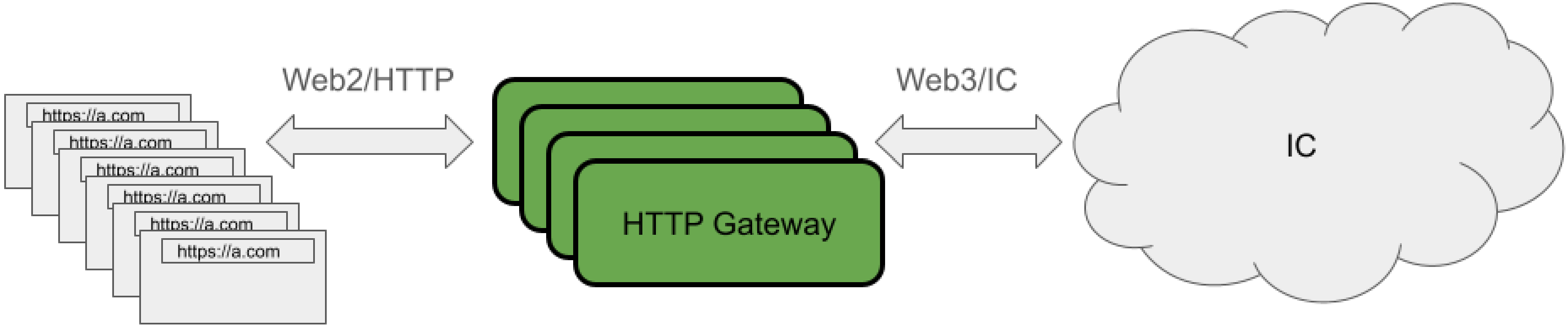
Site Attestation: Browser-based Remote Attestation

Luca Preibsch, Maxim Ritter von Onciul, Rüdiger Kapitza
Friedrich-Alexander-Universität Erlangen-Nürnberg

Presentation for the Confidential Computing Consortium Technical Advisory Committee

Background

How can a *browser* be securely connected to Web3?



Set of works to make this happen, but the use case is, of course, more generic:

- Anna Galanou, Khushboo Bindlish, Luca Preibsch, Yvonne-Anne Pignolet, Christof Fetzer, and Rüdiger Kapitza. 2023. **Trustworthy confidential virtual machines for the masses.** In Proceedings of the 24th International Middleware Conference (Middleware '23). https://doi.org/10.1145/3590140.3629124

- Anna Galanou, Florian Lubitz, Hajeong Jeon, Christof Fetzer, and Rüdiger Kapitza. 2025. **Full Trust Alchemist: Reforging Attestation for Cloud-based Confidential Workloads**. In Proceedings of the 26th International Middleware Conference (Middleware '25). https://doi.org/10.1145/3721462.3770778

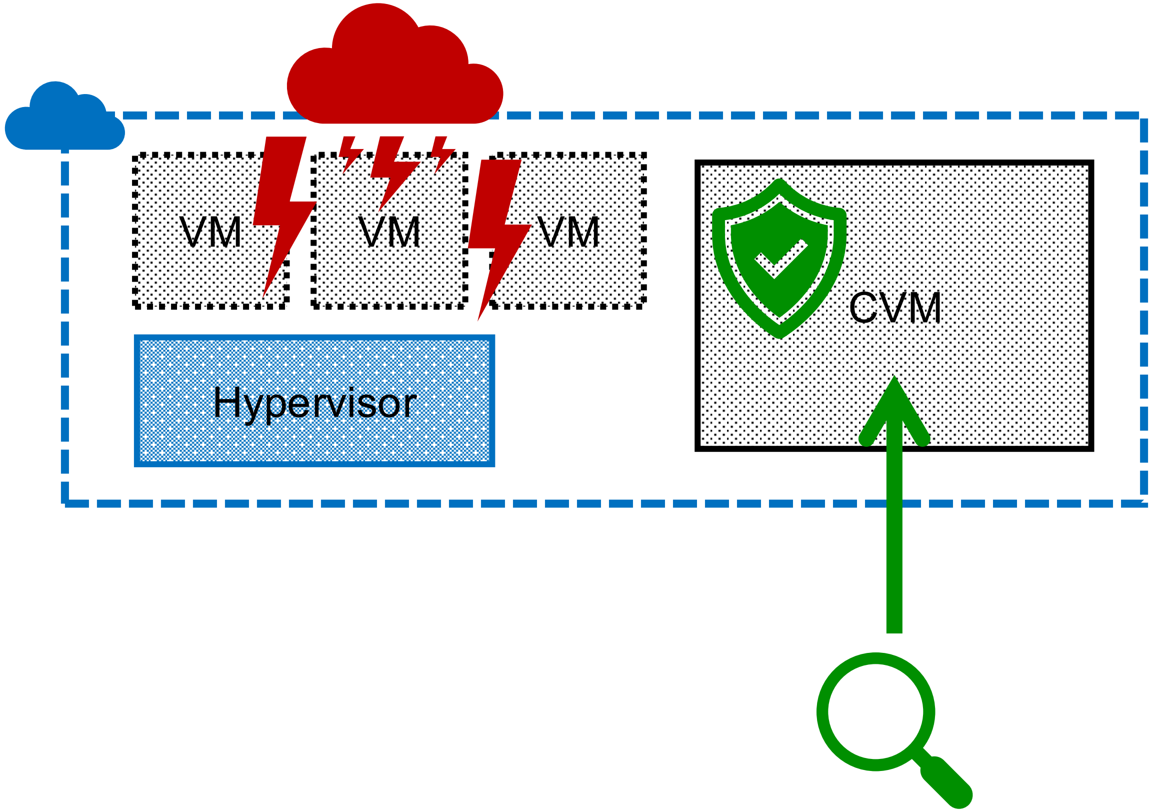
- Luca Preibsch, Maxim Ritter von Onciul, and Rüdiger Kapitza. 2025. **Site Attestation: Browser-based Remote Attestation**. In Proceedings of the 18th European Workshop on Systems Security (EuroSec'25). https://doi.org/10.1145/3722041.3723095

- Arne Vogel, Luca Preibsch, Rüdiger Birkner, Raymond Khalife, Or Ricon, Yvonne-Anne Pignolet and Rüdiger Kapitza. **More secure access for everyone using confidential computing**. 2026. (under submission)

Motivation

Confidential Computing

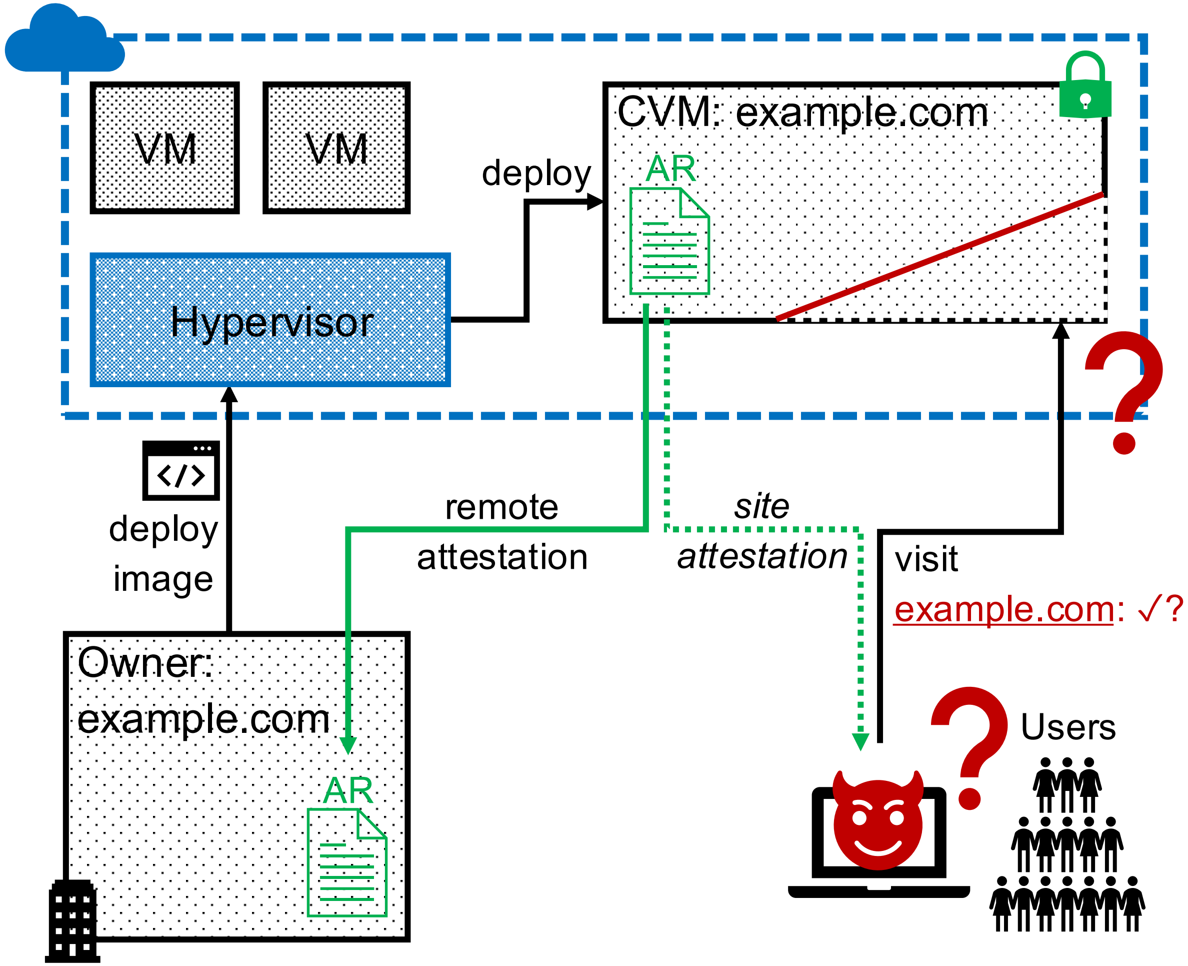
- Generic idea of **confidential computing** is to
  - **protect services from external access** and
  - make this property **remotely attestable**
- Especially useful for outsourcing services to **cloud computing providers**
- Several flavors of confidential computing:
  1. Enclave based
  2. **Confidential Virtual Machines** (CVM)
     - **AMD SEV-SNP**
     - Intel TDX, …
- Site Attestation is applicable to **all forms of confidential computing**



Motivation

Remote Attestation & the User Perspective

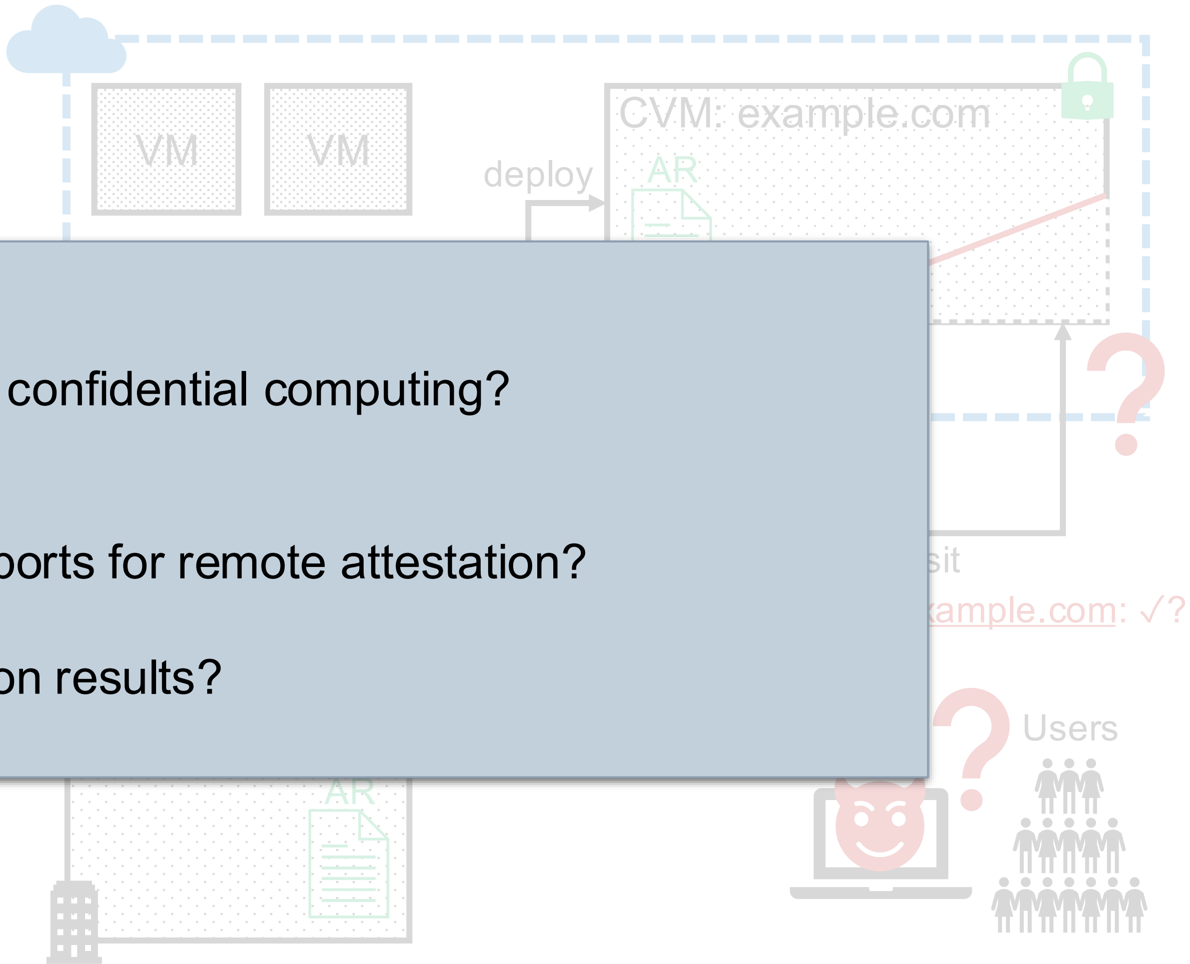
- Service owner hosts website in the cloud
  - Confidential computing protects VM runtime
  - Protection can be validated via remote attestation
- Security guarantees on the user's side are lacking:
  - HTTPS:
    connection terminates with certificate owner,
    protects data in transit
  - Domain reputation?
- **Users could greatly benefit from the advantages of confidential computing**



Motivation

Remote Attestation & the User Perspective

- Service owner hosts website in the cloud
  - Confidential computing protects VM runtime
  - Protection can be validated via remote attestation
- Security
  - HTTPS
    conne
    protec
  - Domai
- **Users c**
  of confidential computing



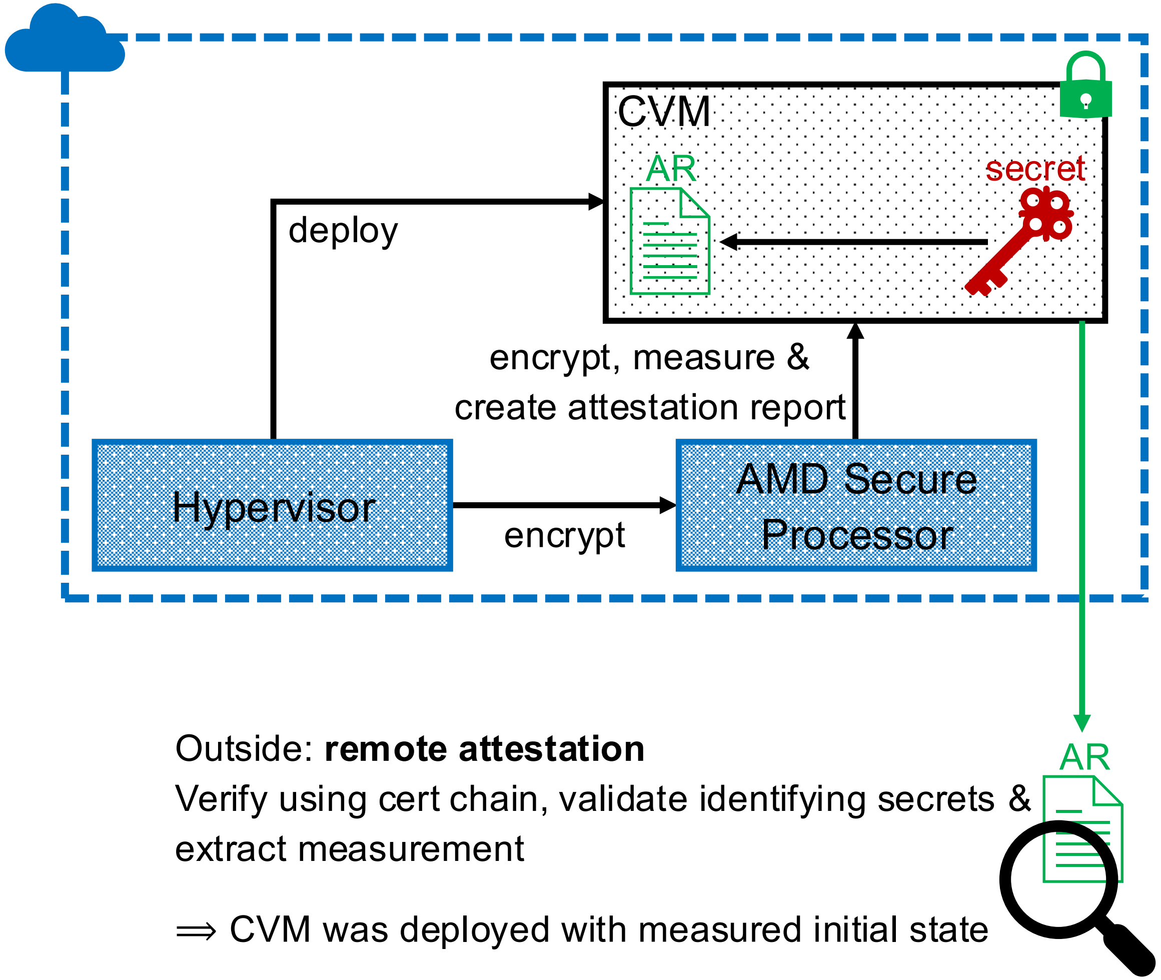
Challenges:

1. How to identify websites protected by confidential computing?
   Could there be a standard way?
2. How can users validate attestation reports for remote attestation?
3. How could users interpret the validation results?

Background

AMD SEV-SNP & Remote Attestation

- **AMD Secure Processor**
  - encrypts CVM’s memory and registers
  - measures the CVM’s initial state
- **Attestation Report**
  - carries initial measurement & any user defined data
  - is signed, can be verified using an AMD cert chain
- Adding full disc encryption and without interfaces for reconfiguration ⟹ **CVM is sealed**[1]
  ⟹ **its state cannot deviate from its initial one**
- Using public source code, the **measurement can be reproduced externally**



Outside: **remote attestation**
Verify using cert chain, validate identifying secrets &
extract measurement

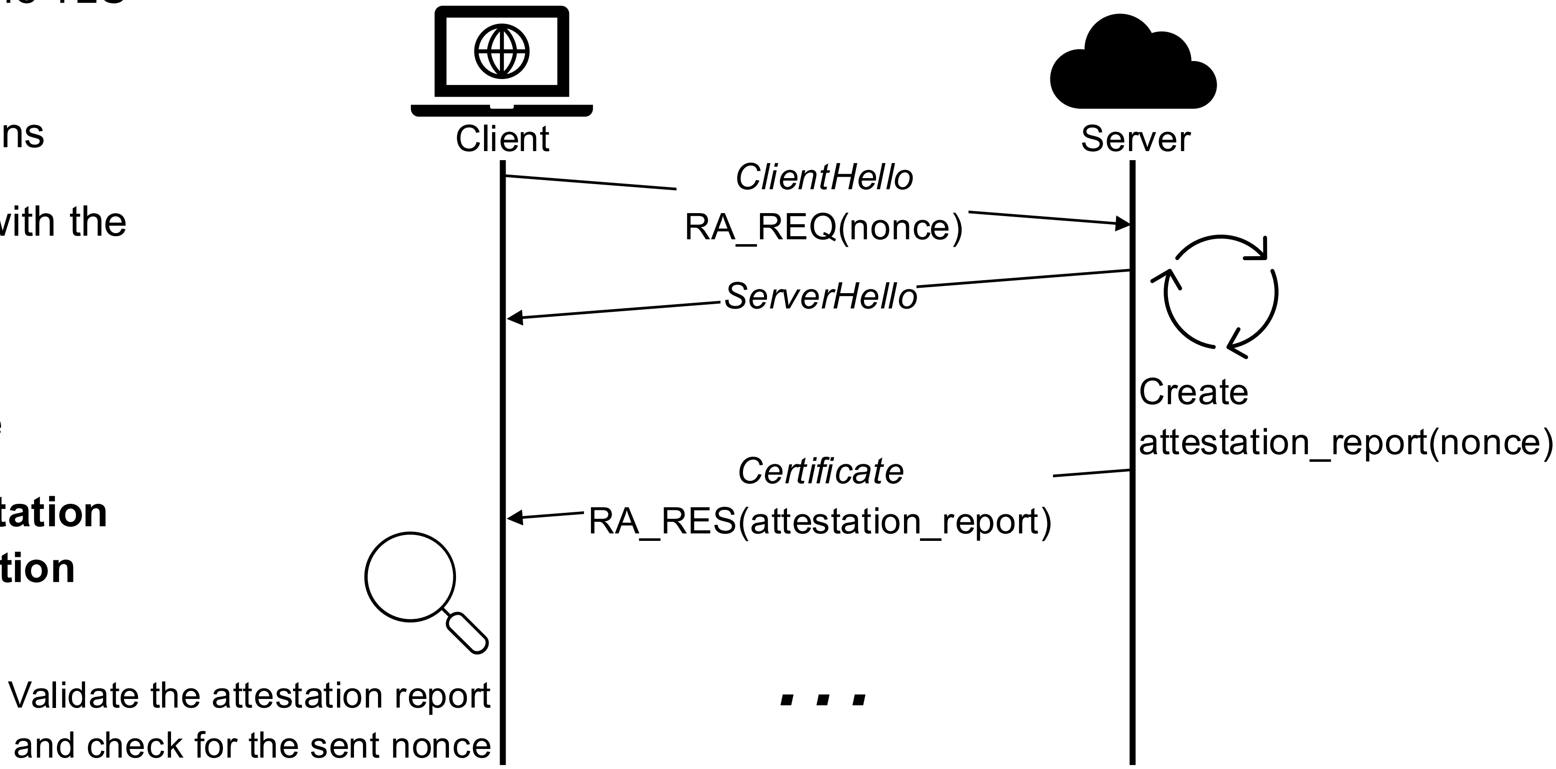
⟹ CVM was deployed with measured initial state

[1] Anna Galanou, Khushboo Bindlish, Luca Preibsch, Yvonne-Anne Pignolet, Christof Fetzer, and Rüdiger Kapitza. 2023. Trustworthy confidential virtual machines for the masses. In Proceedings of the 24th International Middleware Conference. 316–328. https://doi.org/10.1145/3590140.3629124

Background

Remote Attestation over TLS (RATLS)[2]

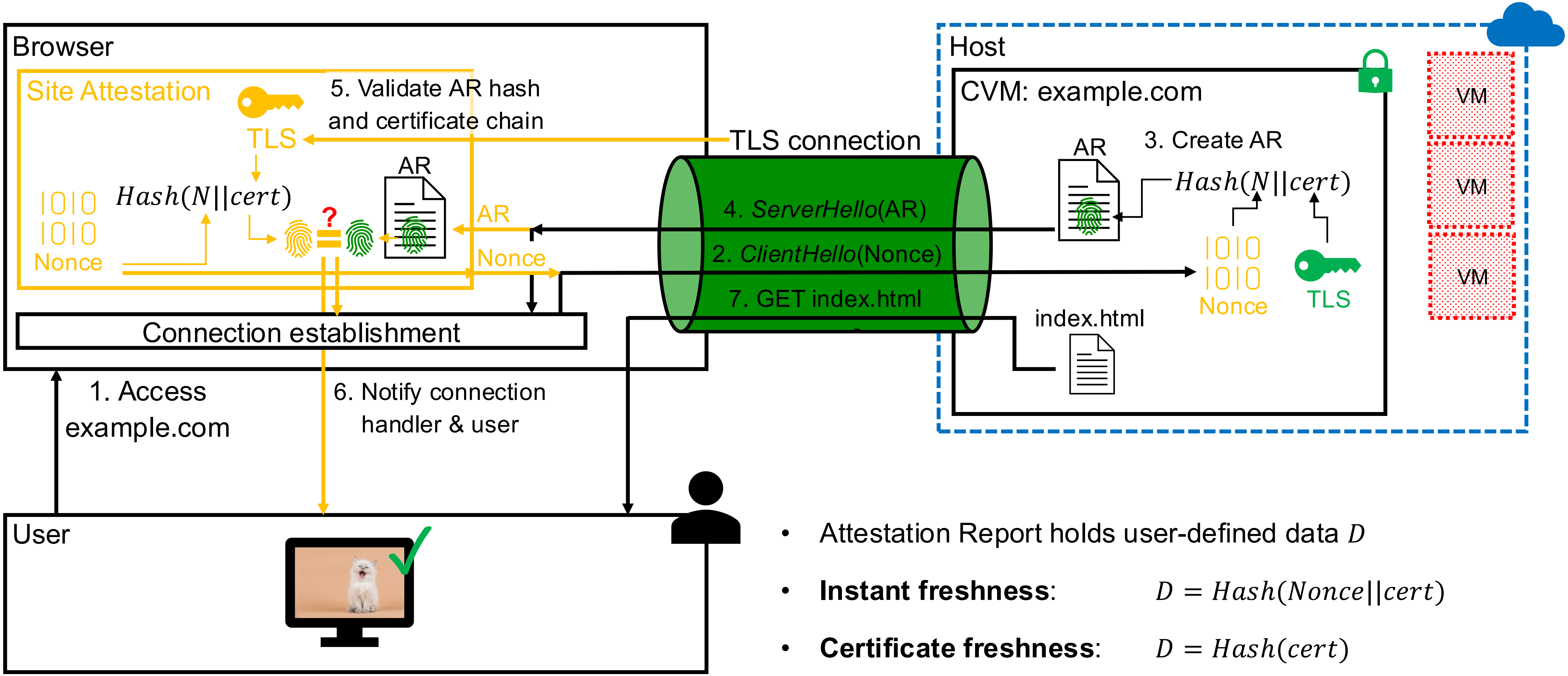
- Idea: perform remote attestation as part of the TLS connection handshake
- Approach: Use TLS 1.3 handshake extensions
  1. Piggyback the attestation report request with the *ClientHello*-message
  2. Receive the attestation report with the corresponding *ServerCertificate*-message
- Should a server **not respond with an attestation report**, it **does not support remote attestation**


Client
Server
ClientHello
RA_REQ(nonce)
ServerHello
Create
attestation_report(nonce)
Certificate
RA_RES(attestation_report)
Validate the attestation report
and check for the sent nonce
. . .

[2] Robert Walther, Carsten Weinhold, and Michael Roitzsch. 2022. RATLS: Integrating Transport Layer Security with Remote Attestation. In Applied Cryptography and Network Security Workshops. Vol. 13285. 361–379. https://doi.org/10.1007/978-3-031-16815-4_20

Site Attestation

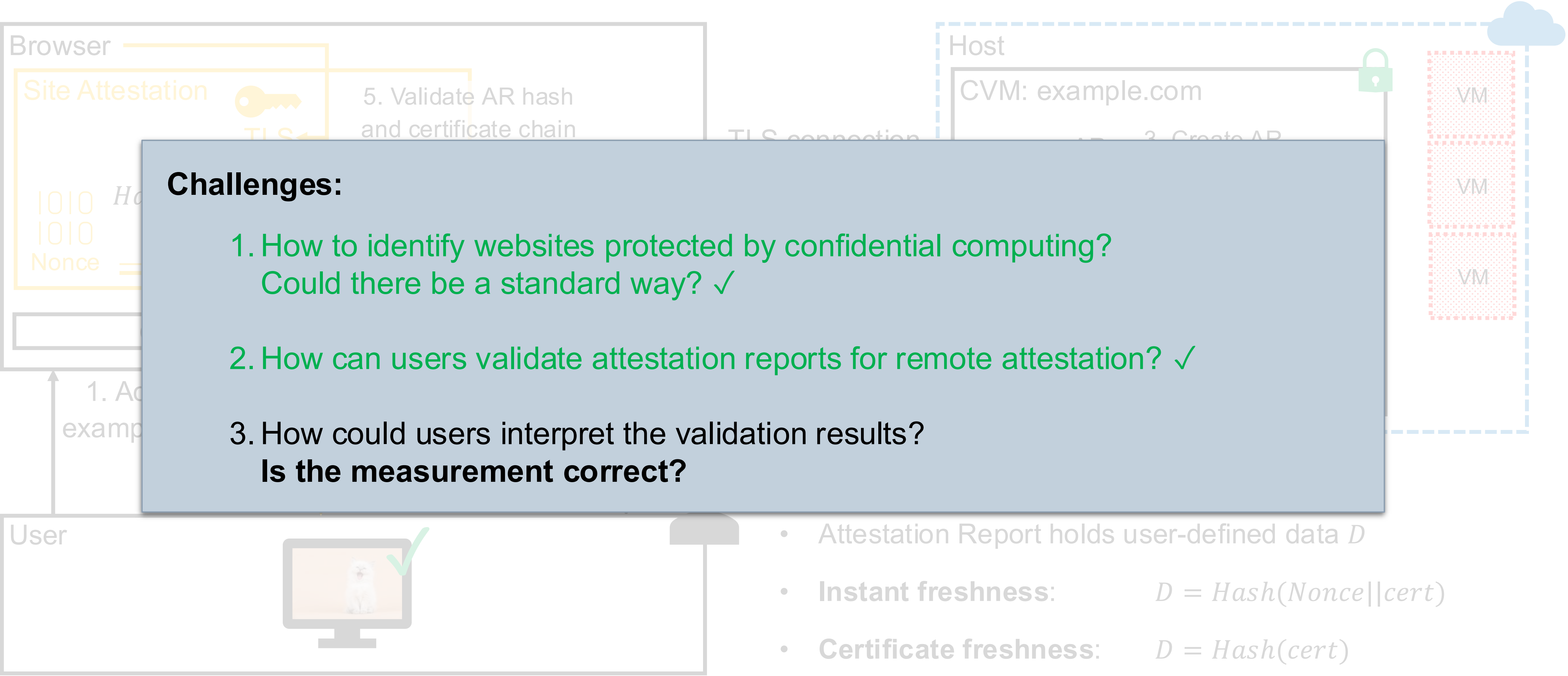
Requesting & Validating the Attestation Report



- Attestation Report holds user-defined data D
- **Instant freshness**: $D = Hash(Nonce||cert)$
- **Certificate freshness**: $D = Hash(cert)$

Site Attestation

Requesting & Validating the Attestation Report



Challenges:

1. How to identify websites protected by confidential computing?
   Could there be a standard way? ✓
2. How can users validate attestation reports for remote attestation? ✓
3. How could users interpret the validation results?
   Is the measurement correct?

- Attestation Report holds user-defined data D
- **Instant freshness**: $D = Hash(Nonce||cert)$
- **Certificate freshness**: $D = Hash(cert)$

Site Attestation

Trust Policies: To trust or not to trust

- **Attestation report** is valid ✓
- But how to decide if a measurement is deemed trustworthy ?
- **3 policies:**

1. *Exact Match Policy:* The **exact measurement** is configured per domain
2. *Trusted Remote Source Policy:* Remote sources are configured as **repository of exact measurements** to trust
3. *Owner Keys Policy:* Attestation reports are **signed via owner key**

- If no policy can be applied based on previous configuration, the user will be prompted to decide via dialog

Site Attestation

Trust Policies: To trust or not to trust

- **Attestation report** is valid ✓
- But how to decide if a measurement is deemed trustworthy ?
- **3 polici**

1. *Exact*
2. *Truste* trust
3. *Owne*

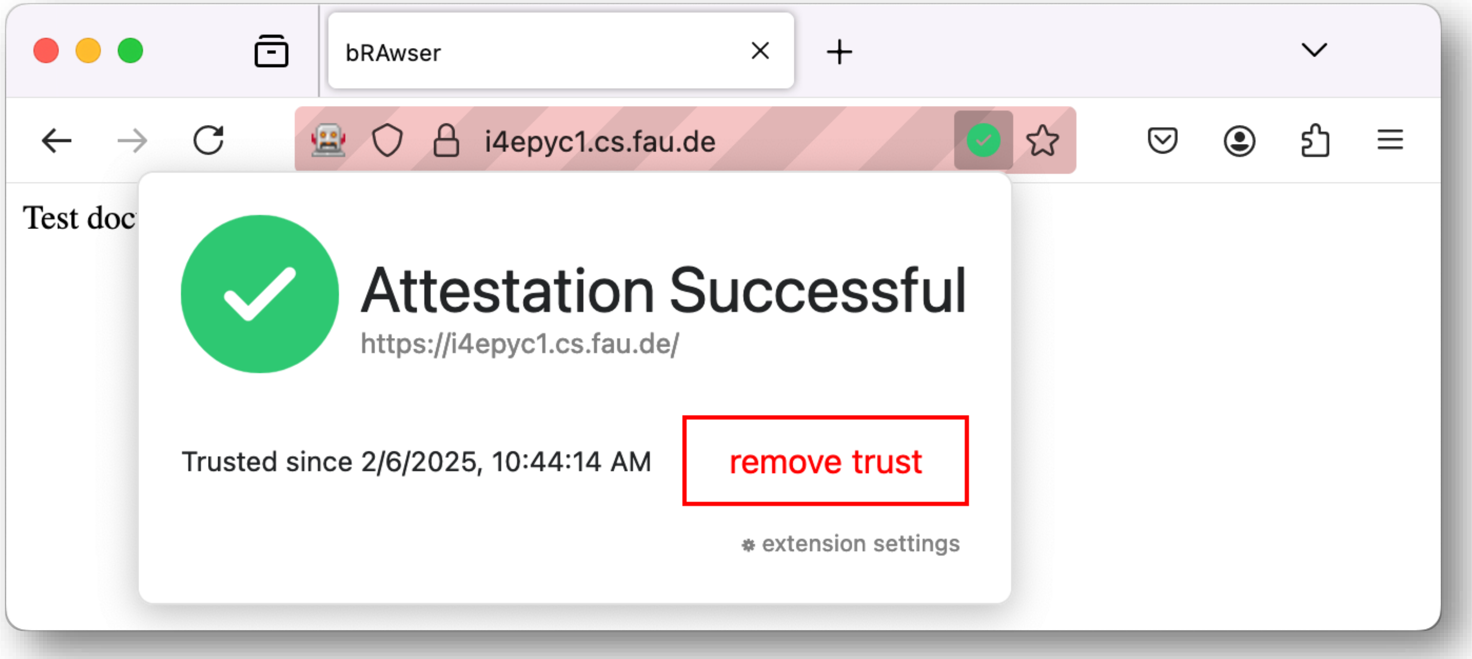
- If no pol

Challenges:

1. How to identify websites protected by confidential computing?
   Could there be a standard way? ✓
2. How can users validate attestation reports for remote attestation? ✓
3. How could users interpret the validation results?
   Is the measurement correct? ✓

Site Attestation

Indicating Trusted Websites



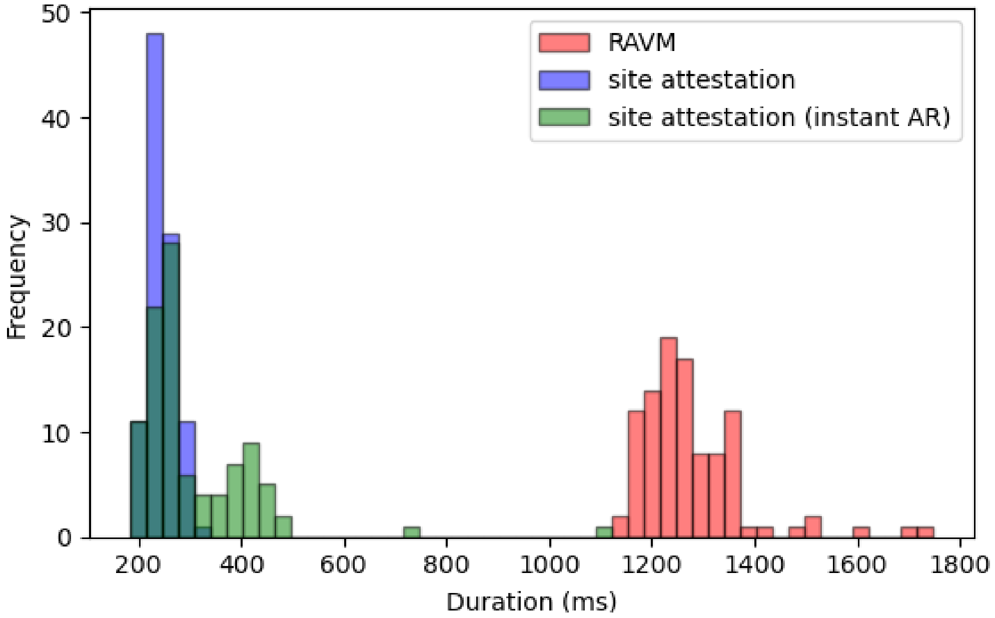
Attestation Successful

https://i4epyc1.cs.fau.de/

Trusted since 2/6/2025, 10:44:14 AM

Evaluation

- Questions:
  - What is the cost of generating attestation reports?
  - How does it affect server-side scalability?
  - **What is the increase in latency when establishing a connection to a *site attestation*-enabled website?**
  - **What about already established connections?**
- **Baseline:** latency of one HTTP GET
  145.95ms (new connection); 69.59ms (existing connection)
- **Preconfigured Exact Match Trust Policy:**
  - **Instant freshness:** 305.26ms (new connection)
  - **Certificate freshness:** 242.75ms (new connection)
  - **Both:** 70.30ms (existing connection)



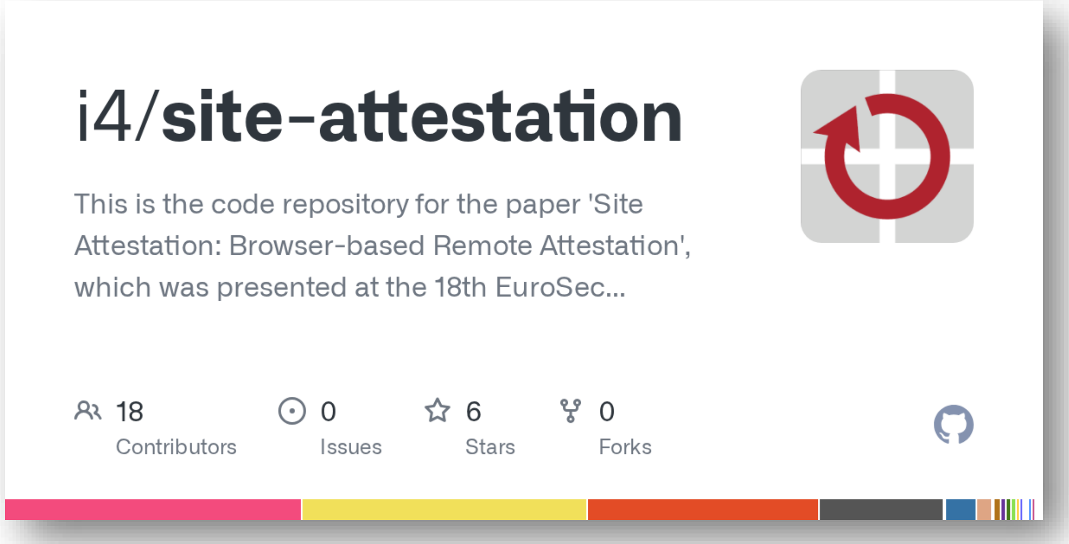
Conclusion

- *Site attestation* proposes to
  a) secure websites through **confidential computing**
  b) and to perform **remote attestation** using
  c) **trustworthiness policies** while surfing the web

- This makes the advantages of **confidential computing accessible** to end users
- and **reduces the need to blindly rely** on a website's reputation,
- while **keeping the browser quick and responsive** with little latency increase.

What can the industry do?

- **Enable TLS connection access in major browsers**
- **Build a standard for remote attestation in the browser**
- Implemented through browser extensions



i4/**site-attestation**

This is the code repository for the paper 'Site Attestation: Browser-based Remote Attestation', which was presented at the 18th EuroSec...

https://github.com/i4/site-attestation

Appendix

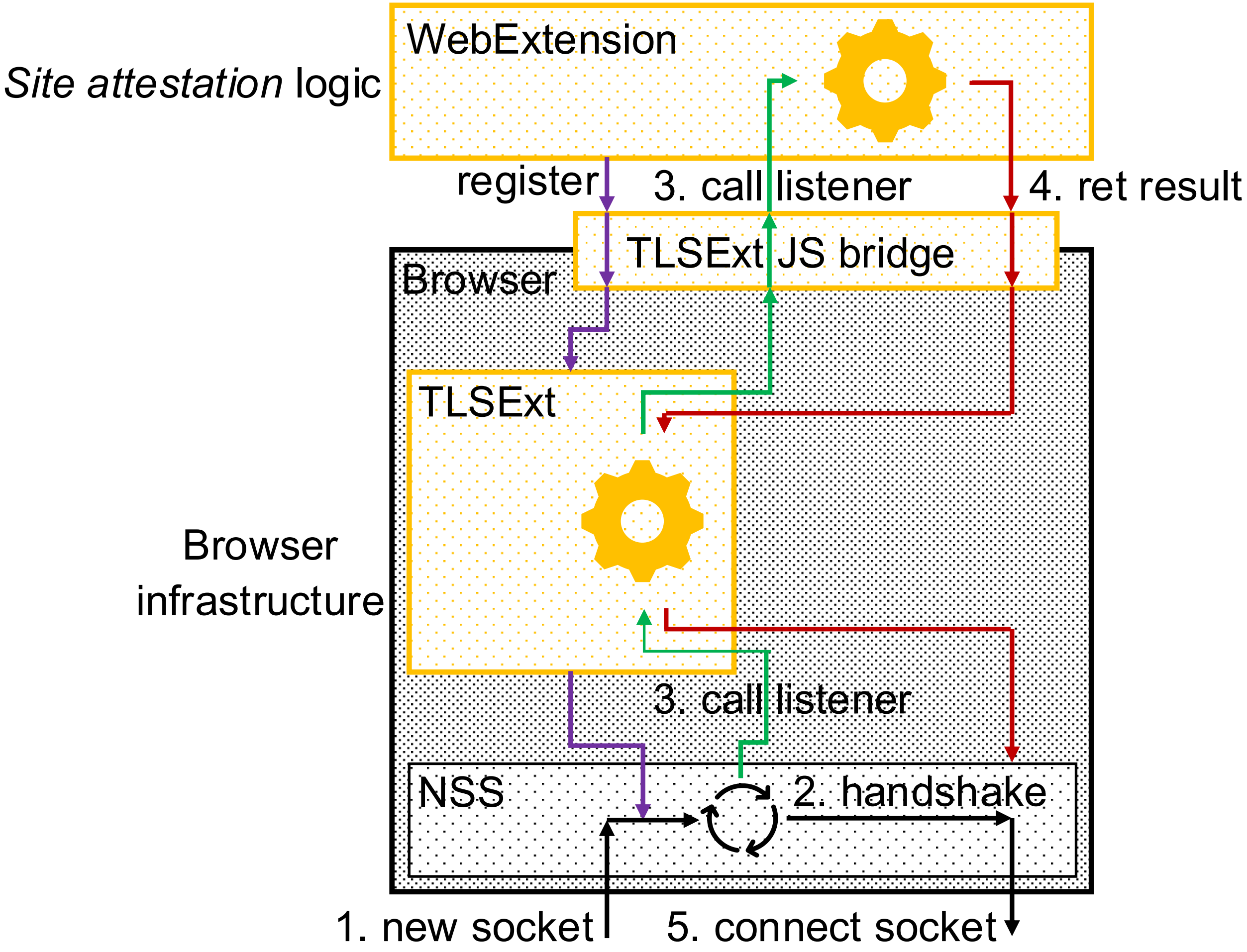
Attacker Model

- Threat model typical for confidential computing
- **Attacker**:
  - Has root access to the host system
  - Cannot break standard cryptography like deducing a private key from cipher text
- **Out of scope**:
  - Attacks targeting application-specific vulnerabilities
  - Side channel attacks
  - Threats to availability: The host can start and stop the secured context, thus the attacker can as well
- We **trust the design and implementation** of the **confidential computing hardware** (i.e. AMD SEV-SNP)
- We assume the **CVM has been fully sealed** and its configuration can be measured as proposed by Revelio[4]

[4]: Anna Galanou, Khushboo Bindlish, Luca Preibsch, Yvonne-Anne Pignolet, Christof Fetzer, and Rüdiger Kapitza. 2023. Trustworthy confidential virtual machines for the masses. In Proceedings of the 24th International Middleware Conference. 316–328. https://doi.org/10.1145/3590140.3629124

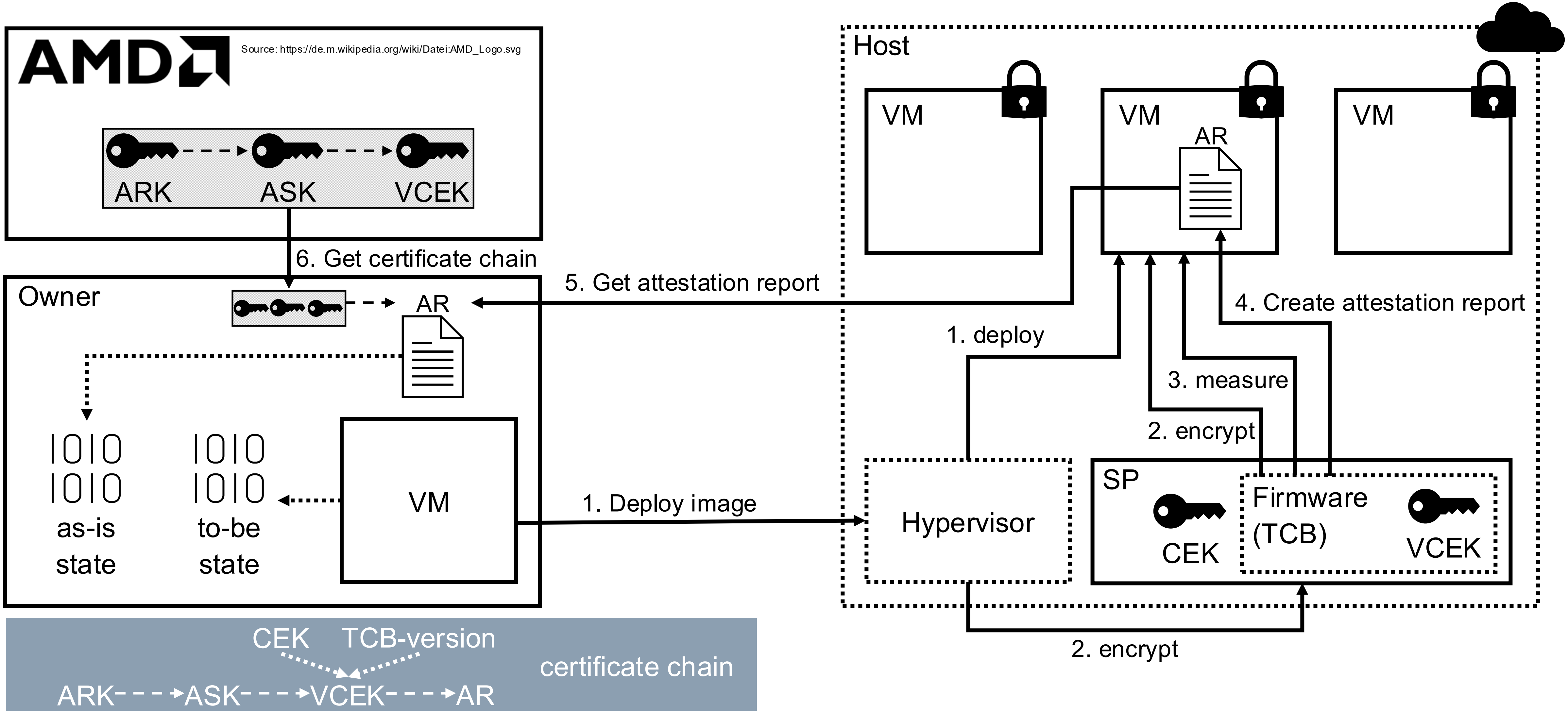
TLS Extension API (TLSExt)

- **Generic API** to create and parse TLS 1.3 extensions
- This keeps *site attestation's* application logic out of the browser
- TLSExt works by exposing functionality of Firefox's TLS library NSS
- To the right: The example representation of how a WebExtension writes TLS extensions using TLSExt



Background

AMD SEV-SNP & Remote Attestation



Evaluation

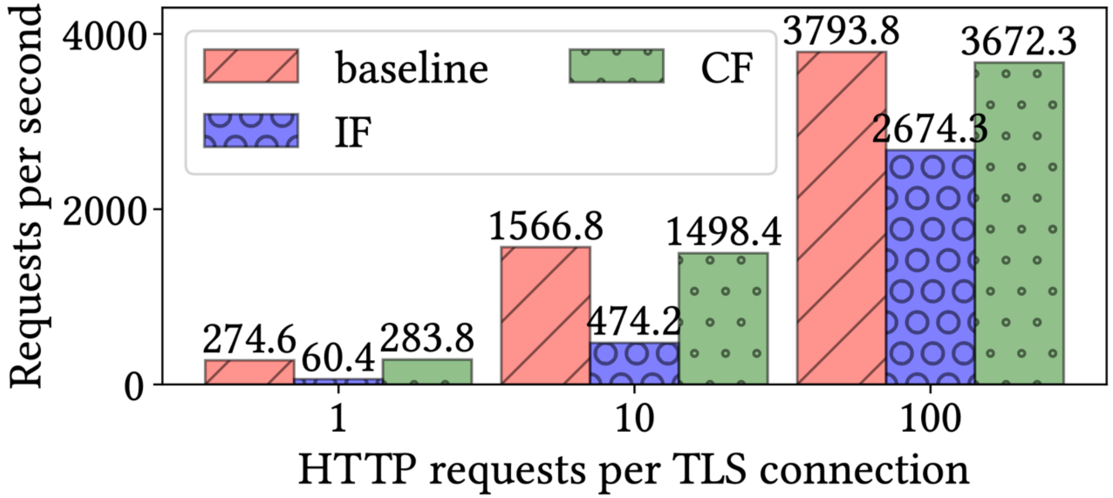
Server Side

- **Impact of attestation report generation**

	Min	Max	Mean
Sequential	4.35ms	6.41ms	4.67ms
Parallel	4.35ms	53.72ms	21.87ms

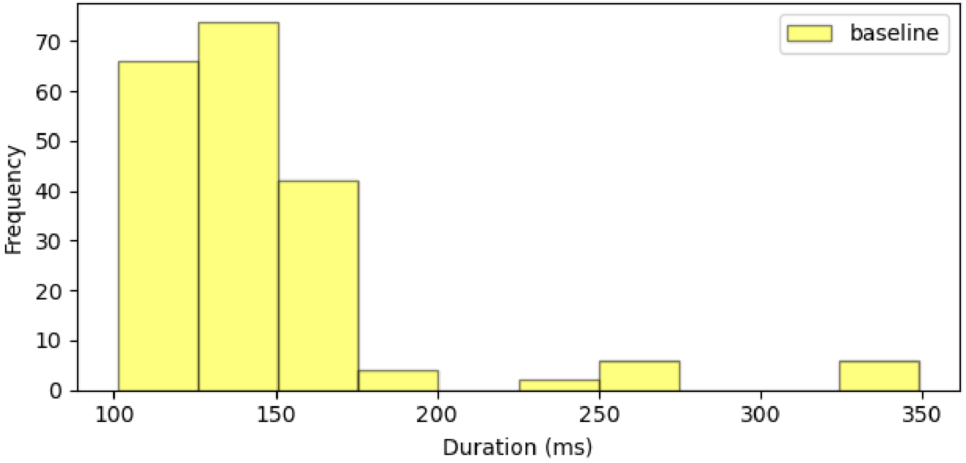
  - Device driver or AMD SP seem to be a bottleneck, parallel generation should be avoided

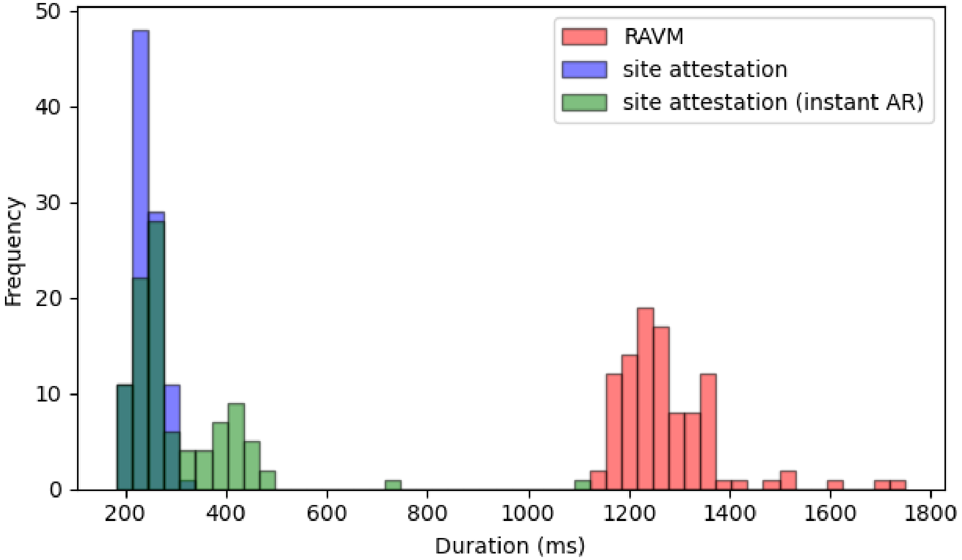
- **Server performance**

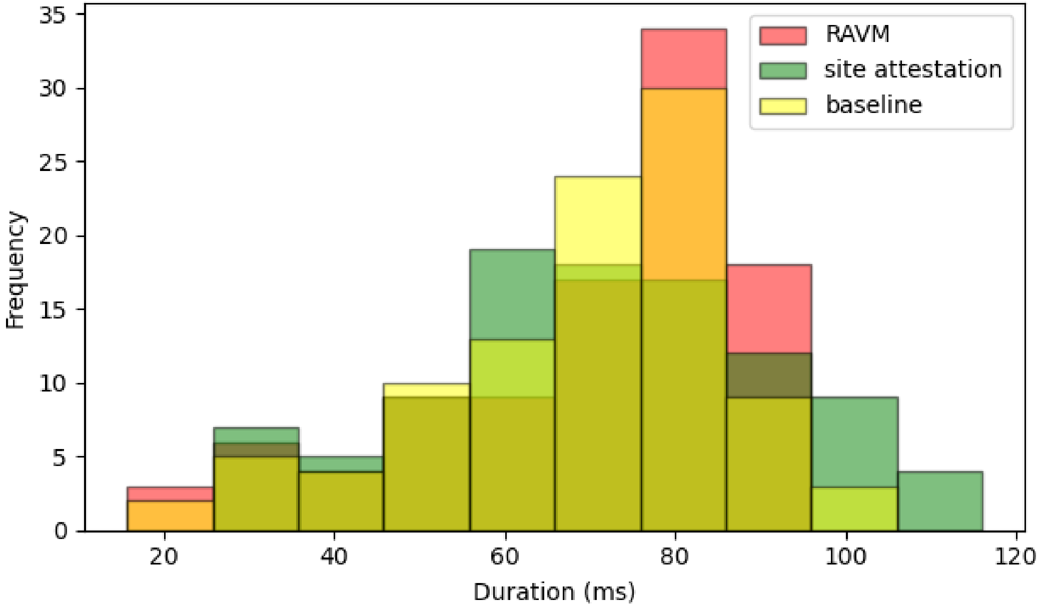


Evaluation

Client Side


baseline
Frequency
Duration (ms)


RAVM
site attestation
site attestation (instant AR)
Frequency
Duration (ms)


RAVM
site attestation
baseline
Frequency
Duration (ms)