eCash 2.0

Inalienably private and quantum-resistant to counterfeiting

David Chaum
XX NETWORK

Thomas Moser
SWISS NATIONAL BANK

Abstract: The digital cash introduced here provides better privacy than paper cash while protecting society against criminal use far better than paper money ever could. In particular, it provides each holder, though their payments are anonymous, with the ability to allow irrefutable tracing of any of their payments—and this ability is "inalienable" in that it simply cannot be given or taken away. This improved control by persons over the privacy of their own payments further allows the adoption of privacy where it might otherwise be blocked by regulation. Without such inalienability, moreover, it is believed that payment privacy intended for particular persons may be taken from them, by malware for instance, and used to protect the privacy of aggregated payments made by others. The supply of currency is completely controlled by its issuer, and the currency is provably protected against counterfeiting even by a quantum computer. Optionally, a blockchain, or individual customer choice of public blockchain, can bring the advantages of such chains, including transparency of the total amount of unspent digital cash outstanding. The design builds on several well-established cryptographic protocols, like public-key digital blind signatures and mix networks, as well as some new cryptographic techniques of its own. Its improved privacy and quantum resistance, when combined with its Visa- or PayPal-like scalability, make it an ideal candidate for central bank digital currency (CBDC).

Introduction

Most central banks are currently exploring the issuance of central bank digital currencies (CBDCs), and a recent BIS survey on the topic found that central banks collectively representing a fifth of the world's population are likely to launch retail CBDCs in the next three years [1]. Many central banks are investigating "wholesale" CBDC, that is, for payments between banks and other institutions. CBDC schemes that meet strong enough requirements, like those used as the example here, can be used for both. Also, the G7 has recently published a set of Public Policy Principles for Retail Central Bank Digital Currencies (CBDC) [2] alongside a G7 Finance Ministers and Central Bank Governors' Statement on CBDCs and digital payments, which emphasize the importance of "rigorous privacy, accountability for the protection of users' data, and transparency on how information will be secured and used, to command trust and confidence by users." [3]. This view is echoed in the July 2021 People's Bank of China report on its CBDC in development, "The Progress of Research & Development of E-CNY in China": "E-CNY follows the principle of 'anonymity for small value and traceable for high value,' and attaches great importance to protecting personal information and privacy." [4] The importance of privacy and its potential impact on design choices was also stressed in the second joint report of a group of central banks and the BIS [5]. The protection of privacy in CBDC design is also a key public demand. A recent whitepaper on digital currency by the World Economic Forum notes: "Of the 8,200 comments received by the European Central Bank (ECB) during its consultation period on the potential for a Euro-denominated CBDC, 41% of all replies centred around privacy." [6]. Similarly, public feedback to the Bank of England's March 2020 Discussion Paper on CBDC emphasized the importance that users place on

having privacy in their transactions [7]. It is hard to imagine that a CBDC that would allow government to track every payment would be welcomed and widely adopted by citizens, especially if there were a superior alternative.

This legitimate interest in protecting privacy must be balanced against the equally legitimate interest in preventing crime. To address these needs, it has been suggested that privacy be limited somehow to low-value transactions, as in the PBOC report on e-CNY. A substantial proportion of the Eurosystem Report's finance-professional respondents concur: "A quarter support selective privacy under which transactions below a given amount would stay private (mostly credit institutions and PSPs)."[8] Such an approach would also seem to be consistent with international standards on combating money laundering and terrorist financing, according to which occasional cash transactions or wire transfers whose value remains below a certain threshold require no or only simplified verification of customer and recipient information. It has additionally been proposed that consumer withdrawal and holding amounts of CBDC be limited, which would also serve as a measure to control the total volume of a CBDC in circulation. For example, the Eurosystem Report notes: "Almost half of citizen respondents mention a need for holding limits, tiered remuneration, or a combination of the two, to manage the amount of digital euro that would be in circulation. A similar share of professional respondents agree."[9]

However, these proposals leave open the loophole that multiple such small amounts can be aggregated to make large but untraceable transfers of value. The CBDC solution introduced here, eCash 2.0, prevents this possibility. eCash 2.0 is anonymous—yet aggregating amounts larger than those issued each user is thwarted. Anonymity is obtained via the "blind signature" technique used by the original eCash (as further detailed below). New here, however, is that each user is given, as part of enrolling in the system, an irrevocable ability to undo the anonymity of any value withdrawn from their account—even if the user wishes to give this ability up. This makes aggregation of value obtained from multiple user accounts very risky. With peer-to-peer payments, if the value issued to a user has already been spent by someone else, a criminal aggregator for instance, the user can at least reveal where it was spent. But if the value is not already spent, the user can spend it first, thereby preventing anyone from spending it later. Together, these properties greatly reduce the risk of criminal aggregation and of subsequent abuse of the privacy afforded.

The system builds on and improves the eCash technology used by some major commercial banks in the 1990s. [10] This technology introduced "digital bearer instruments" that are withdrawn "blinded" and so only entered in a central database when deposited. This provided what was called "one-way privacy," making the system unsuitable for uses such as extortion and bribery. [11] The example of a CBDC architecture illustrated here structurally differs from that of earlier eCash, but preserves these properties. It is structured so that all consumer and merchant interaction is with commercial banks, while money creation and the database of deposited money are provided exclusively by the central bank behind the scenes. Commercial banks authenticate their customers and monitor the extent of withdrawals and deposits, but otherwise the presence of these intermediaries does not affect the underlying cryptographic protocols.

Consumers are first enrolled, ideally, via a visit to the branch of a commercial bank where they are known or identified (see Figure 1). Thereafter, withdrawal can be as simple as withdrawing paper cash via an automated teller machine (ATM) but might typically be conducted online. Because each transaction is separate, system resources scale linearly with growth in transaction volume. Moreover, as validated by the earlier practical deployment of eCash 1.0, operational robustness, cost, and throughput speed are all attractive. Two other differences are that eCash 2.0 is secure against counterfeiters, even those with access to quantum computing; and eCash 2.0 can optionally but flexibly extend to public blockchains and hence bring their various advantages. A way to adapt eCash 2.0 to offline use has been proposed. It uses smartphones in combination with a new type of non-chip physical card to allow secure payments where no online connection is available. [12]

Anonymity and Misuse Prevention

The eCash 2.0 CBDC introduced here can be considered "software only," as it requires no special hardware devices. Merchant or consumer users, if their secret cryptographic key were to be compromised, would stand to lose only the amount of money they are holding in the system. To protect their keys against attack, some users and merchants may choose commercially available key protection devices, such as the digital custody now built into consumer hardware like smartphones. Banks can be expected to continue to use current commercially available hardware devices to protect their keys. Transactions remain quite fast, even if their number becomes large, because additional transaction volume can be efficiently routed to essentially independent but appropriate processing resources, giving the system the kind of linear scalability enjoyed by typical large transaction-processing systems like Visa or PayPal today.

With eCash 2.0, a user can make payments to merchants while remaining anonymous, even if the merchant and the user's bank try to discover the user's identity from all payment information they can obtain.

The commercial banks are in turn assumed to comply with so-called "Know Your Customer" and "Anti-Money Laundering" regulations (KYC/AML). However, if a single user could control large amounts of CBDC, the KYC/AML provisions could be circumvented completely. Preventing aggregation of CBDC implies, at least, that no user should be able to withdraw too large an amount of spendable CBDC, as a few users might be leaders or minions of criminal organizations. This is easily addressed by monitoring or limiting amounts withdrawn per user. Again, an analogy with paper money would be the ATM withdrawal limits on most consumer cards.

However, such restrictions, common to several other CBDC proposals, are only the beginning of preventing criminal misuse by aggregation, not the end. Much more insidious and fundamental potential threats could seek to allow a single person or organized criminal group to control a large sum of CBDC. Malware on smartphones, something that has proven impractical to stop, could for instance simply allow all withdrawal transactions to be with keys centrally controlled by those who created the malware. As to transaction size triggering the suspension of anonymity, such prohibited transactions could be accomplished via numerous smaller payments between what appear to be separate accounts but that are in fact controlled by criminal individuals or organizations, whether though user collaboration or covertly via malware.

In some scenarios, for example, nobody would notice the diversion of fully untraceable money if, once it was spent from compromised phones, it was retrieved and diverted from compromised retail sites. These sites could be one or more popular payment destinations that are unaware that they themselves have also been compromised. Alternatively, the sites could be gray-market or black-market sites that perhaps only accept payments from phones running the modified software, so the user could be incentivized to install the malware on their phone in order to be able to use these sites. It's even possible that parties paid by the aggregator could verify that they themselves control the untraceability of the payments they then would make with funds received from an aggregator.

More technical aggregation attacks that could be widely applicable, but are also thwarted here, include payments for undelivered goods and false refund transactions. The threat model of the protocols presented accordingly includes such apps and user behavior, and thus requires a structural solution. When CBDC is thought of as an electronic replacement for banknotes, the precedents by analogy are once again familiar. For instance, clearly nobody should be able to withdraw cash from your bank account but you. In this respect, eCash 2.0 is already superior to paper money, since, as will be explained, withdrawal is just as quick and simple as taking cash from a virtual ATM, but far more secure. Similarly, if a bad actor were somehow able to take cash from your account, you would want the notes' serial numbers to be known so that the miscreant could be tracked, if not apprehended. On the one hand, banknotes today don't allow such tracing, but CBDC can. On the other, CBDC can be used more easily than banknotes by criminals, in part because it can more easily be hidden when stored or transported but also because it can be used to pay remotely. But if large sums of truly privacy-protected CBDC were at the disposal of criminal organizations—the problem solved fundamentally here—the privacy afforded users could limit ways to stop or apprehend them, and their operations could be greatly facilitated and protected.

A simple example procedure for when a user initially signs up to get CBDC (say, by opening a CBDC facility as part of a current account with a commercial bank) involves a user creating a passphrase that will provide access to the user's private key and can be used to create a corresponding public key. The user enters the passphrase into an app on their phone, but also has it memorized.

At the commercial bank branch, the user is asked to write the passphrase down and then answer a few questions about it. The user's phone app and the bank's system together randomly pick the questions about the passphrase from what is in effect a very large list. The app in the user's phone communicates the passphrase to the bank's system, but with each word encrypted. The banker asks the questions randomly arrived at by the customer's phone app and the bank's system. The banker or bank software then enters the answers provided by the user into the bank's system. The system, by communicating with the user's offline phone app digitally over Bluetooth or the like, obtains a zero-knowledge proof[1] that everything is as it should be and registers the user's public key.

[1]A so-called zero-knowledge proof is a cryptographic technique a computer can use to convince another computer that some underlying data "cleartext" possesses specific properties while the cleartext remains encrypted. In the example of Fig. 1, it allows the user's smartphone to convince the the bank's tablet that the user's passphrase is in effect the user's private key--yet the tablet learns essentially nothing about either the passphrase or the private key.

Of course, neither the user nor their phone has given enough information to the bank to allow it to obtain the user's secret key. The crucial thing is that in the passphrase as a whole, the user retains the secret (private) key and the ability to obtain it at any point in the future. The app on the user's phone knows the passphrase, but it cannot keep the user from memorizing and/or recording it elsewhere or otherwise ever take that knowledge away from the user.
(See Figure 1.)

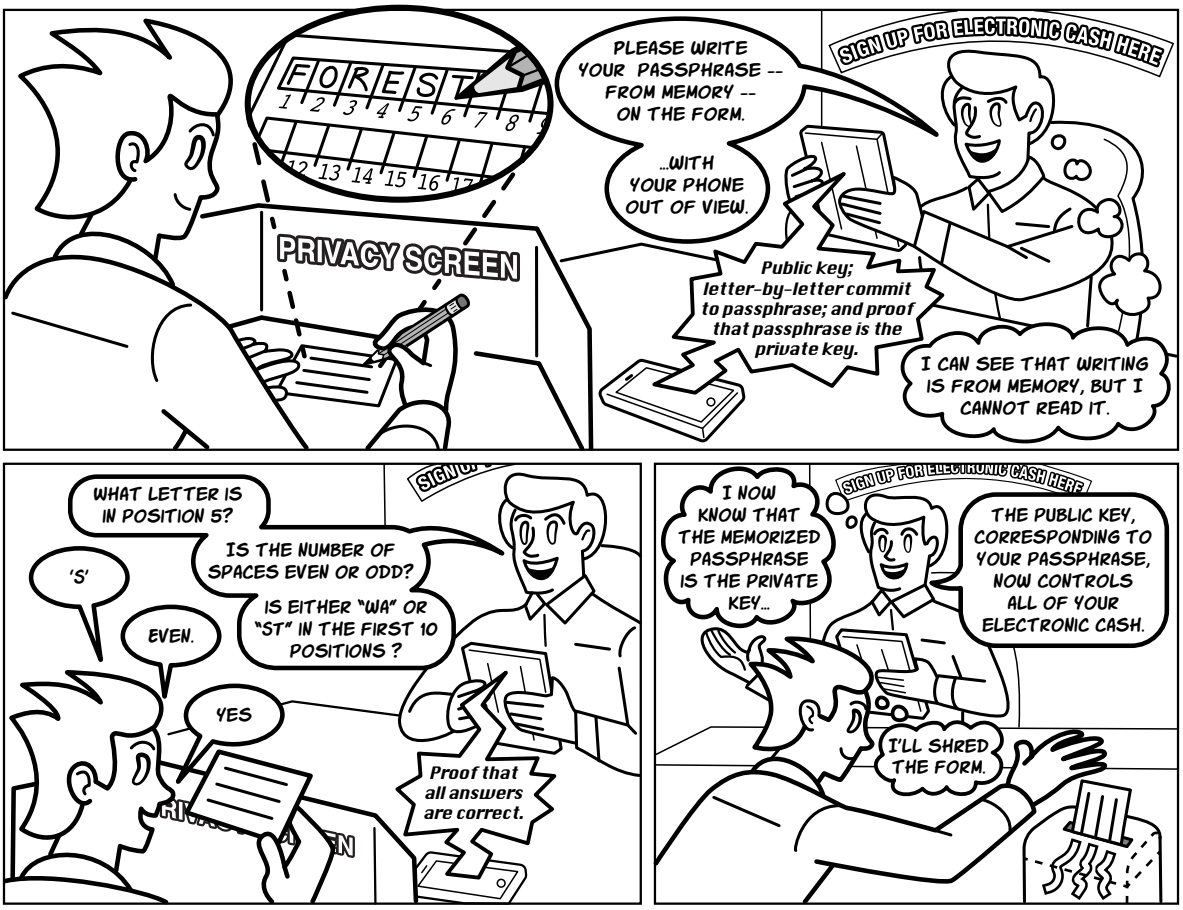

Figure 1: Enrolling for CBDC at a Commercial Bank.

(1) After the user has identified themselves to the bank, they are asked to write down their passphrase from memory on a gridded paper form behind a privacy screen so that the banker can see they are writing from memory but cannot see what they are writing. (2) The user leaves their smartphone to one side so that the app cannot display to the user but can communicate with the bank's system to generate random questions about the passphrase. (3) The banker asks the user these questions, and the user answers them from the passphrase they have written on the form. (4) The banker enters the user's answers, and the system uses a zero-knowledge proof to confirm that the user has the secret key corresponding to the public key that the bank knows. (5) The user destroys the form and has established a secret passphrase that unlocks the signing key for their CBDC account with the bank.

To make this work digitally behind the scenes, cryptographic protocols are employed. It works as follows: The user chooses their passphrase and shows it to their smartphone camera, making sure that the phone has OCRed it correctly. The phone then computes the public key for the user based on the passphrase; and it encrypts, by a special blinding,[2] each letter position of the passphrase separately.

[2]"Blinding": Imagine a randomly numbered card inside an opaque envelope that is stamped from the outside with a seal like the signets once used to seal letters with wax. The impression of the seal embosses the card inside with the signature, but when the envelope is removed, the signer has no way to determine which specific number was on the card signed. Blinding is a cryptographic technique that conceals a cleartext number by transforming it into cyphertext in such a way that it can be digitally "embossed" with a signature. Removing the envelope in the analogy is equivalent to the cyphertext later being decrypted [unblinded] to obtain the now signed form of the cleartext number. This technique, termed "digital blind signature," was developed to create the banknote-like anonymity property of eCash 1.0, whereby the bank would apply a signature with a certain fixed monetary value to a blinded "note" formed by the customer's device and the customer could later unblind and untraceably spend the note.

Both public key and encrypted positions are provided over Bluetooth to the bank tablet along with a zero-knowledge proof that the positions together comprise the private key. Cryptographic "coin flips" between the phone and tablet select the choice of queries from a very large space of predefined possibles. The banker asks each selected query, the phone provides a zero-knowledge proof that the list of encrypted positions—and thus the customer public key, which has already been shown can be reconstructed by the customer from the passphrase—is consistent with the customer's answers.

Quantum-Level Security Against Counterfeiting

If a CBDC were to be counterfeited, as with counterfeiting of banknotes, the potential for systemic harm would depend on scale and detectability. With eCash 1.0, such an attack could be accomplished clandestinely, without triggering an alarm until statistical outflows make the situation evident. For instance, counterfeiters could somehow compromise the central bank's computing resources that have access to signing keys. The central bank could at that point suspend the money, require customers to deposit all unspent money, and then re-issue new money, temporarily disrupting the economy.

However, if counterfeiters were to use a quantum computer to back-derive the bank's signing keys from its public keys, then the replacement system could not simply be another eCash 1.0 instance with different keys, as the quantum computer could break the new keys in effect instantly. Thus, if the system were not quantum-resistant, the mere claim of a quantum attack could arguably require removal of the privacy feature, as well as causing even more serious economic disruption.

The solution proposed here need not affect use of the system by consumers or commercial banks. The additional protective measures are performed by the central bank only. The approach even brings with it the potentially useful advantage of connecting the currency to a blockchain that can be public. (In any system with this architecture, a commercial bank can have a separate "out of band" secure channel with the central bank, which would allow it to periodically check a hash[2] of the withdrawals and deposits made on its behalf at the central bank and thereby ensure that false requests are not being injected.) The essential concept of the quantum resistance is as follows: during each withdrawal transaction, the user's phone prepares a message that includes a quantum-secure hash[3] of the spendable form of the coin being withdrawn.

Only the central bank can allow this prepared message to be included as input to a mix batch of such messages corresponding to the respective denomination. Hence, the corresponding output batch of the mix contains, for each coin that can be spent, a quantum-secure authentication that can automatically be verified when the coin is revealed in payment. Moreover, the mix maintains the unlinkability between user account and payment information. (See Figures 4 and 5 and below on the proposed use of a mix network.)

While a blockchain is not strictly needed as a place to publish the hashes that are output by the mix, it does provide a robust store that can be infeasible in practice to corrupt. Moreover, if each coin is in effect its own "wallet ID" on the blockchain, then the CBDC could be allowed to be transferred between wallet IDs on the blockchain. This would in turn allow use not only of so-called "smart contracts" but also of Liquifinity technology [13].

The Bigger Picture

The decentralization of control over digital currency by the "user-irrevocably-knows-keys" approach introduced here is related to the decentralization of power by voting in democracies. Both involve privacy—of voter choice or of who spent which cash—but the connection runs deeper. In voting, voters may or may not want their vote to be private, such that they control who can see how they vote. But society has an interest in a stronger property, technically often called "ballot secrecy," which is that even if voters want to show others how they voted, they should not be able to do so. This ballot secrecy property is typically enforced by the mandatory physical presence of voters in booths, visually verified by poll workers and other voters. It thwarts so-called "improper influence" of voters, which includes vote buying and coercion.

Similarly, a user may of course wish to have privacy about where they spend their cash. But society has an interest in users themselves always having the keys needed to spend, recognize, and trace their cash. The techniques presented here allow society to ensure that nobody can improperly usurp any user's access to the keys conferring those abilities.

[3]A "cryptographic hash" is the fixed-size output of a standardized cryptographic hash algorithm when it is applied to specific cleartext data. The holder of the cleartext can easily compute the hash and provide it to the recipient; the holder can also later provide the recipient with the cleartext so that the recipient can easily check that the hatches match. But the recipient cannot reverse-engineer the cleartext from the hash without breaking the hash algorithm.

System Architecture

One primary objective of the overall architecture of the CBDC scenario mentioned is ensuring that central banks do not have to interact directly with customers. Rather, authentication is delegated to commercial banks who have the necessary infrastructure (presumably today including KYC/AML support) already in place. Withdrawal and payment protocols are the only two that reach the central bank, each through a commercial bank as intermediary. Thus, before the central bank signs a coin into existence for a commercial bank customer, that customer has been authenticated and the corresponding amount withdrawn from the customer's bank account.

Next, we present an architectural-level description of the workings of the system through the lens of an actual withdrawal transaction, and then, separately, an actual payment transaction. It will be assumed but not shown explicitly that infrastructure providing authentication between banks is in place.

A withdrawal of CBDC by a user would proceed as follows (see Figure 2): Overall, the process is analogous to a customer withdrawing physical cash from an ATM. A customer authenticates to their commercial bank using that particular bank's authentication and authorization procedures, including demonstrating their knowledge of their account keys. The customer's computer (mobile or otherwise) then computes both the coin and the blinding factor that cryptographically conceals the coin from the banks. Next, the customer sends the blinded coin to the commercial bank via an established secure channel together with an authorization to withdraw the coin and debit the customer's account. The commercial bank debits the coin value from the customer's current account and digitally authenticates its authorization of the request on the blinded coin it forwards to the central bank for signing. The central bank deducts the value of the coin from the commercial bank's account at the central bank, signs the coin, and returns the still blinded signature to the commercial bank. Then the commercial bank forwards the blind signature to the customer's electronic wallet. Finally, the customer's wallet unblinds the signature and stores the newly minted electronic cash in its database. (See Figure 2.)

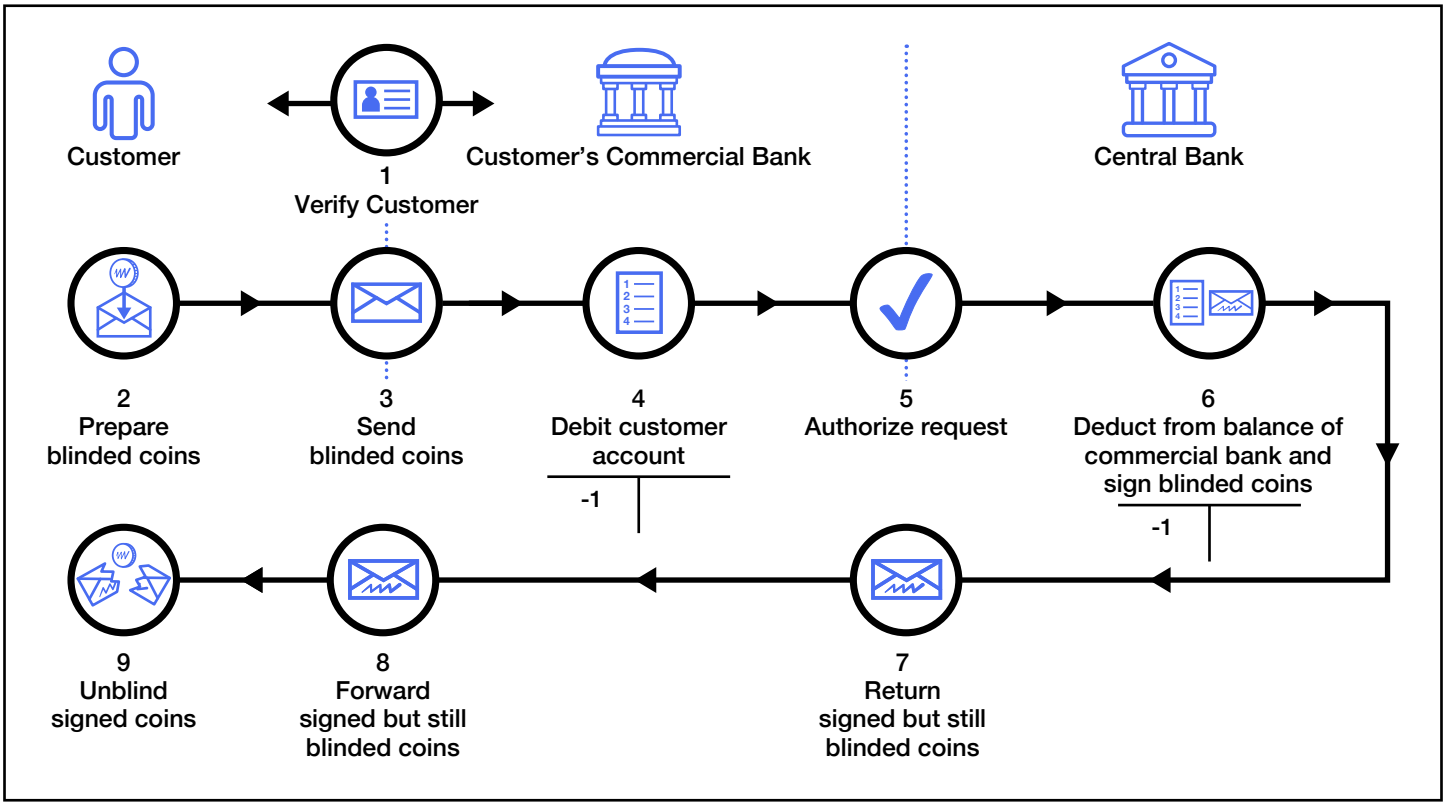

Figure 2: Withdrawal Process

(1) The commercial bank authenticates the customer account holder. (2) The customer's electronic wallet prepares blinded coins. (3) The customer's device sends the blinded coins to the commercial bank's system, which authenticates the request and debits the customer's account accordingly. (4) The commercial bank authorizes the customer's request to the central bank. (5) The central bank deducts the value of the coins from its account for the customer's commercial bank and digitally signs the blinded coins. (6) The central bank returns the signed but still blinded coins to the commercial bank. (7) The commercial bank forwards the signed, blinded coins to the customer's device (8). And (9) the customer's device unblinds the coins—so they are now ready for spending.

When a user spends CBDC, the process is analogous to paying a merchant in cash: the merchant deposits the cash in the merchant's own account at a commercial bank and the commercial bank can deposit the cash in its own account at the central bank.

More specifically, the spending of CBDC proceeds as follows: The customer selects goods they wish to buy, and the customer's phone transmits coins in the payment amount to the merchant. The merchant's system then validates the payment details and passes the coins (together with the merchant's account information) to the merchant's commercial bank. From this point, the process need take only a few hundred milliseconds: The merchant's commercial bank validates that this is one of its merchant customers and forwards the digital coins to the central bank. Since a corrupted customer device might attempt to spend the same coins more than once, the central bank verifies the signature but also checks for double-spending in its own database(s). If everything is in order, the central bank credits the commercial bank's account at the central bank and sends confirmation to the commercial bank. Next, the commercial bank credits the merchant's account and informs the merchant, so the merchant can release the product to the customer. (See Figure 3.)

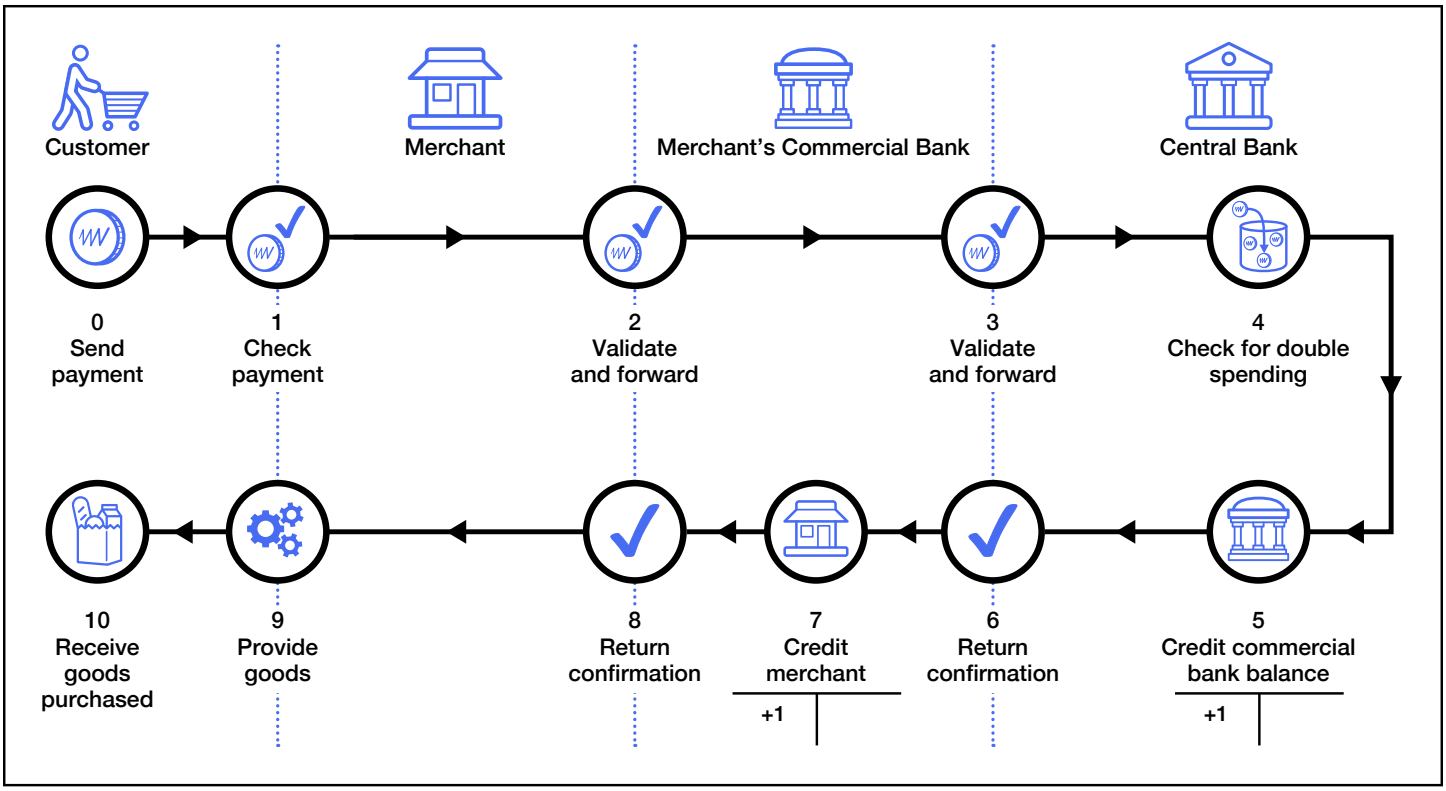

Figure 3: Spending and Merchant Deposit Process.

(0) The customer transmits coins in payment to the merchant. (0) The merchant checks the payment amount. (1) The merchant's system can validate the coins before forwarding to the commercial bank together with the authentication related to its merchant account. (2) The commercial bank can also validate the coins before forwarding them to the central bank. (3) The central bank must both check the validity of the coins and check them against its database(s). (4) The central bank, assuming everything verified, credits the account of that commercial bank. (5) The central bank confirms to the commercial bank that its account has been credited. (6) The commercial bank in turn credits the merchant's account in the same amount. (7) The commercial bank confirms to the merchant that its account has been credited. (8) The merchant provides the goods to the customer, (9) and (10) the customer receives the goods purchased.

Overview of the Basic Cryptographic Protocol

The cryptography that defines the basic system is shown in simplified form in Figure 4. Current standards-based best practices for general use of cryptography, such as for establishing authenticated/private sessions are, however, omitted for clarity, as is customary in describing higher-level protocols like this.

The eCash 2.0 protocol, introduced here in simplified form, is based on the well-known and longstanding RSA cryptosystem. In RSA, each party creates a public key by multiplying two very large suitable primes of their own secret choice; factoring these two numbers apart is believed infeasible (at least without the help of a quantum computer, a topic covered elsewhere here). The central bank's public key, c, which it formed in this way, is used to certify CBDC in the system. While anyone can raise any number to a counting number power modulo the modulus c, only the central bank can raise numbers to fractional powers modulo c, conferring on it the exclusive ability to form its digital signature. Such modular arithmetic, sometimes called clock arithmetic, based on a public modulus c, simply defines "modulo c" as the remainder after dividing out all multiples of c. It allows anyone to verify signatures merely by raising them to a public counting-number power.

For simplicity, user and merchant are here assumed to have a banking relationship with a single commercial bank of their choice and to be able to move money between CBDC and their accounts at that bank. Though not made explicit in the architectural discussion above, each user here also has their own inalienable "secret signing account key" to digitally sign requests for transfer between their accounts. Such digital signatures authenticate ownership of the corresponding account public key and provide durable proof of the withdrawal instruction details and their authorization.

A pair of numbers worth one cent in the system, x | f(x)1/3 (mod c), can be verified by anyone simply raising the second number to the power 3 modulo c and checking that the result equals what is obtained by applying the public one-way function f() to the first number of the pair, x. (The cryptographic assumptions are that it is infeasible for an adversary to compute fractional powers on images under f without access to the randomly chosen information used to form c.)

Here, the value of 1¢ is assigned public exponent 3 in the RSA system with modulus c. The value of 2¢ is assigned exponent 5, 4¢ exponent 7, 8¢ exponent 11, and so on; each successive power-of-two denomination value is represented by the corresponding next prime number as an exponent, all under modulus c. Thus, 13¢ (13 = 1+4+8) corresponds to denominations 1, 4, and 8 cents and exponents 3, 7, and 11. Since only the bank can form the fractional powers 1/3, 1/7, and 1/v, when the bank is presented with x, y, and z and $x^{1/3}$, $y^{1/5}$ and $z^{1/11}$, it knows this should be worth 13¢—but of course it needs to check that x, y, and z have not been deposited before. Put differently, the 13-cent example uses a binary number of only three bits in length; for each additional bit (corresponding to an additional bank secret fractional power) the number of possible payments that can be made doubles. Just by selecting one or zero of each of 16 fractional powers, payments of up to $655.36 can be made in exact cents. This is because 216 = 65536 cents.

As summarized earlier, blind signatures are used here to protect user privacy. A user's smartphone or other device can simply "blind" a desired number f(x) by multiplying it by a random number b that it chooses and raises to a denomination power, for example b^3 for a 1¢ coin. This blinded value $f(x)b^3$ (mod c) can, in exchange for a 1¢ withdrawal, then be signed in blinded form by the central bank. The central bank uses its unique ability to compute the fractional power 1/3, resulting in $\{f(x)b^3\}^{1/3}$ (mod c). Because exponentiation distributes over multiplication, what the user's phone gets back equals $\{f(x)\}^{1/3}b$ (mod c). And since the phone knows b, it can unblind simply by dividing b out, leaving the 1/3 power on f(x) and yielding what turns out, because of the underlying structure of the modular arithmetic, to be a perfectly unlinkable, unblinded 1¢ coin x| $f(x)^{1/3}$ that can then be used in payment (See figure 4). After the payment, which account the value was withdrawn from remains perfectly hidden because of the blinding; however, since the payer knows x, the payer can always reveal x (or a property cryptographically hidden in x) to allow the beneficiary of the payment to be traced.

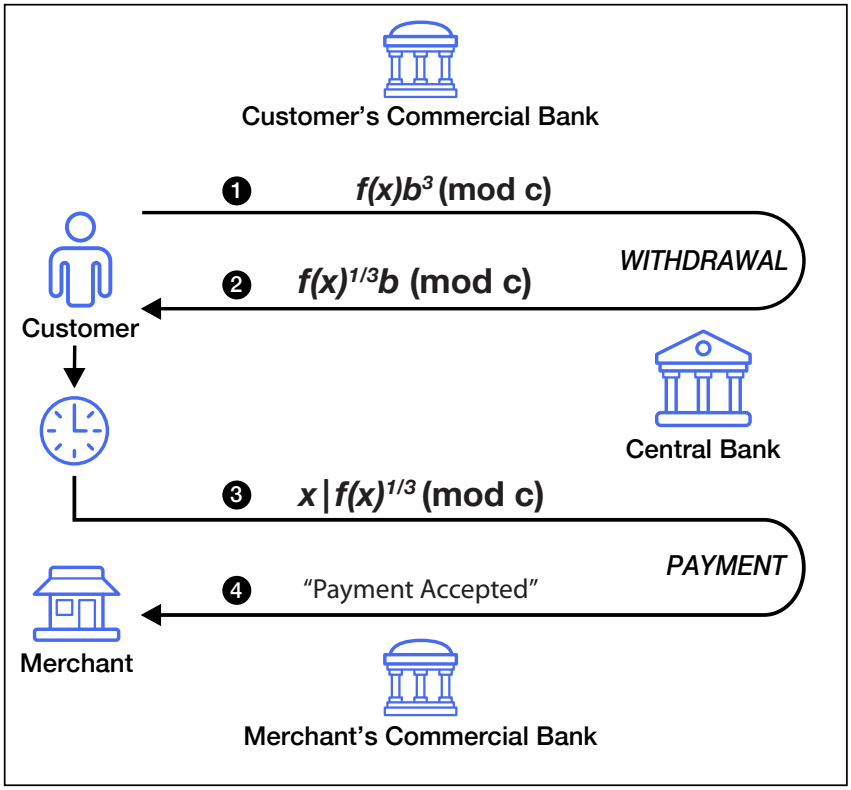

Figure 4: The Basic eCash 2.0 Protocol

(1) The customer's device prepares a blinded coin with value of 1¢ as follows: (a) it generates x as a secret random value; (b) it applies the public one-way function f to x, yielding f(x); (c) it generates a second secret random value b; (d) it raises b to the power 3 (modulo c), yielding b^3 (modulo c); and (e) it "blinds" the coin by multiplying f(x) times b3 (modulo c). The customer's device then sends this blinded coin to the commercial bank (not shown), which forwards it to the central bank. The central bank cryptographically signs the blinded coin by raising it to a fractional power of 1/3, which only it can do. (2) The central bank then returns the signed but still blinded coin to the customer, via the commercial bank. The customer's device unblinds the now signed (valuated) coin by dividing out b. Later, the customer spends (3) the coin with a merchant that sends it on to its bank, which forwards it in turn to the central bank. Because the customer retains the private key formed along with the coin, the customer can always reveal and prove where they spent that coin. This greatly reduces the potential for criminal abuse of the coin. The spent form includes two numbers, shown separated by a "|". The first number is x and the second is the unblinded signed coin $f(x)^{1/3}$ (mod c). The central bank applies f to the first number and cubes the second number and then verifies that the two results are equal (modulo c). The bank also checks to make sure that the coin has not been previously spent by consulting a "double-spending database," not shown, that it maintains by updating to include the x as already spent. Finally (4) the central bank sends back through the commercial bank and merchant the message that the payment has been accepted.

This blind-signature protocol was invented by the first named author in 1982. In the 1990s, DigiCash implemented it and provided it to commercial banks, such as Deutsche Bank, that deployed it online connected to their customer's current accounts [10].

Quantum Resistance and Blockchain

Also invented by the first named author even earlier, in 1979 [14], was the concept of mix networks, which make it possible to send virtually untraceable communications. Here, a mix network is used to preserve privacy while addressing the threat of a quantum computer being used in counterfeiting. Every coin formed using a one-way function f by any user's device is forwarded through a mix network to be checked against a database of spent coins by the central bank. Optionally, the coin can also be published on a blockchain so that any user can also check for it (see Fig. 5). (As mentioned earlier, since users control the hashes of their blinded, unspent coins on that chain, they can make peer-to-peer payments directly on chain or use smart contracts or Liquifinity.) Because there are practical one-way functions known to be quantum-resistant in the strongest sense, even quantum computing cannot be used to forge a coin already on the list, since the counterfeiter cannot find x from the published f(x). This also means that even if a quantum computer reverse-computes the bank's private denomination-signing keys from its public keys, it cannot create spendable coins using those private keys. Only by somehow inserting false payloads into the mix that are not noticed in random checking by customers, could counterfeiters get images in the output database for which they know the pre-image x. (See Figure 5.) Thus, the total amount of CBDC outstanding becomes a matter of public record on the blockchain(s).

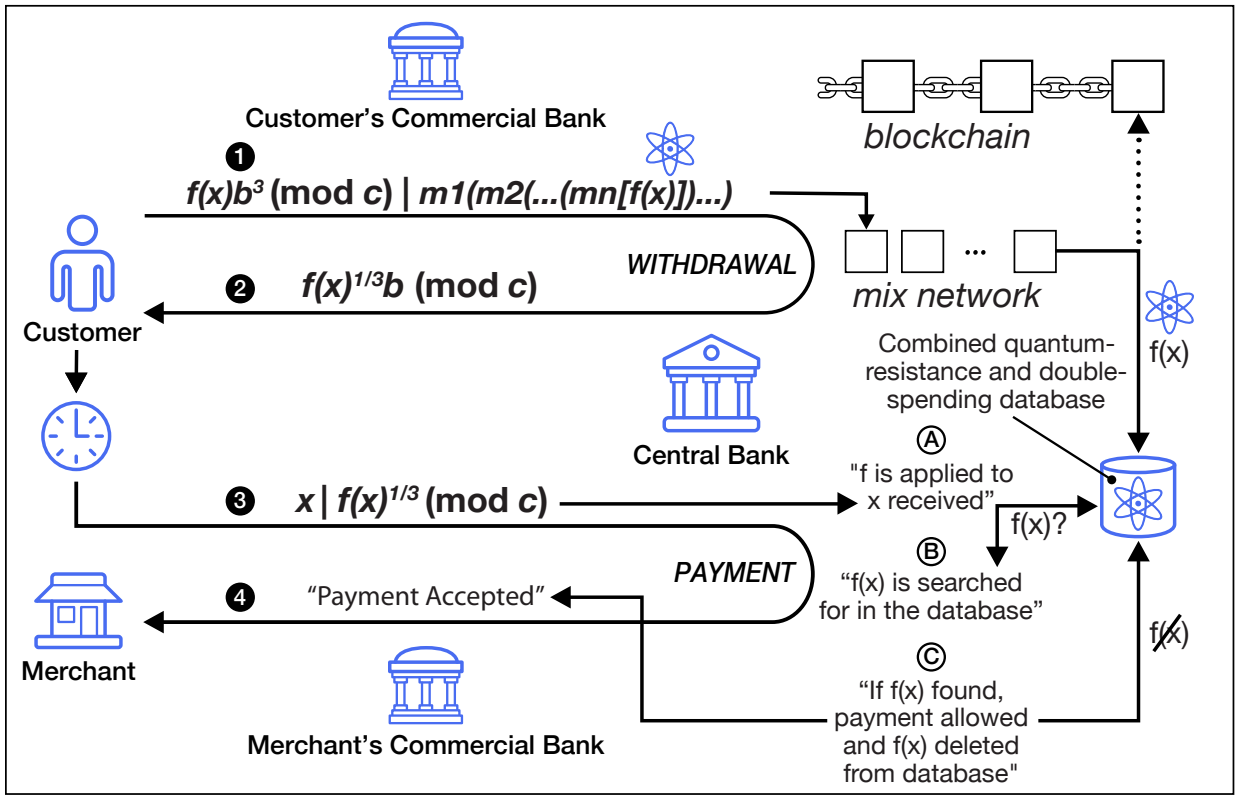

Figure 5: Withdrawal, Spending, and Quantum Resistance

This figure includes everything that was already shown and described with reference to Figure 4. What is new here is the second component of the withdrawal transaction, m1(m2(...(mn[f(x)])...); this value being allowed as input to the mix network by the central bank; the payload output of the mix network, f(x), going into the "combined quantum-resistance and double-spending database"; and the modified checking by the central bank. (Also shown is that f(x) optionally, as indicated by the dotted line, can be recorded on a blockchain, such as a public blockchain selected privately by the customer in the payload.) Essentially, the second component of the withdrawal is recorded in the combined database, but which withdrawal it comes from is hidden by the randomization of the mixing system, in which each "node" successively strips off the respective layer of encryption using its private keys and randomly permutes the batch of items before sending on to the next mix node in the cascade (see [5]). Thus, when a payment is received by the central bank, as in Figure 4, the additional difference would be that the central bank looks in the combined database: if the image under f that it reconstructs, f(x), is already in the database, then the payment is allowed and that image, in the same atomic operation of finding it, is removed from the database. But because of the mixing, which withdrawal corresponds to the payment remains hidden.

Scalability, Availability, and Recoverability

Three questions have interrelated answers: How easy is it to scale the system to accommodate demand as the number of transactions per second needed grows? How can the system be prevented from becoming unavailable and blocking people from making purchases? What happens if the central bank's secret signing key were to be compromised by whatever means?

Scalability can use the same transaction-processing "dispatcher," database "sharding" or website "load balancing" techniques employed by systems with large user bases today, like Visa or the major social media platforms. This is because each transaction coming into the central bank can be recognized almost immediately as relating to one of several separate servers that can fully process it, and it can immediately be dispatched directly to such a server. Fundamentally, what makes this possible is that the transactions processed can be kept independent of each other. The result is what may be considered, at least in principle, orders of magnitude more efficient when compared to solutions requiring every transaction to result in consensus of many nodes on a single blockchain. (By contrast, the xx network's approach to mixing allows the security benefits of the large number of nodes in its network, but only requires replication of the computation by five nodes and offers end-to-end latency of roughly two seconds.)

Since the solution is software-only and its use of cryptography modest, the cost of processing an individual transaction can be low. Performance is also not an issue: computers of the 1990s were able to handle the transaction speeds and database sizes in the production eCash systems. The valid coins are stored only until spent. Since transactions are essentially independent of each other, the amount of additional processing power and bandwidth needed grows by the same amount for each additional spend or deposit transaction per second. This additional power is simply achieved by adding more hardware and sharding; and with so-called consistent hashing, hardware additions need not be disruptive. Any underlying database technology can be used, whether conventional or distributed.

Payments can be urgent, withdrawals less so. Each payment has one or more digitally signed "serial numbers" (called x elsewhere here) and so these parts can in principle be checked for "double-spending" by separate portions of the network. No network can withstand unlimited attack. But if the network can be divided into parts, and each part can process some portion of the transactions' serial numbers, then transactions can be routed to the parts that can handle their serial numbers as mentioned. This provides for a kind of graceful degradation of service, compared to an all-or-nothing failure, and can take advantage of geographically distributed servers.

Withdrawals may not be extremely urgent, but they provide the bedrock security against counterfeiting. They can be made a matter of record and available to the account owner, so that the owner can recover their money from their private key. But otherwise, this data should be protected doubly, by the commercial bank and the central bank. After double encryption, for instance, the data can be backed up in multiple media and locations.

If the central-bank signing key(s) is ever compromised, such as by a quantum computer, a physical attack on data-center vaults, or perhaps some new algorithm, the combined double-spending and quantum-security database detailed with reference to Figure 5 above will prevent counterfeits from being accepted.

Conclusion

A retail CBDC should preserve at least low-value cash-like transactions as a privacy-friendly commons under citizens' individual control. With eCash 2.0, central banks can provide the privacy consumers have shown they care deeply about, while preventing large-scale abuse, with all the advantages of a state-of-the-art CBDC and quantum-resistant security against counterfeiting.

References

[1] Boar, C. and Wehrli, A. (2021). Ready, steady, go?: Results of the third BIS survey on central bank digital currency, BIS Papers No. 114.

[2] G7. Public policy principles for retail central bank digital currencies (14 October, 2021).

[3] G7. Finance Ministers and Central Bank Governors Statement on Central Bank Digital Currencies (CBDCs) and Digital Payments (13 October 2021).

[4] People's Bank of China. The progress of research & development of E-CNY in China (July 2021), 3.2.5.

[5] Bank of Canada, Bank of England, Bank of Japan, European Central Bank, Federal Reserve, Sveriges Riksbank, Swiss National Bank, and BIS. Central bank digital currencies: System design and interoperability (September 2021).

[6] World Economic Forum, Privacy and confidentiality options for central bank digital currency (November 2021) citing European Central Bank Eurosystem. Eurosystem report on the public consultation on a digital euro. (April 2021).

[7] Bank of England. Responses to the Bank of England's March 2020 discussion paper on CBDC. (June 2021).

[8] Eurosystem report on the public consultation on a digital euro, op. cit., p. 19.

[9] ibid., p. 4 .

[10] See https://chaum.com/ecash/ (webpage with a timeline and documents of the development and implementation of eCash 1.0 by Digicash).

[11] Chaum, D., Fiat, A., and Naor, M. Untraceable electronic cash (extended abstract) Presented at CRYPTO 88 (February 1990).

[12] Chaum, D. (forthcoming paper) Offline eCash 2.0: Robust offline payments onlineable later. .

[13] This technology, developed by the first named author, allows cryptocurrencies to be traded securely between any two chains peer to peer, without smart contracts or any third-party intermediary. See https://liquifinity.com/.

[14] Chaum, D. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, vol. 24 no. 2, February 1981. (Also as UCB/ERL M79/9 22 February 1979.)