A Solution for Wireless Privacy and Payments based on E-cash

A. Karygiannis* Aggelos Kiayias[1]† Yiannis Tsiounis[1]‡

Abstract

The IEEE 802.11 Wireless Local Area Network (WLAN) specifications have been the subject of increased attention due to their rapid commercial adaptation and the introduction of new security and privacy concerns. The IEEE 802.1x standard was introduced in order to overcome the initial security shortcomings of the Wired Equivalent Privacy (WEP) protocol. The IEEE 802.1x standard is an extensible standard that couples 802.11 networks with various authentication services through the incorporation of an Extensible Authentication Protocol (EAP) authentication dialog. The existing implementations of EAP dialogs are based on standard cryptographic solutions for authentication and session key generation but do not, however, provide any form of user anonymity or privacy. Anonymity and privacy are currently of pressing interest, especially in the context of WLANs, which are simultaneously the best medium to provide privacy (there is no physical phone number or connection end-point with a predetermined owner) as well as the most threatening medium to user privacy, as they have the potential of disclosing not only the identity of the user, but also their physical location. At the same time, the potential "perfect hiding" capabilities of WLAN users also highlights the need to control anonymity by introducing more flexible authentication mechanisms. Moreover, payment for wireless services is completely decoupled from the above procedures, raising additional efficiency and privacy concerns. In this work we propose a new EAP authentication dialog based on anonymous electronic cash that provides for privacy, anonymity control, payment acceptance and billing, and authentication. Our solution is based on the notion of "public-key embedding e-cash," an e-cash variant we present and formalize in this paper. We present a concrete description of the new EAP authentication dialog in the context of IEEE 802.1x. We also present an efficient implementation of a public-key embedding e-cash scheme based on RSA blind signatures and prove its security.

1 Introduction

WLANs [10] have recently begun proliferating in the marketplace. It is now common practice for most corporations to employ at least one WLAN, while in the consumer market a large portion of laptops and handheld devices are equipped with built-in 802.11-compatible wireless cards. At the same time, service providers have begun deploying wireless access points in public areas. In short, 802.11-based WLANs are in a similar evolutionary state to where the Internet itself was about a decade ago, and they are bound to expand and become the norm in the future – much like cell phones have proliferated and have become the normal form of communication for mobile users.

Under this perspective it becomes clear that a large percentage of WLAN deployments will depend on publicly accessible access points offered by service providers – much in the way cellular phones depend on a similar network. In this setting, anonymity is an important consideration, as in addition to the user's identity, the user's physical location is also disclosed to the service provider or to a sub-contractor of the service provider. Disclosure of the user's location is an important privacy issue: it creates unnecessary liability for the service provider and is also a turn-off for many consumers. On the other hand, a WLAN that can guarantee the anonymity of individuals is a potential medium for the "perfect crime" [11, 17], as the identity as well as the location of the user are hidden. Thus, an actual WLAN implementation would benefit from a controlled privacy solution, i.e., by providing privacy to individuals, but maintaining the ability to revoke that privacy when required by law.

An efficient solution for providing anonymity, however, is paramount, as it is unrealistic to ask service providers to alter or seriously redesign their systems for this incremental feature. Fortunately, in contrast to the current cellular network, there are two major differences which make the goal of providing anonymity in WLANs

*NIST, USA, karygiannis@nist.gov

†CSE Department, University of Connecticut, Storrs, CT, USA, aggelos@cse.uconn.edu

‡Etolian Capital, LP, New York, USA, yiannis@etolian.com

[1] Research supported in part by NIST under SBIR grant SB1341-02-W-1113

feasible: first, cell phones are required to receive connections, whereas in 802.11 WLANs the client always initiates the connection; and second, WLAN authentication protocols are based on open standards and are, therefore, more easily extendable.

The security of 802.11 WLANs, and particularly 802.11b WLANs, has been subjected to severe criticism [4]. On one hand, defects in the underlying encryption mechanism of 802.11b have been widely reported; on the other the standard does not provide an effective access-control/session-key-exchange mechanism. The encryption defects are being addressed in interim and future versions of the 802.11 standard (WPA, WPA2, and RSN), while the authentication deficiency can be addressed by employing the IEEE 802.1x standard, a standard that provides an extensible framework for authentication through the employment of an "EAP authentication dialog." We note that none of the existing solutions that deals effectively with authentication and access control can provide user anonymity against the service providers. It is clear that the employment of WLANs is currently struggling with problems such as user anonymity, data privacy, service availability, and intrusion detection and it becomes more and more apparent that user privacy and anonymity will be sacrificed in favor of other desirable properties such as effective access control and payment for services.

All the above suggest that a successful solution for WLAN privacy, access control and payment for services cannot be piecemeal. Motivated by this, we hereby propose to use a special kind of anonymous e-cash to address all these issues in a single shot. Our solution provides privacy in an efficient and modular way. In addition, it provides more flexibility than current solutions to the orthogonal problem of charging for the actual service. In other words, the ability to use e-cash as the authentication mechanism allows billing and access control to occur simultaneously with (i.e., a customer is billed implicitly by providing a valid e-coin to access the network). This methodology has several advantages over the current billing structures in cellular networks: (a) simplicity (no need to interconnect billing with access control), (b) flexibility (modular architecture allows implementation in existing WLANs), (c) consistent billing (it is much more difficult to circumvent payment, since it is now integral to access control), (d) security (provided "for free" by the e-cash protocol) and, of course, (e) billing privacy (i.e., the ability – but *not the requirement* – to hide one's transactions). We note that using electronic coins for 802.1x authentication has been proposed before [3] using micropayments, but without anonymity. The solution we propose here is as efficient as a micropayment system at payment time, while providing for withdrawals of exact amounts, and of course anonymity.

In this work we formalize the requirements for the most suitable e-cash systems for WLAN authentication. This leads to a variant of e-cash called "public-key embedding e-cash." In a nutshell, public-key embedding e-cash allows the generation of digital coins that incorporate a public-key; this key transforms the coin into a "certification token" that is capable of allowing digital coin-based authentication and key-exchange. Based on this primitive we present a general, albeit efficient, way for adding e-cash authentication into existing WLAN systems. Our solution is in the form of an EAP authentication dialog and is presented within the general framework of the extensible IEEE 802.1x standard (thus, it can be readily incorporated into existing systems). In addition, we show a generic way of using traceable ("fair") e-cash for WLAN authentication, thus providing controlled anonymity to the end system.

We remark that the public-key embedding e-cash idea can be seen as a natural extension of Chaum's RSA-based blind signature and has been implicit in previous works, particularly, Chaum [8], Jakobsson and Yung [13] as well as in Jakobsson and Juels's X-Cash system [12].

We also provide a construction of an efficient public-key embedding e-cash scheme. For this instantiation we utilize a divisible low-exponent RSA-based e-cash scheme that is based on the scheme of Chaum [6, 7]. RSA-based blind signatures [6] have been used before for signing tokens, either in the context of off-line divisibility [15] or in the context of credentials [8].

By tuning RSA to be on the low-exponent extreme, we effectively provide a system with very fast signature verification, which at the same time can be proven secure. In addition, by using more than one exponent we can easily provide a provable multi-denomination system with similar efficiency. We give a formal security analysis and isolate a concise intractability assumption that allows the formalization of the security of our scheme. Our intractability assumption is a natural multi-exponent extension of the "RSA Inversion Oracle Assumptions" that were introduced by Bellare et al. [2].

Organization: We begin by reviewing the model of anonymous electronic cash and presenting the model for public-key embedding e-cash in section 2. Section 3 provides an overview of the current authentication framework in IEEE 802.11 WLANs. Section 4 proceeds to discuss a generic way to incorporate public-key embedding e-cash into WLAN authentication and provides concrete instantiations of efficient low-exponent RSA-based e-cash for that purpose. Section 5 concludes the paper.

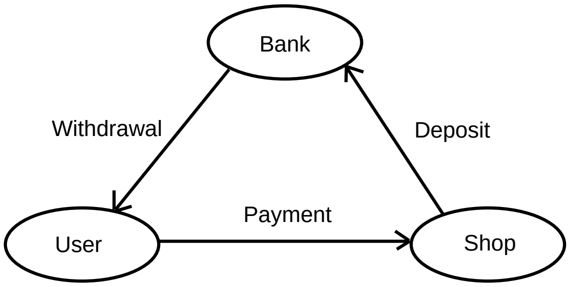

Figure 1: Model of electronic cash.

2 Electronic Cash

An anonymous electronic cash (e-cash) system consists of a collection of probabilistic, polynomially-bounded parties, a bank $\mathcal{B}$, users $\mathcal{U}$, and shops $\mathcal{S}$ and three main procedures: withdrawal, payment and deposit (see Figure 1). Users and shops maintain an account with the bank, while

- $\mathcal{U}$ withdraws *electronic coins* from his account, by performing a *withdrawal protocol* $\mathsf{WITHDRAW}$ with bank $\mathcal{B}$ over an authenticated channel,
- $\mathcal{U}$ spends a coin by participating in a *payment protocol* PAY with a shop $\mathcal{S}$ over an *anonymous channel*, and
- $\mathcal{S}$ performs a *deposit protocol* $\mathsf{DEPOSIT}$ with the bank $\mathcal{B}$, to deposit the user's coin into his account.

An e-cash scheme will be specified by the definitions of the procedures $\langle \mathsf{SETUP}, \mathsf{WITHDRAW}, \mathsf{PAY}, \mathsf{DEPOSIT} \rangle$. The system is *off-line* if during payment the shop $\mathcal{S}$ does not communicate with the bank $\mathcal{B}$, otherwise it is *on-line*. It is *anonymous* if the bank $\mathcal{B}$, in collaboration with the shop $\mathcal{S}$, cannot trace the coin to the user. A *multi-denomination* e-cash scheme supports multiple denominations at withdrawal.

Note that, in the absence of tamper-proof hardware, electronic coins can be copied and spent multiple times by the user $\mathcal{U}$. This has been traditionally referred to as *double-spending*. In anonymous on-line e-cash, double-spending is prevented by having the bank check if the coin has been deposited before. In off-line anonymous e-cash, however, this solution is not possible; instead, as proposed by Chaum et. al. [9], the system guarantees that if a coin is double-spent the user's identity is revealed with overwhelming probability.

2.1 Public-key embedding e-cash

In this section we formalize a variant of electronic cash, called "Public-key embedding e-cash" that is based on the work of Chaum from [8] and of Jakobsson and Yung [13]. Public-key embedding e-cash has the following twist: every coin has a public-key embedded in it with the corresponding secret-key only available to the owner of the coin. In our work the emphasis is on the fact that the e-coin can be used as a certification token that *binds* a coin to a secure session (through the public-key embedding) and as such can be used as the basis of our wireless authentication protocol.

More specifically, in a public-key embedding e-cash scheme the user employs a key-generation algorithm GEN during the $\mathsf{WITHDRAW}$ protocol, in order to create a key-pair $\langle \mathsf{pk}_C, \mathsf{sk}_C \rangle$. We remark that the public-key pk_C is not transmitted to the bank but it is nevertheless bound to the coin that is produced by the protocol. During the PAY protocol the user transmits the tuple $\langle i, \mathsf{pk}_C, C \rangle$ where pk_C is the public-key embedded into the coin C as well as a proof that the public-key pk_C is well formed. The coin verification algorithm executed by the shop $\mathcal{S}$ and the bank $\mathcal{B}$ also verifies whether pk_C is the embedded public-key as well as whether it is well formed. This key transforms a coin to a **certification token** that certifies a public-key that can be used to perform coin-related (i.e., coin-bound) authentication and session key generation. Formally, a multi-denomination public-key embedding e-cash scheme is a collection of protocols and algorithms, $\langle \mathsf{SETUP}, \mathsf{WITHDRAW}, \mathsf{PAY}, \mathsf{DEPOSIT} \rangle$ with the following specifications:

1. SETUP is an algorithm executed by the bank $\mathcal{B}$ so that $\mathsf{SETUP}(1^n, 1^k)$ outputs a public-key $e(k)$, where k is the number of different denominations that the bank employs and n is a security parameter.

2. $\mathsf{WITHDRAW}$ is a protocol executed by the bank $\mathcal{B}$ and the user $\mathcal{U}$. It is assumed that the user $\mathcal{U}$ creates and maintains an account with the bank $\mathcal{B}$ ahead of time. User $\mathcal{U}$ initiates the $\mathsf{WITHDRAW}$ protocol and authenticates to the bank $\mathcal{B}$. $\mathcal{U}$ also submits the denomination $i \in \{1, \ldots, k\}$ that he wishes to withdraw from his account. $\mathcal{B}$ removes the amount of funds that correspond to the denomination i from $\mathcal{U}$'s account upon verifying the correct construction of $\mathcal{U}$'s coin. The output to $\mathcal{U}$ is the electronic coin C into which a public key pk_C is embedded, and the secret key sk_C corresponding to pk_C.

3. PAY is a protocol between user $\mathcal{U}$ and a shop $\mathcal{S}$. $\mathcal{U}$ transmits to $\mathcal{S}$ the tuple $\langle i, \mathsf{pk}_C, C \rangle$ (also called the payment transcript) as well as a proof that pk_C is well formed. The shop $\mathcal{S}$ uses the public-key $e(k)$ to verify the validity of the payment transcript.

4. DEPOSIT is protocol between the shop $\mathcal{S}$ and the bank $\mathcal{B}$. $\mathcal{S}$ transmits payment transcript $\langle i, \mathsf{pk}_C, C\rangle$; the bank verifies its validity again and checks whether the tuple $\langle i, \mathsf{pk}_C, C\rangle$ has been deposited before by looking it up in its database of deposits. If the payment transcript has never been deposited before, $\mathcal{B}$ proceeds to pay the shop $\mathcal{S}$ with the funds that correspond to the i-th denomination and enters $\langle i, \mathsf{pk}_C, C\rangle$ into the database.

   On-line/Off-line. In the case of on-line e-cash the DEPOSIT protocol is executed by $\mathcal{S}$ prior to accepting in the PAY protocol. Furthermore in the off-line case two payment deposits that use the same coin C reveal the identity of the owner of the coin.

In addition to the above, in practice it is useful (if not mandatory in most cases) that coins can be "divided" and a user can spend an arbitrary amount from a coin. In e-cash schemes that allow such divisibility a payment request also includes the amount of funds from a certain coin the user wishes to use. We remark that the DEPOSIT protocol is modified accordingly so that the bank allows a coin to be deposited more than once until all of its funds are used. Coin divisibility is a straightforward application of public-key embedding e-cash schemes; we omit the details here and we refer to our main protocol in section 4 that employs divisibility. We remark that off-line e-cash does not easily couple with divisibility, i.e., divisible off-line e-cash requires in general much more complex constructions compared to the on-line case, e.g., [14, 5].

2.2 Security

Security of e-cash schemes is defined in terms of four requirements: *Anonymity/Unlinkability* (of the user and the user's payments respectively), *Unreusability* (i.e., prevention of double-spending or identification of double-spenders), *Unforgeability* (of electronic coins) and *Unexpandability* (N withdrawn coins cannot be expanded to $N + 1$ valid coins). Formal definitions of these notions have been studied and used before (see, e.g., [16]) and due to lack of space we omit a formal treatment in this extended abstract. With respect to the public-key embedding, we require the following property:
Embedding Robustness: Let W be a withdrawal transcript for a coin C, that was executed for the embedded public-key $\langle \mathsf{pk}_C, \mathsf{sk}_C\rangle$. Then, any payment transcript based on C that is accepted must include pk_C and a proof of well-formedness of pk_C as well.

Informally the above property ensures that the embedding of a coin cannot be altered after the initial execution of the protocol (essentially the coin acts as a secure signature on the embedded public-key).

3 WLAN IEEE 802.11 and Access Control

First we briefly overview the IEEE 802.11 "infrastructure" architecture. The system is subdivided into cells, where each cell is called a Basic Service Set (BSS) and is controlled by a base station called the Access Point (AP). Access points play the role of a wired hub in regular networks with physical connectivity. The AP is the intermediate between a mobile device and a physical (wired) network.

We call a workstation with wireless connectivity "wireless STAtion" (*STA*). It is very critical in the wireless setting to provide authentication services for the purpose of authorizing mobile devices to channel traffic through an access point.

In general, access control in a wireless 802.11 network can be managed (admittedly not very effectively) by employing the following basic mechanisms:

- SSID (Service Set Identifier). Every AP possesses a SSID. A common network configuration requires various STAs to know the SSID of the AP they attempt to connect. The level of security allowed by SSID is practically non-existent (as the AP broadcasts its SSID in intervals).

- MAC Filtering. Every device in a 802.11 network possesses a MAC (Medium Access Control) address. This is a 48-bit value assigned to the WLAN card by the manufacturer, although some devices allow their MAC address to be changed by software. If an AP is capable of checking the MAC address of an STA this provides an additional layer of access control, since it would be possible to block certain MAC addresses or allow access to a certain list of MAC addresses. The downside of this approach is that the AP must maintain a listing of the MAC addresses that are allowed to channel their traffic and lookup this list whenever a packet is received. Such bookkeeping can prove to be a cumbersome task for a large or a frequently changing WLAN. Moreover for WLAN cards with fixed MACs this approach leaves no room for user privacy.

- Static WEP Keys. WEP (Wired Equivalent Privacy) suggests that the AP and the STAs share a small set of keys (4 keys in the implementation). This allows

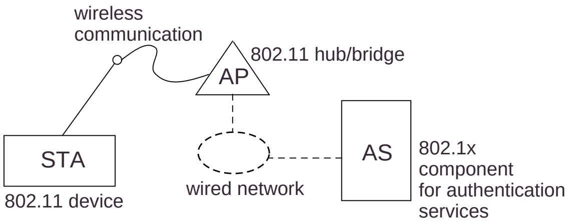

Figure 2: The 802.11 infrastructure for authentication.

wireless communication between STAs and AP to be encrypted. We remark that for decryption all four keys are tried, until the one that allows successful decryption is discovered — this allows the arbitrary choice of the encryption key (among the four possible choices).

In the IEEE 802.11 standard one employs WEP by instructing the MAC (medium access control) layer of the standard to enter the WEP mode that encrypts the communication with the RC4 stream cipher. Naturally this approach requires shared keys between the STA and the AP. Such keys must be changed frequently, and the 802.11 standard does not allow an effective method to change the shared keys. For this reason the WEP approach is not used or if employed in most cases the keys are left unchanged for long periods of time something that compromises security.

It is apparent from the above that access control in 802.11 networks is not properly addressed. To cover this insufficiency another standard can be used in conjunction to 802.11, in particular the IEEE 802.1x standard. This standard provides an effective framework for authentication and key update. In 802.1x an authentication server *(AS)* is accessible through a wired network from the AP. This infrastructure is presented in figure 2. We remark that the AS is part of an AAA (Authentication, Authorization and Accounting) Server. The AAA server may run the RADIUS protocol (Remote Authentication Dial-in User Service). We describe the authentication method of the 802.1x standard in the next section.

3.1 WLAN Authentication Overview — IEEE 802.1x

Mutual authentication of the STA and the AP is crucial for the successful employment of any wireless network, regardless of the underlying line-encryption protocol utilized.

The standard 802.1x is an authentication dialog between the system that needs to access the network and the network. This dialog is using the IETF Extensible Authentication Protocol (EAP). The principal component of the network is the Network Access Port, which is controlled by the Port Access Entity (PAE) which manages what packets will be accepted.

Systems that support PAEs are the “supplicants” (the STAs in the WLAN infrastructure) and “authenticators” (the APs in the WLAN infrastructure). Controlled ports accept packets only from authenticated systems, whereas uncontrolled ports accept packets from anywhere. The latter type, naturally, is used only for authentication purposes. These packets for 802.1x are the EAPOL packets (EAP Over LAN packets).

Authentication is accomplished by the communication of the supplicant (the STA) with the Authentication Server (AS), while the authenticator (the AP) plays the role of a proxy in this communication (forwards EAPOL packets submitted by STA over to AS, and it forwards all AS EAP packets back to the supplicant). We review the basic authentication steps below.

1. The STA sends an EAP-start message to the AP.
2. The AP requests an identity from the STA using EAPOL.
3. STA forwards its identity to AP via EAPOL.
4. AP forwards the STA identity to the AS via EAP.
5. AS and STA have an “EAP authentication dialog.” There are several such methods already standardized, see below.
6. If the dialog terminates successfully, both parties share a common key (this is optional, but highly recommended).
7. AS communicates to the AP (perhaps using RADIUS) whether it accepts or rejects the STA and if it accepts it also communicates the session key.
8. The AP enables its controlled port for the newly authenticated MAC addess. The authenticator (AP) and the supplicant (STA) can now employ the shared key to perform encrypted authenticated “per-packet” communication.

We remark that 802.1x is not the complete solution for authentication; this is so because 802.1x does not specify the exact method that is used in the EAP authentication dialog step. Some known types of EAP authentication dialogs are listed below.

- The EAP-TLS method, which provides mutual authentication between the AS and the supplicant (the STA in the WLAN setting). It is based on digital certificates as specified in X.509. This suggests that the STA must have a certificate which can be verified by the AS and the AS must have a certificate that can be verified by the STA. The main concern with this approach is efficiency of the public key operations, as well as the fact that a Certification Authority is required. Furthermore standard criticisms against X.509 also apply.

- The EAP-SRP method, which employs standard "User ID and Password" based authentication. In a variant called EAP-MD5, the authenticator sends a challenge to the supplicant: a random string along with some serial number. The supplicant appends its password to the challenge and hashes the result using the MD5 algorithm. Then it proceeds to submit the hash to the authenticator. There are a few problems with this approach, the most important being that only the supplicant is authenticated and not the authenticator; further this process does not create a shared key.

4 The E-Cash based Authentication Dialog - EAP-E-Coin

In this section we describe a novel method, standardized in terms of 802.1x, for EAP authentication based on public-key embedding e-cash.

In a nutshell, in the new EAP authentication dialog that we introduce, the STA selects a digital coin and submits this coin, including the client's MAC address, AP address and the SSID, as well as a time-stamp to the AS. In our authentication dialog we take advantage of public-key embedding e-cash to create a coin-specific secure communication dialog and key-exchange. Our EAP dialog also creates a shared key at the end of the interaction.

Remark. While the employment of e-cash provides a layer of anonymity and unlinkability, this would be useless if the STA revealed itself automatically in the communication layer of the WLAN protocol by using the same MAC address all the time. Thus, we emphasize that the real MAC address should be reset by the STA software before each session (that is, if the wireless card provides such capability); this will be our working assumption for the description of our protocol. If this does not hold true (i.e., the wireless card lacks such MAC address resetting capability) one would have to rely on the AP to conceal the MAC address of the card and communicate to the AS a random string that will be used to handle all AP to AS communication (whereas the AP to STA communication will use the standard MAC address of the STA).

4.1 The EAP-E-Coin dialog: a generic description

Let $\langle \mathsf{SETUP}, \mathsf{WITHDRAW}, \mathsf{PAY}, \mathsf{DEPOSIT} \rangle$ be a public-key embedding e-cash scheme as described in section 2. We describe the authentication dialog below. We will denote by $\langle \mathsf{GEN}, \mathsf{ENC}, \mathsf{DEC}, \ \mathsf{SIG}, \mathsf{VER} \rangle$ a chosen ciphertext secure encryption and chosen message secure digital signature scheme.

- (Prior to the Authentication Dialog). We assume the existence of a bank $\mathcal{B}$ that executes the SETUP algorithm for a certain number of denominations. The corresponding public-keys are published and certified and we assume that they are available to the AS.

  The user of an STA obtains some digital coins by employing the withdrawal protocol $\mathsf{WITHDRAW}$ with the bank $\mathcal{B}$. We remark that such communication should be prior to using the WLAN or alternatively AP's can be instructed to allow $\mathsf{WITHDRAW}$ packets to pass through without authentication.

  Every coin has a public-key embedded in it that will be denoted by pk_C with corresponding secret-key sk_C (that is only known to the user of the STA). Without loss of generality we will assume that the coin embedded key-pair $\mathsf{pk}_C, \mathsf{sk}_C$ can also be used in conjunction with the scheme $\langle \mathsf{GEN}$, ENC, $\mathsf{DEC}, \mathsf{SIG}, \mathsf{VER} \rangle$ that is employed by the AS.

  On the other hand the AS uses GEN to create a public-key and a secret-key $\langle \mathsf{pk}_{AS}, \mathsf{sk}_{AS} \rangle \leftarrow \mathsf{GEN}(1^{u'})$ where u' is a security parameter.

- *Authentication Dialog Step 1.* The AS sends a timestamp used both for replay protection and for determining the start of the wireless session to the STA:

  $$\mathsf{SIG}(\mathsf{sk}_{AS}, [I, \mathsf{pk}_{AS}, time\_stamp])$$

  We note that I specifies the type of digital coin denominations that can be used for accessing through the AP from which the STA's authentication request has been directed. Formally, we will require that I is a description of a relation $\mathcal{R}$ such that $\mathcal{R}_I$ contains tuples of the form $\langle i, V \rangle$ where i is a digital cash denomination and V a valid division of the i-th denomination.

- *Authentication Dialog Step 2.* The STA forms the authentication message as follows:

$$\Big\langle \mathsf{SIG}(\mathsf{sk}_C, [i, \mathsf{pk}_C, C, V, \mathrm{MAC}, \mathrm{AP}_{address}, \mathrm{SSID}, time\_stamp]), \mathsf{ENC}(\mathsf{pk}_{AS}, session\_key_{STA})\Big\rangle$$

  where V is the amount to be spent (i.e., the payment required for logging in to the network), and $session\_key_{STA}$ is a random string, that will function as the part of the STA for the creation of the session key later on in the dialog.

- *Authentication Dialog Step 3.* The AS receives the authentication message, it checks whether $\langle i, V\rangle \in \mathcal{R}_I$, and verifies the validity of $\langle i, \mathsf{pk}_C, C\rangle$ as a payment transcript. By employing a "shop" module it submits the coin to the bank to receive the funds. In other words, the AS acts as a shop in the e-cash scheme that we utilize.

- *Authentication Dialog Step 4.* The AS, provided that the payment was accepted, submits the message

$$\Big\langle \mathsf{SIG}(\mathsf{sk}_{AS}, [\mathrm{access\_granted}, \mathrm{time\_frame}, \mathrm{MAC}, \mathrm{AP}_{address}, \mathrm{SSID}]), \mathsf{ENC}(\mathsf{pk}_C, session\_key_{AS}, session\_key)\Big\rangle$$

  where $session\_key = \mathcal{M}(session\_key_{AS}, session\_key_{STA})$ and $\mathcal{M}$ is a Hash-based message Authentication Code (*HMAC* [1]) with message body $session\_key_{STA}$ and key $session\_key_{AS}$. Note that the construction of the session key can be verified by the STA by reconstruction and that the session key is created using random data from both the STA and the AS.

  The STA verifies the signature and the session key generation, and the protocol terminates. The AP, as the intermediate of the STA and AS, parses the above message and adds the MAC address of the STA to the controlled port MAC address listing. Subsequently, the STA uses the controlled port to access the network services.

We present pictorially the phases of the EAP-E-Coin protocol in figure 3.

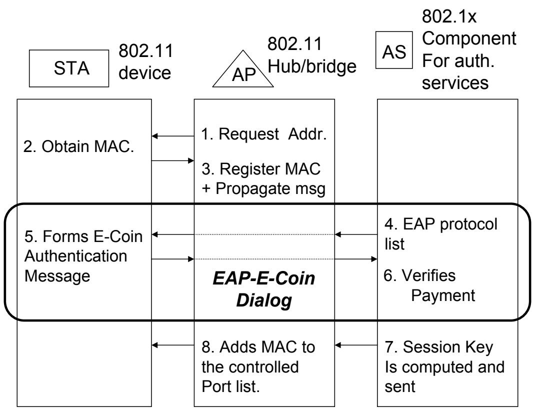

Figure 3: WLAN Authentication using EAP-E-Coin.

4.2 Authentication Server integration

A modular component that lets existing RADIUS servers accept and validate the e-coins used for wireless access can be marketed to the service providers. The component can act as a "shop payment" component, by contacting a central "bank" which verifies the e-coin(s) and instructs the RADIUS server to permit access.

4.3 Client component integration

The e-cash can be distributed to a client-component directly by the service provider, in a sign-up procedure where the client is charged to receive the e-cash. Charging can use any existing method – credit card, separate bill, etc. Customers may (optionally) be given the ability to refund their unused coins after some specified time period. The client component must integrate into STA in order to pass the appropriate request messages. Support for the major WLAN cards should be provided.

4.4 An Efficient design of a Public-key Embedding On-line e-cash Scheme

Mobile devices are typically restricted in terms of computational resources. For this reason our description of the EAP-E-coin authentication dialog would be incomplete without a concrete and efficient public-key embedding e-cash scheme that can be used in conjunction with it.

In this section we present a detailed description of such a scheme. The scheme is an extension of Chaum's blind-signature based e-cash. Its specification will be described in the procedures: $\langle \mathsf{SETUP}, \mathsf{WITHDRAW}, \mathsf{PAY}, \mathsf{DEPOSIT}\rangle$. The e-cash scheme we will present is on-line and it is public-key embedding. In our description we will use $\mathcal{H} : \{0,1\}^* \rightarrow \mathbf{Z}_N^*$ to be a publicly known hash function which can be modeled as a random oracle for the purpose of the security analysis.

SETUP. The bank $\mathcal{B}$ decides on a number of k different denominations and generates an RSA modulus N and the small distinct primes $e_1, e_2, \ldots, e_k$. The bank publishes $N, e_1, \ldots, e_k$, while it keeps the values $\phi(N)$ and $e_1^{-1} \bmod \phi(N), \ldots, e_k^{-1} \bmod \phi(N)$ secret. The bank also selects a public header μ for the coin generation such that $\mu \in \{0,1\}^l$, where $l \in \mathbb{N}$.

We will also assume that each user $\mathcal{U}$ employs an encryption scheme $\langle \mathsf{GEN}, \mathsf{ENC}, \mathsf{DEC} \rangle$ that allows an efficient *proof* of public-key well formedness[1]. This scheme will be used for the purpose of public-key embedding.

WITHDRAW. A user initializes an authenticated channel with the bank with which he maintains an account. Then, $\mathcal{U}$ initializes the digital signature scheme $\langle \mathsf{pk}, \mathsf{sk} \rangle \leftarrow \mathsf{GEN}(1^n)$. The user decides on the denomination $i \in \{1, \ldots, k\}$ he wants to withdraw, forms the "blinded" coin $B = \langle b^{e_i} \mathcal{H}(\mu, \mathsf{pk}) \pmod N, i \rangle$ where $b \in_U \mathbf{Z}_N^*$ and submits B to the bank through the authenticated channel. Observe that the public-key pk is embedded into the coin. The bank removes funds from the user's account that correspond to the denomination i and returns to the user the value $B' = B^{e_i^{-1} \bmod \phi(N)} \pmod N$. Subsequently the user computes the value $C = B'/b \pmod N$. Observe that $C = (\mathcal{H}(\mu, \mathsf{pk}))^{e_i^{-1}} \bmod N$.

PAY. A payment transcript would be of the form $\langle i, \mathsf{pk}, C \rangle$ followed by the proof of well formedness of the public-key pk. The shop verifies the payment by checking the equality $C^{e_i} \overset{?}{=} \mathcal{H}(\mu, \mathsf{pk})$ (note that the value e_i is public-key that corresponds to the i-th denomination published by the bank) and the proof of well-formedness of the public-key pk.

DEPOSIT. If the payment is verified the shop forwards the payment to the bank, which also verifies all equalities above. If the tests fail, the bank rejects the coin as invalid. If the tests succeed, the bank enters the payment transcript into the database and pays the shop as needed.

4.4.1 Security

In this section we establish that our scheme is a secure divisible anonymous e-cash scheme, Proofs of our claim will appear in the full version. Our intractability assumption is related to the RSA problem and is introduced below:

The Intractability Assumption. The intractability assumption that we employ for the security of our scheme is a **multi-exponent** generalization of the RSA-Chosen-Target-Inversion Problem that was defined by Bellare et al. [2] to formalize the security of Chaum's blind signature scheme. We formally state the problem below:

Multi-Exponent-RSA Chosen-Target Inversion problem: ME-RSA-CTI$[n, m]$

Input:	N and random target points $y_1, \ldots, y_{n+1} \in \mathbf{Z}_N^*$ $e_1, \ldots, e_z$ primes such that $\gcd(e_1 \ldots e_z, \phi(N)) = 1$.
Oracle:	Multi-Exponent-RSA inversion oracle: Given $\langle \ell, y \rangle$ with $\ell \in \{1, \ldots, z\}$, it replies by $y^{1/e_\ell} \pmod N$ Only m queries allowed
Find:	Index-set $J \subseteq \{1, \ldots, n\}$, $\lvert J \rvert = m + 1$, $\left\langle \langle \ell_j, y_j^{1/e_{\ell_j}} \pmod N \rangle \right\rangle_{j \in J}$ with $\ell_j \in \{1, \ldots, z\}$ for all $j \in J$.

We will assume that ME-RSA-CTI$[n, m]$ is hard when n is polynomially related to m. The relation of the ME-RSA-CTI problem to the unexpandability of our public-key embedding e-cash scheme (and in general to the unexpandability of Chaum's **multi-denomination** e-cash) is revealed in the following lemma:

Lemma 4.1 *Let $\mathcal{A}$ be a p.p.t. adversary that is allowed to (i) query the Random-Oracle $\mathcal{H}$ n times, (ii) invokes the withdrawal protocol m times, and returns $m + 1$ valid payment transcripts with non-negligible probability. Then the ME-RSA-CTI$[n, m]$ is solvable with non-negligible probability.*

As an immediate corollary to the lemma, we obtain the unexpandability of our public-key embedding e-cash scheme:

Theorem 4.2 *Under the* ME-RSA-CTI$[\mathsf{poly}(m), m]$ *the public-key embedding e-cash scheme we propose is m-unexpandable in the random-oracle model.*

Next we deal with anonymity. We show that it is satisfied in the information theoretic sense by our scheme:

Theorem 4.3 *Our public-key embedding e-cash scheme satisfies anonymity unconditionally.*

Finally observe that the embedding robustness property will follow easily from the fact that the withdrawal protocol is a blind-signature on the user's public-key.

4.5 Traceable E-Cash based Authentication Dialog - EAP-TE-Coin

In this section we consider the additional functionalities of the EAP-E-coin authentication dialog when the un-

[1] For example, any variant of ElGamal encryption would allow such proof in the form of a Schnorr proof of knowledge of the secret discrete-logarithm of the public-key.

derlying public-key embedding e-cash scheme is off-line and allows traceability (i.e., the scheme is a traceable (or "fair") e-cash scheme, see e.g., [11]). Such schemes have been designed in the past and their transformation to the public-key embedding setting can be accomplished in a similar fashion with the one we have demonstrated in section 4.4 (informally, by hashing a public-key into the coin). Fair e-cash schemes, in addition to offering anonymity, allow a third party (e.g., an auditor or a party designated by the service provider) to open the payment and reveal the actual identity of the user. This functionality could be important, for example, in a commercial or military setting, where mobile clients are normally anonymous but if misbehavior of the client is detected then a designated party (e.g., the chief security officer of a corporation or an officer of sufficient rank in the military) can obtain the true identity of a specific mobile client, or can trace the path of a mobile client. In particular fair off-line cash provides two tracing procedures called (i) *coin-tracing*: given the identity of a user it matches the payment requests that were submitted by this user, (ii) *owner-tracing* given a payment transcript (essentially a coin) it reveals the identity of the owner of a coin. We remark that these procedures can only be activated in extraordinary occasions and usually they can be executed only if a quorum of authorities consents.

The authentication dialog follows the paradigm of EAP-E-Coin as described in section 4 with the only difference being in the properties of the underlying e-cash scheme. For this reason we will refer to this authentication dialog as EAP-TE-Coin. In the remaining of this section we describe how the strong traceability properties of the e-cash scheme can augment the authentication functionality in the context of WLANs.

Enhanced Functionalities of the EAP-TE-Coin Authentication Dialog.

Off-Line Authentication. As the employed e-coin protocol in EAP-TE-Coin is off-line, it is not mandatory for the AS to contact the bank (issuer of the e-coins) in order to verify accessibility. Instead the AS batches each $payment\_transcript$ obtained in authentication step 1 of the dialog and submits them to the bank in an offline fashion. Users of STA's that have used an e-coin twice (in order to gain illegal access to the mobile network) are identified (due to the double-spending identification property of off-line e-cash). We remark that the frequency of this procedure can be calibrated (even in the course of the run-time operation of the system) according to the system's security management directives.

Note that if off-line authentication is not an issue, the scheme can be used as "on-line" (i.e., with on-line checking of the payment transcripts). This gives the advantage that e-coins can be spend in fractions (i.e., divisible cash) with an additional digital signature operation for each payment transcript (as we show in section 4.4). This improves the management of the authentication procedure as less coins are required to be withdrawn by the STA.

Revealing the source of an Authentication Dialog. If, for any reason, the source of an authentication dialog (i.e., the user of the STA) needs to be identified the Trustee can be contacted by the Bank (or the AS) and, by employing the owner-tracing protocol its identity can be revealed by submitting the $payment\_transcript$ to the Trustee. The AP can also be programmed to perform this operation in extraordinary cases.

Search whether a given source has initiated Authentication Dialogs. Recall that STA users perform the e-coin withdrawal protocol in an authenticated fashion, and withdrawal transcripts are stored by the Bank with the STA user's identity. The Bank can request from the Trustee to open a certain withdrawal transcript (using the coin-tracing protocol to reveal information about the e-coin that was withdrawn; subsequently the Bank can check whether the specific e-coin has been employed in an Authentication Dialog, simply by going through all payment transcripts).

5 Conclusion

In this paper we express the view that privacy, anonymity control, payment acceptance/billing and authentication are features that will become more and more important in the wireless LAN arena, especially in the main sector of the market occupied by IEEE 802.11 networks. The justification lies in the evident proliferation of 802.11 WLANs and their support by a growing number of service providers in publicly accessible Access Points.

Under this view, it becomes clear that an efficient implementation for all these features is beneficial for (and can provide a market advantage to) the WLAN service providers. We thus propose to use anonymous e-cash (or fair anonymous e-cash) to implement all these features in one shot. We proceed to define the requirements for e-cash schemes that can be used under this setting, called "public-key embedding e-cash", and to give a general framework for utilizing public-key embedding e-cash in WLAN authentication.

Our technical contributions also include a multi-denomination, divisible low exponent RSA-based on-line (public-key embedding) e-cash system that extends the scheme of Chaum [6] and its security analysis that is based on a RSA-derived intractability assumption, which in turn extends the work of Bellare et.al. [2]). The system is very efficient and can be applied to mobile battery-powered devices with small computational power.

References

[1] M. Bellare, R. Canetti, and H. Krawzcyk. Keying hash functions for message authentication. In N. Koblitz, editor, *Advances in Cryptology — Crypto '96, Proceedings (Lecture Notes in Computer Science 1109)*, pages 1–15, Santa Barbara, California, U.S.A., August 1996. Springer-Verlag.

[2] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The power of rsa inversion oracles and the security of chaum's rsa-based blind signature scheme. In *Financial Cryptography 2001*. LNCS, 2001.

[3] M. Blaze, J. Ioannidis, S. Ioannidis, A. Keromytis, P. Nkander, and V. Prevelakis. Tapi: Transactions for accessing public infrastructure. In *Proceedings of the 8th IFIP Personal Wireless Communications (PWC) Conference*, Venice, Italy, September 2003.

[4] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: The insecurity of 802.11. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (MOBICOM-01)*, pages 180–188, New York, July 16–21 2001. ACM Press.

[5] A. Chan, Y. Frankel, and Y. Tsiounis. Easy come–easy go divisible cash. In *Advances in Cryptology — Proceedings of Eurocrypt '98 (Lecture Notes in Computer Science)*, Elsinki, Finland, May 31–June 4 1998. Springer-Verlag. International patent pending. Available at http://www.ccs.neu.edu/home/yiannis/pubs.html.

[6] D. Chaum. Blind signatures for untraceable payments. In D. Chaum, R.L. Rivest, and A. T. Sherman, editors, *Advances in Cryptology. Proc. Crypto'82*, pages 199–203, Santa Barbara, 1983. Plenum Press N. Y.

[7] D. Chaum. Security without identification: transaction systems to make Big Brother obsolete. *Commun. ACM*, 28(10):1030–1044, October 1985.

[8] D. Chaum. Showing credentials without identification: Transfeering signatures between unconditionally unlinkable pseudonyms. In J. Seberry and J. Pieprzyk, editors, *Advances in Cryptology, Proc. of AUSCRYPT '90 (Lecture Notes in Computer Science 453)*, pages 246–264. Springer-Verlag, 1991.

[9] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Advances in Cryptology — Crypto '88 (Lecture Notes in Computer Science)*, pages 319–327. Springer-Verlag, 1990.

[10] IEEE Computer Society LAN MAN Standards Committee. *IEEE 802.11: Wireless LAN Medium Access Control and Physical Layer Specifications*, August 1999.

[11] G. Davida, Y. Frankel, Y. Tsiounis, and M. Yung. Anonymity control in e-cash. In *Proceedings of the 1st Financial Cryptography conference (Lecture Notes in Computer Science 1318)*, Anguilla, BWI, February 24-28 1997. Springer-Verlag. Available at http://www.ccs.neu.edu/home/yiannis/pubs.html.

[12] M. Jakobsson and A. Juels. X-cash: Executable digital cash. In *Proceedings of Financial Cryptography '98*, volume 1465, pages 16–27. Springer, LNCS, 1998.

[13] M. Jakobsson and M. Yung. Revokable and versatile electronic money (extended abstract). In *Proceedings of ACM-CCS 1996*, pages 76–87. ACM Press, 1996.

[14] T. Okamoto. An efficient divisible electronic cash scheme. In Don Coppersmith, editor, *Advances in Cryptology, Proc. of Crypto '95 (Lecture Notes in Computer Science 963)*, pages 438–451. Springer-Verlag, 1995. Santa Barbara, California, U.S.A., August 27–31.

[15] T. Okamoto and K. Ohta. Universal electronic cash. In *Advances in Cryptology — Crypto '91 (Lecture Notes in Computer Science)*, pages 324–337. Springer-Verlag, 1992.

[16] Y. Tsiounis. *Efficient Electronic Cash: New Notions and Techniques*. PhD thesis, College of Computer Science, Northeastern University, Boston, MA, 1997. Available at http://www.ccs.neu.edu/home/yiannis/pubs.html.

[17] B. von Solms and D. Naccache. On blind signatures and perfect crimes. *Computers and Security*, 11(6):581–583, October 1992.