Maintaining the Anonymity of Direct Anonymous Attestation with Subverted Platforms

MIT PRIMES Computer Science Conference
October 13, 2018

By: Ethan Mendes and Patrick Zhang
Mentor: Kyle Hogan

What is an attestation?

A message to state a fact.

You are in a group and want to prove you belong in the group without telling people your name.

Analogy: You, a student, want to buy supplies at the store but want the student discount. You must prove you are a student.

Why does anonymity matter?

Anonymity: the inability to prove that a party was the sender of a message

Your host now belongs to a group that can create attestations with other hosts, but you don’t want to give them your identity.

Extended Analogy: The store knows you are a student, but cannot know your name or any other information besides the fact that you are a student.

Analogy

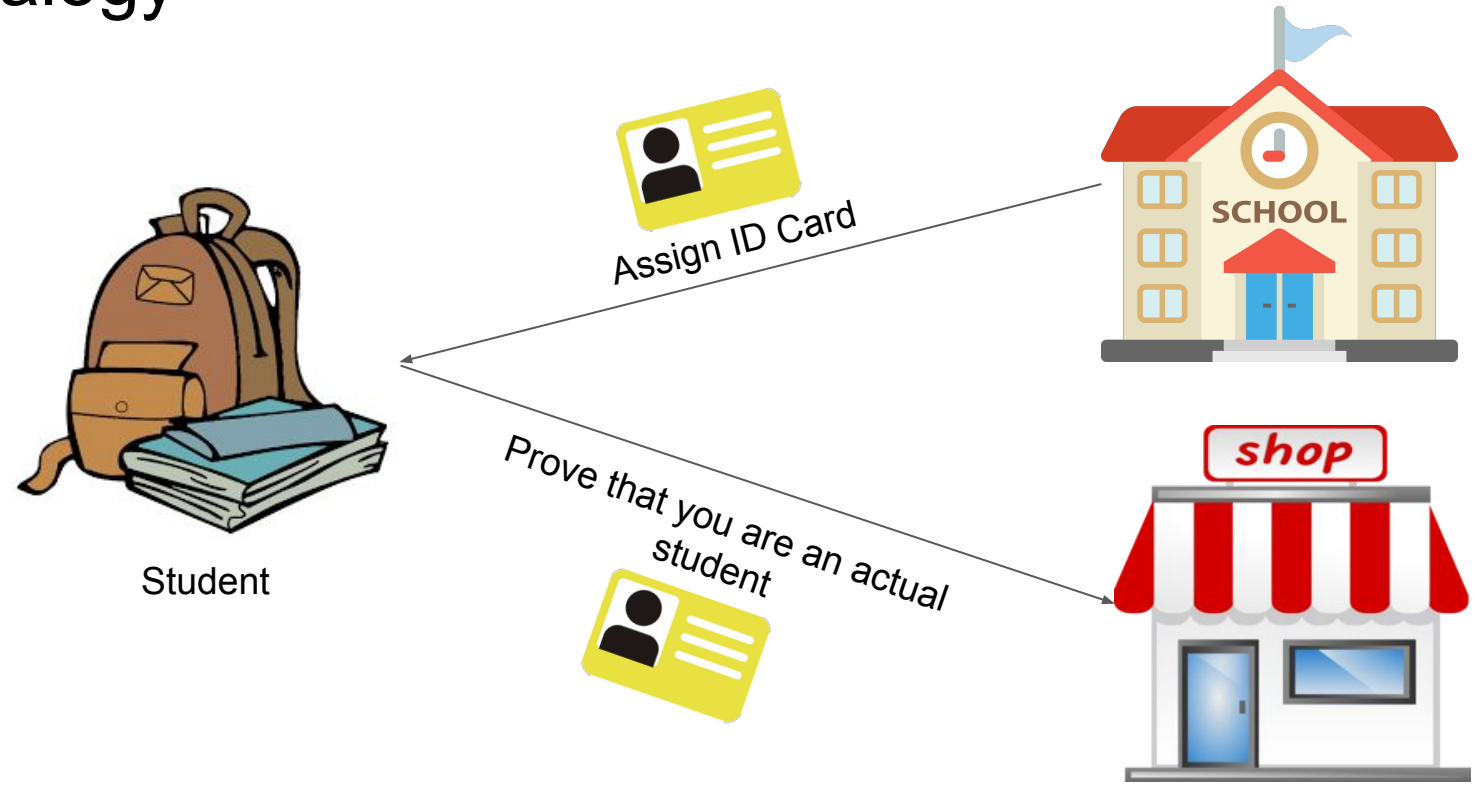
Student

Analogy

Student

Assign ID Card

Analogy


Assign ID Card
SCHOOL
Prove that you are an actual student
shop
Student

Analogy

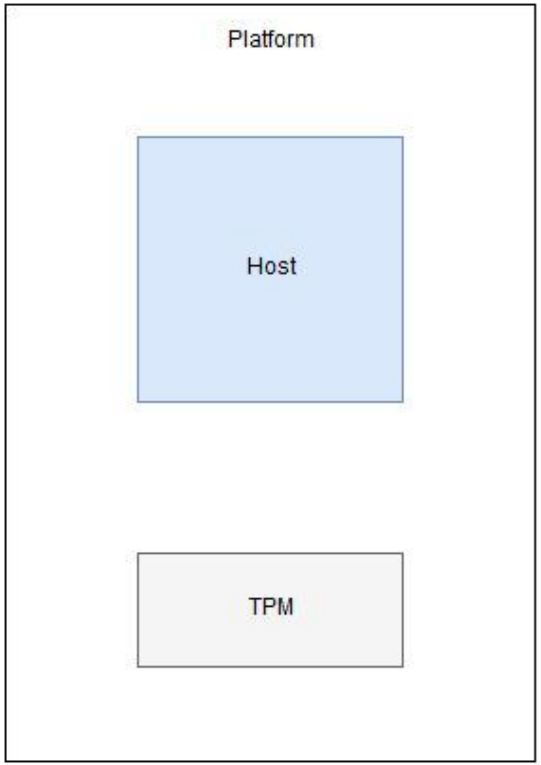
Student

Assign ID Card

Prove that you are an actual student

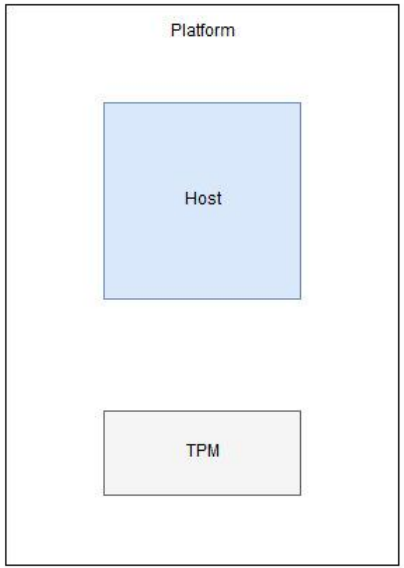
This process is linkable and NOT anonymous!!!

Basic Setup



What is the Platform?

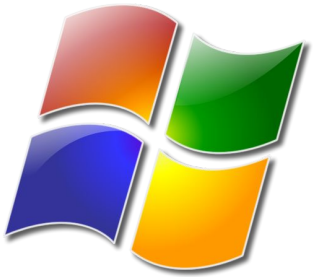
- Contains both the host and the TPM
- Wishes to stay anonymous
  - Analogy: “The student”



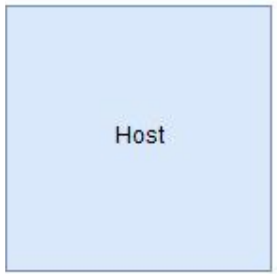
=

What is the Host?

- The host is generally seen as the operating system
- It has full control over what leaves and enters the platform



=



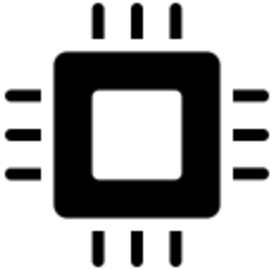
What is a TPM?

Trusted Platform Module:

- a microcontroller with limited functionality for hardware-based, integrated cryptographic keys
- tamper resistant due to physical barriers of altering extracting keys

How is it used?

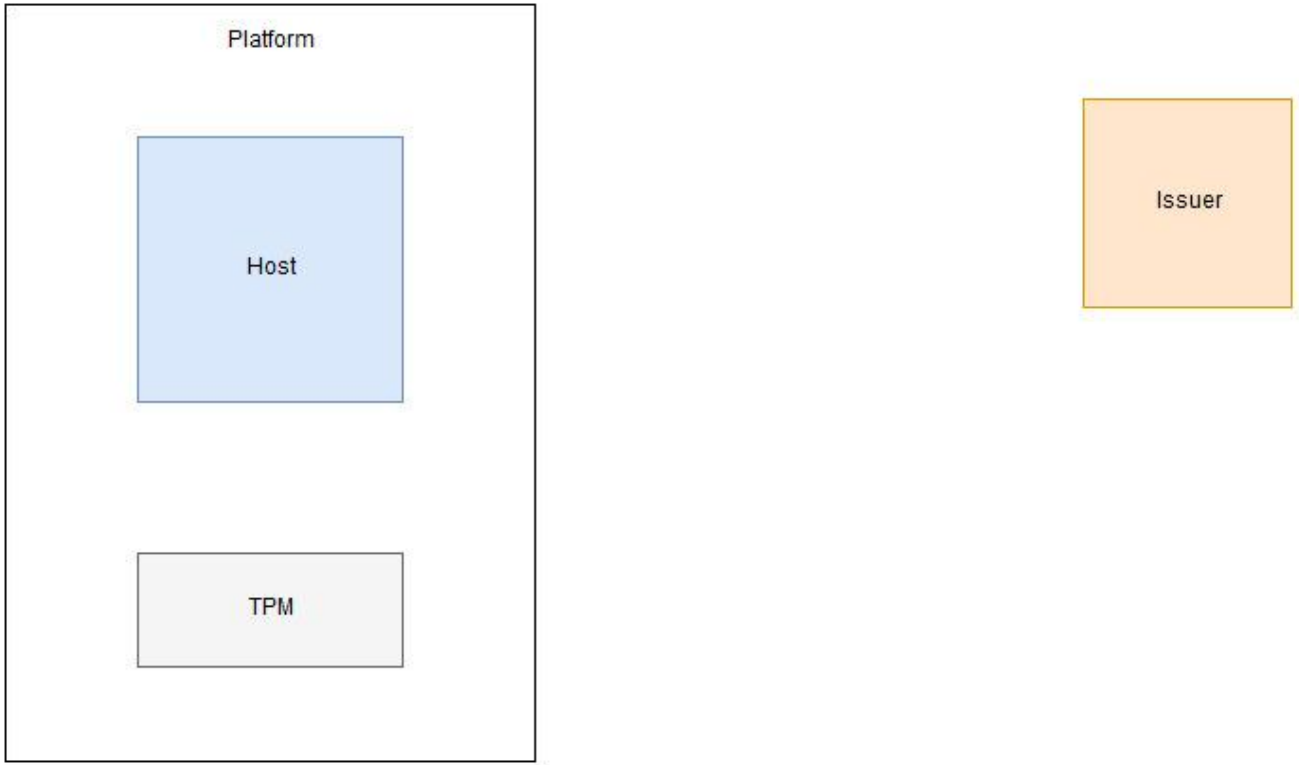
- Creates attestations of the state of the host machine



=

TPM

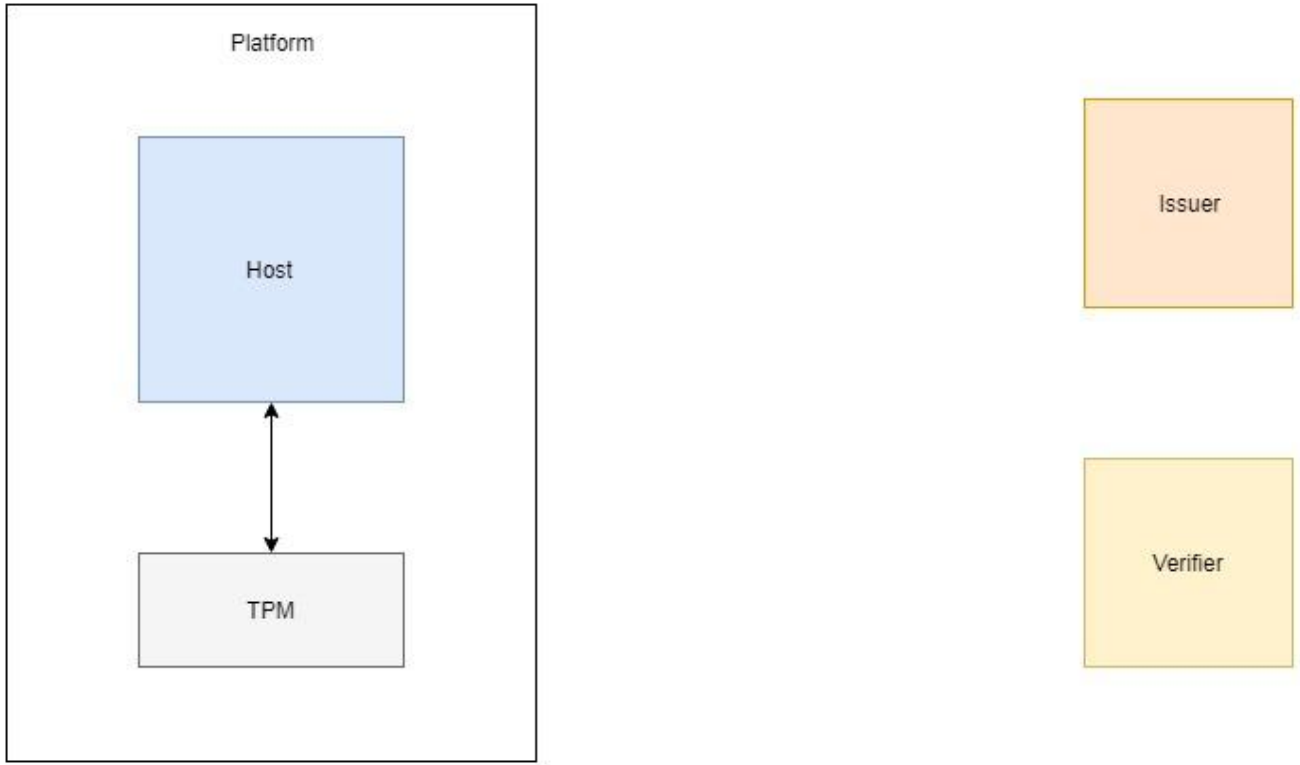
Basic Setup



What is the Issuer?

- The owner of the group
  - Analogy: “The school”
- Can’t be trusted with your personal information for security reasons
- Can be trusted to verify you are a member of the group
  - Analogy: “You trust them that they believe you are a student”
  - If this were not true, you wouldn’t want to be in the group

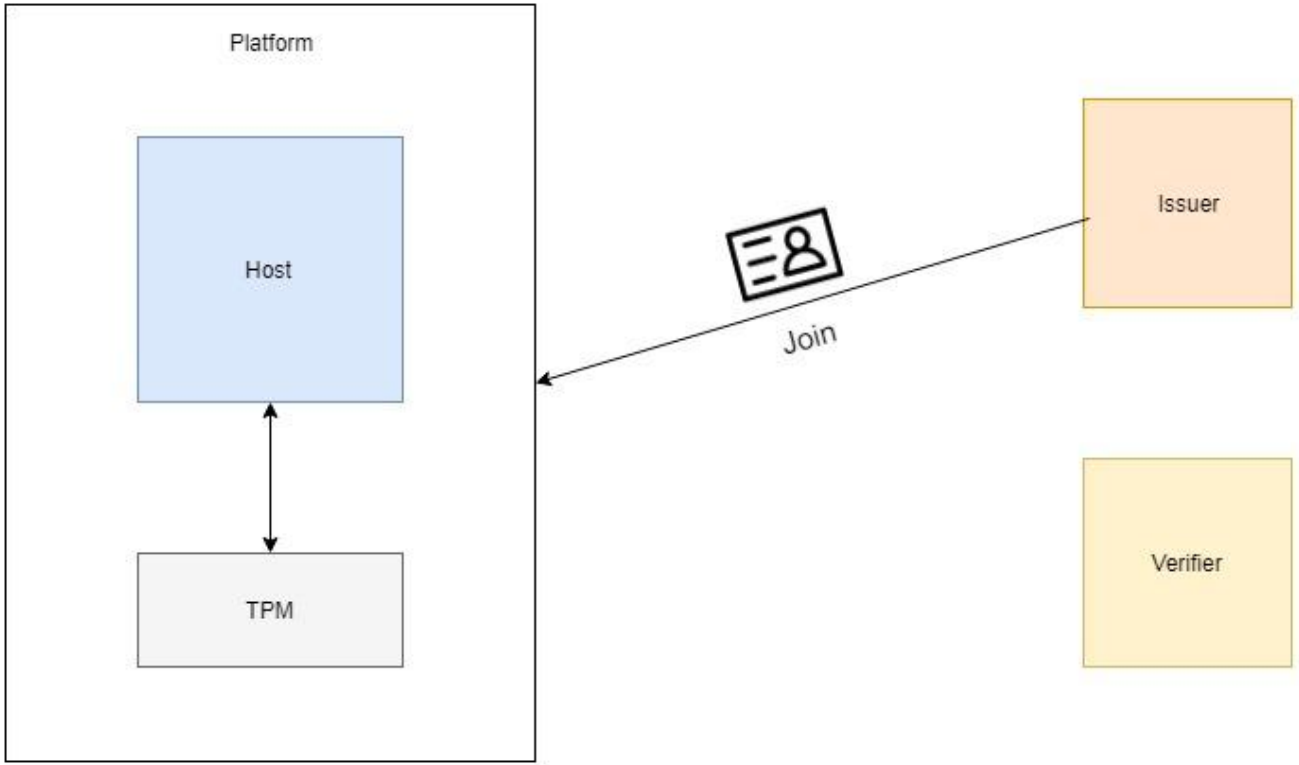
Basic Setup



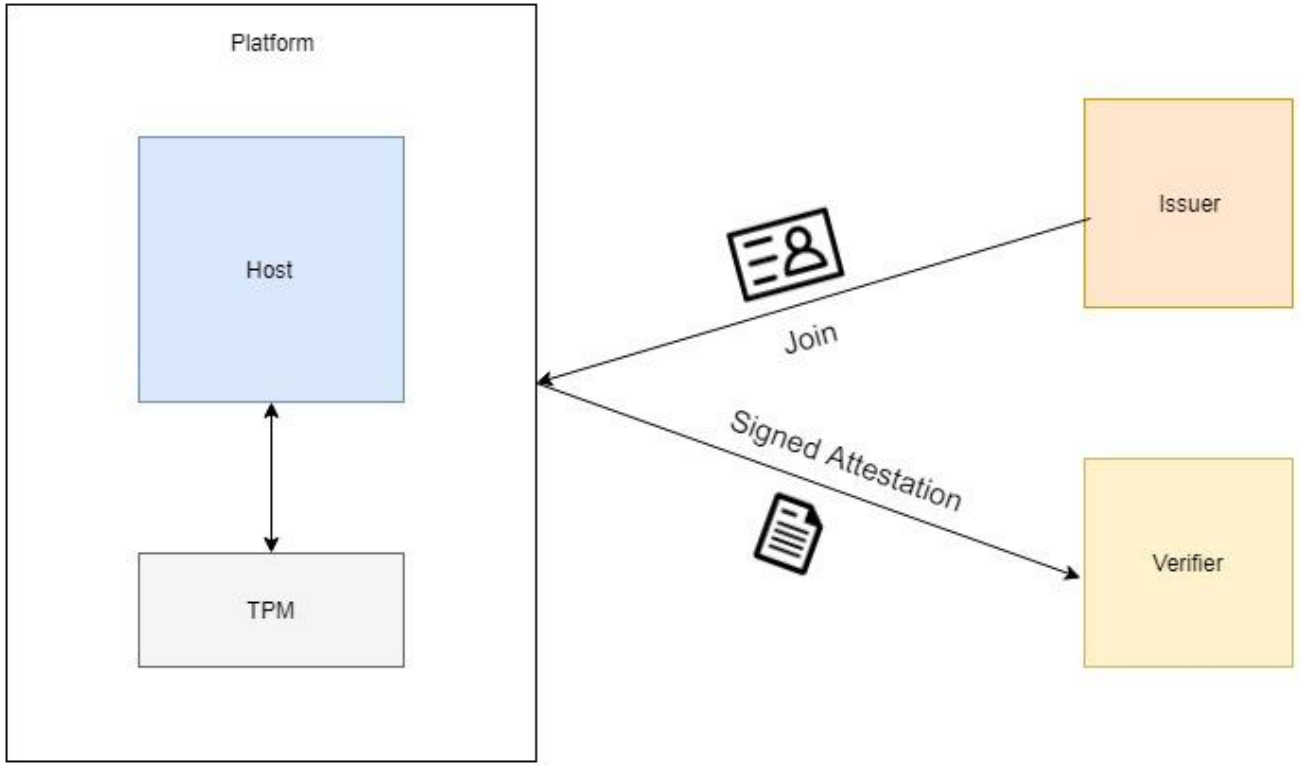
What is the Verifier?

- The device to which you want to prove to that you are in the group
  - Analogy: “the store”
- This is the party that receives the attestation from the platform
  - They make the transaction knowing you are a student without knowing which student
- You trust them enough to make a connection with them, but not enough to release your identity
  - Analogy: the student trusts the store to give them the supplies at the discounted price but can’t trust them with their identity

Approach #1: Remote Attestation



Approach #1: Remote Attestation



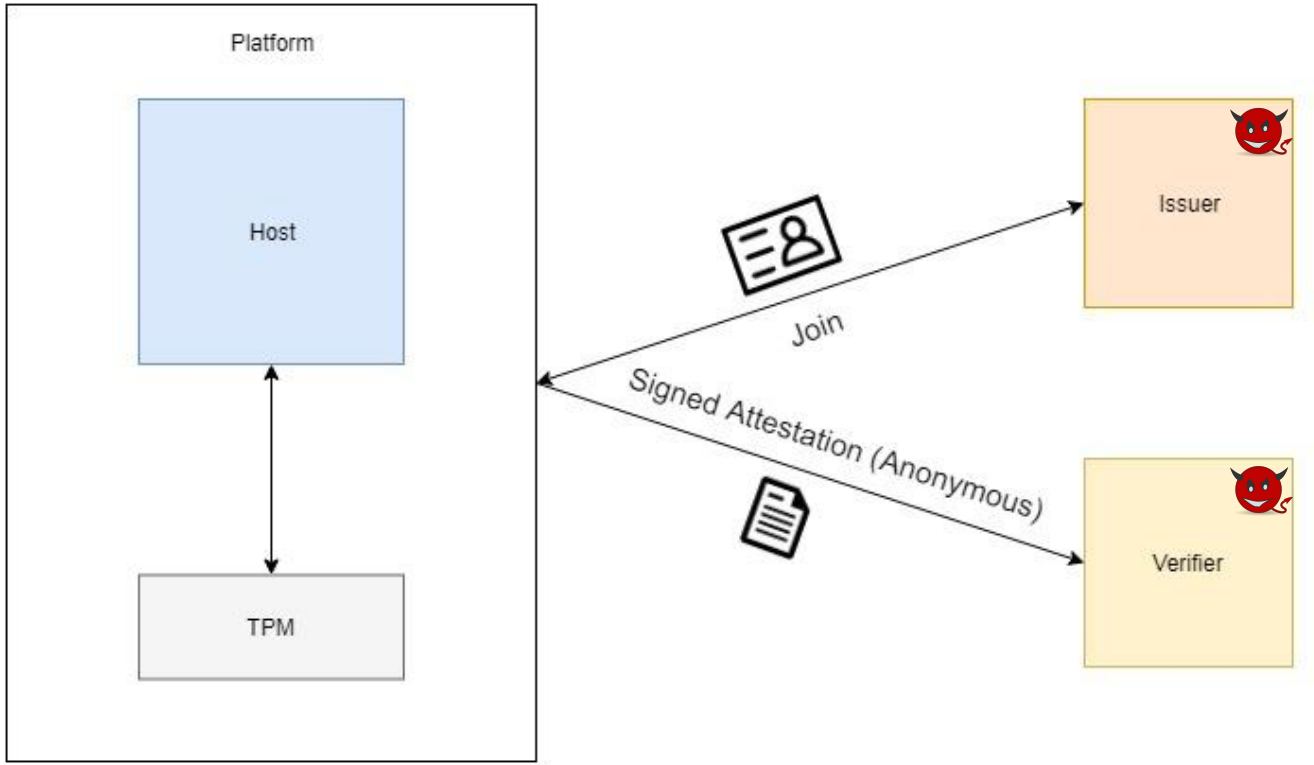
Shortcomings

- This process is not anonymous - the signature of the TPM gives away the platform’s identity
- The TPM is **linkable** - the verifier is able to link the signatures from a single TPM

- Why is this undesirable? Digital Rights Management

We are motivated to send an attestation which does not reveal anything about the identity of the platform and also does not allow for linkability

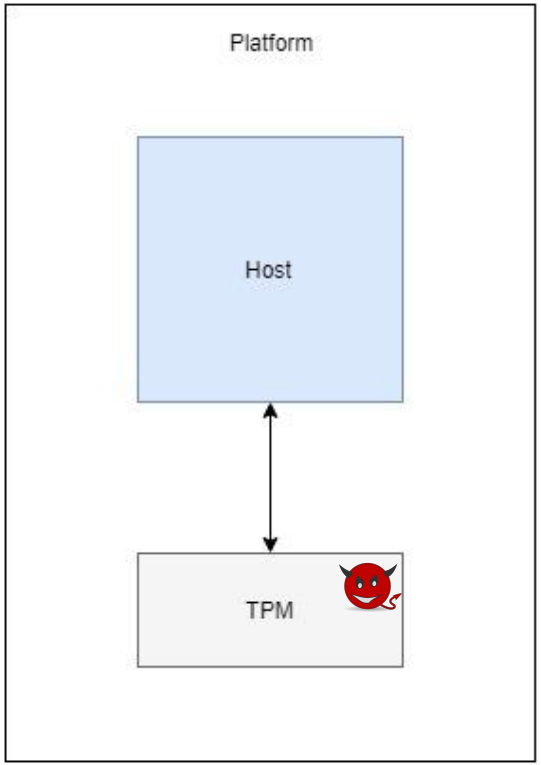
Approach #2: Direct Anonymous Attestation



Shortcomings

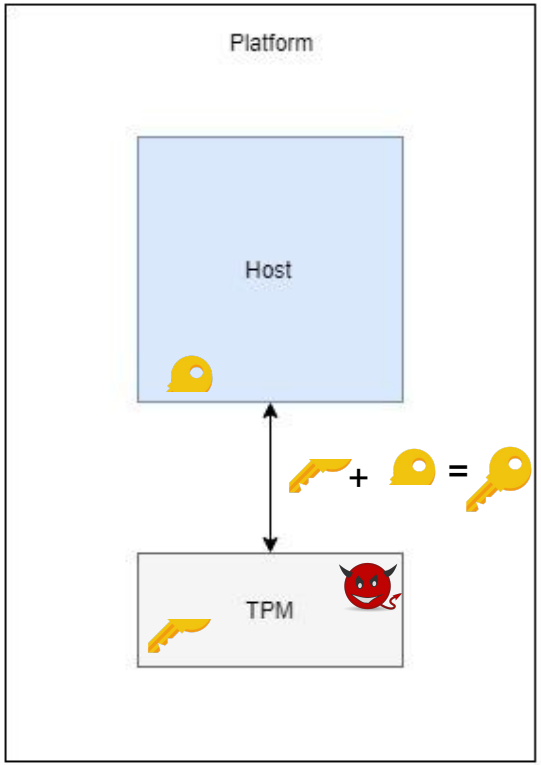
- Platform is completely trusted (host, TPM, every part of the platform)
- Does not stop operating system from attaching identification information on outgoing packets
- TPM has the keys to sign anything, even a deanonymizing message

Approach #3: Anonymous Attestation with a Subverted TPM[1]



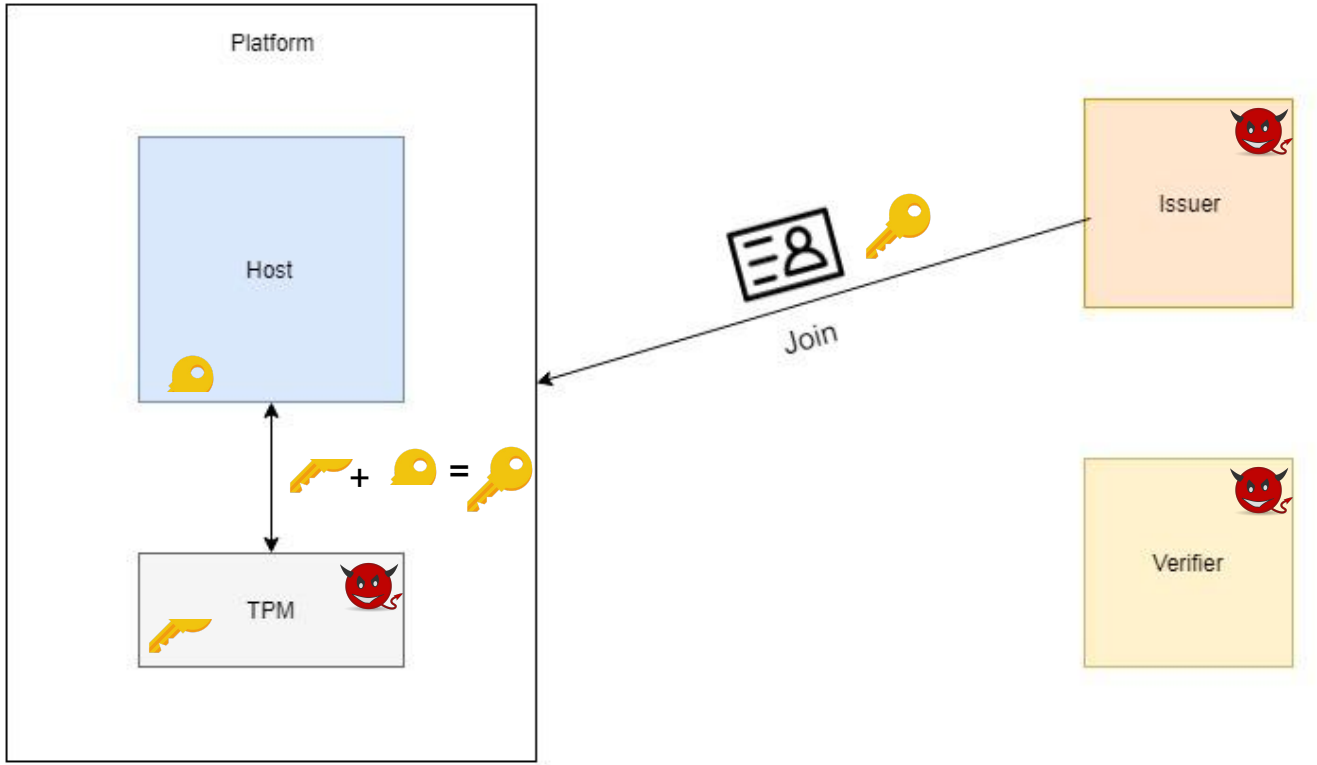
[1]Camenisch, Drijvers, Lehmann: "Anonymous Attestation with Subverted TPMs"

Approach #3: Anonymous Attestation with a Subverted TPM[1]



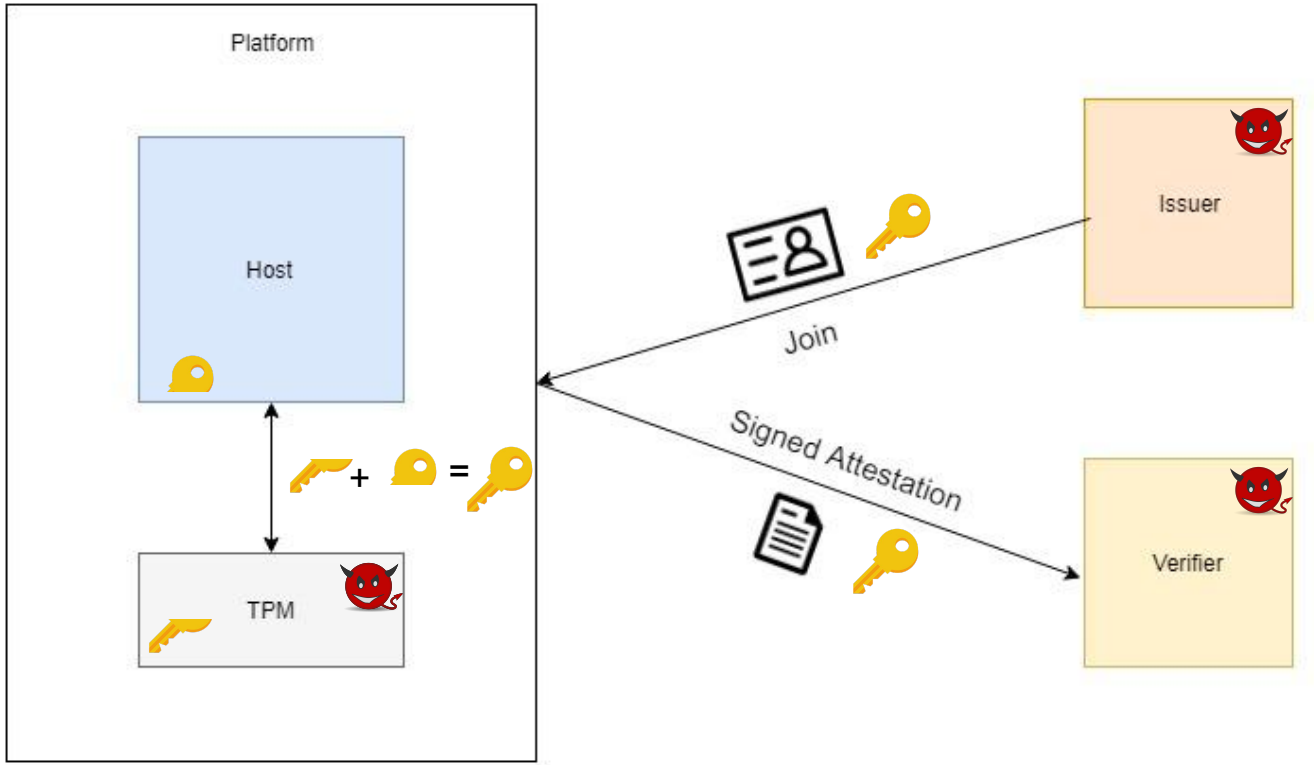
[1]Camenisch, Drijvers, Lehmann: "Anonymous Attestation with Subverted TPMs"

Approach #3: Anonymous Attestation with a Subverted TPM[1]



[1]Camenisch, Drijvers, Lehmann: "Anonymous Attestation with Subverted TPMs"

Approach #3: Anonymous Attestation with a Subverted TPM[1]



[1]Camenisch, Drijvers, Lehmann: "Anonymous Attestation with Subverted TPMs"

Shortcomings

- This process assumes that the host (OS) is fully honest
  - Host can still append any deanonymizing information within the protocol
    - Because the information is inside the attestation, parties can trust its validity

Our Approach

- An adversarial host is plausible in the real world
- We use Intel Software Guard Extensions (SGX) and The Onion Router (TOR) to account for a corrupted host
- For the most part we keep the same cryptographic protocol as Approach #3, but the role of the host is given to an enclave secured by SGX

Intel Software Guard Extensions (SGX)

- Allows the creation of secure **enclaves** - isolated region of encrypted memory that can’t be accessed by other software (even the OS)
- Summary of the contents of the enclave is taken and verified by a trusted party to ensure it is running the correct code
    - verification of small amount of code within the enclave is much easier than that of an operating system

Motivation

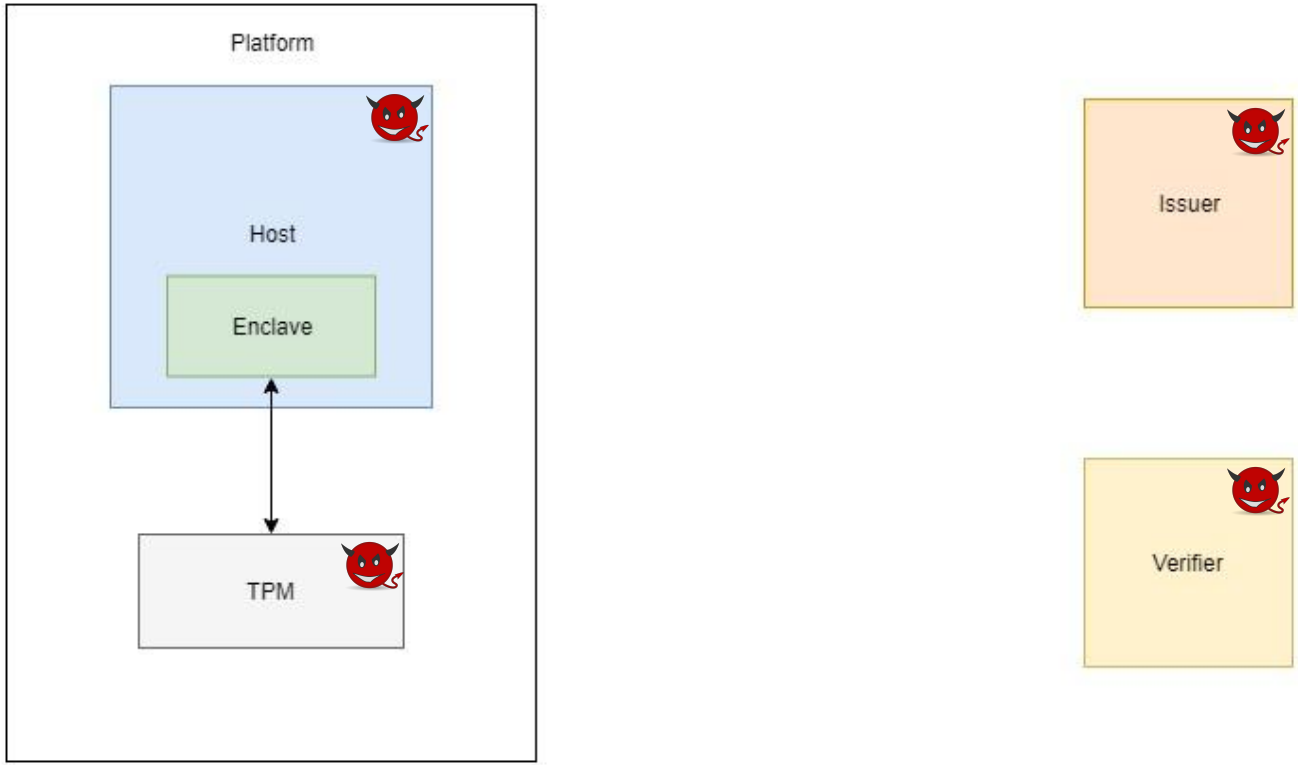
- Hosts (operating systems) are generally very complex and have millions of lines of code
- For every 1000 lines of code approximately 15-50 logic errors can be expected [1]
- It is possible that the operating system has vulnerabilities which can be exploited by the adversary

Operating System	Windows 7	Linux 3.1	Mac OS X
Lines of code (LOC) [2]	~40 million	~15 million	~86 million
Approximate Number of Errors	~600,000 - 2 Million	~225,000 - 750,000	~1.29 Million - 4.3 Million

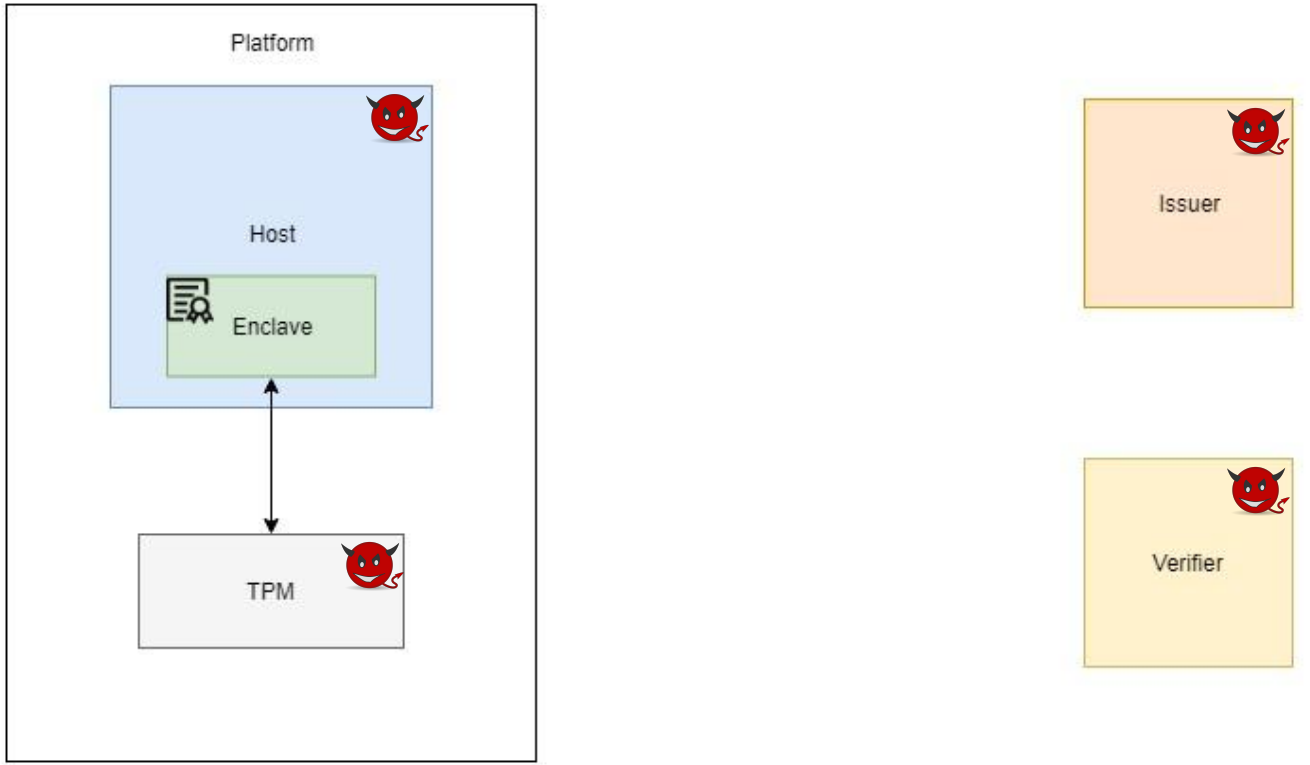
[1] Code Complete, Steve McConnell

[2] Data (with links) : https://docs.google.com/spreadsheets/d/1s9u0uprmuJvwR2fkRqxJ4W5Wfomimmk9pwGTK4Dn_UI/edit#gid=5

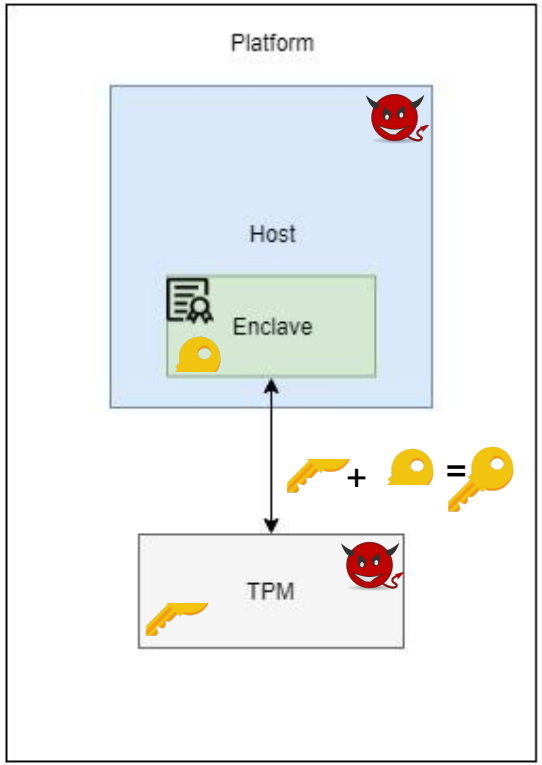
New Model



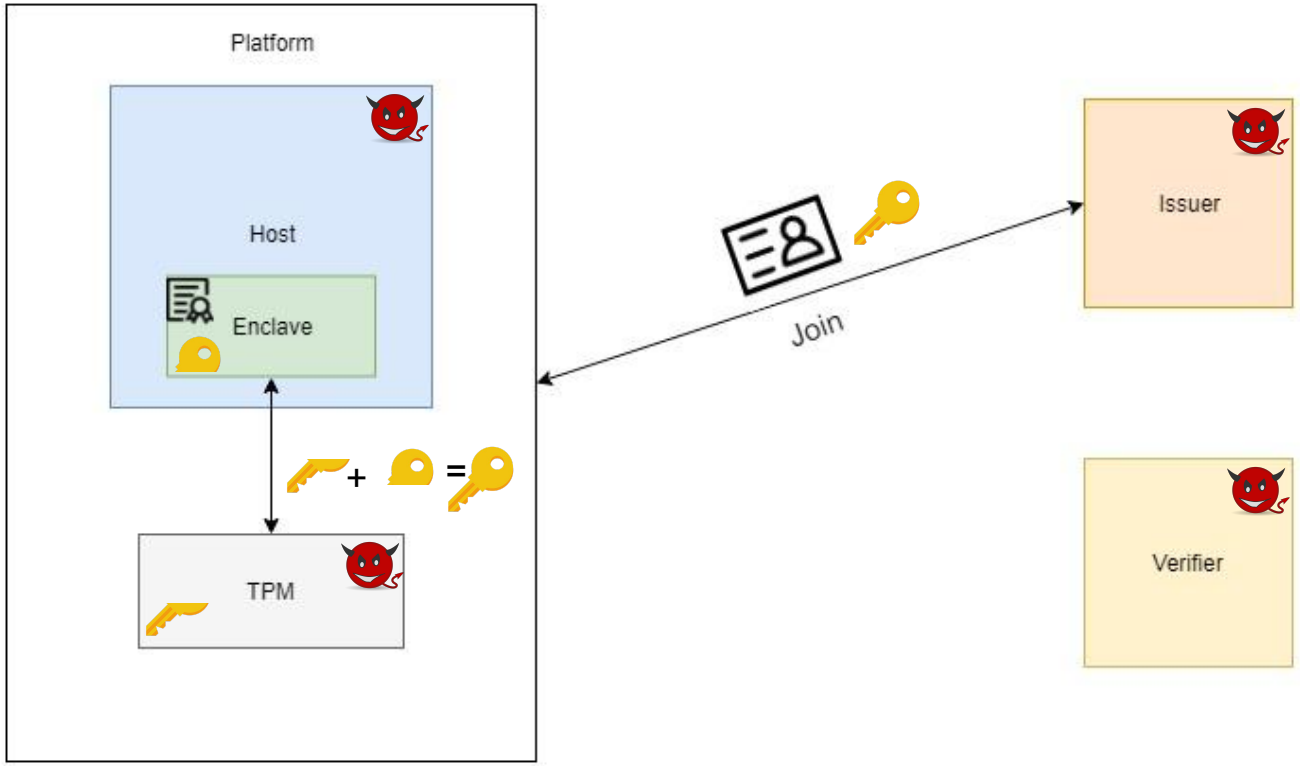
New Model



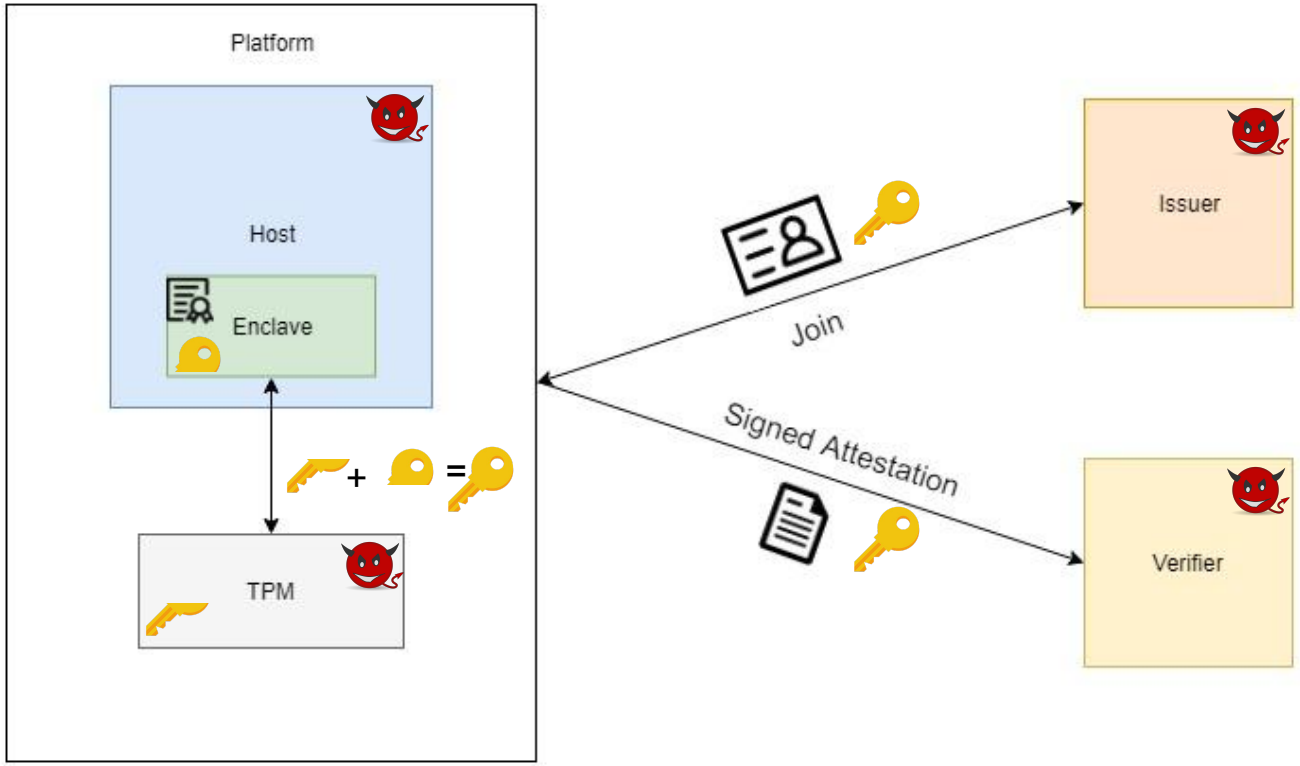
New Model



New Model



New Model

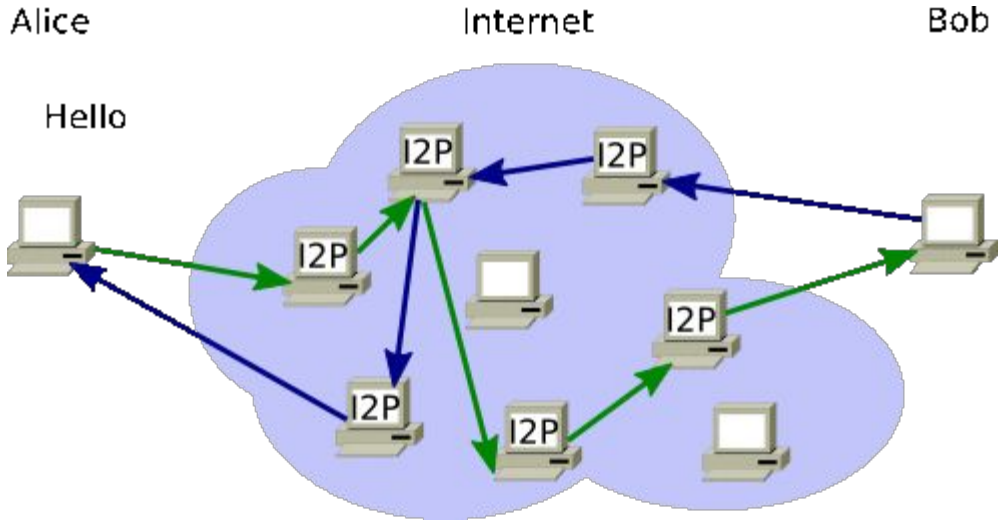

Platform
Host
Enclave
TPM
Issuer
Verifier
Join
Signed Attestation

Shortcomings

- The operating system still has full control over all packets being sent out of the platform
- It can append identifying information onto outgoing messages

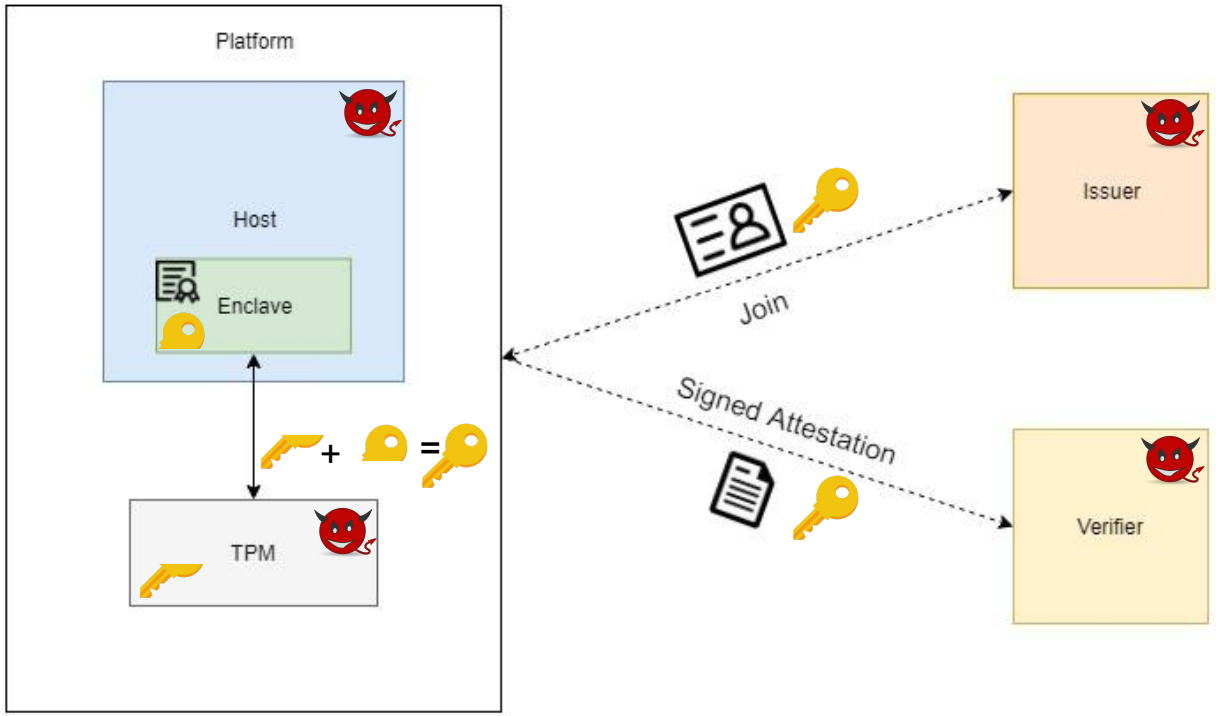
The Onion Router: TOR

Proposal: the protocol will use TOR to encrypt the message several times. If the OS appends identification information anytime after the TOR encryption, the decryption performed at TOR nodes will remove all such information



https://trac.i2p2.de/wiki/Content

New Model


Platform
Host
Enclave
TPM
Issuer
Verifier
Join
Signed Attestation
= Encrypted with TOR

Why security is now achieved

- This protocol keeps the platform’s identity hidden (upholds anonymity) even if the host is corrupted:
  - The enclave secured by SGX prevents the host from adding identifying information to the attestation before it is encrypted
  - TOR removes identifying information that the host may have appended onto the outgoing attestation

Conclusions

- Existing anonymous attestation protocols do not provide a strong sense of anonymity within a more realistic threat model
- The use of an SGX enclave and TOR can be used to strengthen the notion of anonymity for a more adversarial threat model

Future Work

- Develop a concrete instantiation for the protocol in order to generate runtime data

Acknowledgements

- We thank MIT PRIMES for this incredible opportunity

- Our Mentor: Kyle Hogan

- Our Parents

Questions?