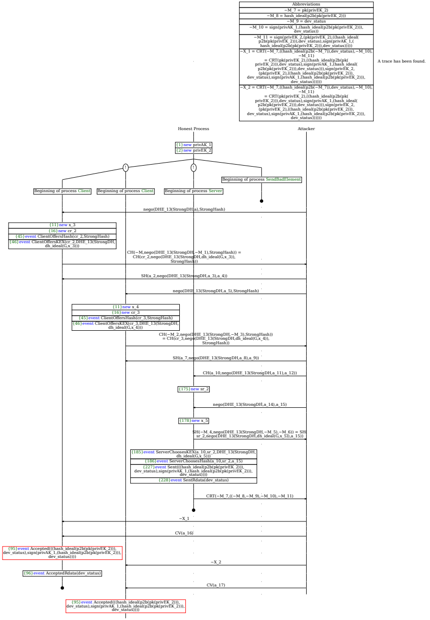
$$inj - event(Accepted(ev)) ==> inj - event(Sent(ev)) \quad (1)$$



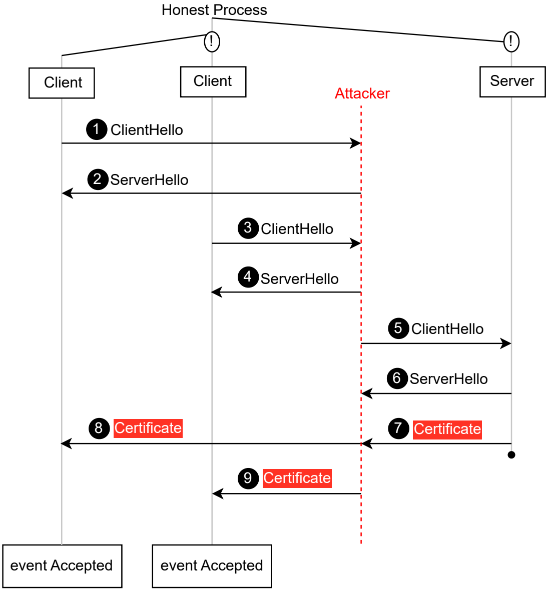
Eliminate Reasons of Failure

$$\forall\ ev.$$
$$\exists\ cr, sr, e.$$
$$inj - event(Accepted(ev)) ==> inj - event(Sent(ev))\ \|$$
$$event(ServerChoosesKEX(cr, sr, DHE\_13(WeakDH, e)))\ \|$$
$$event(ServerChoosesHash(cr, sr, WeakHash))\ \|$$
$$event(SentBadElement).$$

Attack Trace by ProVerif



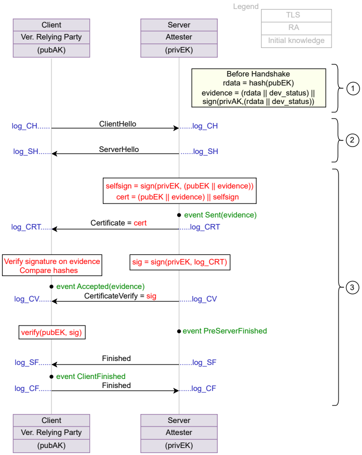
Simplified Attack Trace



TLS layer: Server authentication

$$\forall\ cr, sr, sid.$$

$$inj - event(ClientFinished(cr, sr, sid)) ==> inj - event(PreServerFinished(cr, sr, sid)) \quad (2)$$



Eliminate Reasons of Failure

$$\begin{aligned}
&\forall\ cr, sr, sid.\\
&\exists\ cr', sr', e.\\
&inj - event(ClientFinished(cr, sr, sid)) ==>\\
&inj - event(PreServerFinished(cr, sr, sid)) \parallel\\
&event(ServerChoosesKEX(cr, sr, DHE_13(\color{red}{WeakDH}, e))) \parallel\\
&event(ServerChoosesHash(cr', sr', \color{red}{WeakHash})) \parallel\\
&event(\color{red}{SentBadElement}).
\end{aligned}$$

Summary so far

Property	Without `privEK` leak	With `privEK` leak
Freshness of evidence	$\times$ (1.7 s)	$\times$ (6 min 56 s)
Server authentication	✓ (4.6 s)	$\times$ (2 min 08 s)

Table: Verification results and times for RA-TLS protocol

Agenda

5. Formal Analysis of Attested TLS
   - Flow
   - Threat Model
   - Properties
   - Proposed Mitigations/Fix

Mitigations within Pre-HS attestation

- Reduce the validity period of certificates

Mitigations within Pre-HS attestation

- Reduce the validity period of certificates
  - Gramine: 20301231235959

Mitigations within Pre-HS attestation

- Reduce the validity period of certificates
  - Gramine: 20301231235959
  - Open Enclave SDK: 20501231235959

Mitigations within Pre-HS attestation

- Reduce the validity period of certificates
  - Gramine: 20301231235959
  - Open Enclave SDK: 20501231235959
  - Intel SGX libraries: 20501231235959

Mitigations within Pre-HS attestation

- Reduce the validity period of certificates
  - Gramine: 20301231235959
  - Open Enclave SDK: 20501231235959
  - Intel SGX libraries: 20501231235959
- Client maintains a database of hash of all evidences it has seen and each time compares with the seen evidences

Mitigations within Pre-HS attestation

- Reduce the validity period of certificates
  - Gramine: 20301231235959
  - Open Enclave SDK: 20501231235959
  - Intel SGX libraries: 20501231235959
- Client maintains a database of hash of all evidences it has seen and each time compares with the seen evidences
  - Makes TLS stateful

Mitigations within Pre-HS attestation

- Reduce the validity period of certificates
  - Gramine: 20301231235959
  - Open Enclave SDK: 20501231235959
  - Intel SGX libraries: 20501231235959
- Client maintains a database of hash of all evidences it has seen and each time compares with the seen evidences
  - Makes TLS stateful
  - What if the client would like to reconnect later (with the same TCB, i.e., same evidence)?

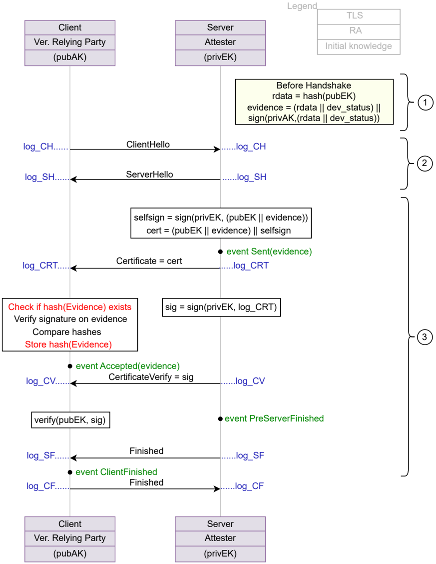
Mitigations within Pre-HS attestation

- Reduce the validity period of certificates
  - Gramine: 20301231235959
  - Open Enclave SDK: 20501231235959
  - Intel SGX libraries: 20501231235959
- Client maintains a database of hash of all evidences it has seen and each time compares with the seen evidences
  - Makes TLS stateful
  - What if the client would like to reconnect later (with the same TCB, i.e., same evidence)?
    - Perhaps it can use PSK then

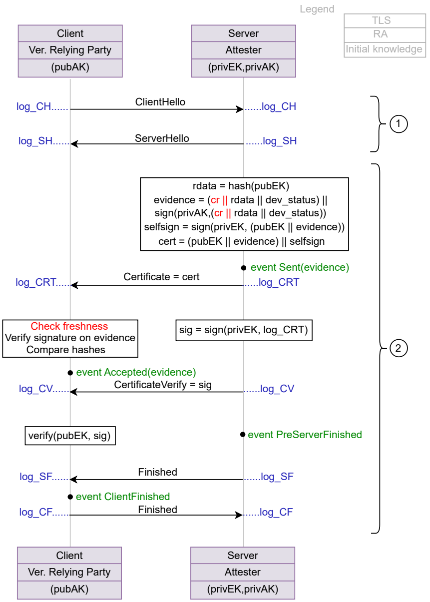
Mitigations within Pre-HS attestation

- Reduce the validity period of certificates
  - Gramine: 20301231235959
  - Open Enclave SDK: 20501231235959
  - Intel SGX libraries: 20501231235959
- Client maintains a database of hash of all evidences it has seen and each time compares with the seen evidences
  - Makes TLS stateful
  - What if the client would like to reconnect later (with the same TCB, i.e., same evidence)?
    - Perhaps it can use PSK then
- Discussion: other thoughts?

Mitigations within Pre-HS attestation



Proposed Fix: Intra-HS attestation



Proposed Fix: Intra-HS attestation

- Leakage of `privEK` can be detected (as long as `privAK` is not leaked)

Property	Without `privEK` leak	With `privEK` leak
Freshness of evidence	✓ (02.5 s)	✓ (02 min 43.7 s)
Server authentication	✓ (13.9 s)	× (15 min 55.3 s)

Table: Verification results and times for our proposed minimal fix to RA-TLS protocol

Outline

Summary

- Intel's RA-TLS is potentially vulnerable to evidence replay attacks

Summary

- Intel's RA-TLS is potentially vulnerable to evidence replay attacks
  - Need for standardized and formally verified attested TLS

Summary

- Intel's RA-TLS is potentially vulnerable to evidence replay attacks
  - Need for standardized and formally verified attested TLS
- Questions for discussion

Summary

- Intel's RA-TLS is potentially vulnerable to evidence replay attacks
  - Need for standardized and formally verified attested TLS
- Questions for discussion
  - Is there any *practical* attack if the derive_secret for ‘ms‘ is skipped?

Summary

- Intel's RA-TLS is potentially vulnerable to evidence replay attacks
  - Need for standardized and formally verified attested TLS
- Questions for discussion
  - Is there any *practical* attack if the derive_secret for 'ms' is skipped?
  - Is a fix for RA-TLS within pre-HS attestation possible without storing state (evidence)?

Summary

- Intel's RA-TLS is potentially vulnerable to evidence replay attacks
  - Need for standardized and formally verified attested TLS
- Questions for discussion
  - Is there any *practical* attack if the derive_secret for 'ms' is skipped?
  - Is a fix for RA-TLS within pre-HS attestation possible without storing state (evidence)?
  - How to deal with underspecification?

Key References

Barbosa, Manuel, Gilles Barthe, Karthik Bhargavan, Bruno Blanchet, Cas Cremers, Kevin Liao, and Bryan Parno. "SoK : Computer-Aided Cryptography". In: *42nd IEEE Symposium on Security and Privacy*. 2021. URL: https://eprint.iacr.org/2019/1393.pdf.

Bhargavan, Karthikeyan, Bruno Blanchet, and Nadim Kobeissi. "Verified Models and Reference Implementations for the TLS 1.3 Standard Candidate". In: *2017 IEEE Symposium on Security and Privacy (SP)*. 2017, pp. 483–502. DOI: 10.1109/SP.2017.26.

Blanchet, Bruno, Vincent Cheval, and Véronique Cortier. "ProVerif with lemmas, induction, fast subsumption, and much more". In: *IEEE Symposium on Security and Privacy (S&P'22)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2022, pp. 205–222. DOI: 10.1109/SP46214.2022.00013.

Knauth, T., M. Steiner, S. Chakrabarti, L. Lei, C. Xing, and M. Vij. *Integrating Remote Attestation with Transport Layer Security*. Tech. rep. Intel Labs, 2018. URL: https://arxiv.org/abs/1801.05863.

ACK

- Ionut Mihalcea (Arm)
- Yaron Sheffer (Intuit)
- Thore Sommer (Kiel University)
- Carsten Weinhold (Barkhausen Institut)
- Michael Roitzsch (Barkhausen Institut)
- Yogesh Deshpande (Arm)
- Anonymous HCVS reviewer # 3

CONFIDENTIAL COMPUTING CONSORTIUM