TC4SE: Trusted Channel for Secure Enclave

Gilang Mentari Hamidy[1], Sri Yulianti[2], Pieter Philippaerts[1], Wouter Joosen[1]

[1]DistriNet – KU Leuven, Belgium, [2]Independent, Indonesia

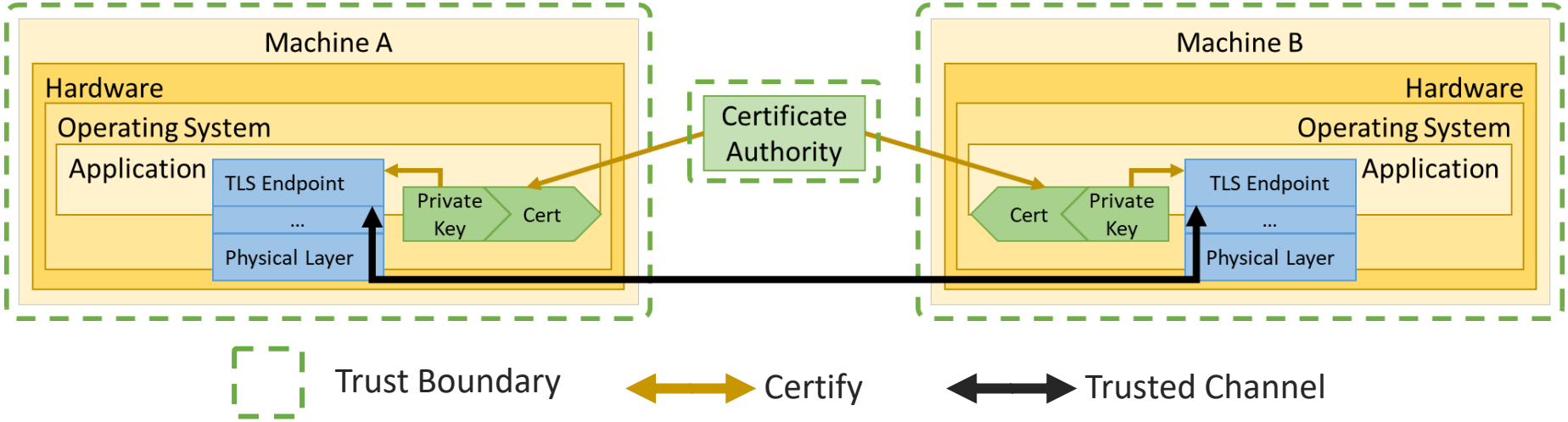
Published in November 2023 – (Re)-presented at FOSDEM 2025

KU LEUVEN DistriNet

Trusted Channel Between Two Remote Executions

Trusted Channel: *a secure channel where the trustworthiness is bound to the configuration of the endpoints* ***[Gasmi et al. - 2007]***

Baseline Trust TLS Secure Channel

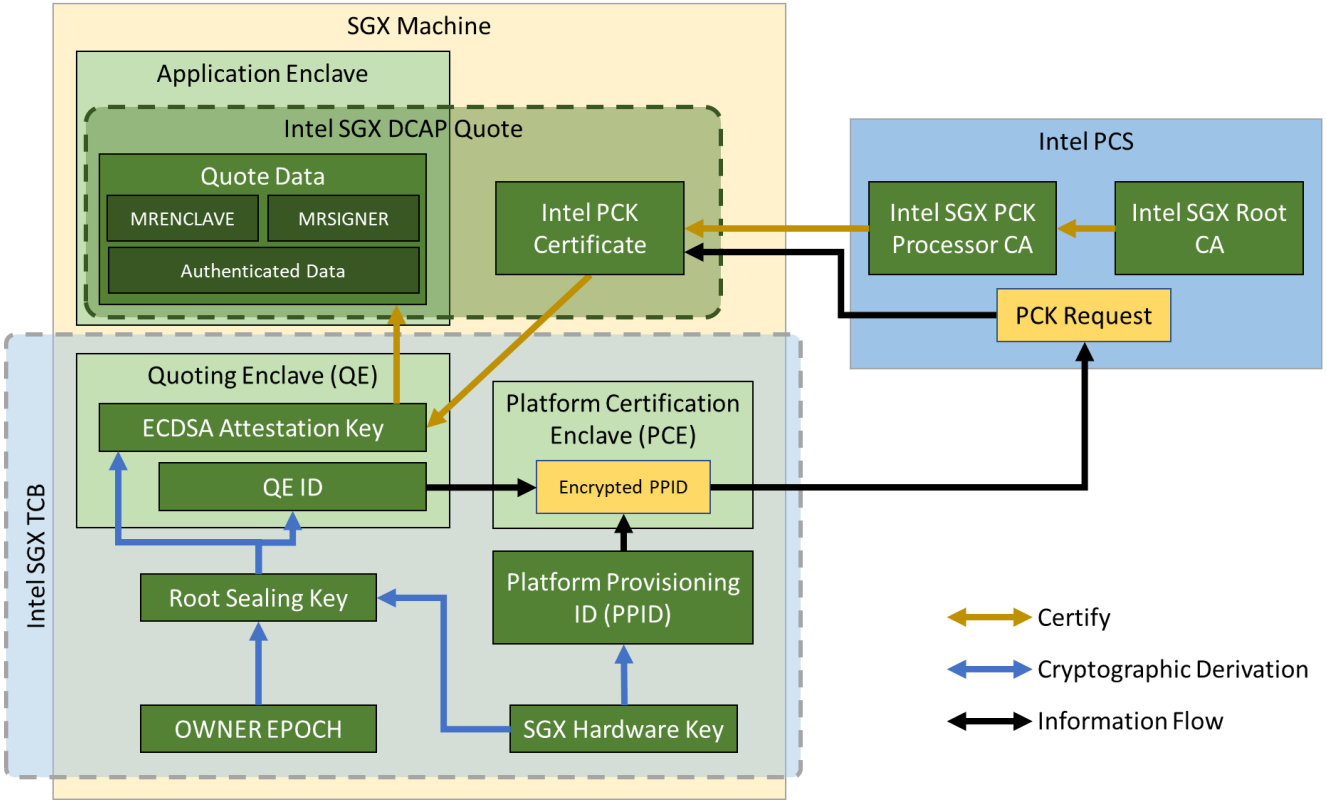


KU LEUVEN DistriNet

The Initial Question

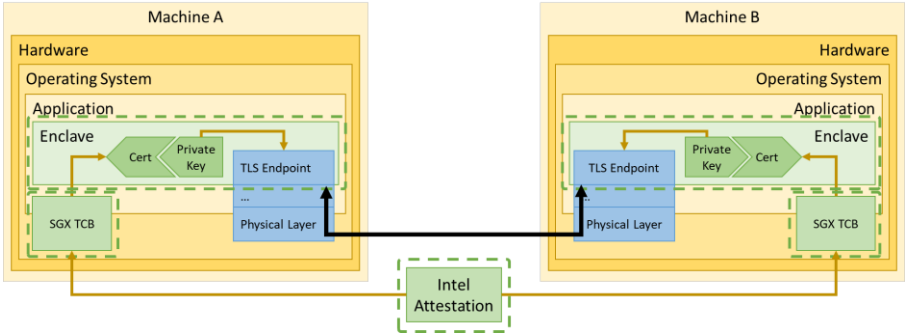
- **How can we establish trusted channel to communicate between two remote secure enclave?**
  - Where should the trusted channel endpoint terminate?
    - **Inside the enclave boundary**
  - How can we verify the trusted channel is established by the rightful trusted execution?
    - **Attestation**
  - How to mutually identify each other to achieve full trust on the trusted channel?
    - **Key Attestation**

Trust Chain of Intel SGX DCAP



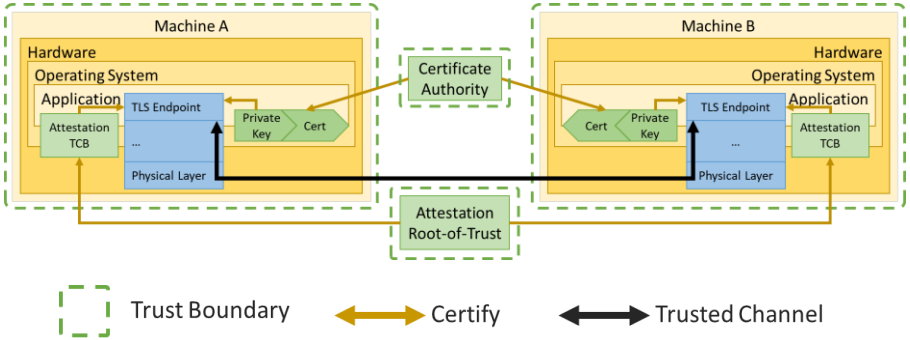
State-of-the-Art of Trusted Channel for SGX

Intel RA-TLS (Knauth, et al.)



- Quotes the public key and attach it in a self-signed certificate, generated on the start of the enclave alongside with the keypair.
- Quote verification occurs during the TLS handshake, verifying the trust chain to Intel SGX root-of-trust

Trusted Socket Layer (Niemi, et al.)



- Not necessarily specific to SGX, but TEE in general
- Quotes the TLS handshake parameters, generated and verified during the handshake
- Binds the trusted channel session to the trust chain

Identified Limitation on Trusted Channel State-of-the-Arts

Dependency on Attestation Infrastructure

- Depends on the availability of the PCS infrastructure to establish trusted channel
- Observable behavior to differentiate with regular mutual TLS handshake
- Potential DoS-ing by severing connection to PCS infrastructure

Performance & Implementation Complexity

- Added overhead to verify the attestation each trusted channel handshake
- Adding extra attestation generation and verification logic within handshaking steps

Trust Chain and Boundary

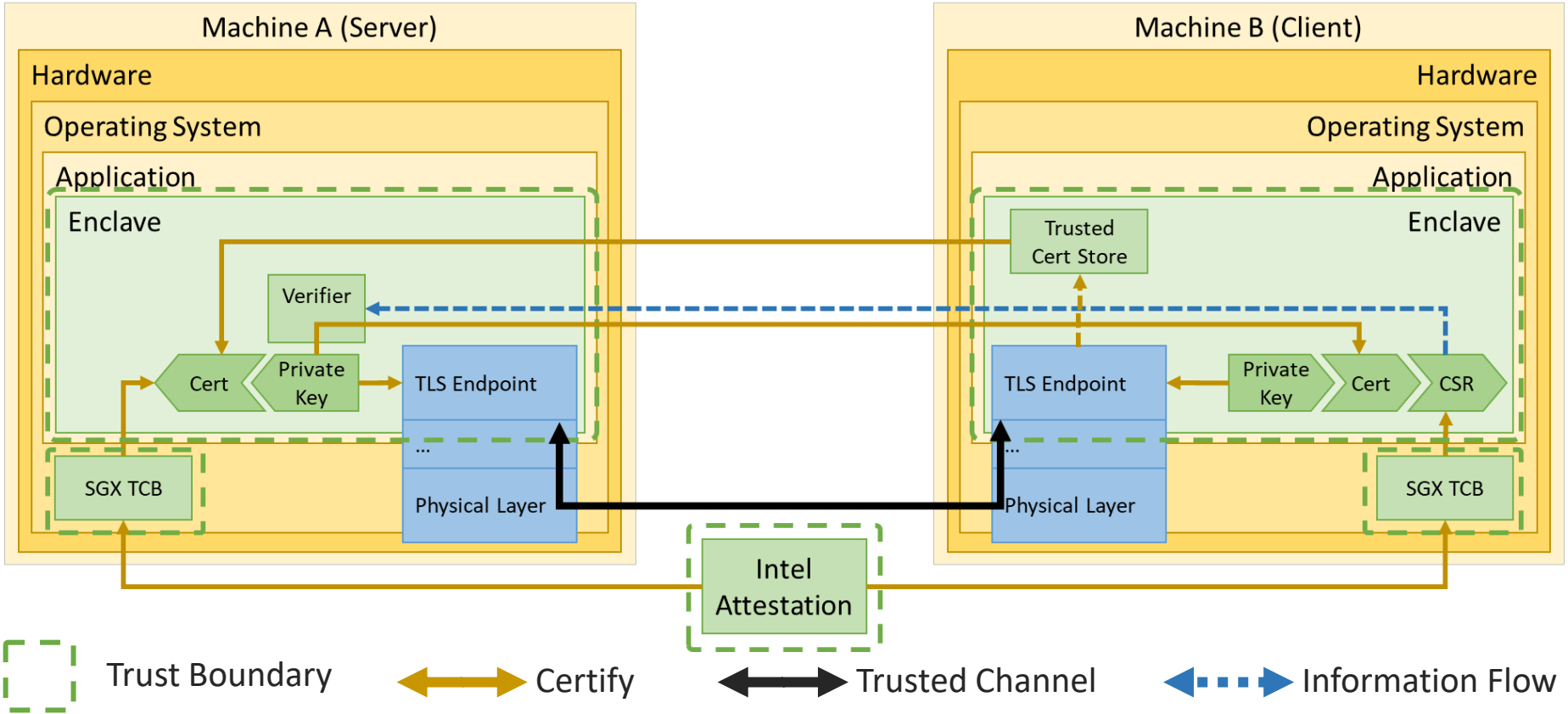
- Trust chain between the CA and attestation remains detached, require separate verification
- Intel RA-TLS design relies entirely on SGX root of trust, disregarding the usage of CA certificate (i.e., X.509 is used only as attestation container)

TC4SE: Trusted Channel for Secure Enclave

- **Simplifies trusted channel creation by delegating the trust to a private key within the secure enclave boundary**
  - Mutual trust is achieved by linking the cryptographic keypair to attestation protocol, i.e. **Key Attestation**
  - Trusted channel is then established through regular mutual-TLS authentication protocol
  - The trust can be preserved across multiple enclave instantiation through **sealing**

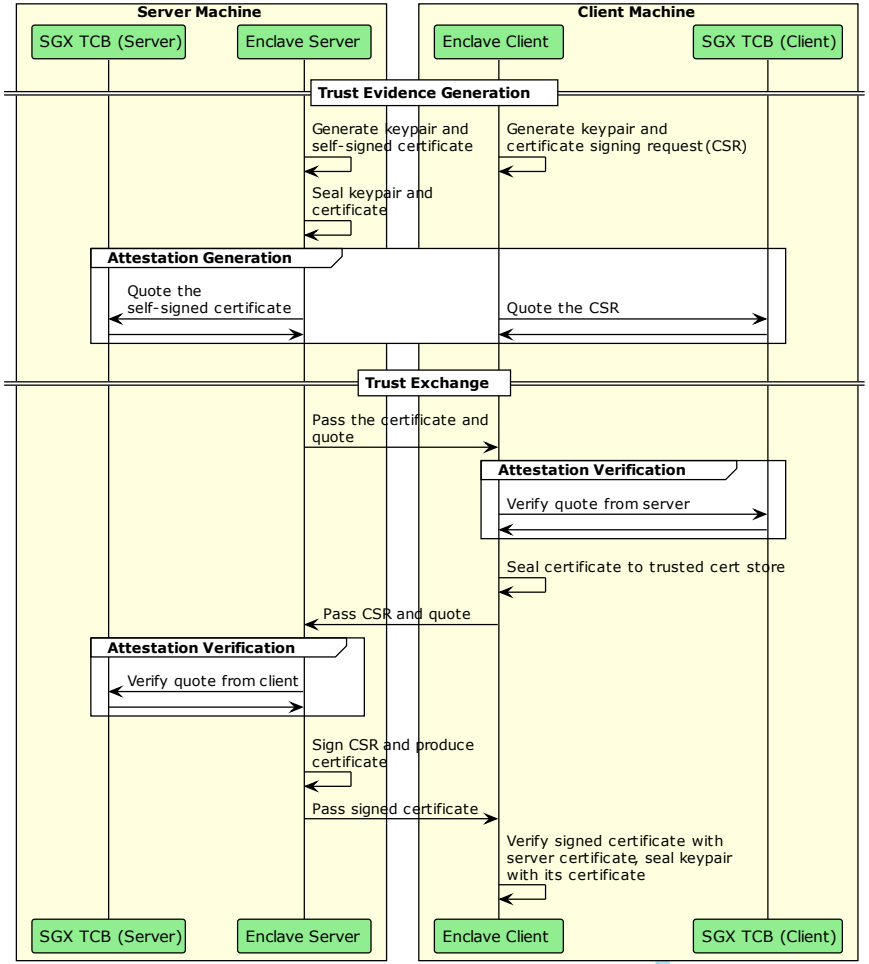
Action	RA-TLS		TSL		TC4SE	
	Initial Setup	Channel Initiation	Initial Setup	Channel Initiation	Initial Setup	Channel Initiation
Key Generation	●	-	●	-	●	-
Quote Generation	●	-	-	●	●	-
Quote Verification	-	●	-	●	●	-
TLS Handshake	-	●	-	●	●	●

TC4SE: Trusted Channel for Secure Enclave



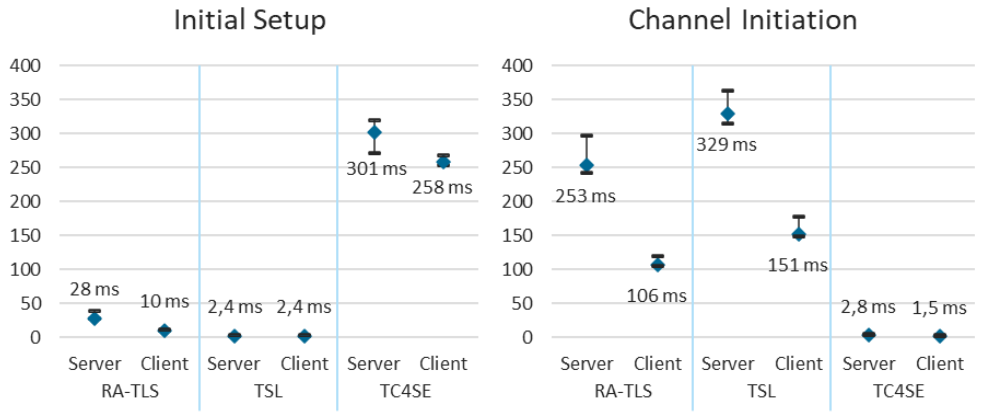
Establishing the Trust in TC4SE

- **Both parties generate their own keypair as the trust anchor**
  - Both parties link their public key through attestation
  - The server quotes the self-signed certificate to be sent to the client, which then client verifies and trust it as CA
  - The client quotes the Certificate Signing Request (CSR) to be sent to the server, which then server signs the CSR for the client
  - The trusted CA certificate and the signed certificate is then used to establish trusted channel via mTLS



Evaluation

- **TC4SE outperforms other previous works**
  - TC4SE performs as much work as possible during the one-time initial setup phase
- **TC4SE takes a significantly longer initial setup phase, but also significantly shorter channel initiation phase**
- **TC4SE uses lower network load especially for channel initiation phase where the enclaves are establishing the secure channel**



Direction	RA-TLS		TSL		TC4SE	
	Initial Setup	Channel Initiation	Initial Setup	Channel Initiation	Initial Setup	Channel Initiation
Client to Server	-	5,80 KB	-	5,81 KB	5,33 KB	1,89 KB
Server to Client	-	16,59 KB	-	17,01 KB	7,71 KB	3,14 KB
Combined	**-**	**22,39 KB**	**-**	**22,82 KB**	13,04 KB	**5,02 KB**

Implementation and Its Challenges

- **TC4SE provides a reference implementation**

- **For airtight scenario, it is possible to strip-down the Intel DCAP infrastructure**

- **This scenario may also be applied in the enclave container solution (Gramine, Occlum) or possibly Intel TDX as well (as TDX also use DCAP for its attestation processes)**

Conclusion

- **TC4SE simplify establishing trusted channel between two remote enclaves**
  - Leverages common protocol (TLS 1.3) and mutual TLS
  - Links the security properties of TLS 1.3 with the SGX root-of-trust
- **We qualitatively compared several proposed approaches and identified potential issues that are addressed in TC4SE**
  - Reduce dependency on attestation infrastructure
  - Improve performance of the trusted channel
- **Basically a bootleg version of lamps-csr-attestation for Attested TLS**

TC4SE is Open Sourced @ GitHub
https://github.com/DistriNet/TC4SE

Thank You

Gilang Mentari Hamidy

gilang.hamidy@kuleuven.be

KU LEUVEN DistriNet

Security Considerations

R1 **End-to-End**	*Provide end-to-end encryption between two enclaves in separate physical machine*	TC4SE is built on top of existing TLS 1.3 protocol, which already guarantees end-to-end encryption
R2 **Authenticated**	*Every party involved must be able to identify themselves and verify the peer authenticity*	• TC4SE uses mutual-TLS (mTLS) to authenticate peers • The certificate/key used in the mTLS is attested through the TEE/SGX attestation mechanism, thus cryptographically bound to the TCB
R3 **Indistinguishable**	*MITM that inspects the communication between two endpoints cannot distinguish the handshake over other regular TLS handshake*	• TC4SE performs attestation verification outside of the TLS handshake, hence external observers see the regular mTLS handshake sequence during handshaking