Σ-State Authentication Language, an Alternative to Bitcoin Script

Alexander Chepurnoy

IOHK Research
{alex.chepurnoy}@iohk.io

Every coin in Bitcoin is protected by a program in stack-based Script language. An interpreter for the language is evaluating the program against a redeeming program (in the same language) as well as a context (few variables containing information about a spending transaction and the blockchain), producing a single boolean value as a result. While Bitcoin Script allows for some contracts to be programmed, its abilities are limited while many instructions were removed after denial-of-service or security issues discovered. To add new cryptographic primitives, for example, ring signatures, a hard-fork is required.

Generalizing the Bitcoin Script, we introduce a notion of an *authentication language* where a verifier is running an interpreter which three inputs are a *proposition* defined in terms of the language, a *context* and also a *proof* (not necessarily defined in the same language) generated by a prover for the proposition against the same context. The interpreter is producing a boolean value and must finish evaluation for any possible inputs within constant time.

We propose an alternative authentication language, named Σ-State. It defines guarding proposition for a coin as a logic formula which combines predicates over a context and cryptographic statements provable via Σ-protocols with AND, OR, k-out-of-n connectives. A prover willing to spend the coin first reduces the compound proposition to a compound cryptographic statement by evaluating predicates over known shared context (state of the blockchain system and a spending transaction). Then the prover is turning a corresponding Σ-protocol into a signature with the help of a Fiat-Shamir transformation. A verifier (a full-node in a blockchain setting) checks the proposition against the context and the signature with an interpreter. Language expressiveness is defined by a set of predicates over context and a set of cryptographic statements. We show how the latter could be updated with a soft-fork by using a language like ZKPDL [1], and how the former could be updated with a soft-fork by using versioning conventions. We propose a set of context predicates for a Bitcoin-like cryptocurrency with a guarantee of constant verification time. We provide several examples: ring and threshold signatures, pre-issued mining rewards, crowdfunding, demurrage currency.

References

1. Sarah Meiklejohn et al. Zkpdl: A language-based system for efficient zero-knowledge proofs and electronic cash. In *USENIX Security Symposium*, volume 10, pages 193–206, 2010.