Riktig - Technical Overview v. 1

Tigerström, Viktor
viktor@riktig.io

Larsson, John
john@riktig.io

1 Introduction

In this document we propose Riktig, an open source authenticity protocol for physical items which allows for anyone with a compatible reader to trustlessly and globally establish whether an item has been authorized by the expected issuer, i.e. is authentic.

2 Motivation

There is good reason to be suspicious of physical item authenticity. The only way to trust that a physical item is authentic, is to have trust in that item's entire chain of custody since it's inception.

The current state of authenticity verification of physical items is as much art as science. Depending on the type of item, the most reliable choice for verification ranges from comparing the item with images of known replicas, to consulting expensive professionals.

Closed and centralized solutions exist whereby items are tagged with identifying data through barcodes, RFID, DNA tagging or through other means. All of these methods are reliant on a central repository of information against which identity data will be checked. In other words, authenticity is deduced from querying a repository to assert that it is aware of the identity.

This method is unreliable because the identification data can easily be replicated by a malicious actor and used to tag inauthentic items. It is also unsuitable for certain types items that have an expected lifetime long enough for it not to be certain that these central repositories will always be available. For example the provider of such a verification mechanism may go out of business.

These problems would be resolved if we could replace the centralized repositories with a public decentralized store of information, and tagging the items in a way that makes it impossible to clone the identity data.

3 Problem Statement

A satisfactory solution must have the following properties.

1. It must be trustless, i.e. without third-party risk to either the issuing party or authorizing party
2. It must be open to participation by anyone
3. It's validation mechanism must be extensible to suit different use cases
4. It must be easy to verify globally with ubiquitous hardware
5. It must withstand cloning attempts, even where an attacker has physical access to the item
6. It must be economically feasible

4 Solution

We propose a solution that combines the digital signatures we know from digital files, and bring them into the physical world. This is done by adding a digital component (a chip) to the item which has a secure element and is capable of asymmetric key cryptography, which thereby can identify itself without revealing sensitive information.

The chip contains a cryptographic key pair, of which the public key can be verified by sending a signing challenge to the chip by the verifying party. At the time of issuance, the authorizing party makes a transaction to a UTXO-based blockchain address which is deterministically generated from the data contained in the group of chips to be issued. A merkle tree structure is created from the data contained in the chip(s). This allows for authorizing many chips with only one blockchain transaction which makes it economically feasible on open public blockchains at scale. The merkle path for each chip is inserted into the specific chip, and can later be used to derive the merkle root.

Given that a verifying party knows the expected authorizing party's public key, they can now trustlessly verify that a chip attached to an item is linked to the authorizing party with the following steps:

1. Verifying that the chip contains the private key corresponding to the public key it exposed by sending a message to the chip and verifying the resulting message.

2. Deriving the address for the group of chips without the need to connect to a centralized party, as all required information to derive it is stored on the chips themselves.

3. Verifying that a transaction between an authorizing party and the address derived from the chips exist on the blockchain. These transactions are easily filterable by a Neutrino node due to the ScriptPubKey being derivable from the chip data.

5 Issuance Group Address

Once the public key of the chip has been verified, the address of the chip must be derived. This is the address of the group of chips that were authorized at the same time, and to which the authorizing party has sent an authentication proof. This is called the issuance group address.

To derive this address using the information stored on a chip the following steps are required

1. Concatenate the compressed public key represented as a byte array followed by the metadata stored on the chips as a byte array. The resulting byte array will hereafter be called the chip information byte array.

2. Hash the chip information byte array multiple times with the merkle path contained in the merkle path data array. The resulting array after hashing is the merkle root.

3. The merkle root is converted into a blockchain address. The address encoding version is stored as metadata on the chip and is used to encode the merkle root hash, which results in the issuance group address.

Because multiple chips are authorized with just one output the protocol becomes scalable and economically feasible even on expensive-to-use blockchains. The theoretical limit for the amount of items that can be authorized with just one output is 2^{50} (based on 1100 bytes merkle path storage, with the first protocol version and Hash160).

6 Authentication Transactions

An authorizing party has a known master public key. An authentication proof between an authorizing party and an item is a transaction on the blockchain to the issuance group address. How this transaction is structured and is to be evaluated is specified by a protocol version. This is how the protocol supports customizing the validation rules for authentication proofs to suit different use cases.

The characteristics of such authentication transactions are generally that

1. One input of the transaction's inputs contains a signature by a private key corresponding to a public key which is derived from the authorising party's master public key. The message signed by the private key is the transaction information, but the sighash flag used may vary. This input is called the authorizing input.

2. One output of the transaction's outputs is referencing an address which has been derived using the information stored on the chip. For many protocol versions this output is generally required to be the same index in the transaction's outputs, as the index of the authorizing input in the transaction's inputs. However this is ultimately decided by the specific protocol version used for the authentication proof.

7 Hardware requirements

In order to authorize an item using the protocol, a secure hardware chip with specific features must be attached to, incorporated into, or otherwise be present together with the item throughout it's life cycle. An example of a class of RFID chips that are compatible and readily available, are contactless Javacard chips. However, any other class of RFID chips that meet the below criteria will work.

1. It must support asymmetric cryptography such as ECC or RSA.
2. It must be hard and costly enough to extract the private key from the chip that it becomes economically infeasible to extract it for an attacker, even with physical access to the chip.
3. A public key must be easily read by a compatible reader.
4. Support for signing of messages using the stored private key, which are sent to the chip by a compatible reader.
5. It must support inserting of merkle path data as well as arbitrary extra data by the authorizing party, as well as supporting reading it by the verifying party
6. For ease of verification the chip should ideally (but not necessarily) support RFID because all modern smartphones include a reader.

By using such a chip we can ensure that the chip can be identified without being vulnerable to cloning attacks.