Implementing regular cash with blind fixed-value electronic coins[1,2]

Dimitrios Lekkas[1,*], Diomidis Spinellis[2]

[1] Department of Product and Systems Design Engineering, University of the Aegean, Syros Island GR-84100, Greece, email: dlek@aegean.gr

[2] Department of Management Science and Technology, Athens University of Economics & Business, 76 Patission St., Athens GR-10434, Greece, email: dds@aueb.gr

Abstract

We propose a generic framework for the development of an off-line peer-to-peer electronic payment system with fixed-value electronic coins. The proposed scheme simulates the issuance, circulation and characteristics of conventional cash and satisfies important security, privacy and usability requirements, offering a comparative advantage over other existing solutions. We exploit the strong security characteristics of a typical tamperproof device such as a smart card, and the trust services provided by a Public Key Infrastructure. The electronic coins are fixed, signed data structures that may be distributed, stored and processed by external applications only in their blind (encrypted) form. Non-divisibility is counterbalanced by the low complexity of the proposed solution, which enables the efficient exchange of multiple coins per transaction.

Keywords: Security; Electronic payments; Smart card; Trust; Public Key Infrastructure

1. Introduction

Electronic payment schemes provide an alternative form of cash to traditional notes and coins. The benefits of electronic cash, such as the reduced handling cost, the convenience, the ability to perform remote payments and the suitability for unattended vending machines, are evident. However, today the business case for various bank-issued electronic purse applications looks quite poor. In fact, except of the widely recognized credit/debit systems, it is still early days in the world of digital money and we are within a transition period towards a globally acceptable electronic cash scheme. At the end of this transition period it is even likely that a significant fraction of the retail payments will be performed outside of the traditional banking system [1].

An ideal digital cash system has to satisfy several baseline requirements: fraud protection, privacy of its users, no dependency on specific means for storage and transmission, off-line usage, peer-to-peer transferability, value divisibility and low complexity. Many e-cash schemes have been proposed, but none of them satisfies the above requirements as a whole in a feasible, low-cost and efficient manner.

---

[1] Dimitrios Lekkas and Diomidis Spinellis, Implementing regular cash with blind fixed-value electronic coins, *Computer Standards & Interfaces*, 29(3):277–288, March 2007.

[2] This is a machine-readable rendering of a working paper draft that led to a publication. The publication should always be cited in preference to this draft using the reference in the previous footnote. This material is presented to ensure timely dissemination of scholarly and technical work.

[*] Corresponding Author, Tel. +302281097100, Fax. +3012281097099

Most of the known digital payment systems sacrifice one or more of the requirements, in favor of some others, as briefly described in the next section.

The main contribution of this paper is the presentation of a practical, simple and secure approach to electronic payment schemes, which simulates the issuance and circulation of conventional money. The power of public key cryptography [2] combined with the security features offered by smart cards [3] with crypto-processor and the wide acceptance of the Extensible Markup Language (XML) [4], provide the valuable means for the construction of an open electronic payments scheme based on transferable, anonymous, verifiable, interoperable but invisible electronic coins. We argue that the proposed e-cash solution offers a comparative advantage over other existing schemes, because it simultaneously satisfies all the required characteristics of a payment system, except that of coin divisibility, at a low-cost and low-complexity approach. The characteristics that differentiate our scheme from other solutions is the elimination of the need for online authorities, the peer-to-peer (P2P) transferability through open networks such as the Internet, the open XML-based structure and their possible usage in a variety of systems. Such systems could be a Point-of-Sale (POS) device, a PC equipped with an ISO-7816 compliant smart card reader or an Automatic Teller Machine (ATM). The scheme offers desirable characteristics of electronic cash such as the anonymity and the protection against forgery and double-spending. The lack of coin divisibility is counterbalanced completely by the efficiency and the simplicity of the system, which enable the exchange of multiple coins per transaction, in an acceptable time and overhead.

2. Existing Standards and Approaches

Electronic cash has been a popular subject for academic and business research for many years. A plethora of standards and approaches on e-cash have been proposed. A direct comparison of their individual characteristics is not always appropriate since each of them is designed under different perspectives and conditions (e.g. micropayments, point-of-sales presence, internet transactions etc.). Existing e-cash schemes are generally categorized into *online* systems where the intervention of a bank or an authority is required in every transaction and the *off-line* systems where a payment can be executed directly between a merchant and a customer. Perfect e-cash would be off-line, anonymous, peer-to-peer transferable, independent of physical locations and devices, divisible to any value and secured against copy and reuse.

Approaches that tried to satisfy all the above requirements [13] introduce very complex mathematics as well as heavy network traffic and their implementation tends to be inefficient and risky. Systems that satisfy the properties of value divisibility and double-spending protection for example, require either an online authority that records the transactions and provides the ‘cash pools’ for credit/debit or special ‘point of sale’ devices which can only communicate to other devices through private circuits or dedicated telephone lines. Additionally, the electronic coins presented so far cannot be stored in repositories other than specially designed secure tokens or online databases, while in most cases the cash’s authenticity cannot be verified during its transfer between e-purses. Similarly, other systems require the distinction in the infrastructure between a payer and a payee (typically a bank in this case) and therefore cash is not transferable between simple users [6].

Off-line schemes propose various solutions with non-divisible or divisible electronic coins [7] [8]. In both cases the problem is the increased complexity of the schemes, which affects the efficiency of the system and makes them extremely difficult to understand, implement and debug. A second problem is that the detection of a possible double-payment with the same electronic value is performed asynchronously. In other words the fraudulent transaction can be successfully executed off-line and the double usage of value will be detected at a later time, after a communication with the central authoritative database is restored.

NetCash satisfies many practical requirements for e-payments but lacks anonymity [9]. Card-based purses such as Mondex and Visa Cash and their online equivalents such as Ecash and Cybercoin did not gain any real traction, for a variety of reasons [10] [11]. A drawback is the need for a substantial investment on infrastructure, while the deployment of new payment mechanisms in several environments offers little comparative advantage against credit cards or traditional cash. Another

negative aspect is the reduced openness of the systems and the lack of interoperability, although the Common Electronic Purse Specifications (CEPS) [12] is continuously gaining common consent.

3. Overview and Technology Background

3.1 Motivation and basic idea

In conventional monetary systems a note has a fixed non-divisible value, is identified by a serial number and is protected against forgery by special material characteristics. Accordingly, in the electronic world a coin may be a fixed data structure accompanied by a digital signature [2] issued by the e-cash issuer, which identifies it uniquely, ensures its integrity and proves its authenticity. As in any digital cash system, the main difficulty arising is that a soft coin — unlike a regular coin — can be indefinitely duplicated at no considerable cost and effort. At the same time the distinction between 'original' and 'duplicate' data is not possible. The challenge is to protect the duplication of an e-coin, satisfying the requirement of unreusability (no double-spending) of the same value. The idea towards this solution is that the content of an e-coin must be kept secret, but verifiable, through-out its life cycle. Such coins can have an open XML-based structure and can be stored in plain text only in tamperproof tokens such as smart cards. Their 'invisible' encrypted form may be stored in any means, circulated locally or via open networks while they can only be further processed by a specific beneficiary. The decryption and the verification of their authenticity are performed within the secure token of the payee. Therefore no one — except the cash issuer — can possess an original coin in clear text, and therefore conceivably reproduce it for multiple payees. Further considerations on double-spending protection, as well as other security issues, mechanism details and additional requirements for the electronic coins are discussed in the next sections.

3.2 Fixed-value coins versus variable-value e-cash

Cash is by default made up of non-divisible basic units (e.g. one eurocent) both for simplifying accounting and because cash is traditionally implemented by coins representing a fixed value. Theoretically, we do not need anything more than coins of one eurocent to complete any transaction. Practically, a payment executed only by using the smallest cash values is inefficient in both its material and its electronic form. The majority of the known commercial or research e-cash systems use e-cash values that can be divided in any portion and perform a transaction of any value up to the balance of the e-purse [5]. The basic advantage of variable-value e-cash is the convenience it offers in making an exact value payment and the limited amount of data exchanged within a transaction. However the drawback of these systems is typically a requirement for an online connection for authorization, verification and double-spending protection purposes. Other solutions with divisible coins [13] have the drawback of increased complexity.

Fixed value coins on the other hand can be handled as autonomous structures and they exhibit a number of significant advantages:

+ simplicity and autonomy of storage and transmission procedures;
+ as fixed autonomous structures, their originality and integrity can be traceable at any point of their lifecycle;
+ the e-cash issuer is always recognizable;
+ they may be distributed and stored in a protected form, which ensures that they cannot be duplicated and/or reused;
+ they promote interoperability as they may be independently used by heterogeneous systems;
+ their structure can be open and portable.

We cannot neglect though that fixed value coins have also some disadvantages:

– exact amount payments may fail in some cases, unless fixed values are set to the smallest possible unit;

– the exchanges needed to complete a transaction may be multiplied, since more than one coin exchange may be needed to make a payment;

– the size of an electronic purse is increased, since each coin is an independent data structure.

However, capacity and efficiency problems can be overcome (see sections 6.2 and 6.3) and therefore the eventual exchange of multiple coins per transaction, does not considerably increase the total overhead. Consequently, the aforementioned important advantages of the fixed-value coins, give them a clear superiority in terms of simplicity, traceability and efficiency, as we will further analyze in the next sections.

3.3 Smart card security features and Public Key Infrastructure

Smart cards are widely considered as tamperproof devices that can be deployed in a variety of portable and secure applications. Smart cards in conjunction with relevant internal (executed within the smart card) and external processes contribute to the construction of highly secure information infrastructures [14]. Multi-application or application-specific smart cards are essential for the full exploitation of public key cryptography since they provide a highly secure means for generating, storing and using cryptographic keys [3], for encrypting and decrypting data, for producing and verifying electronic signatures as well as for integrating other secure internal applications. Many sSmart cards also have built-in functions for the management of digital certificates [15], which are issued by trusted authorities namely the Certification Services Providers (CSP) and constitute public evidence that a specific public key belongs to a named physical or logical entity. Such smart cards are able to perform basic and advanced cryptographic functions internally, providing confidentiality, authentication, information integrity and non-repudiation, in an indisputable manner.

The most important characteristic of smart cards that gives value to the proposed solution is their capability of providing various levels of Access Conditions (AC) for different areas (directories) within their file structure. A storage area in a smart card can be configured for storing sensitive or confidential data (e.g. keys) that must never be exported to any external application, but can be used only by specific internal functions. This is a key feature of smart cards that is exploited by our proposed solution for the secure storage of electronic coins, keys and related information in clear text that is never exposed to the public, including the cardholder herself.

A typical smart card with a crypto-processor that is widely used in the market for public key applications as a personal identification token or a digital signature generator device, has a customizable file structure [16] and a number of built-in commands providing basic functionality. A typical file structure of a PKI enabled smart card, has built-in directories with Access Conditions such as ‘NEVER’, ‘ALWAYS’, ‘AUT KEY’ denoting respectively that data cannot be externally accessed or they are public or they can be accessed by a user authenticated with the relevant key or PIN. The file system of a smart card is of course further customizable, while it can also host custom applications and code written in Java or .NET platforms.

4. Solution Requirements and Features

4.1 Assumptions

The proposed solution is built upon two basic assumptions that are very common in any relevant security-sensitive application:

1) A smart card is considered as tamperproof device for storing and processing sensitive data such as cryptographic keys [3].

2) The Public Key Infrastructure is an acceptable trust-establishment mechanism for all parties involved and the relevant algorithms used for data encrypting, hashing and signing are strong and valid [17].

4.2 Functional architecture

The principle of the proposed mechanism is that *electronic coins (e-coins*) are data structures that represent real fixed and non-divisible values, stored in a protected area of the EEPROM of a smart card, which also offers specific processing capabilities. E-coins can be securely transferred from one smart card to another by a cryptography-based application that consists of several internal (protected within the smart card) and external (in the Application Program Interface (API) or a device) functions. According to the relevant abstract functional model shown in Fig. 1 we distinguish the 'Protected space' where data is accessible only by internal methods and the 'API space' where data may be available to any external API that handles the communication with the smart card and the public network. The two separate spaces are represented by UML (Unified Modeling Language) classes. The smart card enabled for this application will be referred to as an *electronic purse (e-purse*). An e-purse can be used in any card reader mounted in a device such as an ATM, a POS terminal device or a PC. Transactions may be executed over any open or private network.

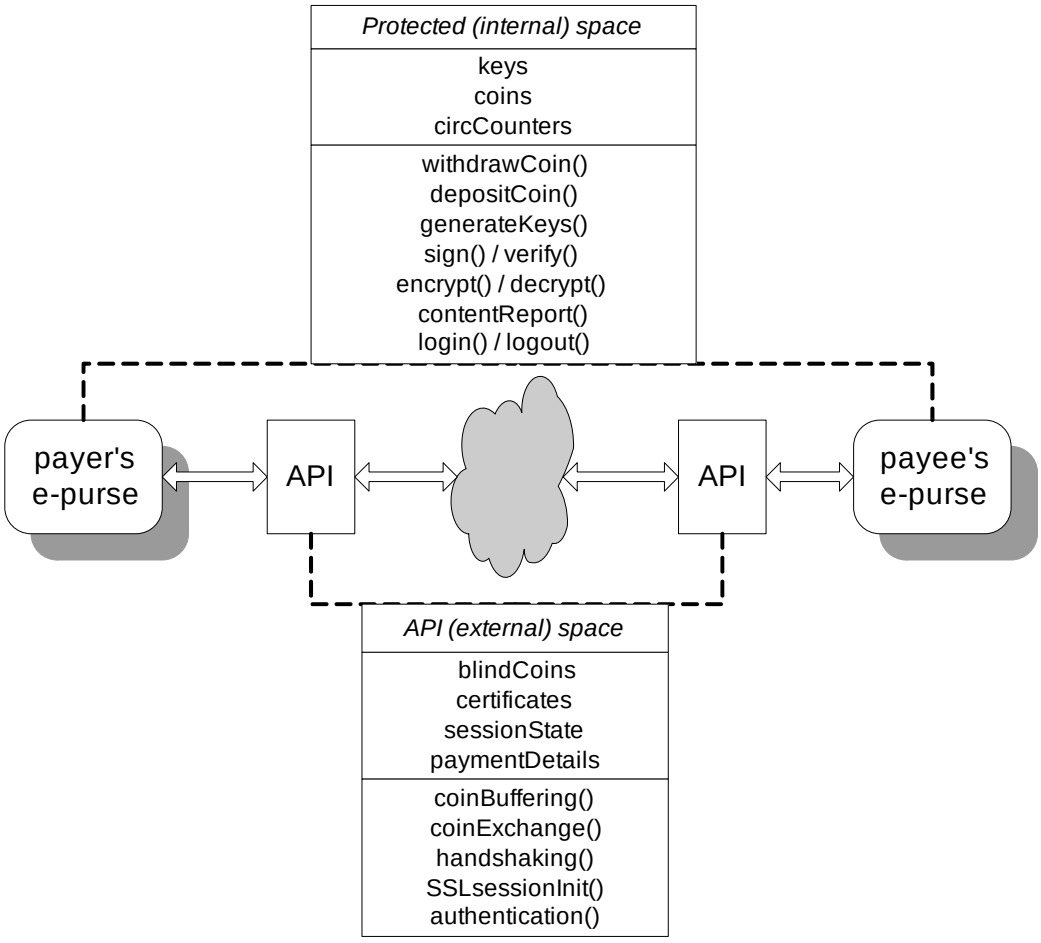

Fig. 1. Abstract Functional Model

4.3 Personalization of e-purse

The construction of the smart card and the personalization of the e-purse are security-critical procedures that must be performed by authorized entities with a security level analogous to a mint that

prints regular notes [18]. A typical PKI-enabled smart card [16] has all the necessary built-in functionality for generating and storing keys, as well as for encrypting/decrypting and signing data needed for the implementation of the solution. Except of the basic file structure of a smart card with PKI crypto-processor, some additional elementary files are needed for the storage of plain-text e-coins and the related Circulation Counters (see section 4.7) with strict access conditions, as described in [Table I]. Due to smart card EEPROM capacity restrictions, the data structures are stored in raw binary values rather than in their XML form, requiring approximately 150 bytes per e-coin, according to the schema presented in section 5.1. The trusted internal functions for depositing and withdrawing an e-coin by decrypting and encrypting XML data respectively must be also installed in the smart card during its formatting-personalization phase.

Directory ID	Elementary File ID	Description	File Type	Length	Access Conditions			
					Read Bin or Dir Next	Update Bin or Decrease	Increase or Delete File	Create Record or File
5F01		Coin Application Directory	DF		AutKey1	N/A	AutKey1	AutKey1
	0020	Min e-coin version accepted	Transparent EF	2 bytes	ALW	NEV	NEV	NEV
	0100	Coin Records	Fixed-Length Linear EF	~150 bytes per record	NEV	AutKey1	NEV	NEV
	0200	Circulation Counters	Fixed-Length Linear EF	6 bytes per record	NEV	AutKey1	NEV	NEV

Table I. Additional smart card files needed for the e-coin application

4.4 Offline trust model

The framework presented is based on specific trust relationships that are established between its basic actors, which must also be available off-line. The three basic actors of the system model, namely the *e-cash issuer*, the *payer* and the *payee*, have to trust each other that:

- the entity is authorized to act as an *e-coin issuer* or an *e-coin user;*
- the cryptographic keys used for e-coin operations are valid and proper (i.e. stored and used within a smart card and non-exportable to any external application);
- the identity details an entity presents for authentication and signature purposes, are real.

A PKI gives us a strong solution for the fulfillment of the above requirements. A fourth trusted actor, a *Certification Services Provider (CSP)* is added to the model and allows the establishment of trust relationships in a highly scalable manner, with the aid of the technology of digital certificates. The CSP issues digital certificates to the other participants providing them with the basic PKI functionalities of authentication, digital signature and encryption, by binding their smart card–based keys to their identity details. A certificate usage extension (such as one based on the X.509 certificate extensions [15]) determines whether an entity is authorized as an e-coin issuer or an e-coin user. It is also important to ensure that certificates are issued explicitly for keys generated within the e-purse, for security reasons explained in section 6.1.2.

Consequently, the trust relationships needed for the operation of the model are centralized, with every participant acting as a ‘relying party’ and trusting the CSP [20] rather than building a web-of-trust. The derived trust relationships between the e-cash issuer, the payer and the payee, as depicted on Fig. 2, are the result of trust transitivity, a common characteristic of the PKI. Technically, an entity that trusts the CSP is recognizing the validity of its digital certificate and therefore the CSP certificate (usually self-signed) has to be distributed in a trusted way. A key property of our trust model is that the functions for the establishment of trust are only preliminary and the management of trust thereafter remains mainly off-line. There is, however, a requirement for periodical online connection

(according to the policy of the CSP) in order to retrieve the Certificate Status Information that denotes eventual certificate revocations due to e-purse compromise or other reasons.

Besides the technical details, there are many other considerations on the establishment and the management of trust against a CSP [17]. The notion of trust against a CSP could be defined as the customer's certainty that the CSP is capable of providing the required services accurately and infallibly, a certainty which also expresses the customer's faith in its moral integrity, in the soundness of its operation, in the effectiveness of its security mechanisms, in its expertise and in its abidance by all regulations and laws, while at the same time, it also contains the acknowledgement of a minimum risk factor, by the relying party. In any case, the issuer of a digital certificate may be an individual Certification Services Provider or a separate role within the banking system itself, acting as a Trusted Third party. Nevertheless, it is permitted and in some cases desired [18] that the roles of e-cash issuer and of CSP are performed by separate non-bank entities.

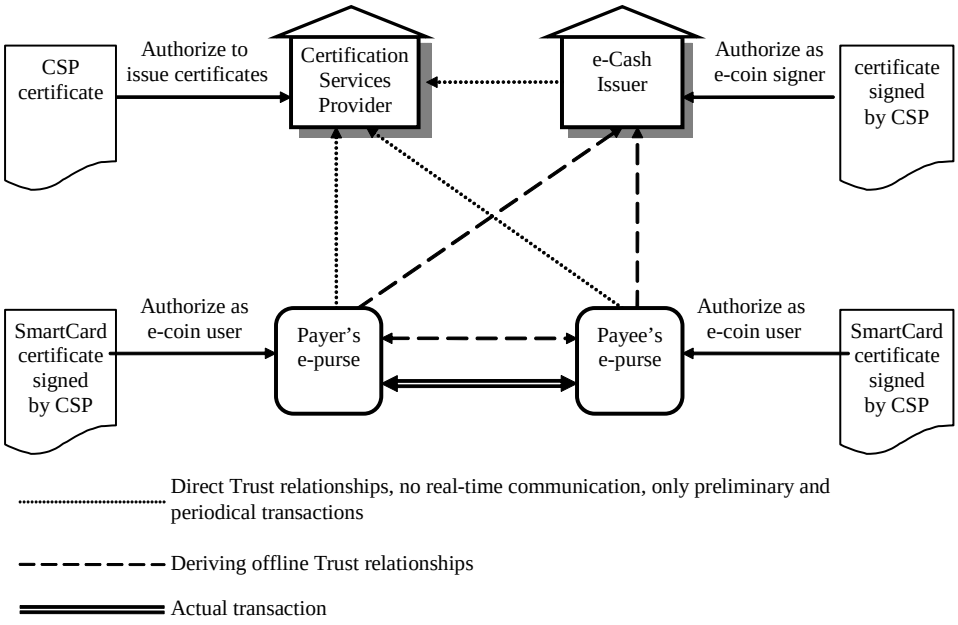

Fig. 2. Off-line Trust Model

4.5 Origin of electronic coins

In a conventional monetary system, a sole central bank is responsible for producing and issuing value in the form of cash of a specific currency. In electronic cash again a central bank or another authorized entity is responsible for manufacturing electronic value in an unforgeable form, which is then securely distributed via the banking system. The electronic cash manufacturer will be referred to as the *e-cash issuer*. The construction of e-coins of specific values is performed by the e-cash issuer by digitally signing an XML data structure as described in section 5.1. Any user thereafter will be able to trace the originality of the e-coins by validating the digital signature of the issuer. The e-cash issuer will also be responsible for performing auditable procedures, for keeping track of the issued coins and for destroying them at the end of their life-cycle. As in conventional banking systems the processes of money issuance and destruction should be governed by strict regulations.

The *origin traceability* of the exchanged e-coins at any point of their life-cycle is one of the strongest characteristics that differentiates the proposed solution from all other schemes. The advantages of the approach are that it protects the e-coins against forgery, while it enables their easy peer-to-peer offline storage and exchange, in the same way as regular cash is handled.

4.6 e-coin life cycle

The time of validity of electronic coins is limited, in order to reduce its exposure to the public, limit the size of the smart card–based coin circulation counter database (see next section) and to keep a control on its circulation. The life cycle of an e-coin ends upon the fulfillment of one or more of the following conditions:

1. The expiration date of the e-coin is reached.
2. The circulation counter reaches its limit (the limit is either denoted by the e-cash issuer or by the physical size of its storage).
3. The certificate of the e-cash issuer who signs the e-coin is expired or revoked.

Upon its invalidation the e-coin remains in the e-purse, while it cannot be further used for transactions. It can only be returned to a distributor, who exchanges it with valid coins. The distributor, in turn, returns the invalidated e-coins to the e-cash issuer who records their destruction and issues new valid coins, abiding by the rules of the monetary system.

4.7 Double-spending protection

Conventional coins and notes are distinct entities, which cannot be double-spent or duplicated at a reasonable cost and effort. Contrarily, e-coins are made of data, which is the only commodity that can be unconditionally copied and reused without any measurable cost. As in any e-payment system, the protection against double-spending (i.e. unreusability) of the e-coins is a main security concern. In our proposal, unreuseability is achieved in two phases:

1) *No replay to different payees*: Once an e-coin is withdrawn (and deleted) from a smart card by the trusted smart processes, it is encrypted with the public key of the intended payee. Consequently the blind e-coin can only be processed and deposited by one only beneficiary, while it is useless to any other party.

2) *No replay to the same payee*: It is obvious that a replayed (uniquely identified) e-coin that is already stored in the e-purse of the payee will not be accepted. In case, however, the payee withdraws (spends) the specific e-coin before it is replayed to her, we need an additional mechanism to keep track of the e-coins that are already processed by the e-purse and thus reject any attempt to re-deposit them. We introduce the *Coin Circulation Counter (CCC)* which accompanies each e-coin throughout its life-span, it is attached out of the signed part of the e-coin but within the encrypted data and it is increased by one unit each time a coin is deposited in a smart card. The e-purse keeps track of the ID of any e-coin deposited together with its CCC at that moment, building a 'mini database' within the tamperproof device, according to the paradigm of '*observers*' [19]. The e-purse will accordingly reject any incoming e-coin with the same (or less) CCC since this will indicate an attempt to replay a payment that is already executed and completed. An e-coin with a greater CCC than the one stored in the e-purse will be accepted, since this indicates that the e-coin was legally transferred to other payees and then it was legally circulated back to the initial e-purse. The usage of CCC in the exchange protocol is further analyzed in section 5.2.

The Coin Circulation Counters may serve also as an *audit mechanism*. The CCC database of an e-purse can be transmitted and recorded by the central e-cash issuer, when the e-purse eventually goes online. This central database of CCCs can be used as a secondary fraud-protection mechanism, even in the case of smart-card compromise, since it can (asynchronously) detect the existence of identical CCCs in different e-purses, indicating double-spending.

4.8 Store e-coins anywhere

Smart cards impose some capacity restrictions in exchange for their small size and mobility. An important feature that adds value to our solution is the possibility of using any external storage media to 'buffer' e-coins. Especially in cases where the transaction volume is high and mobility is not a primary concern (e.g. a bank or a store) the blind e-coins may be stored in any media. This is achieved

by simply withdrawing e-coins form the e-purse while the declared beneficiary is the payer herself. Specifically, the e-coins are encrypted with the public key of the payer (and not the payee), they are not transmitted but stored and the payer is the only person who can reuse them. As described in the detailed steps in section 5.2 an e-coin is legally withdrawn, stored elsewhere and then legally deposited back in the same e-purse.

5. Implementation

5.1 Data structures

The basic data structures used by the proposed framework are the plain-text e-coin stored in the smart card and the encrypted e-coin that may be externally stored. A generic XML schema [4] that incorporates all the requirements and features described in the previous sections is shown below.

The basic e-coin structure consists of three parts: a) The main e-coin information provided by the e-cash issuer such as value (integer expressed in the smallest unit of the currency, multiplied by a scaling factor), currency, version and identification that are signed; b) the information necessary for the recognition and the verification of the signature and the signature value; and c) the coin circulation counter that resides out of the signed part and is changing while the e-coin is circulating.

The structure of the blind e-coin that can be externally stored consists of the binary value of the encrypted e-coin and the digital certificate of the beneficiary, for reference purposes.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema  targetNamespace="http://coins.org"
         xmlns:c="http://coins.org"
         xmlns="http://www.w3.org/2001/XMLSchema">

    <!-- 1. e-coin structure processed within the smart card -->
    <element name="coin">
      <complexType name="coinType">
        <sequence>
          <element ref="c:signedData"/>
          <element ref="c:signatureInfo"/>
          <element name="coinCirculationCounter" type="unsignedShort"/>
        </sequence>
      </complexType>
    </element>

    <!-- 1a. signed part of e-coin -->
    <element name="signedData">
      <complexType name="sdType">
        <sequence>
          <element name="version" type="unsignedShort"/>
          <element name="coinID" type="unsignedInt"/>
          <element name="currency" type="c:currencyID"/>
          <element name="value" type="c:valueType"/>
          <element name="scalingFactor" type="unsignedShort"/>
          <element name="issuerID" type="unsignedShort"/>
          <element name="expires" type="unsignedInt"/>
          <element name="circulationLimit" type="unsignedShort"/>
        </sequence>
      </complexType>
    </element>

    <!-- 1b. signature part of e-coin -->
    <element name="signatureInfo">
      <complexType name="siType">
        <sequence>
          <element name="signatureMethod" type="string"/>
          <element name="signatureValue" type="base64Binary"/>
        </sequence>
      </complexType>
    </element>

    <!-- 2. structure for externally stored (buffered) blind e-coins -->
    <element name="bufferedCoin">
      <complexType name="atType">
```

```
        <sequence>
        <element name="blindCoin" type="hexBinary" />
        <element name="payeeX509Certificate" type="base64Binary" />
        </sequence>
      </complexType>
    </element>

    <!-- user defined data types -->
    <!-- currency data type – ISO code, 3 characters -->
    <simpleType name="currencyID">
      <restriction base="string">
        <minLength value="3"/>
        <maxLength value="3"/>
      </restriction>
    </simpleType>

    <!–data type for coin value – min 1 unit (e.g. cent) multiplied by scalingFactor -->
    <simpleType name="valueType">
      <restriction base="unsignedInt">
        <minInclusive value="1"/>
      </restriction>
    </simpleType>

</schema>
```

5.2 Transaction protocol

The procedures needed to perform a complete transaction on both the payer and the payee sides are either internal (built-in within the smart card) or external (executed in the API level) as described in section 4.2. This section describes the protocol for the exchange of one e-coin between two parties Alice (payer) and Bob (payee), as well as the steps required for the external buffering of e-coins. The protocol for *Blind Electronic Coin (BEC)* exchange incorporates and satisfies all the requirements listed in section 4. It is assumed that the basic cryptographic functionality of the smart card is already implemented (e.g. PIN authentication, key generation, sign/verify, encrypt/decrypt). The notation used is presented in Table II and the basic procedures of coin exchange and coin buffering follow:

$C_{i,v}$:	Plain-text e-coin of value v with unique ID i (XML element “coin”)
$B_{i,v}$:	Blind e-coin of value v with ID i
P_A and S_A :	Public and Secret key of payer *Alice* respectively
P_B and S_B :	Public and Secret key of payee *Bob* respectively
CRT_A and CRT_B :	Digital Certificates of *Alice* and *Bob* respectively
$CCC_{i,A}$ and $CCC_{i,B}$:	Circulation Counter of Coin i stored in e-purse of *Alice* and in e-purse of *Bob* respectively
e*(D, K)* :	RSA encryption of data D using key K
d*(D, K)* :	RSA decryption of cipher D using key K

Table II. Notation used for the BEC exchange protocol:

<u>*Coin exchange transaction:*</u>

1. $B \rightarrow A : v, CRT_B$

   Bob requests from *Alice* an e-coin of value v and sends his certificate for convenience in payee’s identification and keys usage.

2. *A* (internal processes) : *IF* e-coin of value v exists in e-purse *THEN* <u>withdraw coin</u>:

a. Retrieve $C_{i,v}$ including $CCC_{i,A}$ from internal protected storage

b. Compute $B_{i,v} = \boldsymbol{e}(C_{i,v}, P_B)$ and return $B_{i,v}$ to API

c. Delete $C_{i,v}$ from internal storage but keep $CCC_{i,A}$

3. $A \rightarrow B : B_{i,v}$

   Transmit blind e-coin or failure code to *Bob.*

4. *B* (internal processes) : *IF* e-purse space is sufficient *THEN* deposit coin:

   a. Compute $C_{i,v} = \boldsymbol{d}(B_{i,v}, S_B)$

   b. *IF* expiration date of $C_{i,v}$ <= *Today*
   AND

   c. $CCC_{i,A}$ <= *Circulation limit*
   AND

   d. $CCC_{i,B}$ does not exist *OR* $CCC_{i,A} > CCC_{i,B}$
   AND

   e. Signature of $C_{v,i}$ is valid and trusted
   AND

   f. Version of $C_{v,i}$ >= card's minimum acceptable coin version
   THEN

   g. Store $C_{i,v}$ in the smart card

5. $B \rightarrow A : ACK$

   a. Return acknowledgement or failure code to the payer *Alice*.

Coin external buffering:

1. *A* : Alice withdraws coin from her e-purse as per step 2 above, except $B_{i,v} = \boldsymbol{e}(C_{i,v}, P_A)$ in step 2b.
2. *A* : $B_{v,i}$ is now handled by external API and can be stored in any media for a required period, together with CRT_A (XML element "bufferedCoin")
3. *A* : Alice deposits coin back to her e-purse as per step 4 above, except $C_{i,v} = \boldsymbol{d}(B_{i,v}, S_A)$ in step 4a.

Some obvious informational internal functions are necessary for reporting the status of the e-purse, such as:

Total Balance: $\sum_{i=1}^{N} C_i$ where C_i is an e-coin of any value and N is the total number of e-coins

Coin Report: $\{\sum_{i=1}^{n1} C_{i,v1}, \sum_{i=1}^{n2} C_{i,v2}, ..., \sum_{i=1}^{nk} C_{i,vk}\}$ where $\{v_1, v_2, \ldots, v_k\}$ is the set of all possible coin values.

6. Security and Usability Considerations

6.1 Characteristics of presented electronic coins

The evaluation of an electronic payments scheme mainly depends on the satisfaction of specific characteristics from technological, usability and security points of view [21]. The optimum balance between usability, security, privacy, scalability, interoperability, cost and other nonfunctional characteristics will determine the acceptance of a payment system by the market. The proposed framework satisfies all the basic requirements at a very high level as it is demonstrated by the following presentation of its characteristics.

6.1.1 Usability characteristics:

P2P Transferability: The e-coin can be theoretically indefinitely transferable between e-purse users without the intervention of any authority and freely transmitted or stored in its encrypted form, in insecure means. There is no distinction between clients and merchants in terms of infrastructure, protocols and functionality. In order to reduce the exposure of an e-coin to possible cryptanalysis or other malicious treatment, it is invalidated either when its expiration date is reached or when its circulation counter reaches its physical or configured limit.

Openess and Interoperability: The protocol and the data structures used by the system are completely open. Standard PKI-enabled smart cards are used, data structures are XML-based, open and interoperable and APIs may be implemented in any platform. Existing applications may easily integrate an interface to use an e-purse, to store and process e-coins and to communicate with other e-coin users.

Applicability and Cost: The payment system is suitable for personal, merchant, bank and remote Internet-based usage. In any case the same applicability rules apply, being that a) the user possesses a smart card based e-purse with the relevant keys, b) the user is authorized as an e-coin user by possessing a proper digital certificate and c) the user has the necessary infrastructure. The infrastructure is simple, low-cost and consists of a common ISO-7816 smart card reader and a simple application program interface which may reside either on a client's PC, on a POS device or offered as an ASP (Application Service Provision) web-based application. A present average cost of a smart card with a crypto-processor and 32Kbytes of EEPROM [16], which can be indefinitely reused as an e-purse is €10, while a smart card reader with serial or USB interface costs approximately €25.

Ease of use: The coin exchange process is transparent for the end-user, who practically performs a transaction in three steps: a) identifies the payee through a public directory, by means of her digital certificate, b) authenticates to the smart card by entering PIN, c) selects amount to transfer. The preliminary registration phase is also transparent, since the e-coin user will receive a pre-personalized smart card with built-in application, keys and digital certificate. Re-personalization of e-purse (actually, a key renewal process) has to be made upon the expiration of certificates (e.g. every two years).

Efficiency: Efficiency considerations are particularly discussed in sections 6.2 and 6.3, focusing on the storage capacity of an e-purse and the times of response for various processes.

Scalability: Transactions are peer-to-peer and not dependant on any centralized system which could possibly create bottlenecks and therefore there is no limit on the number of both e-coin issuers and users. CSP related functions, such as directory queries for Certificate Status Information (CSI) [22] and certificates retrieval, are centralized and massive usage may occur in a large scale deployment. However this is not an important scalability concern, due to the low complexity and the sparse nature of the relevant directory queries.

Off-line usage: The proposed system operates in off-line mode and there is no need for an online centralized authority or application. A transaction can be completed on a peer-to-peer basis between two parties either remotely or physically present at the same location, without the need for any online third party. Online communication should be periodically available with the CSP for retrieving CSI data and with an e-coin distributor (e.g. a bank) for coin exchange/deposit/withdrawal.

Mobility: Since smart card based applications have the physical property of mobility, the e-coins stored in a personal e-purse can be used at any location where the necessary infrastructure is available.

6.1.2 Security characteristics:

Unforgeability: The authenticity of each e-coin is verifiable at any stage of its circulation, since it is digitally signed by its e-cash issuer. A forgery of an e-coin would mean that the signature creation data (i.e. private keys) or the infrastructure of the e-cash issuer is compromised. Such a case should be as unlikely as the compromise of the infrastructure of a central bank issuing regular money. But even if the unlikely happens, an efficient certificate revocation mechanism [22] would reestablish order in a short period with minimal consequences. Controlling also the version of the circulating e-coins is an additional mechanism against forgery. When the system suffers a serious compromise banks and merchants are upwards updating the minimum acceptable version of e-coins and thus, older version coins are quickly removed from the circulation.

Integrity: Any alteration in the content of the e-coin (in either its plain-text or encrypted form) would invalidate it immediately, since it would render the signature of the e-cash issuer invalid.

Unreusability (protection against double-spending): An e-coin is never exported by a smart card in plain text, but is only visible in its encrypted form, addressed to a sole payee. Thus, once it is withdrawn from the e-purse, it has value only for the payee and it is useless for any other party. An attempt by an adversary to replay an e-coin multiple times to the same intended payee, will fail due to the existence of the Coin Circulation Counter, as it is described in detail in section 4.7.

Anonymity and Untraceability: The e-coins stored in an e-purse do not include any information that connects them to any specific user [23]. Tracing money flows is impossible, since a specific coin cannot be correlated by an eavesdropper to one of the transaction parties, because the coin-ID and the coin-value are invisible during the transaction. As in regular cash and in all e-payment systems, the identification of the beneficiary is inevitable in order to initiate a transaction. Anonymity is assured for the payer, since according to the presented protocol she is not obliged to present any identification details.

Auditability: Audit information about the issuance and the circulation of e-coins may be collected at four levels: a) At the level of certification services where the procedures of registration and authorization of e-coin users and the issuance of digital certificates must be documented, controlled and auditable. b) At the e-cash issuer level who is responsible for keeping track of the issued and destructed coins. c) At the e-coin distributor level who may record transactions with authenticated users. According to the proposed scheme the distributor cannot record e-coin IDs but only value exchanges. d) At the e-coin user level who may record the values of her personal transactions, but not necessarily the identity of the peer party and definitely not coin IDs. Additionally, the logging of Circulation Counters by the e-cash issuer when an e-purse goes online can serve as a secondary fraud-protection mechanism, as described in section 4.7

6.2 Capacity considerations

Since smart cards are portable devices with limited EEPROM and ROM capacity, the applications based on them, must encounter this restriction. According to the indicative file structure of the smart card for the proposed application, the EEPROM must be able to store basic card information, keys, certificates, coins and circulation counters, while the built-in software must fit in the ROM of the smart card. The need for a large general purpose XML-parser is eliminated, since the internal functions themselves undertake to parse and built the specific XML structures described in section 5.1, using minimal dedicated code.

The basic card information, such as serial number, PINs, external keys and cardholder information, together with the cryptographic key pairs, do not occupy more than 1000 Bytes of EEPROM. A digital certificate for keys of 1024 bits, without long extensions, does not exceed the size of 3 KBytes.

A circulation counter for one coin consists of the coin ID plus a two-byte integer counter, 6 bytes in total. A circulation counter must be stored in the card for every distinct coin ever deposited in the card, presently and in the past and they can be deleted only after PKI keys renewal. Assume that an

active e-purse user performs 4 transactions per day with an average of 3 coins exchange per transaction, giving 8760 distinct coin deposits within a 2-year period (a life cycle of 2 years for a 1024 PKI key pair is common). Assuming that all 8760 e-coins are different and none of them is re-circulated, 8760 circulation counters must be stored and therefore 51 KBytes are needed for their storage. This structure can be further optimized by adding to each circulation counter the coin's expiration date in the form of a scaled, rounded-up byte-sized value. Entries for coins that have reached their expiration date can then be removed from the smart card's storage.

A usual EEPROM capacity for smart cards that are today in mass production by various vendors is 128 Kbytes, while smart cards of 512 KB or more EEPROM are already announced [24]. Considering all the abovementioned storage requirements, a 128 KB smart card will leave 73 KB free for storing electronic coins. An electronic coin in plain text in the previously presented XML form, but with fixed-length binary values only and without the XML tags has a maximum size of 22 Bytes plus the size of the e-cash issuer's digital signature (see XML-schema in Appendix). An RSA-SHA-1 digital signature produced by a 1024-bit key is 128 Bytes long [16]. Therefore each coin will occupy a fixed length of 150 Bytes. Accordingly, a 128 KB smart card will be able to store approximately 500 e-coins of various values, a quite satisfactory number for a personal e-purse. The usage of signature algorithms that produce smaller signature values [25] can further increase the number of stored e-coins. The smart card capacity requirements for the proposed application are summarized in [Table III].

Item stored	Size per record (bytes)	Number of records	Approximate total space required
Basic card information & cryptographic keys	1024	1	1KB
Digital certificate	3000	1	3KB
Coins	150	500	73KB
Circulation counters	6	8760	51KB
		Total:	**128KB**

Table III. Indicative smart card capacity requirements

At the side of a bank or a merchant, the e-purse capacity requirements are increased and since mobility is not required, the smart cards should be used in conjunction with external coin buffering as described in section 4.8, which practically removes any limit restrictions from the capacity of the application. Alternative solutions also exist, such as the secure emulation of smart cards by software [26] in which case the whole application resides on a regular computational system, without any measurable capacity restrictions.

6.3 Performance considerations

The basic factor that seriously affects the performance of a smart card–based application is the response of various cryptographic operations performed within the processor of the smart card. The described framework must not be considered as suitable for tick-based payments (e.g. phone-cards) since it is based on public key cryptographic functions executed within the crypto-processor of the smart card [27]. However the efficiency of the system is perfectly acceptable for automatic or online purchasing, based on the following measures for a typical PKI-enabled smart card [16]:

- Generation of a 1024 bit RSA key pair takes approximately 15 seconds.
- RSA signature or encryption of a 150-Byte e-coin is on the level of 1 second. Smart card implementations based on elliptic curves may reduce this time to tens of milliseconds [28].

- Storage and retrieval of e-coins and circulation counters have no measurable influence on performance, since they are stored in fixed-length EFs in the smart card (see section 4.3) which require minimal time for R/W operations.
- Based on an 115200-bps communication speed between the card and the reader, the transmission of a blind e-coin does not take more than 10 milliseconds.
- There is no online centralized communication that may affect performance.

Based on the above facts and since key generation is a one-time initial computation not affecting the overall performance, a complete transaction of withdrawing, transmitting and depositing an e-coin will last approximately 3 seconds (it includes three cryptographic operations: encryption, decryption and signature verification as described in section 5.2).

6.4 Threat scenarios

An electronic payment system based on the proposed framework is coping efficiently with most threats and attacks, either autonomously or by additionally establishing SSL sessions whenever necessary, as discussed in the following scenarios:

A "man in the middle" attack would only be attempted by an adversary who is duly authorized as an e-coin user and possess a valid e-purse and certificate. Transmission to any other entity that does not possess a valid e-purse certificate signed by the trusted authority will fail. The transmission of an e-coin to the intended party and not to an erroneous user depends on the awareness of the payer to check the identity of the payee as shown through a valid digital certificate.

Interference and replay: Further to the above case, the eavesdropper interferes in the communication of the two parties, withholds the blind coins exchanged and then attempts to send the coins to the same payee for his account, as a payer. In a critical transaction, the establishment of an SSL session with one party (payee) authentication will cope with this threat, still preserving the anonymity of the payer.

Traffic monitoring: An eavesdropper who monitors the handshaking communication between two transacting parties may recognize the identity of the payee and conclude the values of the coins exchanged. When the confidentiality of the information transmitted over open networks is considered as critical, it can be achieved by establishing an SSL session.

Coin reuse: An attempt to replay multiple copies of a blind e-coin to different receivers or to the same payee will fail, as described in section 4.7.

Gain access to plain text e-coin: Attempt to gain access to a plain text e-coin, which is then encrypted and sent to multiple recipients with means outside the smart card, is not possible, as long as smart card security is not compromised and cryptographic keys are generated and kept within the smart card.

Smart card emulation by external application: An adversary attempts to build an external application that simulates a smart card. The application can then receive e-coins, decrypt them and then reuse them as described in the previous case. This scenario is not possible, since the application has to possess a valid digital certificate from the CSP that authorizes the usage of cryptographic keys by the e-coin user. Such certificates are issued by the CSP only to card-generated non-exportable keys.

Smart card reverse engineering: An adversary attempts to reverse engineer the smart card by building its internal application in a way that satisfies the requirements of the API but has security trapdoors, such as the access to plain text e-coins or the possibility of spending an e-coin multiple times. The solution is the inspection of the smart card software integrity by secure mechanisms, such as the reflection-based technique proposed in [29] prior to the issuance of a digital certificate for the keys it contains. It is also required that the smart card authorized by the CSP as an e-purse, conforms to certain specifications in terms of structure, protocols and ROM capacity, excluding any proprietary devices. Essentially, the smart cards should be constructed and programmed by an authorized manufacturer in a secure environment, similar to the regular money typographic procedures.

Buffering exploitation: Buffering of e-coins in insecure storage (see section 4.8) such as local disks or cache memory is possible for increasing the capacity of the application or during network communication failures. Buffered e-coins have no risk of misuse since they remain encrypted and can only be processed by the intended beneficiary.

Smart card security compromised: This scenario is against one of the main assumptions of the proposed solution. If the smart card is compromised the attacker will have access to clear text e-coins, which she can then encrypt and spend them multiple times to different payees. These double spent coins may only be identified and invalidated after further circulation, only when a e-purse goes online and Circulation Counters are recorded. Further work on this subject could propose a mechanism such as the distribution of a Coin Invalidation List maintained by the e-cash issuer, similar to the CRL maintained by the CSP.

Compromise of CSP key: This is again an unlikely scenario, since the proposed solution is based on the principles of a PKI, where CSP's secret key protection is crucial. An attacker who gains access to the keys of the CSP could create certificates that authorize other entities as e-coin users or even as e-cash issuers. Efficient certificate revocation mechanisms [22] may mitigate the damage in such situations. The same applies to the case of e-cash issuer key compromise. It is obvious that the protection of the secret keys of the CSP and of the e-cash issuer has to be considered as crucial as the protection of the infrastructure of a traditional mint.

6.5 Summary of usability and security characteristics

The design of the proposed solution is simple and easy to implement, since it is based on known cryptographic algorithms and open data structures. Despite its simplicity, the proposed framework exhibits an overall advantage against other existing or proposed payment systems [5]. Examining the specific usability and security characteristics of our scheme, we conclude that the lack of value divisibility is counterbalanced by the decreased complexity, the increased security, mobility, portability, openness, offline usage, scalability and the almost unique properties of P2P transferability and origin verifiability, which are not addressed as a whole in other systems. Additionally, protection against double-spending is achieved in real-time in contrast to other schemes where the detection of a double spending is asynchronously detected. Finally, the proposed scheme fully supports anonymity, a characteristic that is not supported in schemes such as SET or credit cards. In Table IV we present a summary of how these characteristics are achieved in our solution and how the same characteristics are implemented in regular money.

Characteristic of e-coin	How it is achieved in our framework	Relevant characteristic of regular cash
Verifiable origin and unforgeability	- Distinct authorized issuer (section 4.5) - Authenticity of digitally signed data structure (section 5.1)	Authenticity by means of material secure characteristics
Anonymity	- E-coins are anonymous (section 5.1) - Payer remains anonymous within a transaction (section 5.2)	Regular coins are anonymous. Payee is normally identifiable.
Untraceability	- E-coin IDs cannot be seen and logged (section 5.2)	Serial numbers of notes though, can be recorded
Non-divisibility	- E-coins are signed data structures that cannot be modified (section 5.1)	Distinct material objects
Mobility	- Smart card used as an e-purse (section 4.2)	Small size of material
Scalability	- No centralized transactions (section 5.2) - Storage capacity extended by external coin buffering (section 4.8)	Only physical constrains of production, storage and transfer

P2P Transferability	- Exchange of encrypted, non-modifiable, verifiable data (section 5.2) - No intermediate entities and no distinction in clients and merchants (section 5.2)	Physical objects exchange
Unreusability	- Coin Circulation Counters and (section 4.7) - Smart card security (section 3.3)	Non-duplicable distinct physical objects
Offline	- PKI enabled offline trust model (section 4.4)	No reference to authority while circulating
Life-cycle	- Coin expiration to reduce exposure to cryptanalysis (section 5.2)	Material physical deterioration and destruction
Openness	- Open protocols (XML & Public Key cryptography – sections 5.1, 5.2) - Common hardware (Smart cards and readers – section 3.3)	No infrastructure needed on the user's side

Table IV. Characteristics of blind e-coins

7. Conclusions

The electronic payments framework presented in this paper provides the basis for implementing a strong off-line electronic alternative to conventional coins and notes as well as to other online or closed electronic cash systems. The proposed scheme satisfies important security and usability requirements by exploiting the security features of PKI and smart card technologies. The PKI services provide the means for building trust and authorizing the actors of the scheme, while the smart cards provide a tamper-proof, mobile and cheap device for storing and processing the sensitive data of the application.

A trust model and an abstract functional model are proposed. The trust model is implemented by a series of preliminary procedures such as the key generation, the digital certificate issuance and the smart card personalization. The functional model is based on the separation of internal (protected within the smart card) and external (on the API level) procedures, which are then used to build a simple and secure transaction protocol. The data structures used are based on an XML-schema and therefore they are open and portable.

Main usability requirements, such as offline usage, peer-to-peer transferability, efficiency, openness, scalability and mobility are adequately satisfied. Various security and privacy characteristics, such as e-coin integrity, authenticity, unforgeability, unreusability, untraceability and anonymity are also satisfied. The proposed scheme proved to be resistant to various threats and attacks against its security and reliability. The capacity of a typical smart card is quite satisfactory for a personal e-purse, while external e-coin buffering is possible for applications requiring extended storage capacity.

Further work regarding the proposed framework may focus on the design of a Coin Invalidation List mechanism, similar to the Certificate Revocation List, coping with a smart card compromise and providing status information about the circulating e-coins. Another topic is the enhancement of the efficiency of an e-purse to make it suitable for tick-based micro-payments.

References

[1] D. Birch, New life for the electronic purse?, Card Technology Today, 14 (7–8) (2002) 10.

[2] R.L. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM 21 (2) (1978) 120–126.

[3] C. Lambrinoudakis, Smart Card Technology for Deploying a Secure Information Management Framework, Information Management and Computer Security 8 (4) (2000) 173–183.

[4] J. Cowan and R. Tobin, XML Information Set. W3C Recommendations (2001) available at: http://www.w3.org/TR/xml-infoset.

[5] W. Jung, A. Kirby, R. Kolluri, K. Shannon and Y. Yoon, A Comparison Of Electronic Cash Schemes and Their Implementations, ECE 646 AS-3 Group Project, George Mason University, Virginia, USA, 2001

[6] S.A. Brands, Electronic Cash on the Internet, in: Proc. Symposium on Network and Distributed Systems Security, Eds. (IEEE Computer Society, 1995) 64–84.

[7] D. Chaum, A. Fiat and M. Naor, Untraceable electronic cash, in: Proc. Crypto'88 — Advances in Cryptology, Santa Barbara, California, Lecture Notes in Computer Science vol. 403 (Springer, Berlin, 1990) 319–327.

[8] V. Varadharajan, K.O. Nguyen and Y. Mu, On the design of efficient RSA-based off-line electronic cash schemes, Theoretical Computer Science 226 (1999) 173–184.

[9] G. Medvinsky, B.C. Neuman, NetCash: A design for practical electronic currency on the Internet, in: Proc. 1st Annual ACM conference on Computer and Communications Security, (ACM, 1993) 102–106.

[10] I. Mavridis, G. Pangalos, T. Koukouvinos and S. Muftic, A Secure Payment System for Electronic Commerce, in: Proc. DEXA'99 Workshop, Eds. IEEE Computer Society (Florence, Italy, 1999) 832–836.

[11] Mondex International, Introduction to Mondex and electronic cash, Document reference: tec-gui-760 v.2-0, 1999.

[12] CEPS, Common Electronic Purse Specifications, CEPS Technical specification, Version 2.3 (2001) available at: http://www.cepsco.com

[13] T. Okamoto, An efficient divisible electronic cash scheme, in: Proc. CRYPTO'95 — Advances in Cryptology, Lecture Notes in Computer Science vol. 963, (Springer, Berlin, 1995) 438–451.

[14] S. Brands, Off-line cash transfer by smart cards, in: Proc. First smart card research and advanced application conference (Lille, France, 1994) 101–117.

[15] ITU X-509 | ISO/IEC 9594-8. The directory: Public-key and attribute certificate frameworks. International Telecommunication Union X-Series (2001).

[16] Schlumberger, Cryptoflex Programmer's Guide, Release 3C, 2000.

[17] D. Lekkas, Establishing and managing trust within the Public Key Infrastructure, Computer Communications 26 (16) (2003) 1815–1825.

[18] P. Jones, Time for non-banks to take over the electronic purse, Card Technology Today 12 (3) (2000) 2–3.

[19] D. Chaum, T.P. Pedersen, Wallet Databases with Observers, in: Proc. CRYPTO'86, Advances in Cryptology, (Springer, Berlin, 1986) 192–211.

[20] A. Rensburg and B. Solms, A comparison of schemes for Certification Authorities / Trusted Third Parties, in: L. Yngström and J. Carlsen Eds, Proc. SEC'97 IFIP-TC11 13th International Conference on Information Security, (Copenhagen, Denmark, 1997) 222–240.

[21] D. Abrazhevich, Classification and Characteristics of Electronic Payment Systems, in: Proc. Electronic Commerce and Web Technologies, 2nd International Conference, EC-Web'01, (Munich, Germany, 2001) 81–90.

[22] J. Iliadis, S. Gritzalis, D. Spinellis, D. de Cock, B. Preneel and D. Gritzalis, Towards a framework for evaluating certificate status information mechanisms, Computer Communications 26 (16) (2003) 1839–1850.

[23] Y. Tsiounis, Efficient electronic cash: New notions and techniques, Ph.D. Thesis, Northeastern University, Boston, Massachusetts, 1997.

[24] Samsung Electronics, Samsung electronics mass produces world's first smart card chip with onboard 512KB EEPROM. Press Release, 2004.

[25] F. Zhang, R. Safavi-Naini and W. Susilo, Efficient verifiably encrypted signature and partially blind signature from Bilinear Pairings, in: Proc. INFOCRYPT'03, Lecture Notes in Computer Science vol. 2904 (Springer, Berlin, 2003) 191–204.

[26] D.N. Hoover and B.N. Kausik, Software smart-cards via Cryptographic Camouflage, in: Proc. IEEE Symposium on Security & Privacy (Oakland, USA, 1999) 208–215.

[27] T. Pedersen, Electronic Payments of Small Amounts, in: Proc. Security Protocols International Workshop, Eds. Springer, (Cambridge, UK, 1997) 59–68.

[28] K. Nguyen, Elliptic Curve Cryptography — The state of the art in smart card environments, in: Proc. ECC'02 6th Workshop on Elliptic Curve Cryptography (Essen, Germany, 2002)

[29] D. Spinellis, Reflection as a mechanism for software integrity verification, ACM Transactions on Information and System Security 3 (1) (2000) 51–62.