SUPPORT POOL
OF EXPERTS PROGRAMME

Expert Opinion

The Digital Euro and Its Token-Based Offline Modality

by Prof. Dr.-Ing. **Tibor JAGER**

As part of the SPE programme, the EDPB may commission contractors to provide reports and tools on specific topics.

The views expressed in the deliverables are those of their authors and they do not necessarily reflect the official position of the EDPB. The EDPB does not guarantee the accuracy of the information included in the deliverables. Neither the EDPB nor any person acting on the EDPB's behalf may be held responsible for any use that may be made of the information contained in the deliverables.

Some excerpts may be redacted or removed from the deliverables as their publication would undermine the protection of legitimate interests, including, inter alia, the privacy and integrity of an individual regarding the protection of personal data in accordance with Regulation (EU) 2018/1725 and/or the commercial interests of a natural or legal person.

Summary

This document explores the technical feasibility, inherent limitations, possible approaches, and security considerations surrounding the development of a cash-like, anonymous, and double-spending-resistant offline modality for the Digital Euro.

Key Insights:

- The vision of an offline digital currency that fully replicates the anonymity, security, and usability of physical cash is conceptually appealing, but the absence of physical constraints, the need for robust double-spending resistance, and the tension between anonymity and traceability pose a significant technical challenge.
- Realising an offline, secure, and privacy-preserving digital currency in practise requires navigating a complex landscape of technical trade-offs. There exists a large variety of cryptographic tools that can be used to achieve unforgeability and anonymity in a very strong way. These must be combined with suitable techniques for double-spending prevention, e.g., via secure hardware elements or by carefully weakening the offline requirement.
- The design must be guided by clear requirements, realistic threat models, and close collaboration across disciplines, including cryptography, systems security, hardware engineering, legal frameworks, and policy. Public transparency in both the design and implementation of all system components is essential, because scrutiny by independent experts helps uncover vulnerabilities early and is critical to fostering public confidence, broad adoption, and long-term legitimacy of the system.

In conclusion, **an anonymous modality of the Digital Euro appears feasible, provided that the system is designed based on a suitable combination of techniques and trade-offs**. Currently available cryptographic techniques offer a credible path forward for the design of a privacy-preserving and secure offline modality of the Digital Euro.

Outline: This document is structured around five questions that are stated in Section 1. Section 2 contains definitions of essential terminology and notions such as "cash-like" currency. Section 3 discusses a well-known fundamental limitation that applies to any distributed system and its applicability to offline digital currencies. Section 4 considers the possibility and the challenges of using secure hardware to overcome this limitation. Section 5 discusses inherent challenges and techniques to achieve "cash-likeness", including possible hybrid solutions. Section 6 considers the question of double-spending prevention. Section 7 focuses on technical approaches to design a cash-like offline digital currency based on standard cryptographic techniques. While it is beyond the scope of this document to provide a comprehensive overview of all available cryptographic methods, we aim to illustrate what is technically feasible by focusing on *blind signatures* as one particularly influential and well-understood tool and discuss their functionality, benefits, limitations, and approaches to address these limitations.

Document submitted in September 2025

Contents

1 Considered Questions

This expert opinion is commissioned by the European Data Protection Board (EDPB) to address the following questions:

1. In what way can the CAP theorem, if it is valid, be applied to the setup/design of the token-based offline modality of the Digital Euro and what do each of its three properties (consistency, partition tolerance and availability) mean in the context of electronic payment services?

2. Considering the history and past (successful) attacks on secure (hardware) elements, is multi-spending thinkable and – if yes – what technical measures could be added to build further lines of defence against attempts to compromise the offline Digital Euro infrastructure's end-user devices (e.g. smartphones)?

3. Are there any feasible alternative technical solutions to render the offline modality of the Digital Euro "cash-like", meaning both resilient in terms of double-spending and at the same time anonymous to use ...

   (a) while respecting the CAP theorem?

   (b) while accepting one out of three properties not to be fulfilled, e.g. an offline modality at the expense of "availability", and the consequences for practical implementation and usability?

   (c) with anonymity not for all parties but only/at least for the consumer/payee (natural person) as opposed to the merchant ("sender anonymity"), potentially resulting in a hybrid solution?

4. How can the token-based offline modality prevent double-spending in situations with several transactions in sequence and without any form of reconciliation in between? Conversely, how can a reconciliation process be implemented while preserving the cash-likeness of the token-based offline modality?

5. Considering existing research on the design choices of the Digital Euro, which technical solutions might be helpful for implementing the token-based offline modality while preserving anonymity?

2 Definition of Terminology

In order to be able to answer the questions about "offline", "token-based", and "cash-like" digital currencies, we have to make these terms more precise. It will be useful to start from a simplified perspective on physical cash.

2.1 Physical Cash

Figure 1 gives a simplified and abstracted illustration of physical cash. It depicts the separation between the financial institutions involved in issuing and redeeming money and the transfer of value between individual users. The picture considers three different types of users:

- The *first payer* is the user that withdraws cash from its bank and spends it by transferring it to a different user, the *last payee* or an *intermediary* user in a sequence of payers and payees.

- There may exist an unbounded number of *intermediaries*, which are users that receive value from one user and transfer it to another. These intermediaries act both as payees and payers.

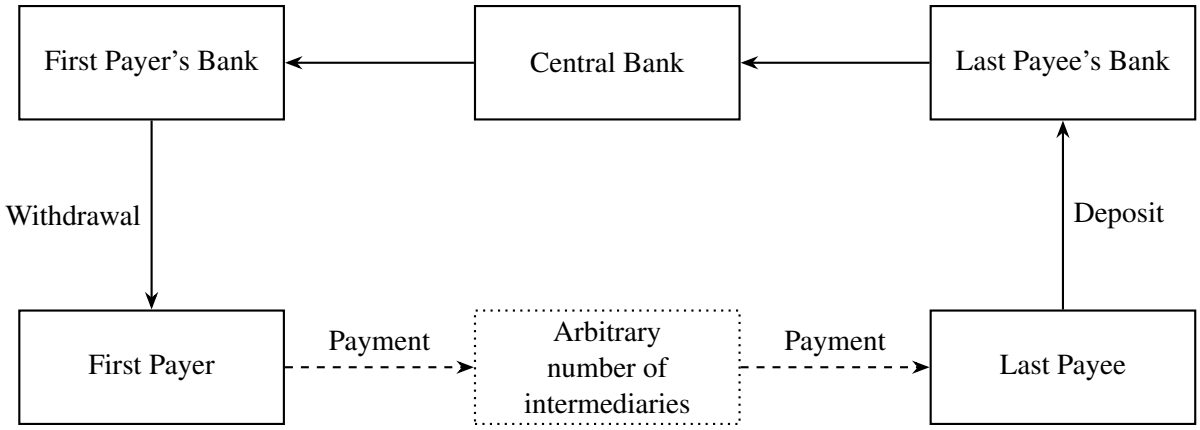


Figure 1: Illustration of physical cash. Solid lines depict transactions that may be considered "online", i.e., with an active communication channel to a central entity, such as a (central) bank. The dashed lines depict "offline" transactions, which can be performed without communication to a central entity.

- The *last payee* is the final recipient of the cash, which then deposits it at its bank.

The dashed arrows in Figure 1 represent the cash payments, flowing from the first payer to the last payee through an arbitrary number of intermediaries. This highlights a core feature of physical cash: once withdrawn, it can be freely transferred between parties without the involvement or oversight of banks or the central authority. The dotted *intermediary* node abstracts that cash may pass through hands of an arbitrary number of intermediaries (e.g., businesses, individual users) before reaching the final recipient. None of the intermediaries needs to be identified or recorded for functionality purposes.

Physical cash can be used without the need to identify or record any intermediary. Typically the first payer is identified when withdrawing, and the last payee may be identified when depositing money.

2.2 "Online" and "Offline" Transactions

For our definition of "online" and "offline" transactions, we will follow the definition given in the European Commission's proposal for a regulation on the establishment of the Digital Euro [Eur23]:

- Art. 2 No. 14 of [Eur23] defines an *"online digital euro payment transaction"* as *"a digital euro payment transaction where the settlement takes place in the digital euro settlement infrastructure"*.
- Art. 2 No. 15 of [Eur23] defines an *"offline digital euro payment transaction"* as *"a digital euro payment transaction, made in physical proximity, where authorisation and settlement take place in the local storage devices of both payer and payee"*.

Following this approach and adapting it to physical cash, we define an "online" transaction as one that requires an active communication channel to a central entity (i.e., a user's bank or the central bank), and an "offline" transaction as one that does not. Then physical cash has the following properties:

- The *first* payer must be *online* for withdrawal of cash. However, no active communication is required when the cash is transferred to an intermediate or final payee. Thus, the payer may be *offline* when transferring the cash to another user.

- The intermediaries may be *offline* when receiving or transferring cash to another user.
- The payee may be *offline* when receiving a payment, but must be *online* for the deposit.

2.3 "Tokens" are Digital Coins

In the sequel, we will understand a *"token"* as the digital equivalent of a physical coin or banknote: a self-contained unit of value that can be transferred directly between users without requiring immediate online connectivity. Each token represents a specific denomination and embodies the monetary value it carries, much like tangible cash. For simplicity, in the sequel we will mainly consider tokens of fixed denomination (e.g., 1 EUR), as this facilitates clarity in illustrating the core concepts. Extensions will be discussed in Section 7.3.3.

2.4 "Cash-like" Digital Currency

In this subsection, we will identify several different properties that are satisfied by physical cash, and that may be expected from a "cash-like" digital currency. This will provide a basis for discussing possibilities and limitations in achieving these properties in the digital world. We will also explain that *physical proximity* is a property of physical cash that is very difficult to reliably enforce in a digital currency.

2.4.1 Properties Expected from a Cash-Like Currency

In a cash-like currency, we require at least the following four properties, which are satisfied by physical cash:

1. **Unforgeability** refers to the property that ensures that it is infeasible to create counterfeit money in a way that it is accepted as genuine by a payee or a bank with high probability (or at least irrational because the costs of counterfeiting exceed the value of the counterfeit money).

   Physical cash is unforgeable in the following sense. Central banks implement various measures to make forgery very difficult, or at least prohibitively expensive. This includes watermarks, holograms, special printing techniques, specialised paper, and UV and magnetic features, for example. These features are designed to be very difficult to reproduce, in particular several of them in combination. If the costs of producing counterfeit money exceed its value, then it is prohibitively expensive (and thus irrational) to forge money. Thus, one may consider this as unlikely, in particular at large scale.

   Unlike physical cash, digital data can be copied or cloned effortlessly. Therefore in this context unforgeability refers only to the inability of creating *new* tokens. Cloning a digital token is not considered a forgery (but still *double-spending* of tokens must be prevented, cf. Item 3 below). In the digital context, unforgeability can be enforced using cryptographic techniques such as digital signature schemes and related, privacy-preserving primitives such as blind signatures [Cha82, Cha86, CFN90] (see Section 7).

2. **Anonymity** is, intuitively, the property that ensures that a user's identity is hidden within a set of possible users, the so-called *anonymity set*. Depending on the concrete application context, we have different interpretations of what the general term "anonymity" precisely refers to, and one must use very different techniques to achieve it. For example, standard techniques in the context of databases include *statistical aggregation*, *differential privacy*, *data minimisation*, or adding *noise*, but digital currencies require different techniques because they aim to achieve a different form of anonymity.

In the context of digital currencies, anonymity refers to the fact that users can make payments without revealing their identity or allowing their transactions to be traced back to them. Therefore we define anonymity as the combination of *pseudonyms* that cannot be traced to the individual with *unlinkability* of actions:

- **Pseudonymity** refers to the property that users operate under identifiers (pseudonyms) that are not directly linked to their real identities.

  Physical cash is pseudonymous, in the sense that the vast majority of all cash payments is made without the need to show an ID card or revealing any other information about the users participating in the transaction.

  Very importantly, note that pseudonymity alone is not sufficient to achieve anonymity. This is because even if the pseudonym does not reveal the user's real identity, it may still be possible to identify different transactions that belong to the same user. For example, if a user performs multiple actions under the same pseudonym, those actions become linkable. Over time, patterns of behaviour or interaction may reveal enough contextual information to identify the individual behind the pseudonym with high success probability. Anonymity requires that users are indistinguishable within an anonymity set and that their actions cannot be reliably linked to each other or to a persistent identifier, which is not achieved only by the use of pseudonyms. Therefore, pseudonymity can offer a layer of identity protection, but it alone does not prevent long-term tracking or profiling.
- **Unlinkability** refers to the property that individual transactions cannot be associated with one another or with a specific entity. In the context of a digital currency, unlinkability of transactions prevents observers or attackers from connecting multiple activities belonging to the same user, even if they occur across different transactions and over time.

  Physical cash is unlinkable in the following sense. Even though many physical cash transactions are in principle linkable through the unique serial numbers of banknotes, cash payments can be viewed as practically unlinkable, **assuming** that neither individuals, nor businesses or banks record serial numbers in ordinary transactions. Furthermore, when a banknote passes through many intermediaries, each one introduces uncertainty about the original source and final destination of the note, which may make linking different transaction across (multiple) intermediaries significantly more difficult. Thus, if a payment is observed at some point, it is practically infeasible to link it to other transactions of the same user. The more intermediaries involved, the larger the number of plausible origins and endpoints, effectively expanding the anonymity set. There are, however, exceptions. For example, in exceptional cases law enforcement might collaborate with banks and use serial numbers to track specific notes involved in crimes.

Pseudonyms hide the users' identities and unlinkability ensures that different activities performed by the same user cannot be connected, i.e., it is infeasible to detect whether these activities belong to the same user or to different users. Together, these properties make it impossible to trace actions back to an individual or to build behavioural profiles, and thereby effectively protect the user's anonymity. Thus, **anonymity is achieved when users are pseudonymous and their actions are unlinkable.**

Together, its pseudonymity and unlinkability properties make physical cash a paradigmatic example of an anonymous payment system. In Section 5.1, we will also explain why it seems necessary to expect even stronger anonymity properties from a digital equivalent of physical cash.

There are **powerful cryptographic techniques** such as blind signatures, which can be used to achieve unforgeability and anonymity simultaneously. These will be discussed in Section 7.

3. **Double-spending resistance** is the property that it is practically infeasible to spend the same coin or banknote more than once.

   In the case of physical cash, resistance to double-spending arises naturally as a direct consequence of its existence as a unique physical object. Once this object is handed over in a transaction, the original holder no longer possesses it, making it impossible to spend it again. **The physical transfer of a coin or banknote inherently prevents the same unit of currency from being spent more than once.** Note that the existence as a unique physical object does *not* hold for digital data.

   In a digital currency system, resistance to double-spending can be achieved in two ways:

   (a) By technically *preventing* the same token from being spent more than once (e.g., a secure hardware component might be used to ensure that a token cannot be double-spent by a malicious user, or an online connection might be used to check whether a given token has been spent before or not).

   (b) By employing auditing mechanisms for *detecting* instances of double-spending after they occur, enabling authorities to identify and penalise attackers through legal measures.

   If detection is used, then also *non-frameability* [Bra94] becomes relevant, which refers to the guarantee that an honest user cannot be falsely accused of double-spending, not even by the bank. When mechanisms are in place to detect double-spending (such as tracing or identity-revealing protocols), non-frameability ensures that only the actual double spender can be identified, and no third party, including malicious actors or even system operators, can manipulate the system to falsely implicate an innocent user.

4. **Transferability** refers to the possibility of exchanging tokens across several users without the need to involve a bank, i.e., in a way such that there is at least one intermediary in the process depicted in Figure 1.

   A non-transferable currency may still allow to directly transfer value between two users (concretely, directly from the first payer to the last payee in Figure 1, without any intermediaries) in an offline manner. However, once a coin is received from a user, it cannot be forwarded to a different user, but only be deposited at a bank.

   Physical cash is inherently transferable because it can be handed directly from one user to another, and then on to another user, and so on, without requiring involvement of a bank.

   In the context of a digital currency, there exist techniques to achieve transferability, either by relying on secure hardware (cf. Section 4) or by using cryptographic techniques [CHL05, FPV09, BCFK15, BFQ21] (cf. Section 7.3.2). Note also that **transferability may not be an essential requirement for a digital currency, as it is possible to design workarounds**. For example, a device could show two balances: (1) payments received from another user, which cannot be spent to a different user (and thus are "frozen" until final settlement with a bank), and (2) digital token that were received from a bank and can be spent to a different user. Not requiring transferability may simplify the system design or reduce the security requirements on hardware components.

Physical cash provides **unforgeability**, **double-spending resistance**, **anonymity**, and **transferability**. In the sequel, we will discuss how and to which extent an (offline) digital currency can achieve these properties, in order to be "cash-like".

2.4.2 The Difficulty of Enforcing Physical Proximity in a Digital Currency

Another notable feature of physical cash is that it requires some form of physical proximity in order to make a transfer. This makes long-distance transfers rather difficult, especially for large amounts and if the transfer is supposed to be covert (e.g., hidden from authorities). At the same time, proximity is easy to verify, because every party involved in a cash transaction can immediately confirm that the other party is physically present, simply because they are exchanging the notes face-to-face. This property, which is trivial for physical cash, cannot easily be reproduced in a digital currency.

Digital systems cannot "see" distance. A byte arriving over a data connection looks the same whether it came from two centimetres away or from a different continent. Therefore, enforcing physical proximity in a digital currency (regardless whether online or offline) is intrinsically hard, because digital communication channels are location-agnostic. This holds also when the currency is implemented by devices that use short-range communication protocols such as NFC or Bluetooth: nothing in the bytes themselves proves the sender and receiver are actually co-located.

This gap is exploited by *relay attacks*, where an attacker places a proxy next to one device (the payer's) and another next to the other device (the payee's). The proxies forward messages over a fast connection so that each honest device "believes" that the other device is in direct proximity. Such *relay attacks* are well-known. For example, car thieves can use relays to forward the car fob's signal from inside a house to the car on the driveway [FDC11]. In the context of electronic identities, relay attacks can be used to make a remote passport appear present at a reader [KW05].

Unfortunately, the available countermeasures are very limited. Distance-bounding protocols (e.g. [BC94, ABB+18]) try to estimate upper-bound distance by timing rapid challenge–response exchanges. The core idea is that signals cannot travel faster than the speed of light. So if one device issues a challenge and measures how quickly the other device responds, it can upper-bound the distance, because a genuine nearby device should be able to answer very quickly, while a relayed message would take longer. In practise, however, this is extremely difficult to enforce reliably. Commodity devices add unpredictable processing delays, so the verifier cannot isolate just the physical propagation time. Attackers can also use specialised low-latency hardware to shave down forwarding delays, bringing a distant device's response within the acceptable threshold.

Another approach is to consider *vincinity-based pairing* [ZPZ+16], where properties of the physical radio channel between two devices are used as additional inputs to a key derivation and authentication protocol. While this is an interesting approach to enforce proximity, it currently seems to be at a rather experimental technology readiness level.

Reliably preventing relay attacks in everyday devices remains a major challenge. If proximity is an inherent requirement for an offline Digital Euro, then either effective techniques must be developed and deployed to guarantee it, or the system must be designed to acknowledge this limitation and take the risks of relay attacks into account.

In this document, we will not consider physical proximity as a property of cash that can be reliably

enforced in a digital currency.

2.5 Secure Hardware

In the sequel, the term "secure hardware" refers to either a dedicated hardware, such as a smart card, or a secure environment within a general-purpose computer such as a smartphone, which is assumed to be resilient against certain forms of unauthorised manipulation.

It is important to emphasise that we will not use the term "secure hardware" to express that a system *is* actually secure, i.e., that it *achieves* certain security properties. This is because the notion of a hardware "being secure" is not an absolute and persistent state, but a moving target, shaped by the evolving landscape of threats and techniques. Whether the system achieves the security properties required from it may change over time as new attacks and vulnerabilities are discovered.

Instead, we will use the term "secure hardware" to indicate that a certain form of security is *required* from the device by the system using the device. This requirement remains stable over time. For example, a system might require confidentiality of user data or integrity of transaction records, and these requirements do not change (unless the system is changed), even if the methods for achieving them must adapt in response to new attack techniques. For a discussion of such assumed hardware security properties, we refer to Section 4. See also Table 2 for an illustration how different approaches require different hardware security properties.

It is important to note that the expected hardware security properties become progressively stronger (i.e., less realistic and more difficult to guarantee) as the number of different security requirements expected from the hardware increases. Hardware that achieves certain security properties in the context of one application may be insufficient to provide security in a different application context (cf. Section 4.1). Extending existing hardware with additional functionality specifically needed to support a digital currency may be technically challenging or even infeasible, particularly if backwards compatibility is required. In cases where adaptation of existing platforms is not possible, the design and deployment of new hardware may be necessary, which is often both costly and time-consuming.

There are security certifications for hardware devices, such as Common Criteria certification. Such a certification can be viewed as an element to ensure that a given hardware device indeed satisfies the expected security properties. However, importantly, such a certification merely checks certain criteria, but does not guarantee that a device actually is and will remain secure, as this would require to guarantee the absence of *any* current and future attack.[1]

> The term "secure" in "secure hardware" refers not to the device being actually secure, but to the fact that one **requires** and **assumes** a certain security property from the device. The expected hardware security properties become less realistic and more difficult to guarantee as the number of different security requirements increases.

[1] For example, this is also reflected in Art. 25 (3) and 32 (3) GDPR. Here, it is stated that certification can be an element to achieve compliance, however, it is generally not considered sufficient on its own.

3 The CAP Theorem and its Application to Offline Digital Currencies

Considered Question

1. In what way can the CAP theorem, if it is valid, be applied to the setup/design of the token-based offline modality of the Digital Euro and what do each of its three properties (consistency, partition tolerance and availability) mean in the context of electronic payment services?

3.1 Distributed Systems and the CAP Theorem

The CAP theorem [Bre00, GL02] is a fundamental mathematical statement in distributed systems theory. These systems involve nodes that exchange data over a network through requests and responses. The theorem states that such a distributed system can only guarantee two out of the following three properties simultaneously:

- **Consistency (C)**: All nodes see the same data at the same time.
- **Availability (A)**: Every request receives a response, even if it may not contain the latest data.
- **Partition Tolerance (P)**: The system continues to operate, even in case of network partitions or communication breakdowns. A "partition" is a point in time where there exist two nodes A and B such that there is no communication path from A to B.

In order to answer the question considered in this section, we first need to adopt the terms "consistency", "availability", and "partition tolerance" to the context of offline digital currencies.

The CAP theorem considers a certain type of distributed systems, consisting of nodes that exchange data over a network through requests and responses. In particular, it does not consider systems outside this model, e.g., systems that additionally use secure hardware.

3.2 The CAP Theorem in the Context of Offline Digital Currencies

An offline digital currency, such as a token-based offline modality of the Digital Euro, for example, allows *peer-to-peer* transactions without real-time connectivity to any central entity at the time of the transaction. The CAP theorem applies to such a system in the following way.

- **Consistency (C)** refers to the situation that all parts of the system (wallets, devices, and any decentralised nodes) maintain a shared and correct view of the money in circulation.

  This includes ensuring that digital funds cannot be spent more than once ("double-spending") and that users cannot trick the system into thinking they have more money than they actually do. Even when transactions occur offline (for example, between two smartphones without internet access), the system must enforce that each transaction is valid and that balances are updated correctly.

  A consistent offline currency system also needs to ensure that, when offline devices later reconnect, the transactions performed offline are correctly and consistently integrated into the wider network without conflicts: in the end, all parts of the system must agree on who owns what, even if different parts of that information were temporarily stored or exchanged offline.

- **Availability (A)**: In the context of an offline digital currency, availability means that payments can be made at any time, regardless of whether a device is connected to the Internet or a central system. Thus, the system must allow two parties to exchange digital money directly and immediately, without waiting for online verification.

  Note that availability is **inherent to any offline-capable currency**, as it ensures that the digital currency system is usable when offline: payments are not blocked or delayed by connectivity issues. It guarantees that users can rely on the currency to function continuously, making it practical for everyday use, much like physical cash.

- **Partition Tolerance (P)**: Offline transactions naturally create network partitions, i.e., situations where devices or users are isolated from the rest of the system due to a lack of internet or network access. Despite these partitions, an offline currency must still allow users to make secure payments and manage their balances without relying on immediate communication with a server.

  Note that partition tolerance is **inherent to any offline-capable currency**, because being able to operate without online access to a central entity is the very definition of "offline". A partition-tolerant system ensures resilience and usability in the real world, where uninterrupted connectivity cannot be guaranteed.

CAP Property	Meaning in Digital Currency	Application to Offline Digital Euro
Consistency (C)	All devices reflect the same data	Sacrificed offline, restored when online
Availability (A)	Transactions processed anytime	Fundamental requirement for offline use
Partition Tolerance (P)	Tolerates loss of connection	Fundamental requirement for offline use

Table 1: Meaning and application of consistency, availability, and partition tolerance in the context of offline digital currency.

Since **Partition Tolerance** and **Availability** are fundamental requirements, the CAP theorem implies a fundamental limitation for an offline digital currency: the system cannot guarantee **Consistency**.
However, it is important to note that this implication is limited, as it applies only if an offline digital currency is viewed as a certain type of a distributed system. The theorem can be circumvented by considering additional safeguards, such as secure hardware elements or a semi-offline setting (Section 5.2.3).

The CAP theorem is often discussed in the context of distributed databases, but this is only a special case of its broader applicability to distributed systems in general. In particular, the "consistency" in CAP does not mean that every device must redundantly store the entire dataset. Instead, it refers to the guarantee that all nodes present a coherent view of the system's state according to the specified consistency model. This still holds when nodes only store partial data, such as a local transaction history for example, because what matters is that the data remains logically consistent across the system. Thus, the CAP theorem applies whether the system replicates all data everywhere or distributes subsets of data among its components.

3.3 On the Validity of the CAP Theorem

The question considered in this section asks about applicability of the CAP theorem "if it is valid". However, note that the above discussion and conclusion may hold independently of the validity of the CAP theorem as a general mathematical statement in distributed systems theory. That is, even if eventually a gap in the proof of the theorem is found, or the theorem turns out to be ambiguous [Kle15], less general than claimed, or even false (which is not expected, but it is not impossible), the main arguments discussed in this section may still apply to offline digital currencies when viewed as distributed systems in the above sense. **Although the question focuses on the CAP theorem, the reasoning applied to the specific case of offline digital currency, as well as the drawn conclusions, stand on their own and may remain valid independent of the CAP theorem.**

4 Achieving Consistency Through Secure Hardware

Considered Question

2. Considering the history and past (successful) attacks on secure (hardware) elements, is multi-spending thinkable and – if yes – what technical measures could be added to build further lines of defence against attempts to compromise the offline Digital Euro infrastructure's end-user devices (e.g. smartphones)?

4.1 Double-Spending Attacks and Secure Hardware

There is a long history of successful attacks on secure hardware elements such as smartcards, mobile secure enclaves, and Trusted Execution Environments (TEEs). See [Koc96, KJJ99, KJJR11, OP11, MIE17, WCP+17, DDE+18, VMW+18, CCX+19, MOG+20, CYS+21, CVM+21, vSY+24, SMvH+21, FQF+24] and the references therein, for example. These research papers are a non-exhaustive selection of works that have shown that it is possible to extract cryptographic keys, bypass authentication mechanisms, clone secure elements, inject malicious code, and exfiltrate sensitive data from hardware, often with minimal physical access or through side-channel and software-based techniques. More concretely, side-channel attacks on smartcards have enabled attackers to recover secret keys by analysing variations in power consumption during cryptographic operations [KJJ99, KJJR11]. In mobile secure enclaves and TEEs, researchers have demonstrated the ability to break isolation guarantees and extract secrets using microarchitectural attacks. Foreshadow [VMW+18] and Plundervolt [MOG+20] showed how TEEs such as Intel's SGX could be compromised via transient execution or voltage fault injection. Attacks like Checkm8, a bootrom exploit for iOS devices (see `https://github.com/axi0mX/ipwndfu`), demonstrate the potential for low-level, persistent, and unpatchable exploits that are capable of compromising millions of smartphone devices.

In particular the attacks demonstrated against mobile secure enclaves and TEEs highlight a fundamental challenge: the more complex the application logic running inside a supposedly-secure device, the harder it becomes to protect against vulnerabilities. Notably, some of these attacks can be carried out even remotely, without requiring the attacker to have direct access to the underlying hardware. Defending against adversaries who actually possess the physical device is an even greater challenge, since physical access typically enables more powerful attack vectors.

These references are just a small and rather random selection from a very large body of research on hardware security. There are many further examples of attacks where skilled attackers with technical knowledge

and appropriate tools were able to defeat the defences implemented by "secure" hardware components.

Thus, although secure hardware components are designed to protect sensitive operations, such as managing cryptographic keys, they have **repeatedly been shown to be vulnerable to a large variety of attacks** in both academic and real-world settings. In a setting where end-user devices like smartphones or smartcards are expected to store and transfer value autonomously and without real-time oversight, as in an offline digital currency, a compromised supposedly-secure but actually-insecure hardware element could allow a malicious actor to clone tokens, manipulate counters, or reset balances, thereby enabling double-spending or other attacks.

In this context, it is also very important to recognise that in the case of digital currencies, **the "attacker" and the "user" can be the very same individual**. This significantly raises the difficulty of achieving security, as an attacking user may have unlimited access and a lot of time to tamper with or analyse their own device.

Furthermore, standard hardware security modules focus on ensuring certain specific forms of confidentiality and integrity of data, e.g. as a secure key store for elementary cryptographic functionality. However, enforcing complex transactional protocol logic or ensuring the uniqueness of offline value transfers against malicious, and potentially technically very skilled, users is significantly more difficult to achieve.

The examples presented here should be understood as examples of concrete attacks that illustrate the existence of a broader threat. Some of these attacks are already well-mitigated in current hardware designs, but still the underlying threat remains very real. History shows that new attack techniques appear frequently and often just one clever new idea can yield a new practical exploit. Therefore it seems unwise to assume that hardware deemed secure today will remain secure indefinitely.

Here it should also be taken into account that replacing vulnerable hardware can be much more difficult, time-consuming, and expensive than patching insecure software. This dynamic must be carefully considered when designing systems that rely on hardware-based security assumptions, especially at large scale, as in the case of a Digital Euro.

Double-spending is clearly thinkable and must be considered very carefully, even when hardware elements are used as an additional layer of protection. Even if the hardware is considered secure for traditional cryptographic use cases, such as secure key storage, it may not provide adequate protection against the more complex fraud scenarios in offline digital currencies, such as double-spending attacks.

4.2 Mitigating Double-Spending Through Defence-in-Depth

To build stronger lines of defence and mitigate the risk of double-spending in offline digital currency systems, a comprehensive approach that combines and integrates technical, architectural, and operational safeguards is required. This may include:

1. Hardware-based safeguards, such as:
   (a) **Use secure hardware.** Employ tamper-resistant hardware elements that provide resistance to physical and logical attacks.
   (b) **Secure and auditable boot processes.** Ensure that end-user devices boot into a verified and trusted state using cryptographically signed firmware. Implement adequate measures to prevent unauthorised software from accessing or modifying secure modules.

(c) **Hardware diversity or homogeneity?** A homogeneous hardware monoculture, where many systems rely on the same or very similar platforms, can create a single point of failure. A successful attack against one instance of a widely deployed hardware architecture can be replicated at scale, compromising a large segment of the ecosystem. Using diverse secure hardware designs and implementations across device types may reduce the risk of a systemic failure caused by a universal vulnerability in a single hardware design.

However, diversity also comes with significant drawbacks. While it may mitigate the risk of universal failure, it also increases the complexity of the system very significantly. A diverse hardware landscape may introduce a wider range of potential vulnerabilities, arising from differences in design, implementation, supply chain, and update mechanisms. This can lead to more weaknesses, such as fragmented patching practises and a broader attack surface for attackers to explore.

In a technical landscape where end-user devices are predominantly smartphones or smart cards produced by a small number of manufacturers, the high degree of hardware homogeneity across the ecosystem increases the potential impact of a platform-specific vulnerability. This must be taken into account in the system design.

2. Resilient systems design, for example by implementing the following concepts:

(a) **Minimal reliance on hardware assumptions and hedged security** are advisable to maximise robustness against future hardware- or protocol-level attacks and to ensure broad compatibility across diverse devices and manufacturers.

While hardware-based solutions may be necessary in certain cases (particularly for enforcing double-spending resistance in offline settings), they inevitably introduce critical risks, including vulnerabilities to device compromise, dependence on trusted manufacturing processes, and susceptibility to side-channel attacks. Consequently, the design of a secure digital currency should, wherever possible, minimise such dependencies.

Cryptographic primitives like blind signatures (see Section 7) may offer well-understood and rigorously analysed guarantees for unforgeability and anonymity (along with possible additional security properties) that may hold independently of hardware security assumptions on a user's device.

Ideally, system components should be designed so that security holds as long as either the hardware or the cryptographic layer remains uncompromised, allowing one to serve as a safeguard in case the other fails. This dual-layered approach provides defence-in-depth, where hardware and cryptography complement each other, providing mutual resilience and enhancing overall robustness.

(b) **Transfer limits** and other risk-based rules, implementing controls to manage the risks associated with offline transactions. This may include limits on how much value transaction may have without an online connection, and how frequently they can occur. Additionally, configurable policies, such as capping the maximum offline balance a user can hold or limiting the number of peer-to-peer transaction hops, may help to contain the potential damage if a device or a class of devices is compromised. These rules must balance usability with privacy and security, ensuring that offline functionality is available without exposing the system to significant risk.

(c) **Delayed finality** refers to the approach where offline transactions may be considered provisional until they are confirmed upon reconnection. This means that the final settlement of such

transactions is delayed until proper validation can occur. To support this process, devices may maintain logs of their transactions, allowing for later auditing. These logs may help to ensure accountability and traceability in the event of disputes or suspected fraud during offline activity. However, maintaining such logs may pose risks to user privacy, making this a trade-off that must be carefully considered.

(d) **Token design with expiry or freshness constraints** involves using short-lived or time-limited tokens to reduce the risk associated with stolen or duplicated tokens. These tokens may include expiration timestamps to ensure they remain valid only for a limited period of times.This may limit the potential damage from token misuse and enhance overall security, particularly in offline or semi-connected environments. However, this approach must be designed in a way that avoids or minimises potential conflicts with availability, user experience, and the use of a digital currency as a long-term store of value.

(e) **Transparency** in the design and implementation of all system components is essential. This includes open specifications of the overall system design, exact security goals and attacker models for the system components, open-source software implementations and hardware design, and publicly documented architectures to allow independent security researchers and ethical hackers to scrutinise the system design and to identify vulnerabilities and contribute to strengthening its security. Systems that rely on obscurity or proprietary black-box designs may leave hidden flaws unaddressed, which may then be found and exploited by resourceful attackers, such as nation-state attackers or organised crime, and should therefore not be considered. By enabling external review and fostering a culture of openness, more robust security can be achieved. It also allows faster adaption to emerging threats and shifting attacker capabilities.

3. Monitoring on a technical and systemic level, such as:

(a) **Monitoring of emerging hardware vulnerabilities and new attack techniques.** The system operator(s) must implement a continuous monitoring of evolving threats and attack techniques. The monitoring should ensure that possible new techniques for cryptanalytic attacks, hardware tampering, or other system compromises are detected early and responded to appropriately. This reduces the window of exposure, helps maintain the integrity of the value stored on devices, and supports trustworthiness of the digital payment system.

(b) **Anomaly detection.** One could consider techniques that monitor the system for suspicious patterns, such as the same digital token being used more than once, which may indicate double-spending. Importantly, there are techniques that allow to perform this kind of analysis in a privacy-preserving way, such that user identities and transaction details remain hidden unless double-spending is detected. Only in this case the system would learn the identity linked to the offending transaction. See also Section 7.3 and Section 7.4. While a detailed analysis of such specific system designs and their applicability to an offline currency at the scale of a Digital Euro is a major effort and out of scope of the current document, such techniques should be considered and evaluated for the design of a secure offline modality of the Digital Euro.

(c) **Responsible disclosure framework.** Establishing a framework for responsible disclosure of vulnerabilities is essential. It must provide a clear communication channel to report security issues and ensure that the system operator is informed discreetly over discovered vulnerabilities, a patch or mitigation is developed, and users and the system are protected before the issue becomes public. Organisations that respond well to responsible disclosure demonstrate maturity in their security posture, openness to improvement, and commitment to protecting users.

Security researchers who uncover vulnerabilities should be able to report them without fear of legal repercussions. Encouraging responsible disclosure not only promotes a safer ecosystem, but also builds trust between researchers and system developers. Reward mechanisms such as bug bounty programs may incentivize the identification and reporting of flaws, accelerating the discovery and resolution of security issues. By supporting a safe and structured disclosure process, developers and system operators will benefit from a broader, engaged security community working collaboratively to improve the system's security and resilience.

A suitable combination of adequate security mechanisms depends on the concrete requirements, protocols, assumptions, threat- and attacker models, security goals, and system design. It may include a subset or all of the aforementioned techniques, as well as additional techniques.

4.3 Conclusions

Secure hardware elements may play a critical role in protecting the offline Digital Euro against double-spending and other attacks. However, potential vulnerabilities are a real and credible risk. A resilient design therefore should **not rely solely on hardware security**, but rather adopt a multi-layered defence approach, combining hardware assurances with cryptographic techniques, software safeguards, analytics, and risk-aware operational constraints. Transparency and a framework for trustworthy responsible disclosure of vulnerabilities are essential. These measures together can significantly reduce the feasibility and the impact of attacks and may significantly improve the system's security and resilience.

5 Alternative Technical Solutions

Considered Question

3. Are there any feasible alternative technical solutions to render the offline modality of the Digital Euro "cash-like", meaning both resilient in terms of double-spending and at the same time anonymous to use ...
   (a) while respecting the CAP theorem?
   (b) while accepting one out of three properties not to be fulfilled, e.g. an offline modality at the expense of "availability", and the consequences for practical implementation and usability?
   (c) with anonymity not for all parties but only/at least for the consumer/payee (natural person) as opposed to the merchant ("sender anonymity"), potentially resulting in a hybrid solution?

5.1 Challenges of Achieving Cash-Like Properties in an Offline Digital Currency

Achieving unforgeability, double-spending resistance, anonymity, and transferability in an offline digital currency presents significant technical and conceptual challenges. These challenges arise because these

properties can be in tension with one another, and because the digital environment lacks the physical constraints that naturally enforce these properties in traditional cash. Digital information is fundamentally very easily and perfectly copy-able, unlike physical objects such as coins or banknotes. This absence of inherent physical limitations creates core challenges for digital cash systems:

- **Double-spending resistance** in a digital currency requires technical safeguards, because the natural uniqueness of physical objects, which inherently prevent the same coin or banknote from being spent more than once, does not exist in the digital realm. This can be achieved either by designing the system in a way such that double-spending is *prevented* (cf. Section 7.3.1), or by enabling its *detection* (cf. Section 7.3.2) after the fact by implementing mechanisms to trace and identify dishonest users.

  In an *online* digital currency, where for every transaction there exists a communication channel to a trusted authority (such as a central bank or a distributed ledger as in blockchain-based cryptocurrencies), double-spending can be prevented by involving the trusted authority to verify every transaction in real-time before it is accepted. Every time a user wants to spend a digital token, the transaction is sent to this online system, which checks whether the token has already been spent. If not, the transaction is approved and recorded. Otherwise, the transaction is rejected. This process ensures that each digital token can be spent only once, because the central authority (possibly implemented by a distributed consensus protocol) keeps an up-to-date and consistent global record. **Importantly, this is also possible in a privacy-preserving way, without a global log of all transactions, using the techniques described in Section 7.** This provides strong guarantees against double-spending, without requiring the tokens themselves to be physically unique or tied to secure hardware, but inherently requires online connectivity of users (more precisely, of the recipient of the transaction) to the trusted authority.

  In an *offline* digital currency, double-spending prevention is much more challenging, because there is no access to a central authority at the time of the transaction. Since digital data can be easily and perfectly copied, the system must use technical mechanisms to either prevent or detect double-spending without real-time verification and in a way that enables anonymity.

  - One approach is to rely on secure hardware that enforces locally on the device that a token is spent only once. This hardware must be trusted not to leak tokens, private keys, or transaction data. Considering the risks discussed in Section 4 that are associated with this approach, secure hardware should be complemented by other technical and systemic measures. For instance, Section 7 describes an approach that uses cryptographic techniques to achieve strong anonymity properties in a way which is independent of the security of the hardware. A related approach is used in [ACK+24], for example.
  - Alternatively, systems can aim to detect double-spending retrospectively, by embedding cryptographic features in tokens that enable the system to identify dishonest users later. This allows the system to link and trace double-spent tokens to the attacker when the network is reconnected, while preserving user anonymity during honest use (cf. Section 7.3.2).

    Note that this approach raises additional risks. The longer the gap between a double-spending event and the moment of detection, the greater the attacker's opportunity to benefit from fraudulent transactions and to evade penalties. Offline conditions may even be adversarially induced, for example, if an attacker deliberately isolates devices or networks to extend this window of opportunity. These challenges become even more complex in cross-jurisdictional settings, such as when one party to an offline transaction operates outside EU oversight, where enforcement may be uncertain.

Moreover, the possibility of state actors engaging in or facilitating double-spending attacks adds another dimension, since they may possess the resources to manipulate network availability, compromise hardware, or exploit jurisdictional gaps at a significantly larger scale.

Thus, because offline environments lack the physical constraint of uniqueness and the online constraint of synchronised records, double-spending resistance must be carefully engineered with cryptographic techniques, hardware assumptions, or a combination of both.

- **Digital cash requires stronger unlinkability than physical cash.** Physical cash is often regarded as the paradigmatic example of an anonymous payment system, due to its pseudonymity and *practical* unlinkability. Crucially, however, cash is considered unlinkable *assuming* that the serial numbers on banknotes are not routinely recorded by individuals, merchants, or banks.

  In contrast, digital currencies operate on devices whose inner workings are not transparent to users, making hidden tracking much easier to implement and much more difficult to detect. As a result, ensuring that an offline digital currency provides the same "cash-like" anonymity as physical cash requires significantly stronger unlinkability guarantees. In Section 7, we discuss how advanced cryptographic techniques, such as blind signatures, can be used to achieve such strong unlinkability.

  > Physical cash remains effectively anonymous because (covert) banknote tracking is hard to implement, in particular at large scale. In contrast, digital currencies operate on devices whose opaque inner workings allow hidden surveillance to be implemented more easily and at much greater scale. This makes it necessary to require stronger forms of anonymity for digital currencies. These can be reliably enforced through appropriate cryptographic techniques (see Section 7).

- **Tension between accountable double-spending detection and anonymity.** If attacks cannot be reliably *prevented* from the outset, but only be *detected* after-the-fact, then it is necessary to identify malicious users, so that the attacking party can be penalised. This requires some form of user traceability, which may undermine anonymity to a certain degree. However, there are protocols which use advanced cryptographic techniques to implement detection tags that are revealed only if a coin is spent more than once. We will discuss such protocols in Section 7.4.

- **Long-term security.** An (offline) digital currency requires cryptographic techniques to ensure properties such as unforgeability, anonymity, or authentication between secure hardware components. Since the offline modality of the Digital Euro will be built for long-term use over many years, the long-term security of all system components must be taken into account.

  With respect to secure hardware assumptions, there must be careful consideration of the life-cycle of the devices, the robustness of their tamper-resistance, and the ability to withstand evolving attack techniques over time. This includes not only ensuring that hardware security modules, secure elements, or trusted execution environments provide strong protections at launch, but also that they can be updated, replaced, or revoked if vulnerabilities are discovered. Furthermore, assumptions about the trustworthiness of hardware must be realistic and minimise single points of failure, ensuring that even if some devices are compromised, systemic risks remain contained.

  With respect to the cryptographic algorithms, it is strongly advisable to design the system in a way that enables easy replacement of algorithms in the running system ("cryptographic agility"). This holds in

particular with the threat of potential future quantum attacks in mind. The system should be prepared for a potential situation where a replacement of cryptographic algorithms becomes necessary due to (quantum) cryptanalytic advances.

The vision of an offline digital currency that fully replicates the anonymity, security, and usability of physical cash is conceptually appealing, but realising it in practise requires navigating a complex landscape of technical trade-offs. The absence of physical constraints, the need for robust long-term security, the possibility of implementing hidden surveillance more easily and at much greater scale, and the tension between anonymity and traceability demand a carefully designed combination of privacy-preserving cryptographic techniques and may require trusted hardware.

5.2 Discussion of Solutions

5.2.1 Feasibility of "Cash-Like" Solutions under the CAP Theorem

As outlined in Section 3, any offline-capable digital currency system by definition requires partition-tolerance (P) and availability (A). Consequently, the CAP theorem implies that consistency (C) must be achieved using tools and techniques beyond the scope of the system model considered by the CAP theorem. Thus, to design a cash-like system, one approach is to accept the risk of temporary inconsistencies (i.e., double-spending) during offline use, and managing these risks through system-level safeguards. Such solutions include for example:

- **Secure hardware enforcement**: Devices enforce local consistency and prevent double-spending using secure hardware. However, this is subject to the limitations and attack vectors of secure elements that were discussed in Section 4.
- **Use of transaction limits**: Transaction and value limits per device reduce the risk if a device is compromised. However, such a solution must be carefully designed. Limits that are set too low may significantly reduce usability in practise, especially in environments where offline payments are expected to cover a wide range of daily expenses. Finding the right balance between security and practicality is therefore crucial: overly restrictive limits may discourage adoption, while overly generous limits increase the potential damage from compromise. A further caveat is that if enforcement of these limits relies solely on the compromised device itself, an attacker might circumvent them, particularly in an offline scenario where no central authority can immediately verify compliance. This must be taken into account when implementing such techniques. For example, it may require strong secure hardware assumptions, possibly in combination with a remote attestation infrastructure that allows a device to verify that it is communicating with a secure hardware element.
- **Reconciliation and detection**: One possible solution is to accept that inconsistencies may occur, but ensuring that they can be detected and traced upon reconnection, possibly through adequate cryptographic techniques that protect the anonymity of users while enabling reliable and non-framable tracing of double-spending users.

Achieving a cash-like experience under the CAP theorem necessitates a compromise on consistency and must instead rely on hardware-backed enforcement, after-the-fact fraud detection, and transaction constraints to manage the resulting risks.

5.2.2 Selective Violation of CAP Properties

Selective violation of the CAP theorem has the following consequences:

- **Violating Consistency (C)**: This case corresponds to accepting that double-spending may occur temporarily, if hardware assumptions fail. In this case, the risk can be addressed by safeguards such as transfer limits and auditability. This path prioritises usability at the cost of some systemic risk.
- **Violating Availability (A)**: Prioritising Consistency and Partition Tolerance would mean that transactions cannot proceed during offline periods. In effect, this leads to an *online-only* system, where offline payments are not possible, which is not a valid option for an *offline* Digital Euro. It also introduces strong usability limitations, in rural or emergency contexts with limited connectivity, for example.
- **Violating Partition Tolerance (P)**: This is also equivalent to abandoning the offline modality. However, one can also consider a *semi-offline* mode as a trade-off, which will be described as the first hybrid approach in Section 5.2.3 and in a more concrete context in Section 7.

Any distributed system intending to enable fully-offline, cash-like payments must accept that consistency cannot be achieved purely on the network protocol level. This impossibility can be circumvented through trade-offs in the offline requirement or secure hardware assumptions (see also Section 5.2.3 and Section 7).

5.2.3 Hybrid Solutions

There are also hybrid approaches that may be used to balance privacy, accountability, and regulatory compliance.

1. One approach to *prevent* double-spending is to consider a *semi-offline* setting, in which the sender of a transaction is offline, but the recipient is online at the time of payment. In this model, the sender transfers a digitally signed token (representing value) to the recipient, who then immediately checks the validity and uniqueness of the token by querying a central online authority as a "validation service". If the token has already been spent, the transaction is rejected. Otherwise, it is accepted and marked as redeemed by the central authority, preventing reuse and thus double-spending. The communication model in this case resembles the architecture of traditional credit card or point-of-sale systems, where only the merchant is required to have online access to authorise transactions. Importantly, strong privacy for the sender can still be preserved through the use of cryptographic techniques, such as blind signatures [Cha82], which allow tokens to be signed by the issuer without revealing their content. This approach is discussed in more detail in Section 7. As a result, the issuer and verifier can confirm the legitimacy of a token without learning who it belongs to. This approach offers a practical trade-off: while it does not support fully offline payments, it enables robust double-spending prevention with strong privacy guarantees, as long as the recipient maintains an online connection.
2. Another approach is to *detect* double-spending after-the-fact, by introducing an *auditor* as a designated entity that can reveal the identity of users only in cases of double-spending. In this setting,

cryptographic techniques can achieve that honest users remain fully anonymous (even to the auditor) during normal, so that it impossible to trace *honest* users. However, the user's identity is revealed if a user spends the same token more than once [Bra94, CHL05, FPV09]. Conditional traceability may provide accountability without undermining the anonymity of honest users. However, it is a trade-off that also introduces new risks. For instance, if a significant delay occurs between a double-spending event and its detection, it may become difficult to apprehend and penalise the attacker, especially if it has already moved to a different jurisdiction.

5.3 Conclusions

There exists ideas and basic concepts for technical solutions that might enable a "cash-like" offline Digital Euro, provided that one accepts fundamental limitations and trade-offs. Any offline modality inherently requires partition-tolerance and availability, thus, consistency must be achieved using additional techniques or assumptions, such as hardware enforcement or by considering a semi-offline approach, or the absence of consistency is accepted and mitigated through auditability-on-abuse mechanisms.

By carefully combining privacy-preserving cryptographic tools with secure hardware, a semi-offline system design, or conditional traceability, it may be possible to design a system that aligns with user- and security expectations and regulatory requirements. Hybrid designs, considering a semi-offline setting or using detection that preserves anonymity for honest users, may offer better trade-offs between security, anonymity, and usability and thus yield more practical solutions. However, such designs must be carefully studied and evaluated to ensure they provide robust guarantees in practise. Ultimately, the design must reflect a carefully calibrated balance between different requirements, guided by realistic threat models and aligned with societal goals.

A cash-like offline Digital Euro appears feasible, if trade-offs are accepted. Hardware assumptions may be considered to prevent double-spending attacks. Hybrid approaches that consider a semi-offline setting or enable detection of malicious users while preserving anonymity for honest users may offer a practical way to circumvent the impossibility implied by the CAP theorem.

6 Double-Spending Prevention for Transferable Digital Tokens

Considered Question

4. How can the token-based offline modality prevent double-spending in situations with several transactions in sequence and without any form of reconciliation in between? Conversely, how can a reconciliation process be implemented while preserving the cash-likeness of the token-based offline modality?

How can the token-based offline modality prevent double-spending in situations with several transactions in sequence and without any form of reconciliation in between? The prevention of double-spending in an *fully-offline* digital currency currently seems only possible under the assumption that the user's device contains secure hardware capable of reliably enforcing double-spending constraints, or by considering a weaker form of "offline" usability, such as the *semi-offline* setting. This applies to any form of offline transaction, be it a single transaction or a sequence of several transactions.

Even if the hardware itself is secure, the system must also ensure that users communicate with genuine, uncompromised devices. This typically requires *remote attestation*, a cryptographic protocol in which the device proves to a verifier (e.g., another user's device) that it is running authorised software in a trusted hardware environment. Implementing secure and reliable remote attestation is itself a complex challenge, particularly in a diverse and hostile device ecosystem. Thus, while hardware-based enforcement is necessary for offline double-spending prevention, it introduces substantial technical and operational challenges that must be carefully addressed.

Without secure hardware, a malicious user could duplicate tokens, enabling double-spending which is undetectable until devices reconnect. If the system is not able to rely on real-time reconciliation to enforce consistency, it must fully trust the device's internal safeguards to act as a proxy for the central authority present in an online system.

The prevention of double-spending in situations with several transactions in sequence and without any form of reconciliation in between hinges entirely on the security of the hardware and its resistance to tampering, cloning, or compromise. However, one may also consider hybrid approaches.

Conversely, how can a reconciliation process be implemented while preserving the cash-likeness of the token-based offline modality? To maintain cash-likeness, the system must uphold the key properties **unforgeability**, **double-spending resistance**, **anonymity**, and (possibly) **transferability**.

Unforgeability and anonymity of a token-based offline currency can be achieved using standard cryptographic techniques, such as blind signatures [Cha82], for example (cf. Section 7.2). Thus, the reconciliation process must focus on the double-spending resistance, both in case of transferable and non-transferable currencies. See Section 7.3 for a detailed discussion.

7 Technical Solutions for an Anonymous Token-Based Offline Modality

Considered Question

5. Considering existing research on the design choices of the Digital Euro, which technical solutions might be helpful for implementing the token-based offline modality while preserving anonymity?

A wide range of cryptographic techniques have been proposed to support privacy-preserving, token-based digital currencies. We will illustrate what is technically feasible by focusing on basic *blind signatures* as one particularly influential, well-understood, and readily available tool. This technique forms the basis of some of the earliest and most foundational approaches to anonymous digital payments. By examining its functionality, benefits, and limitations, we can gain concrete insight into how cryptographic techniques can support the implementation of an anonymous, token-based offline modality of the Digital Euro.

In what follows, we explain how blind signatures work, how they can be used to build a privacy-preserving, token-based, offline digital currency, and what additional mechanisms are needed to address challenges such as double-spending and transferability.

7.1 Blind Signatures

Blind signatures are a classical, very well-studied cryptographic tool. They are a form of digital signatures that enables anonymous digital payments. A token representing a "digital coin" corresponds to a digital signature of a random ID, and the signer (e.g., a bank) cannot link a given signature to a particular user.

7.1.1 Basic Functionality

A blind signature scheme involves two parties: the *user*, who wants a message m to be signed, and the *signer*, who is in possession of a secret signing key sk with corresponding public key pk. The signer uses sk to produce a digital signature s on m, but without ever seeing the message m or s. Using the public key pk, anyone is able to verify that s is indeed a valid signature for m, which must have been issued by the signer.

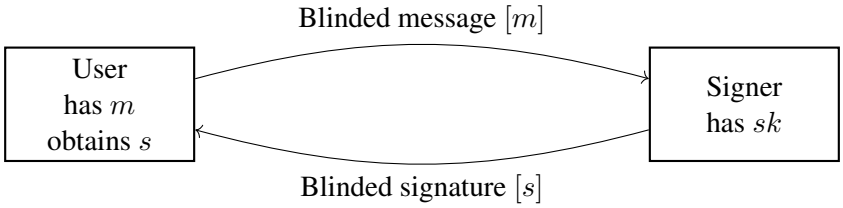


Figure 2: Basic blind signing functionality. s is a digital signature of message m under the signer's secret key sk. The corresponding public key pk is publicly available to any party.

7.1.2 The Blind Signing Protocol

The user and the signer interact in a basic *blind signing protocol* (cf. Figure 2) as follows:[2]

1. **Blinding:** The user begins with a message m and applies a blinding transformation to produce a hidden version of the message, denoted $[m]$. This blinded message is then sent to the signer.

   Conceptually, this is akin to writing the message on a letter and placing the letter inside an envelope.

2. **Signing:** The signer, possessing a private signing key sk, applies the blind-signing algorithm to the blinded message $[m]$ without learning anything about the original message. The result is a blinded signature, denoted $[s]$, which is returned to the user.

   Continuing the metaphor, this step corresponds to the signer placing their signature on the letter, but without opening the envelope. For example, one may imagine that the user puts the message m together with a carbon paper into the envelope. The signer signs the letter on the outside, and the carbon paper transfers the signature through the envelope onto the letter inside the envelope. In the digital world, blind signature schemes can achieve this by using well-known and practical standard cryptographic techniques.

[2]This blind signature protocol is accurate for certain constructions of blind signatures, but only a simplified illustration of the basic functionality for certain other constructions. For some blind signature schemes the protocol is more complex, e.g., requiring multiple rounds of interaction.

3. **Opening:** Upon receiving the blinded signature $[s]$, the user removes the blinding to obtain the final signature s on the original message m.

   This is analogous to opening the envelope and obtaining the original message m along with the signature s.

Note that the signer sees neither the message m that he signs, nor the signature s, but only the blinded values $[m]$ and $[s]$.

7.1.3 Security of Blind Signatures

The two standard properties expected from a blind signature scheme are:

- **One-more unforgeability:** Note that the user may obtain signatures for arbitrary messages from the signer. If a user interacts with the signer N times to obtain signatures for N (different) messages, it should be impossible for the user to obtain signatures for $N + 1$ different messages (i.e., it should be impossible to forge a signature for an additional message).

  Note that this property provides **security for the signer**, in the sense that the user is not able to create signed messages without interacting with the signer. When used in a digital currency, this will provide unforgeability of tokens.

- **Blindness:** Even if the signer receives a pair (m, s) consisting of a message m and a corresponding signature s, which were obtained by the user in a prior execution of the blind signing protocol, the signer is not able to link (m, s) to a specific execution of the blind signing protocol, and thus not to a specific user.

  Note that this property provides **security for the user**, in the sense that the signer is not able to link a given pair (m, s) to the user that has chosen m and received s from the signer. When used in a digital currency, this property will provide anonymity of users.

7.1.4 Realising Blind Signatures

Blind signatures can be realised in multiple ways and based on a large variety of different cryptographic techniques. For example, there exist constructions based on the *RSA* scheme [Cha82]), based on *discrete logarithm* techniques [Sch91, Sch90]), or based on *lattice* techniques [BLNS23], and many more.

As already mentioned in Section 5.1, the threat of quantum attacks must be taken into account. Currently, quantum computers are not considered an active threat for the *unforgeability* of modern blind signature schemes, and many schemes achieve *blindness* unconditionally (i.e., even against future quantum attacks). For such schemes, quantum-security is *currently* not required. However, this will change immediately with the advent of a quantum computer capable of breaking the security of the schemes in use. Therefore, all cryptographic algorithms should either be quantum-secure from the outset, or at least be implemented in a way that makes it possible to update or replace the algorithms in the running system to prepare for a future situation where such a replacement may become necessary ("cryptographic agility"). Independent of quantum attacks, cryptographic agility is strongly advisable, in order to be able to replace schemes in case of potential cryptanalytic advances.

7.2 Using Blind Signatures for Privacy-Preserving Digital Cash

Blind signatures can be used as a foundational building block in the design of privacy-preserving, token-based, offline digital cash systems. In such a system, a (central) bank can issue tokens that users are able to spend anonymously. This concept was first introduced in a seminal work by David Chaum [Cha82], a foundational result for anonymous electronic payment schemes. In the following, we outline the core ideas behind Chaum's protocol. This serves to illustrate the applicability of blind signatures as a technical solutions that might be helpful for implementing the token-based offline modality of a Digital Euro, and as a foundation for a discussion of limitations that require additional tools.

7.2.1 Basic Token-Based Digital Currency

For the basic token-based offline digital currency, we assume the following system setup:

- The (central) bank has a secret signing key sk for a blind signature scheme. The secret key sk is used to generate cryptographically-authenticated *tokens* as a digital equivalent of physical coins. The corresponding public key pk is publicly known.
- Every token has a unique *token-ID*, which "labels" the token and serves as a unique identifier. A new token is created upon withdrawal from a bank, and destroyed upon deposit. The central bank maintains a list of all token-IDs that have been "destroyed", i.e., of all tokens that have been in circulation and then deposited at a bank.
- For simplicity, we assume that tokens have a fixed value, e.g., every token corresponds to a fixed value of, say, 1 EUR. We will discuss later how to generalise this.

Based on this setup, the tokens are created, transferred, and deposited as follows.

- **Creation of tokens via blind withdrawal.** The withdrawal process creates new tokens. The user (more precisely, the "first payer" in Figure 3) interacts with the payer's bank, which in turn interacts with the central bank, as follows.

  1. The payer picks a token-ID m, which is a random number that uniquely identifies the token.

     By choosing the random number from a sufficiently large interval, one can ensure that token-IDs are *unique*, i.e., no two different token share the same token-ID, except for a small error probability. This error probability can easily be made small enough to make it reasonable to expect that it will never happen in practise. For example, choosing m as a random 32-byte number is sufficient for all practical purposes.

     Note that the list of token-IDs grows infinitely, and thus may become large. However, there are possible trade-offs that can be used to reduce the size of the list. For example, one could embed a timestamp in the token-ID, such as the month of withdrawal, and introduce a validity period for digital tokens. Then it is only necessary to store the list of token-IDs that belong to the current validity period. However, this is a trade-off in usability (due to the limited validity of digital tokens, which is parameterised by the size of the validity period) and also in anonymity: embedding a timestamp in the token-ID reduces the *anonymity set* from "all users of the system" to "all users of the system that withdrew a token at a time that is consistent with the timestamp in the token-ID". A more fine-grained timestamp reduces the list size and the anonymity set, while a more coarse-grained timestamp increases both anonymity and the list size.

2. The first payer starts the withdrawal process with the payer's bank. During this process, it runs a blind signing protocol (such as the one described in Section 7.1) with the central bank, where the messages are relayed between the payer and the central bank by the payer's bank, as illustrated in Figure 3. Thereby, the user obtains a token (m, s) consisting of a random token-ID m along with a signature s over m by the central bank that authenticates the token.
3. The payer's bank charges the corresponding amount to the user's account.

Importantly, the withdrawal is "blind", in the sense that neither the payer's bank, nor the central bank see the token-ID m or the corresponding signature s. They will only see the corresponding blinded values $[m]$ and $[s]$.

Note that the link between payer and payee is hidden by the blind signature. However, neither the withdrawal (despite being "blind"), nor the deposit of coins needs to be anonymous. Indeed, it rather seems that withdrawal must be tied to the user's identity to ensure funds are taken from the correct bank account. Thus, the blindness of the token-ID and the signature in the withdrawal process is merely to provide unlinkability between payer and payee, but not to achieve anonymous withdrawal. Intuitively, a token-ID resembles the serial number of a banknote, and the blindness during withdrawal technically ensures that the bank is not able to record the serial number of the banknote given to a customer.

- **Payment via transferal of tokens.** In order to transfer the token (m, s) from one user to another, i.e., from the "payer" to the "payee" in Figure 3, the payer simply transmits (m, s) to the payee. Upon receiving (m, s), the payee accepts the token only after checking that s is a valid signature for m with respect to the central bank's public key pk. This ensures that (m, s) is indeed a valid token, authenticated by the central bank. In this case, the payee accepts the token and the transaction is completed.

  Double-spending attacks. A crucial and fundamental technical challenge that we need to consider here are *double-spending* attacks, where a user transfers the same token to two different other entities (other users or banks). There are two approaches to deal with such attacks, which are either outright *prevention* or after-the-fact *detection*. This challenge can be approached in various ways, but it requires a thorough and detailed discussion of different approaches and their functionality, benefits, and limitations. We provide this in Section 7.3. For now, let us assume that there is some way to ensure tokens are not double-spent.

- **Deposit and destruction of the token.** In order to deposit the token in its bank account, the final payee transmits (m, s) to its bank. The bank accepts the token only after checking that s is a valid signature for m with respect to the central bank's public key pk. If this check is passed, then the payee's bank accepts the token and the corresponding amount is credited to the payee's bank account. Furthermore, the payee's bank tells the central bank to record the token with token-ID m as "destroyed".

  It is crucial to consider **double-spending attacks** here as well. However, since we assume that a bank is "online", it is significantly easier to detect double-spending during deposit of tokens. For instance, to check that a token (m, s) with pseudonym m has not been deposited at a bank before, the payee's bank may communicate with the central bank, to check whether the token with id m has been "destroyed". If yes, a double-spending attack is detected and prevented by not accepting the token.

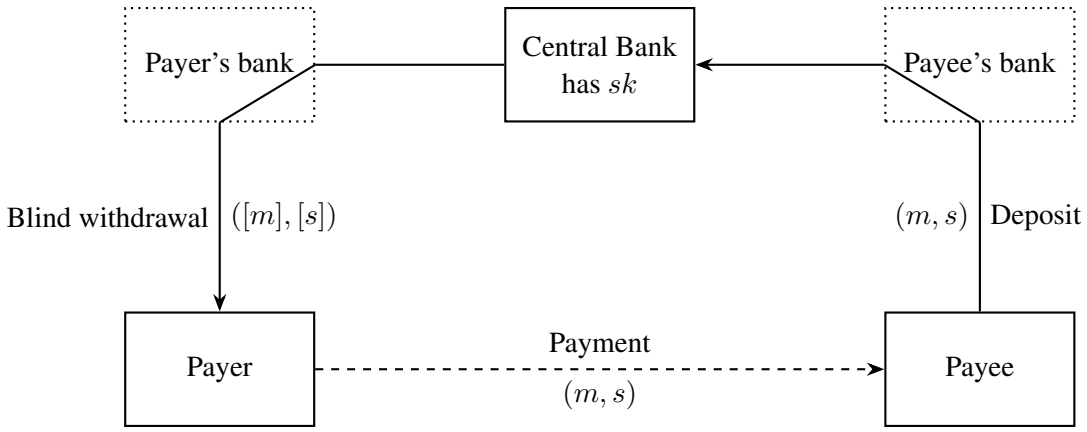


Figure 3: Illustration of digital cash using blind signatures. Solid lines depict transactions that may be considered "online", i.e., with an active communication channel to the central bank. While users are communicating with their retail bank, the retail banks may involve the central bank for certain operations, either by directly communicating with the central bank or by relaying messages between the user and the central bank. The dashed line depicts an"offline" transaction, which is performed directly between two users and possibly without communication to a central entity.

The digital currency described in this section is not a full-fledged design of a digital currency. It is meant as a simple example, to illustrate the use of technical solutions that may be helpful for implementing the token-based offline modality while preserving anonymity. It also provides a basis for the discussion of functionality, benefits, and limitations.

7.3 Analysis of Functionality, Benefits, and Limitations

We examine to which extent the basic blind-signature-based digital currency described above is "cash-like", in the sense defined in Section 2.4. We need to consider the four properties *unforgeability*, *double-spending resistance*, *anonymity*, and *transferability*. Let us first consider *unforgeability* and *anonymity*:

1. **Unforgeability:** Recall that unforgeability refers to the ability of creating *new* tokens. The *one-more unforgeability* property of the blind signature scheme provides that validly authenticated tokens can only be created by interacting with the central bank, i.e., with the approval of the central bank. No user is able to create more tokens than those that were signed by the central bank, i.e., more than those withdrawn from the bank.

   The basic digital currency is unforgeable, provided that the blind signature scheme is one-more unforgeable.

2. **Anonymity:** As explained in Section 2.4, *anonymity* requires that tokens are *pseudonymous* and *unlinkable*. Note that both properties are achieved by the basic digital currency described above. The token-ID m used to identify an individual token is a random number, without any reference to a user or

another entity of the system. When creating the token in the withdrawal process, the bank learns only that the user has withdrawn a token, and thus can reflect this in the user's bank account. However, due to the *blindness* property of the blind signature, the bank is *not* able to link the withdrawal process to the token-ID m of the token. Furthermore, the *blindness* property of the signature scheme also makes it impossible to link a token (m, s) which is transferred to a user or deposited at a bank to the user that withdrew the token.

> The basic digital currency is anonymous, provided that the blind signature scheme has the *blindness* security property.

It remains to discuss *double-spending resistance* and *transferability*.

7.3.1 Double-Spending Resistance

To understand how double-spending can be *prevented*, we consider three distinct settings:

1. *Fully offline, no hardware assumptions:* In this setting, users operate without internet connectivity and without relying on secure hardware. To ensure usability, the system must guarantee *availability*, and due to offline operation, it must tolerate *network partitions*. According to the CAP theorem (cf. Section 3), these two properties necessarily come at the cost of *consistency*.

   This lack of consistency is not merely a theoretical concern, but there is a concrete *double-spending* attack on the system that breaks consistency. This attack works as follows: A user withdraws a token (m, s) and transfers it to two different payees. Since both payees are offline, neither can check if the token has already been spent. As a result, both accept the same token. When the payees later attempt to deposit the token with the bank, only the first succeeds. The second is rejected because the token has already been "destroyed".

   > In a fully offline setting without secure hardware, double-spending cannot be prevented. This is a direct consequence of the CAP theorem.

2. *Fully offline, assuming secure hardware:* To overcome the above limitation, we can rely on *secure hardware*. Recall that this secure hardware may either be a dedicated device like a smart card or a secure environment within a smartphone (cf. Section 2.5).

   In this setting, tokens are stored and managed entirely within the secure hardware. The hardware ensures that:

   - Tokens can never be exported to untrusted environments.
   - All actions (withdrawal, transfer, deposit) happen through the secure hardware.
   - When a token is spent, the payer's hardware *permanently deletes* it.
   - The payee's hardware also enforces deletion upon onward transfer to a different user or deposit at a bank.

This guarantees that tokens cannot be duplicated or spent more than once, even when all parties are offline. Moreover, it allows for offline *transferability* between users without bank interaction.

It is important to emphasise that the security of this approach critically depends on ensuring that tokens are never exposed to untrusted environments. Moreover, whenever tokens are withdrawn, transferred, or deposited, it must be guaranteed that they remain within secure hardware at all times. This means that only secure hardware elements are allowed to participate in these operations. Consequently, each secure device and the banks must be able to verify that their communication partner is also a trusted secure hardware module, and that the communication channel is cryptographically secure. Achieving this typically requires additional infrastructure, such as remote attestation protocols, which are essential for maintaining the integrity and security of the system.

This is a very strong, but crucial requirement on the secure hardware: the system must remain secure even against malicious (and potentially resourceful) users that are in possession of a device and can attempt arbitrary tampering (cf. Section 4). This requirement must be fulfilled at any point in time, by all devices, and even against very resourceful attacks. Due to the large variety of known hardware attacks, it seems necessary to implement this approach only together with additional system-level security measures in place (see Section 4).

> Assuming that secure hardware that reliably prevents double-spending, it is possible to implement an offline digital currency with "cash-like" properties, that is, which is unforgeable, resistant to double-spending, anonymous, and transferable. However, this approach relies on the very strong assumption that the hardware remains secure even in the hands of malicious and resourceful users.

3. *Semi-offline, no hardware assumptions:* In this setting, the *payer is offline*, but the *payee is online* at the time of the transaction. This setup resembles traditional systems such as credit cards or point-of-sale terminals, where the merchant (payee) is online, but the customer (payer) is not (cf. Section 5.2).

   Upon receiving a token from the payer, the payee can immediately deposit it with the bank or query the central system to check whether the token has been spent and destroyed before. If not, then the system accepts it and marks it as destroyed, thereby preventing double-spending.

   Although this setting does not support true *transferability*, where tokens can be passed directly from one user to another, this functionality can be effectively emulated. Since the payee is online, they can immediately withdraw a new token after depositing the received one. This simulates the behaviour of a transferable digital currency while maintaining security against double-spending.

   > The semi-offline setting makes it possible to obtain a digital currency which is "cash-like", i.e., unforgeable, double-spending resistant, anonymous, and transferable, without very strong hardware assumptions. It is a trade-off between online requirements and trust in secure hardware. Importantly, note that the semi-offline setting provides anonymity despite there is some online verification of the validity of tokens.

7.3.2 Detecting Double-Spending

Rather than aiming to *prevent* double-spending entirely, an alternative approach is to focus on its *detection*, coupled with *accountability*, i.e., the ability to reliably identify users who double spend. The threat of penalties for detected misuse can serve as a deterrent. The basic idea of this approach is to introduce an *auditor* that can reveal the identity of users *only* in cases of double-spending, while honest users remain fully anonymous (even to the auditor) during normal use.

Digital currencies based on the basic protocol outlined in Section 7.2 can support such detection mechanisms by replacing the underlying blind signature scheme with a more advanced construction. One example is the e-cash scheme by Brands [Bra94], a cryptographic protocol for anonymous digital payments that provides strong privacy, unforgeability, and double-spending detection with accountability, and several follow-up works such as [CHL05, FPV09]. In these schemes, tokens remain unlinkable to a user's identity, *unless the same token is spent more than once*. In that case, the double-spending behaviour can be detected, and the misbehaving user's identity is revealed in a *non-framable* way. Although the original Brands scheme does not support *transferability* of tokens between users, later works have proposed transferable variants [BCFK15, BFQ21].

However, this approach requires an additional security property, namely *non-frameability* [Bra94], which provides that no entity (not even the central bank) can generate evidence of double-spending that falsely implicates an honest user.

A key drawback of the double-spending-detection-based approach is that it reveals identities of malicious users. It is presented here as another conceivable option that enables a fully-offline setting that provides anonymity for honest users without relying on strong hardware assumptions (cf. Table 2), thereby offering another option to circumvent the limitations implied by the CAP theorem.

If deanonymization of dishonest users at a global level (i.e., the Eurosystem) is not considered acceptable even when honest users remain unconditionally protected, a more decentralised approach could be considered, for example, through retail banks or payment service providers.

Furthermore, one could use cryptographic techniques such as secret sharing or threshold cryptography to ensure that multiple independent parties must cooperate before deanonymization of dishonest users is possible. While designing such a solution may be possible in principle with known cryptographic techniques, developing a concrete design would first require a precise specification of requirements.

If deanonymization of dishonest users is generally not acceptable, then alternative approaches can be considered to balance security, privacy, and system design requirements.

7.3.3 Beyond Fixed-Value Tokens

We have focused on fixed-value tokens, realised via basic blind signatures, because their conceptual simplicity makes it easy to describe and concretely illustrate the possibility of achieving unforgeability and strong anonymity, and the challenges and different options of achieving double-spending resistance. However, there is a large variety of cryptographic techniques that can be used to extend the functionality of this basic approach.

For example, Chaum's basic protocol based on blind signatures described in Section 7.2 issues tokens of fixed denominations. Supporting arbitrary payments (e.g., paying 2.75 EUR with a 5 EUR token) requires splitting tokens, or generating new tokens for change. At the same time, it must be ensured that no extra value is created or misused, and anonymity must be preserved. There are several cryptographic techniques to address the challenge of allowing a user to make a payment for less than the value of a token and receive

the remaining value as a new token (or tokens), without compromising privacy, unforgeability, or double-spending detection, such as [Oka95, CHL05, CG07, CPST15b, CPST15a, BPS19], for example.

Another practical approach that could be considered is to introduce tokens of varying denominations (e.g., 1 ct, 5ct, 1 EUR, 2 EUR, 10 EUR, 50 EUR etc.) and let users withdraw a collection of such tokens. The total value of withdrawn tokens may even exceed the desired withdrawal amount, if the secure hardware on the user's device ensures that the total value of tokens spent never exceeds the total value withdrawn. During a payment, the device selects and transfers the appropriate combination of tokens to match the payment amount, while enforcing that leftover tokens remain unused. This approach avoids real-time token splitting while preserving both unforgeability and anonymity, but assumes security of the hardware.

7.3.4 Discussion of Benefits and Limitations

The blind-signature-based approach offers several benefits but also comes with limitations. Its most important benefits are:

1. **Unforgeability:** The unforgeability of blind-signature-based tokens can be achieved in a very strong way, based on well-understood standard cryptographic techniques.

2. **Anonymity:** Thanks to the blindness property, the bank signs tokens without learning their identifiers. This ensures in particular that deposited tokens cannot be linked back to the withdrawal process, providing unlinkability and strong user privacy.

3. **Simplicity:** The basic design is conceptually simple and can be realised with standard techniques. This simple foundations can be extended with advanced techniques. For instance, it is possible to support double-spending detection, certain forms of accountability, or flexible denominations, if required by the overall system design.

4. **Cash-like payment flow:** Users can withdraw tokens from a bank and later spend them directly with merchants, closely resembling the way physical cash works.

The following limitations must be considered:

1. **Double-spending resistance:** The CAP theorem applies also to blind-signature-based protocols. Thus, like any other offline-capable digital currency, also the blind-signature-based approach requires secure hardware or semi-offline verification or a combination of both.

2. **Transferability:** Basic schemes do not allow *the same* token to be freely transferred between users without eventually involving the bank. If transferability is a requirement, then additional mechanisms or secure hardware assumptions are required.

3. **Accountability:** Since accountability inherently requires deanomymization of users, it fundamentally contradicts anonymity. This applies to digital cash in the same way as to physical cash. However, the digital world enables different trade-offs to balance user privacy with traceability by authorities. Both organisational and technical measures can be considered here, if this is required.

A digital currency where tokens are realised using blind signatures can achieve strong guarantees of anonymity and unforgeability without requiring secure hardware to achieve these properties. The basic protocol is conceptually simple and implements a cash-like payment flow. Double-spending resistance, transferability, and accountability depend heavily on system design choices. A fully offline setting fundamentally requires secure hardware. A semi-offline architecture can provide a different trade-off in terms of trust and usability. More advanced schemes can extend functionality by enabling accountability or flexible denominations, illustrating a rich design space for building truly "cash-like" digital currencies.

7.4 Other cryptographic tools for privacy-preserving digital tokens

The type of blind signature schemes considered in this section correspond to the original notion of blind signatures, as introduced by Chaum in [Cha82]. Numerous extensions of this concept have been developed, some of them quite recently, including *non-interactive* blind signatures [Han23, BCGY24, HPZ25] and *partially-blind* signatures [MS98, AO00, Oka06, BPV12, dPK22, KLLQ23], for example. Furthermore, there is a very large body of other cryptographic tools, which enable other forms of "privacy-preserving, unforgeable digital tokens" that may be useful for the design of a privacy-preserving digital currency. These include, for example, anonymous credentials [CL01, CV02, CL04, BCGS09, CKLM14], zero knowledge proofs [GMR89, BFM88, Gol01, GOS06, GS08, Gro16, BBB$^+$18], so-called OPRFs [CHL22], and general secure computation protocols [EKR18], for example.

The selection of appropriate cryptographic tools depends heavily on the specific requirements and constraints of the digital currency system under consideration. Without committing to a particular system design, it is arguably unproductive to delve into the subtle distinctions between these different notions. For this reason, we focus here on basic blind signatures, as a concrete example of a cryptographic tool that may be used in an offline modality of a Digital Euro. Our goal here is primarily to convey its essential workings, to clarify what guarantees privacy-preserving cryptography may offer and where its limitations lie.

When properly integrated into the system architecture, modern cryptographic techniques can enable digital cash to achieve key properties of physical cash, such as unforgeability and anonymity.

7.5 Conclusions

The design of a secure privacy-preserving digital currency requires a careful and holistic integration of different tools and techniques. For example, blind signatures can be easily used to build a digital currency that is *unforgeable* and *anonymous*. A fundamental challenge for any offline digital currency is to achieve *double-spending resistance*, which inherently requires trade-offs. A fully-offline setting requires secure hardware that prevents double-spending (see the second setting considered in Section 7.3.1). Alternatively, one may consider a *semi-offline* setting (third setting in Section 7.3.1), which uses online verification by the recipient of a transaction without sacrificing anonymity. One may also consider combinations of both approaches, such as using the semi-offline approach by default and the hardware assumption as a fallback.

The appropriate selection and integration of these tools depends on the overall system design and the specific security and functionality properties required by the system. A robust design must therefore select appropriate cryptographic tools and ensure that they are composed correctly within the system's architecture, balancing usability, efficiency, and threat resilience.

An anonymous offline modality of the Digital Euro appears feasible, provided that the system is designed based on a suitable combination of adequate cryptographic tools. A fully offline variant requires secure hardware (cf. Section 4.3) to prevent double-spending and enable transferability. A semi-offline modality could possibly be realised even without or based on weaker secure hardware assumptions. Importantly, while it appears technically feasible to implement a secure and anonymous offline modality for the Digital Euro, doing so would constitute a significant research and development effort.

The development of an offline modality of the Digital Euro requires a precise and comprehensive specification of all functional and security requirements, which can be used as a basis for the system design. This system design must clearly state the assumptions made about the underlying cryptographic primitives and the secure hardware components involved.

Both the requirement specification and the system design should be developed in a transparent manner to support and actively encourage external scrutiny and independent review.

An overview of different approaches and how they can achieve unforgeability, anonymity, double-spending resistance, and transferability is given in Table 2.

Scheme	**Payer offline**	**Payee offline**	**Unforgeability**	**Anonymity**	**Double-spending**	**Transferable**
Secure hardware only	✓	✓	HWA	HWA	Prevention, HWA	HWA
Blind sigs. + secure HW	✓	✓	Cryptographic	Cryptographic	Prevention, HWA	HWA
Blind sigs. (semi-offline)	✓	✗	Cryptographic	Cryptographic	Prevention	✓$^{(a)}$
Blind signatures online	✗	✗	Cryptographic	Cryptographic	Prevention	✓$^{(a)}$
Transferable e-cash	✓	✓	Cryptographic	Cryptographic$^{(b)}$	Detection only	✓

Table 2: Comparison of different basic approaches by their properties and required assumptions. Importantly, this table merely aims to illustrate some possible trade-offs, and many variants with different trade-offs and additional properties can be considered. A concrete recommendation of a specific variant can only be made based on a precise formulation of all functional requirements, security requirements, and acceptable risks and assumptions.
The columns "payer offline" and "payee offline" refer to the possibility of the respective user to be offline at the time of the transaction. "HWA" means that the considered property must be enforced by a secure hardware assumption. Note that the expected security properties get stronger (i.e., less realistic and more difficult to achieve) with an increasing number of different properties that the hardware has to provide. Extending existing hardware with additional functionality required by a digital currency may be difficult or impossible, in particular in a backwards-compatible way. Developing new hardware may be expensive and time-consuming. $^{(a)}$Can be emulated to the user. $^{(b)}$Only for honest users.

8 Conclusions

This report comes to the conclusion that an anonymous, cash-like offline modality of the Digital Euro seems technically feasible, but it requires a carefully orchestrated combination of cryptographic techniques, hardware assumptions, and systemic safeguards (Section 5.1).

Certain properties (most notably unforgeability and anonymity) can be guaranteed very strongly using well-understood cryptographic methods. Using adequate cryptographic tools, even certain forms of accountability, which seemingly contradicts anonymity, can be realised in a way that honest users are protected and deanonymization is only possible for malicious users (Section 7.3.2). In contrast, other key requirements, such as double-spending resistance and transferability, face inherent constraints rooted in the CAP theorem (Section 5.1, Section 6). These limitations do not preclude a solution, but they necessitate careful and pragmatic choices, such as the use of secure hardware assumptions or weakening the offline requirements by considering a semi-offline setting (Section 5.2.3). Importantly, the required security assumptions should be as weak (i.e., as realistic) as possible, and used only where inherently required (Section 2.5, Section 4.3). For example, secure hardware assumptions should not be used to achieve anonymity when this property can be achieved much more reliably using well-understood cryptographic techniques (Section 7.5). It must also be taken into account that even if a certain hardware device is considered secure for traditional cryptographic use cases, such as secure key storage, it may not provide adequate protection against the more complex fraud scenarios in offline digital currencies, such as double-spending attacks. Physical proximity emerges as a property that is difficult to enforce reliably in a digital context, and thus it appears more promising to pursue designs whose security does not depend on robust distance-bounding (Section 2.4.2).

The design of a Digital Euro must balance these trade-offs transparently, with clear requirements and robust threat models.

An anonymous, offline, cash-like Digital Euro is a credible prospect, but also a demanding design challenge. Achieving it requires a multi-faceted approach that combines strong cryptographic guarantees with realistic and well-understood hardware assumptions, trade-offs, and transparent and open design. If these elements are aligned, it seems possible to build an offline modality that offers the expected security, privacy, and usability properties.

On the necessity of open and transparent system design. Lack of cash-like anonymity and the risk of attacks are real and credible threats in a digital currency system, which may severely hinder its acceptance. The best way to mitigate and respond to these risks in a democratic and trustworthy way is to open up the entire system for public scrutiny. The digital currency system must:

1. Clearly document the requirements, security goals, and attacker models for every component.
2. Publish its protocols and the underlying security assumptions.
3. Provide open-source implementations of both software and hardware.
4. Actively encourage independent experts and ethical hackers to test, analyse, and help improve the system.

An open and transparent design not only delivers stronger and more future-proof security, it also ensures legitimacy in the eyes of citizens. By demonstrating that the Digital Euro is built in line with Europe's traditions of openness and protection of fundamental rights, one can foster the trust and acceptance necessary for its success.

A prime example of the great success of a transparent design approach is the DP-3T *("Decentralized Privacy-Preserving Proximity Tracing")* project [TBB+22], developed during the COVID-19 pandemic to enable contact tracing while preserving user privacy. Its core protocols, cryptographic assumptions, and threat models were made fully public from the outset, and the implementation was released as open source. This transparency allowed for wide-ranging independent analysis and critique, resulting in high public trust. Several European governments ultimately adopted DP-3T-based systems, precisely because their openness fostered both technical robustness and societal acceptance. This demonstrates how transparency and community engagement are practical prerequisites for secure and widely accepted digital systems.

References

[ABB+18] Gildas Avoine, Muhammed Ali Bingöl, Ioana Boureanu, Srdjan čapkun, Gerhard Hancke, Süleyman Kardaş, Chong Hee Kim, Cédric Lauradoux, Benjamin Martin, Jorge Munilla, Alberto Peinado, Kasper Bonne Rasmussen, Dave Singelée, Aslan Tchamkerten, Rolando Trujillo-Rasua, and Serge Vaudenay. Security of distance-bounding: A survey. *ACM Comput. Surv.*, 51(5), September 2018.

[ACK+24] Elli Androulaki, Angelo De Caro, Kaoutar El Khiyaoui, Romain Gay, Rebekah Mercer, and Alessandro Sorniotti. Secure and privacy-preserving CBDC offline payments using a secure element. Cryptology ePrint Archive, Paper 2024/1746, 2024.

[AO00] Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 271–286. Springer, Berlin, Heidelberg, August 2000.

[BBB+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018.

[BC94] Stefan Brands and David Chaum. Distance-bounding protocols (extended abstract). In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 344–359. Springer, Berlin, Heidelberg, May 1994.

[BCFK15] Foteini Baldimtsi, Melissa Chase, Georg Fuchsbauer, and Markulf Kohlweiss. Anonymous transferable E-cash. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 101–124. Springer, Berlin, Heidelberg, March / April 2015.

[BCGS09] Patrik Bichsel, Jan Camenisch, Thomas Groß, and Victor Shoup. Anonymous credentials on a standard java card. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM CCS 2009*, pages 600–610. ACM Press, November 2009.

[BCGY24] Foteini Baldimtsi, Jiaqi Cheng, Rishab Goyal, and Aayush Yadav. Non-interactive blind signatures: Post-quantum and stronger security. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part II*, volume 15485 of *LNCS*, pages 70–104. Springer, Singapore, December 2024.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988.

[BFQ21] Balthazar Bauer, Georg Fuchsbauer, and Chen Qian. Transferable E-cash: A cleaner model and the first practical instantiation. In Juan Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 559–590. Springer, Cham, May 2021.

[BLNS23] Ward Beullens, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Lattice-based blind signatures: Short, efficient, and round-optimal. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 16–29. ACM Press, November 2023.

[BPS19] Florian Bourse, David Pointcheval, and Olivier Sanders. Divisible E-cash from constrained pseudo-random functions. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 679–708. Springer, Cham, December 2019.

[BPV12] Olivier Blazy, David Pointcheval, and Damien Vergnaud. Compact round-optimal partially-blind signatures. In Ivan Visconti and Roberto De Prisco, editors, *SCN 12*, volume 7485 of *LNCS*, pages 95–112. Springer, Berlin, Heidelberg, September 2012.

[Bra94] Stefan Brands. Untraceable off-line cash in wallets with observers (extended abstract). In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 302–318. Springer, Berlin, Heidelberg, August 1994.

[Bre00] Eric A. Brewer. Towards robust distributed systems (abstract). In *Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing*, page 7. ACM, 2000. Invited Talk.

[CCX+19] Guoxing Chen, Sanchuan Chen, Yuan Xiao, Yinqian Zhang, Zhiqiang Lin, and Ten-Hwang Lai. SgxPectre: Stealing intel secrets from SGX enclaves via speculative execution. In *2019 IEEE European Symposium on Security and Privacy*, pages 142–157. IEEE Computer Society Press, June 2019.

[CFN90] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 319–327. Springer, New York, August 1990.

[CG07] Sébastien Canard and Aline Gouget. Divisible e-cash systems can be truly anonymous. In Moni Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 482–497. Springer, Berlin, Heidelberg, May 2007.

[Cha82] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 199–203. Plenum Press, New York, USA, 1982.

[Cha86] David Chaum. Showing credentials without identification: Signatures transferred between unconditionally unlinkable pseudonyms. In Franz Pichler, editor, *EUROCRYPT'85*, volume 219 of *LNCS*, pages 241–244. Springer, Berlin, Heidelberg, April 1986.

[CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 302–321. Springer, Berlin, Heidelberg, May 2005.

[CHL22] Sílvia Casacuberta, Julia Hesse, and Anja Lehmann. SoK: Oblivious pseudorandom functions. In *2022 IEEE European Symposium on Security and Privacy*, pages 625–646. IEEE Computer Society Press, June 2022.

[CKLM14] Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable signatures: New definitions and delegatable anonymous credentials. In Anupam Datta and Cedric Fournet, editors, *CSF 2014 Computer Security Foundations Symposium*, pages 199–213. IEEE Computer Society Press, 2014.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, Berlin, Heidelberg, May 2001.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, Berlin, Heidelberg, August 2004.

[CPST15a] Sébastien Canard, David Pointcheval, Olivier Sanders, and Jacques Traoré. Divisible E-cash made practical. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 77–100. Springer, Berlin, Heidelberg, March / April 2015.

[CPST15b] Sébastien Canard, David Pointcheval, Olivier Sanders, and Jacques Traoré. Scalable divisible E-cash. In Tal Malkin, Vladimir Kolesnikov, Allison Bishop Lewko, and Michalis Polychronakis, editors, *ACNS 2015*, volume 9092 of *LNCS*, pages 287–306. Springer, Cham, June 2015.

[CV02] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In Vijayalakshmi Atluri, editor, *ACM CCS 2002*, pages 21–30. ACM Press, November 2002.

[CVM+21] Zitai Chen, Georgios Vasilakis, Kit Murdock, Edward Dean, David Oswald, and Flavio D. Garcia. VoltPillager: Hardware-based fault injection attacks against intel SGX enclaves using the SVID voltage scaling interface. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 699–716. USENIX Association, August 2021.

[CYS+21] Jinhua Cui, Jason Zhijingcheng Yu, Shweta Shinde, Prateek Saxena, and Zhiping Cai. SmashEx: Smashing SGX enclaves using exceptions. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 779–793. ACM Press, November 2021.

[DDE+18] Fergus Dall, Gabrielle De Micheli, Thomas Eisenbarth, Daniel Genkin, Nadia Heninger, Ahmad Moghimi, and Yuval Yarom. CacheQuote: Efficiently recovering long-term secrets of SGX EPID via cache attacks. *IACR TCHES*, 2018(2):171–191, 2018.

[dPK22] Rafaël del Pino and Shuichi Katsumata. A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 306–336. Springer, Cham, August 2022.

[EKR18] David Evans, Vladimir Kolesnikov, and Mike Rosulek. A pragmatic introduction to secure multi-party computation. *Found. Trends Priv. Secur.*, 2(2-3):70–246, 2018.

[Eur23] European Commission. Proposal for a regulation of the european parliament and of the council on the establishment of the digital euro. `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52023PC0369`, June 2023. COM(2023) 369 final, CELEX: 52023PC0369.

[FDC11] Aurélien Francillon, Boris Danev, and Srdjan Capkun. Relay attacks on passive keyless entry and start systems in modern cars. In *NDSS 2011*. The Internet Society, February 2011.

[FPV09] Georg Fuchsbauer, David Pointcheval, and Damien Vergnaud. Transferable constant-size fair e-cash. In Juan A. Garay, Atsuko Miyaji, and Akira Otsuka, editors, *CANS 09*, volume 5888 of *LNCS*, pages 226–247. Springer, Berlin, Heidelberg, December 2009.

[FQF+24] Dengguo Feng, Yu Qin, Wei Feng, Wei Li, Ketong Shang, and Hongzhan Ma. Survey of research on confidential computing. *IET Commun.*, 18(9):535–556, 2024.

[GL02] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33(2):51–59, 2002.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, Berlin, Heidelberg, May / June 2006.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Berlin, Heidelberg, May 2016.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Berlin, Heidelberg, April 2008.

[Han23] Lucjan Hanzlik. Non-interactive blind signatures for random messages. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 722–752. Springer, Cham, April 2023.

[HPZ25] Lucjan Hanzlik, Eugenio Paracucchi, and Riccardo Zanotto. Non-interactive blind signatures from RSA assumption and more. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part II*, volume 15602 of *LNCS*, pages 365–394. Springer, Cham, May 2025.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer, Berlin, Heidelberg, August 1999.

[KJJR11] Paul C. Kocher, Joshua Jaffe, Benjamin Jun, and Pankaj Rohatgi. Introduction to differential power analysis. *Journal of Cryptographic Engineering*, 1(1):5–27, April 2011.

[Kle15] Martin Kleppmann. A critique of the CAP theorem. *CoRR*, abs/1509.05393, 2015.

[KLLQ23] Shuichi Katsumata, Yi-Fu Lai, Jason T. LeGrow, and Ling Qin. CSI-Otter: Isogeny-based (partially) blind signatures from the class group action with a twist. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part III*, volume 14083 of *LNCS*, pages 729–761. Springer, Cham, August 2023.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 104–113. Springer, Berlin, Heidelberg, August 1996.

[KW05] Ziv Kfir and Avishai Wool. Picking virtual pockets using relay attacks on contactless smartcard systems. Cryptology ePrint Archive, Report 2005/052, 2005.

[MIE17] Ahmad Moghimi, Gorka Irazoqui, and Thomas Eisenbarth. CacheZoom: How SGX amplifies the power of cache attacks. In Wieland Fischer and Naofumi Homma, editors, *CHES 2017*, volume 10529 of *LNCS*, pages 69–90. Springer, Cham, September 2017.

[MOG+20] Kit Murdock, David Oswald, Flavio D. Garcia, Jo Van Bulck, Daniel Gruss, and Frank Piessens. Plundervolt: Software-based fault injection attacks against intel SGX. In *2020 IEEE Symposium on Security and Privacy*, pages 1466–1482. IEEE Computer Society Press, May 2020.

[MS98] Shingo Miyazaki and Kouichi Sakurai. A more efficient untraceable e-cash system with partially blind signatures based on the discrete logarithm problem. In Rafael Hirschfeld, editor, *FC'98*, volume 1465 of *LNCS*, pages 296–308. Springer, Berlin, Heidelberg, February 1998.

[Oka95] Tatsuaki Okamoto. An efficient divisible electronic cash scheme. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 438–451. Springer, Berlin, Heidelberg, August 1995.

[Oka06] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 80–99. Springer, Berlin, Heidelberg, March 2006.

[OP11] David Oswald and Christof Paar. Breaking Mifare DESFire MF3ICD40: Power analysis and templates in the real world. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES 2011*, volume 6917 of *LNCS*, pages 207–222. Springer, Berlin, Heidelberg, September / October 2011.

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, New York, August 1990.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, January 1991.

[SMvH+21] Carlton Shepherd, Konstantinos Markantonakis, Nico van Heijningen, Driss Aboulkassimi, Clément Gaine, Thibaut Heckmann, and David Naccache. Physical fault injection and side-channel attacks on mobile devices: A comprehensive analysis. *Comput. Secur.*, 111:102471, 2021.

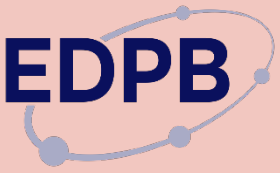
[TBB+22] Carmela Troncoso, Dan Bogdanov, Edouard Bugnion, Sylvain Chatel, Cas Cremers, Seda F. Gürses, Jean-Pierre Hubaux, Dennis Jackson, James R. Larus, Wouter Lueks, Rui Oliveira, Mathias Payer, Bart Preneel, Apostolos Pyrgelis, Marcel Salathé, Theresa Stadler, and Michael Veale. Deploying decentralized, privacy-preserving proximity tracing. *Commun. ACM*, 65(9):48–57, 2022.

[VMW+18] Jo Van Bulck, Marina Minkin, Ofir Weisse, Daniel Genkin, Baris Kasikci, Frank Piessens, Mark Silberstein, Thomas F. Wenisch, Yuval Yarom, and Raoul Strackx. Foreshadow: Extracting the keys to the intel SGX kingdom with transient out-of-order execution. In William Enck and Adrienne Porter Felt, editors, *USENIX Security 2018*, pages 991–1008. USENIX Association, August 2018.

[vSY+24] Stephan van Schaik, Alexander Seto, Thomas Yurek, Adam Batori, Bader AlBassam, Daniel Genkin, Andrew Miller, Eyal Ronen, Yuval Yarom, and Christina Garman. SoK: SGX.fail: How stuff gets eXposed. In *2024 IEEE Symposium on Security and Privacy*, pages 4143–4162. IEEE Computer Society Press, May 2024.

[WCP+17] Wenhao Wang, Guoxing Chen, Xiaorui Pan, Yinqian Zhang, XiaoFeng Wang, Vincent Bindschaedler, Haixu Tang, and Carl A. Gunter. Leaky cauldron on the dark land: Understanding memory side-channel hazards in SGX. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 2421–2434. ACM Press, October / November 2017.

[ZPZ+16] Christian T. Zenger, Mario Pietersz, Jan Zimmer, Jan-Felix Posielek, Thorben Lenze, and Christof Paar. Authenticated key establishment for low-resource devices exploiting correlated random channels. *Comput. Networks*, 109:105–123, 2016.


EDPB