GoldEuro: Offline e-cash survey
— Student Project —

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

I. Introduction

For the past seven years, the number of cash payments has been declining in the European Union [1]. According to the Dutch national bank, De Nederlandsche Bank, on average 79% of all payments in the Euro area were settled with cash in 2016. This dropped to 59% in 2022. Within the Euro area are member states to which this share is only 19% (Finland) or 21% (the Netherlands) of all payments. In other countries, the share is even lower, namely in China (18%) and South Korea (14%) [2]. Some expect that a cashless society will be reached eventually [3]. Nearly all the other payments are settled with a digital payment method, like a debit card or smartphone.

However, those types of payments are heavily dependent on being able to contact one or more financial institutions. These methods are obsolete and void when such a party can not be reached. For example, this could happen during outages, which occur more often and with an increasing duration [4].

Another concern with these digital payment methods is that they are not anonymous, unlike cash. Therefore, they require trust in a third party, like a bank, to handle the personal data confidentially and not use them for commercial gain [5]. Examples of this data are balance, transaction details and name and address details. Additionally, those parties must be trusted to secure the data adequately to prevent data breaches.

Other increasingly popular digital payment methods, such as cryptocurrencies [6], avoid this problem by providing (pseudo)anonymity in a decentralized network. However, they depend on the same condition that the ledger should be reachable during a transaction. This is needed to ensure that the transaction is valid and completed. Moreover, they are not regarded as legal tender in most countries.

As these cryptocurrencies are unregulated, they weaken the control of central banks on the economy. To provide a payment method with the advantages of cryptocurrencies, central banks have started developing their own Central Bank Digital Currencies (CBDC) [2, 7, 8, 9] or have expressed their interest in them. These CBDCs could be used as an alternative to cash, providing a digital payment method with the benefits of cash [10, 11].

Additionally, (central) banks, like the European Central Bank [12] and The People's Bank of China [13], also highly desire that the CBDC remains functional offline. This functionality would make it possible to execute transactions, even when an outage occurs or when the bank cannot be reached.

However, there is not (yet) a CBDC that satisfies the desired properties that the e-cash should have. This survey focuses on different solutions posed by literature regarding offline cash solutions, which could be seen as predecessors of the offline CBDC schemes. The survey is structured as follows: in section II the desired properties of an offline e-cash scheme are listed, section III covers the double spending problem found in (offline) digital currencies, section IV lists the different schemes found in literature and section V contains a conclusion.

II. Desired properties

For (offline) e-cash to be usable like physical cash, it should have the same benefits and properties as physical cash. These properties would ensure that the scheme is sound, secure and privacy-protecting. However, these properties combined pose a major technological challenge to all be included in a single scheme.

Unforgeability. It should not be possible for a user to create some e-cash that appears to be valid but is fake, in the name of the issuer with non-negligible probability in Probabilistic Polynomial Time [14]. For physical cash, unforgeability is ensured by physical measures, such as special paper, ultraviolet ink and holograms, that make forging coins and banknotes difficult [15].

Unlinkability. It should be infeasible to link any two payments executed by the same user, even when its identity is known unless the payments lead to double spending [16].

Anonymity. To ensure the user's privacy, it must be impossible to link a user to a transaction. This is known as *Weak Anonymity* (WA) [15]. *Strong Anonymity* poses that it should also be impossible to decide whether two transactions are initiated by the same user.

Transferability. Even though this is a less studied property in the literature [17], transferability is a highly desired property and benefit of physical cash. Transferability allows users to re-use the coins they received earlier to pay for something else or give change during a transaction. For digital cash schemes, transferability enables users to re-use their received e-cash without depositing and withdrawing the e-cash first.

III. Double Spending

Unlike online e-cash schemes, offline schemes do not have access to a trusted third party, or the entity responsible for issuing and retrieving e-cash. Therefore, it is impossible to verify whether a token has been spent already during a transaction. Additionally, e-cash tokens can be duplicated easily, as it is just digital data. Combined with its anonymity, it could be abused effortlessly. Therefore, measures should be taken to prevent or discourage malicious users from double spending their e-cash. Solutions posed for this problem in an online scenario, such as modifying the balance of an account in real-time [18] or by checking if a token is spent earlier [19, 20], are not possible in an offline scenario.

In offline e-cash schemes, there are two ways of handling the double spending problem. One way is ensuring that double spending can be detected later, for example, when depositing the coins. Another way is by using specific hardware or special (trusted) software.

The first of the two, identifying double spending, can be achieved whilst guaranteeing the anonymity and privacy of non-malicious users through cryptography. Several solutions, such as [21, 22, 23], make it impossible to find the identity of a spender when a token is spent only once. However, if the token has been spent at least twice, the spender's identity can be resolved from the token. As the identity can be revealed, the user can be held accountable. This makes it possible to punish or prosecute the malicious users, discouraging them from double-spending.

The second solution, based on hardware, makes it impossible to spend the same token twice and thus prevents double-spending. For this, one could use specific hardware that cannot be tampered with, such as a wallet [21, 24] or a secured chip integrated into a smart card [25, 18].

Alternatively, double-spending can be prevented by storing secret values in a Trusted Execution Environment (TEE) [26, 27]. Following the definition of Sabt et al. [28]: "A TEE is a tamper-resistant processing environment that runs on a separation kernel. It guarantees the authenticity of the executed code, the integrity of the runtime states (e.g. CPU registers, memory and sensitive I/O), and the confidentiality of its code, data and runtime states stored on a persistent memory."

Using the TEE, one could store a secret value needed to sign a token and delete it upon signing. Therefore, the token can only be signed and thus spent once. Another option would be to store the tokens in the TEE and remove or update them accordingly when they are used. However, preventing double-spending with either specialised hardware or a TEE heavily relies on its assumed security. This security could be broken [29, 30]. When it is breached and the scheme does not have the option to identify double-spending, users can freely copy and spend valid e-coins without repercussions. This would break the integrity of the entire scheme as users could generate unlimited money.

IV. Evolution of offline e-cash

Many researchers [22, 24, 25, 31, 32, 33] see the introduction of the concept of blind signatures by Chaum [34] as the foundation digital cash schemes. As they can be used to create anonymous, untraceable transactions. When creating a blind signature, the signer of the message does not know the content of the message. However, the message and the signature can be verified with the signer's public key [35].

In an e-cash scheme, a user can construct a valid token in collaboration with the issuer, which creates a blind signature of the token. This way, the issuer does not know the exact content of the token, providing anonymity and untraceability for the user. The token can be verified with the public key of the issuer by potential receivers of the token.

Brands [21] used this principle to design the first untraceable privacy-protecting offline e-cash scheme. Withdrawing a token is done through an interaction between the account holder and the bank. This process is shown in Figure 1. The bank first generates a random variable (w) from the generator group and creates two variables, a and b, of which b is constructed by using the identity of the account holder. With these and five other randomly generated variables, the account holder can construct token c' and create the blinded challenge c to send back to the bank. The bank can then sign the challenge and send it back to the account holder.

$$\begin{array}{lcr}
\text{Account holder} & & \text{Bank} \\
 & & w \in_R \mathbb{Z}_q \\
 & & a \leftarrow g^w \\
 & & b \leftarrow (Ig_2)^w \\
 & \xleftarrow{\text{a,b}} & \\
w \in_R \mathbb{Z}_q^* & & \\
A \leftarrow (Ig_2)^s & & \\
z \leftarrow z^s & & \\
x_1, x_2, u, v \in_R \mathbb{Z}_q & & \\
B \leftarrow g_1^{x_1} g_2^{x_2} & & \\
a' \leftarrow a^u g^v & & \\
b' \leftarrow b^{su} A^v & & \\
c' \leftarrow \mathcal{H}(A, B, z', a', b') & & \\
c \leftarrow c'/u \mod q & & \\
 & \xrightarrow{\text{c}} & \\
 & & r \leftarrow cx + w \mod q \\
 & \xleftarrow{r} & \\
r' \leftarrow ru + v \mod q & &
\end{array}$$

Fig. 1: Withdraw protocol from Brand's [21] offline e-cash scheme.

The account holder now has a representation of a token of which the bank does not know the representation. This makes it impossible to link the account holder to the token, providing anonymity. The scheme also offers unlinkability as the tokens are constructed with mostly random values. On the other hand, the token also requires the bank's signature to be valid. Under the assumption that the account holder cannot forge the signatures of the bank, this satisfies the unforgeability property of the e-cash.

When the account holder wants to spend the token(s), it sends two variables and the signature (of the bank) of these variables to the receiver. The receiver can then verify the validity of the token and compute a challenge for the account holder. This challenge includes the identity of the receiver and a unique identifier of the transaction, e.g. the date-time of the transaction. The account holder responds with two variables, computed with the challenge of the receiver, and three of the random variables generated during the withdrawal phase. With these variables and the public values of the bank, the receiver can verify if the token is valid. Additionally, the receiver now has proof that the account holder has paid with that token. Otherwise, it would be impossible to know these values.

Upon the deposit of the token by the receiver, the bank checks the database of deposited tokens. The bank stores the token's identifier, the timestamp of the transaction, and the two variables received by the payee in the database. If the identifier of the token is not yet in the database, the bank adds the deposit to the database and credits the receiver. However, if the token was added earlier, there are two possibilities. Namely, the receiver is trying to deposit the same token twice, or the account holder has spent the token before.

The bank can trivially check for the first case by comparing the two computed values with values already stored in the database. If they are equal, the receiver tries to deposit the same token twice. This would imply that the same challenge would be sent to the account holder upon receiving the token. This would be impossible as the challenge includes a unique identifier and the identity of the receiver.

On the other hand, if the variables differ, the account holder must have spent the token twice. With these variables, the bank can find the identity embedded in the representation of the token. With this identity, the anonymity of the account holder can be revoked. This would allow the bank to take legal action against the account holder with the variables given by the receivers as proof of double spending.

Brands [21] extended this system by adding a non-malleable observer provided by the bank when the account holder opens an account. This observer stores the values needed to construct the token. These are in this case not known by the account holder. When a token is being spent, the observer will find the unknown variables in its memory and remove them after they have been used. This makes it impossible for the user to spend the same token twice, preventing double-spending.

If the tamper resistance of the observer is broken, the account holder can find the hidden variables. With these values, the account holder can double-spend the tokens. However, that double-spending would still be detected like before.

A. Recoverability of tokens

Liu et al. [24] noted in 2005 that most of the earlier (offline) e-cash schemes, including the work of Brands [21], do not support the property of recoverability. They argue that recoverability is needed in the system as there are several risks of losing e-cash. First of all, since e-cash is digital, there exists a risk that the files containing the representation of cash or other relevant files can be corrupted, or the entire computer could crash.

Additionally, losing the medium on which the cash is stored would imply losing all access to the e-cash. If another malicious person finds the card, that person could spend all cash stored on that medium. When using a credit or debit card, one could block the card, preventing it from being used in future transactions.

Adding recoverability to a traceable scheme is trivial, as the bank can find the number of coins credited to and spent by the user. However, this is impossible when the cash in the scheme is untraceable, as the bank does not know how many tokens the user holds or has spent.

The solution that Liu et al. [24] posed was to add a *Recovery Center* (RC) to the scheme of Brands [21]. The purpose of the RC is to store information needed to recover e-cash. After the account holder has withdrawn e-cash from the bank, the account holder sends the tokens to RC. The RC then computes two signatures. The first one is a signature of the token with an additional variable x_i, with i being the counter of tokens sent by the account holder to the RC. The second signature is computed with the combination of a hash of x_i and i. The account holder then adds the first signature to the representation of the token and can store the second signature, needed for the recovery, in a different place.

The receiver must now check if a token is recovered by the RC when it is received. This can be done by doing a lookup in the blacklist maintained by the RC. If the token was recovered before, and therefore added to the blacklist, the receiver has to stop the transaction.

For the recovery, the account holder must reveal their identity to the bank and the RC and send the second signature received from the RC to the bank. The bank then checks if the token(s) are spent earlier and refunds them if they are not. Additionally, the RC has to add the hash found in the signature to the blacklist and notify all users, such that they will not accept coins that map to the hash.

As this scheme is both computationally heavy and the fact that the account holder has to reveal its identity is considered undesirable, Juan [25] has designed $Ro-Cash$. To provide more anonymity in the system, the scheme uses digital pseudonyms.

Following the definition of Chaum [36]: ”A digital pseudonym is a public key used to verify the signatures made by the anonymous holder of the corresponding private key”. A list of pseudonyms is created and maintained by a trusted third party. This is combined with bilinear pairing for relatively short and highly secure keys.

When creating an account with the bank, the account holder receives a tamper-proof smartcard from the bank that contains the pseudonym of the account holder. This smartcard is used to prevent double-spending. Upon withdrawing e-cash the account holder requests a partially blind signature from the bank for the e-cash. Additionally, the encrypted blinding factors are sent to an auditor.

In contrast to the scheme of Liu et al. [24], the account holder can reconstruct the same token with the help of the bank and the auditor. This removes the need to maintain a blacklist of tokens that are recovered and thus invalid. Therefore it is no longer required to forward that information to other parties.

B. Storage reduction

Another issue that Juan (2005) noted in the scheme of Brands [21] was that the bank has to maintain an enormous database to detect double spending [26]. Besides that, he also raised the concern that the bank could issue additional e-tokens if they are untrustworthy. To combat both problems, *AOMPS*, anonymous offline multi-authority payment scheme, was designed. In AOMPS, the issuing of tokens is assigned to a group of n parties using a blind threshold signature scheme.

In a blind threshold signature scheme, the authority of (blindly) signing a message is transferred from one individual to multiple individuals [37]. One can create a blind signature of the group by requesting blind signatures from at least t, the threshold, out of the total of n individuals. Others can verify the validity and authenticity of the signatures by decrypting the message with the public key of the group.

In AOMPS, the e-token issuers collaborate to generate their individual threshold verifiable public keys and shares based on the public parameters of the bank. To get an e-token, an account holder must first set up a pseudonym with the bank and then receives a tamper-proof device. After that, the account holder must use the blind threshold signature protocol to get a blind e-token from at least t honest issuers. Then, the account holder can send another message to the issuers, which then contact the bank to verify whether the account holder has enough money. If so, an issuer-specific signature is sent back to the account holder. When the account holder has all signatures, the e-token can be constructed and stored in the database of the tamper-proof device.

When transferring money, the user sends a token, an identification of the receiver and the amount to pay to the tamper-proof device. The device then checks if the token is in the database and if the value of the token is higher than the amount to pay. If so, the device will send a certificate and the corresponding token back to the account holder and store a new token with the remaining value in the database. The account holder then sends the e-token along with two certificates to the receiver, which then can verify whether the tokens and transaction details are valid.

C. Token expiration

In 2011, Eslami and Talebi [38] proposed a different scheme that solves the problem of the bank having to maintain a large dataset to detect double-spending. They solved the problem by giving the tokens expiration dates. Additionally, they suggest adding a Central Authority (CA) to the scheme to separate the authentication infrastructure from the bank. The CA handles identity-related proofs and maps public keys to entities. Even though there is a CA, the bank is still required to store information that can be used to identify the account holder. However, the identity is constructed using a variable only known to the account holder. This value is used to validate the account holder's identity and for fraud control. Furthermore, the bank now keeps a table for deposited and exchanged coins.

During the withdrawal protocol, the account holder constructs a coin by sharing computed values with the bank. These values are generated by a combination of variables related to the account holder's identity and random values. The bank also adds a date-time value to the coin, giving the coin an expiration date. With this expiration date, the number of coins the bank has to keep in the database is reduced significantly.

During a transaction, the receiver can first check if the coin has not expired and if the coin is valid. Then, the receiver computes a challenge based on the hash of his identity, elements of the coin, and the timestamp of the transaction. This is used to detect double-spending. The payer then uses ElGamel's to compute γ. The receiver finally checks if the γ is valid.

As the coins can expire, the bank also has to offer an option to exchange expired coins. Coins can be exchanged if they are in neither the exchanged coins table nor the deposited coins table. Firstly, the account holder must present the expired token and the secret identifying variable to the bank. After that, the account holder and bank construct a new token like in the withdrawal phase.

To deposit a coin, one must send the coin and the corresponding challenge and γ to the bank. The bank then checks if the coin has been exchanged or deposited before. If not, the coin is accepted by the bank. Otherwise, the bank has to find out who committed fraud.

Due to the challenge the receiver sent during the transaction, it is not possible to either deposit the same token twice or to spend a token that you received. This is because it is computationally infeasible to find a challenge and γ that are valid without the secret values that the initial account holder knows. If the account holder spends the same token twice, the identity can be found due to the two unique challenges and γ's.

Baseri et al. [32] found three flaws in the scheme designed by Eslami and Talebi [38]. The first flaw is that a malicious account holder could forge the identifying value, such that the forged identity would be found when double spending was detected.

Secondly, the expiration date of the coins could be forged, resulting in coins that would remain valid for a longer period. The final flaw was that during the exchange of an expired coin, the bank only checks the account holder's identity and the coin's validity but not the relation between the two. This makes it possible to exchange the coins of others.

The first issue is solved by constructing the account holder's identity differently. In the proposed version, the account holder chooses a random number and computes its identity by raising a public value of the CA to that exponent.

Year	Author(s)	Novelty	DS Detection	DS Prevention	Recoverable?	Transferable?	Dual Anonymous?
1993	[21]	Observers and DS detection	✓	✓	X	X	X
2005	[26]	Multi authority token issuing	X	✓	X	X	X
2005	[24]	Lost token recovery	✓	X	✓	X	X
2010	[25]	Bilinear pairing to reconstruct tokens	X	✓	✓	X	X
2011	[38]	Integrated token expiration	✓	X	✓	X	X
2013	[32]	Irrefutable token expiration	✓	X	✓	X	X
2014	[39]	Metadata addition during deposit	✓	X	✓	X	X
2015	[22]	Malleable signatures	✓	X	X	✓	X
2021	[23]	Commit transactions	✓	X	X	✓	X
2023	[40]	Re-randomize tokens for dual anonymity	✓	X	X	✓	✓

TABLE I: List of all described offline e-cash schemes

The bank then asks for a zero-knowledge proof of the identity of the account holder. After that, the bank will reply with two values, based on another public value of the CA and one on the bank's private key.

The second problem is solved by embedding the date-time part into the encoding in the coin, making it impossible to change the expiration date of the coin. Lastly, to solve the problem where the relation between the account holder and the coin to be exchanged was not checked, they proposed to add that to the validation when a coin is exchanged.

Fan et al. [39] created a scheme with an attachable deposit date besides the expiration date. With this date, it would be possible to determine how much interest a depositor should get. Additionally, in their scheme, the trusted hardware is used by the bank to protect the privacy of the account holders. If a coin is spent twice, the bank will be able to revoke the anonymity of the account holder. The renewal method of expired e-cash is also more efficient than the method proposed by Baseri et al. [32] computation-wise.

D. Transferable tokens

In 2015, Baldimtsi et al. [22] designed a solution that focuses on a different property of e-cash, namely transferability. The communication costs in the network can be reduced, when exchanging e-cash that one received from others without contacting the bank is possible. To achieve this, Baldimsti et al. used improvements of malleable signatures by Chase et al. [41] a year before. With malleable signatures, one could transform a signature on message m to a signature on message m' when there exists an allowed transformation $T(m) = m'$.

First of all, they stated that most schemes use a set-up where a (deposited) token is composed of three parts: SN, which is the token its serial number or identifier, σ, the signature of the bank on SN, and DS, the tag with which double spending can be detected when the token is deposited. The token can then be represented as $(SN||\sigma||DS)$. When double spending occurs, the bank finds two coins with the same SN and different DS.

Upon receiving a coin, the receiver will give the coinholder a nonce to create the double-spending tag. This nonce is later used to determine who had spent the same token twice.

In the scheme of Baldimsti et al., a token will have the form $(SN_1||\sigma)$, where SN_1 is the serial number created by the account holder and σ the malleable signature of the bank. When that token is transferred k times between users the representation will be $(SN_1..SN_k||\sigma_k||DS_1..DS_{k-1})$. In this representation is SN_k the kth serial number of the token, σ_k the malleable signature on SN_k and DS_{k-1} the double-spending tag generated by user $k-1$ when transferring the coin.

Since the identity of the users is embedded in the SN tag, the bank can find the user responsible for double spending when it detects two coins with the same serial number.

Bauer et al. [23] reviewed this scheme in 2021 and found that it is inefficient due to the malleable signatures, as every coin has to store all the transformations it has undergone. They also proposed a new scheme that uses similar tags. However, instead of malleable signatures, users now have to *commit* the transaction. This is done by using a commit-and-prove scheme.

When a user *commits* a transaction, the coin its structure will be updated to contain the commit tag, the encryption of that tag, and proof values proving that the commit tag and its encryption are equal. Due to these tags, the coin its structure is (partly) randomized, resulting in re-randomizable encryption and removing the need for malleable signatures. This makes the scheme more efficient.

E. Dual Anonymity

In 2023, Jianbing et al. [40] proposed a scheme that has dual anonymous transactions. Dual anonymity implies that the identity of the payer and the payee remain secret during the transaction and the deposit of e-cash. In most e-cash schemes the identity of the payee is linked to a transaction when the coin(s) are deposited. However, this means that if the bank colludes with the payer of the transaction, they could reveal and potentially abuse sensitive information regarding the payee.

During the transaction in this scheme, the payee proves that he is a valid user by showing a zero-knowledge proof without revealing his identity. Then the payer generates a unique transaction identifier R and calculates a traceable tag. The payer can now anonymously prove that he possesses a coin and that the tag corresponds to the identifier of the coin. With this information, the payee can generate a transcript which can be used to re-randomize the received coin at the bank. The payee can now choose to deposit the newly randomized to his account or he could use the coin to pay in a different transaction. Due to the randomization, it is impossible to link either the payer or the payee of the transaction prior to the deposit. Additionally, the re-randomization also makes the coins transferable under the requirement that the tokens are re-randomized by the bank.

Table I lists all discussed schemes with their properties. As all schemes satisfy the properties of (strong) anonymity and unlinkability due to the blind signatures introduced by Chaum, these are omitted from the table. As seen from the table, no scheme satisfies all the (desired) properties of offline e-cash. Additionally, almost all cash schemes seem fully theoretical and have no implemented prototype to test for feasibility and scalability. Only [40] included an efficiency analysis based on an implemented prototype.

The entries in Table I, which have the feature that the scheme prevents double-spending, all require specialised and secure hardware or software such that malicious users cannot abuse the system. The security of that system is thus largely dependent on the security of the tamper-proof device, which can be questionable.

V. Conclusion

For now, no proposed e-cash scheme satisfies all properties that regular cash also has. Anonymity and unlinkability can be guaranteed with blind signatures and some randomness. However combining more complex properties, like recoverability and transferability, is harder while combating double-spending. Some schemes rely on trusted hardware to prevent this. However, that raises the need for more research into such components to verify their security. On the other hand, detecting double spending once it has occurred, would require the involvement of a legal department to prosecute malicious users.

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] Yeonouk Chu et al. “Review of offline payment function of CBDC considering security requirements”. In: *Applied sciences* 12.9 (2022), p. 4488.

[3] Daniel D Garcia-Swartz, Robert W Hahn, and Anne Layne-Farrar. “The move toward a cashless society: a closer look at payment instrument economics”. In: *Review of network economics* 5.2 (2006).

[4] Markus K Brunnermeier, Harold James, and Jean-Pierre Landau. *The digitalization of money*. Tech. rep. National Bureau of Economic Research, 2019.

[5] Younghoon Chang et al. “The role of privacy policy on consumers’ perceived privacy”. In: *Government Information Quarterly* 35.3 (2018), pp. 445–459.

[6] Tobias Bamert et al. “Bluewallet: The secure bitcoin wallet”. In: *Security and Trust Management: 10th International Workshop, STM 2014, Wroclaw, Poland, September 10-11, 2014. Proceedings 10*. Springer. 2014, pp. 65–80.

[7] David Kuo Chuen Lee, Li Yan, and Yu Wang. “A global perspective on central bank digital currency”. In: *China Economic Journal* 14.1 (2021), pp. 52–66.

[8] Peterson K Ozili. “Central bank digital currency research around the World: a review of literature”. In: *Journal of Money Laundering Control* 26.2 (2023), pp. 215–226.

[9] Gabriel Soderberg et al. “Behind the scenes of central bank digital currency: Emerging trends, insights, and policy lessons”. In: (2022).

[10] Jonathan Chiu and Seyed Mohammadreza Davoodalhosseini. “Central bank digital currency and banking: Macroeconomic benefits of a cash-like design”. In: *Management Science* (2023).

[11] Jonas Gross et al. “Designing a central bank digital currency with support for cash-like privacy”. In: *Available at SSRN 3891121* (2021).

[12] European Central Bank. *Progress on the investigation phase of a digital euro – third report*. Tech. rep. European Central Bank, 2023.

[13] John Kiff. “Taking Digital Currencies Offline”. In: *International Monetary Fund* (2022).

[14] Lynn Batten and Xun Yi. “Off-line digital cash schemes providing untraceability, anonymity and change”. In: *Electronic Commerce Research* 19 (2019), pp. 81–110.

[15] Jannik Dreier, Ali Kassem, and Pascal Lafourcade. “Formal analysis of e-cash protocols”. In: *2015 12th International Joint Conference on e-Business and Telecommunications (ICETE)*. Vol. 04. 2015, pp. 65–75.

[16] Pawel Pszona and Grzegorz Stachowiak. “Unlinkable Divisible Digital Cash without Trusted Third Party”. In: *Cryptology ePrint Archive* (2007).

[17] Sébastien Canard and Aline Gouget. “Anonymity in transferable e-cash”. In: *Applied Cryptography and Network Security: 6th International Conference, ACNS 2008, New York, NY, USA, June 3-6, 2008. Proceedings 6*. Springer. 2008, pp. 207–223.

[18] 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.

[19] R Sai Anand and CE Veni Madhavan. “An online, transferable e-cash payment system”. In: *Progress in Cryptology—INDOCRYPT 2000: First International Conference in Cryptology in India Calcutta, India, December 10–13, 2000 Proceedings 1*. Springer. 2000, pp. 93–103.

[20] SK Hafizul Islam et al. “Provably secure pairing-free identity-based partially blind signature scheme and its application in online e-cash system”. In: *Arabian Journal for Science and Engineering* 41 (2016), pp. 3163–3176.

[21] Stefan Brands. “Untraceable off-line cash in wallet with observers”. In: *Advances in Cryptology—CRYPTO’93: 13th Annual International Cryp-*

tology Conference Santa Barbara, California, USA August 22–26, 1993 Proceedings 13. Springer. 1994, pp. 302–318.

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

[23] 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.

[24] 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.

[25] 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.

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

[27] Henrique de Carvalho Videira. “The offline digital currency puzzle solved by a local blockchain”. In: *arXiv preprint arXiv:2305.02290* (2023).

[28] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. “Trusted execution environment: what it is, and what it is not”. In: *2015 IEEE Trustcom/BigDataSE/Ispa*. Vol. 1. IEEE. 2015, pp. 57–64.

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

[30] 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.

[31] Berry Schoenmakers. “Security aspects of the EcashTM payment system”. In: *Lecture notes in computer science* (1998), pp. 338–352.

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

[33] Karl Wüst et al. “Platypus: a central bank digital currency with unlinkable transactions and privacy-preserving regulation”. In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 2022, pp. 2947–2960.

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

[35] Elsayed Mohammed, A-E Emarah, and K El-Shennaway. “A blind signature scheme based on El-Gamal signature”. In: *Proceedings of the Seventeenth National Radio Science Conference. 17th NRSC’2000 (IEEE Cat. No. 00EX396)*. IEEE. 2000, pp. C25–1.

[36] David L Chaum. “Untraceable electronic mail, return addresses, and digital pseudonyms”. In: *Communications of the ACM* 24.2 (1981), pp. 84–90.

[37] Wen-Shenq Juang and Chin-Laung Lei. “Blind threshold signatures based on discrete logarithm”. In: *Annual Asian Computing Science Conference*. Springer. 1996, pp. 172–181.

[38] 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.

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

[40] 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`.

[41] 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.