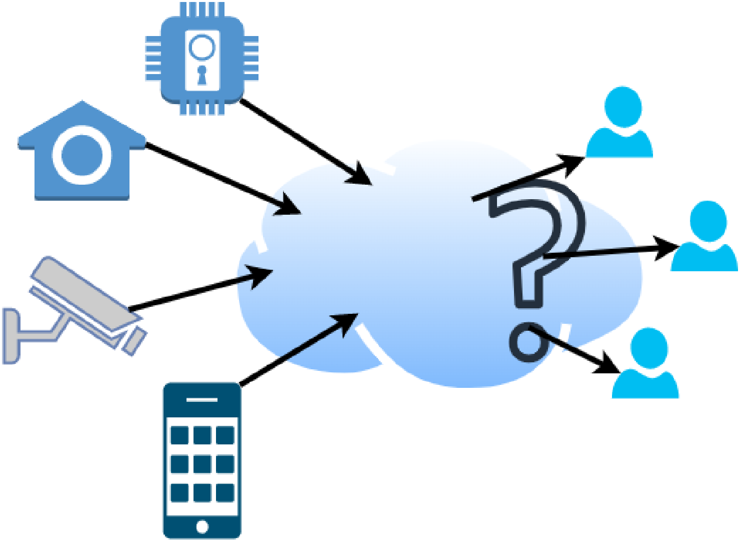
Trust and verify: Formally verified and attested computations in the cloud

Marcus Birgersson
Supervisors: Cyrille Artho & Musard Balliu

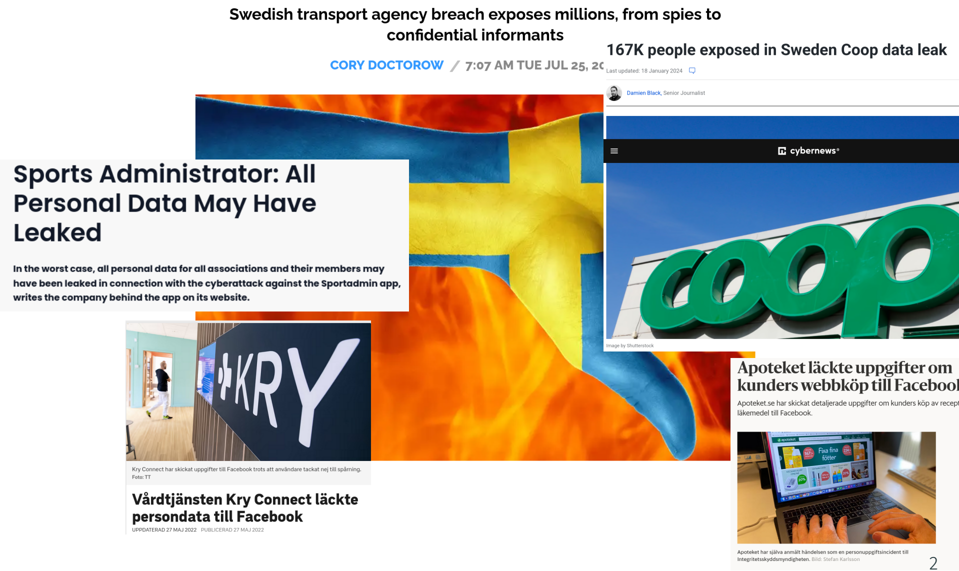
2025-05-22

KTH Royal Institute of Technology

Problem



Problem - Mistakes and breaches



Swedish transport agency breach exposes millions, from spies to confidential informants

CORY DOCTOROW / 7:07 AM TUE JUL 25, 20

167K people exposed in Sweden Coop data leak

Last updated: 18 January 2024

Damien Black, Senior Journalist

Image by Shutterstock

Sports Administrator: All Personal Data May Have Leaked

In the worst case, all personal data for all associations and their members may have been leaked in connection with the cyberattack against the Sportadmin app, writes the company behind the app on its website.

Kry Connect har skickat uppgifter till Facebook trots att användare tackat nej till spårning.
Foto: TT

Vårdtjänsten Kry Connect läckte persondata till Facebook

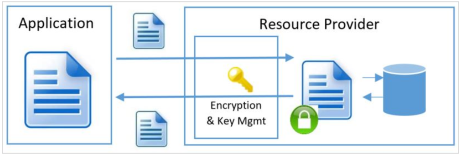
UPPDATERAD 27 MAJ 2022 PUBLICERAD 27 MAJ 2022

Apoteket läckte uppgifter om kunders webbköp till Faceboo

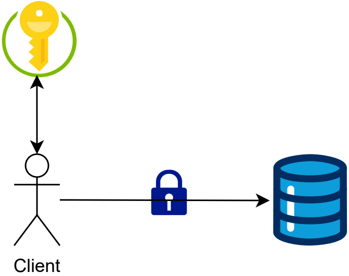
Apoteket.se har skickat detaljerade uppgifter om kunders köp av recept
läkemedel till Facebook.

Apoteket har själva anmält händelsen som en personuppgiftsincident till Integritetsskyddsmyndigheten. Bild: Stefan Karlsson

Examples - How to protect confidential data

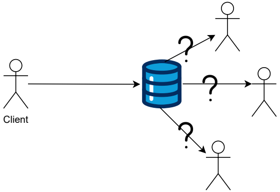


Server-side encryption

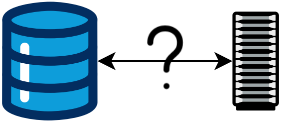


Client-side encryption

Examples - Sharing and computation over confidential data

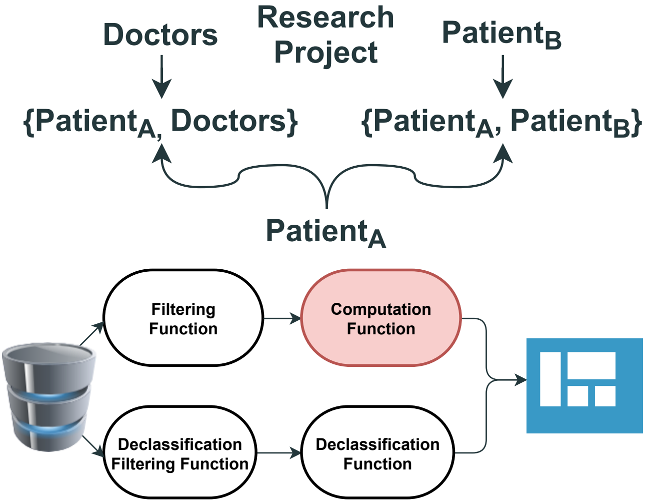


Sharing



Computation

Access Controll: Decentralized label model



Doctors

Research Project

Patient_B

{Patient_A, Doctors}

{Patient_A, Patient_B}

Patient_A

Filtering Function

Computation Function

Declassification Filtering Function

Declassification Function

Goal: User aware data protection

Attestation
The client should be able to get a proof for how the data is managed.

Goal: User aware data protection

Attestation
The client should be able to get a proof for how the data is managed.

Confidentiality

- The service provider should not be able to access data that it is not explicitly allowed to access.
- The service provider should not be able to use data in any other way that has been previously agreed upon.

Goal: User aware data protection

Attestation

The client should be able to get a proof for how the data is managed.

Confidentiality

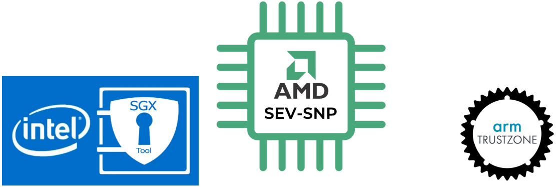
- The service provider should not be able to access data that it is not explicitly allowed to access.
- The service provider should not be able to use data in any other way that has been previously agreed upon.

Practical

- No limitations on computation
- Both data and computations should be carried out securely in the cloud.

Trusted Execution Environments

A trusted execution environment (TEE), is a tamper-resistant processing environment that guarantees the integrity and confidentiality of its run-time states.



Remote attestation:
Cryptographic proof of environment and software

Trusted Execution Environments - The good and the bad

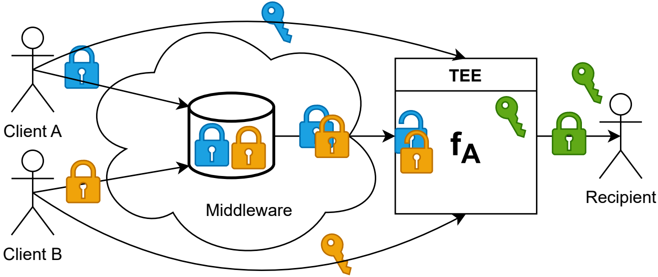
1. Cryptographic proof of software and hardware
2. Reproducible build generates transparent computation
3. Immutable

intel SGX Tool

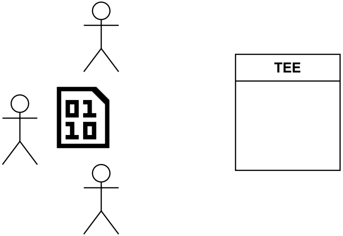
AMD SEV-SNP

arm TRUSTZONE

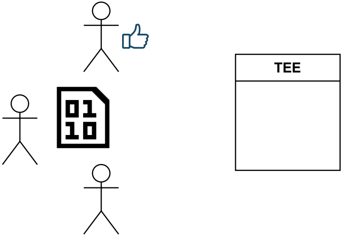
Secure multi-party computation in cloud



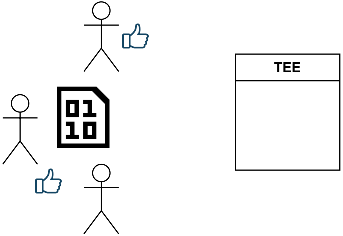
Problem: Slow and tedious updates



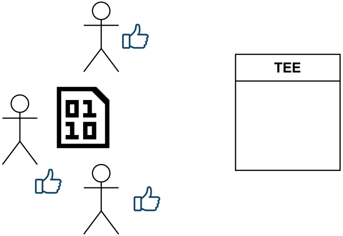
Problem: Slow and tedious updates



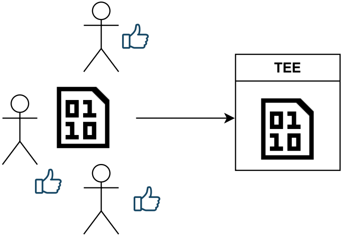
Problem: Slow and tedious updates



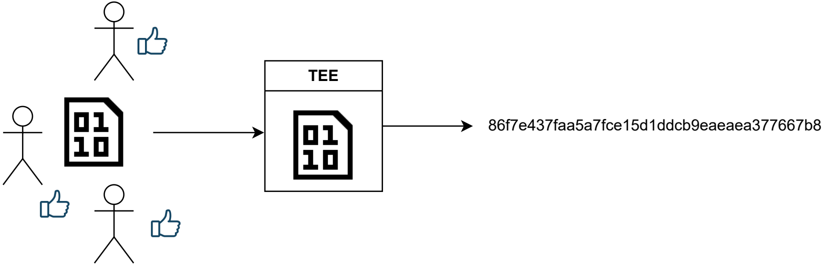
Problem: Slow and tedious updates



Problem: Slow and tedious updates



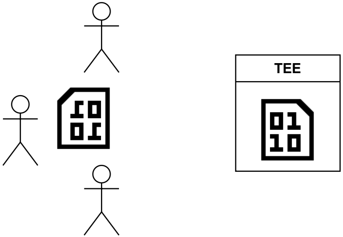
Problem: Slow and tedious updates



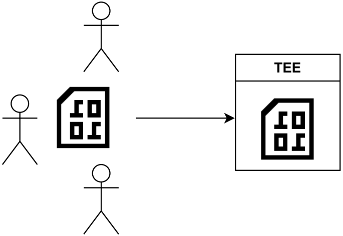
TEE

86f7e437faa5a7fce15d1ddcb9eaeaea377667b8

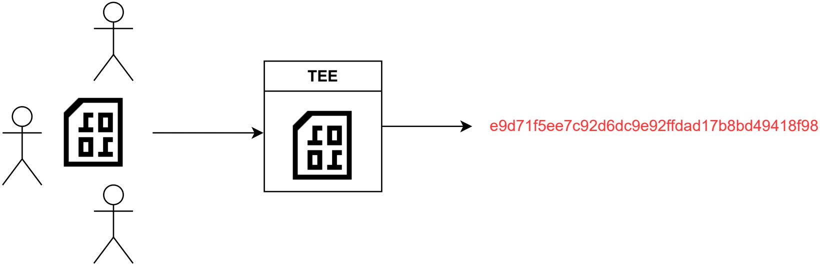
Problem: Slow and tedious updates



Problem: Slow and tedious updates



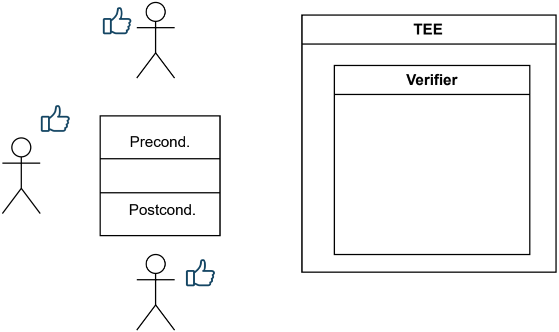
Problem: Slow and tedious updates



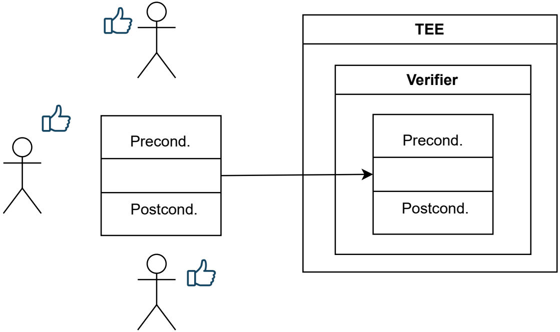
TEE

e9d71f5ee7c92d6dc9e92ffdad17b8bd49418f98

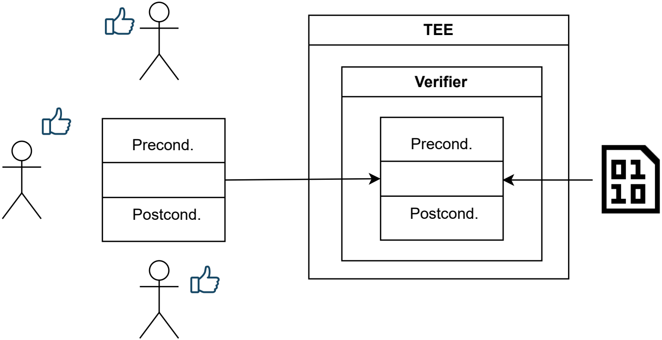
Trust the What, not the How



Trust the What, not the How



Trust the What, not the How



Example - Upgradable sorting implementation

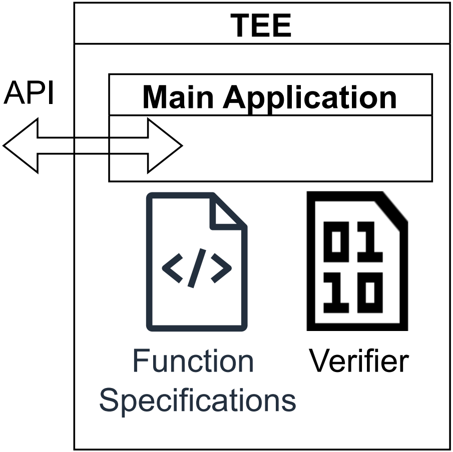
```
 predicate perm(a:array<int>, b:array<int>)
    requires a != null && b != null
    reads a,b
{
    multiset(a[..]) == multiset(b[..])
}


 predicate sorted(a:array<int>, min:int, max:int)
    requires a != null
    requires 0 <= min <= max <= a.Length
    reads a
{
    forall i,j | min <= i < j < max :: a[i] <= a[j]
}
```

Example - Upgradable sorting implementation

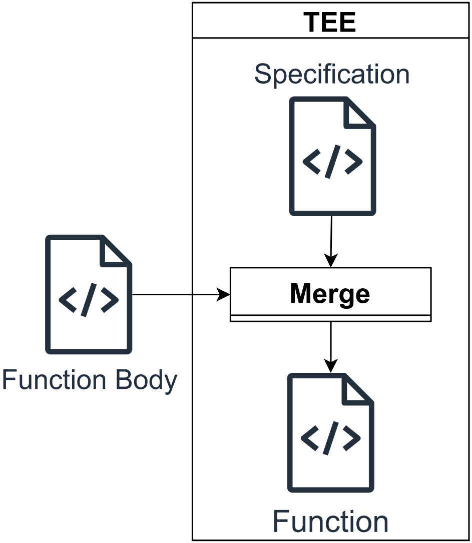
```
 method sort(a:array<int>)
    requires a != null
    requires a.Length >= 1
    modifies a
    ensures perm(a,old(a))
    ensures sorted(a, 0, a.Length)

{
// Implementation goes here
}
```

TRUVALT framework

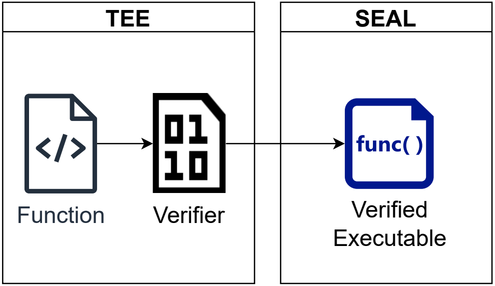
TRUVALT - TRUsted VAlidation system in TEE



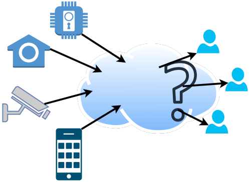
Deployment

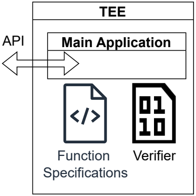


Deployment



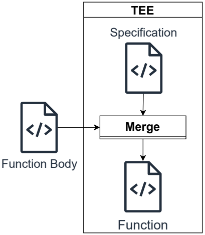
Questions

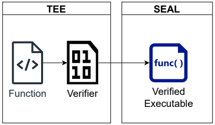




Thanks!

Questions?


TEE
Specification
Function Body
Merge
Function


TEE
SEAL
Function
Verifier
func( )
Verified Executable