Offline Euro POC
— Student Project —

Leon Kempen
Delft University of Technology
Delft, The Netherlands
L.M.Kempen@student.tudelft.nl

Abstract— **Digital payments are becoming more important in the current financial system. However, this increases the critical dependency of an online service that can be used to perform and validate transactions. Whenever this service cannot be contacted the digital payment systems are useless. This scenario will occur more frequently as climate change increases the likelihood of extreme weather, leading to more power outages. Another problem with the current digital payment options is that people desire more privacy regarding their financial decisions. This can be seen in the rise in popularity and adoption of cryptocurrencies. Both issues can be solved with a CBDC that can be used in an offline scenario like cash. This thesis proposes a prototype implementation for the digital euro. The prototype uses EBSI for passport-grade digital identification and zero-knowledge proofs for privacy-protecting transactions. The protocol offers offline transactions without an active third party and with retroactive double-spending detection. The protocol has a working proof of concept showcasing its usability and demonstrating its functionality.**

I. Introduction

For the past decade, the share of digital payments has increased and the number of cash payments has declined [1]. However, the dependency on having a connection to an online infrastructure during the transaction has also increased. When you pay at a store with a debit or credit card, a connection to your bank is needed to verify whether you have enough balance to pay for the goods. Additionally, the money must also be transferred from the account of the payer to the account of the payee.

Other digital payment options, such as most cryptocurrencies, have the same dependency on being connected. In the case of Bitcoin [2], a connection to the ledger is needed to verify whether the transaction is included in the global blockchain.

The result of these dependencies on online infrastructures is that they are unusable whenever they cannot be reached. This could for example be in regions with no Internet coverage, when the servers of a bank are down or during a power outage.

The number of outages has increased for the past years [3] and it is expected that the likelihood of power outages will increase in the future [4, 5]. A significant share of these outages are caused by extreme weather events, such as heatwaves, blizzards, hurricanes and floods [6, 7, 8, 9].

Due to climate change, the likelihood and extremity of these weather events have increased [10, 11, 12], which could cause more frequent outages. To have a digital payment option available during those conditions, the transaction must be possible in an offline manner. This implies that no other party but the payer and payee can be involved during the transaction.

Another issue with the current digital payment methods is that they are not privacy-protecting. The bank has a complete list of all transactions involving the account holders and their balances. In case of a breach, this data could be abused.

For most cryptocurrencies, transactions are stored in a public ledger, using a wallet address as a pseudonym. Some of those cryptocurrencies, like Ethereum [13], users have a fixed wallet address. If you know which address belongs to someone, the transactions executed with that wallet can be traced. For other cryptocurrencies like Bitcoin [14] it is feasible to change the wallet addresses with every transaction. However, an address becomes tainted with each transaction and can be tracked with a taint analysis [15].

Another digital payment option that could be used offline and with more privacy is electronic cash (e-cash). Depending on the protocol, e-cash has similar properties to physical cash. Comparable to regular cash, a user must first withdraw money from the bank. With e-cash, this money is represented as a digital token and can be stored on a device. At a later stage, the holder can spend the token(s) by transferring the tokens to the receiver. Finally, the receiver can deposit the tokens at the bank to redeem the value of the tokens.

In an offline scenario, no bank, ledger, or other third party is involved in the transaction between the spender and the receiver. Therefore, the transaction can be executed in an offline manner.

Many Central Banks have expressed their interest in e-cash and some Central Banks are providing digital versions of their currencies as e-cash. These digital versions of currencies backed by a Central Bank are named Central Bank Digital Currencies (CBDCs). In December 2023, 130 countries, contributing to 98% of the global GDP, have expressed their interest in a CBDC, are researching and developing it, or have a CBDC in circulation [16]. Examples of CBDCs in circulation are: *e-Naira* (Nigeria), *Sand Dollar* (The Bahamas) and *JAM-DEX* (Jamaica). Several CBDCs of countries in the G20 that are currently in the pilot phase are: *Digital Yen* (Japan), *e-CNY* (China) and *eAUD* (Australia).

However, a survey from the International Monetary Fund (IMF) [17] found that most CBDCs in development can only be used online. The ones that can be used offline typically rely on tamper-resistant hardware to maintain the integrity of the CBDCs stored on a device. As Liu et al. [18] and Lee et al. [19] have shown, even the current state-of-the-art tamper-resistant, secure hardware can be breached. Therefore, the design of the CBDC must rely on established cryptographic protocols to maintain the system's integrity, rather than 'tamper-resistant' hardware.

Currently, the European Central Bank (ECB) is in the preparation stage of designing the Digital Euro [20]. Two of the main design goals of the Digital Euro are protecting privacy as much as possible and support for offline transactions [21].

This thesis proposes a design for the Digital Euro, that fulfils these goals. The system relies on zero-knowledge proofs to transfer Digital Euros between users. As those proofs embed the identity of users in a hidden way, the participants can not identified by other users or banks.

The anonymity of users is further protected by the integration of the European Blockchain Services Infrastructure (EBSI) to handle digital identity, users can act under a passport-grade key pair that works as a pseudonym during transactions. During those transactions, there is no need for a connection to the bank or other party to verify the legitimacy of the Digital Euro or the participants. This makes it possible to transfer euros offline in areas with no network coverage or during a power outage.

II. Problem description

One of the main problems with offline e-cash is the balance between privacy and fraud prevention. On one side, offline e-cash transactions should provide anonymity and be untraceable. However, they are more prone to malicious actions since no central party can be reached to verify transactions. One of those actions is the act of double-spending.

Receivers of an e-cash token cannot check if the same token is spent in an earlier transaction. Therefore, in a fully anonymous setting, malicious users could freely duplicate e-cash and spend the tokens at different places. In the literature, there are two ways to mitigate double-spending.

Several e-cash schemes, such as [22, 23, 24], prevent double-spending utilizing secure and tamperproof hardware or software. Those implementations rely on the hardware or software to remove or mark a token used after the transaction. However as Liu et al. [18] and Lee et al. [19] have proven, such hardware and software are not fully secure and tamperproof and can thus be breached. This allows malicious users to freely double-spend their e-cash.

The other solution relies on cryptographic principles to detect double spending and revoke the anonymity of the malicious user. This often occurs when the e-cash is deposited, such as in [25, 26, 27, 28, 29, 30].

Another challenge in offline e-cash is the property of transferability. e-cash is in most schemes not transferable, meaning that a token can only be used for one transaction. After that transaction, the receiver must deposit at the bank and cannot use it for another transaction. This implies that during a longer period in which the bank cannot be reached the number of transactions is limited by the number of e-cash circulating.

On the other hand, transferable e-cash can be used in multiple transactions like physical cash. Whenever someone receives e-cash it can be reused for the next transaction. This reduces the dependency on the infrastructure of the bank. Furthermore, it allows for a more efficient implementation of e-cash with multiple denominations since users can use the change they receive in future transactions. However, the downside of transferable e-cash is that every transaction must be included with the e-cash to detect double-spending. This implies that the size of the e-cash grows with every transaction [31]. This also makes hiding the identity of spenders more complex.

Some e-cash schemes [28, 29] are based on a combination of several difficult cryptographic principles, making them efficient and powerful, but also very complex and hard to understand. Given that simplicity [32] can be a key factor in generating trust in a system, the protocol of the Digital Euro must be transparent and understandable. This trust in the system could play a vital role in the adaptation of CBDCs by the masses.

III. Related Work

A. *Evolution of offline e-cash*

In 1983 e-cash was first introduced by Chaum [33], with the creation of blind signatures. Blind signatures can be used to construct a signature for a message, while the signer does not know the content. Others can verify the message's signature using the signer's public key.

This concept was used by Brands [25] to create the first offline e-cash protocol in 1994. In the protocol, the user creates a token that embeds his identity in a hidden way. This token is then blinded and sent to the bank for a blind signature. As the token is blinded before it is sent to the bank, the bank cannot link the token to the user trying to withdraw. The token's signature makes it impossible for users to create valid e-cash without the bank.

When spending a token, the spender sends two variables used in the withdrawal with their signature to the receiver. After verification, the receiver can compute a challenge based on the received variables with a unique identifier. This challenge is used to generate a proof of transaction from the spender, as solving this challenge requires knowledge of the token that only the spender has.

The combination of the variables first received, the challenge and the response are a transcript of the transaction and can be used to deposit the token at the bank. The bank checks if the token's identifier is already in the database. If not, the bank stores the transcript in the database. If the identifier was already in the database, the bank has to take action to identify the double spender. The double-spending could result from two actors, the spender and the receiver.

In the latter case, the token's receiver tries to deposit the same token twice. This can be detected trivially as the two transcripts are equal. This could only happen if the same challenge is used twice, which is impossible as the challenge is dependent on a unique identifier.

When the two transcripts differ, the same token must have been used in two transactions. The bank can reveal the identity embedded in the token and thus identify the double spender. The two transcripts can be used as proof of double-spending when the bank takes legal action.

Brands [25] also stated that tamper-proof hardware to prevent double-spending can be used as an extension of this scheme. This hardware would make it harder to double-spend tokens, making it less likely to happen. However, if the hardware is breached and users double-spend their tokens, the bank could reveal their identity like before.

In 2005, Liu et al. [27] designed an e-cash scheme that supports recoverability. They deemed recoverability important as there are several ways that someone could lose their e-cash. This could for example be when the database or file in which the e-cash is stored is corrupted. Moreover, e-cash would be lost whenever the device on which the e-cash is stored is lost.

As the e-cash scheme is untraceable, adding recoverability is complex. This is because the bank can't determine how much e-cash someone holds or has spent before losing access to the e-cash.

Liu et al. extended the Brands' scheme by adding a *Recovery Center* (RC). This RC could be used to recover e-cash. The user would send their tokens to the RC after withdrawing them. The RC responds with two signatures, one is used as a proof of registration and the other for recovery.

When a user wants to recover a token, he must identify himself at the bank and the RC and show the second signature. The bank refunds the tokens if the token(s) have not been deposited yet. The RC will then store the signature of the token to a blacklist. Additionally, the RC forwards this signature to all users to notify them that the token is invalid.

Upon receiving a token, users must now also check if the token is not added to the blacklist. This leads to a scheme that is computationally heavy and requires all users to maintain a blacklist of all recovered tokens. The scheme would also require an online connection between the RC and the users to keep the blacklist up to date.

In 2010, Juang [23] improved the scheme of Liu et al. by proposing a scheme using digital pseudonyms and bilinear pairings. Instead of using an RC, the users receive a tamper-proof smart card when registering at the bank. When a user withdraws a token, a partially blind signature is created and the blinding factors are sent to an auditor.

Together with the bank and the auditor, the users can reconstruct a token identical to the token to be recovered. In this process, the user would not have to identify himself to the bank or the auditor, providing more anonymity. There is also no need for a blacklist of tokens since the reconstructed token is identical to the recovered token.

Juan [22] also found other potential issues in Brands' scheme in 2005. For double spending detection, the bank has to maintain a large database with all transactions. The other issue is that the bank could issue additional tokens if they are malicious.

To mitigate these problems, Juan proposed *AOMPS*. In this scheme, multiple parties are assigned to issue e-cash. Instead of one blind signature, a blind threshold signature scheme (BTSS) is used. A user can construct a blind signature on a token if he received at least the threshold number of signatures from the group of signers in BTSS. The blind signature can then be verified using the public key of the group of signers.

However, this scheme only prevents double-spending by relying on tamper-proof hardware. Moreover, the promised storage reduction is achieved by letting the receivers store their transactions. Combined with the reliance on hardware, double spending would not even be detected in this scheme if the hardware is broken.

In 2011, Eslami and Talebi [34] proposed a different solution to solve the storage problem, token expiration dates. In a scheme with expiration dates, a specific future date is included in the representation of the token. If this date has passed the token is no longer valid.

The bank now has to offer an exchange service to handle token expiration. Tokens that have expired can be exchanged for new ones through this service, implying that the bank must keep track of exchanged and deposited tokens.

In 2013, Baseri et al. [35] found multiple problems in the scheme of Eslami and Talebi and proposed a new scheme that solved the issues.

This scheme was further improved by Fan et al. [26] in 2014. They made the exchange steps more efficient and included a deposit date to calculate how much interest a depositor would receive.

However, the question remains if these schemes solve the problem of storage to detect double-spending. Having the option to recover expired e-cash will not lead to a decrease in transactions and thus a decrease in the number of deposits. This means that the size of the deposit table will not be affected by adding an expiration date. Tokens that have been deposited and expired after can not be removed from the storage, because they are needed to check if a token has been spent when it is sent for exchange. Furthermore, by offering an exchange service for expired tokens, the bank should store the exchanged tokens leading to a larger required storage to detect double-spending.

The first transferable offline e-cash was proposed in 2015 by Baldimtsi et al. [28]. They used malleable signatures proposed by Chase et al. [36] for this. Baldimtsi et al. state that a non-transferable token can be described as $(SN||\sigma||DS)$. In this description, SN is the token's serial number, σ the bank's signature and DS information which can be used to detect double-spending and to revoke double-spender's identity.

In the scheme of Baldimsti et al., this description is extended with tags, in which every transaction generates a new DS tag and a new SN. For example, after k transactions, the token can be described as $(SN_1..SN_k||\sigma_k||DS_1..DS_{k-1})$.

σ_k is the malleable signature on SN_k and DS_{k-1}.

The bank can identify double spending when it detects two tokens with the same SN_i tag but different DS_i tags. As the identity of the spenders is embedded in the SN_i tags, the bank can use the two DS_i tags to identify the double-spender.

In 2021, Bauer et al. [29] found that using malleable signatures was inefficient. They improved the scheme by using a commit-and-prove scheme instead of malleable signatures. In a transaction, the token is updated with a commit tag, an encryption of the tag and proof values to show that the commit tag is encrypted correctly. Additionally, the commits randomize the token's structure and encryption. Therefore the inefficient malleable signatures are no longer needed.

Jianbing et al. (2023) [30] noted that all the previous e-cash schemes require the payee to identify himself during the transaction. Therefore they proposed a protocol that provides dual anonymity. This guarantees that the identity of the payer and payee remain hidden during the transaction or when the tokens are deposited.

Before a transaction, the receiver can prove that he is a verified user using a zero-knowledge proof. After that, the payer can generate a transaction identifier and create a traceable tag. The receiver of the token can compute a receipt of the transaction. This receipt can be used to deposit the token at the bank or re-randomize the token with the help of the bank.

Even though Jianbing et al. claim that the scheme is transferable, tokens can only be spent multiple times after it is randomized after each transaction. This implies that the bank has to be contacted after each transaction, making the transferable aspect of the scheme significantly less useable.

B. Eurotoken

Blokzijl [37] and Koning [38] did earlier work regarding a CBDC, named Eurotoken, that the EU could use. Initially, the bank mints a token by defining a serial number, a face value and a nonce. Upon withdrawal, the bank sends the user the minted token, a tuple of the receiver's public key and a signature of the bank on the minted token and the receiver's public key.

The signature tuple is the start of a chain of proofs of ownership. This chain of ownership is sent with the token and is extended with each transaction. As the bank's signature includes the withdrawer's public key, the withdrawer can prove he owns the token. When the user spends the token, the user will send the token and extend the chain of ownership with a tuple of the receiver's public key and a signature, singing the previous proof of ownership and the recipient's public key. The deposit of the token is similar to a transaction between users. However, now the bank is the receiver of the token.

Token holders can verify the chain of ownership after k transactions starting from the bank's signature. This signature can be used to find the public key of the first receiver. The found public key can then be used to validate the next proof and to find the next recipient's public key. After k transactions the last found public key maps to the current holder of the token.

The bank can detect double spending upon deposit of the tokens. Whenever the bank has received two tokens with the same first proof double spending must have occurred. The bank can then compare the chain of proofs of ownership to find the double spender. After some i proofs there must be two proofs where proof $i+1$ from the first chain differs from proof $i+1$ from the second chain. This implies that proof i is used in two transactions and thus doubly spent. The identity of the double spender can then easily be found, as that is the receiver's public key used to create proof i.

The problem with this proposal is that it offers no privacy and the token's history is fully traceable. Whenever someone receives a token, all the public keys of the previous holders can be found. Malicious people who know which public keys map to which identity could use and abuse that information to obtain sensitive personal information. Moreover, all transactions are visible to the bank. This makes it possible for the bank to construct a graph which can be used to trace the payment system.

Privacy is an important factor in why people use cash for payments [39]. The current implementation of Eurotoken offers less privacy than the online payment infrastructure of banks. This combined will have a detrimental effect on the adoption rate of the CBDC, as the bonus of paying offline will cost you your privacy. Moreover, the provided protocol does not align with the main design goal of the ECB, namely privacy protection [40].

IV. Security assumptions

A. Discrete Logarithm problem

The Discrete Logarithm Problem (DLP) states that given a finite cyclic group G, generator $\langle g \rangle$ of G and $h \in G$, it is hard to find an integer a, such that $g^a = h$.

B. Decisional Diffie-Hellman assumption

V. Building Blocks

A. Blind Signatures

Chaum [33] first introduced blind signatures in 1983. A blind signature scheme can be used to obtain a valid signature on a message M, without the signer knowing the exact content of M. This makes it possible for e-cash to have a valid signature of a bank for an unknown token. When this token is deposited later, the bank cannot recognize which user has withdrawn the token. This makes it impossible for the bank to link the user who withdrew the token to the user who deposited it, proving more anonymity.

In this thesis, an implementation of the Blind RSA Signature is used. However, any blind signature protocol could be used. A blind RSA signature is obtained as follows:

1) The signing party generates RSA parameters e, d and N and publishes d and N. Additionally, the signing party also publishes a hash function H.
2) The client then picks a random blinding factor r and calculates e^r.

3) With that the client computes the blinded message M' for message M to sign: $M' = H(M)e^r \mod N$, and sends M' to the signing party.
4) The signing party then signs the blinded message as: $\sigma' = M'^d \mod N$ and returns σ'.
5) To obtain the signature on message M the client computes: $\sigma = \sigma'^{-r} \mod N$.
6) Other parties can verify the validity of σ by checking: $H(M) \overset{?}{=} \sigma^e$.

A more formal protocol description can be found in Figure 1.

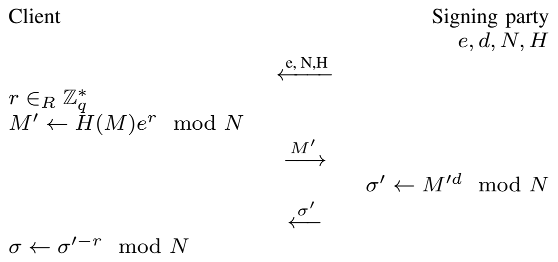
Client
Signing party
e, d, N, H
$\xleftarrow{\text{e, N,H}}$
$r \in_R \mathbb{Z}_q^*$
$M' \leftarrow H(M)e^r \mod N$
$\xrightarrow{M'}$
$\sigma' \leftarrow M'^d \mod N$
$\xleftarrow{\sigma'}$
$\sigma \leftarrow \sigma'^{-r} \mod N$

Fig. 1: Blind RSA signature protocol to obtain a signature σ on message M

The blind signature is done over the hash of the message to prevent malicious clients from creating more valid signatures from an earlier received signature. Without the hash, malicious clients could also compute valid signatures on multiples of message M, due to the multiplicative homomorphic property of RSA.

Given that the hash function is collision-resistant, it is hard for a malicious client to find the message corresponding to the malled signature. Therefore it is impossible to create more valid signatures, based on an earlier received signature.

B. Bilinear Map

A bilinear map e is an operation that takes two elements from, potentially, different elliptic curve groups of order p and maps them to an element of a third group, the target group. More formally, given source groups G, H and target group G_T, a bilinear map is denoted as:

$$e : G \times H \rightarrow G_T$$

Additionally, the pairing must satisfy the following three properties:

- **Bilinearity:** For all items $P, Q \in G$ and $R, S \in H$, the following holds:

$$e(P + Q, R) = e(P, R) \cdot e(Q, R)$$
$$e(P, R + S) = e(P, R) \cdot e(P, S)$$

  Moreover, given generators g, h such that $G = \langle g \rangle$ and $H = \langle h \rangle$, for all $a, b \in \mathbb{Z}_{|}$ the following holds:

$$e(g^a, h^b) = e(g, h)^{ab}$$

- **Non-degeneracy**: $e(P, R) \neq 1$.
- **Efficient computability**: There must be an efficient method to calculate the pairing efficiently.

An extended bilinear map E is a mapping of two elements of G and two elements of H to four elements of G_T:

$$E : G^2 \times H^2 \rightarrow G_T^4$$

As an example, given $g_1, g_2 \in G$ and $h_1, h_2 \in H$:

$$E\left(\begin{pmatrix} g_1 \\ g_2 \end{pmatrix}, \begin{pmatrix} h_1 & h_2 \end{pmatrix}\right) = \begin{pmatrix} e(g_1, h_1) & e(g_1, h_2) \\ e(g_2, h_1) & e(g_2, h_2) \end{pmatrix} \quad (1)$$

Similarly to regular bilinear maps, the extended bilinear maps are also bilinear, using entry-wise product operations for the vectors and matrices. Given $g_1, g_2, g_3, g_4 \in G$ and $h_1, h_2 \in H$:

$$E\left(\begin{pmatrix} g_1 \\ g_2 \end{pmatrix}\begin{pmatrix} g_3 \\ g_4 \end{pmatrix}, \begin{pmatrix} h_1 & h_2 \end{pmatrix}\right) = E\left(\begin{pmatrix} g_1 \\ g_2 \end{pmatrix}, \begin{pmatrix} h_1 & h_2 \end{pmatrix}\right) E\left(\begin{pmatrix} g_3 \\ g_4 \end{pmatrix}, \begin{pmatrix} h_1 & h_2 \end{pmatrix}\right)$$

C. Groth-Sahai Proofs

In 2008, Groth and Sahai [41] presented a proof framework that can be used to efficiently create non-interactive zero-knowledge (NIZK) proofs and non-interactive witness-indistinguishable (NIWI) proofs. Before this, NIZK proofs used to be very efficient and thus not useable. The Groth-Sahai (GS) proofs are designed to prove statements in pairing-based equations.

As a setup, a (trusted) party must publish an asymmetric bilinear pairing description and a Common Reference String (CRS). The asymmetric bilinear pairing description is defined as:

$$(G_1, G_2, G_T, p, e, g_1, g_2)$$

in which G_1 and G_2 are two different bilinear groups of order p. These groups have a mapping e to target group G_T. g_1 and g_2 are generators of respectively G_1 and G_2.

The CRS is constructed with two pairs of four random group elements, four from G and four from G_2 and is defined as:

$$CRS = (g, u, g', u', h, v, h', v')$$

Depending on the structure of the GS proofs, the CRS can be used in a trapdoor function. In some structures, this will reveal the input. However, in other structures, no secret information can be found. The setup can be done with public randomness and multiple parties to fully remove the trust needed in a (central) party.

Each proof consists of three parts, namely the target T, the commitment values c_1, c_2, d_1, d_2 and proof elements $\theta_1, \theta_2, \pi_1, \pi_2$. The target represents the value that the prover wants to prove. The commitment values are used to randomized encryptions of values with which the proof is constructed. Elements from G_1 are encrypted in c_1 and c_2, whereas elements from G_2 are encrypted in d_1 and d_2. Lastly, the proof elements are used to derandomize the commitment values without revealing the exact values.

A full proof can be verified with an equation similar to equation 2:

$$E\left(\begin{pmatrix}c_1\\c_2\end{pmatrix},(d_1,d_2)\right)\overset{?}{=}E\left(\begin{pmatrix}g_1\\u\end{pmatrix},(\pi_1,\pi_2)\right)E\left(\begin{pmatrix}\theta_1\\\theta_2\end{pmatrix},(g_2,v)\right)\begin{pmatrix}1&1\\1&T\end{pmatrix} \tag{2}$$

More specifically, the verification can be done elementwise after expanding the extended bilinear maps as in equation 1. For example, to verify $e(c_1, d_1)$, the following must hold:

$$e(c_1,d_1)\overset{?}{=}e(g_1,\pi_1)\cdot e(\theta_1,g_2)\cdot 1$$

In this thesis, the implementation of the Groth-Sahai proofs is as follows. The equation to prove is $e(X,Y) = T$ in which $X \in G_1$ and $Y \in G_2$ and T is the target of the proof. The commitment values are randomized with values $r, s \in Z_p$, and computed as:

$$\begin{array}{ll} c_1 = g_1^r & d_1 = g_2^s \\ c_2 = u^r X & d_2 = v^s Y \end{array}$$

The prover now picks a random value $t \in Z_p$ and computes the proof elements as:

$$\begin{array}{ll} \pi_1 = d_1^r g_2^t & \theta_1 = g_1^{-t} \\ \pi_2 = d_2^r v^t & \theta_2 = X^s u^{-t} \end{array}$$

The full proof is now defined as $(c_1, c_2, d_1, d_2, \pi_1, \pi_2, \theta_1, \theta_2)$ and can be verified by others with equation 2. If someone knows the exponents used to create u and v from the CRS, one could find the committed values of X and Y. Let $u = g_1^\alpha$ and $v = g_2^\beta$, the committed values can be retrieved with the equations 3a and 3b.

$$X = c_1^{-\alpha} c_2 \tag{3a}$$

$$Y = d_1^{-\beta} d_2 \tag{3b}$$

VI. System Overview

The protocol is divided into four phases: Initialization, withdrawal, transactions and deposit. The initialization phase is executed only once by the trusted third party (TTP) and the users. The other three phases are related to the cycle of a single digital euro.

A. Initialization

In the initialization phase, the TTP responsible for managing identification publishes a bilinear pairing description and a common reference string (CRS), as found in section V-C. The exponents used to generate the group elements are stored for later use by the TTP but remain private. The participants in the protocol will use the bilinear pairing description and CRS.

Every participant has to register at the TTP as well. Upon registering the user picks a random private key x, calculates the public key $X = g_1^x$ and registers X at the TTP.

The user can register at a bank with the public key, certified by the TTP. The EBSI identification service can be used to prove the user's identity. The bank can use this public key to keep track of the user's balance.

B. Withdrawal

At the start of the withdrawal phase, the user can prove his identity to the bank in the same way as during the initialization phase. After that, the user generates a serial number (SN) and collaborates with the bank to obtain a blind signature from figure 1 of that serial number. Upon returning the blind signature, the bank deducts the euro from the user's balance.

C. Transactions

Every transaction the digital euro has undergone must be stored with the euro to combat double-spending. To find the user that double-spent a euro, the details of the malicious transaction must be known to retrieve the identity of the double-spender, as shown in [31]. This scheme stores the required information as a Groth-Sahai (GS) proof. By storing the information in a zero-knowledge proof, participants in later transactions, or the bank, cannot deduce any information related to the transaction from the proof. They can, however, verify if the proofs and thus the transactions are valid.

A digital euro is described as:

$$(SN, \sigma_{sn}, GS)$$

in which, SN is the serial number of the digital euro, σ_{sn} is the signature of the bank on SN and GS is an ordered list of GS proofs of previous transactions. Upon withdrawal GS is empty.

During a transaction, the spender and the receiver collaborate to create a GS proof, which is stored with the digital euro.

To start a transaction the receiver generates a random t and sends the randomization elements g_2^t, v^t, g_1^{-t} and u^{-t} to the spender, whilst keeping t secret. This prevents the spender from deciding on all randomness and trying to obfuscate double-spending by using the same randomness for two transactions with the same digital euro. Furthermore, t is used to prove knowledge of the randomization elements used in the previous transaction, as the t, will be used to determine randomization in the next transaction. The spender will use these randomization elements when creating the GS proof for the transaction.

The target of the proof, T, depends on whether the digital euro is spent earlier. When the euro has not been spent before, the target is $T = e(g_1, g_2)^\sigma$. Otherwise, after i transactions the target can be computed as $T_i = e(g_1, g_2)^{T_{i-1}}$. This way, the targets of the proofs can be used to describe a chain of transactions, in which the current proof links to the previous proof.

With this target, the spender can compute $y = \frac{T}{x}$ and $Y = g_2^y$, in which x is the spender's private key. The spender can now use the GS proof, to prove $e(g_1^x, g_2^y)$. Note that g_1^x is equal to the spender's public key. Additionally, due to the property of bilinearity, $e(g_1^x, g_2^y) = e(g_1, g_2)^{xy} = e(g_1, g_2)^T$.

The value of s in the proof is set to the inverse of t_{prev}, the t used in the previous transaction to provide the randomization elements. This implies that the spender must know the value of t used in the previous transaction and cannot generate a valid proof if he does not.

The spender sends the values of v^s and Y together with the proof elements to the receiver. With these, the receiver can verify the proof, if $e(X, Y) = T$ and check if d_2 is constructed correctly.

Additionally, the receiver must check if the previous proofs included with the digital euro are correct and verify the links between the proofs. Given the proofs for transaction $i-1 = j$ and i as:

$$(c_{1j}, c_{2j}, d_{1j}, d_{2j}, \theta_{1j}, \theta_{2j}, \pi_{1j}, \pi_{2j}, T_j)$$

and

$$(c_{1i}, c_{2i}, d_{1i}, d_{2i}, \theta_{1i}, \theta_{2i}, \pi_{1i}, \pi_{2i}, T_i)$$

the equations 4a and 4b must hold:

$$T_i \overset{?}{=} e(g_1, g_2)^{T_j} \tag{4a}$$

$$e(\theta_{1j}, d_{1i}) \overset{?}{=} e(g_1, g_2)^1 \tag{4b}$$

Equation 4b must hold to verify that every spender knew the randomization element t in the previous transaction. As g_1 and g_2 are part of the bilinear pairing description and thus constant, the equation expands to $e(g_1^{-t_j}, g_2^{s_i})$, which is equal to $e(g_1, g_2)^{-t_j s_i}$. For the transaction to be valid s should be the inverse of t of the previous transaction, implying that $-t_j s_i = 1$. This results in the verification form $e(g_1, g_2)^1$.

D. Deposit

A euro can be deposited to the bank in the same way as a euro is transferred between users in section VI-C. However in this case the bank is the receiver. As the user that wants to deposit the euro has to share their public key, the bank knows to which account the balance should be added. The bank also checks if the digital euro is doubly spent or not.

E. Double spending detection

The bank detects double spending when two digital euros DE and DE' with the same signature σ_{sn} are deposited. There are two possible scenarios in this case.

The first trivial case is when GS of DE equals GS of DE', excluding the last proof created in VI-D. This occurs if, and only if, the same user tries to deposit the same digital euro twice. To deposit the euro the user must identify himself, therefore the identity of the double spender is revealed.

In the second scenario, when GS of DE does not equal GS of DE', the bank must take additional actions to reveal the identity of the double spender. Given that the two lists of proofs are different, there must be an index i, such that $GS_{DE}[i] \neq GS_{DE'}[i]$. Assuming that the odds that the double spender retrieved the randomization elements generated by the same t are extremely unlikely, the proofs have, at least, different values for the θ_1 and θ_2 proof elements.

The bank can then send both proofs to the TTP. The TTP can extract the public key X with equation 3a, for both proofs and check if X is the same for both proofs sent by the bank. If they are the same, the TTP can retrieve the legal identity, registered with this public key, and return it to the bank. Otherwise, this transaction is no occurrence of double-spending. This could for example occur when the double spender did receive the same randomization parameters.

F. Efficiency Analysis

As mentioned earlier, the size of the Digital Euro must grow to detect double spending and revoke the anonymity of the double-spender. As seen in section VI-C, every transaction included in the Digital Euro is defined as a GS-proof. This means that the size of the Digital Euro grows with 8 or 9 group elements for each transaction. The number of group elements depends on whether the value of T is explicitly included in the proofs. Given that the target T can be calculated from the proof elements of the previous proof, it can be omitted for size optimizations. This means that the size of the Digital Euro after n transactions can be computed as:

$$size = |SN| + |\sigma_{SN}| + n \cdot 8|G|$$

in which $|SN|$ denotes the size of the serial number, $|\sigma_{SN}|$ the size of the signature of the bank on SN and $|G|$ the size of a group element.

VII. Implementation

The described protocol is implemented in Kotlin as a proof of concept. The implementation can be found [HERE]. The Java Pairing Based Cryptography (JPBC) library is used for group and bilinear map operations. As this is a proof of concept, it is not a fully implemented financial system and users can freely withdraw and deposit Digital Euros without affecting their balances. To mimic the current payment options, the prototype is built as a mobile application.

Offline data transfer between clients is implemented through NFC. In a real-world scenario, this method would still be viable when only local connections are possible. Alternatives for local data exchange, such as BlueTooth and local Wi-Fi, require more preparation for an existing connection. Data transfer through audio waves would not work when multiple transactions are done simultaneously in the same space or somewhere with significant background noises. This would for example become an issue at cash registers in supermarkets.

VIII. Limitations and future work

The current protocol relies on a TTP to revoke the anonymity of users in case double spending is detected. However, the TTP can revoke anyone's identity based on a single transaction. This makes it possible for a malicious TTP to fully trace transactions when it receives a Digital Euro with the full list of proofs. In most literature, the TTP requires two proofs of the double-spend transaction to revoke the user's anonymity. Even though this protocol offers more privacy and anonymity than the traditional banking system, a 'once concealed twice revealed' approach might be more desirable.

Such an approach might be feasible by using a different type of GS-proof. For example, by changing how the targets of the proofs are constructed. If it is possible to create the proofs such that two targets generated for the double-spending transaction would reveal the identity of the double-spender a commitment scheme that always hides the spender's identity can be used.

On the other hand, the ability to revoke the anonymity from one transaction also has legal advantages. When a perpetrator would only spend e-cash obtained through theft or a forced money transfer once, the perpetrator can be identified. The perpetrator would not be identifiable from a single valid transaction without this possibility.

To further protect users' privacy, the CRS used in the protocol can be constructed by a collaboration of multiple parties. The ability of a single party to revoke the anonymity of all users is then removed. To revoke the anonymity of users all parties are needed.

Another limitation of the protocol is that users can recognize e-cash, which they had before. The signature and transaction proofs are not randomized with each transaction. Therefore if a user notices that it had the same e-cash before, it is possible to gain some knowledge regarding the traceability of the e-cash. This knowledge allows the user to link the identity of the receiver of the earlier transaction to the identity from whom the user received the e-cash and the number of transactions in between. This linkability could be avoided by randomizing both the signature and transaction proofs for every transfer as is done in [28] and [29].

IX. Conclusion

This thesis proposes an offline transferrable e-cash scheme that could be used as a prototype for the CBDC of the ECB. The protocol is based on bilinear pairings through GS-proofs. Using these proofs, the identity of the users is encrypted into the commit values of the proof. However other users can only verify that the transactions are valid and cannot obtain information from the proofs. As every transaction with the same Digital Euro is linked to the previous one in two ways, malicious users cannot alter the proof history. Additionally, as the users must know a secret variable used in the previous proof to generate a new valid proof, users cannot spend Digital Euros which they did not directly receive.

The scheme relies on a TTP to handle the users' identities and to revoke their anonymity when needed. Whenever the bank detects double spending when receiving two tokens with the same serial number and signature, the TTP extracts the identity from the proofs. Even though the TTP only needs one proof to revoke the anonymity of users, the protocol gives more privacy towards the bank than the traditional banking systems and Eurotoken.

Another problem is that users can recognize digital euros they have had before. However, they can extract little information from this recognition. This problem could potentially be mitigated by randomizing the proofs with each transaction. However, this will increase the cryptographic complexity of the system, which could hurt the adoption rate.

The protocol also has a public proof of concept implementation. This implementation can be seen as a real-world example of how the system could be used. Additionally, the proof of concept also makes it easier to reason about bottlenecks and other potential problems in the system.

In conclusion, this protocol is a prototype for the digital euro, useable by the European Central Bank. The protocol offers a transferrable offline e-cash scheme and more privacy than current digital payment options or the earlier proposed solution. Therefore, including this implementation of the digital euro will enhance the digital payment ecosystem and make the economic system more durable in areas with low coverage or during power outages.

References

[1] DNB. *Use of cash lower in Euro Area Countries*. Dec. 2022. URL: `https://www.dnb.nl/en/general-news/dnbulletin-2022/use-of-cash-lower-in-euro-area-countries`.

[2] Satoshi Nakamoto. "Bitcoin whitepaper". In: *URL: https://bitcoin. org/bitcoin. pdf-(: 17.07. 2019)* (2008).

[3] Narayan Bhusal et al. "Power system resilience: Current practices, challenges, and future directions". In: *Ieee Access* 8 (2020), pp. 18064–18086.

[4] Adam X Andresen et al. "Understanding the social impacts of power outages in North America: a systematic review". In: *Environmental Research Letters* 18.5 (2023), p. 053004.

[5] ATD Perera et al. "Quantifying the impacts of climate change and extreme climate events on energy systems". In: *Nature Energy* 5.2 (2020), pp. 150–159.

[6] Laiz Souto et al. "Identification of weather patterns and transitions likely to cause power outages in the United Kingdom". In: *Communications Earth & Environment* 5.1 (2024), p. 49.

[7] J Schaller and S Ekisheva. "Leading causes of outages for transmission elements of the North American bulk power system". In: *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE. 2016, pp. 1–5.

[8] Stephen A Shield et al. "Major impacts of weather events on the electrical power delivery system in the United States". In: *Energy* 218 (2021), p. 119434.

[9] Joan A Casey et al. "Power outages and community health: a narrative review". In: *Current environmental health reports* 7 (2020), pp. 371–383.

[10] Peter Stott. "How climate change affects extreme weather events". In: *Science* 352.6293 (2016), pp. 1517–1518.

[11] Kristie L Ebi et al. "Extreme weather and climate change: population health and health system implications". In: *Annual review of public health* 42.1 (2021), pp. 293–315.

[12] Intergovernmental Panel on Climate Change (IPCC). "Weather and Climate Extreme Events in a Changing Climate". In: *Climate Change 2021 – The Physical Science Basis: Working Group I Contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2023, pp. 1513–1766.

[13] *Ethereum.* Accessed: 2024-03-04. URL: https://ethereum.org/en/.

[14] bitcoin.org. *Protect your privacy*. Accessed: 2024-03-04. URL: https://bitcoin.org/en/protect-your-privacy.

[15] Tin Tironsakkul et al. "Context matters: Methods for Bitcoin tracking". In: *Forensic Science International: Digital Investigation* 42 (2022), p. 301475.

[16] Atlantic Council. *Central Bank Digital Currency Tracker*. Accessed: 2024-03-04. URL: https://www.atlanticcouncil.org/cbdctracker/.

[17] John Kiff. *Taking digital currencies offline*. July 2022. URL: https://www.imf.org/en/Publications/fandd/issues/2022/09/kiff-taking-digital-currencies-offline.

[18] Weijie Liu et al. "Understanding TEE containers, easy to use? Hard to trust". In: *arXiv preprint arXiv:2109.01923* (2021).

[19] Jaehyuk Lee et al. "Hacking in darkness: Return-oriented programming against secure enclaves". In: *26th USENIX Security Symposium (USENIX Security 17)*. 2017, pp. 523–539.

[20] European Central Bank. *Where does the project stand?* Oct. 2023. URL: https://www.ecb.europa.eu/paym/digital_euro/timeline/html/index.en.html.

[21] European Central Bank. "A stocktake on the digital euro". In: *Eurosystem* (Oct. 2023).

[22] Wen-Shenq Juang. "A practical anonymous off-line multi-authority payment scheme". In: *Electronic Commerce Research and Applications* 4.3 (2005), pp. 240–249.

[23] Wen-Shenq Juang. "RO-cash: An efficient and practical recoverable pre-paid offline e-cash scheme using bilinear pairings". In: *Journal of Systems and Software* 83.4 (2010), pp. 638–645.

[24] Zhexuan Hong and Jiageng Chen. "A Solution for the Offline Double-Spending Issue of Digital Currencies". In: *International Conference on Science of Cyber Security*. Springer. 2022, pp. 455–471.

[25] Stefan Brands. "Untraceable off-line cash in wallet with observers". In: *Advances in Cryptology—CRYPTO'93: 13th Annual International Cryptology Conference Santa Barbara, California, USA August 22–26, 1993 Proceedings 13*. Springer. 1994, pp. 302–318.

[26] Chun-I Fan, Wei-Zhe Sun, Hoi-Tung Hau, et al. "Date attachable offline electronic cash scheme". In: *The Scientific World Journal* 2014 (2014).

[27] Joseph K Liu, Patrick P Tsang, and Duncan S Wong. "Recoverable and untraceable e-cash". In: *Public Key Infrastructure: Second European PKI Workshop: Research and Applications, EuroPKI 2005, Canterbury, UK, June 30-July 1, 2005, Revised Selected Papers 2*. Springer. 2005, pp. 206–214.

[28] Foteini Baldimtsi et al. "Anonymous transferable e-cash". In: *IACR International Workshop on Public Key Cryptography*. Springer. 2015, pp. 101–124.

[29] Balthazar Bauer, Georg Fuchsbauer, and Chen Qian. "Transferable E-cash: A cleaner model and the first practical instantiation". In: *IACR International Conference on Public-Key Cryptography*. Springer. 2021, pp. 559–590.

[30] Jianbing Ni et al. "Dual-Anonymous Off-Line Electronic Cash for Mobile Payment". In: *IEEE Transactions on Mobile Computing* 22.6 (2023), pp. 3303–3317. DOI: 10.1109/TMC.2021.3135301.

[31] David Chaum and Torben Pryds Pedersen. "Transferred cash grows in size". In: *Workshop on the Theory and Application of Cryptographic Techniques*. Springer. 1992, pp. 390–407.

[32] Johannes Strikwerda. "Simplicity and Complexity". In: *Organized Complexity in Business: Understanding, Concepts and Tools*. Springer, 2023, pp. 31–49.

[33] David Chaum. "Blind signatures for untraceable payments". In: *Advances in Cryptology: Proceedings of Crypto 82*. Springer. 1983, pp. 199–203.

[34] Ziba Eslami and Mehdi Talebi. "A new untraceable off-line electronic cash system". In: *Electronic Commerce Research and Applications* 10.1 (2011), pp. 59–66.

[35] Yaser Baseri, Benyamin Takhtaei, and Javad Mohajeri. "Secure untraceable off-line electronic cash system". In: *Scientia Iranica* 20.3 (2013), pp. 637–646.

[36] Melissa Chase et al. "Malleable signatures: New definitions and delegatable anonymous credentials". In: *2014 IEEE 27th computer security foundations symposium*. IEEE. 2014, pp. 199–213.

[37] Wessel Blokzijl. "EuroToken: An offline capable Central Bank Digital Currency". In: (2021).

[38] Robbert Koning. "Performance analysis of an offline digital Euro prototype". In: (2023).

[39] ECB. *The role of cash*. URL: https://www.ecb.europa.eu/paym/digital_euro/timeline/html/index.en.html.

[40] European Central Bank. "The case for a digital euro: key objectives and design considerations". In: *Eurosystem* (July 2022).

[41] Jens Groth and Amit Sahai. "Efficient non-interactive proof systems for bilinear groups". In: *Advances in Cryptology–EUROCRYPT 2008: 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings 27*. Springer. 2008, pp. 415–432.