

Go Language Support in Hyperledger Fabric Private Chaincode

Ricardo Zappoli

Master thesis in Computer Science

Smart contracts are a great invention that enables running applications in a distributed way and enables them to take advantage of distributed ledgers' capabilities. Since their inception, the goal of a lot of new systems and frameworks has been to make these smart contracts easier and easier to write and to use, more and more powerful and with ever-increasing features. Hyperledger Fabric made the developing process more straightforward, by allowing smart contracts, here called chaincodes, to be written in well-known programming languages, i.e., Java and Go. Hyperledger Fabric Private Chaincode (FPC) went a step further, by allowing them to treat confidential data within Intel SGX enclaves. Thus, it is in this perspective that this thesis aims to go another step further in that direction, by allowing FPC chaincodes to be written in Go, making them as easy to use as with Fabric, and as secure as with FPC.

Our approach is to recreate a Go Chaincode Enclave and to run it inside an Intel SGX enclave with EGo, a new commercial framework whose purpose is to make Golang applications confidential by allowing them to run with Intel SGX. This is implemented in a new module within FPC, where we have translated the FPC Stub Interface to a Go version and took advantage of an existing mock enclave to make it handle real chaincodes. After evaluating the project, we found a slightly higher end-to-end latency with our solution compared to the classic version of FPC, but with a difference not significant enough to outweigh the benefits of the extension. We show these benefits in a usability evaluation of the extension, where we run existing Golang chaincodes samples with our solution. While remote attestations are not supported due to compatibility issues, making the extension an unsafe proof-of-concept at the moment, we are confident in its ability to ever so simplify Fabric confidential chaincodes in the future.

Mark: 5

Prof. Christian Cachin