Cash: The once and future king

Eduard de Jong*

23rd February 2024

(version 1.3)
Originally published January 31, 2023

Abstract

It's time to bring order to digital money through the proper issuance of Central Bank Digital Currency (**CBDC**). While many financial services moved to the digital realm based on centralised account ledgers, there remains an unmet need for a cash-like, distributed, digital, bearer payment instrument. A hodgepodge of existing proprietary *electronic cash* (e-cash) products have shown this need can be met, but only in fragmented, disconnected walled gardens. Only a central bank can be the issuer of an e-cash payment instrument that brings the traditional benefits of cash to the digital world. Every economy in the world will benefit when its central bank issues a digital currency. A formal model for money as a distributed information system is provided that is validated by showing cash as one of its instantiations. Guided by this model the aggregating receipt token technology is presented as an e-cash implementation that will deliver key benefits of physical cash to the digital world. It supports online and offline payments that can be securely received with a software-only implementation. A money system based on this technology supports both monetary and security management. This aggregating receipt token e-cash solution is the only technology that meets all of the key requirements for **CBDC**: strong protection of privacy, increased financial inclusion, protecting the availability of money as a public good and reduced costs to society of payments.

Keywords: electronic cash, e-cash, digital payment, offline payment, money, digital money, digital token, **CBDC**, formal model.

1 Introduction

In English there is a saying "Cash is King" to express that, if you intend to buy anything, a payment in cash will be key in convincing a seller to accept your offer. The alternative to accepting cash is to give credit. For many centuries giving credit used to be the main way for merchants to retain customers, which was then at regular times settled in cash. To give credit the seller needs to know the buyer and estimate the risk of losing money; with

*The author thanks Peter Cattaneo for his continued support in bringing this paper into its current shape.

cash there is strong confidence in the absence of risk[1]. Hence, payment in cash has for a very long time in human history been the much more preferred way to receive payment. Before the modern age, it also was the only way to pay for customers not local to a merchant. In 2022 cash is still main means of payments in many countries around the world, especially in rural regions.

King Cash

What, then, makes cash king?

1. Cash has a recognisable value, the receiver of cash knows it can spend that same value[2] in the future;
2. Possession of cash means ownership, the owner has full control of when to spend, how much to spend, and who to pay;
3. Payment in cash is instantaneous, every piece of cash, whether coins or banknotes, handed over to a payee changes ownership at that very moment;
4. Cash payments have no fees, anyone paying or being paid with cash only requires the exact amount involved;
5. Payment in cash involves only two parties, it is finished there and then and no other party is directly involved in the transaction;
6. Payment in cash is irrevocable, there is no way to retrieve the value paid except, maybe, much later by starting a legal procedure with uncertain outcome;
7. A value paid in cash has no memory of its previous owner, so a payer is anonymous;
8. The ability to pay is only constrained by the amount of value owned and not, for instance, by system limitations;
9. Knowledge of monetary value owned is exclusive to the owner.

There are also a number of drawbacks of using cash:

a) Storing cash entails the risk of theft, it needs physical and organisational protective measures with a strength appropriate for the amount being stored;
b) An amount of cash can be bulky, which limits the amount than can practically stored or used in payment;
c) Creating a receipt for a cash payment is a separate operation, as is receiving the receipt;
d) Cash facilitates laundering money.

[1] Aglietta and Orléan analyse[1] the various aspects of the trust in money that provides the basis for the perceived minimal risk in cash. Brunton, in the first chapter of his book "Digital cash"[5] argues that a key characteristic of money is precisely the trust in the future use as a value. Complementing these descriptions, in his PhD thesis[26], Walton investigates the perception of trust in money by its users, particularly in digital forms.

[2] As a related but separate issue, the *monetary system* in which cash is being issued and used is concerned with providing trust that the same monetary value carried as cash can at a later time pay for about the same amount of goods or services.

King cash in battle

In a Welsh legend, dating from the early European middle ages, a king, named Arthur, successfully, through a number of battles he won, brought safety, stability and prosperity to parts of what we currently know as England and Wales. Years later, after his death from receiving fatal wounds in a battle against his incestuous bastard son, turbulent and incertain times in these parts resumed. The legend further tells[3] us that King Arthur will, at some time in the future, return to restore order and bring renewed prosperity, returning from a place where his wounds will be magically healed.

In our times, another King is doing battle: it is King Cash, fighting in the War on Cash[14].

During its over two and a half thousand year long reign, King Cash enabled marketplaces around the globe to trade in a wide range of goods, with customers and traders both local and from far away. In this way king Cash brought prosperity to communities large and small.

In the competition with electronic payments enabled by banking cards, a more recent invention of banks, which earlier invented banknotes as a form of cash as a complement to coins, one could argue that King Cash is, like Arthur in the legend, doing battle in this war with an incestuous bastard!

Electronic cash

Starting in the early nineteen nighties many e-cash systems have been deployed successfully, especially for fare payment in public transit systems with multiple transit providers. A transit e-cash system like the Octopus card in Hong Kong additionally supports payments in shops.

Electronic cash (e-cash) is money stored as persistent information in the memory of a personal computing device together with an application to transfer some, or all, of it to another computer. Cryptography is used to protect the value in an e-cash payment.

An e-purse is the IT device that stores an amount of e-cash and provides control of spending to its owner. An e-purse leverages the strong hardware security provided by a smartcard to keep cryptographic keys secret and protect the application program and money information against tampering. The implementation of an e-purse can enhance the functionality and security provided by a smartcard chip, e.g. with BlueTooth communication to communicate with the mobile phone and with tamper detection to enhance security. With BlueTooth support a phone app can then provide a rich user interface, e.g. including planning recurring payments.

A payment in e-cash is a digital interaction between two e-purses where a cryptographic protocol protects the digital transfer of money. An e-cash payment is instantaneous, does not incur a fee to be completed, only involves two parties, and cannot be revoked: E-cash has the cash properties 3, 4, 5 and 6, respectively.

The portability of an e-purse, the peer–to–peer nature of a payments and the control of its owner over spending effectively makes e-cash a bearer payment instrument.

[3] In 1958 T.H. White published a retelling of this legend under the title "*Arthur: the once and future king*[27]"

The main cash property that e-cash never can provide is the physical experience of seeing and touching bank notes or coins that allows them to be recognized as money property 1; it can, however, *emulate* some of that experience, e.g. a payment made by holding the e-purse, against another device that accepts it, supported by branding of the e-purse for issuing central bank.

Support for payer privacy in e-cash, cash property 7, depends on the details of the cryptographic protocol used in payments.

Support for cash property 8 (limits on amounts) in an e-cash system is affected by the need to actively maintain its security. As detailed in the 1996 report published by the BIS[4], managing system security in an e-cash system requires to set limits on both the amount held in an e-purse and used in a single payment. In practice these *security* limits[4] will not be noticeable by a large majority of users. Hence, for these users cash property 8 can be fully supported.

An e-cash implementation can reduce the risk of theft, cash drawback a), by implementing payment authorization in the e-purse hardware.

Aggregating receipt token technology

Aggregating receipt token[5] (**ART**) technology[6] is a way to implement offline, transferrable e-cash, which effectively has all the nine properties of Plain Old Cash listed above, including payer privacy, as well as greatly reducing the four drawbacks mentioned. Aggregating receipt token technology enables a secure peer–to–peer e-cash payment infrastructure with strong control by the user of spending. An **ART** technology e-cash system can neatly be integrated with the existing monetary system, e.g. with banks by operating an e-purse to accept deposits into an account and to make payments as withdrawal from it. A user of an **ART** technology e-purse can receive payments from any other user and does not need to have a bank account.

The **ART** technology supports the BIS security management recommendations (c.f. [4]), with auditing of pseudonymous transaction logs and the dynamic configuration of the security limits on amounts stored and paid.

An **ART** technology e-cash system supports *income transparency*[7] as basis for conforming to Anti Money Laundering (**AML**) regulation.

[4]The security limits on holding and spending are based on the (conservatively) estimated strength of the security provided by the secure hardware in the e-purse, and the estimated risk of a security breach of a single e-purse. Stronger e-purse implementations could be available for users that require to hold and pay larger amounts.

[5]This paper uses the word *'token'* in its traditional meaning as an object of intrinsic value for use in a specific environment; it is a generic term for a coin. In the digital domain a token is a digital *'object'* that has a monetary value. An aggregating token is digital; it is a message secured by cryptography and specially constructed for use in a electronic cash systems. A more recent use of the term *'token'* is in the context of blockchain technology. In that context a token is a digital reference of information that has been recorded in one of the blocks with the recorded information representing something of value.

[6]The concept of an aggregating receipt token has been conceived by Chris Stanford in 1994. In 1996 a collaboration with the author resulted in an awarded patent[19].

[7]The term *income transparency* first appears in David Chaum et.al [9] to describe a feature of a digital payment system, in this case using anonymous *coins*, whereby a user cannot hide any moneys received, at least when challenged by authorities. The concept, without using the term, is also presented by the author in his proposal[16] for a low-level modification of blockchain protocols to enable enforcing regulations.

Reusable, payee–bound payment tokens as e-cash have been proposed, among others, by Manasse[22] as Millicent and by Rivest and Shamir[25] as PayWord. However, none of these e-cash methods provide reusability of the same token with different payers as **ART** technology does.

Central bank digital currency (CBDC)

Over the past centuries the technology for money and payment has continuously further developed: there was the introduction of banks as way to share the costs of safely storing large amounts of moneys; the invention of checks to make payments against stored money; the invention of banknotes that lead to a race in bank note printing technology to stay ahead of counterfeiting; the invention of the credit card, the debit card, and debit and credit cards as smartcards to enhance security in payments; the emergence of crypto "currencies"; the ubiquity of online banking. **CBDC** is the next step in this development.

As stated by Christine Lagarde and Fabio Panetta, president and member of the board of the European Central Bank (**ECB**), respectively, in a blog post[21], "*A digital Euro would complement cash—not replace it—by allowing central bank money to also be used in digital form.*"

The main purposes of **CBDC** is to protect *money as a public good* against encroachment from large IT companies like Apple, Google and Meta and to reduce friction in payments by reducing the costs of payments to society. To be succesful **CBDC** must be very convenient to use and widely accepted as payment instrument; payment from person to persom, from customer to merchants, form business to business, consumer and government, and from governments to all. Payment in **CBDC** must be possible both offline and online and it must protect the privacy of the payer: **CBDC** must be "*cash–like*[8]."

Formal modeling

Formal modelling is a specialised discipline in software engineering that aims to establish trust that software does what it is intended to do. Formal modelling can also be used to show that software cannot misbehave when it responds to faulty inputs like from an attacker. It uses mathematical proof techniques to show that a program actually responds to input and does so correctly. It can also show that implementation choices in program code are consistent with the purpose of that program.

A formal model starts with a very abstract description of the system that needs to be implemented, with mathematical formulas to precisely describe the key system features. The formulas express how a system evolves over time as it responds to repeated varying input data; they specify conditions for internal values that must be maintained at all times, the system *constraints*.

An actual e-cash system, as an implementation of *money*, is expected to operate flawlessly as any glitch could lead to reducing user trust in the system even when actual financial damage is absent or very small.

[8] An overview of requirements for **CBDC** is given by Bindseil et al. in [3, p. 30]. The importance of **CBDC** as cash-like for its adoption is mentioned in a report by the ECB [28, p. 9] and other reports on **CBDC** requirements, e.g. [8, 12].

A number of researches has applied formal modelling to e-cash. The European Central Bank presented a formalised model of electronic money security objectives[11]. Gouda and Liu use a common formal model[13] applied to several e-cash message protocols to verify that these protocols are secure against message loss, modification and replay attacks. Butler and Yadev analyse in [6] how the Mondex e-cash system[23] detects loss of value during payment. Inega, Oyama and Yasuura in [15] develop a model to compare two different e-cash protocols; they establish fundamental differences in communication and security properties between the protocols. The previous formal model work has had a narrow focus on the electronic payment operation. This paper presents the basis for a model of a more comprehensive digital money system.

To tackle the complexity of typical system a formal model is developed in a number of small steps, where each step adds a single feature or design choice. The result of a modelling step is a *refinement* of the model. The purpose of a refinement it to get a formal specification for a more functional system or closer to the actual implementation. A modelling step precisely specifies the new feature with additional formulas. A new feature typically affect a few of the earlier formulas, which are replaced by one or more other formulas.

The art of formal modelling is to find the right model to start with and to find the sequence of refinements that after many steps can easily be implemented in program code. Modelling tools support this process by generating mathematical proofs that a refinement meets all the constraints of the basis model.

The formal model of money presented in this paper shows that with **ART** technology e-cash can properly implement **CBDC**.

Return of the king

This paper is organised as follows: The next section develops a formal model of an information system that implements money and identifies two fundamentally different ways for further refinements towards actual implementations[9]. That section also discusses three different ways to implement e-cash. Section 3 describes the use of **ART** technology, one of these three options, to implement e-cash. It extends the formal model of money from the previous section into a formal model for the implementation of aggregating receipt token e-cash. The final, concluding section shows that, issued as **CBDC** based on aggregating receipt token e-cash technology, like King Arthur of the legend,

King Cash will return!

2 Modeling money

Formal model 1 specifies money as an information system with a set of numbers and a set of users. The numbers in this abstract model are "*balances*," indicated as $\mathbb{B}$, that each represent something we can call an amount of money. Each balance is associated with a user u by labeling it, as in $\mathbb{B}_u$. The

[9] A concrete implementation of money can be seen as a distinct *type of money.*

model presents constraints to be obeyed as the system evolves, which happens when applying the operation that is also presented. How a user can technically be associated with its balance $\mathbb{B}$ is not further specified here; that association is something that a concrete implementation of a money *system* will address.

The text style used for the symbol $\mathbb{B}$ is an indication that its value should be considered as persistent and protected; other symbols that will be introduced with the same text style also will be persistent and protected. Implementations of money will differ in how to achieve these data storage properties.

Model 1: Money

$$\sum_{\text{users}} \mathbb{B}_u = C \tag{1}$$

$$\forall u : \mathbb{B}_u \geq 0 \tag{2}$$

$$\mathscr{P}(a, \mathbb{B}_{\text{Rob}}, \mathbb{B}_{\text{Eve}}) =: \mathbb{B}'_{\text{Rob}} = \mathbb{B}_{\text{Rob}} - a;\ \mathbb{B}'_{\text{Eve}} = \mathbb{B}_{\text{Eve}} + a \tag{3}$$

Formula 1 in model 1 shows a key property of a money system: the total amount of money is constant. That property means that no money can disappear and that money can't be created out of thin air. Formula 2 specifies that any balance $\mathbb{B}_u$ is either zero or positive. The operation, defined by formula 3, specifies how the system can evolve by moving money from one user to another. That operation is indicated with the symbol $\mathscr{P}$, for *payment*.

Payment is the only operation a money system needs to support. The definition of the operation $\mathscr{P}$ shows that a change in the value of a balance $\mathbb{B}$ is done pair–wise: $\mathscr{P}$ affects two balances, $\mathbb{B}_{\text{Rob}}$ and $\mathbb{B}_{\text{Eve}}$, where Rob is the identier for the payer and Eve is the identifier for the payee. The change in both balances is by an amount a, with the payer balance value $\mathbb{B}_{\text{Rob}}$ being reduced by a into a new value $\mathbb{B}'_{\text{Rob}}$ and the payee balance $\mathbb{B}_{\text{Eve}}$ being incremented by a into a new value $\mathbb{B}'_{\text{Eve}}$[10].

A balance $\mathbb{B}_u$ thus represents the amount of money a user u has available to spend and formula 3 shows spending money. The three formulas in model 1 completely specify the behavior of a money system.

A first refinement of model 1 is needed to address the issue that the constraints in it imply that a payment can only be with an amount $a = 0$, which may be formally correct but is not very useful. To blow life into the system, a specific *user* is added to the system, the *issuer*.

Refinement 2 shows the issuing of money by the issuer. As specified by formula 4, at a certain point in time, $t = 0$, the issuer balance $\mathbb{B}_{\text{Issuer}}$ is set to a specific value, $C_0 > 0$. After the issuer balance has been set to this value, the issuer can start making non–zero payments to other users, which subsequently can make useful payments to further users, enabling useful payments among all the other users. The consequence of issuing money is also reflected in refinement 2 by replacing formula 1 by two new ones, formula 5 for the total amount before issuing money, 0, and formula 6 with the issued amount C_0 as the new total.

[10] By convention the tick in $\mathbb{B}'$ indicates that after the computations in the formula have been done the computed value replaces the old one, $\mathbb{B}' \Rightarrow \mathbb{B}$

Refinement 2: Money issuer

$$\forall u : \mathbb{B}_u \geq 0 \tag{2}$$

$$\mathscr{P}(a, \mathbb{B}_{\text{Rob}}, \mathbb{B}_{\text{Eve}}) =: \mathbb{B}'_{\text{Rob}} = \mathbb{B}_{\text{Rob}} - a;\ \mathbb{B}'_{\text{Eve}} = \mathbb{B}_{\text{Eve}} + a \tag{3}$$

$$\mathbb{B}^{t=0}_{\text{Issuer}} = C_0,\ C_0 > 0 \tag{4}$$

$$\sum_{\text{users}} \mathbb{B}^{t<0}_u = 0 \tag{5}$$

$$\sum_{\text{users}} \mathbb{B}^{t\geq 0}_u = C_0 \tag{6}$$

Together, the formulas in refinement 2 specify a minimal, complete and useful money system. This lack of specificity can be leveraged in a further refinement that shows how the issuer can change the amount of issued money multiple times without violating the constraints in model 1. In the present context such a refinement is not needed.

Refinement 2 also shows that the introduction of an issuer in the system main result is that the constant C is greater than 0. To prevent cluttering their presentation with issuer details, further refinements will be directly based on model 1 adding the condition $C > 0$ as sufficient and necessary to make it be alive.

As a second, separate refinement step for the abstract model 1 of money, the concept of *ownership* is introduced: refinement 3 specifies that a user in the system, as an owner, has exclusive control over how much to pay and to whom to pay. Ownership gives the *meaning of* ***money*** to the information represented by the persistent, tamperproof number $\mathbb{B}_u$.

Ownership is reflected in formulas 8 and 9. In formula 8 a user specific *authorisation* operation is introduced with the symbol $\mathscr{A}$. With the operation $\mathscr{A}_u$ the user u determines for an amount a that a payment to another user u' should be done, or not. Equation 3 is rewritten in this model as formula 9, specifying that the payer is explicitly involved in the payment and that the payment only can proceed if the authorisation for the payment is given.

Refinement 3: Money with ownership

$$\forall u : \mathbb{B}_u \geq 0 \tag{2}$$

$$\sum_{\text{users}} \mathbb{B}_u = C, C > 0 \tag{7}$$

$$\mathscr{A}_u(a, u') : \{\text{true}, \text{false}\} \tag{8}$$

$$\mathscr{P}(a, B_{\text{Rob}}, B_{\text{Eve}}) =: \text{if } \mathscr{A}_{\text{Rob}}(a, \text{Eve}) \Rightarrow \mathbb{B}'_{\text{Rob}} = \mathbb{B}_{\text{Rob}} - a;\ \mathbb{B}'_{\text{Eve}} = \mathbb{B}_{\text{Eve}} + a \tag{9}$$

Further refinement steps could be, for instance, with details on the payment operation $\mathscr{P}$ and authorisation operation $\mathscr{A}$. Details can also be specified on how to protect the number $\mathbb{B}_u$ stored in a persistent computer memory against tampering.

At this point in the refinement process we find that there are two fundamentally different ways to implement a money system, implying fundamentally different ways for further refinements:

- As a *central* system using ledgers to record all the balances $\mathbb{B}_u$, with ledger management software providing the assurance that in an update to the records the system invariants (formulas 2, 7) are respected and the authorisation $\mathscr{A}_u(a, \text{Eve})$ is performed. An example of centralised money system is a bank providing a debit card payment function. In a ledger system, implementing authorisation also requires recording user related data, such as the details of the user's payment card.
- As a *distributed* system, where each user has its own private store for its balance $\mathbb{B}_u$ and privately performs the authorisation $\mathscr{A}_u$ to allow a payment to another user to proceed. An example of distributed money system is cash, i.e. banknotes and coins.

In refining the formal model to specify the implementation of a money system, authorisation can be treated as a subsystem with its own abstract model and refinements. The refinements for an authorisation subsystem in a distributed money system, with a secure device operated by a user will be very different from the refinements needed for a central money system. A central money system, for instance, needs refinements to describe i) a way to recognise the owner of a balance among the set of all users, ii) a way the determine the intent by the owner to make a payment to another specific user, and iii) presenting to the user the correct identity information on the intended payee[11].

As mentioned, Plain Old Cash is an example of a distributed implementation of the money system specified in refinement 3. In the implementation as cash each balance $\mathbb{B}$ is represented by a collection of physical objects, coins and banknotes with various denominations[12]. Ownership is *'implemented'* by a physical container under control of the user, that holds the objects that represent the monetary value. The value of $\mathbb{B}$ is tamper resistent by the owner's physical container and it is tamper detecting by the way a banknote or coin are manufactured. Payment is authorised by payer Rob by removing coins and banknotes representing the amount to pay a from the owner's container, which implements $\mathbb{B}'_{\text{Rob}} = \mathbb{B}_{\text{Rob}} - a$ in formula 9. Payment is completed by payee Eve receiving the objects removed by Rob from his container and adding these to her container implementing $\mathbb{B}'_{\text{Eve}} = \mathbb{B}_{\text{Eve}} + a$ in that formula. Dreier et al.[10, pp. 3, 4] also present Plain Old Cash as matching a formal model of digital payment.

This paper focuses on another implementation of a distributed money system, one that uses digital technology, i.e. the implementation of e-cash. In an e-cash system the *balance* $\mathbb{B}_u$ is stored in an e-purse, a dedicated, secure personal computer. Historically, e-purses have mostly been implemented as smartcards with the balance $\mathbb{B}_u$, represented by digital information stored in the card chip's persistent memory. In addition to the balance, the e-purse stores a program to implement the payment operation together with secret keys to cryptographically secure the messages to synchronise the updating of both balances, $\mathbb{B}_{\text{Rob}}, \mathbb{B}_{\text{Eve}}$.

[11] In the example above of a bank supporting a debit card payment, identification of the payee is provided by a secure payment terminal which is registered for that purpose.

[12] To be formal, the numeric value of $\mathbb{B}$ is encoded using a mixed radix system (c.f. Donald Knuth in [20, p. 209]) with the denomination value of a money object as unit value for each radix. A number in a particular radix is represented by the number of coins or banknotes for that unit value.

Diagram 1: E-cash value transfer

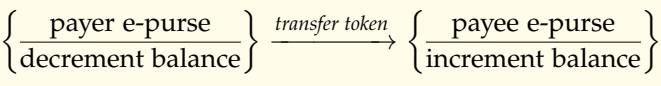


Diagram 1 describes the implementation of formula 3, payment, in a distributed digital money system implementation as a peer-to-peer interaction between two devices, the two e-purses, that can act either as payer or as payee. The payer e-purse sends a message, labeled *transfer token*[13], that is received by a payee as specified in a particular payment protocol implemented by the e-purses. The diagram also summarises the operations these devices perform before and after sending and receiving, respectively, the transfer token, which implement the two balance computations specified in formula 9.

The transfer token represents the amount a in the payment and it typically also contains data to assure that the balances in both devices are updated in accordance with the system invariants.

A practical implementation of diagram 1 contains at least one extra message, a payment request, send by the payee to specify the payment amount a. In some implementations additional messages will be needed, for instance, if a multi–step cryptographic protocol is used to secure the transfer token or to prevent loss of value due to communication loss.

The e-purse can provide the owner with a PIN, password, passphrase or biometric to implement the user specific authorisation function $\mathscr{A}_u$. The payment authorisation may be configured to always allow small payments or require authorisation for an aggregate total of small payments. It may use a user interface to present the amount to pay and the name of the payee.

There are three options to implement e-cash, that can be characterised by the form of the transfer token used:

- ⇒ *Authenticated-device transfer*, where both the payer and the payee need to use secure devices and the transfer token is a secure message containing the payment value in a multi-message cryptographic protocol to authenticate both devices;
- ⇒ One or more *digital coins*, each a distinct token[14], generated by a specific trusted system component, the issuer, and both the payer and payee can use any computer;
- ⇒ *Aggregating receipt token technology*, the transfer token is an authenticated message, an *aggregating receipt token*, generated by a third party for a particular payee, reusable to receive multiple payments from different payers, the payer uses a secure device and the payee can use any computer.

In addition to implementing the transfer token to securely move monetary value, an implementation of an e-cash system needs to adress a range of operational and security issues as described by the author in [17]. These issues include payer privacy, loss of value due to loss of communication and

[13]The word "*token*" in *transfer token* emphasises that this message between the two devices is secured, that it can't be tampered with.

[14]This paper uses the word *coin* as reference to a *token* that has a pre–arranged, fixed monetary value. A digital coin is usually generated to be anonymous.

double spending by copying data. Payer privacy-by-default is only provided by digital coins and **ART** technology. In these systems a payment does not reveal any data that could identify the payer.

With e-cash the balance of spendable money can be carried around[15] by the payer. However, it depends on the type of transfer token implemented if an e-purse can fully realise an e-cash *bearer payment instrument* with suport for the key properties of i) a payment only involving two parties, ii) immediate finality of **all** payments and iii) absence of per-payment costs.

With an authenticated-device transfer the implementation addresses the issue of loss of value by loss of communication by recording details of each attempted and completed payment in both e-purses. This solution requires the e-purses to regularly communicate with the issuer to upload the recorded transaction details for analysis to detect non-completed payments. Compensation for a detected loss of value requires a bank account linked with the owner of the e-purse. Because of these issues, authenticated-device transfercan not meet all of the requirements of a cash-like bearer payment instrument.

Digital coin e-cash can effectively only be used for an online setting as the issuer needs to be involved to detect double spending of a coin. Digital coins provide strong privacy protection of both users, which could prevent an effective implementation of **AML** measures[16]. As digital coins can be easily copied, implementations an e-cash system gets complicated by the need to add mechanisms to transfer ownership while copying data from payer to payee and to prevent theft.

Offline use of e-cash with digital coins has been proposed by using specially constructed coins, either a coin for one-time use or a coin with restricted multiple use, a so called "*transferrable coin*[17]." However, prevention of double spending in limited-use coins requires that the issuer inspects it after it has been deposited by the payee. A transferable coin must be be deposited after a specified number of transfers, or on expiry of its validity period. With offline use, the double spending protection relies on the availability of an identity management infrastructure to identify, and possibly apprehend, the payer that did spend a coin twice. All digital coin systems presume that each user has a bank account to debet on withdrawing a coin and banks implicitly provide the needed identity infrastructure.

Adding a smartcard to a digital coin e-cash system can enhances its security in offline payments. A smart card can also add stronger ownership control to digital coin payments. However, with a smartcard as key component in the system, the key benefit of digital coins, strong payer privacy in combination with *software only acceptance of payments,* is much harder to realize.

[15] An e-purse that is configured to store larger amounts can conceivably be realised as a device that is hard to steal, e.g. it could be fixed to a building or be too heavy and bulky to be easily carried around. Such an e-purse would be the digital equivalent of a wall safe.

[16] As mentioned above, a recent proposal by Chaum et al. [9] supports income transparency to enable **AML** in the Taler payment system that uses digital coins.

[17] Transferrable coins are still an area of active cryptographic research. In 2020 Bauer et.al claim, in [2], to present the design of a **first** practical version of a secure transferable coins that also provides a good level of (privacy) protection of the users. No complete implementation exists today of this design.

The remainder of this paper describes e-cash based on **ART** technology that combines the best properties from both alternatives.

3 E-cash with an aggregating receipt token

An *aggregating receipt token* is specifically created for use by a particular payee to be reused in receiving money in multiple payments from, possibly, different payers. In each subsequent payment where an aggregating receipt token is used again, its value is increased by the payment amount. An aggregating receipt token is configured to be used for a specific number of times and up to a specific aggregate value. A user obtains aggregating receipt tokens from a dedicated service provider, one of possibly many "*token factories*" that support the e-cash system.

Notably, a token factory is not an issuer of money as the value of a created token is 0. It is a part of the security supporting infrastructure of an e-cash system based on **ART** technology. As the token factory creates a new **ART**, it includes data to specify a usage configuration for the token and it uses a secret signing key to protect the new token against tampering. The token control information supports managing monetary and security aspects of the e-cash system, which can be different for specific (groups of) users. The token control information can include an expiry date, a maximum amount of a single payment or the maximum amount of payments to be collected in the token. The user information needed to determine the token configuration is stored in the e-purse, which provides this information to the token factory in a digitally signed request for one or more new tokens.

With **ART** technology the balance of money owned by a user is split into two parts:

i) a *spendable* part, digitally stored as a number in a secure device
ii) a *redeemable* part consisting of a number of **ART**s stored in the memory of the payee's computer.

The spendable part of the e-purse balance is similar to the way an e-cash balance is stored in a system using an authenticated-device transfer. The redeemable part of the balance resembles the balance for a system using digital coins. To receive money an **ART** is used; to spend money the secure device is used. The e-purse in **ART** technology now consists of two parts, a secure device and complementary software in a computer operated by the user. The spendable balance is protected by the secure hardware, and the redeemable balance is protected against tampering and theft with cryptography.

A complete payment with an **ART**, from requesting a payment to spending the received money, is done in two different operations that both involve a single secure device interacting with the software: i) In *payment* the e-purse hardware of the payer receives a payment request with an **ART** from the software in the payee's e-purse and ii) In *redemption* the software in an e-purse communicates with its own hardware part to make money received with an **ART** available to be spendable in future payments.

Payer Rob uses the secure device in his e-purse to process the payment request he received from Eve's software. The payment request consists of Eve's **ART** and the amount to be paid. After validating Eve's **ART**, the e-

purse computes the payment operation

$$\text{if } \mathscr{A}_{\text{Rob}}(a, \text{Eve}) \Rightarrow \mathbb{B}'_{\text{Rob}} = \mathbb{B}_{\text{Rob}} - a;\ \mathbb{T}'_{\text{Eve}} = \mathbb{T}_{\text{Eve}} + a$$

Where $\mathbb{T}_{\text{Eve}}$ refers to the received aggregating token and $\mathbb{T}'_{\text{Eve}}$ to the result of the payment. The cryptographic protection in the updated aggregating receipt token, $\mathbb{T}'$, is computed by the secure hardware in the e-purse using a secret payment key. The payment key is certified by the issuer using a blind digital signature algorithm to provide payer anonymity.

Payee Eve does not need the secure device of her e-purse to accept e-cash. The e-purse software in her computer does that: It first validates the **ART** received as payment result, then checks that the amount added to its value is the expected payment. An accepted **ART** is put back in persistent memory, to be reused or redeemed at a suitable future time. For instance, if payee Eve is a webshop and her computer is a (cloud) server, an **ART** could be put back in a queue for immediate reuse to accept a payment by the next online customer.

Figure 1 shows how a payee first obtains an aggregating receipt token, as T_0 from a token factory and then uses it to receive a series of payments from a number of payers, which can possibly be the same. The token T_0 obtained from the token factory contains the token configuration, its value is zero. Token T_0 is contained in the most recent token value T_i and each e-purse can validate that the received token has been used in the previous payment conforming to its configuration. The e-purse also determines that the current payment request also matches the configuration. After accepting the aggregating receipt token each of the paying e-purses returns it with a new higher value computed from the previous value and the amount requested as payment, $T_0 \rightarrow T_1$, $T_1 \rightarrow T_2$ etc. until the last shown payment result $T_{n-1} \rightarrow T_n$. Figure 1 shows that the token after having been used in a number of payments, is returned to the token factory including as the full set of received tokens: $T_{0\cdots n}$.

Eve's e-purse software implements a function to keep the collection of stored aggregating receipt tokens up to date. When needed, the software sends a request to the token factory and involves the secure part of Eve's e-purse to provide a cryptographic identifier for her to be included in the new token. Eve's identifier in her e-purse will later be used to validate the token during redemption. The e-purse digitally signs the request for an **ART** to prove that the request comes form a genuine user.

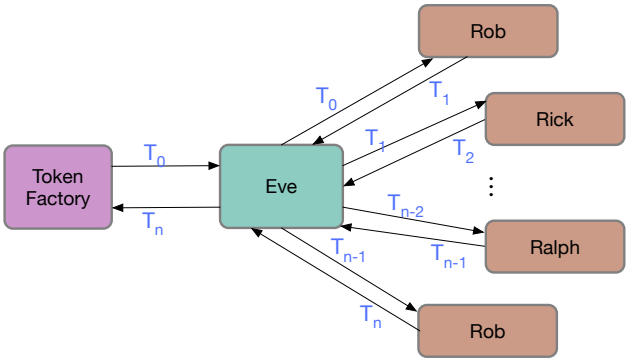


Figure 1: An aggregating receipt token receiving payments.

Redemption of the token value with the e-purse is not shown in fig. 1, it could have been done after any of the payments or just before, or even after,

returning the token to the factory. Also not shown is that returning an **ART** is typically done in a request for one or more new tokens.

Monitoring system security is the primary purpose of analysing returned tokens. With this analysis and adaptable configuration of the e-purse and **ART**s, e-cash can support the comprehensive security management framework recommended in 1996 by the Bank for International Settlement in a report[4] on the security of an e-cash system.

An e-cash system with aggregating receipt token can support Anti Money Laundering (**AML**) by providing *income transparency*[18] via an additional analysis of expired tokens that have been returned by the user when requesting new tokens. As the identifier in an aggregating receipt token is pseudonymous, first–level AML monitoring can be done without needing the full identity of the payee.

Aggregating receipt token e-cash as a money system

This section extends the formal model of money developed in section 2 to show that payment with aggregating receipt token e-cash can be implemented as a security–critical software component that is performed completely within the secure device of a payer's e-purse. In this implementation loss of communication can be addressed by repeating the payment request as the computation is idempotent. A less mathematically inclined reader can skip this section.

Based on the formal model of a money system developed in section 2 (model 1) the implementation of an aggregating receipt token e-cash system can be specified as further refinements based on refinement 3.

Refinement 4 shows a first refinement step of introducing an aggregating receipt token in formal money system as a split of the balance of e-cash into two parts, one used to make payments and the other to receive payments. In refinement 4 equation 10 specifies that each user, in addition to its balance $\mathbb{B}$, has an aggregating receipt token, $\mathbb{T}$. The token $\mathbb{T}$ also has a value that is positive or zero. Formula 11 specifies that all the money in the system is held either in balances $\mathbb{B}$ or in **ART**s, $\mathbb{T}$. Equation 12 is an updated version of 9 that shows that payment in this model involves the balance of the payer, $\mathbb{B}_{\text{Rob}}$, and the aggregating token of the payee, $\mathbb{T}_{\text{Eve}}$.

The *redemption* of the **ART** for a specific user is indicated by the symbol $\mathscr{R}_u$ and defined in formula 13. It shows that in redemption the value collected in one or more payments with its aggregating token $\mathbb{T}_u$ is transferred to the balance $\mathbb{B}_u$. After redemption the value of the token T_u is set to 0, ready for reuse in receiving one or more future payments for Eve.

[18] The term income transparency is introduced by Chaum et al. in [9]. In [16] the author applies the concept of income transparency in a proposal to modify implementations of crypto currency systems in support of oversight and regulations.

Refinement 4: Money with an aggregating receipt token

$$\forall u : \mathbb{B}_u \geq 0 \tag{2}$$
$$\mathscr{A}_u(a, u') : \{\text{true}, \text{false}\} \tag{8}$$
$$\forall u : \mathbb{T}_u \geq 0 \tag{10}$$
$$\sum_{\text{users}} \mathbb{B}_u + \sum_{\text{users}} \mathbb{T}_u = C, C > 0 \tag{11}$$
$$\mathscr{P}(a, \mathbb{B}_{\text{Rob}}, \mathbb{T}_{\text{Eve}}) =: \text{if } \mathscr{A}_{\text{Rob}}(a, \text{Eve}) \Rightarrow \mathbb{B}'_{\text{Rob}} = \mathbb{B}_{\text{Rob}} - a;\ \mathbb{T}'_{\text{Eve}} = \mathbb{T}_{\text{Eve}} + a \tag{12}$$
$$\mathscr{R}(\mathbb{B}_u, \mathbb{T}_u) =: \mathbb{B}'_u = \mathbb{B}_u + \mathbb{T}_u; \mathbb{T}'_u = 0 \tag{13}$$

Refinement 4 shows that the **ART** is modified twice, first in payment (formula 12) and then in redemption (formula 13). In further refinements to formal modeling the protection of an aggregating receipt token against tampering it will be helpful if an **ART** is only modified in only a single computation. Refinement 5 shows a refinement with the changes in the model to accomplishes that the aggregating receipt token is only modified when it receives a payment value. The change affects the redemption operation $\mathscr{R}$. It introduces a copy $\mathbb{T}^{\dagger}$ of the redeemed aggregating receipt token in the e-purse memory to remember the token value in the last redemption that is to be subtracted from the token value in the next redemption. With these changes the value of an aggregating receipt token is monotonously increasing, a nice property to have when validating an aggregating receipt token during use as a correctly composed, cryptographically protected chain of payment records.

Refinement 5: Monotonously aggregating receipt token

$$\forall u : \mathbb{B}_u \geq 0 \tag{2}$$
$$\mathscr{A}_u(a, u') : \{\text{true}, \text{false}\} \tag{8}$$
$$\forall u : \mathbb{T}_u \geq 0 \tag{10}$$
$$\sum_{\text{users}} \mathbb{B}_u + \sum_{\text{users}} \mathbb{T}_u = C, C > 0 \tag{11}$$
$$\mathscr{P}(a, \mathbb{B}_{\text{Rob}}, \mathbb{T}_{\text{Eve}}) =: \text{if } \mathscr{A}_{\text{Rob}}(a, \text{Eve}) \Rightarrow \mathbb{B}'_{\text{Rob}} = \mathbb{B}_{\text{Rob}} - a;\ \mathbb{T}'_{\text{Eve}} = \mathbb{T}_{\text{Eve}} + a \tag{12}$$
$$\forall u : \mathbb{T}^{\dagger}_u \geq 0 \tag{14}$$
$$\mathscr{R}(\mathbb{B}_u, \mathbb{T}_u) =: \mathbb{B}'_u = \mathbb{B}_u + \mathbb{T}_u - \mathbb{T}^{\dagger}_u; \mathbb{T}^{\dagger\prime}_u = \mathbb{T}_u \tag{15}$$

The next refinement step of the formal money model toward a specification of an implementation for an aggregating receipt token e-cash system is shown in refinement 6. It introduces a copy of Eve's **ART** in the persistent memory of Rob's e-purse and modifies the payment operation $\mathscr{P}$. Formula 17 introduces a user specific function to compute the new value of the payee's aggregating token $\mathbb{T}$ during a payment of an amount a. The last step of function $\texttt{atxr}_u(\cdot)$ stores it's result in the token $\mathbb{T}^{\ddagger}_u$, which is introduced in formula

16. Like $\mathbb{T}^{\dagger}$, $\mathbb{T}^{\ddagger}$ is a copy of a token, so they are not include in formula 11.

In a payment with an aggregating receipt token by payer Rob to payee Eve the function $\texttt{atxr}_{\text{Rob}}$ is used to compute, and persistently store, the payment result. Formula 18 shows this, it is based on formula 12. After this computation the stored result $\mathbb{T}^{\ddagger}_{\text{Rob}}$ is copied as the payment result $\mathbb{T}'_{\text{Eve}}$.

Refinement 6: Atomic payment

$$\forall u : \mathbb{B}_u \geq 0 \tag{2}$$
$$\mathscr{A}_u(a, u') : \{\text{true}, \text{false}\} \tag{8}$$
$$\forall u : \mathbb{T}_u \geq 0 \tag{10}$$
$$\sum_{\text{users}} \mathbb{B}_u + \sum_{\text{users}} \mathbb{T}_u = C, C > 0 \tag{11}$$
$$\forall u : \mathbb{T}^{\dagger}_u \geq 0 \tag{14}$$
$$\mathscr{R}(\mathbb{B}_u, \mathbb{T}_u) =: \mathbb{B}'_u = \mathbb{B}_u + \mathbb{T}_u - \mathbb{T}^{\dagger}_u; \mathbb{T}^{\dagger\prime}_u = \mathbb{T}_u \tag{15}$$
$$\forall u : \mathbb{T}^{\ddagger}_u \geq 0 \tag{16}$$
$$\texttt{atxr}_u(a, \mathbb{T}) =: \mathbb{B}'_u = \mathbb{B}_u - a;\ \mathbb{T}^{\ddagger}_u = \mathbb{T} + a \tag{17}$$
$$\begin{aligned} \mathscr{P}(a, \mathbb{B}_{\text{Rob}}, \mathbb{T}_{\text{Eve}}) =: \text{if } & \mathscr{A}_{\text{Rob}}(a, \text{Eve}) \Rightarrow \\ & \texttt{atxr}_{\text{Rob}}(a, \mathbb{T}_{\text{Eve}}); \\ & \mathbb{T}'_{\text{Eve}} = \mathbb{T}^{\ddagger}_{\text{Rob}} \end{aligned} \tag{18}$$

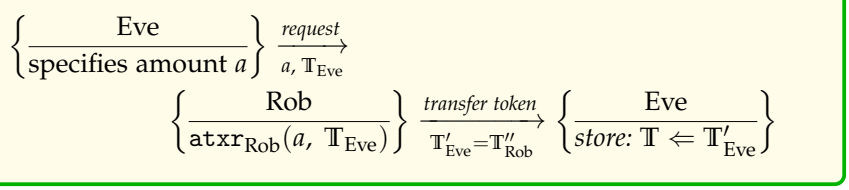
The e-purse of user u can implement the function $\texttt{atxr}(a, \mathbb{T})$ to be atomic, i.e. to make sure that the computed values, token $\mathbb{T}^{\ddagger}$ and the new balance value $\mathbb{B}'$, are only stored persistently if the computation has not been interrupted. A secure device can be programmed to support atomic updates of its persistent memory. How this could be done is shown by Butler et al.[7] with a formal model[19] of the implementation of a so–called transacted memory[20].

Comparing formula 18 with diagram 1 shows that the assignment $\mathbb{T}'_{\text{Eve}} = \mathbb{T}^{\ddagger}_{\text{Rob}}$ corresponds with sending the payment result $\mathbb{T}^{\ddagger}_{\text{Rob}}$ as *transfer token* from Rob's e-purse to Eve's computer, where its is received as $\mathbb{T}'_{\text{Eve}}$, the new value of her aggregating token, ready for storage and later reuse or redemption. Diagram 2 shows this. The diagram also shows clearly that, once the input a and $\mathbb{T}_{\text{Eve}}$ has been received in the payment request, the computation $\texttt{atxr}$ can be computed by Rob's e-purse as it only updates data local to that computer.

[19] This model was based on a patent[18] issued to Jurjen Bos and the author in 2000.

[20] In a transacted memory a write operation is guaranteed to write all data that has been specified to written as a unit. This means that the persistent memory is always in a consistent state.

Diagram 2: Aggregating receipt token value transfer



The refinements use a single aggregating token per user; as a user can only use a single token each time it receives a payment the model actually, implicitly supports multiple tokens. In further refinements, where details on managing tokens are introduced it will be needed to make the use of multiple tokens explicit.

Further formal model refinements can further be derived to drill down on many important implementation details, including cryptographic security enforcing functions, in the e-purse and security supporting operations like a token factory.

E-cash versus cash and accounts

Payment with e-cash is a digital payment and can be done both offline and online, whereas traditional cash only can be offline. In addition, in reviewing the four disadvantages of cash presented on page 2 it becomes clear that e-cash can be better than Plain Old Cash: i) The size of the e-purse if fixed[21], it can store any amount in the same size[22] (item b); ii) The payee token computed as result of payment is stored in the e-purse implicitly acting as a receipt, including a payee reference for the payment in the request message and storing that reference with the computed token enhances the receipt (item c); iii) e-cash cannot be stolen, the e-purse requires owner authorisation and an aggregating receipt token can only be redeemed at the owner's e-purse (item a) and iv) Aggregating receipt token technology e-cash supports **AML** so it isn't very suitable for laundering money (item d).

A distributed peer-to-peer system e-cash based on **ART** technology provides: i) no operational costs to make a payment; ii) zero latency in in–person payments; iii) low latency in online payments; iv) no constraints to scalability and v) capability to operate without data communication networks. A digital payment system using a ledger and intermediated updates to account records cannot provide these features.

The software in the **ART** technology e-purse implement four central functions i) payment, ii) redemption, iii) aggregating receipt token factory request preparation and iv) key management. A formal model, like the one outlined in this paper, as solid basis for the correctness of the implementation, is a good way to convey trust in the correctness of system operations to users, operators and regulators.

[21] As mentioned above, the enhanced physical security of an e-purse intended for larger amounts can require more bulk and weight. Such an e-purse would still be much smaller than the corresponding amount of cash.

[22] As a computer an e-purse has memory constraints limiting the amount stored. It could be configured to store less then that technical maximum.

4 Aggregating receipt token e-cash as CBDC

With **ART** technology, **CBDC** can be realised as a fully decentralised, resilient infrastructure for secure online and offline payments. This solution delivers high throughput, low latency and a user experience very similar to those for Plain Old Cash. Aggregating receipt token technology e-cash meets the requirements for **CBDC** presented by the ECB in its 2020 report on the Digital Euro[24]. As a digital form of cash, it can be managed with the conventional central bank monetary controls; E-cash is a new, publicly accessible, digital component in the the monetary system, which complements the three conventional components: physical cash, digital commercial bank accounts and digital accounts with bank reserves.

A **CBDC** with **ART** e-cash can meet all use cases identified by Bindseil et al. in [3, p. 16]: Consumer-to-Consumer (**C2C**), Physical point of Interaction (**POI**), E-commerce, Recurrent payments and Corporate/business to business (**B2B**). As a digital payment instrument e-cash payments can be made across borders; special cross-currency tokens can be constructed by a money–changing service provider to facilitate receiving payments in foreign currencies. Privacy is unconditionally protected by design: a received amount of **ART** technology e-cash cannot identify the payer.

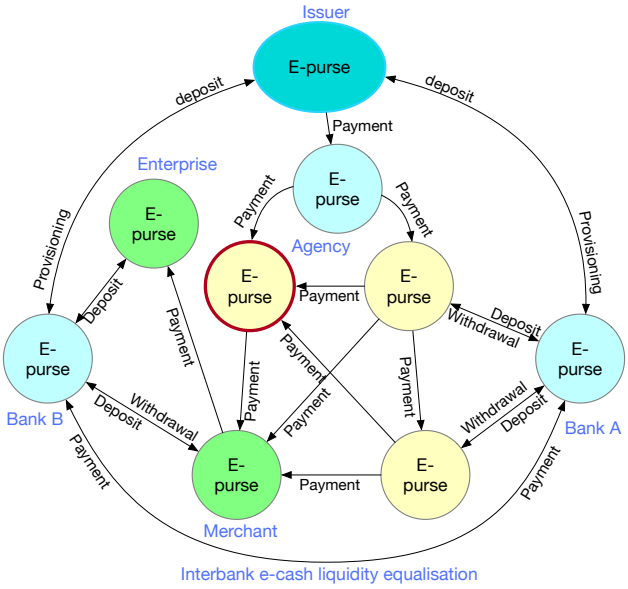


Figure 2: Users with e-purses in e-cash **CBDC**

Only two devices are involved in any payment, on one side the secure device in the payer's e-purse and on the other side a computer operated by a payee. This means that there are no external transaction costs for either parties to pay in e-cash. Transaction capacity for a single payer is constrained by the time needed to validate and compute digital signatures. Transaction capacity for a payee is practically unlimited as multiple payments can be handled concurrently. An in–person payment can be completed in milliseconds and an online payment can typically be done in less than half a second.

Figure 2 schematically shows an e-cash **CBDC** with nine functionally identical e-purses that interact with each other in various payments. To demonstrate different system operations the figure shows e-purses owned and operated by a number of different prototypical users indicated by the different coloured circles:

- Three generic users, the yellow circles, which each can make and receive payments to and from any of the e-purses operated by other users;
- Two banks with the e-purses communicating with the bank's adminis-

trative system, where e-cash using customers can deposit and withdraw e-cash into or from their respective bank accounts;
- A non-bank agency, like social services or a postoffice, that supports users that have no bank account enabling them to use e-cash to make online and offline payments;
- A money issuer, typically the central bank, that provisions banks with e-cash, or can remove e-cash from circulation by requesting banks to make payments;
- Two commercial users, a merchant and another enterprises, the green circles, of which the merchant primarily receives e-cash from customers using e-cash and uses it to pay a supplier or delivery company depositing the remaining proceeds in a bank account.

A user without a bank account, as shown by the top left yellow circle, can fully partake in the e-cash system with receiving and making payments Each arrow shown in fig. 2 is a payment in e-cash that in each instance follows the same two-message **ART** payment protocol. Different users may have differently built hardware and software parts in their e-purses to reflect different use cases and differences in amounts stored or paid. The differences in the e-purse can be in: i) the hardware security features, ii) the software in the user's IT system to interact with it, iii) the kind of payment signature algorithm or the size of the payment key, iv) the user's interface device, v) the programability of payments, and vi) the method(s) of authorising a payment.

The payment in e-cash is gratis, as no costs are involved in making or receiving a payment. Operational costs in the system are in creating aggregating receipt tokens and in periodically certifying the cryptographic keys in an e-purse. Different bussines models may be applied to cover the operational costs, these would typically apply a fee structure designed to promote financial inclusion, e.g. by providing a free tier with an ample amount of received payments of small amounts.

E-cash deployment

To deploy **CBDC** as e-cash, every resident and corporation in a jurisdiction can be provided with a tamper resistent secure device as the core component of an e-purse. Security is designed with a layered architecture with at its base a smartcard chip that implements the payment protocol and stores the secret cryptographic keys. The smartcard chip is embedded in the e-purse security device with additional processing power to manage communication and provide tamper detection. Communication interface and user interface software complements the secure hardware to become a fully functioning secure **ART** technology e-purse.

For most users a mobile phone app can provide the needed complementary e-purse functions, in particular payment approval with PIN, password or user authentication based on the phone's biometric features. With the app a user can configure the approval methods, review past payments and program recurring or scheduled payments or separate funds for budgetting.

The e-purse hardware device can be provided without cost to a user by preloading it with an amount of spendable e-cash that can exceed its costs of manufacturing, in which case seigniorage of the loaded amount of cash covers the costs of the e-purse. Like a banknote the secure hardware device

could be branded with the name of the central bank and the currency symbol, to also make it recognisable as a bearer payment instruments.

For merchants and other commercial users, suppliers of existing administrative systems, e.g a point of sale (**POS**) app for a tablet, could enhance their products with e-purse software functions. To promote this integration, e-purse software modules for various IT platforms could be made available at low costs.

For people who cannot use a bank account e-cash enables online purchases. As it has no per-transaction fees such purchases could even be possible for people with the lowest income. Deployment of e-purses could help achieve the often stated goal of financial inclusion for **CBDC**.

The e-purses for banks could be provided by approved vendors that incorporate the aggregating receipt token software implementation in a Hardware Security Module (**HSM**). These vendors could also provide the driver modules for integration into the banking software. The bank integration software for its e-purse can be associated with a web *API* front-end, to handle deposit and withdrawal of e-cash. Easy digital access to deposit and withdrawal allows user to backup e-cash from their device to a bank account. Software modules for a standardised web API implementation could be made available to promote this integration in the banking IT infrastructure.

The e-purse of the issuer could be implemented with an **HSM** like any other bank and similarly be integrated in the bank's administrative IT system. An implementation with an air gap can be realised with optical communication for e-cash provisioning and removal. In that case the HSM uses scanner and printer to receive and send an aggregating receipt token to respond to a provisioning request. The issuer **HSM** implementation of its e-purse can be configured to require multi-user authorisation, e.g. a group signature, to make a payment that issues e-cash. In that case the input QR code could be extended with the required number of digital signatures for the operators to authorise this e-cash payment.

5 Conclusion

A comprehensive e-cash system based on my **ART** technology is the best way to realise **CBDC**. It has a fundamentally lower operational cost base, inherently high resilience and efficient scalability in comparison with alternatives. This **ART** technology provides the foundation for a country-wide e-cash system that strongly protects privacy while enabling widely accepted, high-speed, cash-like, online and offline payments: from citizen to citizen, from consumer to merchant, from business to business, from citizens and businesses to government, and from government to all.

A formal model of the implementation of **ART** technology e-cash demonstrates to security professionals and other experts, using mathematical proofs, that the system strongly protects both the monetary value stored and paid, and the privacy of its users. With a transparent system architecture, this model provides a sound basis for an open discussion with a critical citizenry on the objectives and all implementation aspects of CBDC. As e-cash, **CBDC** brings citizens a true bearer payment instrument that is fully in line with the digital age; e-cash is cash that is better than Plain Old Cash.

6 Bibliography

[1] Michel Aggietta et al. *La monnaie souveraine*. Odile Jacob, 1998. ISBN: 9782738106315.

[2] Balthazar Bauer, Georg Fuchsbauer and Chen Qian. *Transferable E-cash: A Cleaner Model and the First Practical Instantiation*. Cryptology ePrint Archive, Paper 2020/1400. https://eprint.iacr.org/2020/1400. 2020. URL: https://eprint.iacr.org/2020/1400.

[3] Ulrich Bindseil, Fabio Panetta and Ignacio Terol. 'Central Bank Digital Currency: functional scope, pricing and controls'. In: *Occasional Paper Series*. 286. European Central Bank, Dec. 2021. URL: https://www.ecb.europa.eu/pub/pdf/scpops/ecb.op286~9d472374ea.en.pdf.

[4] BIS task force on security of electronic money. 'Security of electronic money'. In: Report 1. Report produced by the Committee on Payment and Settlement Systems and the Group of Computer Experts of the central banks of the Group of Ten countries. Basle: Bank Iinternational Settlements, 1996, p. 64. ISBN: 92-9131-119-7. URL: https://www.bis.org/cpmi/publ/d18.htm.

[5] Finn Brunton. *Digital Cash: The Unknown History of the Anarchists, Utopians, and Technologists Who Created Cryptocurrency*. Princeton University Press, 2019. ISBN: 9780691179490. URL: http://www.jstor.org/stable/j.ctvc77f9r (visited on 21/05/2022).

[6] Michael Butler and Divakar Yadav. 'An incremental development of the Mondex system in Event-B'. In: *Formal Aspects of Computing* 20.1 (2008), pp. 61–77. ISSN: 1433-299X. DOI: 10.1007/s00165-007-0061-4. URL: https://doi.org/10.1007/s00165-007-0061-4.

[7] Michael Butler et al. 'Transacted Memory for Smart Cards'. In: *FME 2001, Formal Methods for Increasing Software Productivity (01/03/01)*. Ed. by J. N. Oliveira and P. Zave. Address: Berlin. Mar. 2001, pp. 478–99. URL: https://eprints.soton.ac.uk/253695/.

[8] *Central bank digital currency: opportunities, challenges and design*. Tech. rep. 4. Bank of England, Oct. 2020. URL: https://www.bankofengland.co.uk/paper/2020/central-bank-digital-currency-opportunities-challenges-and-design-discussion-paper.

[9] David Chaum, Christian Grothoff and Thomas Moser. 'How to issue a central bank digital currency'. In: SNB Working Papers. Swiss National Bank, Mar. 2021. URL: https://www.snb.ch/en/mmr/papers/id/working_paper_2021_03.

[10] J. Dreier, A. Kassem and P. Lafourcade. 'Formal analysis of e-cash protocols'. In: *2015 12th International Joint Conference on e-Business and Telecommunications (ICETE)*. Vol. 04. July 2015, pp. 65–75.

[11] *Electronic money system security objectives*. Tech. rep. Based on common criteria. European Central Bank, May 2003. eprint: ISBN92-9181-362-1.

[12] 'Exploring anonymity in central bank digital currencies'. In: vol. 4. In focus. European Central Bank, 2019. URL: https://www.bis.org/cpmi/publ/d18.htm.

[13] M.G. Gouda and A.X. Liu. 'Formal specification and verification of a micropayment protocol'. In: *Proceedings. 13th International Conference on Computer Communications and Networks (IEEE Cat. No.04EX969)*. Oct. 2004, pp. 489–494. DOI: 10.1109/ICCCN.2004.1401715. URL: https://ieeexplore.ieee.org/document/1401715.

[14] Gesine Hinterwälder et al. 'Efficient E-cash in practice: NFC–based payments for public transportation systems'. In: *Privacy Enhancing Technologies*. Ed. by Emiliano De Cristofaro and Matthew Wright. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 40–59. ISBN: 978-3-642-39077-7.

[15] Shunsuke Inenaga, Kenichiro Oyama and Hiroto Yasuura. 'Towards Modeling Stored-value Electronic Money Systems'. In: *IPSJ Online Transactions* 3 (2010), pp. 176–185. DOI: 10.2197/ipsjtrans.3.176.

[16] Eduard de Jong. 'How to regulate crypto currencies'. In: *SUERF Policy brief*. 330. SUERF, May 2022. URL: https://www.suerf.org/suer-policy-brief/45357/how-to-regulate-crypto-currencies.

[17] Eduard de Jong. 'The implementation of e-cash: challenges and options'. To be published. Jan. 2023.

[18] Eduard de Jong and Jurjen Bos. *Arrangement storing different versions of a set of data in separate memory areas and method for updating a set of data in a memory*. EN. Pat. Dec. 2000. URL: https://www.freepatentsonline.com/6769053.html.

[19] Eduard Karel de Jong and Christopher John Stanford. *System with and method of cryptographically protecting communications*. Patent filed May 24 1996. Apr. 1997. URL: http://www.freepatentsonline.com/6553351.html.

[20] Donald Knuth. *The art of comuputer programming*. Vol. 2: *Seminumerical algorithms*. 3rd ed. The art of computer programming. Addison–Wesley, 1998. ISBN: 0-201-89684-2.

[21] Christine Lagarde and Fabio Panetta. *Key objectives of digital Euro*. European Central Bank. July 2022. URL: https://www.ecb.europa.eu/press/blog/date/2022/html/ecb.blog220713~34e21c3240.en.html.

[22] Mark S. Manasse. 'The Millicent Protocols for Electronic Commerce'. In: *Proceedings of the 1st Conference on USENIX Workshop on Electronic Commerce - Volume 1*. WOEC'95. New York, New York: USENIX Association, 1995, p. 9.

[23] *Mondex*. Accessed on June 23, 2022. URL: https://en.wikipedia.org/wiki/Mondex.

[24] *Report on a digital Euro*. Tech. rep. Frankurt am Main, Germany: European Central Bank, Oct. 2020. URL: https://www.ecb.europa.eu/euro/html/digitaleuro-report.en.html.

[25] Ronald L. Rivest and Adi Shamir. 'PayWord and MicroMint: Two Simple Micropayment Schemes'. In: *Proceedings of the International Workshop on Security Protocols*. Berlin, Heidelberg: Springer-Verlag, 1996, pp. 69–87. ISBN: 3540624945.

[26] Joseph B. Walton. *Why we use a new currency: the role of trust and control in explaining the perception and usage of Bitcoin*. PhD. 2020.

[27] T. H. White. *Arthur: The once and future king*. Collins, 1958.

[28] Alejandro Zamora–Pérez, Eliana Coschignana and lorena Barreiro. 'Ensuring adoption of central bank digital currencies – an easy task or Gordian knot?' In: Occasional paper series 307. European Central bank, Oct. 2022. URL: https://www.ecb.europa.eu/pub/pdf/scpops/ecb.op307~c85ee17bc5.en.pdf.