Smart Contract

Xiaojie Zhu

April 17, 2018

1 Smart Contract

A smart contract is a program that runs on the blockchain and has its correct execution enforced by the consensus protocol. Smart contracts operate as *autonomous actors*, whose behaviour is completely predictable. Any on-chain logic can be expressed as a function of on-chain data inputs. Since smart contracts can handle large number of virtual coins worth hundreds of dollars, many attacks are launched to steal virtual coins from them. Specifically, there are four attacks shown as follows.

- Transaction-ordering dependence. The reason includes two aspects. The first is that even a benign invocation to the contract may yield an unexpected result to users there are concurrent invocations. The second is that malicious user intensionally exploit the order of transactions to gain more profits.
- Timestamp dependence. Some contracts may use the block timestamps as a triggering condition to execute some critical operations. Since a miner is able to set the block timestamp based on its own interests, it is risky to set timestamp as *salt*.
- Mishandled exceptions. The result of calling a contract is easily ignored to check, which results in many exceptions handled improperly.
- Reentrancy vulnerability. The well-known attack to *TheDao* is vulnerable to this attack, which results in 3600000 Ether loss. A naive example of this attack is to execute contract many times to send virtual coins from one address to another address before the sender's balance reaches zero.

To handle above threats, following improvements are proposed.

- Guarded transactions. The state and guard condition is added such that transaction execution must satisfy the requirement.
- Deterministic timestamp. One solution is to translate from block timestamp to block numbers.

- Better exception handling. The new solution is to automatically propagate the exception at the level of EVM from callee to caller.