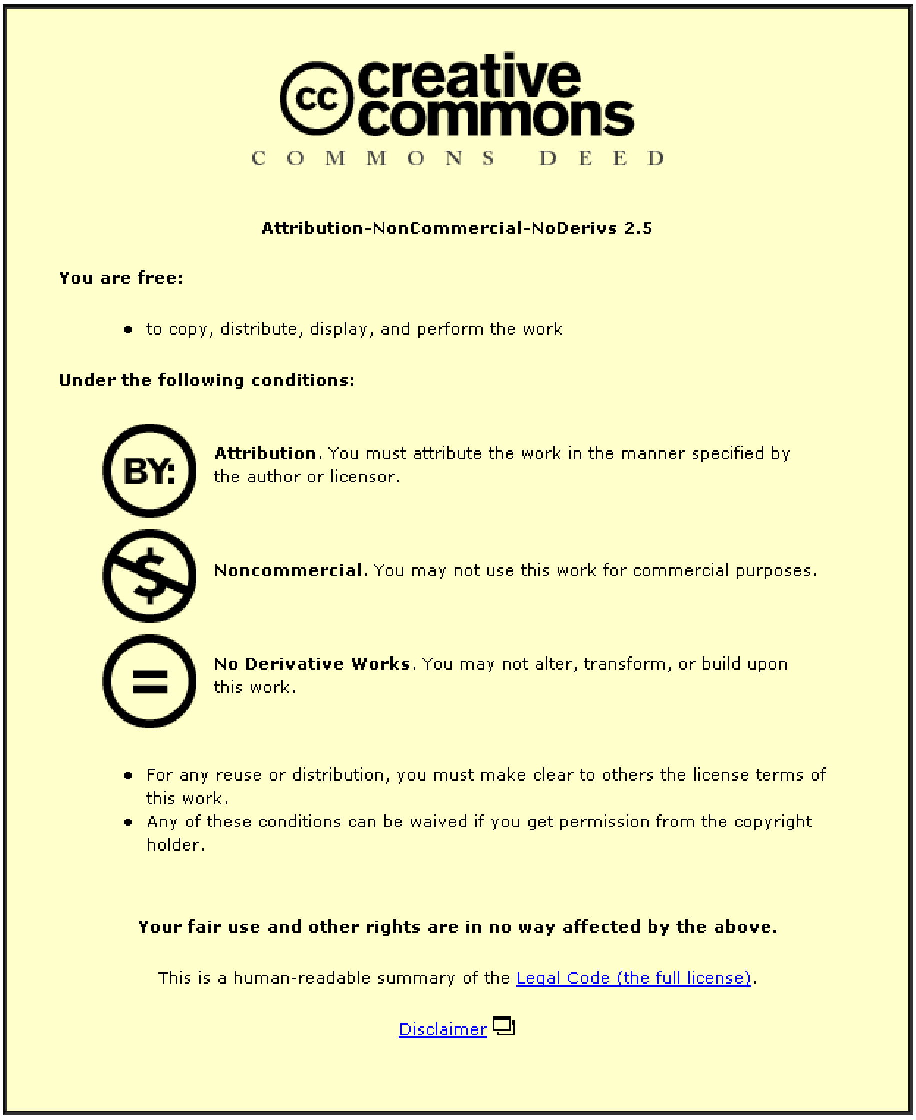

Noname manuscript No.
(will be inserted by the editor)

Non-Repudiable Authentication and Billing Architecture for Wireless Mesh Networks

Raphael C.-W. Phan

Received: date / Accepted: date

Abstract Wireless mesh networks (WMNs) are a kind of wireless ad hoc networks that are multi-hop where packets are forwarded from source to destination by intermediate notes as well as routers that form a kind of network infrastructure backbone. We investigate the security of the recently proposed first known secure authentication and billing architecture for WMNs which eliminates the need for bilateral roaming agreements and that for traditional home-foreign domains. We show that this architecture does not securely provide incontestable billing contrary to designer claims and furthermore it does not achieve entity authentication. We then present an enhanced scheme that achieves entity authentication and nonrepudiable billing.

Keywords Mobile ad hoc networks, mesh networks, security, authentication, billing, non-repudiation, cheating

1 Introduction

UPASS [23] is the first known authentication and billing architecture for wireless mesh networks (WMN) [2,3,13,16,18,22,23,26] that is claimed to provide entity authentication and incontestable billing. The main idea is to motivate potentially selfish intermediate users and routers to forward packets within multi-hop wireless networks by rewarding them with incentives [24].

Rewarding packet forwarders positively instead of penalizing those that do not forward is a good economic approach. Certainly, parties respond more enthusiastically to positive encouragement rather than negative compulsion. Therefore, the billing mechanism needs to function properly and in this sense its incontestable (also known as nonrepudiation) property cannot be flawed. Otherwise, well-behaving packet forwarders might not be rewarded and/or misbehaving nodes could frame innocent ones to cause the latter to be billed extra.

Electronic and Electrical Engineering, Loughborough University, LE11 3TU, UK.
Email: r.phan@lboro.ac.uk

In this paper, we first present our security analysis results of the authentication and billing schemes of UPASS, showing that they do not achieve entity authentication and nonrepudiation. These are the first known security analysis results on UPASS. In doing so, we discuss the reasons behind the cause of these problems. We then present enhanced nonrepudiable authentication and billing schemes that overcome these security problems.

2 Preliminaries

UPASS makes use of identity-based cryptography that is constructed from bilinear pairings over elliptic curves. Let p and q be two large primes and $\mathbb{E}/\mathbb{Z}_p$ denote the elliptic curve $y^2 = x^3 + ax + b$ over $\mathbb{Z}_p$. Let $\mathbb{G}_1$ denote a subgroup of order q of the additive group of points on $\mathbb{E}/\mathbb{Z}_p$, and $\mathbb{G}_2$ denote a subgroup of order q of the multiplicative group of the finite field $\mathbb{F}^*_{p^2}$. A pairing is defined as a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfying the following:

- Bilinear: For all $P, Q \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}^*_q$, we have that

$$\hat{e}(aP, bQ) = \hat{e}(aP, Q)^b = \hat{e}(P, bQ)^a = \hat{e}(P, Q)^{ab}$$

- Non-degenerate: If P is a generator of $\mathbb{G}_1$, then $\hat{e}(P, P)$ is therefore a generator of $\mathbb{G}_2$.
- Computable: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

Some other notations required for the rest of this paper are as follows. $h_k(m)$ denotes a message authentication code (MAC) of message m under key k, $E_{pk}(m)$ denotes ID-based encryption of m under public key pk, $S_{sk}(m)$ denotes ID-based signature of m under private key sk.

2.1 Known Types of Attacks on Authentication and Key Establishment Schemes

The basic requirements of Authentication [12, 19–21] and/or Key Establishment [25] (AKE) schemes can be found in literature, e.g. [17, 5]. In particular, they include the following.

- **Entity authentication (EA):** Each party is assured of the identity of the other party involved in a protocol and that the latter has actually participated.
- **Key-indistinguishability (IND):** An adversary should not be able to obtain even one bit of information about the secret key established by a key establishment scheme.
- **Unknown key-share attack (UKS) resilience:** UKS is an attack where a party A believes that he shares a key with another party B upon completion of a protocol run (this is in fact the case), but B falsely believes that the key is instead shared with a party $E \neq A$. A basic AKE protocol should be resilient to this.
- **Perfect forward secrecy (PFS):** If long-term private keys or master secrets of any party are compromised, the secrecy of previously established session keys should not be affected. This is an attempt to still offer some form of security guarantee in spite of the fact that the long-term secret has been leaked.

- **Key-compromise impersonation (KCI) resilience:** The compromise of any party's long-term private key or master secret should not enable the adversary to impersonate any other parties.
- **Key control (KC) resilience:** No party should be able to control or predict the value (or even some bits) of the established key.

It is important for a security protocol, as is the UPASS protocol, to be secure at least against known types of attacks [7,10,11] including those listed above.

In this paper, we show unfortunately that for the UPASS protocol, it is susceptible to unknown key share (UKS) attacks [11,7], and furthermore, is contestable.

3 Authentication and Billing Architecture for WMNs: UPASS

UPASS is an architecture for authentication and billing for WMNs, proposed by Zhang and Fang [23]. It is based on the usage of identity-based cryptography. UPASS is designed to achieve entity authentication and incontestable (non-repudiable) billing.

Rather than being based on the notion of requiring users to register with a home network and then having foreign networks contact the home network when the user requires access while roaming away from home, UPASS is based on the credit card model, i.e. there exists brokers analogous to banks to which both users and WMN access providers (also known as operators) register and have long-term relationships with. When a user accesses the network within the domain of a WMN operator, the user pays by way of tokens which the operator uses to contact the broker to claim payment for the amount of user access corresponding to the tokens, and the broker then charges the user in order to pay to the operator. Note therefore that unlike the relationship with the broker which is long term, the relationship between user and operator is short term and on a per session basis, i.e. analogous to a customer visiting a supermarket.

3.1 Trust Setup

Within the UPASS architecture, the wireless mesh network is divided into trust domains, each of which is managed by a WMN operator or a broker. Setup consists of the following steps:

1. Generate parameters $\langle p, q, \mathbb{E}/\mathbb{Z}_p, \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$.
2. Choose an arbitrary generator P of $\mathbb{G}_1$.
3. Choose a cryptographic hash function H_1 that maps arbitrary-length strings to non-zero elements in $\mathbb{G}_1$.
4. Choose a random $\kappa \in \mathbb{Z}_q^*$ to be the `domain-master-secret` and generate a public `domain-public-key` from this by $P_{pub} = \kappa P$.

Each router (respectively user) in the WMN domain is assumed to be uniquely identifiable by a network access identifier R-NAI (respectively U-NAI) obtained from his enrolled broker.

Before deployment, a WMN operator supplies each of its mesh routers within its domain with a router pass R-`pass` $= \langle R$-NAI, `expiry-date`$\rangle$ which forms a timestamped ID of the router and a pass-based key R-`key` $= \kappa H_1(R$-`pass`$)$ where

κ is the operator's domain-master-secret. R-pass is made public but R-key is kept secret and known only to the operator and its router.

Meanwhile, for the WMN user, at the point of registration with a broker (analogous to a bank in the real world), the broker than issues a user pass to the user, defined as $U\texttt{-pass} = \langle U\text{-NAI}, \texttt{expiry-date}, \texttt{otherTerms}\rangle$ where otherTerms is used to specify any instance-specific terms and conditions of the user registration. The broker also issues a pass-based user key $U\texttt{-key} = \kappa H_1(U\texttt{-pass})$, where κ is the broker's domain-master-secret. U-pass is made public but U-key is kept secret by the user.

3.2 Entity Authentication

The entity authentication phase between user and WMN operator's router, and between any two users, are described in this section.

Inter-domain User-Router Authentication (URA). The inter-domain authentication between user and router needs to occur when a user migrates to another WMN domain from its current domain.

Consider a user U_1 having the key pair $\langle U_1\texttt{-pass}, \texttt{U1-key}\rangle$ and router R_1 having $\langle R_1\texttt{-pass}, R_1\texttt{-key}\rangle$. A three-way mutual authentication protocol then follows:

1. $R_1 \rightarrow * : R_1\texttt{-pass},\ S_{R_1\texttt{-key}}(t_1)$
2. $U_1 \rightarrow R_1 : U_1\texttt{-pass},\ S_{U_1\texttt{-key}}(t_2)$
3. $R_1 \rightarrow U_1 : \widetilde{U_1\texttt{-pass}},\ E_{U_1\texttt{-pass}}(\widetilde{U_1\texttt{-key}})$

More precisely, U_1 wanders into a new domain and encounters a beacon message broadcast by router R_1. This is indicated above as message (1.). This beacon message includes the router's R_1-pass as well as access fee rates and an R_1-signed timestamp t_1 to prevent replay attacks. U_1 then performs the following:

i. Check if t_1 is fresh.
ii. Check that R_1-pass is still valid, via its expiry-date element.
iii. Verify the signature $S_{R_1\texttt{-key}}(t_1)$ by using R_1-pass.

U_1 then sends message (2.) which includes its own U_1-pass and its signature of a timestamp t_2. R_1 then performs similar steps as (i.) to (iii.) as above. R_1 contacts the domain operator for a temporary key pair for U_1 defined as

$$\widetilde{U_1\texttt{-pass}} = \langle \widetilde{U_1\text{-NAI}}, \texttt{expiry-date}\rangle$$

$$\widetilde{U_1\texttt{-key}} = \kappa H_1(\widetilde{U_1\texttt{-pass}})$$

R_1 sends message (3.) to U_1 including the temporary public key $\widetilde{U_1\texttt{-pass}}$ and the encryption of the temporary private key $\widetilde{U_1\texttt{-key}}$. U_1 decrypts this to obtain $\widetilde{U_1\texttt{-key}}$ and checks if the following holds:

$$\hat{e}(\widetilde{U_1\texttt{-key}}, P) = \hat{e}(H_1(\widetilde{U_1\texttt{-pass}}), P_{pub}).$$

U_1 stores this temporary key pair for its use within the WMN domain that it just joined. Note that upon completion of this protocol, both U_1 and R_1 have also implicitly established a shared secret key

$$\begin{aligned} K_{R_1,U_1} &= \hat{e}(R_1\text{-}\mathtt{key}, H_1(\widetilde{U_1\text{-}\mathtt{pass}})) \\ &= \hat{e}(H_1(R_1\text{-}\mathtt{pass}), H_1(\widetilde{U_1\text{-}\mathtt{pass}}))^{\kappa} \\ &= \hat{e}(H_1(\widetilde{U_1\text{-}\mathtt{pass}}), H_1(R_1\text{-}\mathtt{pass}))^{\kappa} \\ &= \hat{e}(\widetilde{U_1\text{-}\mathtt{key}}, H_1(R_1\text{-}\mathtt{pass})) \\ &= K_{U_1,R_1}. \end{aligned}$$

Intra-Domain User-User Authentication (UUA). Within the same mesh in a WMN domain, the authentication between users is needed since only packets from legitimate users should be forwarded by other users, otherwise the latter forwarding users are not assured to get any incentives for forwarding packets.

The UPASS architecture defines that this kind of authentication be based on the temporary key-pair $\langle\widetilde{U\text{-}\mathtt{pass}}, \widetilde{U\text{-}\mathtt{key}}\rangle$. Note that possession of this key-pair already implies that the user U has been successfully authenticated against a WMN router of the current domain. Let U_1 and U_2 denote the two users wishing to run the user-user authentication protocol. User U_1 (respectively U_2) sends its pass $\widetilde{U_1\text{-}\mathtt{pass}}$ (respectively $\widetilde{U_2\text{-}\mathtt{pass}}$) to each other. Using the received pass of the other user, they can then generate the shared secret key as

$$K_{U_1,U_2} = \hat{e}(H_1(\widetilde{U_1\text{-}\mathtt{pass}}), H_1(\widetilde{U_2\text{-}\mathtt{pass}}))^{\kappa}.$$

This shared key can then be subsequently used for authenticating each other, e.g. U_1 sends to U_2 a random challenge r_1 encrypted with K_{U_1,U_2} to which U_2 responds with the encryption of $r_1 + 1$ under the same shared key.

3.3 Incontestable Billing

Once mutual authentication is achieved, and the user accesses the network, the operator can then charge the user for its accesses via a session-based billing scheme of UPASS that is claimed to be incontestable (nonrepudiable).

A payment structure is used, defined as follows:

$$\langle S_{U_1\text{-}\mathtt{key}}(D_{U_1\to R_1}), \langle a_m\rangle, \langle w_{1,t}\rangle, \langle w_{2,t}\rangle, \ldots, \langle w_{m,t}\rangle\rangle,$$

where we define

$$D_{U_1\to R_1} = \langle R_1\text{-}\mathtt{NAI}, \mathtt{expiry\text{-}date}, L, a_1, t, m\rangle$$

and L represents the monetary worth of each token to be used in the scheme for payment while t and m respectively denote the parameters that define the length of the payment chains $\langle w_{i,t}\rangle$ and proof chain $\langle a_m\rangle$ to be defined in ensuing paragraphs.

$\langle a_m\rangle$ denotes the chain of m hash values $\{a_i | 1 \leq i \leq m\}$ generated as

$$a_i = h(a_{i+1}),$$

where $h(\cdot)$ is a cryptographic hash function, starting from the initial random root a_m and proceeding through index i in descending order, i.e. $i = m$ down to $i = 1$. The nice property of such a hash chain is that given a_{i-1} it is computationally infeasible to derive a_i although computing in opposite direction, i.e. deriving a_i from a_{i-1} is efficient. For the context of this billing scheme, this hash chain $\langle a_m \rangle$ is called a proof chain.

Similarly, $\langle w_{i,t} \rangle$ denotes the chain of t hash values $\{w_{i,j} | 1 \leq j \leq t\}$ generated as

$$w_{i,j} = h(w_{i,j+1}),$$

derived from the initial random root $w_{i,t}$. This hash chain $\langle w_{i,t} \rangle$ is called a payment chain.

$S_{U_1\texttt{-key}}(D_{U_1 \rightarrow R_1})$ denotes U_1's signed commitment on his intention to pay to R_1, and this needs to be sent to R_1 before any session starts. R_1 verifies this with U_1-`pass` and saves it in order to verify subsequent payments from U_1.

For U_1 to start paying for its network accesses by using tokens, recall that the payment commitment $S_{U_1\texttt{-key}}(D_{U_1 \rightarrow R_1})$ structure $D_{U_1 \rightarrow R_1}$ sent from U_1 to R_1 contains the element a_1. This can be used as a proof token to verify the authenticity of its corresponding payment chain $\langle w_{1,t} \rangle$. In more detail, in order for U_1 to spend the payment tokens $\{w_{1,j} | 1 \leq j \leq t\}$ of $\langle w_{1,t} \rangle$, U_1 sends $\langle w_{1,1}, h_{a_1}(w_{1,1}) \rangle$ to R_1. a_1 is viewed as a one-time password of U_1 for this particular payment chain $\langle w_{1,t} \rangle$, and in this sense $h_{a_1}(w_{1,1})$ is a message authentication code (MAC) on the value of $w_{1,1}$. If R_1 successfully verifies $\langle w_{1,1}, h_{a_1}(w_{1,1}) \rangle$, it knows that $w_{1,1}$ is authentic and saves it for later verification of other subsequent payment tokens $w_{1,j}$ (for $j = 2, \ldots, t$) by checking the relation

$$w_{1,j-1} = h(w_{1,j})$$

starting from $w_{1,1}$. When U_1 has used up all the payment tokens of the payment chain $\langle w_{1,t} \rangle$, it moves on to have the next payment chain $\langle w_{i,t} \rangle$ (for $i = 2, \ldots, m$) authenticated by sending $\langle a_i, w_{i,1}, h_{a_i}(w_{i,1}) \rangle$ (for $i = 2, \ldots, m$) to R_1. As like before, R_1 checks if $a_{i-1} = h(a_i)$, and then uses a_i to verify the authenticity of $w_{i,1}$ via $h_{a_i}(\cdot)$.

The router stores a payment record for a user U_i as follows:

$$\langle S_{U_1\texttt{-key}}(D_{U_1 \rightarrow R_1}), a_k, \{w_{i,1}, h_{a_i}(w_{i,1}), w_{i,k_i}, k_i | 1 \leq i \leq k\} \rangle$$

where a_k $(1 \leq k \leq m)$ denotes the highest-indexed proof token and w_{i,k_i} $(1 \leq k_i \leq t)$ denotes the highest-indexed payment token from $\langle w_{i,t} \rangle$.

When R_1 wishes to redeem the users' payments, it reports all stored payment records to the WMN domain operator who then contacts a broker. The steps to be conducted by the broker would be:

1. Verify $S_{U_1\texttt{-key}}(D_{U_1 \rightarrow R_1})$, including to verify the user's signature, and that it has not expired.
2. Verify that $a_1 = h^{k-1}(a_k)$ and store the intermediate values $a_{k-1}, \ldots, a_2$.
3. Compute $h_{a_i}(w_{i,1})$ (for $i = 1, \ldots, k$).
4. Verify that $w_{i,1} = h^{k_i-1}(w_{i,k_i})$ for $(i = 1, \ldots, k)$, and credit the operator's account with k_i number of L-valued monetary units if this is satisfied.

4 Cryptanalysis of UPASS

In this section, we present the results of our analysis of UPASS' security, in particular relevant to its claims of entity authentication and incontestable billing. We treat each one in turn.

4.1 Inter-Domain User-Router Authentication (URA) Scheme

Unknown Key Share Attacks. Message (2.) of the inter-domain user-router authentication (URA) scheme, i.e. $\langle U_1\text{-}\mathtt{pass}, S_{U_1\text{-}\mathtt{key}}(t_2)\rangle$ indicates only the originator i.e. U_1 but does not indicate the intended recipient i.e. R_1. This makes it susceptible to a typical man-in-the-middle attack known as unknown key share (UKS) attack [11,7] that breaks the security goal of entity authentication. Such an attack is known to apply to the conventional Diffie-Hellman (DH) key establishment protocol and thus recent advanced DH protocol are designed to resist this kind of attack.

In more detail, a UKS attack on URA proceeds as follows:

1. U_1 upon seeing message (1.) broadcast from router R_1, replies with message (2.). This is intercepted by the adversary who instead channels it to another router R_2.
2. R_2 therefore thinks U_1 intends to authenticate with it, and replies with message (3.) to U_1.

Therefore, U_1 ends up thinking it has authenticated itself to R_1 when in fact it is authenticated with R_2. The implicitly established shared key is also between U_1 and R_2 instead of U_1 and R_1.

A more devastating UKS attack on URA can be mounted as follows:

1. U_1 on seeing message (1.) responds with message (2.) to R_1 as per normal operational run of the URA protocol.
2. When R_1 responds with message (3.) i.e. $\langle \widetilde{U_1\text{-}\mathtt{pass}}, E_{U_1\text{-}\mathtt{pass}}(\widetilde{U_1\text{-}\mathtt{key}})\rangle$ to U_1, this is intercepted by the adversary U_2 and replaced with $\langle \widetilde{U_2\text{-}\mathtt{pass}}, E_{U_1\text{-}\mathtt{pass}}(\widetilde{U_2\text{-}\mathtt{key}})\rangle$, where $\widetilde{U_2\text{-}\mathtt{key}}$ is known to the adversary U_2. Thus, the established shared key computed by U_1 becomes

$$\begin{aligned}
K'_{U_1,R_1} &= \hat{e}(R_1\text{-}\mathtt{key}, H_1(\widetilde{U_2\text{-}\mathtt{pass}}))\\
&= \hat{e}(H_1(R_1\text{-}\mathtt{pass}), H_1(\widetilde{U_2\text{-}\mathtt{pass}}))^{\kappa}\\
&= \hat{e}(H_1(\widetilde{U_2\text{-}\mathtt{pass}}), H_1(R_1\text{-}\mathtt{pass}))^{\kappa}\\
&= \hat{e}(\widetilde{U_2\text{-}\mathtt{key}}, H_1(R_1\text{-}\mathtt{pass}))\\
&= K_{R_1,U_2}.
\end{aligned}$$

Thus, U_1 ends up computing a shared key which it thinks is shared with R_1 but instead is shared with the adversary U_2. What is worse, the adversary U_2 can even compute the shared key K'_{U_1,R_1}, and therefore take part in subsequent communications with U_1 while U_1 thinks it is communicating with R_1.

Remark on Key Control. Note that as a side remark, URA is not resilient to key control, i.e. a generated key (in this case the user U's temporary key pair) is not jointly established by all involved parties (in this case the router and the user U), but is rather generated altogether by the router's domain operator. This is undesirable [17] within the general context of authentication and key establishment protocols, and is so in the current case where only the broker is fully trusted whereas the user and router (and its operator) do not trust each other [23].

4.2 Intra-Domain User-User Authentication (UUA) Scheme

Unknown Key Share Attacks. UUA is susceptible to the same kind of unknown key share attacks as URA. The reason is that two users wishing to authenticate each other simply exchange their U-`pass` without integrity protection nor entity authentication. As details of the attacks are similar to those on URA in the preceeding subsection, we omit the description here.

Remark on Privacy. Note that UUA makes use of the temporary key pair $\langle \widetilde{U\text{-}\mathtt{pass}}, \widetilde{U\text{-}\mathtt{key}} \rangle$ generated during the URA stage. Nevertheless, this temporary key pair is generated by the operator. Thus, since the temporary key pair is used to derive the UUA shared key, therefore besides the two involved users U_1 and U_2, in fact the operator is also able to derive the key. This may not be desirable with respect to privacy since even third party key exchange servers are typically assumed to be honest but curious [1].

Considering the current UPASS setting where users do not trust the operator [23], it is desired that the operator should not know the key that is established between any pair of users. The motivation for the adversarial operator in mounting this attack is as follows. The operator impersonates the user (client) so that the user client gets charged for accesses due to the impersonating operator, and so the operator benefits from this since charges go into rewarding the forwarding users and towards paying the operator.

4.3 Incontestable Billing Scheme

In the security analysis section of UPASS' billing scheme, the designers considered how the scheme prevents from cheating users and/or cheating operators, basically in terms of users who access the network but not paying later, and operators who get paid but not really delivering on the services paid for. In these situations, the cheatings are incontestable, i.e. the culprit (user or operator) cannot deny that they cheated, and so culprits can be penalized by various means e.g. legal court actions.

We highlight here a more devastating form of cheating on UPASS' billing scheme, where the actual cheating act is repudiable (contestable), i.e. the culprit can deny being involved, and so one cannot then proceed to take the matter to court for any legal actions.

Cheating the User. Consider the billing scheme of UPASS. The router R_1 gets a copy of a_1 (the initial proof token) from the user U_1, and in similar fashion the other subsequent a_i ($2 \leq i \leq m$) at some later time. Note that this value is used to compute the MAC $h_{a_1}(\cdot)$ (subsequently $h_{a_i}(\cdot)$) for verifying the authenticity of the payment token $w_{1,1}$ (subsequently $w_{i,1}$). Thus, a router can use the same a_i ($1 \leq i \leq m$) to forge the MAC $h_{a_i}(\cdot)$ such that a different $w'_{i,1}$ is used instead of $w_{i,1}$. The gist is that this $w'_{i,1}$ is generated by the router in typical iterated hash chain fashion from a root $w'_{i,t}$ that is known to (or in fact arbitrarily chosen by) the router, to satisfy the relation

$$w_{i,j-1} = h(w_{i,j}), \quad j = 2, \ldots, t.$$

In this way, since the router knows all the values of this $\langle w'_{i,t} \rangle$ payment chain, it can replace the user-intended payment chain $\langle w_{i,t} \rangle$ with its own generated $\langle w'_{i,t} \rangle$ and thereby claim all t payments even if the user only accessed once and paid only one payment token $w_{i,1}$.

Indeed, this attack demonstrates the flaw in using symmetric cryptography, as is the case for the MAC $h_{a_i}(\cdot)$, for achieving the incontestability claimed by UPASS' billing scheme. MACs use the same secret key for generation and verification, thus in the current case in addition to the user, the verifier (router) also [8] gets hold of the MAC generation key and hence nonrepudiation (incontestability) cannot hold.

5 Enhanced Schemes for Entity Authentication and Nonrepudiation

In this section, we describe enhancements to the authentication and billing schemes such that they properly provide entity authentication and nonrepudiation.

5.1 Entity Authentication

As discussed in Section 4, the major problem with the entity authentication phase of UPASS is the susceptibility to unknown key share attacks since it is only explicit what the identity of the sender is. Thus, an enhanced inter-domain user-router authentication (URA) can be described as follows:

1. $R_1 \rightarrow * : R_1\texttt{-pass},\ S_{R_1\texttt{-key}}(t_1)$
2. $U_1 \rightarrow R_1 : U_1\texttt{-pass},\ S_{U_1\texttt{-key}}(t_2, R_1\texttt{-pass})$
3. $R_1 \rightarrow U_1 : \widetilde{U_1\texttt{-pass}},\ E_{U_1\texttt{-pass}}(\widetilde{U_1\texttt{-key}}),$

$$S_{R_1\texttt{-key}}(\widetilde{U_1\texttt{-pass}}, t_2)$$

Message (2.) indicates not just the identity of the sender U_1 via its signature, but also explicitly [6] indicates U_1's intended recipient via the inclusion of R_1-`pass` in the signature. The signature by R_1 in message (3.) allows U_1 to verify the authenticity of the temporary public key $\widetilde{U_1\texttt{-pass}}$ as well as know this is not a replay.

5.2 Nonrepudiable Billing

Recall that the payment structure used in the billing scheme is as follows:

$$\langle S_{U_1\text{-}\mathtt{key}}(D_{U_1 \to R_1}), \langle a_m \rangle, \langle w_{1,t} \rangle, \langle w_{2,t} \rangle, \ldots, \langle w_{m,t} \rangle \rangle.$$

In our enhanced billing scheme, we redefine the structure in the signature $S_{U_1\text{-}\mathtt{key}}(D_{U_1 \to R_1})$ from U_1 to R_1 as follows:

$$\begin{aligned} D_{U_1 \to R_1} &= \\ \langle R_1\text{-}\mathtt{NAI}, \mathtt{expiry\text{-}date},\ L,\ a_1, t, m, h_2(w_{1,t}), \ldots, h_2(w_{m,t}) \rangle, \end{aligned}$$

where $h_2(\cdot)$ is some cryptographic hash function. The rest of the billing scheme proceeds as normal, as per the description in section 3.3.

The starting point here is to recall that MACs alone do not provide any non-repudiation [8]. In contrast, the above suggestion to include the roots $w_{i,t}$ $(1 \leq i \leq m)$ of the payment chains $\langle w_{i,t} \rangle$ $(1 \leq i \leq m)$ in U_1's signature commits these roots as the actual values used to generate the payment chains, and therefore are nonrepudiable, i.e. U_1 cannot later deny having used them as payment tokens. Furthermore, a router cannot replace them during the payment stage with payment tokens of its choice and thus cannot frame innocent users for making extra payments which had not been made, so cheating is prevented.

6 Conclusion

We have presented our results of the security analysis of an authentication and billing architecture for wireless mesh networks. It is vital that the security of authentication and billing schemes in literature are analyzed because the only way the public can have faith to use them is if they are assured that the security is sound after having been put through considerable amount of analysis. They would want to be assured that they are only billed for the accesses they had actually made, and that they will definitely be rewarded if they had forwarded packets of other users. Furthermore, in the current setting of mesh networks, user nodes will not be willing to depend on other nodes to forward packets nor willing to themselves forward the packets if there are doubts in the billing scheme especially with respect to nonrepudiation. If the latter occurs, the basic functionality of wireless mesh networks would cease to work. To that end, we proposed enhanced authentication and billing schemes that offer entity authentication and nonrepudiation.

References

1. M. Abdalla, P.-A. Fouque and D. Pointcheval, "Password-based Authenticated Key Exchange in the Three-Party Setting," *IEE Proceedings - Information Security*, Vol. 153, No. 1, pp. 27–39, 2006.
2. I.F. Akyildiz and X. Wang, "A Survey on Wireless Mesh Networks," *IEEE Communications Magazine*, Vol. 43, No. 9, pp. S23–S30, 2005.
3. I.F. Akyildiz, X. Wang and W. Wang, "Wireless Mesh Networks: a Survey," *Computer Networks*, Vol. 47, No. 4, pp. 445–487, 2005.

4. M. Bellare, D. Pointcheval and P. Rogaway, "Authenticated Key Exchange Secure against Dictionary Attacks," *Advances in Cryptology - EUROCRYPT '00*, LNCS 1807, pp. 139–155, 2000.
5. C. Boyd and A. Mathuria. *Protocols for Authentication and Key Establishment.* Springer-Verlag, 2003.
6. K.-K.R. Choo, C. Boyd and Y. Hitchcock, "Examining Indistinguishability-based Proof Models of Key Establishment Protocols," *Advances in Cryptology - ASIACRYPT '05*, LNCS 3788, pp. 585–604, 2005.
7. W. Diffie, P.C. van Oorschot and M.J. Wiener, "Authentication and Authenticated Key Exchanges," *Design, Codes and Cryptography*, Vol. 2, No. 2, pp. 107-125, 1992.
8. R. Gennaro and P. Rohatgi, "How to Sign Digital Streams," *Information and Computation*, Vol. 165, No. 1, pp. 100–116, 2001.
9. Y. Jiang, C. Lin and X.S. Shen, "Mutual Authentication and Key Exchange Protocols for Roaming Services in Wireless Mobile Networks," *IEEE Transactions on Wireless Communications*, Vol. 5, No. 9, pp. 2569–2577, 2006.
10. M. Just and S. Vaudenay, "Authenticated Multi-Party Key Agreement," *Advances in Cryptology - Asiacrypt '96*, LNCS 1163, pp. 36-49, 1996.
11. B.S. Kaliski Jr, "An Unknown Key-Share Attack on the MQV Key Agreement Protocol," *ACM TISSEC*, Vol. 4, No. 3, pp. 275-288, 2001.
12. T.-F. Lee, S.-H. Chang, T. Hwang and S.-K. Chong, "Enhanced Delegation-based Authentication Protocol for PCSs," *IEEE Transactions on Wireless Communications*, Vol. 8, No. 5, pp. 2166–2171, 2009.
13. M.J. Lee, J. Zheng, Y.-B. Ko and D.M. Shrestha, "Emerging Standards for Wireless Mesh Technology," *IEEE Wireless Communications*, Vol. 13, No. 2, pp. 56–63, 2006.
14. Y. Lin and Y. Chen, "Reducing Authentication Signalling Traffic in Third-Generation Mobile Network," *IEEE Transactions on Wireless Communications*, Vol. 2, No. 3, pp. 493–501, 2003.
15. P. Lin, H.-Y. Chen, Y. Fang, J.-Y. Jeng and F.-S. Lu, "A Secure Mobile Electronic Payment Architecture Platform for Wireless Mobile Networks," *IEEE Transactions on Wireless Communications*, Vol. 7, No. 7, pp. 2705–2713, 2008.
16. X. Lin, R. Lu, P.-H. Ho and X.S. Shen, "TUA: A Novel Compromise-Resilient Authentication Architecture for Wireless Mesh Networks," *IEEE Transactions on Wireless Communications*, Vol. 7, No. 4, pp. 1389–1399, 2008.
17. A.J. Menezes, P.C. van Oorschot and S.A. Vanstone. *Handbook of Applied Cryptography*, CRC Press, 1997.
18. M. Portmann and A.A. Pirzada, "Wireless Mesh Networks for Public Safety and Crisis Management Applications," *IEEE Internet Computing*, Vol. 12, No. 1, pp. 18–25, 2008.
19. K. Ren, W. Lou, K. Zeng and P.J. Moran, "On Broadcast Authentication in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, Vol. 6, No. 11, pp. 4136–4144, 2007.
20. C. Tang and D.O. Wu, "Mobile Privacy in Wireless Networks - Revisited," *IEEE Transactions on Wireless Communications*, Vol. 7, No. 3, pp. 1035–1042, 2008.
21. C. Tang and D.O. Wu, "An Efficient Mobile Authentication Scheme for Wireless Networks," *IEEE Transactions on Wireless Communications*, Vol. 7, No. 4, pp. 1408–1416, 2008.
22. Y. Zhang and Y. Fang, "ARSA: An Attack-Resilient Security Architecture for Multihop Wireless Mesh Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 10, pp. 1916–1928, 2006.
23. Y. Zhang and Y. Fang, "A Secure Authentication and Billing Architecture for Wireless Mesh Networks," *Wireless Networks*, Vol. 13, No. 5, pp. 663–678, 2007.
24. Y. Zhang, W. Lou and Y. Fang, "A Secure Incentive Protocol for Mobile Ad Hoc Networks," *Wireless Networks*, Vol. 13, No. 5, pp. 569–582, 2007.
25. Y. Zhou and Y. Fang, "A Two-Layer Key Establishment Scheme for Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, Vol. 6, No. 9, pp. 1009–1020, 2007.
26. H. Zhu, X. Lin, R. Lu, P.-H. Ho and X.S. Shen, "SLAB: A Secure Localized Authentication and Billing Scheme for Wireless Mesh Networks," *IEEE Transactions on Wireless Communications*, Vol. 7, No. 10, pp. 3858–3868, 2008.