Runtime transparency (?)

or

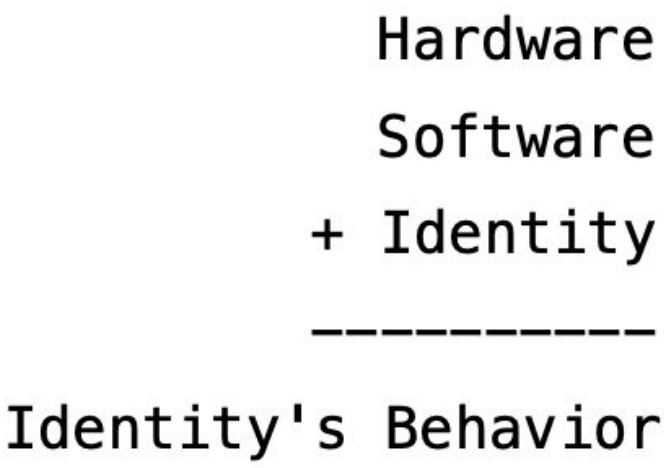
- ~ “transparent server” (Hal Finney ca 2001)
- ~ Keylime (MIT ca 2016, Redhat, IBM)
- ~ System Transparency (Mullvad 2019)
- ~ Project Oak (Google 2019)
- ~ Apple Private Cloud Compute (Apple 2024)
    - **“verifiable transparency”, “runtime transparency”**
- … what else?

Organizations & Core Activities

Organization	Core activities
Amagicom AB	Holding company
– Mullvad VPN AB	– Mullvad VPN – Mullvad Browser – Mullvad Leta (search engine)
– Karlstad Internet Privacy Lab AB	– Traffic analysis defense against AI-based classifiers
– Glasklar Teknik AB	– System Transparency – Sigsum – Debian Snapshot service
– Tillitis AB	– FPGA-based open-source security hardware – TKey (a USB security key)

System Transparency – (system-transparency.org)

– a security architecture for transparent systems



What must I ~~trust~~ be vulnerable to?

- the **hardware** and the **operator** who provisioned it
- the **software** and the **developer** who provided the source code

System Transparency Authentication Mechanism

Fredrik Strömberg

STAM provides accountability for running systems

TLS
(and other secure communication protocols)

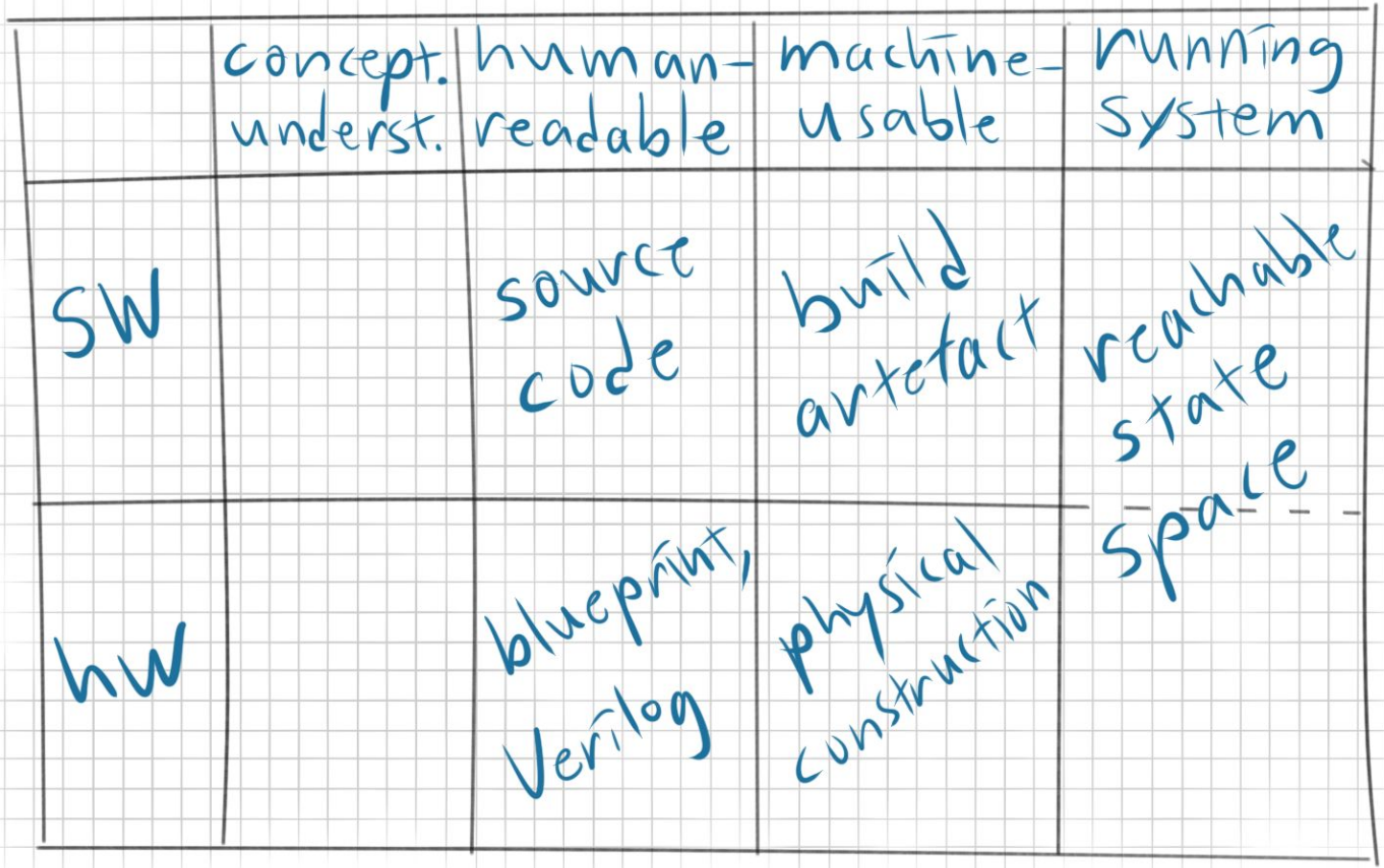
- Authentication is done by establishing trust in a certificate.
- The certificate contains an identity and a public key.
- The authenticated key is used to establish secure communication.
- Secure communication can be established assuming the remote system’s private key has not been compromised.

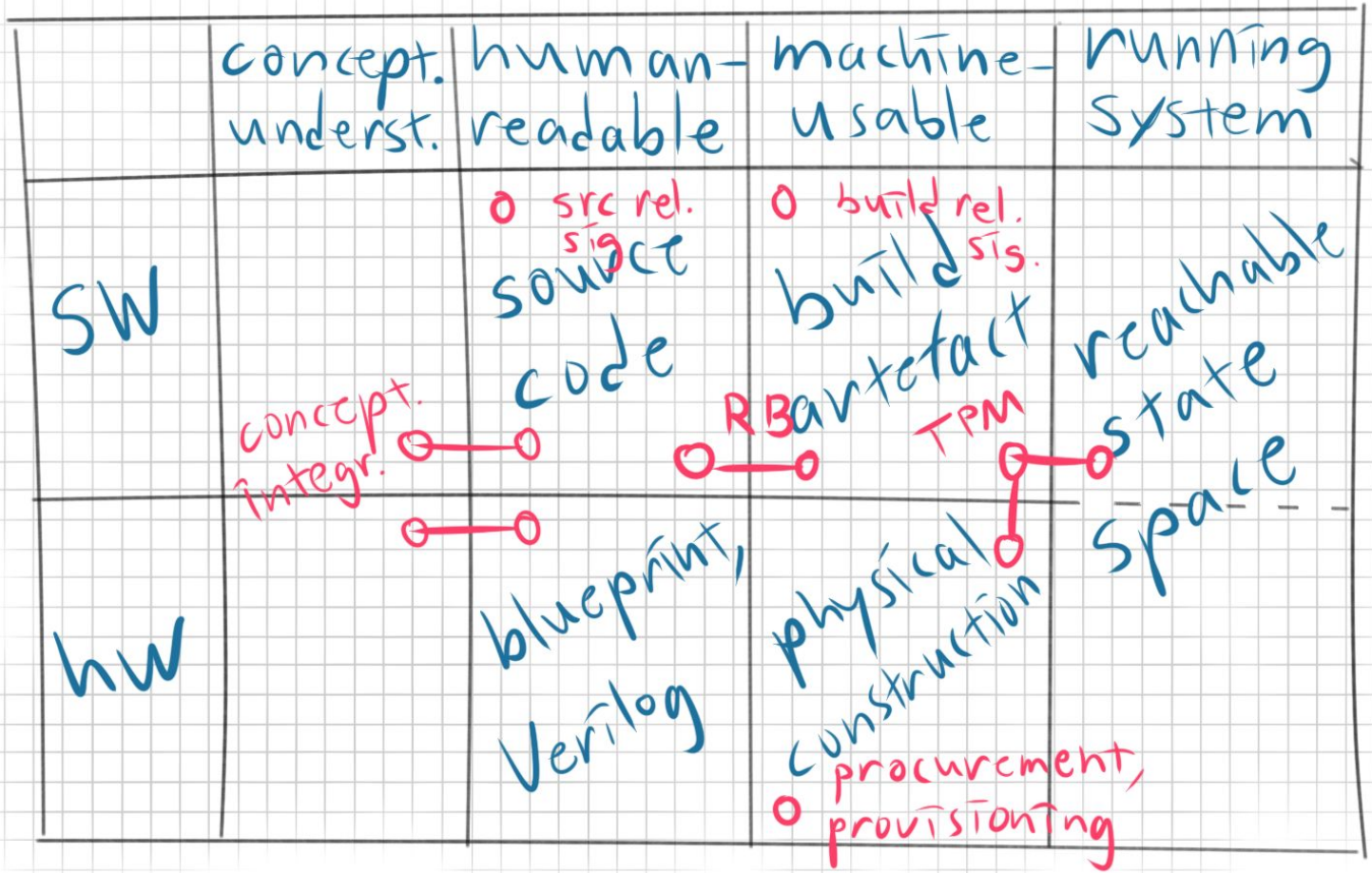
STAM

(and remote attestation mechanisms)

- Authentication is done by establishing trust in a certificate.
- The certificate contains an identity, a public key, **and other claims about the remote system's attributes.**
- The authenticated key is used to establish secure communication.
- Secure communication can be established assuming the remote system's private key has not been compromised.


hw + sw
behavior


concept. understand.
human-readable
machine-usable
running system
SW
source code
build artefact
reachable state space
hw
blueprint, Verilog
physical construction


concept. understand.
human-readable
machine-usable
running system
SW
hw
source code
build artefact
reachable state space
blueprint, Verilog
physical construction
concept. integr.
RB
TPM
src rel. sig.
build sig.
procurement, provisioning

System Transparency – (system-transparency.org)

Builds on:

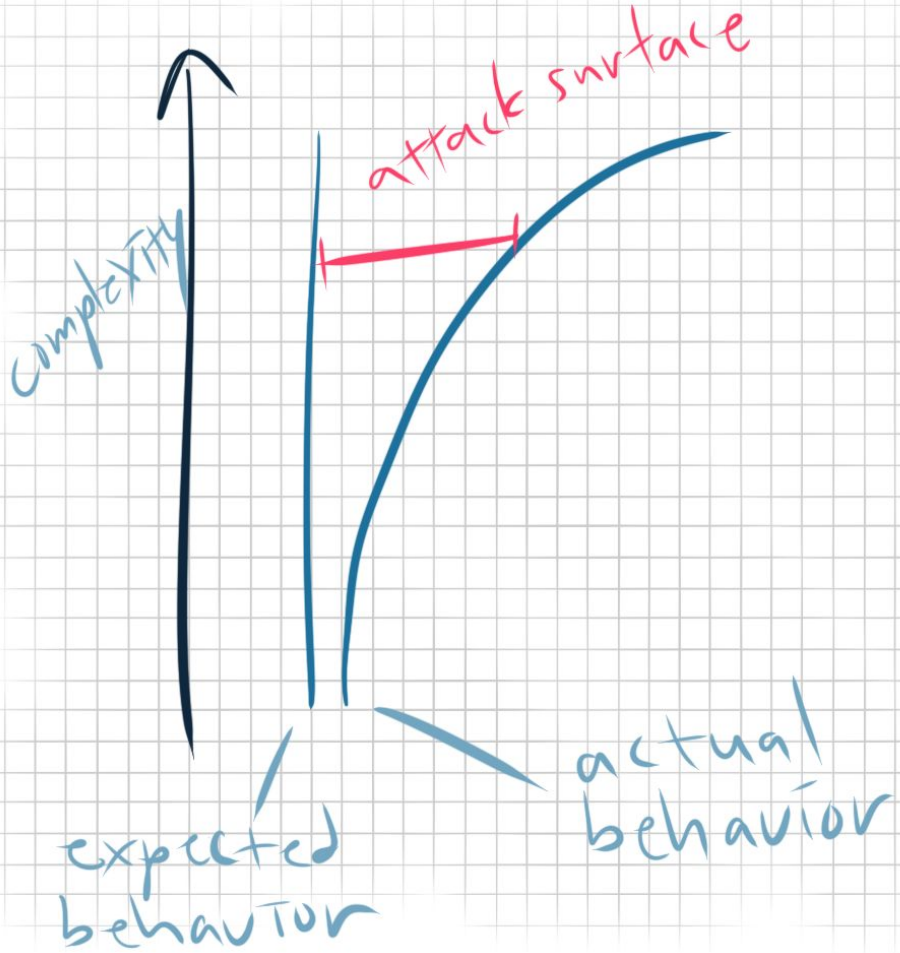
- secure network communication protocols with entity authentication mechanisms
- remote attestation
- reproducible builds
- Sigsum (transparency logging with witness cosigning)

Complexities that matter

Computational complexity is the foundation of modern cryptography.

What would the Internet look like without TLS?

Conceptual complexity is the root cause of most(?) security vulnerabilities.


attack surface
complexity
expected behavior
actual behavior

High-leverage defences

(increase confidence that intended and actual system behaviour are the same)

1. **Maximise use of cryptographic defences**

   (as cryptography relies on computational complexity)

2. **Maximise use of hardware defences**

   (as hardware defines the rules of software)

3. **Maximise conceptual integrity**

   (see quotes)

4. **Constrain the reachable state space whenever possible**

5. **Distribute trust assumptions**

END

STAM

will assure a local system of a remote system's

- platform provenance (signed provisioning manifest)
- platform state (signed TPM measurements)
- software authenticity (signed code)
- source code traceability (signed reproducible builds)
- attestation freshness (measure a recent Sigsum STH)
- certificate transparency (Sigsum log all signatures)
- human-readable identity (e.g. X.509 TLS Certificate)

STAM

signature details

• DC engineer signs	(signed provisioning manifest)
• TPM signs	(signed TPM measurements)
• Developer signs	(signed code)
• Build chain signs	(signed reproducible builds)
• Log, witnesses & TPM sign	(measure a recent Sigsum STH)
• Log & witnesses sign	(Sigsum log all signatures)
• Web CA signs	(e.g. X.509 TLS Certificate)

STAM

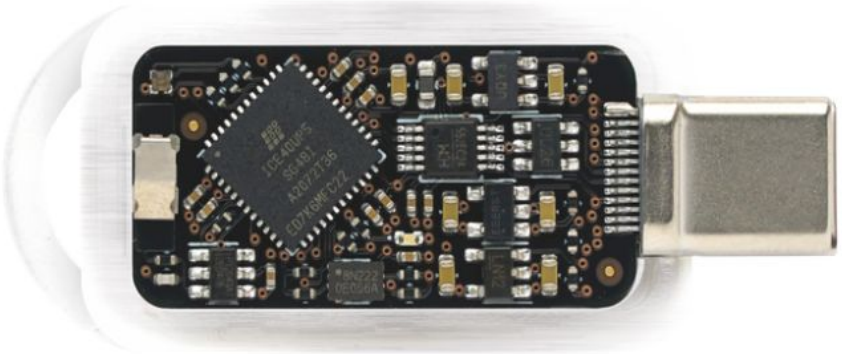
signature details, cont.

- DC engineer signs : TPM event log during provisioning
- TPM signs : build artefacts in boot chain
- Developer signs : build artefacts
- Build chain signs : build artefacts, source code
- Log, witnesses & TPM sign : Sigsum STH
- Log & witnesses sign : all signatures mentioned
- Web CA signs : domain name

TKey - (dev.tillitis.se)

- the most open-source hardware USB security key

- The TKey's schematic, PCB design, circuit design (Verilog) and software is open source. Enjoy!



TKey - (dev.tillitis.se)

Core idea:

- based on the idea of measured boot using TCG DICE

- KDF: blake2s(UDS, blake2s(device_app), USS)