Not so immutable: Upgradeability of Smart Contracts on Ethereum

Mehdi Salehi, Jeremy Clark, Mohammad Mannan

Concordia University, Montreal, Canada

The key promise of a smart contract running on Ethereum is that its code will execute exactly as it is written, and the code that is written can never be changed. While Ethereum cannot maintain this promise unconditionally, its assumptions (*e.g.,* cryptographic primitives are secure and well-intentioned participants outweigh malicious ones) provide a realistic level of assurance.

The immutability of a smart contract's code is related to trust. If Alice can validate the contract code, she can trust her money to it and not be surprised by its behavior. Unfortunately, disguising malicious behavior in innocuous-looking code is possible ('rug pulls'), and many blockchain users have been victims. On the other hand, if the smart contract is long-standing with lots of attention and security assessments from third-party professional auditors, the immutability of the code can add confidence.

The flip side of immutability is that it prevents software updates. Consider the case where a security vulnerability in a smart contract code is discovered. Less urgently, some software projects may want to roll out new features, which are also blocked by immutability. There is an intense debate about whether this is positive or negative, with many claiming that 'upgradeability is a bug.' [1]

Modern smart contracts use software tricks to enable upgradeability, raising the research questions of *how* upgradeability is achieved and *who* is authorized to make changes. In this research, we summarize six upgradeability patterns. We also extensively evaluate different upgradeability methods to give the developers an idea about the pros and cons of each method, and to help them to choose the pattern that fits into their desired system. We develop a measurement framework for finding how many upgradeable contracts are on Ethereum that use certain prominent upgrade patterns called *Upgradeable Proxy Pattern*. We find 1.4 million proxy contracts which *8,225* of them are unique upgradeable proxy contracts. We also measure how they implement access control over their upgradeability: about 50% of them are controlled by a single Externally Owned Address (EOA), and about 14% are controlled by multi-signature wallets in which a limited number of persons can change the whole logic of the contract.

This talk aims to highlight that immutability, as a core property of blockchain, is oversold. Immutability has already been criticized for being dependent on consensus–both technical and social– however, the widespread use of upgradeability patterns further degrades immutability. Finally, the prominence of contracts that can be upgraded with a single private key (i.e., externally owned account) calls into question how decentralized our DApps (decentralized applications) really are. If the upgrade process is corrupted through a key theft or by a rogue insider, the whole logic of the contract can be changed to the attacker's benefit.

[1] "Upgradeability Is a Bug", Steve Marx, Medium, Feb 2019.