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

However those type of payments are heavily dependent on being able to contact one or more trusted third parties. When such a party can not be contacted, for example during an outage, which occur more often and with an increasing duration [4], the method is obsolete and void.

Another issue with these digital payment methods, is that they are not anonymous, unlike cash, and thus require trust in a third party, most often a bank, to handle the personal data such as balance, transaction details and name and address details confidentially and not use them for commercial gain [5], and secure it adequately to prevent data breaches.

Other increasingly popular digital payment methods, such as cryptocurrencies [6], like Bitcoin and Ethereum, avoid this problem by providing (pseudo)anonymity in a decentralized network. However they depent on the same condition that the ledger should be reachable during a transaction to ensure that the transaction is valid and completed, and 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 CBDC's 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 also remains functional offline, for example when there is a power outage or when the bank is unreachable.

However there does not (yet) exist a digital payment scheme that satisfies all desired properties. 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 that the scheme should have 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

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

Unforgeability. It should not be possible for a user to create valid, but fake, e-cash 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 privacy of the user of the e-cash, it should not be possible to link a user to a transaction. This is known as *Weak Anonymity* (WA) [15]. A stronger notion of anonymity, *Strong Anonymity*, poses that, in addition to WA, it should also be impossible to decide if two transactions were 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, as it 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 e-cash they received, without the need to deposit and

withdraw the e-cash first.

III. Double Spending

Contrary to online e-cash schemes, offline schemes do not have access to a trusted third party, or at least the entity that is responsible for issuing and retrieving e-cash, and can thus not verify whether a token is already spent or not. Combined with the ease of copying a token, as it is just digital data, and the property of anonymity of (e-)cash, measures should be taken to prevent or discourage malicious users from doubly 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, either by ensuring that double spending can be detected at a later stage, for example when depositing the coins, or by making use of 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 if the token is spent only once. However, if the token is spent at least twice the identity of the spender can be resolved from the token. As the identity can be revealed, the user can be held accountable and makes it possible to punish or prosecute the malicious user, discouraging potentially malicious users 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 in 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 for example store a secret value needed to sign a token with that is deleted upon signing such that the token can only be signed, and thus spent, once or store the tokens themselves 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, which could be broken [29, 30].

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 it would allow for anonymous untraceable payments. 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 public key of the signer [35]. In an e-cash scheme a user can thus construct a valid token in collaboration with an issuer, and let the issuer create a blind signature of the e-cash. This way the issuer, for example a bank, does not know the exact content of the token, providing anonymity and untraceability for the user, and can the token 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. Upon withdrawing coins from the bank, the bank will send two variables based on the generator group and a random value, of which one of the variables is constructed by using the identity of the user. The account holder will then generate five random numbers, of which one is used to construct another value with the identity of the account holder. With these numbers the account holder computes a blinded challenge and sends it to the bank.

With this set-up the account holder now has a representation of a token of which the bank does not know the representation, making it impossible to link the account holder to the token, thus providing anonymity and unlinkability as the tokens are constructed with random values. On the other hand, the token also requires a signature of the bank in order 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, who can then verify the validity of the token and compute a challenge for the account holder, which contains 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 which are computed with the challenge of the receiver and three of the random variables generated by the account holder during the withdrawal phase. With these variables and the public values of the bank the receiver can verify the validity of the token, and has proof that account holder has payed with that token, otherwise it is impossible to know these values.

Upon the deposit of the token by the receiver the bank checks the database of deposited tokens. In this database the bank stores both the identifier of the token, which is constructed in the withdrawal phase by the account holder and blindly signed by the bank, the date time of the transaction and the two variables received by the payee of the transaction. If there is no entry of the identifier of the token the bank adds the deposit to the database and credits the receiver. However if there is already an entry of that token in the database, there are two possibilities, namely either 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 compar-

ing the two computed values with the values that are already in the database. If they are equal, the receiver tries to deposit the same token twice, as the same challenge would be sent to the account holder upon receiving the token, which would be impossible as the challenge should include the unique identifier, such as the date time, and the identity of the receiver.

Contrary, if the variables differ, the account holder must have spent the token twice. With the combination of the two variables linked to the token, the bank can find the identifier of the account holder which is embedded in the representation of the token and thus revoke its anonymity. 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 is used to store values needed to construct the token, which are in this case not known by the account holder. When the account holder spends a token the observer will find the unknown variables in its memory and removes them after they have been used. This makes it impossible for the user to spend the same token twice which prevents double spending.

Breaking the tamper-resistance of the observer, allowing the account holder to find the hidden variables, gives the account holder the opportunity to double spend the tokens. However, this double spending can still be identified once the tokens are deposited in the same way as the scheme without the observer.

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 recoverabilty is needed in the system as there are several risks of losing e-cash. First of all, since the 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 it stored would imply losing all access to the cash. If another malicious person would find that card he/she could spent all cash stored on that medium. When using a credit or debit card, this situation would not occur, as one could simply block the card such that it cannot be used in future transactions.

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

The solution that Liu et al. [24] posed was to add a *Recovery Center* (RC) to the scheme of Brands [21], with the purpose 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, which 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, and the second a signature over 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, which is needed for the recovery, in a different place.

When the account holder pays with the token, the receiver must now also check if that token is recovered by the RC or not, by doing a look up in the blacklist maintained by the RC. If the token was recovered earlier and therefore on the blacklist, the receiver has to stop the transaction.

For the recovery, the account holder has to reveal its identity to both the bank and the RC. The account holder must sent the second signature received from the RC to the bank. The bank then checks if the token(s) are spent or not 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 makes use of 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 by the bank, the account holder receives a non-tamperable 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 send 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, removing the need 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 in this case had to maintain a enormous database to detect double spending [26]. Besides that, he also raised to concern that the bank could also issue additional e-tokens if they are untrustworthy. To combat both problems, *AOMPS*, anonymous off-line 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]. By requesting blind signatures from at least t, the threshold, out of the total n, the amount of individuals, one can create a blind signature of the group. Others can verify the validity and authenticity of the 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, without a trusted third party, 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. The account holder must then send another message to the issuers, which then contact the bank to verify whether the account holder has enough money. If that is the case, a issuer specific signature is sent back to the account holder. When the account holder has all signatures, the e-token can be constructed and is stored in the database of the tamper proof device.

When transferring money, the user sends a token in combination with the 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 stores a new token with the remaining value back in the database, making the tokens divisible. The account holder then sends the e-token along with two certificates to the receiver, which then has the possibility to verify whether to 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 by giving the tokens expiration dates. Additionally, they also suggest to add a Central Authority (CA) to the scheme to separate the authentication infrastructure from the bank. The CA is meant to handle identity related proofs and maps public keys to entities. Even though there is a CA in the scheme, the bank is still required to store information to identify the account holder with, when the identity values are computed. 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 needs to keep a table for deposited coins and exchanged coins.

During the withdrawal protocol the account holder constructs a coin by sharing computed values to the bank, generated by a combination of variables related to the identity of the account holder and random values. Additionally 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 significantly reduced.

During a transaction the receiver can first check if the coin has not expired and if the coin is valid. Then the receiver computes challenge based on the hash of the identifier of the receiver, elements of the coin and the date time of the transaction. This is used to prevent double spending (by the receiver). The payer then uses ElGamel's to compute γ. The receiver finally checks if the γ is valid.

As the coins now have an expiration date the bank also have to offer an option to exchange expired coins, which are in neither the exchanged coins table and the deposited coins table. Firstly the account holder must present the expired token in combination with the secret identifying variable. After that the account holder and bank construct a new token like in the withdrawal phase.

To deposit a coin, one has to send the coin and the corresponding challenge and γ to the bank. The bank then checks if the coin is already existing in the exchanged coin table or the deposited coin table. If not the coin is accepted by the bank. Otherwise the bank has to find who commit fraud.

Due to the challenge the receiver sent during the transaction, it is not possible to either deposit the same token twice or to spent a token that you received, as 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 different challenges and γ's.

Baseri et al. [32] found three flaws in the scheme designed by Eslami and Talebi [38]. The flaws were that a malicious account holder could forge the identifying value such that the forged identity would be found when double spending was detected, the expiration date of the coins could be forged, such that coins would remain valid for a longer period, and lastly during the exchange the bank only checks the identity of the account holder and the validity of the coin, but not the relation between the two. Therefore it would be possible exchange coins of others. Finally Baseri et al. also noted that the inclusion of the exchange and deposit tables in fact does not decrease the size of the database, as information about outdated coins should still be stored.

The first issue is solved by constructing the identity of the account holder 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. 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, both based on another public value of the CA and one with a private key of the bank.

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 exchanging a coin.

Fan et al. [39] created a scheme that has an attachable deposit date besides the expiration date. With this date it would be possible to determine how much interest a depos-

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

TABLE I

LIST OF ALL DESCRIBED SCHEMES

itor 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] in terms of computation.

D. Transferable tokens

In 2015 Baldimtsi et al. [22] designed a solution that focuses on a different property of e-cash, namely transferability. When it is possible to exchange e-cash that one received from others without having to contact the bank, the communication costs in the network can be reduced. To achieve this property, Baldimsti et al. used improvements of malleable signatures by Chase et al. [40] 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 a serial number, or identifier, of the token created by the account holder, 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 holder of the coin a nonce to create the double spending tag with. This nonce is later used to determined 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 over that serial number. 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 which SN_k represents 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, it is possible for the bank to find the user responsible for double spending when it detects two coins with the same serial number.

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

When a user *commits* a transaction, the structure of the coin will be updated to contain the commit tag, an encryption of that tag, and a proof values proving that the commit tag and its encryption are equal. Due the these tags, the structure of the coin is (partly) randomised, resulting in rerandomisable encryption and removes the need for malleable signatures, making the scheme more efficient.

Table I lists all discussed schemes with their properties. As all schemes satisfy the properties of (strong) anonymity and unlinkabilty due to the blind signatures introduced by Chaum, these are omitted from the table. As can be seen from the table, there is no scheme that satisfies all the desired properties of offline e-cash. Additionally all cash schemes seem to be fully theoretical and have no prototype implemented to test for feasibilty and scaleability.
The entries in Table I that have the feature that the scheme prevents double spending, all need 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, there is no proposed e-cash scheme that satisfies all properties that regular cash also has. Even though anonymity and unlinkabilty can be solved with blind signatures and some randomness, other more complex properties like recoverability, transferability and divisibility are harder to combine, whilst also combating double spending. Some schemes rely on trusted hardware to prevent this, however that raises the need for more research to such components to verify their security. On the other hand, by detecting double spending once it has occurred would require move involvement of 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 Cryptology 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 ElGamal 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] 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.