RUB

RUHR-UNIVERSITÄT BOCHUM

THE FORKING WAY: WHEN TEES MEET CONSENSUS

Annika Wilde, Tim Niklas Gruel, Claudio Soriente, Ghassan Karame

REWIRE

CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

Across

Gefördert durch
DFG Deutsche Forschungsgemeinschaft

TEEs

TEEs

TEEs

- ✗ I/O controlled by untrusted host
- ✗ No freshness guarantees
- ✗ Forking attacks (rollback & cloning)

TEEs

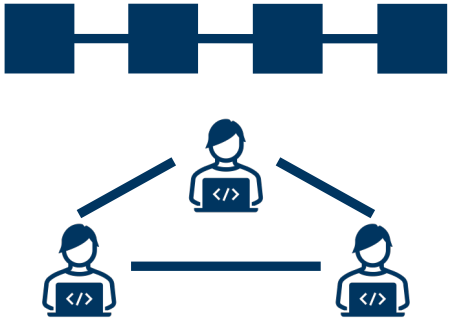
- ✗ I/O controlled by untrusted host
- ✗ No freshness guarantees
- ✗ Forking attacks (rollback & cloning)

Blockchains

TEEs

- ✗ I/O controlled by untrusted host
- ✗ No freshness guarantees
- ✗ Forking attacks (rollback & cloning)

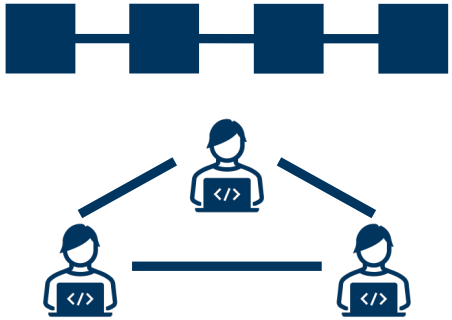
Blockchains



TEEs

- ✗ I/O controlled by untrusted host
- ✗ No freshness guarantees
- ✗ Forking attacks (rollback & cloning)

Blockchains

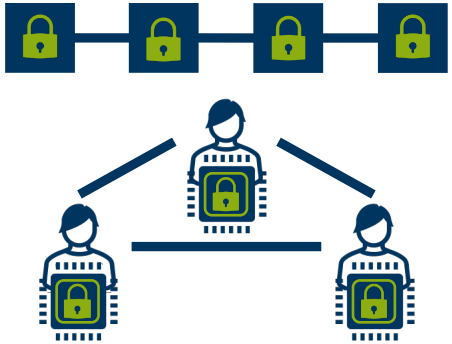


- ✗ Can only run deterministic applications
- ✗ All data must be publicly available

TEE-based blockchains

TEEs

- ✗ I/O controlled by untrusted host
- ✗ No freshness guarantees
- ✗ Forking attacks (rollback & cloning)



Blockchains

- ✗ Can only run deterministic applications
- ✗ All data must be publicly available

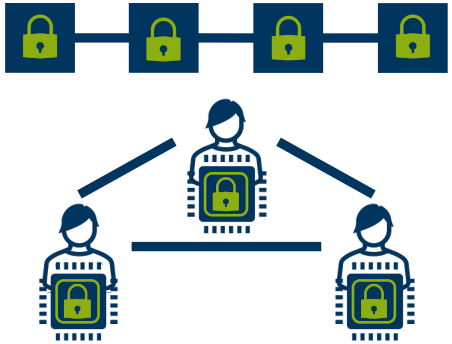
TEE-based blockchains

TEEs

✗ I/O controlled by untrusted host

✗ No freshness guarantees

✗ Forking attacks (rollback & cloning)



Blockchains

✗ Can only run deterministic applications

✗ All data must be publicly available

✓ TEEs provide randomized computing.

✓ TEEs provide confidential computing.

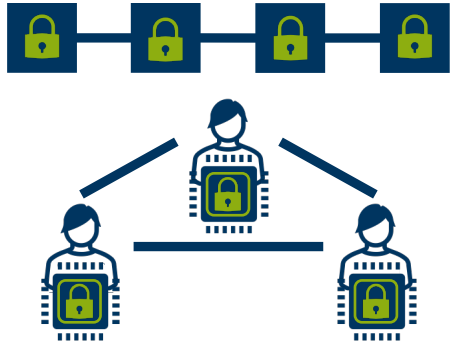
✓ Blockchains provide a total ordering of events.

✓ Blockchains protect against rollback & cloning attacks.

TEE-based blockchains

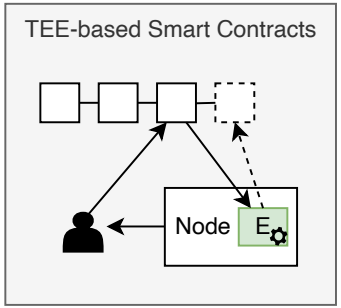
TEEs

- ✗ I/O controlled by untrusted host
- ✗ No freshness guarantees
- ✗ Forking attacks (rollback & cloning)



Blockchains

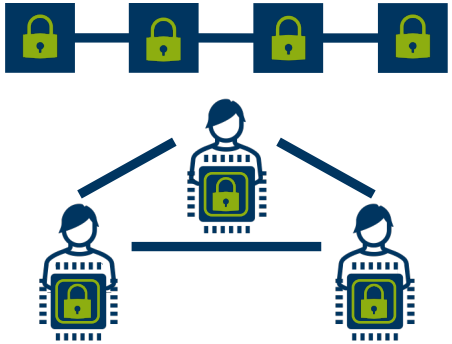
- ✗ Can only run deterministic applications
- ✗ All data must be publicly available



TEE-based blockchains

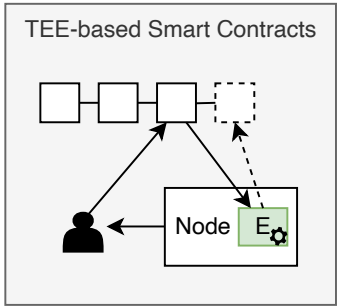
TEEs

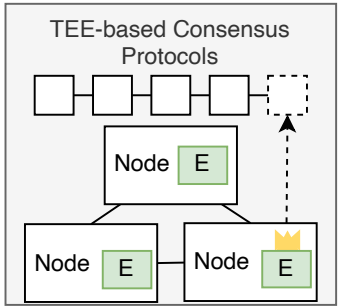
- ✗ I/O controlled by untrusted host
- ✗ No freshness guarantees
- ✗ Forking attacks (rollback & cloning)



Blockchains

- ✗ Can only run deterministic applications
- ✗ All data must be publicly available

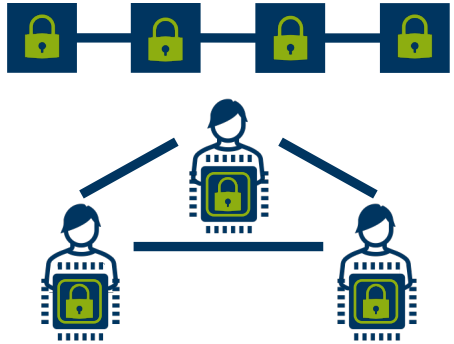




TEE-based blockchains

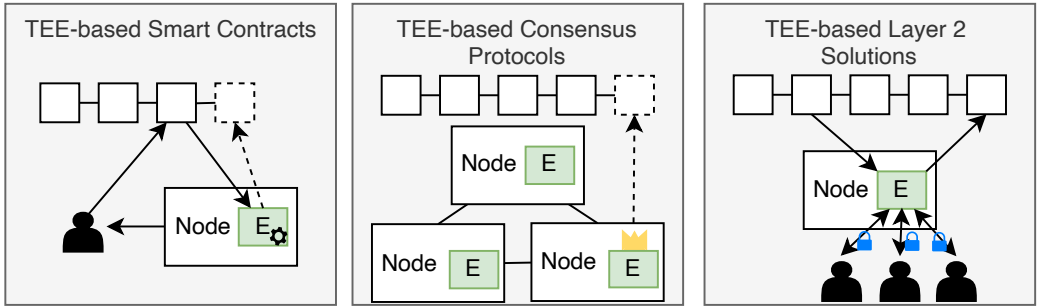
TEEs

- ✗ I/O controlled by untrusted host
- ✗ No freshness guarantees
- ✗ Forking attacks (rollback & cloning)



Blockchains

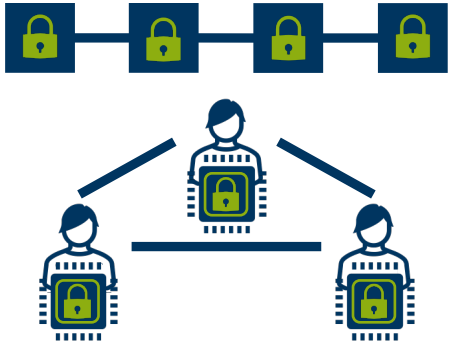
- ✗ Can only run deterministic applications
- ✗ All data must be publicly available



TEE-based blockchains

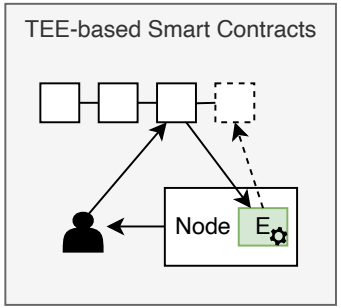
TEEs

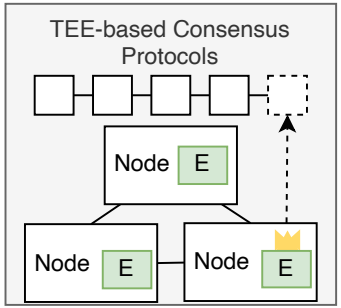
- ✗ I/O controlled by untrusted host
- ✗ No freshness guarantees
- ✗ Forking attacks (rollback & cloning)

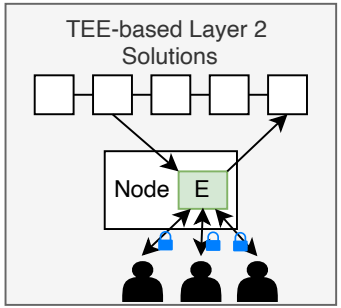


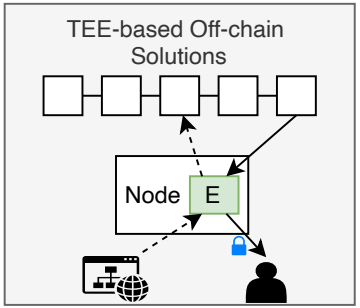
Blockchains

- ✗ Can only run deterministic applications
- ✗ All data must be publicly available



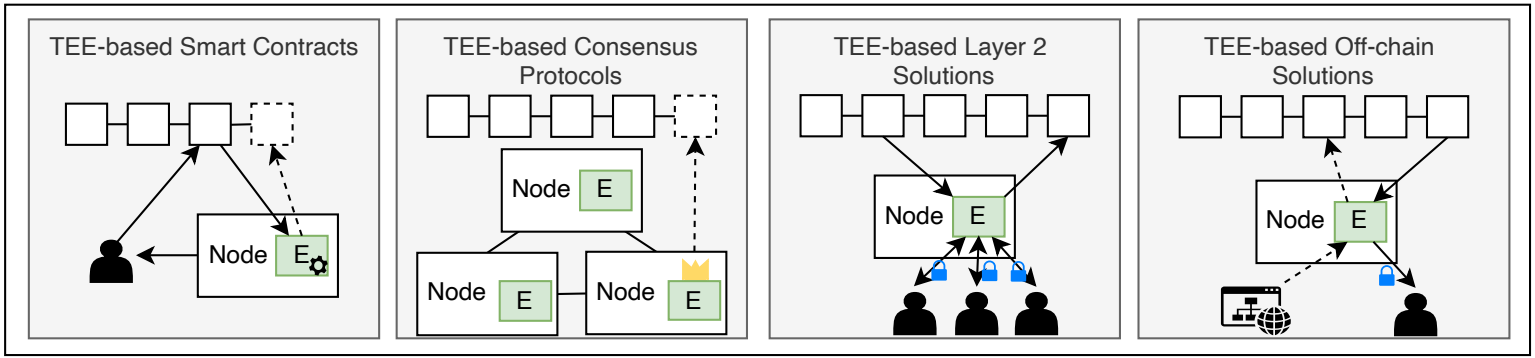




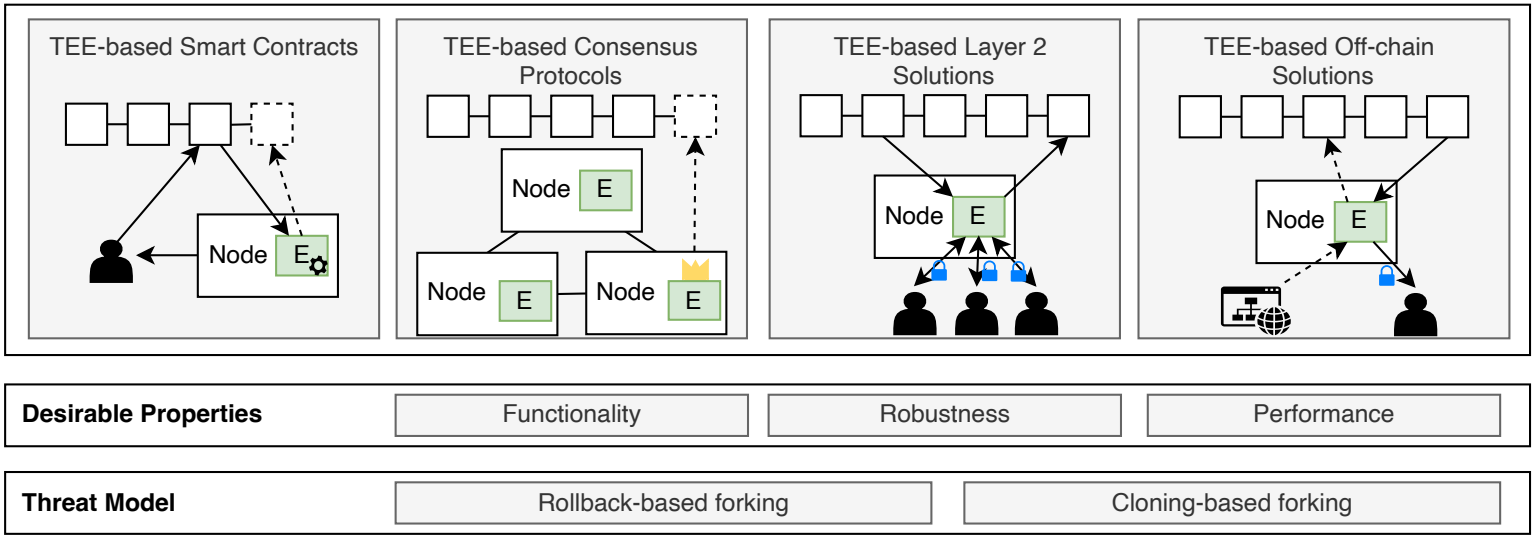


RUHR UNIVERSITÄT BOCHUM RUB

Methodology

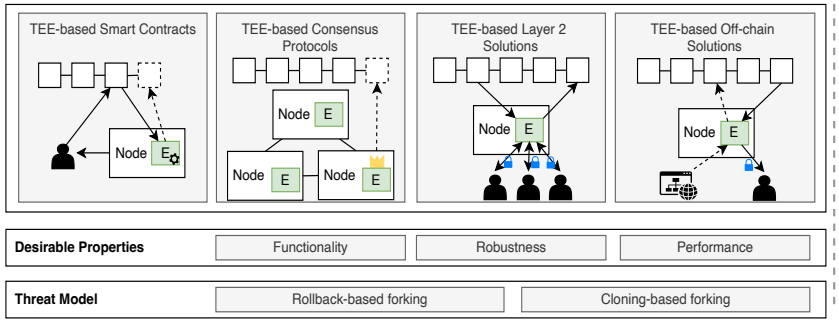


Methodology



Classification & analysis of 29 TEE-based blockchains

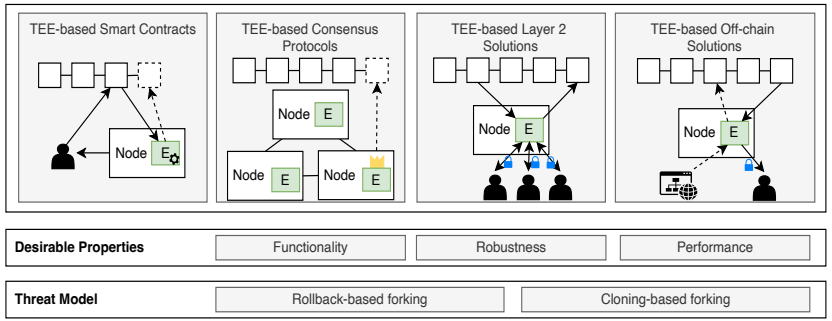
Mitigation Strategies & Pitfalls



Classification of Existing Solutions

- Stateless enclaves
- Ephemeral identities
- Fixed set of clients
- Serialization

Mitigation Strategies & Pitfalls



Classification of Existing Solutions

- Stateless enclaves
- Ephemeral identities
- Fixed set of clients
- Serialization

Limitations of Existing Solutions

- Expressiveness
- Key Management
- Fault tolerance
- Reconfiguration
- Low throughput
- Existential honesty
- Blockchain forks
- Randomized computations

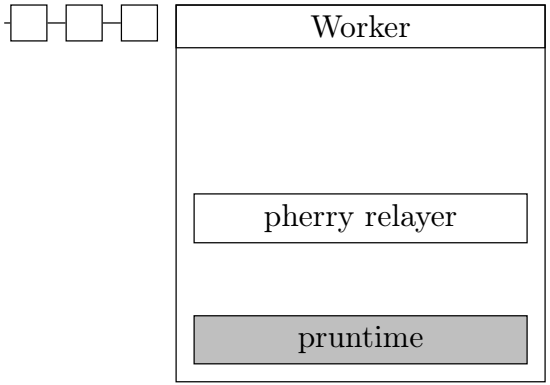
Case study: Phala[1]

- Production-ready Layer 1 blockchain
- Confidential smart contracts

[1] Phala: A Secure Decentralized Cloud Computing Network based on Polkadot, [Online] March 2022

Case study: Phala[1]

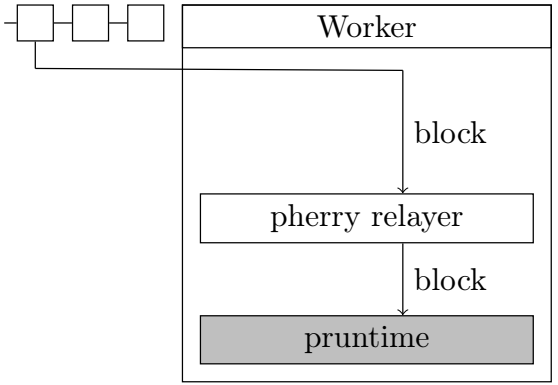
- Production-ready Layer 1 blockchain
- Confidential smart contracts



[1] Phala: A Secure Decentralized Cloud Computing Network based on Polkadot, [Online] March 2022

Case study: Phala[1]

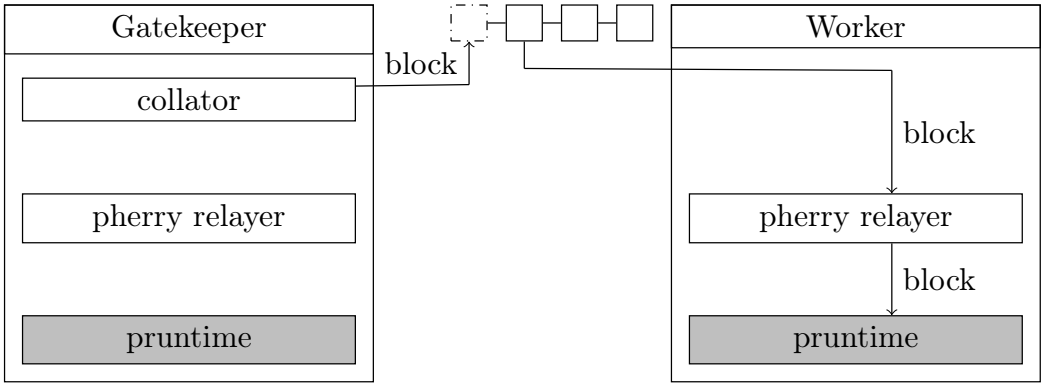
- Production-ready Layer 1 blockchain
- Confidential smart contracts



[1] Phala: A Secure Decentralized Cloud Computing Network based on Polkadot, [Online] March 2022

Case study: Phala[1]

- Production-ready Layer 1 blockchain
- Confidential smart contracts

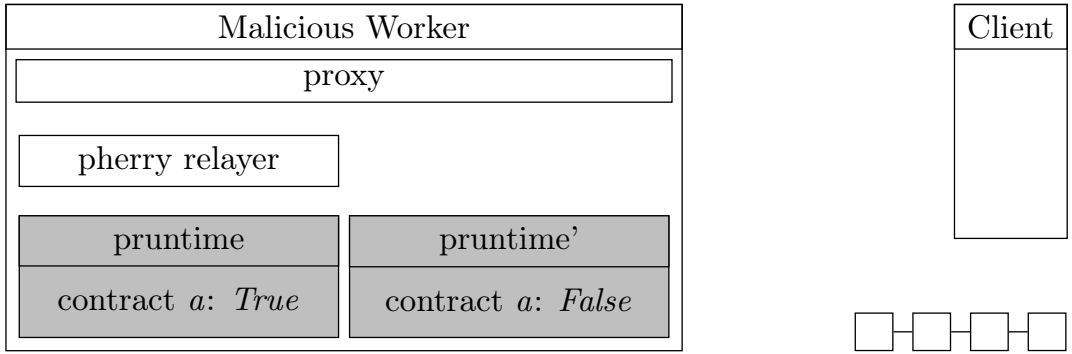


[1] Phala: A Secure Decentralized Cloud Computing Network based on Polkadot, [Online] March 2022

Case study: Phala

Cloning attack:

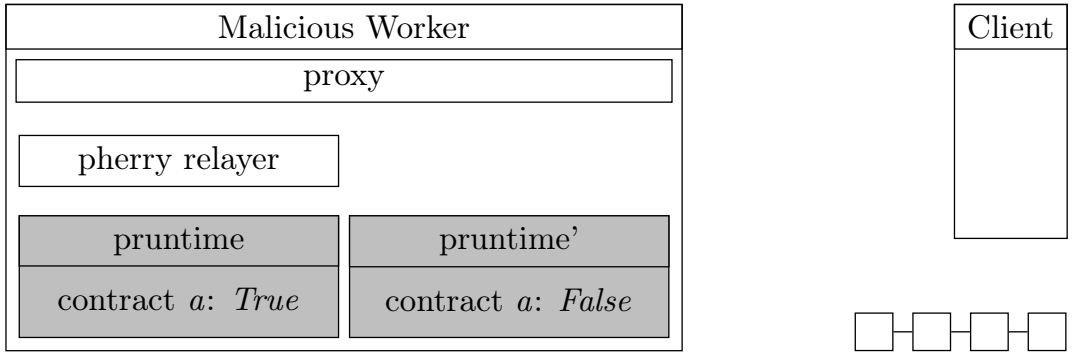
1. Malicious worker starts two enclaves and a proxy



Case study: Phala

Cloning attack:

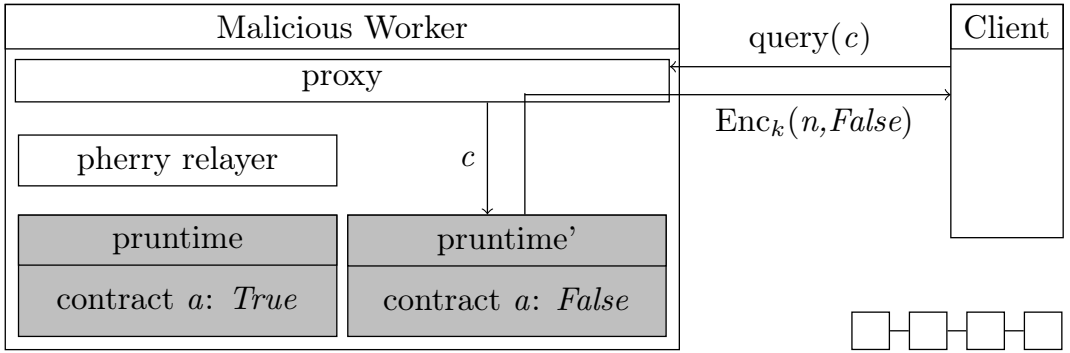
1. Malicious worker starts two enclaves and a proxy
2. Isolate one of the enclaves by terminating the relayer



Case study: Phala

Cloning attack:

1. Malicious worker starts two enclaves and a proxy
2. Isolate one of the enclaves by terminating the relayer
3. Client sends an encrypted request
4. Proxy routes it to the isolated enclave (clone)
5. Clone responds with a stale state



Case study: Phala

Countermeasure 1: Heartbeats

- Enclaves regularly issue heartbeat transactions to prove they are alive

Heartbeat
session_id
challenge_block
challenge_time
iterations
n_clusters
n_contracts

Case study: Phala

Countermeasure 1: Heartbeats

- Enclaves regularly issue heartbeat transactions to prove they are alive

Heartbeat
session_id
challenge_block
challenge_time
iterations
n_clusters
n_contracts

Case study: Phala

Heartbeat

- session_id
- challenge_block
- challenge_time
- iterations
- n_clusters
- n_contracts

Countermeasure 1: Heartbeats

- Enclaves regularly issue heartbeat transactions to prove they are alive
- Exchange heartbeats via a separate P2P network
- Enclaves check they regularly receive heartbeat messages from others

Case study: Phala

Heartbeat
session_id
challenge_block
challenge_time
iterations
n_clusters
n_contracts

Countermeasure 1: Heartbeats

- Enclaves regularly issue heartbeat transactions to prove they are alive
- Exchange heartbeats via a separate P2P network
- Enclaves check they regularly receive heartbeat messages from others
  - ✗ Existential honesty
  - ✗ Randomized computations

Case study: Phala

Countermeasure 1: Heartbeats

- ✗ Existential honesty
- ✗ Randomized computations

Countermeasure 2: Timestamping

- Include the current block height in the response to contract queries

Case study: Phala

Countermeasure 1: Heartbeats

- ✗ Existential honesty
- ✗ Randomized computations

Countermeasure 2: Timestamping

- Include the current block height in the response to contract queries
  - ✗ Randomized computations

Case study: Phala

Countermeasure 1: Heartbeats

- ✗ Existential honesty
- ✗ Randomized computations

Countermeasure 2: Timestamping

- ✗ Randomized computations

Countermeasure 3: Ephemeral IDs

- Rely on ephemeral IDs to ensure only one enclave instance is active on each node

Project	Forking Mitigations						Limitations							
	Stateless enclaves	Ephemeral identities	Fixed set of clients	Transaction replay	Time-stamping	State on the ledger	Functionality			Robustness				Performance
							L1	L4	L8	L2	L3	L6	L7	L5
TEE-based Smart Contracts														
Azure CCF [47]	✓			✓	✓		✗		✗			✗	✗	✗
CONFIDE [32]					✓				✗			✗	✗	✗
CreDB [52]					✓				⋆			⋆	⋆	⋆
Ekiden [11]						✓			✗			✗	✗	✗
Phala [9]	✓			✓	✓		✗		✗			✗	✓	✗
Secret Network [13]	✓			✓			✗		✗			✓	✗	✗
TEE-based Consensus Protocols														
Crust sWorker [53]					✓				✗			✗	✗	✗
ENGRAFT [35]	✓	✓				✓	✗		✗	✓		✗	✗	✗
MobileCoin [49]	✓						✗							
Proof of Luck [34]	✓					✓	✗		✓			✗	✗	✗
REM [33]	✓					✓	✗		✓			✗	✗	✗
TEE-based Layer 2 Solutions														
COMMITEE [42]	✓	✓					✗			✓				
FastKitten [8]			✓					✓			✓			
Hybridchain [51]		✓				✓			✗	✓		✗	✗	✗
IntegriTEE [60]						✓			✓			✗	✓	✗
Obscuro Mixer [39]	✓	✓					✗			✗				
PrivacyGuard [50]	✓						✗							
Private Chaincode [37]						✓			✓			✗	✗	✗
Private Data Objects [38]	✓					✓	✗		✗			✗	✗	✗
ShadowEth [54]					✓	✓			✗			✗	✗	✗
Teechain [40]			✓					✓			✓			
Ten [12]						✓			✓			✗	✗	✗
Tesseract [43]	✓	✓		✓	✓		✗		✗	✗		✗	✗	✗
Twilight [41]	✓	✓					✗			✗				
TEE-based Blockchain Applications														
BITE [1]					✓				✗			✗	✗	✗
LSKV [48]	✓			✓	✓		✗		✗			✗	✗	✗
sgxwallet [44]	✓						✗							
Ternoa Network [45]		✓			✓		✗		✗	✗		✗	✗	✗
Town Crier [46]	✓				✓		✗		✗			✗	✗	✗

Project	Forking Mitigations						Limitations							
	Stateless enclaves	Ephemeral identities	Fixed set of clients	Transaction replay	Time-stamping	State on the ledger	Functionality			Robustness				Performance
							L1	L4	L8	L2	L3	L6	L7	L5
TEE-based Smart Contracts														
Azure CCF [47]	✓			✓	✓		✗		✗			✗	✗	✗
CONFIDE [32]					✓				✗			✗	✗	✗
CreDB [52]					✓				⋆			⋆	⋆	⋆
Ekiden [11]						✓			✗			✗	✗	✗
Phala [9]	✓			✓	✓		✗		✗			✗	✓	✗
Secret Network [13]	✓			✓			✗		✗			✓	✗	✗
TEE-based Consensus Protocols														
Crust sWorker [53]					✓				✗			✗	✗	✗
ENGRAFT [35]	✓	✓				✓	✗		✗	✓		✗	✗	✗
MobileCoin [49]	✓						✗							
Proof of Luck [34]	✓					✓	✗		✓			✗	✗	✗
REM [33]	✓					✓	✗		✓			✗	✗	✗
TEE-based Layer 2 Solutions														
COMMITEE [42]	✓	✓					✗			✓				
FastKitten [8]			✓					✓			✓			
Hybridchain [51]		✓				✓			✗	✓		✗	✗	✗
IntegriTEE [60]						✓			✓			✗	✓	✗
Obscuro Mixer [39]	✓	✓					✗			✗				
PrivacyGuard [50]	✓						✗							
Private Chaincode [37]						✓			✓			✗	✗	✗
Private Data Objects [38]	✓					✓	✗		✗			✗	✗	✗
ShadowEth [54]					✓	✓			✗			✗	✗	✗
Teechain [40]			✓					✓			✓			
Ten [12]						✓			✓			✗	✗	✗
Tesseract [43]	✓	✓		✓	✓		✗		✗	✗		✗	✗	✗
Twilight [41]	✓	✓					✗			✗				
TEE-based Blockchain Applications														
BITE [1]					✓				✗			✗	✗	✗
LSKV [48]	✓			✓	✓		✗		✗			✗	✗	✗
sgxwallet [44]	✓						✗							
Ternoa Network [45]		✓			✓		✗		✗	✗		✗	✗	✗
Town Crier [46]	✓				✓		✗		✗			✗	✗	✗

Takeaways

Takeaways

Stateless Enclaves

- No persistent inside the enclave
- Prevents rollback attacks by design
- Cloning protection needed for non-deterministic enclaves

Takeaways

Stateless Enclaves

- No persistent inside the enclave
- Prevents rollback attacks by design
- Cloning protection needed for non-deterministic enclaves

Ephemeral Identities

- Enclaves use temporary, unique identifiers
- Can prevent cloning attacks
- Persistent state must be protected against rollback

Takeaways

Stateless Enclaves

- No persistent inside the enclave
- Prevents rollback attacks by design
- Cloning protection needed for non-deterministic enclaves

Ephemeral Identities

- Enclaves use temporary, unique identifiers
- Can prevent cloning attacks
- Persistent state must be protected against rollback

Fixed Set of Clients

- Operates with a predefined, trusted set of clients
- Can mitigate rollback attacks
- Cloning protection needed for non-deterministic enclaves

Serializing State

- Logs enclave input/output through a consistent layer
- Can prevent rollback and cloning
- Cloning protection needed for non-deterministic enclaves

Takeaways

Stateless Enclaves

- No persistent inside the enclave
- Prevents rollback attacks by design
- Cloning protection needed for non-deterministic enclaves

Ephemeral Identities

- Enclaves use temporary, unique identifiers
- Can prevent cloning attacks
- Persistent state must be protected against rollback

Fixed Set of Clients

- Operates with a predefined, trusted set of clients
- Can mitigate rollback attacks
- Cloning protection needed for non-deterministic enclaves

Serializing State

- Logs enclave input/output through a consistent layer
- Can prevent rollback and cloning
- Cloning protection needed for non-deterministic enclaves