MobileCoin

December 11, 2017

1 Motivation

Applications that make use of crypto-currency and blockchain technology are often difficult to deploy in practice, particularly in mobile environments. Implementors currently face deployment challenges around device resource constraints, transaction times, and key management, all of which often contribute to a negative user experience. Mobile applications don't have the ability to synchronize an entire multi-gigabyte blockchain, minutes-long transaction times are unacceptable for typical use cases, and end users are not equipped to reliably maintain secret keys over a long period of time.

As a result, most attempts at building a compelling crypto-currency user experience unfortunately resort to trusting a third party service to manage keys and validate transactions. This largely sacrifices the primary benefits offered by crypto-currency to begin with.

MobileCoin is an effort to develop a fast, private, and easy-to-use crypto-currency that can be deployed in resource constrained environments to users who aren't equipped to reliably maintain secret keys over a long period of time, all without giving up control of funds to a payment processing service.

2 Experience

The technical design of MobileCoin is centered around a target user experience for integrating crypto-currency into mobile messaging apps like WhatsApp or Signal. A user should be able to install the app, enter a 4 digit PIN, and send/receive funds to/from other users addressed by their phone number or user identifier. Transactions should take less than a second, funds should immediately be available for use, and neither the messaging service nor any other third party should learn anything about a user's account balance or transaction history (such as who is paying who). At any point, a user should be able to reinstall the app or get a new phone and regain secure access to their funds simply by entering a 4-digit PIN. Payments should also be possible across apps and networks.

3 Design

MobileCoin starts by recognizing that not all clients are capable of participating in a P2P network, and proposes a federated approach instead. The MobileCoin network design is made up of **nodes**, where each node is designed to serve **users**.

Nodes do the heavy lifting of tasks that are ill-suited for user clients, such as maintaining an expansive ledger and processing high-throughput low-latency transactions, but are designed such that a *node operator should not have access to their users' funds nor learn anything about their users' balances and transaction history.*

This is accomplished via a layered approach, combining several levels of protection for defense-in-depth and forward-secrecy.

3.1 SGX

All MobileCoin nodes are designed to run in an SGX secure enclave. An SGX enclave is isolated from the host OS in hardware-encrypted RAM, which prevents the node operator from being able to "see" into the enclave, although care must be taken to avoid information leaks through memory access patterns. SGX also supports a feature known as **remote attestation**, which allows a remote client to determine that a server is running a specific piece of software inside an SGX enclave over a network. By doing remote attestation before establishing encrypted communication channels between nodes, the entire MobileCoin ledger is designed to remain sealed within SGX enclaves across the entire network. This means that the ledger is "public" and distributed to all MobileCoin nodes, but will also simultaneously never be accessible or viewable by humans (even the operators of the MobileCoin nodes) so long as SGX and the MobileCoin software remains secure.

3.2 Transaction privacy

MobileCoin does not rely solely on SGX for maintaining transaction privacy. Transactions are designed to employ CryptoNote[1] one-time addresses and one-time ring signatures, so MobileCoin will still maintain transaction privacy through unlinkable addresses if an attacker is able to defeat SGX and view transactions on the network.

3.3 Consensus

Owing to its federated nature, MobileCoin nodes are designed to use the Stellar Consensus Protocol[2] to synchronize a ledger, which should allow for sub-second transactions under normal circumstances, along with decentralized control and

[1] https://cryptonote.org/whitepaper.pdf

[2] https://www.stellar.org/papers/stellar-consensus-protocol.pdf

flexible trust. This also allows nodes to avoid storing a full blockchain of transaction history, since it is only necessary to maintain a ledger of address → value mappings, as well as the list of used key images to prevent double spending. This provides a certain measure of forward secrecy[3]. Even though one-time ring signatures hide the source of a transaction among a large set of possible candidates, using SCP means that information can be discarded entirely after a transaction completes, rather than being maintained in a block chain forever.

3.4 Key management

Running MobileCoin in an SGX enclave allows nodes to securely manage keys for users. A client can perform remote attestation to its MobileCoin node before transmitting its keys into the remote enclave along with a short recovery PIN. The MobileCoin node can then rate limit authenticated access to the keys, while the enclave prevents the node operator or anyone who compromises the node from circumventing the software and attempting to brute force access to the keys directly. In this way, user keys can reside safely in a node and survive across application reinstalls or lost devices, without having to trust the node operator or the security of the node computer, and without having to memorize or safely store extremely long recovery passphrases.

4 Life of a MobileCoin transaction

1. At intall time, Alice's client generates a MobileCoin keypair and short recovery PIN.

2. At install time, Alice's client performs remote attestation with its MobileCoin node, establishes a secure communication channel into the remote enclave, and transmits its keypair along with its recovery PIN.

3. To send Bob a payment, Alice's client looks up Bob's public key.

4. Alice's client generates a CryptoNote one-time public key for Bob.

5. Alice's client generates a CryptoNote one-time ring signature for the transaction.

6. Alice's client transmits the pending transaction to its MobileCoin node.

7. The node synchronizes the transaction to the network using Stellar Consensus Protocol. The MobileCoin ledger is updated to reflect the transaction's output values, as well as the key image generated as part of the one-time ring signature in order to prevent double spending. Everything else is discarded.

8. Bob's MobileCoin node uses Bob's CryptoNote tracking key to recognize the one-time public key

[3] https://en.wikipedia.org/wiki/Forward_secrecy

9. Bob's MobileCoin node sends Bob's client a message, which can then calculate the private key that corresponds to the generated one-time public key.
10. Bob has now successfully received a payment.

The transaction completes in less than a few seconds, all transaction and balance information is kept private within SGX enclaves across the network such that the transaction itself is never visible, transaction privacy is further protected with CryptoNote one-time addresses and one-time ring signatures if an attacker is able to forge SGX remote attestation in order to connect to the network with modified software, the node operator or attackers who compromise a node never have access to user keys or user data, and users can switch phones or reinstall the app and maintain access to their funds simply by entering a short PIN.

5 MobileCoin wallet

MobileCoin is designed so that a mobile messaging application like WhatsApp, Facebook Messenger, or Signal could integrate with a MobileCoin wallet. The messaging application would be able to securely recover the information it needs in order to construct and validate transactions from its MobileCoin node on install or reinstall, and would receive updates from its MobileCoin node without having to maintain persistent network connectivity. A MobileCoin wallet integrated into a messaging app like WhatsApp, Facebook Messenger, or Signal could also look up payment destination public keys based on username, and transmit an encrypted message with a deniable signature to the recipient of a transaction in order to prove where the payment originated.