Offline Euro POC
— Student Project —

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

***Abstract*— Lorem Ipsum**

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 in the future. 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, with each transaction, an address becomes tainted 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 integrity of the system, 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. Related Work

A. Double Spending

B. Evolution of e-cash

III. Building Blocks

A. Blind Signatures

Chaum [22] 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: $H(M) \stackrel{?}{=} \sigma^e$.

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

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.

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

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

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:

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

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:

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

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 [23] 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 equation 2:

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

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:

$$e(c_1, d_1) = e(g_1, \pi_1) \cdot e(\theta_1, g_2) \cdot 1$$

In this thesis, two implementations of GS proofs are used. In the first proof, 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:

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

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.

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

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

In the second implementation of GS-proof, the target is defined as $xy = T$, in which $x \in Z_p$ and $y \in Z_p$. With random values $r, s \in Z_p$ the commitment values are calculated as:

$$\begin{array}{ll} c_1 = g^r (g')^x & d_1 = h^s (h')^y \\ c_2 = u^r (u'g)^x & d_2 = v^s (v'g)^x \end{array}$$

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

$$\begin{array}{ll} \pi_1 = d_1^r h^t & \pi_2 = d_2^r v^t \\ \theta_1 = g'^{xs} g^{-t} & \theta_2 = (u'g)^{xs} u^{-t} \end{array}$$

Similar to the first type of GS-proof, the full proof is 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. However, if someone knows the exponents used in the CRS, it is impossible to find the values of x and y.

IV. System Overview

V. Conclusion

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

[23] 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.