# How to Physically Hold Your Bitcoins ?

Houda Ferradi[1], Antoine Houssais[2], and David Naccache[2]

AlgoEngitec, 60 rue Francois 1er, 75008 Paris
DIÉNS, ÉNS, CNRS, PSL University, Paris, France
45 rue d'Ulm, 75230, Paris CEDEX 05, France
given_name.family_name@ens.fr,

**Abstract.** The rise of virtual currencies has revolutionized the way we conduct financial transactions. These digital assets, governed by intricate online protocols, have rapidly gained prominence as a viable medium of exchange, offering convenience and security. However, as we delve deeper into the digital realm, a challenge persists: How can we bridge the gap between the virtual and the physical? This paper tackles this challenge by proposing a way to materialize virtual coins and make them physically exchangeable *offline* at the cost of some plausible trust assumptions.

## 1 Introduction

Virtual currencies are digital or electronic forms of currency that can be used as a medium of exchange for goods and services. Their value is determined by market demand and supply. Some well-known examples of virtual currencies include Bitcoin, Ethereum, and Dogecoin.

Virtual currencies are exchanged using rather complex computer programs and involve online interactions allowing to maintain a distributed blockchain that serves as a public financial transaction database.

It is reasonable to assume that users also desire to exchange cryptocurrency off-line by just giving physical objects to each other. This paper describes a way allowing to achieve such a goal.

### 1.1 The Proposed Protocol

We identify three entities: the issuing bank (or authority) $\mathcal{B}$, the payer's physical coin $\mathcal{C}$ and the payee's terminal $\mathcal{T}$.

The idea consists in enclosing value information (called *financial credentials*) in an RFID chip enclosed in a casino token-like plastic casing. The RFID chipundergoes a life-cycle comprising three distinct states: : *sealed*, *unsealed* and *recycled*. We depart from the assumption that unsealing and recycling are rare, as are the deposits of cash at a bank.

When manufactured, the chip is loaded with financial credentials $\omega$ for some face value (e.g. 0.54 finney $\simeq$ \$1) and a PKI-certified authentication key-pair $\mathsf{sk}, \mathsf{pk}$. In the sealed state, the chip will keep $\omega$ secret but accept to prove (typically

using a zero-knowledge proof) that the chip contains $\mathsf{sk}$. When unsealed, the chip will refuse to authenticate itself but will allow free reading of $\omega$. When unsealed, the chip will accept the reloading of new financial credentials $\omega'$ for being re-circulated.

Trading is done using a totally offline phone application that attempts to authenticate the chip. If successful (i.e. sealed), the chip will change hands and be physically accepted as a coin for its face value. Otherwise the (unsealed) chip will be accepted for a fixed deposit value (e.g. ¢50) and returned to the issuing bank. Unsealed chips returned to the bank are reloaded with a new $\omega'$s and circulated again.

The system is not limited to cryptocurrencies as it allows enclosing in circulating plastic tokens any top-up values (e.g. airtime, meal vouchers, etc).

The method is interesting because it lends itself to even less monitoring than classical cryptocurrencies as coins may change hands without leaving any digital traces. In particular coin swapping machines could be installed in public places and, in exchange of a moderate fee, shuffle coins between users. This will break the traceability chain of cryptocurrency and improve anonymity.

## 2 Underlying Technologies & Building-Blocks

### 2.1 RFID Plastic Coins (Casino Chips)

Radio-Frequency Identification (RFID) technology has a wide range of applications across various industries such as access control, inventory management, and supply chain tracking [3], this is due to its ability to efficiently track and identify objects using radio waves.

Many manufacturers produce plastic coins containing RFID chips. One of the biggest consumer of those tokens is the gambling industry. The technology is identical to NFC cards except in its form factor (Figure 1). Such tokens are often used in casinos to track and authenticate gaming chips, monitor player behavior, and prevent fraud.

**Fig. 1.** Typical Casino Chips.

The tamper-resistant microprocessor inside the token $\mathcal{C}$ has basic public-key cryptographic capabilities.

### 2.2 RFID Terminals

The RFID terminal $\mathcal{T}$ interacts with the coin $\mathcal{C}$. $\mathcal{T}$ can take several form factors:

**An ad hoc hardware:** A standalone box with a simple LED turning into either red (authentication failed) or green (authentication succeeded). If several coin values and face values are planned the LED display will also show the value contained in $\mathcal{C}$.

**A smartphone:** In this case the above functions are performed by an application.

**A contactless reader already connected to a shop appliance:** The system can also exploit already existing contactless readers which are already connected to point-of-sale terminals or cash registers.

### 2.3 Identification Protocols

The proposed system requires an identification protocol. The identification protocol allows $\mathcal{C}$, to prove to $\mathcal{T}$ that $\mathcal{C}$ contains a secret $\mathsf{sk}$ without revealing $\mathsf{sk}$ to $\mathcal{T}$.

The increased popularity of IoT implementations recently invigorated the interest in the resource-preserving protocols of the late 1980s initially designed for smart-cards. By then, cryptoprocessors were expensive and cumbersome, hence the research community developed astute ways to identify and sign with scarce resources.

One such particularly elegant procedure is the Fiat-Shamir protocol [1] that we do not restate here. In a nutshell, using $\simeq 20$ modular multiplications, a Fiat-Shamir prover ($\mathcal{C}$) can prove to a verifier ($\mathcal{T}$) that $\mathcal{C}$ knows (contains) a secret $\mathsf{sk}$.

Another option consists in using pre-computed DSA coupons [2]. This limits the number of authentication sessions to the number of coupons pre-loaded into $\mathcal{C}$. Assuming that a coins changes hands less than 2500 times between re-sealings, $2500 \times 28 = 70$K bytes are necessary to store coupons in the chip.

We denote by $\mathsf{auth}(\mathsf{pk}) \in \{\mathsf{T}, \mathsf{F}\}$ the result of an authentication session with respect to a public-key $\mathsf{pk}$. Also, $\mathsf{ver\_cert}(\Delta, \mathsf{cert}) \in \{\mathsf{T}, \mathsf{F}\}$ will represent the result of checking the certificate $\mathsf{cert}$ with on data $\Delta$ with respect to some root of trust.

## 3 Transaction Protocols

In the following sections the underline word Check stands for "In case of failure abort all further operations and inform the user (e.g. red LED or a buzzer signal) that a failure occured".

### 3.1 System Setup

A bank $\mathcal{B}$ generates a PKI allowing to certify public-keys[1] and creates $n$ virtual wallets, each containing a cryptocurrency amount $x$. We denote by $\omega_i$ the credentials (keys) of wallet $i$. A coin $\mathcal{C}$ not containing any currency is worth a deposit value $d$.

### 3.2 Coin Issuance (Sealing)

$\mathcal{B}$ does the following for $i = 1, \ldots, n$:

1. Check that the status of $\mathcal{C}_i$ is unsealed.
2. Load $\omega_i$ into $\mathcal{C}_i$.
3. Trigger the generation of two key-pairs $\mathsf{sk}_i^s, \mathsf{pk}_i^s$ and $\mathsf{sk}_i^u, \mathsf{pk}_i^u$ in $\mathcal{C}_i$.
4. Load into $\mathcal{C}_i$ a certificate $\mathsf{cert}_i^s$ on $\mathsf{pk}_i^s$.
5. Load into $\mathcal{C}_i$ a certificate $\mathsf{cert}_i^u$ on $\omega_i, \mathsf{pk}_i^u$.
6. Switch $\mathcal{C}_i$ into sealed mode.
7. Sell $\mathcal{C}_i$ to a user for the equivalent of $x + d$.

$\mathsf{cert}_i^s$ is only readable in sealed mode and $\mathsf{cert}_i^u$ is only readable in unsealed mode.

### 3.3 Physical Coin Transfer

To transfer a coin $\mathcal{C}_i$ between users the following is done:

1. The payee places the coin $\mathcal{C}_i$ on his terminal $\mathcal{T}$.
2. $\mathcal{T}$ reads the status of $\mathcal{C}_i$.
3. If the status is "sealed"
   (a) $\mathcal{T}$ reads $\mathsf{pk}_i^s$ and $\mathsf{cert}_i^s$.
   (b) $\mathcal{T}$ Checks that $\mathsf{auth}(\mathsf{pk}_i^s) \wedge \mathsf{ver\_cert}(\mathsf{pk}_i^s, \mathsf{cert}_i^s)$
   (c) $\mathcal{T}$ informs the payee that he can accept $\mathcal{C}_i$ for a face value of $x$.
   (d) $\mathcal{C}_i$ physically changes hands.
4. If the status is "unsealed"
   (a) $\mathcal{T}$ reads $\omega_i, \mathsf{pk}_i^u$ and $\mathsf{cert}_i^u$.
   (b) $\mathcal{T}$ Checks that $\mathsf{auth}(\mathsf{pk}_i^u) \wedge \mathsf{ver\_cert}(\omega_i | \mathsf{pk}_i^u, \mathsf{cert}_i^u)$
   (c) $\mathcal{T}$ informs the payee that he can accept $\mathcal{C}_i$ for a deposit value of $d$.
   (d) The payee accepts $\mathcal{C}_i$ against a deposit value of $d$.
   (e) The payee returns $\mathcal{C}_i$ to $\mathcal{B}$ and gets $d$ from $\mathcal{B}$.

### 3.4 Cash-Out (Unsealing)

Verifying that a coin is unsealed does not require any on-line communication but transferring the contents of the coin into a wallet requires going on-line.

1. The user places $\mathcal{C}_i$ on $\mathcal{T}$ for a long duration $\tau$ (e.g. one minute).
2. $\mathcal{T}$ Checks that $\mathcal{C}_i$ is "unsealed".
3. After $\tau$ is elapsed, $\mathcal{C}_i$ switches to "unsealed", disabling the use of $\mathsf{sk}_i^s, \mathsf{pk}_i^s, \mathsf{cert}_i^s$ and allowing the free reading of $\omega_i, \mathsf{pk}_i^u, \mathsf{cert}_i^u$.
4. $\mathcal{T}$ reads $\omega_i, \mathsf{pk}_i^u, \mathsf{cert}_i^u$ and Checks them.
5. $\omega_i$ is used online to access a wallet containing $x$.

[1] If Fiat-Shamir is used, this is not necessary because Fiat-Shamir is identity-based.

### 3.5 Re-Circulating

Upon reception of an unsealed coin $\mathcal{C}_i$, $\mathcal{B}$ uses an administrative symmetric key to erase from $\mathcal{C}_i$ all traces of $\mathsf{sk}_i^s, \mathsf{pk}_i^s, \mathsf{sk}_i^u, \mathsf{pk}_i^u, \mathsf{cert}_i^s, \mathsf{cert}_i^u, \omega_i$.

The coin is then reused as a new coin in the issuance process (Section 3.2).

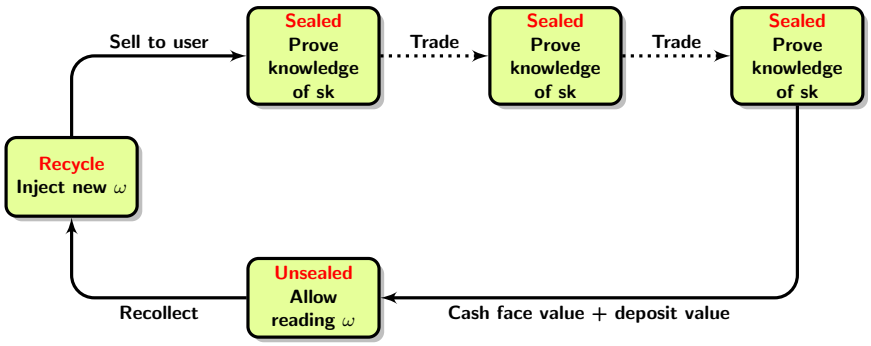


**Fig. 2.** Token Life-Cycle.

## 4 Reducing Confiscation Risks

The (minor) inconvenient of the proposed method is reliance on some $\mathcal{B}$ who may, in theory, confiscate the credentials stored in the coins. Unfortunately, there seems to be no perfect mitigation of this risk given that we deal with material coins whose operation could be simulated in the absence of a proper root of trust.

We nonetheless consider that relying on several $\mathcal{B}$s for the same cryptocurrency (i.e. distributing the risk) and cashing-out amounts at a regular pace will balance anonymity against the very hypothetical risk of a betraying $\mathcal{B}$.

More complex risk mitigation strategies could be used as well. For instance the coin might contain several (e.g. $v = 5$) chips managed $u$ different $\mathcal{B}_i$s. A coin now contains $uv$ independent wallets, each containing $1/(uv)$-th of the total coin value (we call each of those shares a *monetary unit*). During sealing the coins are successively transported from $\mathcal{B}_i$ to $\mathcal{B}_{i+1}$ for sealing. Assuming that $\beta$ chip manufacturers cheat and $\alpha$ banks cheat we still have $uv - \alpha v - \beta u + \alpha\beta$ monetary units that survive.

Assume that the probability that a bank cheats is $p$ and that the probability that a chip manufacturer cheats is $q$. The expectation of the surviving number of monetary units is:

$$\sum_{\beta=0}^{v}\sum_{\alpha=0}^{u} q^{\alpha}p^{\beta}(1-q)^{u-\alpha}(1-p)^{v-\beta}\binom{u}{\alpha}\binom{v}{\beta}(\alpha\beta-\alpha v-\beta u+uv) = (1-q)(1-p)uv$$

## 5 In Conclusion

This paper described a method allowing to increase the fluidity and the anonymity of cryptocurrencies. The core components of the proposed solution are: tamper-resistance, authentication protocols and the partial reliance on issuing authorities.

## References

1. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings. Lecture Notes in Computer Science, vol. 263, pp. 186–194. Springer (1986)
2. Naccache, D., M'Raïhi, D., Vaudenay, S., Raphaeli, D.: Can D.S.A. be improved? — complexity trade-offs with the digital signature standard —. In: De Santis, A. (ed.) Advances in Cryptology — EUROCRYPT'94. pp. 77–85. Springer Berlin Heidelberg, Berlin, Heidelberg (1995)
3. Want, R.: RFID Explained: A Primer on Radio Frequency Identification Technologies. Synthesis lectures on mobile and pervasive computing, Morgan & Claypool Publishers (2006)