The International Journal of Parallel, Emergent and Distributed Systems
Vol. , No. , , 1–28

RESEARCH ARTICLE

Distributed Architectures for Electronic Cash Schemes: A Survey

Isabelle Simplot-Ryl[1,2] and Issa Traoré[3] and Patricia Everaere[1]

[1]Lifl Cnrs Umr 8022/[2]Inria Lille-Nord Europe
Université de Lille I, Cité Scientifique
F-59655 Villeneuve d'Ascq Cedex, France
[3]University of Victoria
Department of Electrical and Computer Engineering
PO Box 3055 STN CSC
Victoria, BC, V8W 3P6, Canada
()

The volume of E-commerce transactions has considerably increased in the last several years. One of the most important aspects of such progress is the efforts made to develop and deploy dependable and secure payment infrastructures. Among these infrastructures is electronic cash, which is an attempt to reproduce the characteristics of paper cash in online transactions. Electronic cash schemes have so far been the purpose of a significant amount of research work. Although real-life deployments of such schemes are expected to take place in highly distributed environments, limited attention has been paid in the literature on underlying architectural issues. So far the focus has mostly been on addressing only security issues. However, for real-life deployment, distributed processing criteria such as performance, scalability, and availability are of prime importance. In this paper, through a survey of the literature, we identify and analyze the different distributed architectural styles underlying existing e-cash schemes. We discuss the strengths and limitations of these architectures with respect to fundamental system distribution criteria. In light of such discussion, we make some recommendations for designing effective distributed e-cash systems from an architectural perspective.

Keywords: Electronic Cash, Distributed Architecture, Security, Dependability, Electronic Commerce, Software Architecture.

1. Introduction

About a decade earlier growing concerns have been expressed in the academic and financial communities about the future and safety of electronic commerce [53]. At that time the consensus was that while the volume of e-commerce transactions and activities was steadily increasing, a significant threat that could limit that growth was the lack of secure electronic payment methods. With the considerable technological progress achieved in this front over the last decade, this discussion seems now to belong to the past. Now, similar concerns are being raised and discussions have started about the future of electronic cash. Considering the amount of work achieved so far in this area, we can reasonably be hopeful that in about a decade from now the usage of e-cash [2, 3, 31, 37] would be common practice as is the case now for other electronic payment methods and systems such as credit cards. For e-cash to reach that level of acceptability, it must exhibit at least all the key characteristics of physical cash such as *anonymity, transferability*, and *security.* Despite the amount of literature produced, a universal e-cash scheme has yet to be enacted. So far, many cash schemes have been proposed, which do not necessarily converge, and tend to focus only on a limited subset of expected properties. Only

ISSN: 1744-5760 print/ISSN 1744-5779 online
© Taylor & Francis
DOI: 10.1080/1744576YYxxxxxxxx
http://www.informaworld.com

few of the proposed schemes are actually being used for online payment without the underlying support of some other electronic payment methods such as credit cards.

To date, research on e-cash has been directed primarily towards addressing security requirements through the design of suitable security protocols and mechanisms. The implementation and real-life deployment of these schemes, which are inherently distributed, have been overlooked or only lightly covered in the literature. However, actual implementation of the proposed protocols and mechanisms raises a lot of practical distributed processing issues including scalability, performance, and availability. We believe that scalability and performance issues are the main threats to real-life deployment and use of e-cash. The main reason being that in general the security protocols and mechanisms underlying current e-cash schemes have not been designed and analyzed by taking into account constraints specific to distributed environments. Such constraints involve various dimensions, as noted by Jakobsson and Juels, "including minimized human involvement, improved distribution of goods and information, and more rapid processing of transactions" [25].

Despite the limited coverage of architectural issues in the literature, different distributed architectural styles have emerged from the various e-cash schemes proposed so far. The goal of this paper is to identify and analyze these architectural styles through a survey of the literature. We highlight and discuss the strengths and weaknesses of these architectures with respect to fundamental distributed processing criteria. Based on the outcome of such discussion, suitable recommendations are made on designing and deploying secure and dependable distributed e-cash schemes that address concerns of customers, merchants, and financial institutions, and make necessary tradeoffs between security and architectural goals.

The rest of the paper is organized as follows. In Section 2, we present the key requirements for universal electronic cash, and summarize the basic approach used to address them in the literature. In section 3, we discuss distribution requirements and challenges, and present possible approaches and architectures proposed in the literature for e-cash systems. In sections 4 and 5, we survey and discuss sample proposals made in the literature in the light of the distributed architectural styles identified in the previous section. In section 6, we make some recommendations concerning e-cash system implementation from security and distribution perspective. Finally in Section 7, we make some concluding remarks.

2. E-Cash Overview

In this section, we review most of the key properties that could be expected from an e-cash system, and summarize general modus-operandi for a typical system.

2.1 E-Cash Requirements

Over decades several key concepts have driven the design of e-cash systems. Initially a small subset of properties were defined and implemented. As the field has been maturing, more and more properties were identified. Some of the properties cover security and privacy including *anonymity, pseudonymity, untraceability, unforgeability, no framing, double spending prevention* and *auditability.* Other properties target practicality or user convenience including *transferability, fairness* and *recoverability.* Several of these concepts such as transferability, anonymity, (and so on) have received many different interpretations in the literature. In this section, we provide our own understanding of these concepts and attempt whenever possible to base our definition on *ISO 15408 common criteria*. Popular concepts include the following:

Anonymity: Impossibility for anyone to determine the true identity of a user associated with a subject, an operation, or an object.

Pseudonymity: Use of pseudonyms allowing a user to remain anonymous in transactions involving an information system, by still remaining liable in case where she commits some illegal actions.

Untraceability: Impossibility for anyone to establish a link between different transactions performed by the same user in one or many different information systems at different times.

Transferability: In [13], transferability is defined as the possibility for a coin to circulate between people, offline, without involving the bank or any other central authority. A broader definition is used in [55], where a transferable coin is a coin that "can be circulated among people" regardless of whether the transactions are online of offline. We adopt in this work, the previous definition which is more meaningful from system distribution perspective.

Double spending prevention: Impossibility for anyone to spend a coin more than once.

Unforgeability: Impossibility for unauthorized parties to create new coins.

No framing: Impossibility for anyone else other than the owner of a coin to spend it.

Fairness: Impossibility for anyone to get away with malicious behavior in a transaction. It should be possible to uncover the identity and actions of an individual behaving maliciously in a particular transaction without revealing any further information about other transactions.

Recoverability: refers to the restoration of the values and integrity of coins in situations where the computing environment or device containing the coins fails or is stolen in case, for instance, of a smart card.

Auditability: a cash scheme is auditable if it involves some built-in mechanims allowing a trusted third-party or authority, referred to as an *auditor* to monitor the money supply. Any new valid coin that is injected in the monetary system should be known to the auditor.

Although all the above e-cash properties are important, only a subset of these properties have direct impact on system distribution. Examples of such properties include *transferability*, and *fairness*. For instance, in some proposals to ensure *fairness*, a group signature scheme in which the consumers are group members and a trusted third-party is the group manager is used [33]. In this case the role of the trusted third party is only to register new group members and to resolve conflict; regular transactions are carried through the remaining participants. To ensure *anonymity*, in some proposals, the amount of (secret) knowledge accessed by the bank is spread between the bank and the central bank [55]. This limits the actual knowledge carried by each of these participants, and as such reduces their capability to trace back payment transactions on their own.

2.2 E-Cash Protocols and Mechanisms

A basic e-cash scheme typically involves three kinds of participants:

- *Payer:* the consumer or customer receiving some service or goods in exchange of some monetary payment.

- *Payee:* the merchant or service provider receiving monetary compensation in exchange of some service or goods provided to the consumer.
- *Financial Institution:* provides required financial infrastructures and services underlying payment transactions between payer and payee. This may include a network of banks, financial brokers, credit card companies etc. This player is commonly referred to as the *bank*; we will use this terminology in the rest of the paper.

Three types of transactions underlie e-cash schemes:

- *Withdrawal:* is the operation during which a consumer purchases or acquires some coins with a bank.
- *Payment:* corresponds to the transfer of e-cash from the payer to the payee in exchange of goods or services.
- *Deposit:* is the operation through which the recipient of some coins redeems these coins at a bank. In general, *payee* and *payer* have different banks. This means that the *deposit* phase is actually more complex; (behind the scene) the payer's bank would have to contact the *payee*'s bank to redeem the coins before crediting his account.

Two kinds of e-cash schemes are considered, according to whether or not the bank is involved (immediately) in payment phase: *online* and *offline*:

- *Online* scheme involves checking the validity of the coins at the bank before accepting the payment.
- *Offline* scheme does not require active participation of the bank during payment transaction; the validity of the coins can be checked by other means on the spot or at a later time (after payment completion) through the bank.

One of the key differences between electronic banking (online) and traditional banking (at a branch) is that the former occurs through untrusted medium (i.e., the Internet) that is not under the control of the bank. Therefore, necessary steps must be taken to ensure the security of the payment transactions over such medium. Two kinds of security properties must be fulfilled in this case, namely *privacy* and *authenticity* [29]. *Privacy* consists of protecting against unauthorized disclosure of sensitive or personal information. *Authenticity* can be achieved by ensuring *user identification*, *message integrity*, and *nonrepudiation. Authenticity* can be achieved by implementing appropriate *authentication infrastructure*. An important aspect of such infrastructure is key management, which is carried out by a *certification authority* (CA).The CA is a trusted third party which issues digital certificate carrying the identity and related proof for the players involved in payment transactions. Hence, the authentication infrastructure or at least part of it (e.g., CA) is an external entity, separate from the bank.

Privacy requires both *payer anonymity* and *payment untraceability.* One of the side-effects of *privacy* is that it provides a fertile ground for counterfeit. E-cash schemes can be subject to two kinds of counterfeit, namely *forgery* and *multiple spending.* While *authenticity* features such as *user identification* and *message integrity* can protect against forgery, multiple spending protection still poses some challenges. The most widely used approach to combat multiple spending consists of maintaining at the bank a database of spent coins. This can be used to reject transactions involving multiple-spent coins in online transactions, or to identify occurrence of multiple-spending in offline payment. Maintaining, however, a database of spent coins poses important system distribution challenges [54].

3. Outline of Distributed Architectures for E-Cash

In this section, we summarize key distribution criteria and challenges, and identify and discuss the main distributed architecture styles emerging from existing e-cash schemes.

3.1 Distribution Criteria and Challenges

Key requirements to be considered when designing a distributed architecture include *scalability, openness, heterogeneity, security, availability,* and *performance.* As indicated earlier, so far the focus in e-cash research has primarily been on addressing security requirements; limited attention has been paid to architectural and implementation issues.

An important aspect of the design of e-cash system architecture is applicability to real-life context and concerns. In effect, for a new payment scheme to be adopted by professionals, it ought either to achieve significant cost reduction so as to justify investing in required new infrastructure, or to bring notable additional service to customers. At the same time, the average customer must enjoy additional benefit when using the new system without incurring extra cost.

There are several benefits in deploying a software-based e-cash system (without requiring any extra hardware) including the following:

- Simplification of payment procedures for customers through the use of the same mechanism anywhere and under all circumstances, whether it is at a merchant or online etc.
- Flexibility in using the payment scheme due to the elimination of security constraints related to the hardware infrastructure underlying payment cards.
- Elimination of special-purpose (hardware-based) payment terminals, which are expensive, difficult to maintain, and quite often have limited lifespan.

The above advantages rely, of course, on deploying a system that is not only highly reliable and secure, but is also efficient. Most papers in the literature tend to focus solely on protocols design, underlying security issues and required functionality. By studying, however, carefully the interactions between the different parties or roles involved in the proposed payment schemes, it is clear that several practical and technical requirements of these highly distributed systems must seriously be considered when it comes to their real-life deployment. We discuss these aspects in the following:

Algorithms Distribution. Several important aspects of system distribution must be taken into account in the design of the proposed e-cash protocols and algorithms. In particular, aspects such as concurrency, synchronization and resource sharing can be the cause of specific security weaknesses. So it is necessary to check the impact of these distribution requirements on the proposed algorithms and protocols, and make necessary trade-offs.

Furthermore, universality is an essential prerequisite for electronic cash scheme to be widely adopted: a consumer must be able to perform some payment at a merchant regardless of whether or not both of them have the same bank. As a matter of fact, for instance, all the schemes requiring a spent coin database must involve some form of distribution. Since online transactions may occur simultaneously, the spent coin database must be updated in real-time for effective detection of double-spending. As a result, to handle the load corresponding to requests for verification, it seems natural to replicate or distribute the spent coin database. However, this would open up classical issues related to coherence and update in distributed database architectures that should be addressed.

Performance. It is essential that proposed e-cash system be efficient; the time required for payment processing should be relatively small, whether it is at a cashier in a supermarket or online. For instance, online, beyond certain duration the customer is likely to drop the whole transaction and may not return anymore to the same site. Therefore, processing load and latency carry an important weight in the success of such system. Although payment card schemes are relatively efficient, in general, they cover only payments above certain amounts and tend to slow down under huge traffic or in peak periods. Proposed e-cash schemes must adopt distribution strategies adapted to the type and volume of transactions targeted, taking into account performance issues.

Scalability. Real-life deployment of e-cash schemes means large scale deployment. For e-cash to be viable from business perspective, the system must be scalable and able to handle large user population generating large volume of transactions. According to Anderson *et al*, peak traffic for current online payment systems typically occurs at 1pm on the Saturday before Christmas [2]. Large scale deployment necessarily requires a distribution of the role of the bank to reallocate the load involved while proposed protocols require global coherence of data during verification operations.

Besides the above quality requirements, distributed systems have several other important requirements, including redundancy management, coherence of replicated data, and availability. These characteristics must be taken into account in the design of e-cash schemes, even in the case of purely online system. In particular, service availability is an important requirement for online retail customers. In effect, system availability and the possibility for the system to deliver key functionalities under failure must be considered during the design of the protocols. Failure of payment infrastructure is a catastrophic scenario for merchants, in particular, when large volumes of transactions are involved. Despite the security benefits they provide (e.g., real-time double-spending detection), online schemes involve important risks of failures. An hybrid scheme providing a fail-safe mode would more likely provide better fault tolerance. According to Anderson *et al.* [2], 99.99% availability is expected for current online payment systems. Failing to achieve such level of quality mean customer dissatisfaction and loss of business for merchants and banks.

3.2 Distributed Architecture Styles

As mentioned earlier an e-cash scheme may involve separate players and entities, who may interact or operate independently, impacting as a matter of fact, positively or negatively, system distribution. These include the payer, the payee, a network of banks, and an authentication infrastructure. Since typically most of these players will naturally be distributed geographically, it is essential to optimize the size and number of messages exchanged in their interactions so as to improve the quality of service delivered by the system and to minimize the overall cost of deploying, using and maintaining the system.

The impact on system distribution will depend on the properties expected from the system and the architecture style adopted in achieving them. According to the level of reliance on particular player in operating the system, the architecture may evolve from a fully centralized model to various strains of distributed models.

Most of the e-cash schemes proposed in the literature assume a basic e-cash architecture, which implicitly is a centralized architecture, built around the bank. However, some of the proposals are based explicitly on dedicated architectures which actually play essential role in implementing some of the required properties. We have identified in the literature seven different categories of e-cash architectures as follows: *Basic Online Scheme, Basic Offline Scheme, Basic Transferable Scheme, Peer-to-Peer Scheme, Distributed Banking Scheme, Randomized Scheme* and *Agent-based Scheme*, respectively.

In *basic online* e-cash schemes, the bank represents the main point of focus or of interactions, which tends to create around it a performance bottleneck and single point of failure. In this case, the key to system distribution would be to remove such bottleneck by dispatching strategically some of the load between other players. *Basic offline* and *basic transferable* schemes are earlier attempts to address this issue by limiting the role of the bank and integrating more flexibility and autonomy in the e-cash architectures. But these open up more security challenges. *Peer-to-Peer Scheme, Distributed Banking Scheme, Randomized Scheme* and *Agent-based Scheme*, are more sophisticated architectural proposals which try to strike the right balance between security and distribution requirements. In all these cases optimizing the number of interactions between the bank and other participants often represents the main source of difficulties.

In most proposed e-cash schemes, the bank is viewed as an abstract entity, without further discussion or consideration of the complexity underlying such notion. The bank as a player is actually a complex distributed system on its own with its own intricacies. The bank is the abstraction for interconnected financial institutions that cooperate in enabling and clearing payment transactions. Typically there are three types of financial institutions involved in E-Cash transactions, namely the *issuer*, which is the bank issuing the coins and possibly hosting the consumer's account, the *acquirer* which is the bank hosting the account of the merchant, and a *clearing house.* The acquiring bank sends received coins to the clearing house to process them and clear the payment. This will involve checking the validity of the coins with the issuing bank, and if successful, performing corresponding funds transfer between issuer and acquirer, and updating the database of spent coins. The above model can further be decentralized by distributing the responsibilities towards surrogates referred to as brokers, or further be complicated by including a trusted third party under the form of a central bank that monitors the operations of the financial institutions.

As mentioned above, in many proposals, protection against double-spending involves maintaining a spent coins database at the bank [54]. This raises some distribution challenges especially in the light of the above discussion, including location, scalability, performance and security. One of the first issues with such database is about its size, which will grow with the time as more transactions take place and pose as a result storage and performance problems. Alternative solution might consist of associating with the coins expiration dates, which however could raise maintenance issues. For the location, decision should be made about whether the database would be deployed at a single location, for instance at the clearing house, or whether it should be distributed, for instance between participating banks. In the two following sections (4 and 5), we describe in more details e-cash architectural styles and illustrate them through sample works from the literature.

4. Basic Architectures

A lot of research works have been produced on electronic cash schemes. We review sample of these works in each of the architectural categories identified in the previous section by focusing on underlying distribution challenges and criteria. In this section we will focus on basic schemes while in the subsequent section the focus will shift to current systems in the advanced architecture category.

4.1 Basic Online Schemes

The basic online architecture style can be seen as a kind of default architecture, which has been used systematically in many works to introduce e-cash mechanisms and protocols; samples of such works include [4, 11, 15, 19, 32, 34, 35, 47, 49, 55, 60, 61].

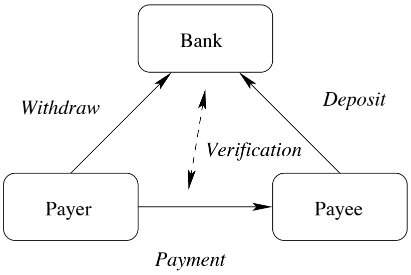

Figure 1. Basic Online Model

Basic online architecture typically involves three kinds of roles: *Consumer*, *Merchant*, and the *Bank* (see Figure 1). In this configuration, the consumer is the *payer*, who owns some coins and use them to purchase some goods or services. The *merchant* is the *payee* who receives coins in exchange of service provided. The *bank* stands for a financial institution that issues and sells the coins to consumers, provides support for checking the validity and integrity of the coins, and redeems the coins when requested by merchants.

The e-cash proposed in most current basic online schemes can generically be represented as a pair (s, s'), where s is a key that is kept secret by the owner of the cash and s' is a unique number that can be made public and is often referred to as a serial number. Typically s' is derived from s using a one-way hash function h. For a typical withdrawal, the consumer generates (s, s') and sends s' along with the amount of the coin and her account information to the bank. The bank checks the identity of the consumer and the balance of the account. If there is sufficient fund, the bank will debit the account and update its coins database by listing s' as reference for a valid coin.

For payment, the customer will send the coin (s, s') to the merchant who will check the validity of the coin by sending it to the bank. In principle, the merchant will also try to take ownership of the e-cash by selecting and sending a new serial number to the bank to replace the current one. The merchant will generate (s_1, s'_1), keep s_1, and send (s, s', s'_1) to the bank. The bank will check firstly that the consumer is really the owner of the coin through a hash operation $(h(s) = s')$, and secondly that the coin exists and has not already been spent by checking the serial number s'. If this is successful, the bank will replace s' by s'_1. This allows avoiding double-spending; in case where the customer attempts to reuse the e-cash at a different location, the money will be rejected because the password will be invalid. Taking ownership of the coin that way ensures the recipient of the coin (i.e. merchant) that its validity is guaranteed by the bank and that she is the sole owner of it.

Most of the basic online schemes proposed in the literature tend to focus on achieving a limited subset of the required e-cash properties. For instance, while

Medvinsky and Neuman's NetCash [34] supports conditional payer anonymity, the NetBill scheme proposed by Sirbu and Tygar [46] provides no payer anonymity. Furthermore while the cash checks scheme proposed by Chaum [11] supports unconditional payer anonymity, the high computational cost involved make it unsuitable for micropayments. This limitation is addressed by Deng *et al* who proposed an improvement of Chaum's cash check by using blind signatures to achieve security and anonymity, and providing support for micropayment by limiting to one the number of cash check used for multiple payments [15].

The basic online architecture is inherently a centralized architecture, where most of the communications go through the bank. The continuous involvement of the bank makes it a performance bottleneck and a single point of failure, which are important weaknesses in a distributed setting. It presents an advantage from security perspective, since it allows real-time double-spending detection. However, achieving payment untraceability with the basic architecture can be very challenging.

In some proposals, the basic architecture is extended by introducing a fourth role played by a trusted third party such as a *central bank*. In these cases, some of the responsibilities of the bank are transferred to the central bank such as issuing or publishing the coins. Sometime, such distribution of responsibilities not only improve system efficiency by reducing the load on the bank but it also allows implementing some required properties such as *anonymity* or *fairness*.

For instance, in [55], Wuu et al. propose an online payment system that involves a new role called the *Issuer* in addition to the three traditional ones of *Consumer, Merchant*, and *Bank*. The issuer is a trusted third party that takes over the role of checking the validity of the coins which, in most schemes is performed by the bank. They claim that this allows addressing the main drawbacks of online payment systems, namely having the bank as a bottleneck and single point of failure. The payment procedure followed in this scheme is similar to the one outlined above, with the difference that in this case coins are issued by separate entity other than the bank. Typically the consumer generates a secret proof and a public serial number, and then sends a withdrawal request to the bank. The bank authenticates the consumer, checks the balance of her account and asks the issuer to register the serial number of the coin.

For payment, the merchant generates secret proof to be kept locally and public serial number to be sent to the consumer, who then forwards to the issuer this information along with the secret proof and serial number of the coin. The issuer checks the validity of the coin and then creates a new coin by converting the old one. To make a deposit, the merchant sends proof of his identity, the serial number of the coin and proof of ownership to the bank. The bank checks the validity of the coin with the issuer, and then credits the merchant's account and asks the issuer to perform the necessary updates. During the withdrawal process, the bank receives the identity of the consumer and the withdrawal information. The withdrawal information, consisting of the coin serial number and a symmetric key, is encrypted using the issuer public key. After authenticating the consumer and checking his account balance, the bank forwards the (encrypted) payment information to the issuer to publish the requested coins. Note that at this stage, the bank knows the consumer but has no knowledge of the correspondence between the serial number of the coins and the consumer; such information is transmitted to the issuer (via the bank) encrypted using the issuer's public key. The issuer has such knowledge, and it knows the bank but does not know the consumer. So the consumer can spend the coin without neither the bank nor the issuer being able to trace corresponding transactions back to her.

According to the authors, the proposed system provides security (i.e. unforgeability, no framing, and double spending prevention and over-spending prevention), fairness, transferability, and anonymity. But collusion between the bank and the issuer, a possibility not to be ruled out, could reveal the consumer's identity. The same can be said for the fairness property which relies on the willingness of participants like the bank or issuer to abide by the protocol. Furthermore, although the proposed scheme allows reusing coins in several payments before redeeming them, the involvement of the bank in these operations make questionable the transferability claim. Wuu *et al.* suggest alleviating the limitations of online payment (i.e., single point of failure and performance bottleneck) by distributing the overheads of coin verification over several *Issuers*.

Overall the *basic online scheme* presents several practical advantages when it comes to implementing key security features such as real-time double spending prevention. This comes however with a loss in efficiency and scalability due to the central role played by the bank. Alternatives may consist, for instance, of distributing the role played by the bank between several players or simply of removing or reducing the central position played by the bank. In subsequent sections we will discuss how this issue is addressed through the remaining architectural styles identified earlier in this survey.

4.2 Basic Offline Schemes

Basic offline schemes stem from an attempt to address some of the weaknesses highlighted previously in *basic online schemes* by including more flexibility and autonomy in e-cash transactions. As shown by Figure 2, basic offline schemes are also based on a three-party model involving the same roles as the previous model. The main difference between both models is that payment transactions take place only between the payee and payer, and do not involve the bank. So the bank does not represent anymore a performance bottleneck and single point of failure as in the basic online architecture. However, double-spending can be detected only after the fact, which can have some damaging consequences.

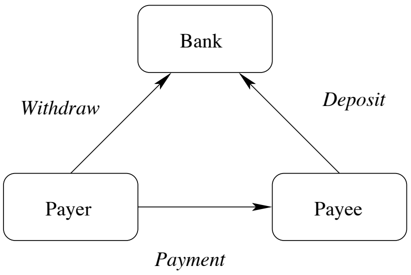

Figure 2. Basic Offline Model

Basic offline schemes have been the purpose of intense research in the last decades. Examples of basic offline payment schemes proposed in the literature include [5–7, 12, 17, 18, 20, 38, 38, 43, 51, 52]. Since the increased flexibility and autonomy of basic offline schemes come at the expense of security (i.e. double spending detection), the focus of these works has mostly been on developing strong security mechanisms that would address this limitation. Unfortunately, in general, these security mechanisms have proven to be heavy and complex, offsetting as a result the performance and scalability gains achieved inherently by the proposed schemes.

For instance, one of the earliest untraceable offline scheme, proposed by Chaum *et al.*, falls in this category [12]. The proposed scheme is based on the *cut-and-choose* methodology, collision free one-way functions, and blind signature scheme.

Blind signature scheme was introduced initially by Chaum to ensure consumer anonymity during electronic payments [10]. The scheme allows the sender of a message to obtain a signature without the signer knowing anything about the content of the message. As such it allows perfect unlinkability, meaning that it is impossible for any party other than the sender to link a message-signature pair to the signer. In digital payments systems, the signer is the bank. By signing blindly issued coins, the bank has no way to trace how the consumer will spend them; this allows the consumer to remain anonymous.

With the offline e-cash payment method proposed by Chaum *et al.* in [12], the consumer initially creates a coin by generating and blinding a message of n pairs of numbers such that the occurrence probability is extremely low; n must be large enough that an event with probability 2^{-n} is less likely to happen in practice. As a result, matching pairs of such numbers can be used to (quasi) uniquely identify the consumer. After generating the message, the consumer must blind it and send it to the bank for signature. The bank will simply check that the message has the required format and properties, and then sign and send it back to the consumer. When the consumer wants to make a payment, the merchant will send her a challenge message consisting of a random sequence of n bits, each corresponding to the position of a piece in one of the pairs of numbers involved in the coin. In return, the consumer must respond to the challenge by sending the appropriate pieces for corresponding pairs. The merchant will store this response, and later to deposit the coin, she will simply have to send the sequence of numbers to the bank along with the coin. In case of double spending the bank will receive two different sequences of numbers for the same coin; by combining them the bank will be able to establish the fraud and link it to the fraudster. Note that this is possible because a random string of bits is used as challenge. Key limitations of the above scheme are that despite its conceptual simplicity, it is inefficient because each coin must carry $2n$ large numbers, which limit scalability and performance. Furthermore the blindness of the signature scheme used by the bank makes it possible for the consumer to trick the bank into signing, for instance, an amount that could be different from what was agreed upon. The method proposed by Ferguson addresses the latter limitation by using randomized blind signature, but this comes with an increased level of complexity [17].

Brand's proposal [5] improves on the above schemes by using zero-knowledge proofs based on the Schnorr protocols instead of cut-and-choose for user identification. Zero-knowledge proof is any cryptographic protocol that allows proving knowledge of some secret information without revealing such information.

The main advantage of the above basic offline schemes is that there is a real distribution of the tasks among participants. Each participant plays an equal role and there is no central component where all or most the communications have to transit through as it is the case in the basic online model. However, the number and size of the messages exchanged between participants could impact negatively on scalability and performance.

To address the above concerns, in [43], Rivest and Shamir study two micropayment systems, namely "PayWord" and "MicroMint" that minimize the amount of public operations required per transaction by using hash operations instead of public key ones, cutting down dramatically underlying performance costs. For micropayment transactions, it is important to keep underlying costs as low as possible to avoid situations where mechanisms costs outweigh payment values. Three roles are considered in the proposed micropayment system, namely users, vendors and brokers.

"PayWord" is based on a chain of hash values referred to as "paywords"; similar scheme is being used to implement one-time password [28] and one-time signature [36]. For the initial payment, the user authenticates to the vendor the complete chain using a public key operation, and then pays by presenting a payword to the vendor. Subsequent payments are made by unraveling the paywords in the chain. With "PayWord", the user opens an account with the Broker who issues a digitally-signed certificate for her; the certificate must be renewed on a regular basis by the broker. The certificate can be viewed as a guarantee for vendors that the holder is allowed to perform PayWord transactions, and corresponding paywords are redeemable by the broker. Since, the primary goal of "PayWord" is to reduce the amount of communications involved in payment transactions, to avoid the broker becoming a bottleneck, computations are performed offline. So "PayWord" is fundamentally an offline system. As such the broker does not have to be online when users and vendors are interacting. Although it is obvious that "PayWord" does not provide user anonymity, it may be possible to achieve anonymity at the vendor side. The vendor simply needs to obtain the public key of the broker to check the user's certificate, and establish a way for the broker to pay redeemed paywords. Assuming that these operations can be achieved anonymously, the anonymity of the vendor can be ensured.

MicroMint' trades security versus efficiency; it eliminates the public key operation altogether, increasing speed at the expense of security. It generates coins using k-way hash-function collisions. Payments are based on coins issued by the broker and sold to users. The user gives the coins to the vendor in exchange of a service; to redeem the coins, the vendor returns them to the broker. MicroMint coins are bit strings easy to check but difficult to produce.

Micropayments are also the target of the scheme proposed in [38]. The authors present an offline scheme that supports unconditional client anonymity, partial untraceability, and protection against double-spending, coin forgery and framing, while minimizing the amount of computation involved. The authors highlight the complexity and heavy computations involved in existing payment systems, and emphasize the need for low cost environments for electronic wallets that would support micro payments. It is argued here that challenge-response initiated by the merchant to the consumer should be avoided as much as possible because of their high computation cost. In this context, the consumer will simply withdraw a series of coins from the bank that would be connected through a hash function. The proposed solution, however, is biased because the consumer must answer to a challenge to sign the first coin in the series but not the remaining ones provided that he would have to spend all them with the same merchant.

Overall, due to the decentralization and autonomy of the components involved, basic offline schemes are inherently more scalable than basic online schemes. But this comes with a price tag of reduced security. To address the security limitation, there is an attempt to deploy heavy security infrastructures which tend to cancel out the performance gain. So in distributed processing where both security and efficiency are equally important, we need to strike the right balance by making appropriate trade-offs when designing and deploying basic offline schemes.

4.3 Basic Transferable Schemes

The basic architectures outlined previously are preliminary and necessary steps towards establishing e-cash as universal currency. However, they still remain too close to existing payment card architectures, with the single improvement being the independence from hardware. To reach the status of true universal currency, achieving transferability is mandatory. This should allow extending the use of electronic payment schemes to private contexts and to exchanges between individuals.

Examples of transferable schemes proposed in the literature include [39, 40, 48, 50]. Transferability is one of the few e-cash properties which have direct impact on system distribution, because by definition it requires payment transactions to be carried out between peers without any involvement of the bank [42]. In effect, traditionally transferability is defined as the capacity to reuse received coins in other payments without involving the bank. Note that (with respect to this definition) several of the basic offline schemes introduced above such as [5, 12, 17, 38, 43] do not support transferability; in these cases the only option for a payee is to redeem a coin after receiving it.

As illustrated by Figure 3, basic transferable architecture involves at least four kinds of participants: *Initial Payer*, *Final Payee*, *Payment Intermediary*, and *Bank*. *Initial Payer*, *Final Payee*, and *Bank* play the same role as *Payer, Payee* and *Bank*, respectively, in the previous schemes. *Payment Intermediary* is an agent who will play the role of *Payee* or *Payer* according to the type of transactions.

For instance, one of the earliest transferable schemes, proposed in [50], involves a bank B and n individuals $\{C_i \mid 1 \leq i \leq n\}$ who may play the roles of consumers or merchants. C_1 withdraws a coin from the bank B and purchases some item with C_2 by transferring the coin to it; C_2 later reuses the same coin in a purchase with C_3; the same process is repeated several times going from C_i to C_{i+1}. Finally after receiving the coin, the last individual in the chain, C_n will deposit it at the bank. Note that all the intermediary steps in this sequence of transactions take place without contacting the bank. Although this approach offers suitable ground for autonomous and distributed processing, it can pose significant scalability and performance issues.

For instance, in [13], the proposed solution requires recording transaction information in coins, which contributes to growing coins sizes as more transactions occur; such information is needed by the bank for double spending detection. Moreover, in this case it is necessary to protect the confidentiality and the integrity of the information carried by the coins; only authorized officials or parties should have access to it.

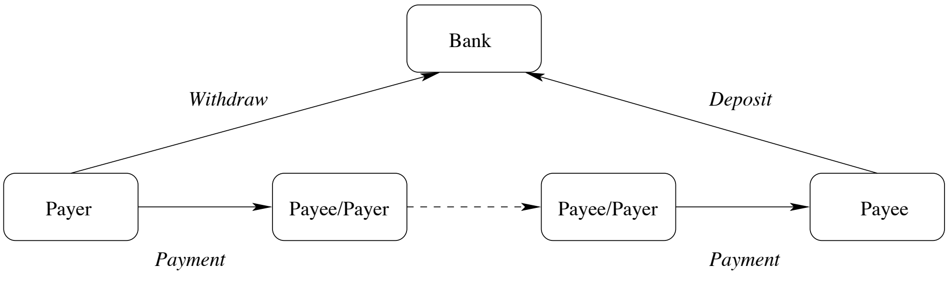

Figure 3. Basic Transferable Model

The main challenge in designing a transferable scheme is double-spending detection; since transferable schemes work inherently offline, double-spending detection happens after the fact. So unless there is a limitation on the number of allowed transfers, the cost of fraudulent transactions (discovered after the fact) can be huge. Thus, transferable schemes need some traceability mechanisms to identify fraudsters, and as such cannot ensure at the same time full anonymity and security. For instance, in [48], an offline and transferable e-cash system based on split secret scheme is proposed. The proposal is interesting in many respects because it highlights most of the problems faced by transferable e-cash schemes. To deal with traceability, the proposed cash model contains two parts: a fix component, signed by the issuer, based on traditional e-cash format, and a variable component, signed during the transaction that records transaction information to ensure traceability. The variable component allows protecting against double spending: the identity of each participant is recorded in the coin, in a list of transactions. Each item in the transaction list consists of a fixed number of pairs resulting from a split secret based on the identity of the participant. To ensure confidentiality and detect possible double spending, the list is randomly blinded in such way that future participants cannot know the identity of previous participants. Furthermore in case of double spending the probability that the same items were randomly blinded can be low: with one pair per transaction item, the fraud detection probability is only 50%, while 6 pairs are required to reach a fraud detection probability of 98.5%.

This way of detecting double spending is very interesting, but it requires a trusted third party referred to as Point of Sale (POS) device in the model that is in charge of blinding and signing the second part of the coin. Thus, the model may be categorized as working offline in the sense that the presence of the bank is not required, but it still needs the presence of an external party that is neither the payer nor the payee.

To protect against double-spending, the bank maintains a database of spent coins. If a coin exists in the database, then the bank will check the entries of the transaction list and detect the first different entry. The underlying idea is that for each entry, one of the two parts of corresponding identity has been masked randomly and the probability for the same parts to be masked is very low. In this case, finding an entry where both parts are different will allow reconstructing the identity of the fraudster. The POS is also responsible for checking the validity of the coin and for checking the reliability of the consumer based on a blacklist maintained and provided by the bank.

There are two main drawbacks to this model. Firstly, the offline characterization of the scheme is debatable considering that the involvement of the POS is necessary; one can simply argue that the POS is simply another form of bank. Furthermore to ensure the reliability of the system, it is necessary for the POS to connect to the bank on a regular basis in order to obtain updates to the blacklists. Most important is the problem of the size of the coins and of the database maintained by the bank to keep track of all the redeemed coins. Using some validity date allows reducing the size of the database maintained by the bank. Moreover, the size of the transaction list is bounded: when the validity of the coin is about to expire or when the transaction list is full, the coin must be returned to the bank which will check that it is valid and then redeem it. Thus, although transferability is achieved with the proposed scheme, the number of possible transfers is bounded by the size of the transaction list. The size of the data circulating is significant and the operation of the POS is constraining; as a result the proposed scheme is not as light as claimed by the authors. The system does not provide anonymity, but claim to ensure pseudonymity. However, if the coin is used for only a single payment, the bank will be able to trace back the transaction.

Transferability is an essential property for e-cash to reach the status of realistic cash scheme. However, in this case protecting against double spending can be achieved only by trading-off anonymity, and furthermore underlying scalability and performance challenges can quickly become daunting. Involving a trusted third party may alleviate some of these challenges, but will represent in itself an extremely constraining solution. The advanced architecture styles attempt to address these issues as discussed in the next section.

5. Advanced Architectures

In our classification, four architectural styles fall in the advanced architectures categories, namely, *distributed banking, peer-to-peer*, *randomized* and *agent-based* architectures. We review in this section the main characteristics of these architectures.

5.1 Distributed Banking Schemes

From the above description of basic online schemes, it appears that the bank is the main choking point. We have seen that offline payment represents a way to remove the bottleneck created by the bank. Alternative approach consists of distributing banking responsibilities. In the literature, distributed banking has been approached from two different perspectives. The first perspective advocated by Lysyanskaya and Ramzan takes into account the diversity of banking institutions actually involved in payment transactions and propose a unifying scheme which improves scalability [33]. The second perspective elaborated by Hoepman and Jacobs consists of spreading the responsibilities of the bank between several other participants, reducing as a result the load on the bank [24].

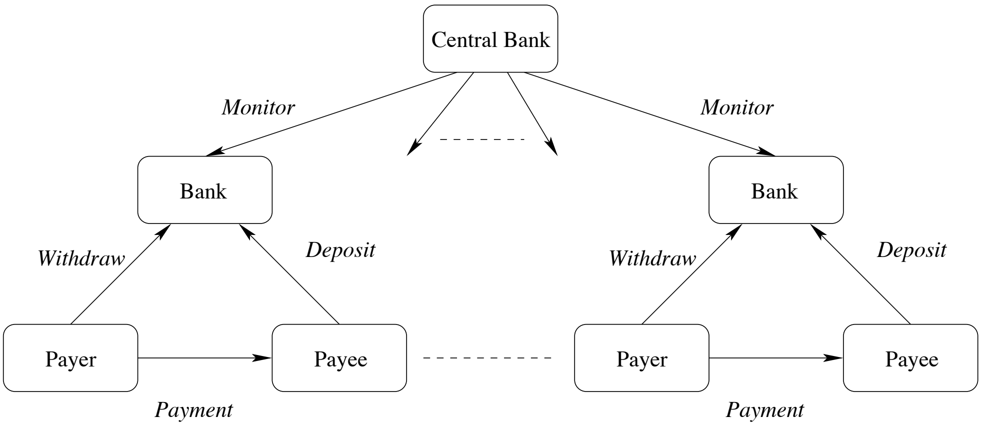

Figure 4. Distributed Electronic Banking Model

The distributed banking model proposed by Lysyanskaya and Ramzan stems from the observation that electronic payment systems are currently being developed and operated separately and independently by banks; each bank maintains its own payment gateway and authentication infrastructure. This is not the best way to optimize resource usage and tends to complicate clearing activities required in reconciling transactions performed at different banks. Lysyanskaya and Ramzan tackled this issue in [33] by modifying the basic three-party model (introduced above) taking into consideration the fact that transactions may be carried out through a large group of banks monitored by a central bank (see Figure 4). They propose an extension of Camenisch and Stadlers' group signature scheme [8] by integrating the notion of blindness, leading to the so-called group blind signature scheme.

Groups signatures schemes were initially introduced by Chaum and van Heyst [14]. Group signatures allow a member of a group to sign on behalf of the rest of the group without revealing the identity of the signer, and making it impossible to link two different signatures issued by the same group member. However, in case of dispute, the scheme involves a designated group member who can determine the signer of the document. A powerful feature of group signatures is that signature verification can be performed using a single group public key. Unfortunately, with the initial schemes, the size of the public key tends to grow with the size of the group, which is unacceptable in situations where scalability is required. The later proposal of Camenisch and Stadler [8] in which the group public key remains independent of the group size addresses such limitation.

Using the proposed group blind signature, Lysyanskaya and Ramzan devise a scheme allowing a group of banks to distribute anonymous and untraceable e-cash, while concealing the identity of the issuing bank. The proposed scheme is particular in not only the fact that it allows multiple banks to distribute the e-cash, but it also allows concealing the identities of both the consumer and its bank. Four roles are considered in the proposed scheme: the consumer, the merchant, the banks which form a group, and the group manager which can be, for instance, the central bank. To purchase a coin, the consumer first generates the coin and sends it to her bank for signature. The bank withdraws the coin's value from the consumer's account, signs blindly the coin and sends it to the consumer. To avoid blindly signing something other than what was agreed upon, the authors suggest that the bank can use different secret signing keys for different coins values. For payment purpose, the consumer gives the signed coin to the vendor, who can check the validity of the coin using the group public key. To redeem the coin, the vendor deposits it at its bank, which in its turn checks the validity of the coin and accordingly update the vendor's account and the list of coins already spent (to avoid double spending). A key limitation of the proposed scheme is that it is an online scheme. To make the scheme offline, the authors propose a modification that unfortunately reduces anonymity. In the modified scheme, consumers form a group as well, the manager of which is a trusted third party referred to as passive trustee. After purchasing a coin as usual, the consumer applies the spender group signature to it prior to using it for payment. The vendor checks the validity of the coins as usual, and in case of conflict, the trusted party is asked to establish the identity of the faulty consumer. Unlike in the online scheme where the identity of the spender was fully concealed, in the proposed offline scheme, there is a compromise in spender anonymity. It is important to highlight that transferability is not addressed at all in the proposed scheme.

Xu and Zhao proposed in [56] a distributed electronic payment model based on the notion of bank union, which actually fully mirrors the distributed banking model proposed in [33]. A bank union consists of a group of banks each with the ability to issue e-cash, and whose transactions are monitored or regulated by a central bank. The payment protocol of the bank union is based on a group blind signature scheme. The main contribution of Xu and Zhao is to have implemented and tested a distributed electronic payment gateway based on this model.

The distributed banking model proposed by Hoepman and Jacobs consists of transferring some of the responsibilities of the bank to other players [24]. Hoepman investigated the issue of distributed double spending prevention in an on-line decentralized environment without a central bank through which all requests for verification must go. They suggest that efficient randomization techniques can be used in such context to prevent a coin from being spent multiple times. The approach consists basically of distributing the function of the (central) bank over a subset of the nodes in the system, referred to as *clerk set*. The main task of the *clerk sets* is to check the validity of the coins during payment transactions. The selection of clerk sets can be done deterministically or randomly, depending on the recipient (i.e. merchant) and the characteristics of the coin. It is shown that by selecting appropriately clerk sets of size above specific bounds double spending can be prevented either deterministically or with strong probability, according to whether the selection is deterministic or random. To verify the validity of a coin, a clerk set maintains the history of coins, which may grow without bounds. Several simplifying assumptions have been made in this proposal, including the consideration that the network is static and that coins are transmitted through atomic operations. These actually underscore some of the open challenges that need to be addressed by the proposed distributed banking scheme.

Distributed banking model attempts to remove the performance bottleneck created by the bank by redistributing and optimizing the tasks performed. It achieves better scalability compared to the basic online model, while maintaining the same level of security.

5.2 Peer-to-peer Schemes

The *peer-to-peer* (P2P) model is one of the instances of architectural schemes attempting to move away from the basic centralized model. This represents a significant progress in terms of system distribution as it introduces several interesting features for decentralizing e-cash payment architectures. Likewise, the P2P model seems a natural fit for capturing communications between individuals without intervention of the bank.

As shown by Figure 5, the P2P model transfers the responsibilities of the bank to *brokers* and empowers consumers and merchants who are treated as peers. Peers can purchase coins and transfer ownerships to other peers through payments, taking place directly or sometimes through the broker without involving the bank. So in the P2P model the bank becomes a remote entity which plays very limited role compared to the basic three-role model; this creates suitable opportunities for distributing effectively the load.

Redistributing the load of the bank among peers contributes to greatly improving overall system performance. In this regard, the *Peer-to-Peer* architecture is fundamentally different from the *distributed banking model* presented previously, in which the main target of the decentralization is the bank, which shifts from an abstract centralized entity to a concrete network of financial institutions.

PPay is an example of micropayment scheme for peer-to-peer environments that provides security, fairness, and scalability, but no anonymity [59]. PPay makes a distinction between the *owner* of a coin and its *holder*. Initially the owner purchases the coin with a *broker*, and uses it for a payment by transferring it to another user who becomes the new *holder*. Subsequent transfers from a holder to another holder are made via the owner, who remains as such for the lifetime of the coin. A coin is represented as $C_H = Sign_O(Sign_B(O, s_n), H, seq))$, where O is the owner of the coin, H is its current holder, s_n is a unique serial number associated with the coin, and seq is a sequence number. [1] The sequence number is maintained and incremented by the coin's owner every time it is issued or transferred. For coin holder H_1 to transfer a coin $C_{H1} = Sign_O(Sign_B(O, s_n), H, seq))$ to H_2, she will send a request with the identity of H_1 to owner O signed with her private key; O will do the transfer and sends coin $C_{H2} = Sign_O(Sign_B(O, s_n), H, seq))$ to H_2. So clearly, PPay does not provide any anonymity since the identities of parties are encoded in the coins, but it allows avoiding double spending performed by participants other than the owner, and allows detecting (and punishing after the

[1] Notations $Sign_0$ or $Sign_K$ are used when some message is signed by entity O or using key K, respectively.

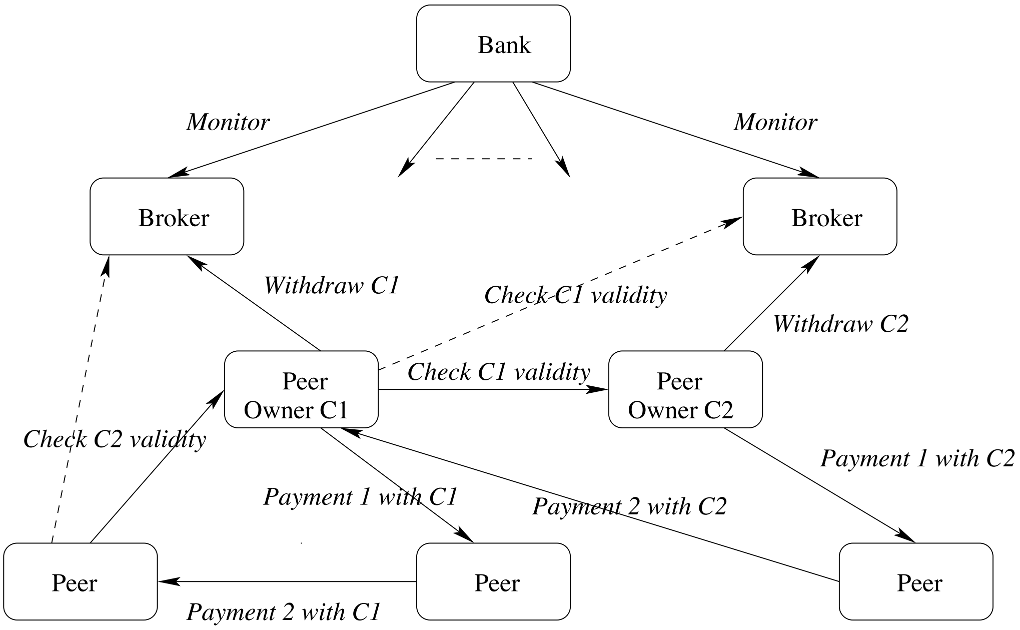

Figure 5. P2P Model

fact) double spending performed by the owner. The advantage of such scheme is the distribution between the different owners of the payment verification function which is entirely supported by the bank in previous models. Major disadvantage of the scheme is that using the coins requires that owners (always) be online.

Wei et al. introduce *WhoPay* [54], a scalable and anonymous extension of *PPay*. It is claimed that *WhoPay* provides security, anonymity, fairness, transferability, in addition to scalability. *WhoPay* involves in addition to the three roles mentioned above, a fourth role named the *Judge* who is a trusted third party that plays the role of group manager for users. *WhoPay* uses group signatures to ensure fairness; every user is required to register with the group manager. In contrast with *PPay*, with *WhoPay* coins are represented with public keys instead of serial numbers, but like *PPay* coins transfer load is distributed across peers to ensure scalability, and coins follow the same lifecycle in both schemes. To obtain a coin, a user H generates a pair of public and private key (pk_H, sk_H), keeps secret sk_H and sends pk_H to the coin's owner O; the subtlety here is that the public key is sent without any explicit identification of its owner. The transfer of the coin will follow the same process as with PPay and the coin will have several of the same fields; the transferred coin will be $C_H = Sign_{sk_O}(Sign_B(O, pk_O), pk_H, seq, exp_{date}))$, where exp_{date} is the expiration date for the coin; coins must be renewed before or by the expiration date to keep their value. The basic scheme proposed for *WhoPay* does not achieve full anonymity; while coin holdership is hidden, coin ownership is exposed. To address this limitation, the authors suggest, among other solutions, to remove the identity of the owner from the coin, and put the onus on the owner of a private key to prove her ownership of a coin.

Also, the basic scheme does not support double spending prevention for coins' owners (although as mentioned above, it allows avoiding double spending performed by participants other than the owner). Double spending can be detected through the group signature scheme, but only after the fact, which as argued earlier can be costly. To address this limitation, the *WhoPay* model provides real-time double spending detection by implementing a publicly viewable list of valid coins. This list can be read and updated by coins owners, and only be viewed by other peers. The authors suggest implementing the coins list as an access-controlled distributed hash table (DHT). One of the key issues arising with such approach is the huge level of trust placed in a single entity, which could have some adverse consequences for the security of the entire system.

But most importantly, both *PPay* and *WhoPay* are essentially online payments schemes. Both of them use a downtime protocol to handle cases where the owner of a coin is not available online. In this case, the owner is replaced by the broker, which temporarily processes transfer requests, and later synchronizes state with corresponding owners when they become available online. This is simply a variant of an online scheme, where the bank plays the role of the broker. Offline payments where peers can exchange coins among themselves without involving any external entity like a broker or a bank are not supported by *PPay*.

Osipkov *et al.* introduces a software-based e-cash scheme that uses a cooperative peer-to-peer architecture to combat double-spending in real-time without requiring an online trusted third party [41]. The proposed scheme follows the guiding principles of the above P2P models by considering merchants as special participants, not similar to other participants. It is assumed here that due to business necessity, merchants are (bound to be) always online, which allows real-time double-spending detection. The proposed framework targets ”mini-payments” (payments small enough while keeping underlying costs in profitable ranges) and is based on the consideration that double-spending should not be prosecuted, allowing as a result fully anonymous and untraceable e-cash. The proposed architecture involves three kinds of participants, namely *broker*, *merchant* and *consumer*. The broker could be online or offline, and may either play the role of the bank or serves as intermediary between the bank and the other participants. A broker deals with a mini-payment network, in which participating merchants play an active role in establishing and checking coins validity. Consumers purchase coins with the Broker; each purchased coin is assigned at that time to a merchant selected randomly and referred to as *witness* who is responsible for certifying the coin in future payments.

For payment, the consumer transmits a coin to the merchant, who then forwards it to the assigned witness for certification. The witness will certify the coin if she considers it valid, and sends it to the merchant, who can redeem it at the broker at any time. Otherwise, if she has already seen a previous instance of the same coin, she will extract some secret information from both instances, and sends these to the merchant as basis to reject the payment.

To deal with cases where some of the merchants might go offline, it is suggested to use *k-out-of-n* schemes ($k \leq n$), where k out of n assigned witnesses would be required to certify a coin before completing a payment. Such extension could also allow achieving fault tolerance and distributing load based on geographic considerations.

The proposed frameworks by distributing verification or double-spending detection functionality between merchants, lighten significantly the load on the bank (or the broker), which is no more a performance bottleneck or single point of failure. This represents significant advance towards the design of e-cash models that supports distributed processing and scalability.

5.3 Randomized Schemes

Previous discussions have highlighted the importance of *connectivity* in e-cash system distribution. *Connectivity* refers to the level of interaction between the system and the bank during the lifetime of a coin. With e-cash systems, we consider three classes of connectivity: online, offline, and hybrid schemes. Hybrid schemes, in their turn, can be subdivided into two subclasses: systems that can be used invariably as (pure) online or (pure) offline schemes (i.e. support both kinds of connectivity), and systems that provide a middle ground between both approaches by conducting randomized audits or verifications. Randomized Schemes are based on probabilistic checking, which represents a middle ground between online verification where every payment transaction is checked by the bank in real-time and offline verification where transactions are checked after the fact. These schemes allow by checking probabilistically payment transactions to optimize the high cost of online verification while alleviating the risk and cost of double-spending underlying offline schemes. In doing that they provide the basis for achieving an improved level of performance and scalability that is lacking in online schemes while partially addressing some of the security issues plaguing offline environments. From system distribution perspectives the latter kind of systems are more meaningful because they explore the continuum between online and offline systems in terms of volume of interactions with the bank [30].

Some of the few papers that have covered randomized audits in the literature include references [21, 24, 27, 57, 58]. The work of Hoepman on distributed double spending prevention [24], which is already discussed in section 5.1 under distributed banking schemes, also falls under the category of randomized schemes. While in [21, 27, 57], double spending is prevented by checking probabilistically coin validity through a (central) bank, in [24] the same goal is achieved by always checking payment validity through a distributed set of banks selected randomly.

Unlike in [57] and [58], both [21] and [27] are pure software systems. In [57], Yacobi proposes a randomized scheme that combines hardware and software solutions with randomized audit. Specifically the proposed scheme uses smart-card id-based wallets storing coins signed by the bank. In [16], e-cash systems based on tamper resistant hardware were organized into two sub-categories referred to as c-wallet (or coin wallet) and b-wallet (or balance wallet), respectively. The former carries individual fixed value coins, while the latter maintains total of the collection of coins as a whole. Yacobi suggests that fraud emits signals, with different frequency that can be leveraged to detect fraudulent behaviors. According to Yacobi, although b-wallets are more space-efficient, they are inherently less effective in detecting fraudulent behavior compared with c-wallets [57]. Fraud detection with b-wallet consists of checking whether the wallet has spent beyond its balance based on a small sample of transactions. This is more difficult to establish than showing that a coin was spent at least twice within a c-wallet, where by definition individual coins are stored and discarded each after payment. Considering that breaking a strong tamper-resistant hardware would require large investment in terms of cost and time, Yacobi investigated in [57] the role of randomized audit in connection with economically motivated adversarial payers. It is established that with randomized audit, there is a middle ground between fully online and fully offline schemes, where the attacker will break even with their investment in attempting to defraud the system. Naturally it is expected that beyond this point defrauding the system will not make sense from a business perspective to rationale adversaries. Specifically it is shown that when the attacker breaks even with their investment in defrauding the system, the probability not to detect her is $O(e^{v/r})$, assuming that v is the value of such investment and $1/r$ is the audit sampling rate. Furthermore, in [58], it is shown that there is an upper bound on theft when applying partial real-time audit of payment transactions for both c-wallet and b-wallet, assuming perfect conditions. However, while the upper bound decreases quadratically as the audit rate increases for c-wallet, it increases exponentially for b-wallet, particularly when a low false alarm rate is required. As a result it is suggested that while partial audit may be appropriate for c-wallets used for small transactions, it is too risky for b-wallet requiring very low false alarm rate.

Overall, randomized architectures approaches represent fertile ground for research into secure distributed e-cash payments. In fact, a perfectly secure system requiring permanent connectivity and systematic checking may not be suitable for real-life applications, where scalability, performance, and low (communications) costs are key concerns. Hybrid approaches seem then to be promising middle-ground between security and practical implementation.

5.4 Agent-based Schemes

The previous architectures, which cover most of the e-cash schemes proposed in the literature, are based on the assumption that there are always two parties, a consumer and a merchant who exactly know each other's location, and are willing to exchange goods and funds. However, a substantial amount of e-commerce transactions occur in settings involving a large number of uncoordinated and distributed parties with limited knowledge of available services. Peer-to-peer environment may allow such parties to discover and interact with prospective trading partners. But, in some cases, the high volume of interactions create communications bottlenecks and may have adverse impact on performance and resource availability for some of the participants. Agent-based schemes address these challenges by providing a mechanism, under the form of a mobile agent that performs remotely various tasks on behalf of the user, including searching, selecting, negotiating, and processing. Mobile agents limit the amount of interactions involved in peer-to-peer communications, and improve as a result the efficiency of distributed processing.

There are two main research directions around multi-agents setting and e-cash. The first direction concerns the creation of multi-agents e-commerce framework, including the notion of payment. For example, Guan and Hua propose a multi-agent mediated electronic payment architecture, which supports diverse electronic payment schemes [22, 23]. Their model decomposes the payment environment into autonomous payment clusters, where specialized agents collaborate to perform payment tasks. The proposed agent architecture is designed to be extensible and scalable. It is structured into the so-called SAFER mobile agent communities. SAFER, the acronym for Secure Agent Fabrication, Evolution and Roaming, is an agent framework designed to support and manage agents in e-commerce environments [62]. A SAFER community is an autonomous agent cluster that consists of various entities. Five different entities are involved in the electronic payment implementation namely the *Interconnected Financial Institutions (IFI), Payment Gateway, Trusted Third Party (TTP), Merchant Host* and *Agent Butler*. The Agent Butler is deployed on the customer host, and typically acts on behalf of the customer also referred to in SAFER as the *owner*. The Agent Butler receives requests from the owner and manages and dispatches mobile agents accordingly; as such the owner does not need to be always online since it can fully rely on the Agent Butler to perform required tasks. The IFI consists of the network of banks involved in the transactions, including the customer's bank that issues the cash, the merchant's bank, and a clearing house that handles inter-bank transactions. The payment gateway serves as front-end for the entities involved in the IFI. TTP is some neutral trusted certified host that handles or ensures trusted operations or services in specific areas or for specific purpose. For instance, the TTP could be some Certificate Authority (CA) that is responsible for delivering trusted digital certificates for the different entities involved in the agent community. In a SAFER community, agents are organized into a multi-layered structure referred to as "agency". Each "agency" represents a group or federation of agents with specific functionality or expertise; for instance agents specialized in information gathering would be part of an "Information Agency", while those providing payment or accounting services might be part of "Financing Agency". Agencies interact and cooperate under the supervision of Agent Butler to carry out the various tasks involved in the system operation. The main interest of this distributed architecture is that it takes into account not only the negotiation process, but also the payment procedures. In particular, it allows agents to choose automatically the best payment option, which is a necessary task in order to make such framework useful in real-life applications.

The second research direction concerns the possibility for an agent, in a multi-agent setting, to carry and spend e-cash. This is an important and difficult task in the sense that carrying digital cash exposes the agent to possible theft due to inherent security weaknesses. Such challenge must be addressed for agent-based e-cash scheme to represent a viable payment option. In order to reduce the amount of peer-to-peer interactions by allowing mobile agents to carry digital cash securely, Jakobsson and Juels propose a new e-cash scheme known as X-cash or executable digital cash, which ties the offer and payment in a common entity [25] . A piece of X-cash consists of a signed certificate issued by the bank and a program ω that generates the amount that the consumer is willing to pay for some goods or services. Initially consumer C will obtain a negotiable certificate from her bank authorizing her to make payments. She will decide the range of offers she would like to make for the goods and based on that the offer function ω will be constructed and encoded in a piece of executable code. The consumer will then generate the X-cash by combining the signed executable code and the certificate issued by the bank. The executable is signed using the private key sk_C corresponding to the public key pk_C contained in the signed certificate issued by the bank. To make some purchase, C will send the X-cash to the merchant M. The merchant will check the correctness of the signature and evaluate corresponding offer by executing ω. If satisfied, M will contact C's bank which will get in touch with M's bank and perform the payment clearing process. By allowing the offer to travel in a common entity with corresponding goods or payments, the proposed architecture allows digital cash to be used in highly distributed settings while ensuring the security of conveyed funds. Although the basic scheme proposed in [25] does not support anonymity, the authors claim that it is possible to extend it to address such property. However, this approach is not really a multi-agent one, in a sense that interactions are limited to a merchant and a consumer, or a merchant and a bank. CyberOrg, a model for hierarchical coordination of resource proposed in [26], also allows the creation of agents carrying e-cash. The proposed approach focuses more on the implementation of agents and their interactions rather than addressing security requirements of e-cash payment.

Researches on using multi-agents systems for e-cash payments are quite recent. A first explanation is that multi-agent setting creates an additional layer of difficulties on top of an already complex set of issues. In the future, we will have to address several difficulties in this setting. On one hand succeeding in using some artificial agents to negotiate and conduct payment transactions on user behalf may represent a considerable boost for e-cash technology, but on the other hand this may be the source of significant security challenges. As a result, suitable trade-offs must be made by taking into account these constraints, when designing multi-agents based e-cash architectures.

6. Summary and Discussions

Issues surrounding e-cash schemes have widely been discussed and studied in the literature. However, the approaches proposed so far have not yet reached the expected level of maturity and as a result the overall field continues to be an area of intense research. In effect, existing electronic payment systems are based on pre-paid schemes or on payment cards, which inherently are very centralized, non-transferable and in general non anonymous.

However, the benefits of e-cash either for consumers and merchants as simple and universal payment scheme, or for banks and countries as a cost-effective alternative to physical cash, are widely recognized.

Other than the possible psychological barriers due to any changes in the usual way of doing business, it is clear from the above literature review that e-cash faces two main kinds of challenges: the need to prove absolute security of the scheme and the possibility to deploy and use it at the scale of a country or worldwide. We discuss these issues in the following.

6.1 Security issues

The security of current cash scheme depends on its physical characteristics, and most of the arguments against using digital cash are related to the risk of counterfeiting and malicious uses. Most of the published work on e-cash schemes focus on such issues. Although the arguments laid out against digital cash are legitimate, it is important to mention that achieving absolute security is wishful thinking, since nowadays even physical cash is still subject to counterfeit and money laundering. Recently in [1], it was argued that even though ordinary counterfeit seems to have limited impact on the economy, when conducted at large scale under the guidance of some hostile country, it could lead to a disaster. The authors suggest integrating some digital certification mechanism within the physical cash while maintaining its fundamental characteristics (i.e. anonymity, transferability etc.), keeping it easy to produce and keeping its current appearance for user acceptability. In addition to that, the digital element must allow checking easily the authenticity of the cash and make duplication worthless. The corresponding new cash scheme referred to as *physical digital cash* involves the same security challenges faced with hardware platforms and online verification as discussed previously. This underscores the fact that the viability of physical cash is still an issue open for debate.

E-cash does not involve necessarily new security weaknesses but it does introduce different ones. In particular, it seems unrealistic that one would use a fully anonymous payment scheme without increasing considerably the risk of frauds. It must be noted that in many countries the maximum amount of cash transactions allowed is limited by the law. Some proposals in the literature suggest handling transactions differently according to their amounts and allowing anonymity revocation for transactions involving large amounts. In [9], issues and challenges underlying anonymous payment systems are discussed. In particular two different classes of systems are considered: systems where anonymity can be removed by authorized people and systems where the anonymity of the consumer is guaranteed for payments involving small amounts of money over a limited period of time. It is argued that proposed systems, in particular escrowed systems (*i.e.* based on pseudonymity) fail to address several important issues, in particular in cases where a bank is attacked. The discussion focuses essentially on financial and legal issues (e.g., money laundering), with limited consideration for technical aspects. The authors claim that to be safe, a system must not support transferability and must forbid anonymous transactions beyond certain amount over specific period of time. This raises two important issues:

- Forcing all transactions to go through the bank would certainly lead to a heavy system, and as a result would not only allow banks to charge huge fees on exchanges but would also impact negatively system distribution.
- It is not clear how to account for all the transactions made by a consumer over specified period of time while remaining anonymous.

To summarize, the arguments put forward by the authors are debatable, in particular the notion of attack seems here to refer to organized crime. In effect three kinds of security objectives are targeted with e-cash :

(1) Ensuring the user that he would be able to use it without any restriction when he behaves honestly and that he would not be a victim of theft, or of any other unpleasant situations (e.g., identity fraud, etc.).
(2) Ensuring that the scheme cannot be used to launch large scale criminal actions.
(3) Ensuring that the scheme is robust and reliable.

Even though the first item seems to be the most striking one and is a necessary precondition for consumer adoptability, the second one is much more essential. In effect, we must recognize that the payment card system, despite its reported limitations, works relatively well and that possible unpleasant situations faced by customers such as losses or thefts are covered by various forms of insurance. Organized crime linked for instance to money laundering is much more difficult to circumvent. For example [9] examines various criminal behaviors like blackmailing and money laundering, and recommends limiting purely anonymous transactions and relying on a blind auditable scheme based on public information instead of secret keys. These solutions can create discomfort for the average user, and make it necessary to make some tradeoffs between the different required properties. However, an attack against the global monetary scheme could have more important fallouts, the most obvious one being large-scale fabrication of counterfeited cash. In [45], an original solution is presented that suggests basing system security on public information rather than on one or several secret keys, the theft of which could jeopardize the entire system. In the proposed approach, the bank would have to maintain a hash tree whose leaves correspond to valid coins. The roots of the trees are public and the proof of validity of a coin consists of identifying (using zero-knowledge proof) in the tree a path from the coin to a valid root. Despite its novelty, the proposed approach faces some challenges concerning its implementation and data updates. It seems that the security of the scheme would require some compromise between user comfort and basic freedom.

6.2 Distribution issues

Most of the literatures on e-cash have dealt with the security issues related to underlying mechanisms and communication protocols since [12]. These works are essential since they have allowed laying down the foundation of e-cash and led to relatively satisfactory solutions. Still there are some important open issues, which according to the previous discussions are related to system implementation and distribution, and can be articulated into two major thrusts as follows:

- Performance and scalability issues.
- Security issues related (specifically) to system distribution and scaling.

Performance issues can be important deterrent for real-life deployment of an e-cash scheme. Nowadays, payment in retail store should not take more than few seconds and it is important to keep up with the pace. Although data replication and load balancing issues in highly distributed servers have been well studied and addressed for traditional applications, there are several unaddressed challenges when it comes to e-cash which involves several new security requirements. In effect, we are faced here with two contradictory requirements. On one hand, the models that apparently are the safest against double-spending and counterfeit are online schemes, which are non-transferable and in which every transaction goes through the bank, and as such are inherently centralized. On the other hand usability and performance needs are arguments in favor of adopting distributed models. These two aspects seem contradictory in the first place; however number of security properties that one might qualify as secondary (in such settings) such as availability also plead for adopting distributed architecture, with multiple redundant backup to ensure a minimum level of service for payment verification in fail-safe mode. A key risk of a fully centralized architecture is a denial of service attack that could paralyze the entire banking system.

Hence, since system distribution seems necessary for deploying e-cash schemes, the existing protocols must be revisited to take such requirement into account. For instance, all the protocols that involve maintaining a database of spent coins are inherently dependent on how it will be accessed. If it is acceptable that the database be consulted for transactions above certain amount, it is inadmissible to have to access it for every transaction. We may consider using distributed database technology; a significant amount of literature exists on such topic since [44]. However, for security verifications, delay in data update cannot be afforded, which is a more likely scenario when using distributed database technology considering the huge volume of data transfers. In effect, it is possible to launch an attack in which a large number of distributed clients use the same coin synchronously to purchase items at different merchants.

The new e-cash schemes based on multi-agents [22, 23, 25] or Peer-to-Peer architectures [41] are promising even though they are not yet well developed and still carry some open issues. They attempt to distribute risks and responsibilities between several different parties. They are based on cooperative schemes in which the certification role played traditionally by the bank is distributed or transferred to other parties. These approaches achieve an effective and practical compromise between solution applicability and risks mitigation. In effect, users may incur minor loss like when using other traditional payment systems (e.g., payment cards), which as long as they are limited and distributed will be covered by various insurances. Future solutions will probably be such kinds of systems where achieving absolute security will only be an attribute of the banking system itself that is responsible for detecting counterfeit.

7. Conclusions

Physical cash is an anachronism in a society that is quickly progressing toward hardware-less or paper-less environments. This can probably be explained for the most part by the attachment of people to their current ways of life and the strong fear of change. On the other hand it must be recognized that the solutions proposed in the literature to move away from physical cash by adopting e-cash have not reached the level of maturity required for large-scale adoption.

In effect, security protocols and mechanisms underlying e-cash schemes have widely been studied in the research literature, but only a few of these proposals have covered implementation and architectural issues. However, such issues are essential for real-life deployment of the proposed schemes. Large-scale deployment of e-cash schemes happen naturally in distributed environments. In this paper, we have surveyed various e-cash proposals by focusing primarily on underlying architectures and discussing corresponding system distribution challenges. However, proposed architectures lack of depth and in some cases carry important flaws from distributed processing perspective. We believe that the transfer of e-cash from the academic world to real-life use will necessitate developing scalable systems that can be sized for real-life demands and workloads, and that will integrate distributed processing constraints in the design and analysis of underlying protocols and mechanisms from the beginning.

References

[1] Acquisti, A., Christin, N., Parno, B., and Perrig, A. Countermeasures against government-scale monetary forgery. In *Proc. 12th International Conference on Financial Cryptography and Data Security (FC08)* (Cozumel, Mexico, 2008), Lecture Notes in Computer Science, Springer.

[2] Anderson, R. J., Manifavas, C., and Sutherland, C. Netcard - a practical electronic-cash system. In *Proc. International Workshop on Security Protocols* (Cambridge, United Kingdom, 1996), vol. 1189 of *Lecture Notes in Computer Science*, Springer, pp. 49–57.

[3] Asokan, N., Janson, P. A., Steiner, M., and Waidner, M. The state of the art in electronic payment systems. *IEEE Computer 30*, 9 (1997), 28–35.

[4] Bellare, M., Garay, J. A., Hauser, R., Herzberg, A., Krawczyk, H., Steiner, M., Tsudik, G., Herreweghen, E., and Waidner, M. Design, implementation, and deployment of the ikp secure electronic payment system. *IEEE Journal on Selected Areas in Communications 18*, 4 (2000), 611–627.

[5] Brands, S. Untraceable off-line cash in wallets with observers (extended abstract). In *Proc. 13th Annual International Cryptology Conference (CRYPTO'93)* (Santa Barbara, California, USA, 1993), vol. 773 of *Lecture Notes in Computer Science*, Springer, pp. 302–318.

[6] Brands, S. Off-line electronic cash based on secret-key certificates. In *Proc. 2nd Latin American Symposium on Theroretical Informatics (LATIN'95)* (Valparaíso, Chile, 1995), vol. 911 of *Lecture Notes in Computer Science*, Springer, pp. 131–166.

[7] Buttyán, L., and Salem, N. B. A payment scheme for broadcast multimedia streams. In *Proc. 6th IEEE Symposium on Computers and Communications (ISCC 2001)* (Hammamet, Tunisia, 2001), IEEE Computer Society, pp. 669–673.

[8] Camenisch, J., and Stadler, M. Efficient group signatures for large groups. In *Proc.17th Annual International Cryptology Conference (CRYPTO'97)* (Santa Barbara, California, USA, 1997), vol. 1294 of *Lecture Notes in Computer Science*, Springer, pp. 410–424.

[9] Cash, O. A. E., and Crime. Tomas sander and amnon ta-shma. In *Proc. 2nd International Workshop on Information Security (ISW'99)* (Kuala Lumpur, Malaysia, 1999), vol. 1729 of *Lecture Notes in Computer Science*, Springer, pp. 202–206.

[10] Chaum, D. Blind signatures for untraceable payments. In *Proc. Advances in Cryptology (CRYPTO'82)* (1982), pp. 199–203.

[11] Chaum, D. Online cash checks. In *Proc. Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT'89)* (Houthalen, Belgium, 1989), vol. 434 of *Lecture Notes in Computer Science*, Springer, pp. 288–293.

[12] Chaum, D., Fiat, A., and Naor, M. Untraceable electronic cash. In *Proc. 8th Annual International Cryptology Conference (CRYPTO'88)* (Santa Barbara, California, USA, 1988), vol. 403 of *Lecture Notes in Computer Science*, Springer, pp. 319–327.

[13] Chaum, D., and Pedersen, T. P. Transferred cash grows in size. In *Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT'92)* (Balatonfüred, Hungary, 1992), vol. 547 of *Lecture Notes in Computer Science*, Springer, pp. 390–407.

[14] Chaum, D., and van Heyst, E. Group signatures. In *Proc. Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT'91)* (Brighton, UK, 1991), vol. 547 of *Lecture Notes in Computer Science*, Springer, pp. 257–265.

[15] Deng, R. H., Han, Y., Jeng, A. B., and Ngair, T.-H. A new on-line cash check scheme. In *Proc. 4th ACM Conference on Computer and Communications Security (CCS'97)* (Zurich, Switzerland, 1997), pp. 111–116.

[16] Even, S., and oded Goldreich. Electronic wallet. In *Proc. Advances in Cryptology (CRYPTO'83)* (1984), Plenum Press, New York, pp. 383–386.

[17] Ferguson, N. Extensions of single-term coins. In *Proc.13th Annual International Cryptology Conference (CRYPTO'93)* (Santa Barbara, California, USA, 1993), vol. 773 of *Lecture Notes in Computer Science*, Springer, pp. 292–301.

[18] Ferguson, N. Single term off-line coins. In *Proc. Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT'93)* (Lofthus, Norway, 1993), vol. 765 of *Lecture Notes in Computer Science*, Springer, pp. 318–328.

[19] Ferrer-Gomila, J. L., Payeras-Capellà, M., and i Rotger, L. H. A fully anonymous electronic payment scheme for b2b. In *Proc. International Conference Web Engineering (ICWE 2003)* (Oviedo, Spain, 2003), vol. 2722 of *Lecture Notes in Computer Science*, Springer, pp. 76–79.

[20] Frankel, Y., Tsiounis, Y., and Yung, M. Fair off-line e-cash made easy. In *Proc. International Conference on the Theory and Applications of Cryptology and Information Security – Advances in Cryptology (ASIACRYPT '98)* (Beijing, China, 1998), vol. 1514 of *Lecture Notes in Computer Science*, Springer, pp. 257–270.

[21] Gabber, E., and Silberschatz, A. Agora: A minimal distributed protocol for electronic commerce. In *Proc. 2nd USENIX Workshop on Electronic Commerce* (Oakland, CA, 1996), pp. 223–232.

[22] Guan, S. U., and Hua, F. A multi-agent architecture for electronic payment. *International Journal of Information Technology and Decision Making 2*, 3 (2003), 497–522.

[23] Guan, S. U., Tan, S. L., and Hua, F. A modularized electronic payment system for agent-based e-commerce. *Journal of Research and Practice in Information Technology 36*, 2 (2004), 67–87.

[24] Hoepman, J.-H., and Jacobs, B. Increased security through open source. *CoRR: Computing Research Repository abs/0801.3924* (2008).

[25] Jakobsson, M., and Juels, A. X-cash: Executable digital cash. In *Proc. 2nd International Conference on Financial Cryptography (FC'98)* (Anguilla, British West Indies, 1998), vol. 1465 of *Lecture Notes in Computer Science*, Springer, pp. 16–27.

[26] Jamali, N., and Zhao, X. A scalable approach to multi-agent resource acquisition and control. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005)* (Utrecht, Netherlands, July 2005), ACM Press, pp. 868–875.

[27] Jarecki, S., and Odlyzko, A. M. An efficient micropayment system based on probabilistic polling. In *Proc. 1st International Conference on Financial Cryptography (FC'97)* (Anguilla, British West Indies, 1997), vol. 1318 of *Lecture Notes in Computer Science*, Springer, pp. 173–192.

[28] Lamport, L. Password authentication with insecure communication. *Communications of the ACM 11*, 24 (1981), 770–772.

[29] Law, L., Sabett, S., and Solinas, J. How to make a mint: the cryptography of anonymous electronic cash. Tech. rep., National Security Agency, Office of Information Security Research and Technology, Cryptology Division, 1996.

[30] Lipton, R. J., and Ostrovsky, R. Micropayments via efficient coin-flipping. In *Proc. 2nd International Conference on Financial Cryptography (FC'98)* (Anguilla, British West Indies, 1998), vol. 1465 of *Lecture Notes in Computer Science*, Springer, pp. 1–15.

[31] Liu, J. K., Tsang, P. P., and Wong, D. S. Recoverable and untraceable e-cash. In *Proc. 2nd European Public Key Infrastructure Workshop: Research and Applications (EuroPKI 2005)* (Canterbury, UK, 2005), vol. 3545 of *Lecture Notes in Computer Science*, Springer, pp. 206–214.

[32] Low, S. H., Maxemchuk, N. F., and Paul, S. Anonymous credit cards and their collusion analysis. *IEEE/ACM Transactions on Networking 4*, 6 (1996), 809–816.

[33] Lysyanskaya, A., and Ramzan, Z. Group blind digital signatures: A scalable solution to electronic cash. In *Proc. 2nd International Conference on Financial Cryptography (FC'98)* (Anguilla, British West Indies, 1998), vol. 1465 of *Lecture Notes in Computer Science*, Springer, pp. 184–197.

[34] Medvinsky, G., and Neuman, B. C. Netcash: A design for practical electronic currency on the internet. In *Proc. 1st ACM Conference on Computer and Communications Security (CCS'93)* (Fairfax, Virginia, USA, 1993), ACM, pp. 102–106.

[35] Meng, B., and Qianxing. Socpt: A secure online card payment protocol. In *Proc. 8th International Conference on Computer Supported Cooperative Work in Design* (2004), pp. 679–684.

[36] Merkle, R. C. A certified digital signature. In *Proc. 9th Annual International Cryptology Conference (CRYPTO'89)* (Santa Barbara, California, 1989), vol. 435 of *Lecture Notes in Computer Science*, Springer, pp. 218–238.

[37] Neuman, B. C., and Medvinsky, G. Requirements for network payment: The netcheque perspective. In *Proc. Technologies for the Information Superhighway (COMPCON'95)* (San Francisco, California, USA, 1995), IEEE-CS, pp. 32–36.

[38] Nguyen, K. Q., Mu, Y., and Varadharajan, V. Secure and efficient digital coins. In *Proc. 13th Annual Computer Security Applications Conference (ACSAC 1997)* (San Diego, CA, USA, 1997), IEEE Computer Society, pp. 9–15.

[39] Okamoto, T., and Ohta, K. Disposable zero-knowledge authentication and their applications to untraceable electronic cash. In *Proc. 9th Annual International Crytology Conference (CRYPTO'89)* (Santa Barbara, California, USA, 1990), vol. 435 of *Lecture Notes in Computer Science*, Springer, pp. 481–496.

[40] Okamoto, T., and Ohta, K. Universal electronic cash. In *Proc. 11th Annual International Cryptology Conference (CRYPTO'91)* (Santa Barbara, California, USA, 1992), vol. 576 of *Lecture Notes in Computer Science*, Springer, pp. 324–337.

[41] Osipkov, I., Vasserman, E. Y., Hopper, N., and Kim, Y. Combating double-spending using cooperative p2p systems. In *Proc. 27th IEEE International Conference on Distributed Computing Systems (ICDCS 2007)* (Toronto, Canada, 2007), IEEE Computer Society, p. 41.

[42] Pagnia, H., and Jansen, R. Towards multiple-payment schemes for digital money. In *Proc. 1st International Conference on Financial Cryptography (FC'97)* (Anguilla, British West Indies, 1997), vol. 1318 of *Lecture Notes in Computer Science*, Springer, pp. 203–216.

[43] Rivest, R. L., and Shamir, A. Payword and micromint: Two simple micropayment schemes. In *Proc. Security Protocols Workshop* (Cambridge, United Kingdom, 1996), vol. 1189 of *Lecture Notes in Computer Science*, Springer, pp. 69–87.

[44] Rothnie, J. B., and Goodman, N. A survey of research and development in distributed database management. In *Proc. 3rd international conference on Very large data bases (VLDB'1977)* (Tokyo,

Japan, 1977), VLDB Endowment, pp. 48–62.

[45] Sander, T., and Ta-Shma, A. Auditable, anonymous electronic cash (extended abstract). In *Proc.19th Annual International Cryptology Conference (CRYPTO'99)* (Santa Barbara, California, USA, 1999), vol. 1666 of *Lecture Notes in Computer Science*, Springer, pp. 555–572.

[46] Sirbu, M., and Tygar, D. Netbill: An internet commerce system optimized for network delivered services. In *Proc. Technologies for the Information Superhighway (COMPCON'95)* (San Francisco, California, USA, 1995), IEEE-CS, pp. 34–39.

[47] Sirbu, M. A. Credits and debits on the internet. *IEEE Spectrum 34*, 2 (1997), 23–29.

[48] Tewari, H., O'Mahony, D., and Peirce, M. Reusable off-line electronic cash using secret splitting. Tech. Rep. TCD-CS-1998-27, Computer Science Department, Trinity College, Dublin, 1998.

[49] Tracz, R., and Wrona, K. Fair electronic cash withdrawal and change return for wireless networks. In *Proc. 1st International Workshop on Mobile Commerce* (Rome, Italy, 2001), ACM, pp. 14–19.

[50] van Antwerpen, H. Electronic cash. Master's thesis, CWI, 1990.

[51] Varadharajan, V., Nguyen, K. Q., and Mu, Y. On the design of efficient rsa-based off-line electronic coin schemes. *Theoretical Computer Science 226*, 1–2 (1999), 173–184.

[52] Wang, H., and Zhang, Y. Untraceable off-line electronic coin flow in e-commerce. In *24th Australian Computer Science Conference (ACSC'01)* (GoldCoast, Australia, 2001), IEEE Computer Society, pp. 191–198.

[53] Watson, A. Electronic cash and set. In *Proc. Internet Crime* (Melbourne, 1998), Australian institute of Criminology.

[54] Wei, K., Smith, A. J., Chen, Y.-F. R., and Vo, B. Whopay: A scalable and anonymous payment system for peer-to-peer environments. In *Proc. 26th IEEE International Conference on Distributed Computing Systems (ICDCS 2006)* (Lisboa, Portugal, 2006), IEEE Computer Society, p. 13.

[55] Wuu, L.-C., Lin, C.-M., and Wang, W.-F. Anonymous and transferable coins in pay-fair e-commerce. *IEICE - Transactions on Information and Systems E89-D*, 12 (2006), 2950–2956.

[56] Xu, Q., and Zhao, H. Distributed electronic payment system based on bank union. In *Proc. 4th International Conference on High-Performance Computing in the Asia-Pacific Region* (Beijing, China, 2000), vol. 1, IEEE Computer Society, pp. 548–551.

[57] Yacobi, Y. On the continuum between on-line and off-line e-cash systems. In *Proc. 1st International Conference on Financial Cryptography (FC'97)* (Anguilla, British West Indies, 1997), vol. 1318 of *Lecture Notes in Computer Science*, Springer, pp. 193–202.

[58] Yacobi, Y. Risk management for e-cash systems with partial real-time audit. *Netnomics 3* (2001), 119–127.

[59] Yang, B., and Garcia-Molina, H. Ppay: micropayments for peer-to-peer systems. In *Proc. 10th ACM Conference on Computer and Communications Security (CCS 2003)* (Washingtion, DC, 2003), ACM Press, pp. 300–310.

[60] Yang, Z., Lang, W., and Tan, Y. A new fair micropayment system based on hash chain. In *Proc. IEEE International Conference on e-Technology, e-Commerce, and e-Services (EEE 04)* (Taipei, Taiwan, 2004), IEEE Computer Society, pp. 139–145.

[61] Yen, S.-M. Payfair: A prepaid internet micropayment scheme ensuring customer fairness. *IEE Proceedings: Computers and Digital Techniques 148*, 6 (2001), 207–213.

[62] Zhu, F., Guan, S.-U., and Yang, Y. *Internet Commerce and Software Agents: Cases, technologies and Opportunities.* IDEA Group Publishing, 2000, ch. SAFER E-Commerce: Secure Agent Fabrication, Evolution & Roaming for E-Commerce, pp. 190–206.