Turing-complete contract execution on L2 is thought to be achieved with TEE like Intel SGX. This seems not to be the best solution simply because this security depends on Intel's compliance.

This post just requests comments about Trultless Network (Ethereum) × Trusted Machine (FPGA) model and its problems.

Background

TEE was already a candidate of L2 solution in 2016, as TEEchan was proposed by the team headed by Emin Gun Sierer ,Joshua Lind.

[https://arxiv.org/abs/1612.07766

](https://arxiv.org/abs/1612.07766)LN was focused by the community, because of the reason that Intel should not be the trustpoint.

Ethereum has a different status quo about this topic.

Bitcoin Script is focused on operations of Bticoin like payments and its escrow, and this can be mostly same of LN functions.

EVM operates storages on Ethereum, not focused on operations of Ether, while cryptographical L2 solutions like OVM does not have same functions of EVM.

Intel SGX has almost of all functions of EVM with a verification of the process of executions, though this needs Intel's hardware level trust.

In Ethereum Community, this kind of ideas are well discussed. [1][2]

And idea about hardware level also appears as a solution about VDF. [1] [3]

Intel SGX

Intel SGX has encrypted memory "enclave".[4]

The private key is inside the circuit (e-fuse ,Provisioning Secret) and its public key is registered by Intel.

Verifications of enclave executions can be shown to validator by "Remote Attestation"[5]

usenix.org

[

](https://www.usenix.org/system/files/conference/atc17/atc17-tsai.pdf)

atc17-tsai.pdf

3.90 MB

FPGA

The theme is whether or not the things above can be samely implemented by FPGA and its relevant modules with external key generation / key import . This simply means removing trust of a maker from TEE.

(abandoned: FPGA is a mutable circuit, thus makers cannot embed backdoors inside it.

)

If FPGA can principally generate/import key externally with HDL, the maker cannot attack users with stealing the private key

(modified)

If SGX's performance is not so needed for smart contracts, FPGA's performance is to a considerable extent.

ECDSA on FPGA

TEE on FPGA is already well researched.(abandoned: This provides mutable TEE circuit.)

[http://www.cs.binghamton.edu/~jelwell1/papers/micro14_evtyushkin.pdf

](http://www.cs.binghamton.edu/~jelwell1/papers/micro14_evtyushkin.pdf)

But it’s hard to find ECDSA on FPGA with certain security and privacy against physical access.

There’s a verify circuit, but not signing.

Problem

Intel SGX’s verification of keys and executions are provided by Intel. How could we do the same thing with FPGA?

Reference

[1] see Question for Justin Drake

[https://docs.ethhub.io/other/ethereum-2.0-ama/

](https://docs.ethhub.io/other/ethereum-2.0-ama/)[2] TEE topics

[https://ethresear.ch/t/trusted-setup-with-intel-sgx/5531

](https://ethresear.ch/t/trusted-setup-with-intel-sgx/5531)

[3] VDF

[https://ethresear.ch/t/verifiable-delay-functions-and-attacks/2365

](https://ethresear.ch/t/verifiable-delay-functions-and-attacks/2365)

[4] Good explanation about TEE (it’s in English but only its title is in Japanese)

seminar-materials.iijlab.net

[

](https://seminar-materials.iijlab.net/iijlab-seminar/iijlab-seminar-20181120.pdf)

iijlab-seminar-20181120.pdf

1761.83 KB

[5] Graphene-SGX:

usenix.org

[

](https://www.usenix.org/system/files/conference/atc17/atc17-tsai.pdf)

atc17-tsai.pdf

3.90 MB