TEE² – Combining Trusted Hardware to Enhance the Security of TEEs

Master-Thesis of Fritz Alder

In cooperation with Aalto University, Finland

TECHNISCHE UNIVERSITÄT DARMSTADT

Aalto University

Supervisor: Prof. Katzenbeisser

Supervisor at Aalto University: Dr. Andrew Paverd and Prof. Asokan

Motivation

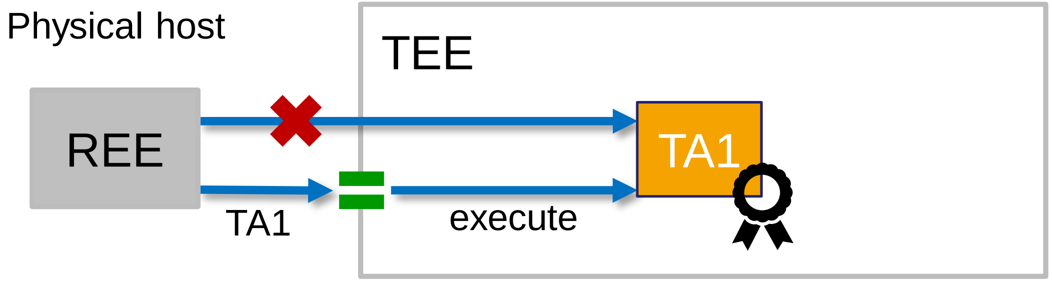
- Trusted Execution Environments (TEEs) provide isolated execution of security sensitive pieces of code that can be attested by remote parties.
  - ➢ TEEs have the potential to ensure security in cloud computing environments
- Real-world TEEs and their implementations are prone to attacks and bugs
  - ➢ Trust in real-world TEEs is difficult to achieve, possibly slowing down adoption

➢ Can we combine multiple TEEs to achieve security even if all but one TEE is compromised?

Trusted Execution Environment (TEE)

Aalto University
TECHNISCHE UNIVERSITÄT DARMSTADT

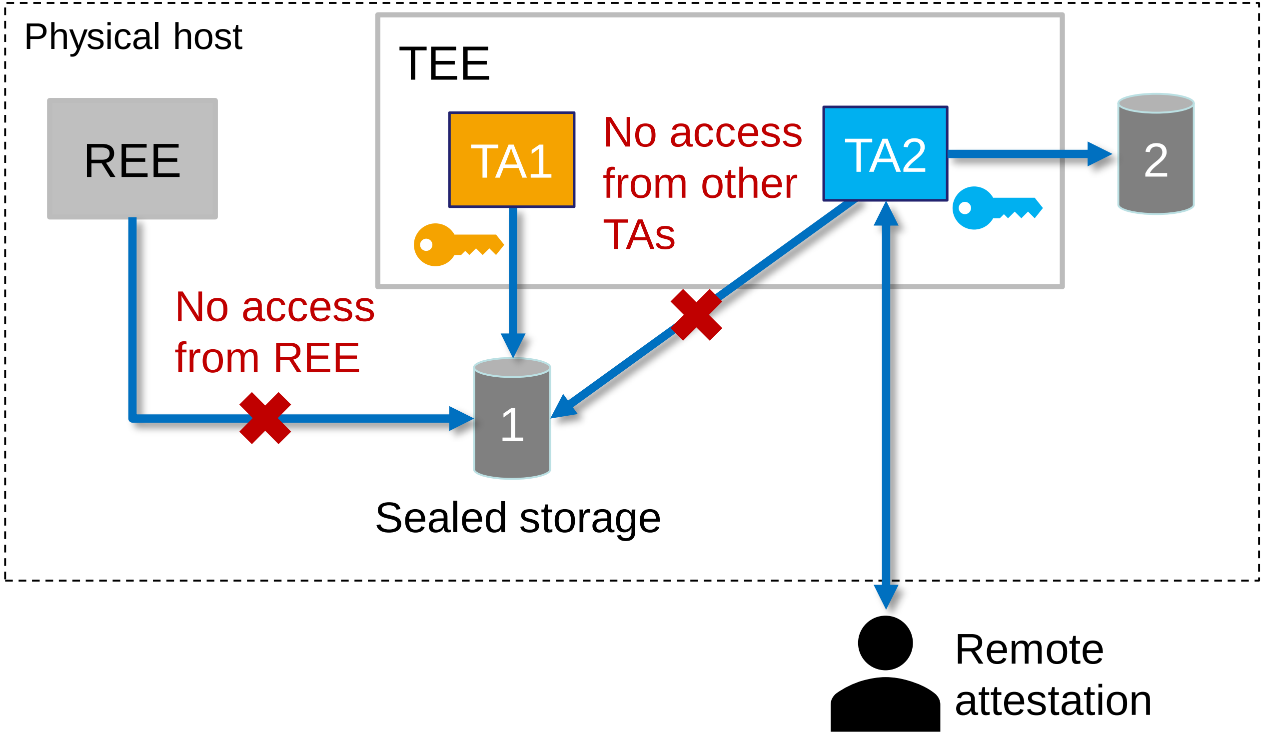
- Separated from Rich Execution Environment (REE)
- Executes Trusted Applications (TAs)
- Provides:
  - Code integrity
  - Isolated execution
  - Sealed data
  - TEE attestation



- Code integrity verified by TEE (e.g. with certificates)
- No direct access to TAs from REE
- Access only through predefined call-gates

Trusted Execution Environment (TEE)

- Separated from Rich Execution Environment (REE)
- Executes Trusted Applications (TAs)
- Provides:
  - Code integrity
  - Isolated execution
  - Sealed data
  - TEE attestation

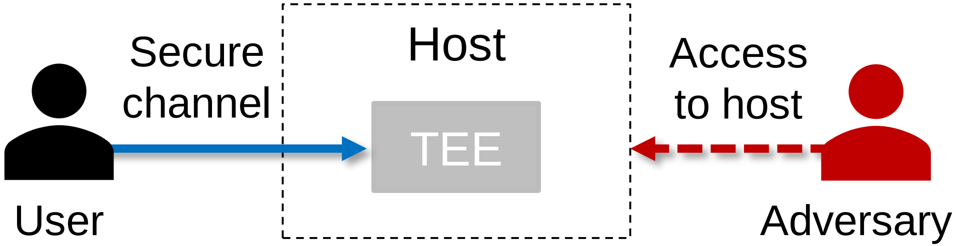


System model – *Ideal TEE*

Aalto University

TECHNISCHE UNIVERSITÄT DARMSTADT

- Similar to Honest but Curious cloud provider model:
  - User communicates with TEE
  - Adversary has full control of host
  - Adversary has no interest in DoS



- Adversary goal: Undermine any of the four TEE properties
  - Code integrity
  - Isolated execution
  - Sealed data
  - TEE attestation

Real-world TEE adversaries

Aalto University

TECHNISCHE UNIVERSITÄT DARMSTADT

Weak attacker W

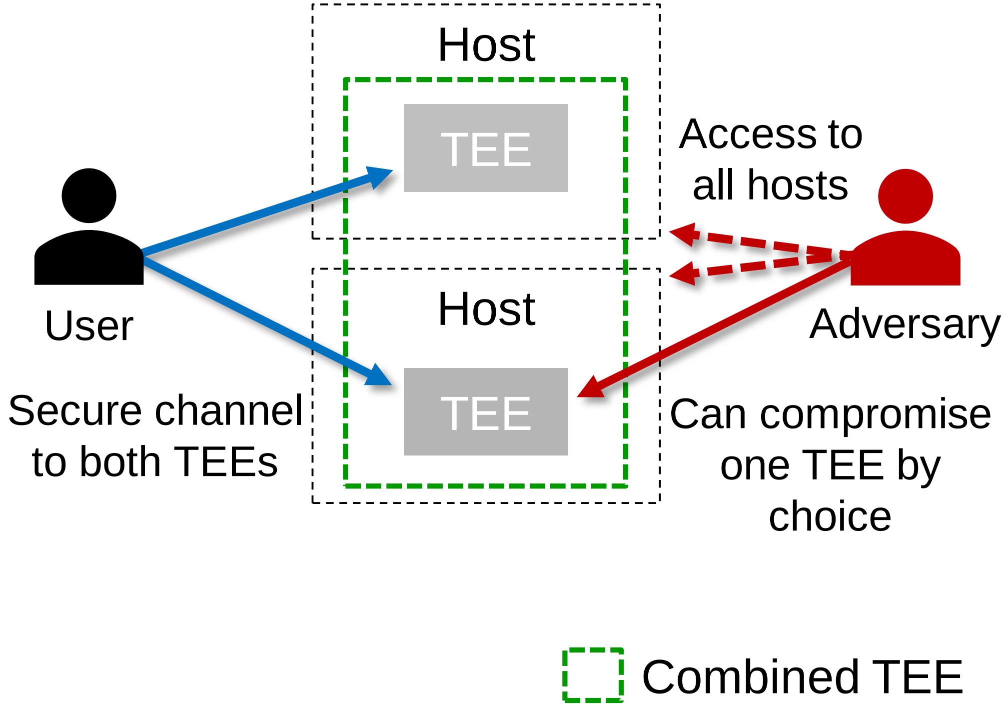
- Compromises TEE confidentiality
  - ➢ Can read run-time secrets and sealed data
  - ➢ But: Can not fake attestations or impact TEE integrity

Strong attacker S

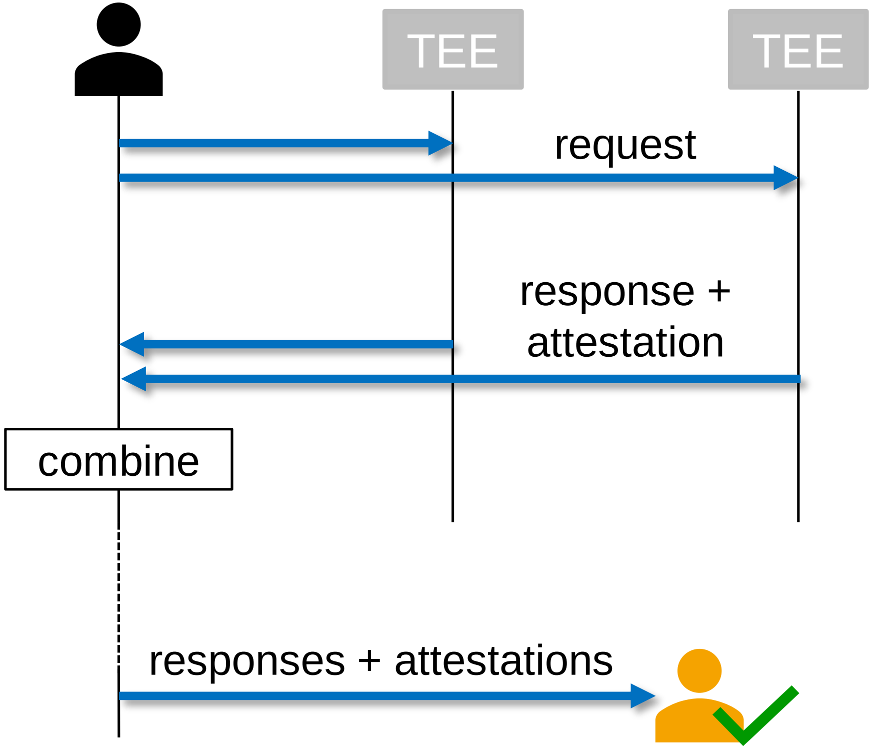
- Compromises TEE confidentiality and integrity
- Has access to architectural secrets or can influence TEE integrity
  - ➢ Can fully impersonate the TEE

Combined TEE – Design

- User communicates with two unique TEEs
- Adversary has full control over both untrusted hosts
- Adversary can choose to compromise any TEE
  - User stays unware of choice
- *Combined TEE* remains secure as long as at least one TEE is uncompromised



Random Number Generation



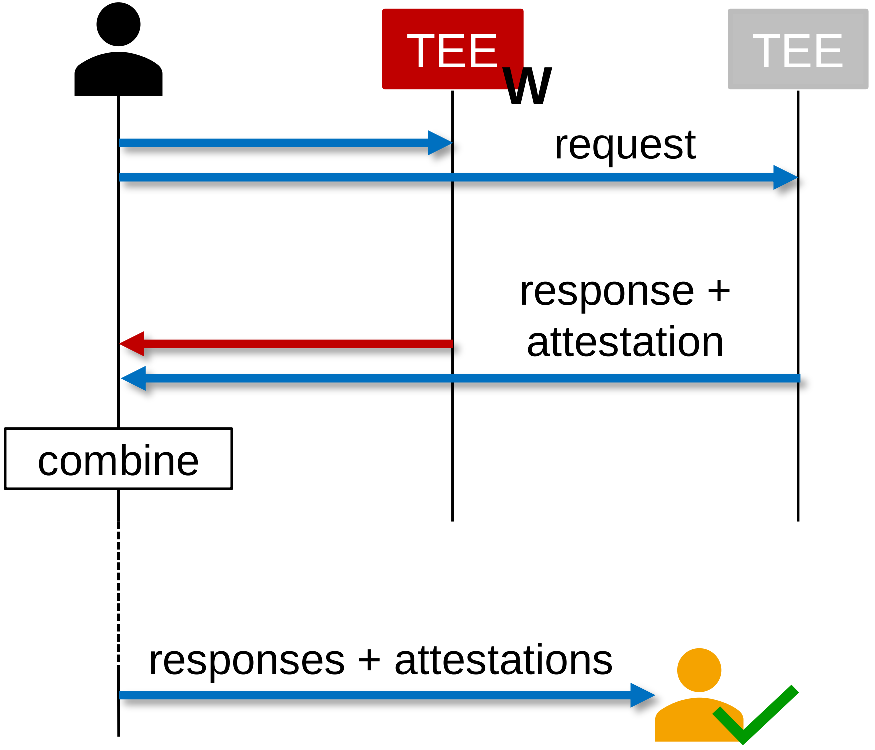
Goal:

1. Generate a random string..
2. ...that is unknown by an attacker
3. ...and can be attested by remote parties as being actually randomly generated

Random Number Generation – weak adversary

Aalto University

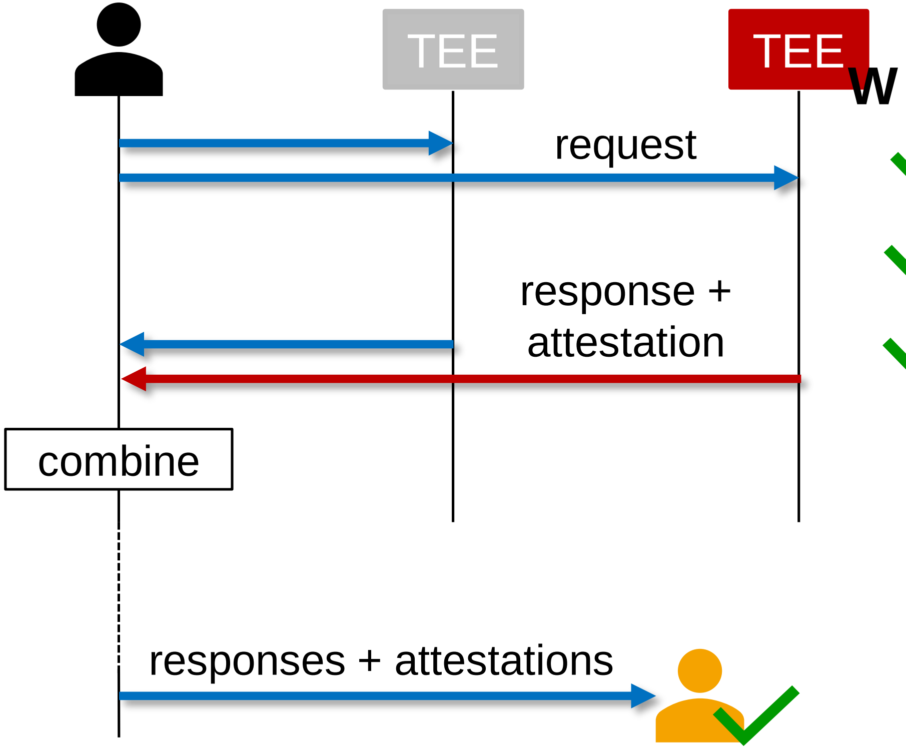
TECHNISCHE UNIVERSITÄT DARMSTADT



Goal:

1. Generate a random string..
2. ...that is unknown by an attacker
3. ...and can be attested by remote parties as being actually randomly generated

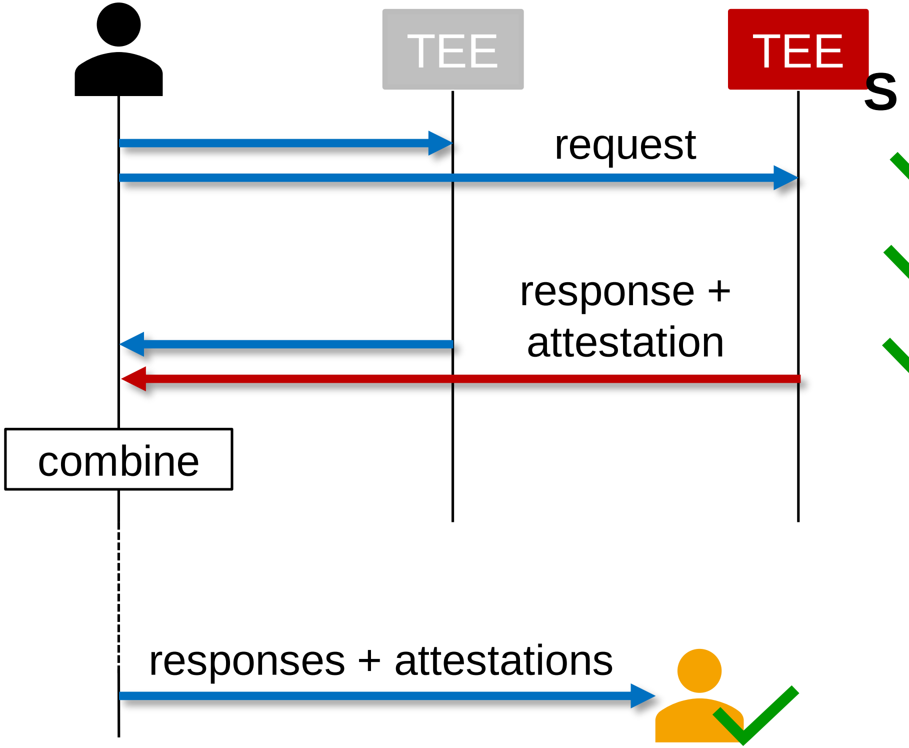
Random Number Generation – weak adversary



Goal:

1. Generate a random string..
2. ...that is unknown by an attacker
3. ...and can be attested by remote parties as being actually randomly generated

Random Number Generation–strong adversary



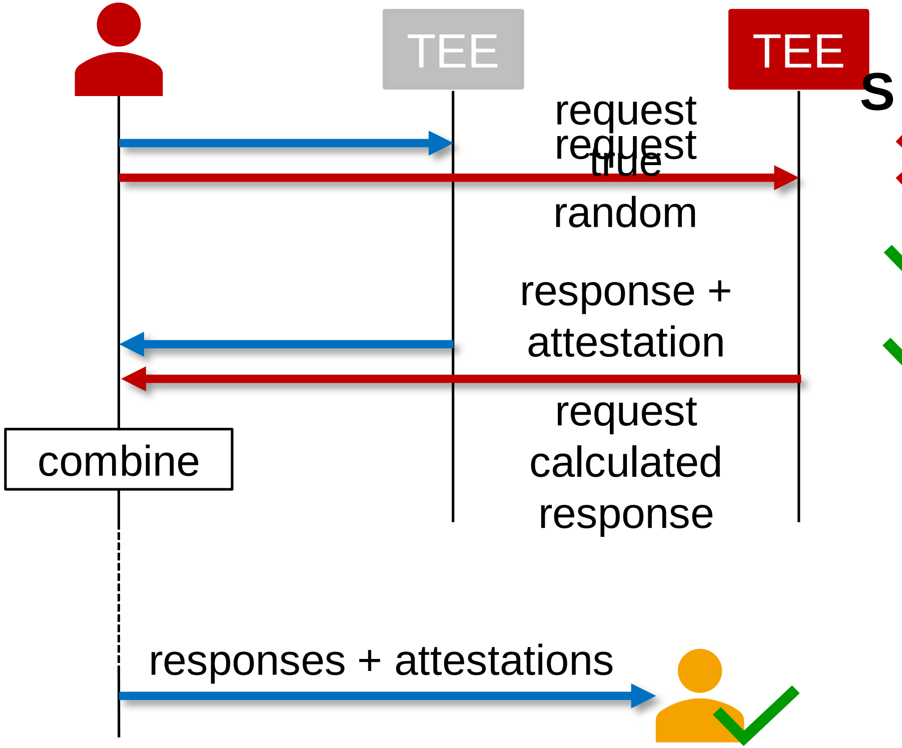
Goal:

1. Generate a random string..
2. ...that is unknown by an attacker
3. ...and can be attested by remote parties as being actually randomly generated

Random Number Generation–strong adversary

Aalto University

TECHNISCHE UNIVERSITÄT DARMSTADT



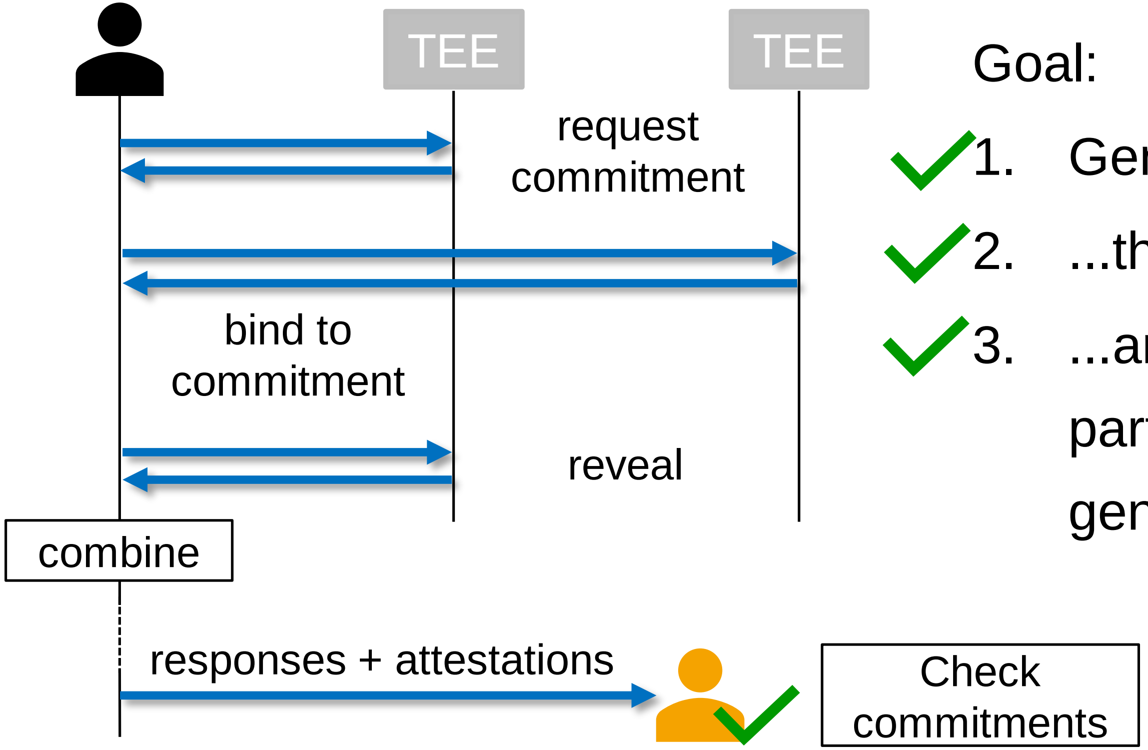
Goal:

1. Generate a random string..
2. ...that is unknown by an attacker
3. ...and can be attested by remote parties as being actually randomly generated

Random Number Generation–strong adversary

Aalto University

TECHNISCHE UNIVERSITÄT DARMSTADT



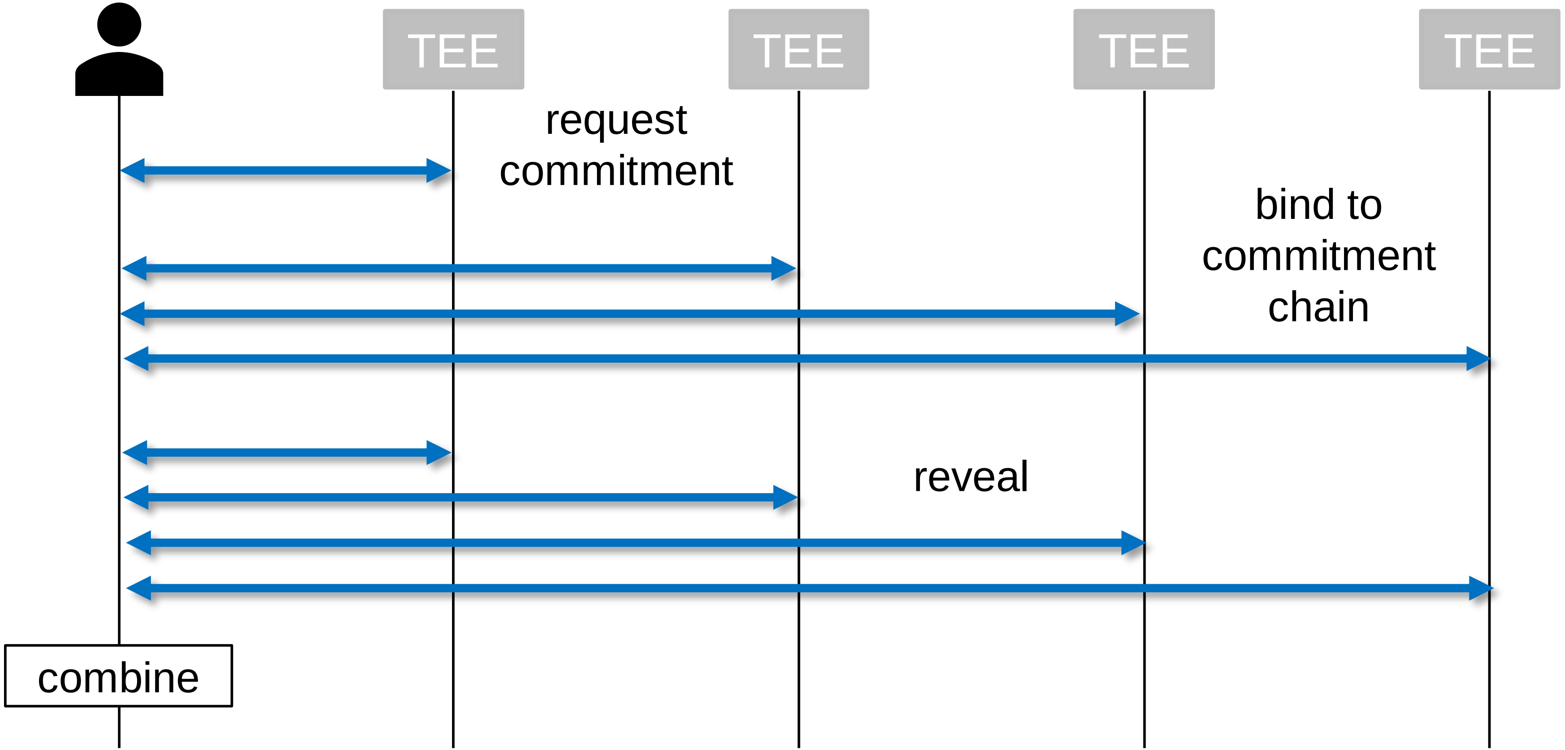
Goal:

1. Generate a random string..
2. ...that is unknown by an attacker
3. ...and can be attested by remote parties as being actually randomly generated

Random Number Generation–strong adversary

Aalto University

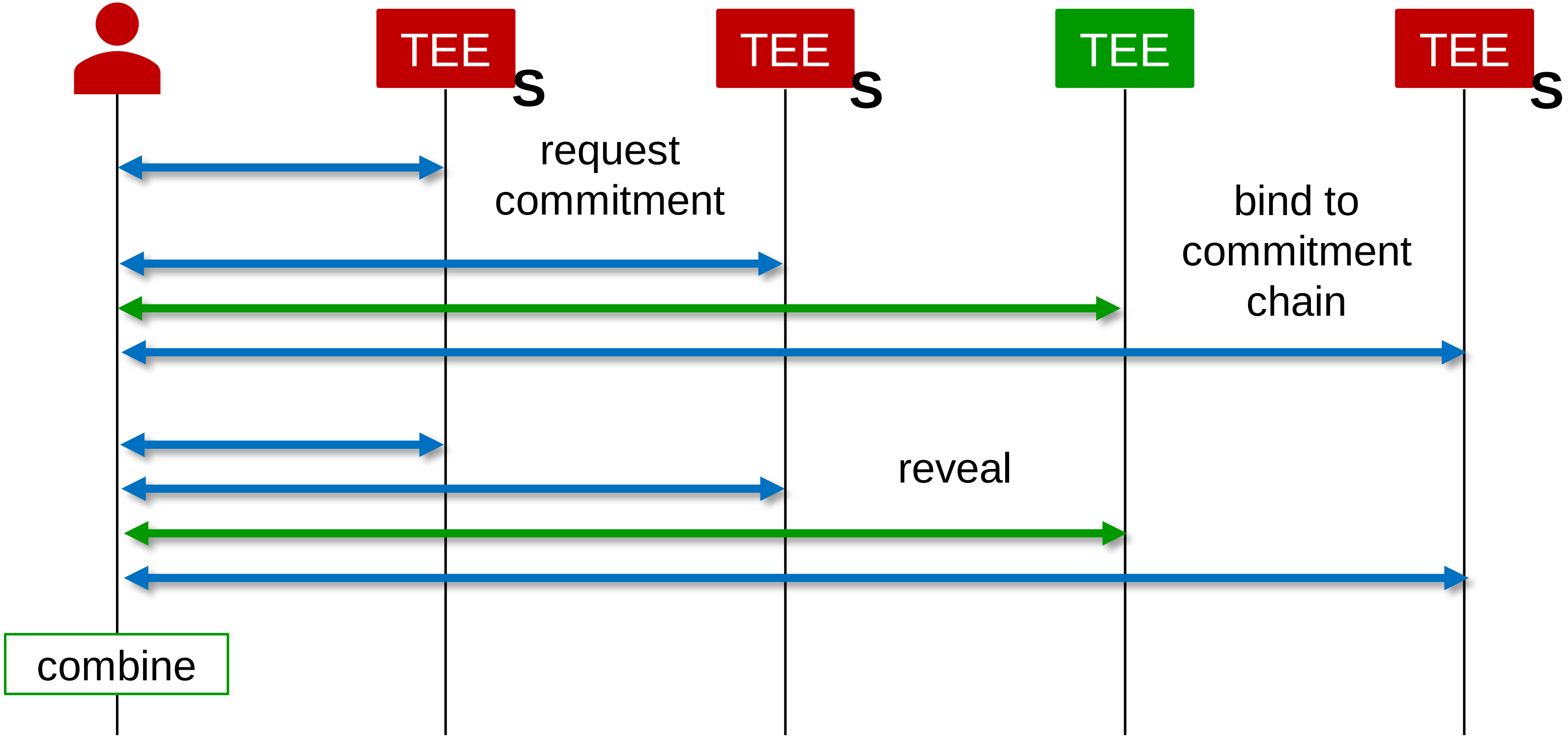
TECHNISCHE UNIVERSITÄT DARMSTADT



Random Number Generation–strong adversary

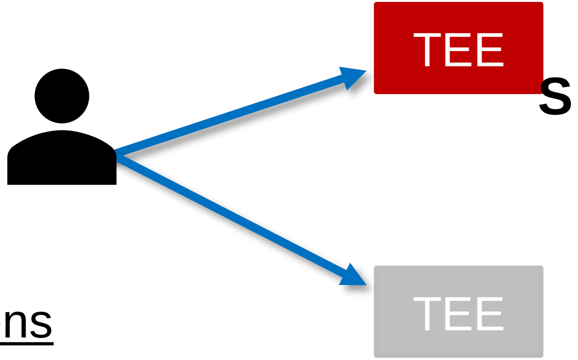
Aalto University

TECHNISCHE UNIVERSITÄT DARMSTADT

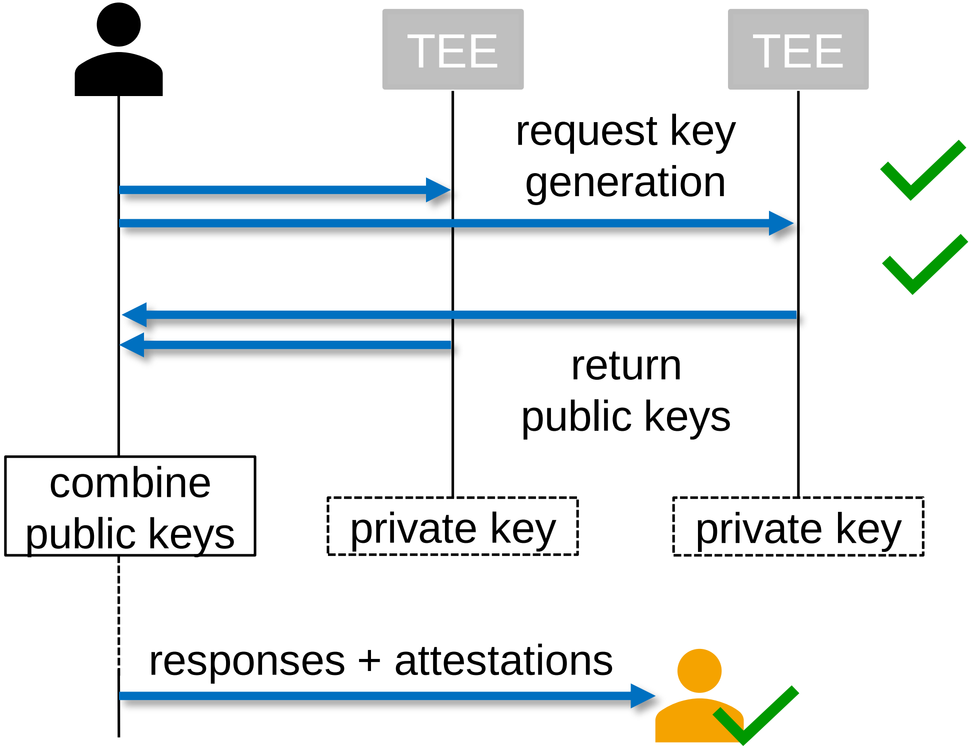


Protocol Design

- *Combined TEE* protocols differ from *Ideal TEE* protocols
  - No TEE can have knowledge of or control over any part of a secret
  - Instead, protocols need to protect against compromised TEEs
- Defined a range of utility, one-party, and two-party protocols
  - Key Exchange
  - Messaging
  - Random Number Generation
  - ElGamal operations
  - Signing
  - Store-and-forward
  - Oblivious Transfer



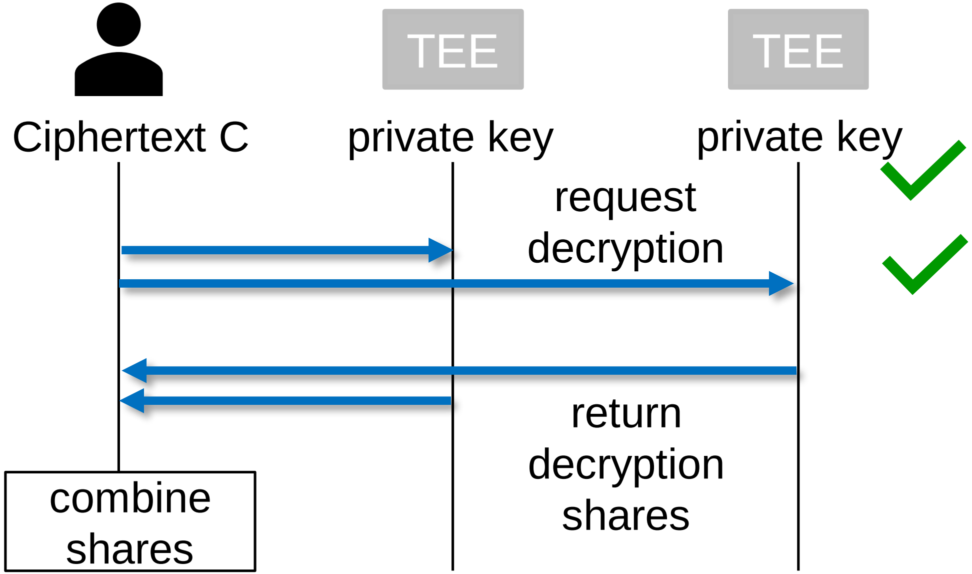
ElGamal operations – key generation



Goal:

1. Operate on private keys held by the TEEs... ✓
2. ...that are attestable ✓
3. ...and can not be learned during decryption
4. ...but can be used for confidential messages to the user

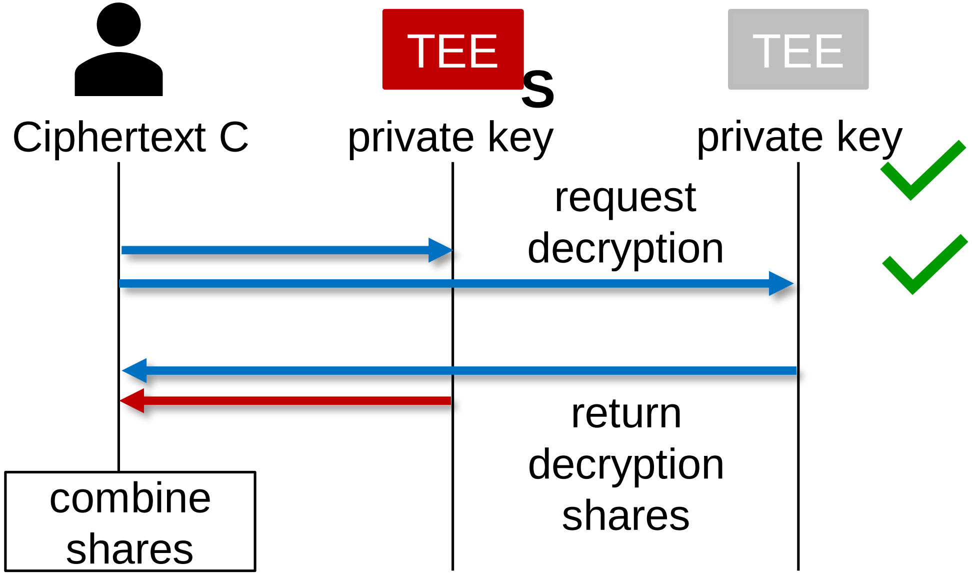
ElGamal operations – decryption



Goal:

1. Operate on private keys held by the TEEs... ✓
2. ...that are attestable ✓
3. ...and can not be learned during decryption
4. ...but can be used for confidential messages to the user

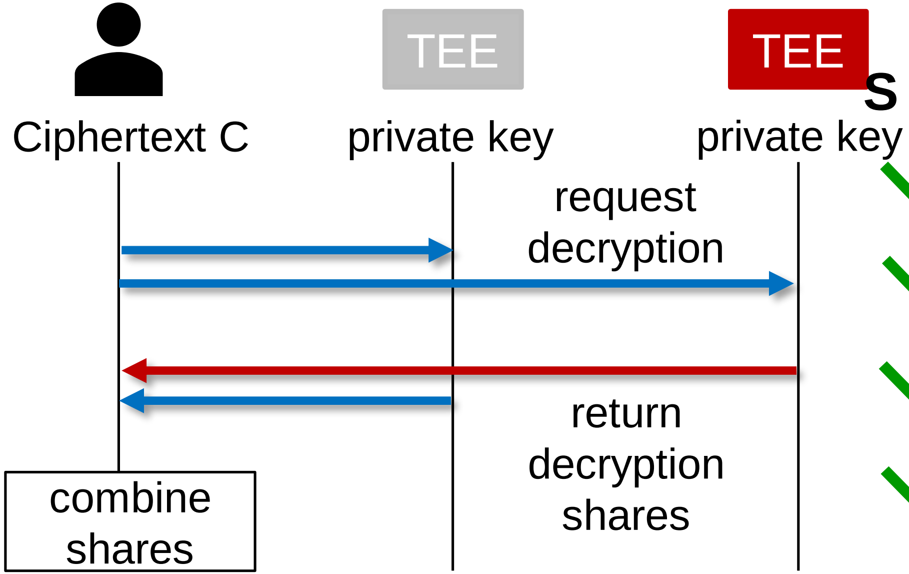
ElGamal operations – decryption



Goal:

1. Operate on private keys held by the TEEs...
2. ...that are attestable
3. ...and can not be learned during decryption
4. ...but can be used for confidential messages to the user

ElGamal operations – decryption



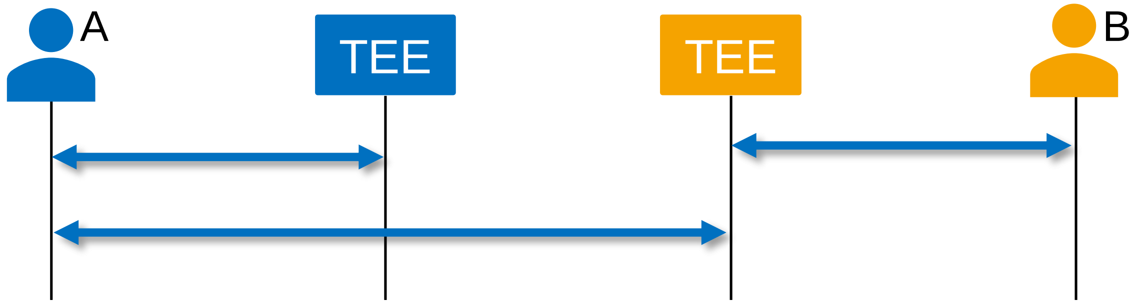
$y1 = x1 * G$; $y2 = x2 * G$; $Y = y1 + y2$;
$C1 = k*G$; $C2 = M + k*x1*G + k*x2*G$
$C = (C1,C2)$; $d1 = -x1 * C1$; $d2 = -x2 * C1$;
$M = C2 + d1 + d2$

Goal:

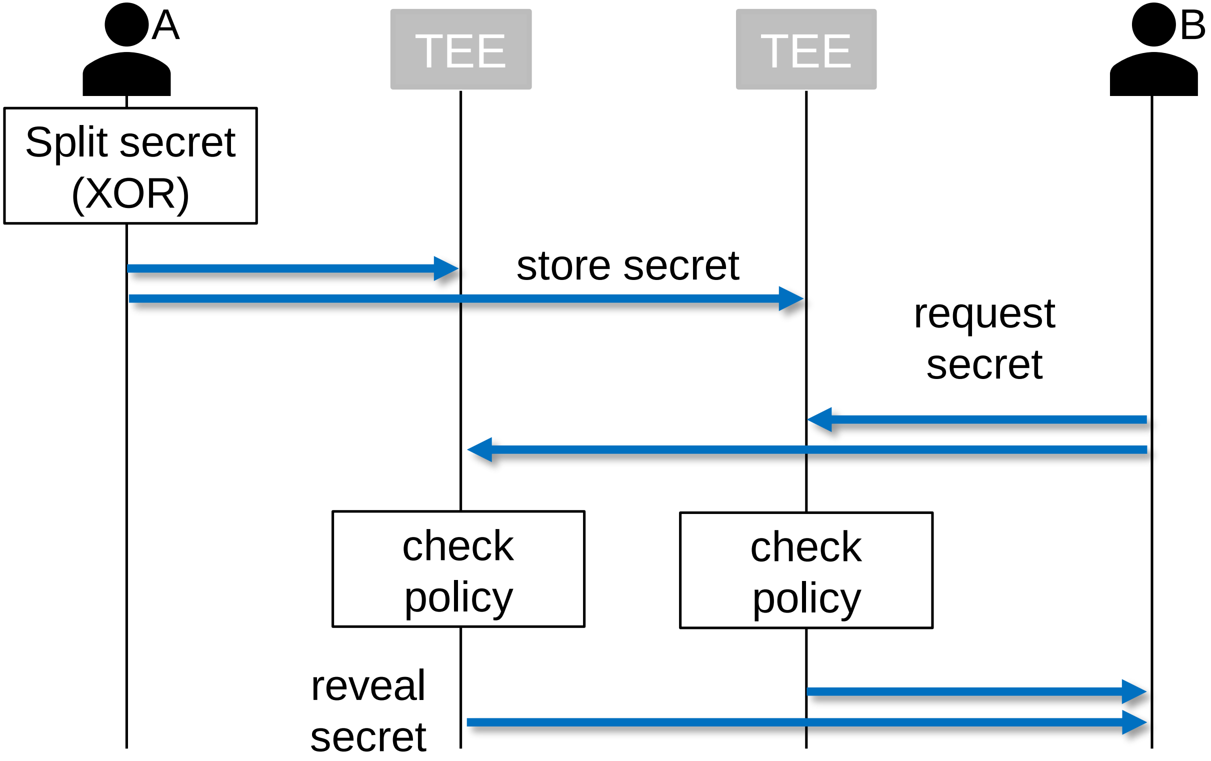
1. Operate on private keys held by the TEEs...
2. ...that are attestable
3. ...and can not be learned during decryption
4. ...but can be used for confidential messages to the user

Two-party protocols

- Both parties can try to cheat and can compromise N-1 TEEs
- ..but do not collaborate
- Protocols require active participation from both users
  - More than a simple attestation verification



Policy based store-and-forward

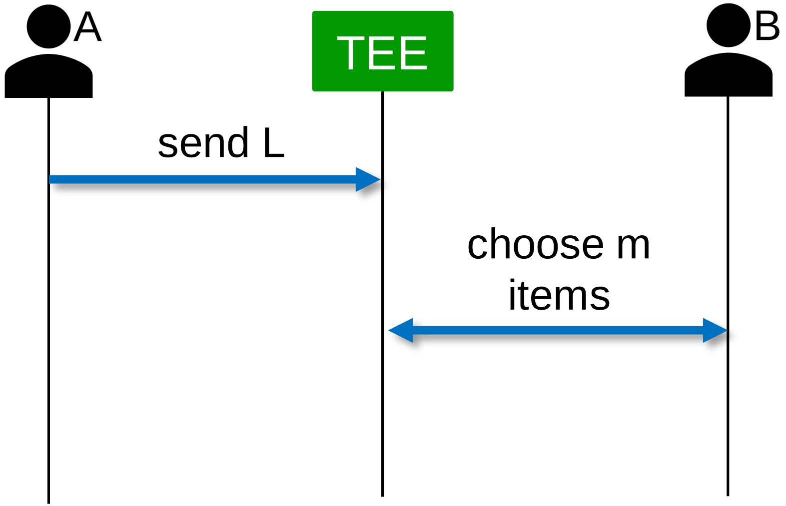


Goal:

1. Secretly share data with user B
2. Only B can reveal the secret
3. B can not reveal the secret if a policy is not matched

Jump to implementation

Oblivious transfer – ideal version

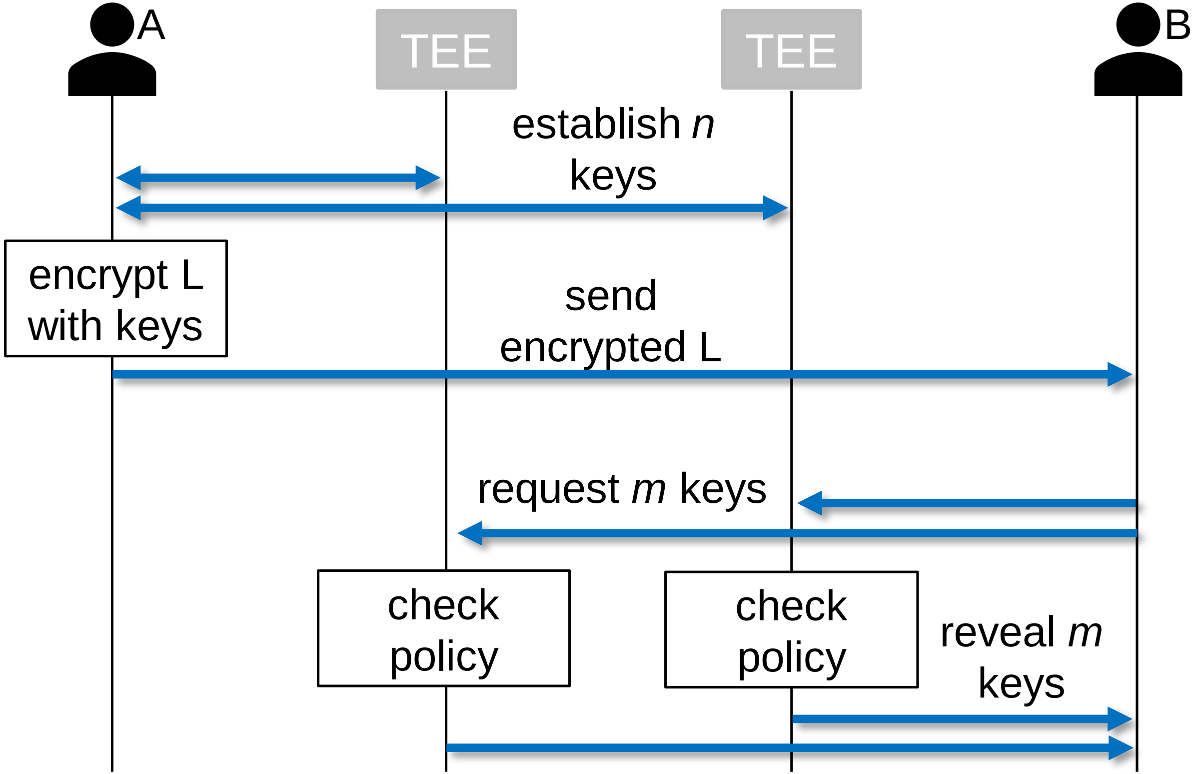


A has a list L of *n* items

Goals:

1. B can select up to *m* items
2. B should not learn more than *m* items
3. A should not learn B‘s choices (except the value of *m*)
4. No third party should learn any items or choices

Oblivious transfer – *Combined TEE*

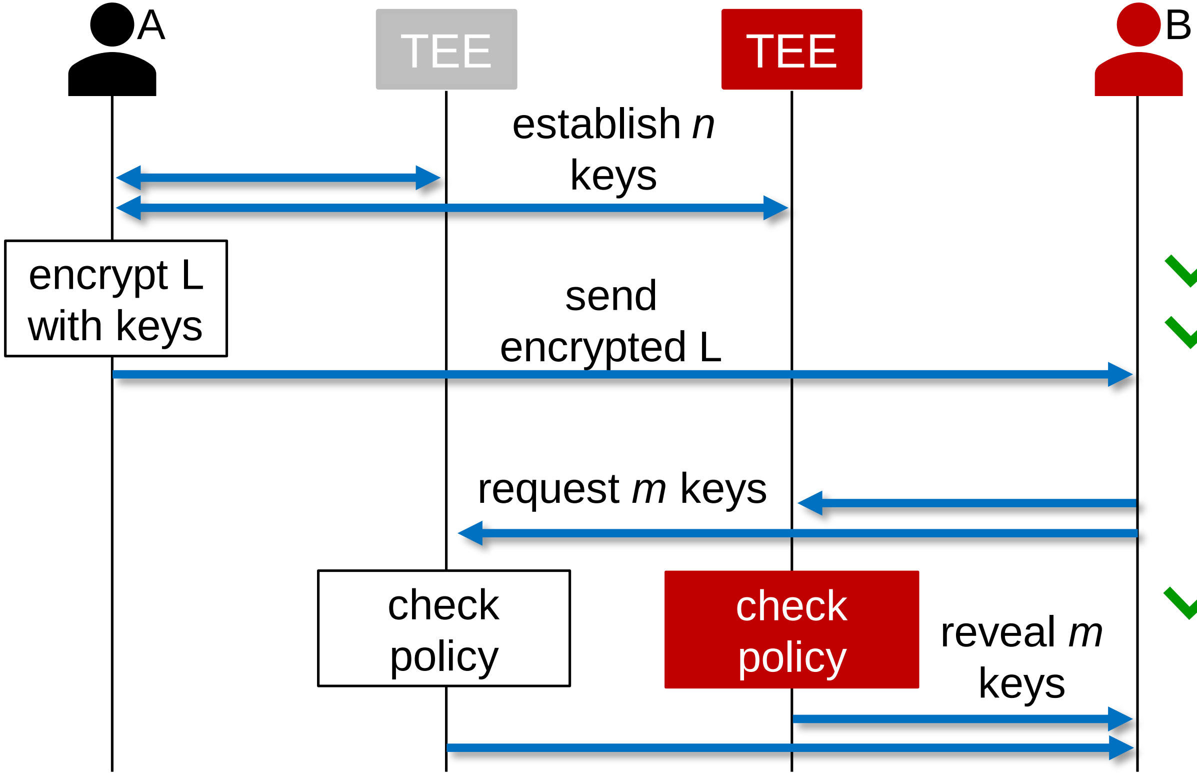


A has a list L of *n* items

Goals:

1. B can select up to *m* items
2. B should not learn more than *m* items
3. A should not learn B‘s choices (except the value of *m*)
4. No third party should learn any items or choices

Oblivious transfer – *Combined TEE*



A has a list L of *n* items

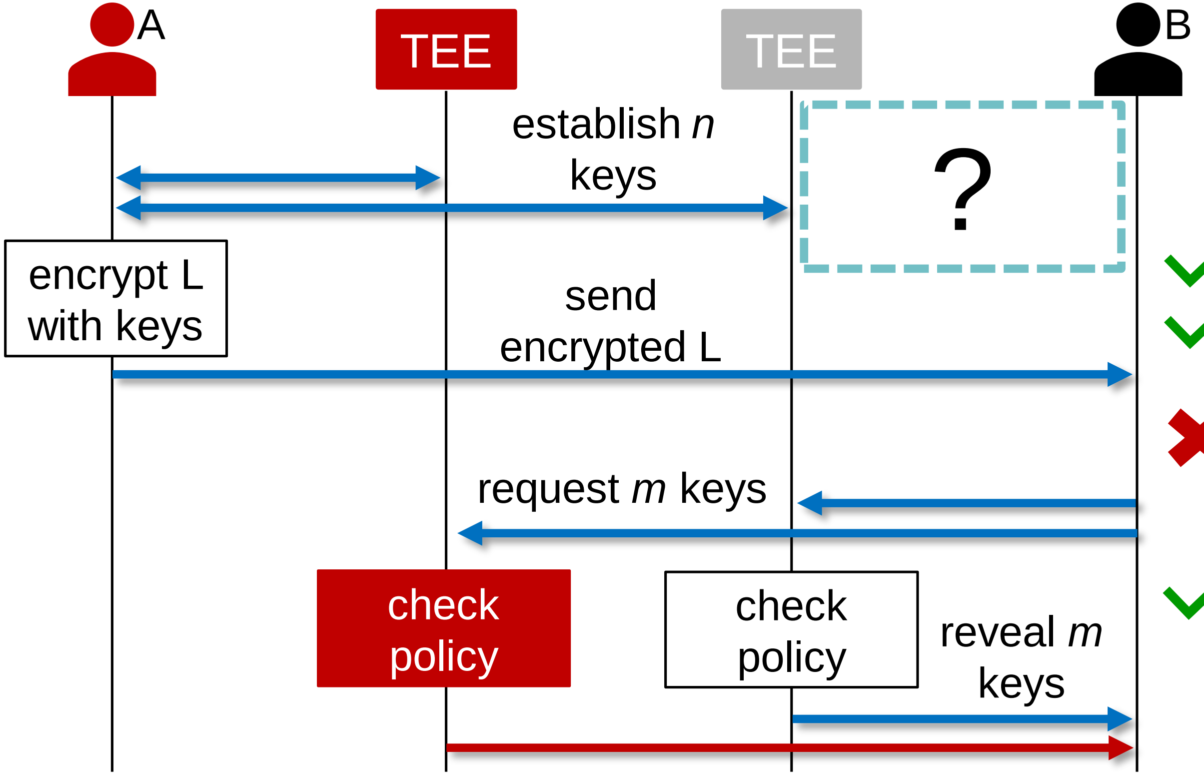
Goals:

1. B can select up to *m* items
2. B should not learn more than *m* items
3. A should not learn B‘s choices (except the value of *m*)
4. No third party should learn any items or choices

Oblivious transfer – *Combined TEE*

Aalto University

TECHNISCHE UNIVERSITÄT DARMSTADT

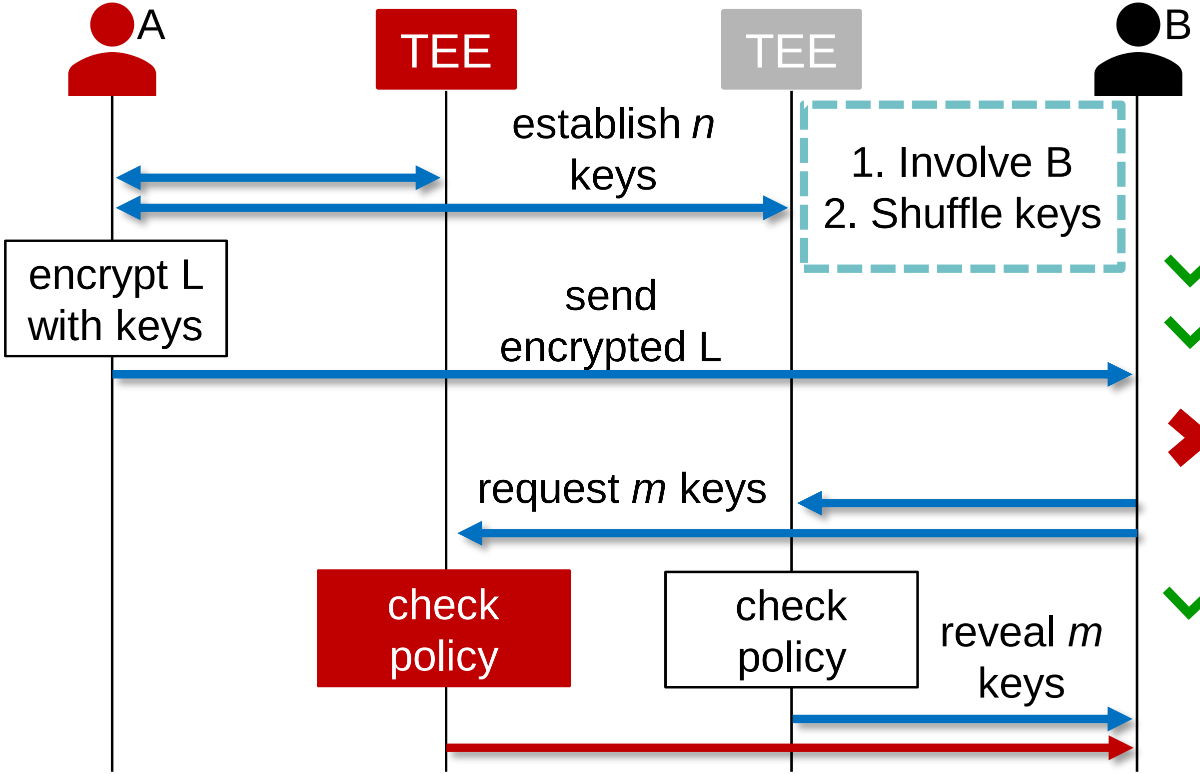


A has a list L of *n* items

Goals:

1. B can select up to *m* items
2. B should not learn more than *m* items
3. A should not learn B‘s choices (except the value of *m*)
4. No third party should learn any items or choices

Oblivious transfer – *Combined TEE*



A has a list L of *n* items

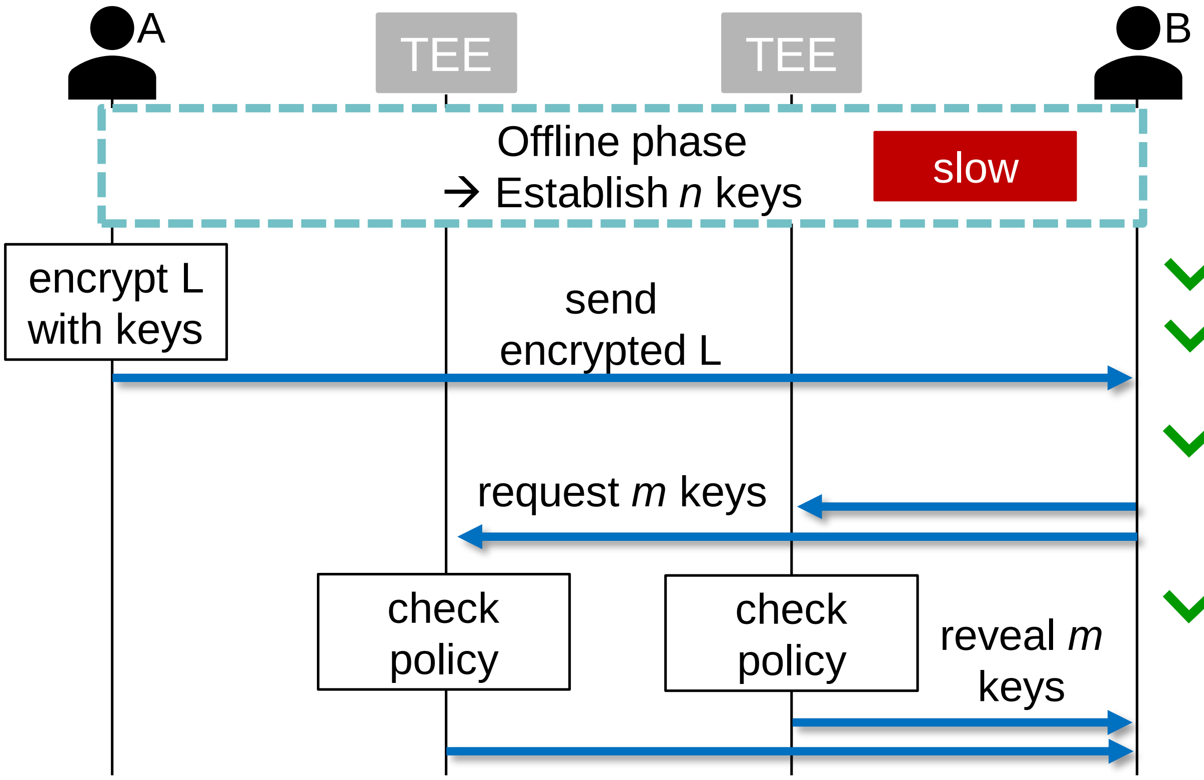
Goals:

1. B can select up to *m* items
2. B should not learn more than *m* items
3. A should not learn B‘s choices (except the value of *m*)
4. No third party should learn any items or choices

Oblivious transfer – *Combined TEE*

Aalto University

TECHNISCHE UNIVERSITÄT DARMSTADT



A has a list L of *n* items

Goals:

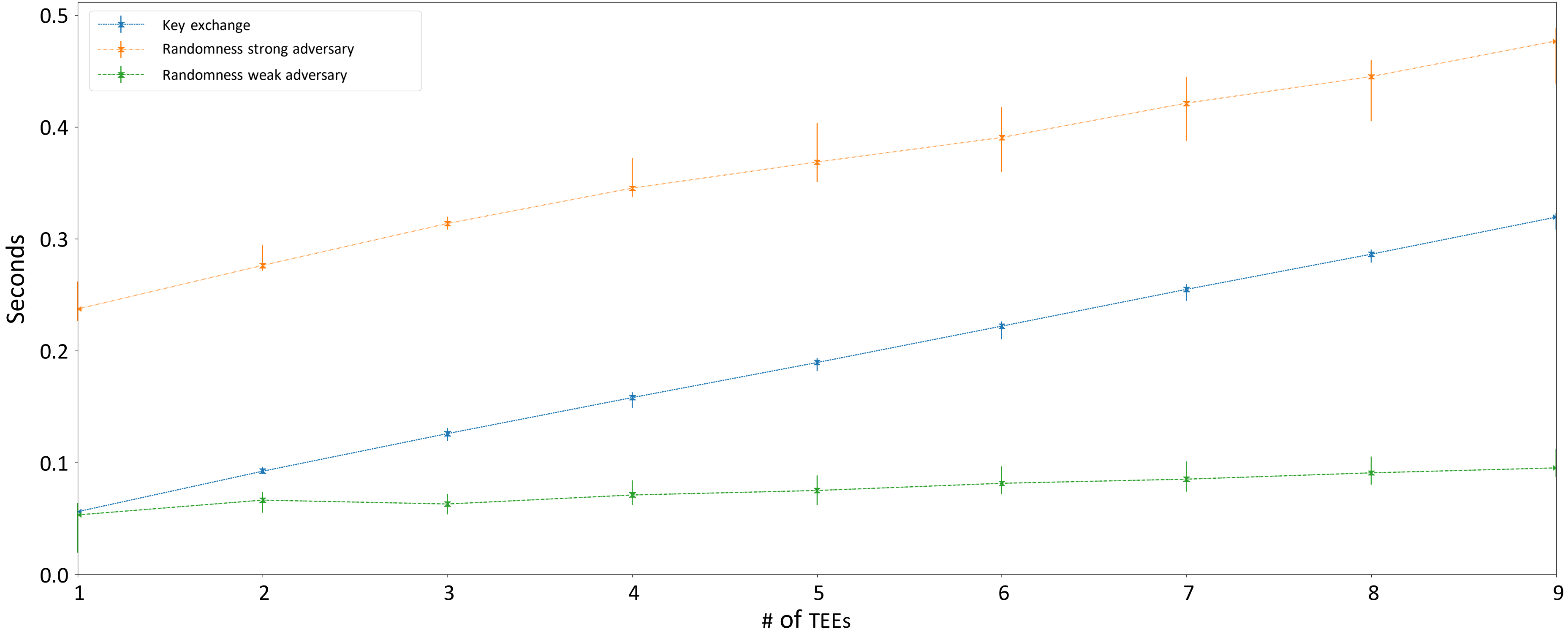
1. B can select up to *m* items
2. B should not learn more than *m* items
3. A should not learn B's choices (except the value of *m*)
4. No third party should learn any items or choices

Verification & Implementation

- Formally verified subset of protocols with the Tamarin Prover
  - Key exchange & RNG with weak adversary
  - Comparing security properties of *Combined TEE* protocols to security properties of *Ideal TEE* protocols

- Implemented subset of protocols with Intel SGX
  - Key exchange, both RNG protocols, and signature generation
  - Based on C++ (SGX) and Python (Client)
    - Passing JSON strings to exchange requests and responses

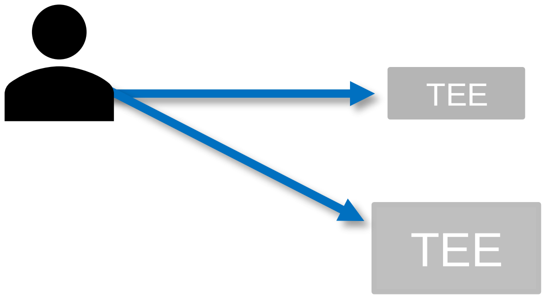
Evaluation

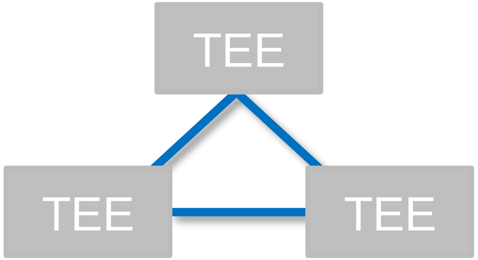


Key exchange
Randomness strong adversary
Randomness weak adversary
Seconds
0.0
0.1
0.2
0.3
0.4
0.5
1
2
3
4
5
6
7
8
9
# of TEEs

Future work

- Explore different system models
  - Small and Big TEE instead of system of equals
    - ARM Trustzone + Intel SGX
    - Key management enclave + disk encryption enclave
  - System of TEEs for safety and reliability
    - E.g. TEEs in a car: Parts may fail frequently
- Port complex applications
  - Only cryptographic building blocks so far
  - No real-world use case

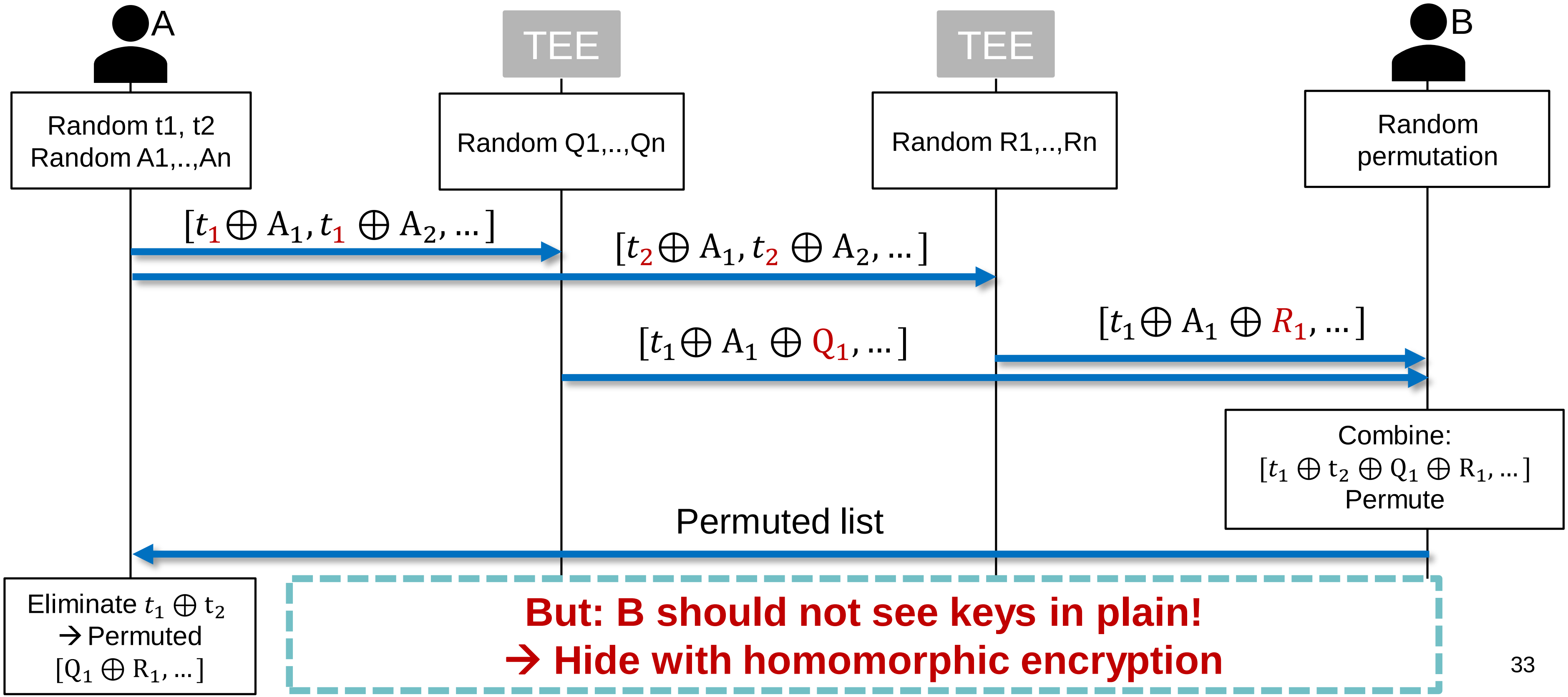




Summary

- TEEs have great potential
- 2 types of real-world adversaries exist: Weak and strong attackers
- Combined TEE alleviates impact of compromises
  - Range of protocols for arbitrary many cooperating TEEs: RNG, PK-Encryption, Signing, Store and forward, Oblivious Transfer
- Subset of protocols formally verified with Tamarin prover
- Implementation based on Intel SGX and Python
  - Shows a reasonable performance overhead for cryptographic building blocks
- Future work: Port complex applications, explore different TEE^2 scenarios (big-small TEE combination, multiple TEEs for safety and reliability, etc)

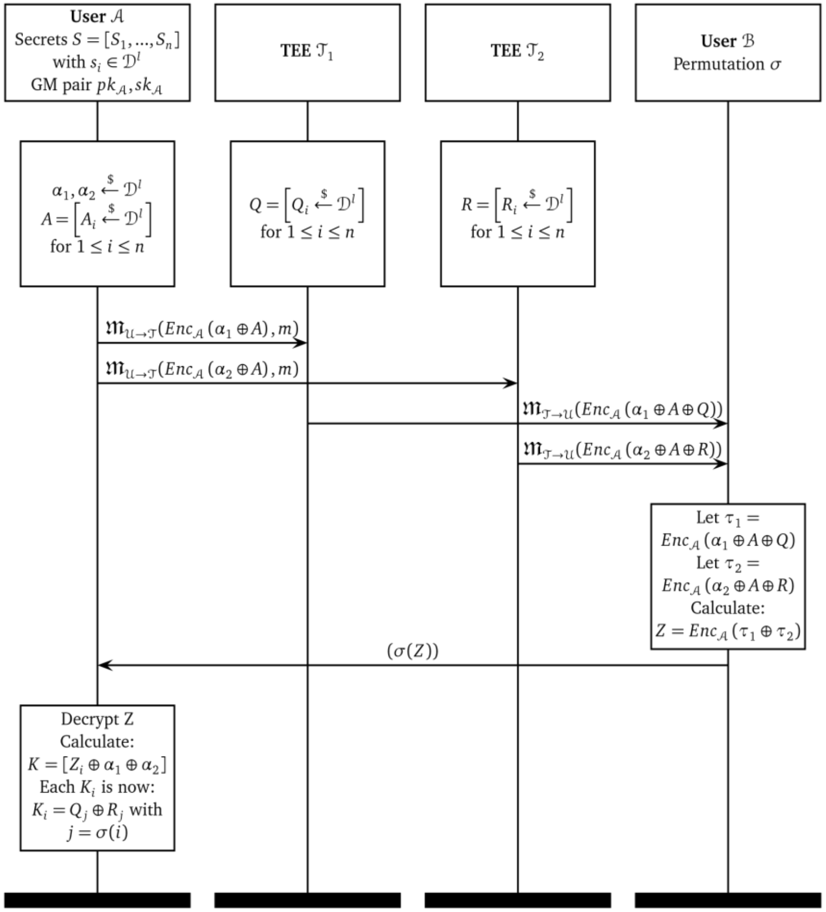
Oblivious transfer – *Offline phase*



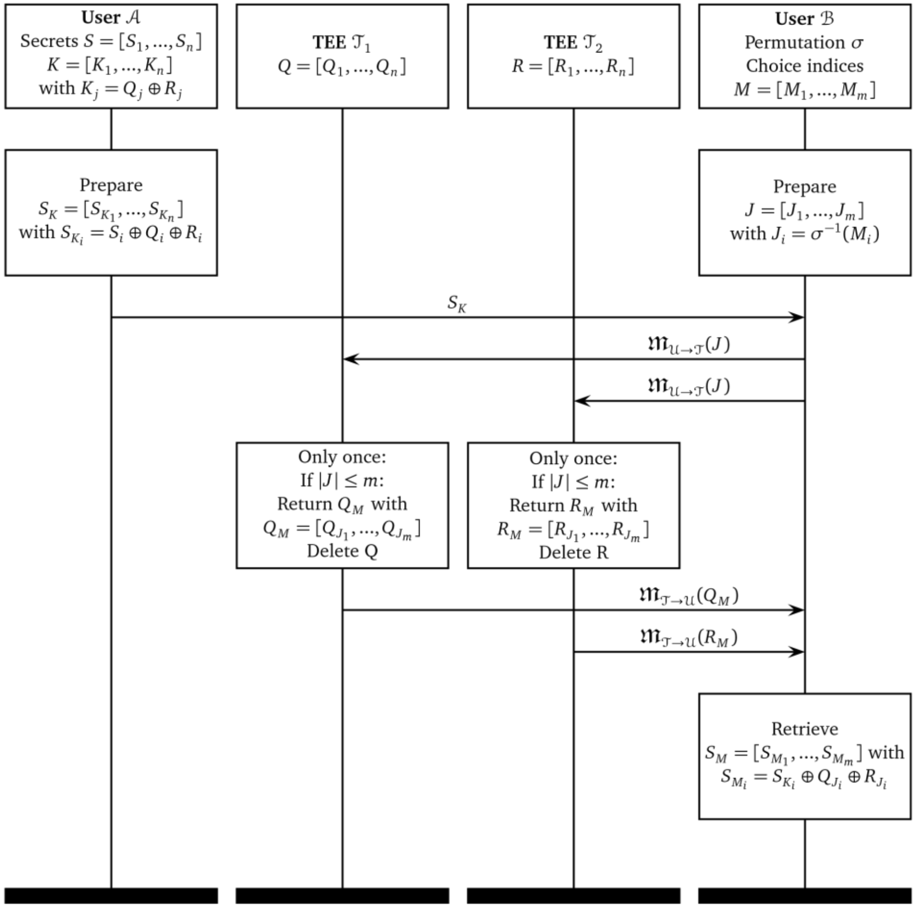
But: B should not see keys in plain!

→ Hide with homomorphic encryption

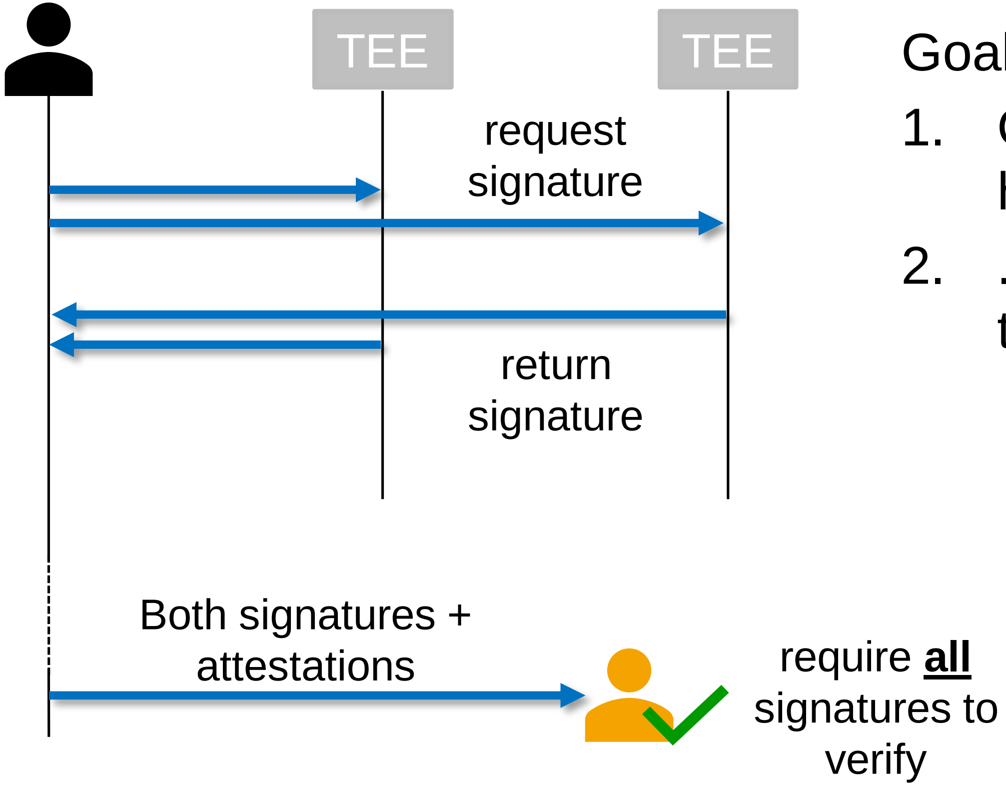
$\mathfrak{OT}^m_n$: Offline phase of oblivious m of n transfer between two users with Combined TEE.



$\mathfrak{OT}^m_n$: Online phase of oblivious m of n transfer between two users with Combined TEE.



Signing



Goal:

1. Operate on private signing keys held by the TEEs...
2. ...that are attestable and linkable to these TEEs