Journal of Safety Engineering 2013, 2(2A): 7-13
DOI: 10.5923/s.safety.201311.02

Proof of the Secrecy Property of Secure WLAN Authentication Scheme (SWAS) Using Extended Protocol Composition Logic

Rajeev Singh[1,*], Teek Parval Sharma[2]

[1]G.B.Pant University, U.S.Nagar, Uttarakhand
[2]National Institute of Technology Hamirpur (H.P.), India

Abstract An effective and secure key exchange is one of the essential feature of a secure communication. Secure WLAN Authentication Scheme (SWAS) is one such mechanism that involves cryptographic operations for providing a secure key exchange. The scheme not only provide entity authentication but also provide authentication to the individual frames involved. It evolves fresh keys for securing data sessions. It enhances the network performance by reducing the number of messages required for authentication and key exchange. The protocol enhances reliability of the WLAN communication system and hence its properties need to be validated. Extended Protocol Composition Logic (PCL) is a formal method used extensively for validating the security properties of a protocol. In this paper, we utilize it for proving the secrecy property of SWAS. Thus, main contribution of the paper is presentation of formal proof for the secrecy property of the SWAS scheme.

Keywords Systems Safety and Security, Network Performance, Communication System Reliability, Extended Protocol Composition Logic (PCL)

1. Introduction

Wireless users are increasing day by day. This calls for need of secure communication between wireless Stations (STA) and Access Point (AP). WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access) and WPA-2 (IEEE 802.11i[1][2]) are three standard ways for controlling the network access and evolving the secret shared key in Wireless Local Area Networks (WLANs). Out of these WEP is currently deprecated due to inherent weaknesses[3-6]. WPA was an interim security standard, requiring software up-gradation of nodes. In year 2004, WPA-2 (IEEE 802.11i[1]) came into existence and is continuing till date. The standard is stable and provides security to data frames. Still the standard has scope for improvisations.

Research community and societies are working to make WLANs robust and secure. In this regard, Secure WLAN Authentication Scheme (SWAS), a delegation based[7-8] scheme is proposed at[9] to provide access and shared key generation between STAs and AP. It involves cryptographic operations for providing a secure key exchange. It evolves fresh keys for securing data sessions. The scheme not only provide entity authentication but also provide authentication to the individual frames involved. It also reduces the entire process of authentication and key generation to only four messages. In 802.11i authentication, with Extensible Authentication Protocol-Transport Layer Security (EAP-TLS) as EAP method, several messages are exchanged between STA & AP and between AP & AS. After this, 4-way handshake is done to evolve fresh keys between STA and AP. Thus, SWAS scheme reduces communication and network overload. The average computation time for SWAS authentication is of the order of 21.8477 ms while on the same set of systems authentication time for 802.11i with EAP-TLS is of the order of 76.8 ms. Thus, the extra computations used in the SWAS does not have much effect.

For enhancing the protocol's security, its properties need to be validated. In this paper, we present a formal proof of the secrecy property of SWAS using extended Protocol Composition Logic (PCL) which is a formal method used for validating the security properties of a protocol.

The rest of the paper is divided into 4 sections. Section 2 presents a brief review of Secure WLAN Authentication Scheme (SWAS). Section 3 discuss the methodology and axioms used. Section 4 provides formal proof of secrecy property of SWAS. Section 5 provides conclusions. Appendix A shows fragment of proof system while Appendix B provides statements of formal proof of secrecy property.

* Corresponding author:
rajeevpec@gmail.com (Rajeev Singh)
Published online at http://journal.sapub.org/safety

2. Review of Secure WLAN Authentication Scheme (SWAS)

We proposed Secure WLAN Authentication Scheme (SWAS) at[9]. It is used for providing entity authentication, access control and fresh key generation. During the authentication process, security of keys and frame contents is maintained. All messages in the scheme either use symmetric key encryption or Message Integrity Code (MIC) for protecting the contents. For this purpose, the scheme has several keys (Master Key (MK), Master Session Key (MSK), Pairwise Master Key (PMK) and Pairwise Transient Key (PTK) mapped to equivalent key hierarchy of 802.11i. The end of SWAS authentication process leads to evolving of a secure key (PTK) for protecting the data frames transmitted between STA and AP.

SWAS is a delegation based[7-8] scheme and has initialization phase, request phase and authentication phase. The *initialization phase* is used to register the STA with the authentication server (AS). Once registered, the STA is given delegation key (σ), securely. The elliptic curve parameters and public keys (PK_{AP}-PK_{STA}) are shared in the *request phase*. These are used for evolving the master key (MK) using elliptic curve diffie-hellman (ECDH) key generation algorithm. In the *authentication phase*, STA uses the delegation key for generating the delegation passcode $(R \text{ and } s)$ which is embedded in message M1. STA also keeps the AS passcode in the M1. The messages used in the scheme are listed and described in table 1 (For simplicity, some of the contents are removed). The purpose and contents of passcodes in the scheme are mentioned in table 2. Delegation passcode is verified at the AP. Its verification authenticates the STA. AP forwards the AS passcode to the AS in M2. It also signs the message M2 digitally. AS authenticates AP via digital signature then decrypts the AS passcode, extracts the random number r3. AS creates the digitally signed message M3 and sends it back to AP. M3 contains the AP passcode and STA passcode. From AP passocode, AP extracts r3 and derives the pairwise master key (PMK) and pairwise transient key (PTK). Using STA passcode, the STA is ensured that the genuine AS is involved in the authentication process. Thus all devices confirm each other's identity.

The AS passcode, AP passcode and STA passcode are all encrypted using keys known only to the sender and receiver. The messages M1 & M4 are protected using MIC while M2 & M3 are digitally signed. The secrecy property imply that the secret keys derived using the material (random numbers r1, r2, r3 and serial number s1) in M1, M2, M3 and M4 remains secret and shared between the desired parties. The keys used in the SWAS scheme are described in table 3. The MK (master key), MSK (master session key), PMK (pairwise master key) and PTK (pairwise transient key) are shared between STA and AP. No other party can calculate these keys. The PTK is used to encrypt the data frames between STA and AP.

SWAS enhances network performance. It utilizes only 4 messages for authentication and key exchange. Two (2) messages are utilized between STA and AP while two (2) are utilized between AP and AS. Whereas, IEEE 802.11i (the standard WLAN security protocol) utilizes large number of messages for the same purpose. A total of nine (9) messages are exchanged between STA and AP while eight (8) messages are exchanged between AP and AS during default EAP-TLS authentication. Four (4) messages are also exchanged between STA and AP during four way handshake that follows EAP-TLS authentication.

Table 1. SWAS Message description

Message	Send from -to	Contents
M1	STA to AP	M1: $\{IDAS, r1, (s1, r2)_{MK}, R, s, (r3, K_{AP-AS})_{\sigma}, MIC_{MSK}\}$
M2	AP to AS	M2: $\{IDSTA, (r3, K_{AP-AS})_{\sigma}, MIC_{K_{AP-AS}}, DigSig_{AP}\}$
M3	AS to AP	M3: $\{(IDSTA, r3)_{K_{AP-AS}}, (IDAP, r3)_{\sigma}, DigSig_{AS}\}$
M4	AP to STA	M4: $\{(s1, r3)_{PTK}, (IDAP, r3)_{\sigma}\}$

Table 2. SWAS Passcodes

Passcode	Contents	Shared between	Purpose
Delegation passcode	$R \text{ and } s$	STA and AP	Initial authentication of the STA to AP,
AS passcode	$(r3, K_{AP-AS})_{\sigma}$	STA and AS	Authentication of the STA to AS, Only after its receipt and verification can STA participate in the communication and can derive the data keys
AP passcode	$(IDSTA, r3)_{K_{AP-AS}}$	AS and AP	Authentication of the AS to AP, Only after its receipt and verification can the AP calculate the PMK and PTK
STA passcode	$(IDAP, r3)_{\sigma}$	AS and STA	Authentication of the AS to STA, only after its receipt and verification can the STA ensure that AP also posses same PTK

Table 3. SWAS Keys

Key	Shared between	Calculated as
MK	STA and AP	Calculated using ECDH
MSK	STA and AP	PRF{ r1, MK }
$K_{AP\text{-}AS}$	STA, AP, and AS	PRF{ r2, MK }
PMK	STA and AP	PRF{ r3, MSK }
PTK	STA and AP	PRF{s1, PMK }

3. Methodology and Axioms Used

The Protocol Composition Logic (PCL)[10-13] is used extensively for proving the security properties like authentication property of a protocol. For proving the secrecy property, in this paper extended PCL[14] by Roy et al. (2008) is used. This extended version of PCL has additional formulas, rules and axioms (in addition to that of PCL). The secrecy properties are formalized using the Has($\hat{X}$,s) predicate. It expresses the fact that principal $\hat{X}$, has the information needed to compute the secret s. SendsSafeMsg(X, s, $\mathcal{K}$) formulates the safe sending of all messages by thread X. SafeNet (s, $\mathcal{K}$) is used for safe sending by all threads. Predicate SafeNet (M, s, $\mathcal{K}$) asserts that every occurrence of s in message M is protected by a key in the set $\mathcal{K}$. The NET rule deals with the safety of all messages in the network assuming all messages in the network were safe prior to that step; provided "each possible protocol step" locally sends out safe messages.

We mention the fragment of PCL proof system considered for proof in Appendix A. It shows only the axioms used in proving the SWAS properties. A comprehensive list of axioms is present in appendix A at[10] and[14]. The Honesty Rule is considered which states that an honest principal is suppose to follow one or more roles of the protocol i.e. it must satisfy every invariant property of the protocol role.

4. Proof of Secrecy Property of SWAS

Secrecy property asserts that some data (keys and other secret information) that are used in the protocol are not revealed to others. In case of encrypted messages, only the agents that have the decryption key can obtain the information. The extended logic involves general approach for the cases where protocol agent receives data encrypted by one of the chosen set of encryption keys, only sends sensitive data under encryption by another key in the set. This approach reduces complexity in proving the secrecy of protocol.

The secrecy properties of SWAS w.r.t. extended PCL logic are listed in table 4. The secrecy objective is expressed in a form that the secret information is known only to certain principals. For ex. SEC^{STA}_{MSK} means that when the thread C finishes executing the client's (STAs) role, the key MSK remains secret. Similarly, other security properties are analogously defined. Hence, we intend to prove:

$SWAS \vdash SEC^{STA}_{MSK}, SEC^{AP}_{MSK}$

$SWAS \vdash SEC^{STA}_{K_{AP\text{-}AS}}, SEC^{AP}_{K_{AP\text{-}AS}}, SEC^{AS}_{K_{AP\text{-}AS}}$

$SWAS \vdash SEC^{STA}_{PTK}, SEC^{AP}_{PTK}$

For proving these properties certain conditions Φ are assumed (Appendix A), along with the honesty of the principals $(\hat{C}-\text{Client/STA}, \hat{A}-\text{AP}, \hat{S}-\text{Server/AS})$. It is then proved (Appendix B) using secrecy induction that this implies SafeNet(MSK,$\mathcal{K}$),

Table 4. SWAS secrecy properties

1. SECMSK : Hon $(\hat{C}, \hat{A}) \supset$ (Has(X,MSK) $\supset \hat{X} \in \{\hat{C}, \hat{A}\}$)
2. $SEC_{K_{AP\text{-}AS}}$: Hon $(\hat{C}, \hat{A}, \hat{S}) \supset$ (Has(X,KAP-AS) $\supset$ $\hat{X} \in \{\hat{C}, \hat{A}, \hat{S}\}$)
3. SECPTK : Hon $(\hat{C}, \hat{A}) \supset$ (Has(X,PTK) $\supset \hat{X} \in \{\hat{C}, \hat{A}\}$)
1. SEC^{STA}_{MSK} :[STA]C SECMSK
2. SEC^{AP}_{MSK} :[AP]A SECMSK
3. $SEC^{STA}_{K_{AP\text{-}AS}}$:[STA]C $SEC_{K_{AP\text{-}AS}}$
4. $SEC^{AP}_{K_{AP\text{-}AS}}$:[AP]A $SEC_{K_{AP\text{-}AS}}$
5. $SEC^{AS}_{K_{AP\text{-}AS}}$:[AS]S $SEC_{K_{AP\text{-}AS}}$
6. SEC^{STA}_{PTK} :[STA]C SECPTK
7. SEC^{AP}_{PTK} :[AP]A SECPTK

SafeNet($K_{AP\text{-}AS}$,$\mathcal{K}$) and SafeNet(PTK,$\mathcal{K}$). POS axiom is then utilized to conclude the SWAS secrecy proof. We are not showing the SWAS secrecy property w.r.t. PMK due to similarity and space constraint.

5. Conclusions

A Secure WLAN Authentication Scheme (SWAS) is an effort towards building a secure authentication with desired security properties. The scheme asserts to improve the security in WLANs as all its packets are protected using cryptographic measures. It maintains message protection while keeping low network overhead and hence, increases network performance. Strengthening the security of the scheme implies enhancing reliability of the WLAN communication system (key and data exchange). Secrecy is one of its major property through which the scheme strengthen the security. In this paper, we proof the secrecy property of the SWAS using extended protocol composition logic method. As future work we intend to prove the other properties like authentication and resistance to DoS (Denial of Service) attacks.

Appendix A

Fragment of the PCL Proof System

Axioms and Rules for proving SWAS secrecy property

SendsSafemsg ($X,s,\mathcal{K}$) ≡ $\forall M.$ (Send (X,M) ⊃ SafeMsg(M,s,K)

SafeNet ($s,\mathcal{K}$) ≡ $\forall X.$ SendsSafeMsg($X,s,\mathcal{K}$)

SAF0 SafeMsg($s,s,\mathcal{K}$) ∧ SafeMsg($x,s,\mathcal{K}$), Where x is an atomic term different from s

SAF1 SafeMsg($M_0,M_1,s,\mathcal{K}$) ≡ SafeMsg ($M_0,s,\mathcal{K}$) ∧ SafeMsg($M_1,s,\mathcal{K}$)

SAF2 SafeMsg($E_{sym}[k]$ (M),$s,\mathcal{K}$) ≡ SafeMsg ($M,s,\mathcal{K}$) ∨ $k \in \mathcal{K}$

SAF4 SafeMsg($HASH$ (M), $s,\mathcal{K}$)

NET $\forall \rho \in \mathcal{Q} \quad \forall P \in BS(\rho)$

$$\frac{SafeNet(s,\mathcal{K})[P]_X\, Honest(\hat{X}) \wedge \Phi \supset SendsSafeMsg(X,s,\mathcal{K})}{\mathcal{Q} \vdash KOHonest(s,\mathcal{K}) \wedge \Phi \supset SafeNet(s,\mathcal{K})}(*)$$

(*): $[P]_A$ does not capture free variables in Φ, $\mathcal{K}$, s, and Φ is prefix closed.

NET0 SafeNet($s,\mathcal{K}$)[]$_x$ SendsSafeMsg($X,s,\mathcal{K}$)

NET1 SafeNet($s,\mathcal{K}$)[receive M]$_x$ SafeMsg($M,s,\mathcal{K}$)

NET2 SendsSafeMsg($X,s,\mathcal{K}$)[a]$_x$ SendsSafeMsg($X,s,\mathcal{K}$),where a is not a send.

NET3 SendsSafeMsg($X,s,\mathcal{K}$)[send M]$_x$ SafeMsg($M,s,\mathcal{K}$) ⊃SendsSafeMsg($X,s,\mathcal{K}$)

POS SafeNet($s,\mathcal{K}$) ∧ Has(X,M) ∧ ¬SafeMsg($M,s,\mathcal{K}$)⊃ ∃ $k \in \mathcal{K}$, Has(X,k) ∧ New(X,s)

Environmental Assumptions:

Secrecy is proved w.r.t. key set $\mathcal{K}$ = ($K_{AP\text{-}AS}$, σ, MK, MSK, PMK, PTK)

Where, r1 ≠ r2 ≠ r3 ≠ s1 ≠ $K_{AP\text{-}AS}$ ≠ MSK ≠ PTK ≠ MK ≠ σ

i.e. all keys in the scheme belongs to set $\mathcal{K}$ and none of them is equal to each other and to the chosen random numbers.

The assumed condition Φ is conjunction of following formulas:

Φ_1: ∀X, M, New(X,k) ⊃ ¬ (Send(X,M) ∧ ContainsOpen (M,k))

Where, ContainsOpen is used to assert that k can be obtained without any decryption.

Φ_2: $\forall X, \hat{A}_0, \hat{S}_0 .r3$. New (X,$K_{AP\text{-}AS}$) ∧ SymEnc(X,r3.$K_{AP\text{-}AS}$, σ) $\supset \hat{X} = \hat{C} \wedge \hat{A}_0 = \hat{A} \wedge \hat{S}_0 = \hat{S}$

New Formulas:

1. Assign(X, x, y): x is assigned value y by thread X
2. Derieve(X, k, x, y): thread X calculates key k from x and y using pseudo random function (PRF)

Derieve(X, k, x, y) ≡ Assign(X,k,Computes(X,PRF{x, y}))

For ex. Derieve(X, MSK, x, y) ≡ Assign(X, MSK, Computes(X,PRF{r1,MK}))

New Axioms:

SAF5: SafeMsg($HASH_k$(M) ,s,$\mathcal{K}$) ≡ SafeMsg(MIC_k,s,$\mathcal{K}$) where MIC_k is MIC of M using k, k∈$\mathcal{K}$

DER : New(X,k_{new}) ≡ New(X,x) ∧ Derieve(X, k_{new}, x, y)

Derivation of key using new random number x leads to new key (k_{new}) formation.

NET4: SafeNet(M,s,$\mathcal{K}$)[derieve k_{new}, x, s)]$_X$SendsSafeMsg($MIC_{k_{New}}$,s,$\mathcal{K}$)

It is safe to send a MIC of the message obtained using the new derieved key

Appendix B

Proof of secrecy property, SWAS:

Let[STA] $_{C'}$: [new r1, r2, r3, s1;
derive MSK, r1, MK;
derive $K_{AP\text{-}AS}$, r2, MK;
enc_{CA}: =symenc s1.r2,MK;
tc := symenc r3. $K_{AP\text{-}AS}$, σ;

MIC_{MSK} = HASH $_{MSK}$(Msg1);
send enc_{CA}.tc. MIC_{MSK} ;] $_{C'}$

case (i) w.r.t. MSK

$$[STA]_{C'}\ New(C', r1) \wedge Derieve(C', MSK, r1, MK) \quad (1)$$

$$[STA]_{C'}\ New(C', r2) \wedge Derieve(C', K_{AP\text{-}AS}, r2, MK) \quad (2)$$

DER, (2) $[STA]_{C'}$ $New(C', K_{AP\text{-}AS}) \wedge Send(C', IDAS.\ E_{sym}[MK]$ (s1.r2).

$$E_{sym}[\sigma]\ (r3.K_{AP\text{-}AS}).MIC_{MSK}) \quad (3)$$

$$\Phi_1,\ (3)\ [STA]_{C'}\ K_{AP\text{-}AS} \neq MK \neq \sigma \quad (4)$$

NET4, (1), (4) $[STA]_{C'}$ SafeMsg(IDAS. E_{sym}[MK] (s1.r2). $E_{sym}[\sigma]$ (r3.

KAP-AS). MICMSK,MSK,) (5)

NET3, (5) SafeNet (MSK,)$[STA]_{C'}$ SendsSafeMsg(C',MSK,) (6)

case (ii) w.r.t. $K_{AP\text{-}AS}$

$$(2)\ [STA]_{C'}\ New(C', K_{AP\text{-}AS}) \wedge SymEnc(C', r3.K_{AP\text{-}AS}, \sigma) \quad (7)$$

$$\Phi_2,\ (7)\ [STA]_{C'}\ \hat{C}' = \hat{C} \wedge \hat{A}' = \hat{A} \wedge \hat{S}' = \hat{S} \quad (8)$$

$$(8)\ [STA]_{C'}\ (MK \in \mathcal{K}) \wedge (\sigma \in \mathcal{K}) \quad (9)$$

(1), NET4, SAF*, (9) SafeNet ($K_{AP\text{-}AS}$, $\mathcal{K}$)$[STA]_{C'}$ SafeMsg(IDAS.E_{sym}[MK] (s1.r2).

$$E_{sym}[\sigma]\ (r3.K_{AP\text{-}AS}).\ MIC_{MSK}, K_{AP\text{-}AS},\ \mathcal{K}) \quad (10)$$

NET*, (10) SafeNet ($K_{AP\text{-}AS}$, $\mathcal{K}$)$[STA]_{C'}$ SendsSafeMsg(C', $K_{AP\text{-}AS}$, $\mathcal{K}$) (11)

Let$[AP]_{A'}$: [receive IDAS.enc_{CA}.tc. MIC_{MSK} ;
$text_{CA}$: =symdec enc_{CA},MK;
match $text_{CA}$ as s1.r2;
derieve $K_{AP\text{-}AS}$, r2, MK;
$MIC_{k_{AP\text{-}AS}} := HASH_{k_{AP\text{-}AS}}$ (Msg2);
send tc. $MIC_{K_{AP-AS}}$. $DigSig_{AP}$;] $_{A'}$

case (i) w.r.t. MSK

$$NET1\ SafeNet\ (MSK, \mathcal{K})[\ receive\ IDAS.enc_{CA}.tc.\ MIC_{MSK}]_{A'}\ SafeMsg(tc, MSK, \mathcal{K}) \quad (12)$$

$$SAF1,\ (12)\ SafeNet\ (MSK, \mathcal{K})[\ AP]_{A'}\ SafeMsg(tc.\ MIC_{K_{AP-AS}}.DigSig_{AP}, MSK, \mathcal{K}) \quad (13)$$

$$NET0,\ (13)\ SafeNet\ (MSK, \mathcal{K})[\ AP]_{A'}\ SendsSafeMsg(A', MSK, \mathcal{K}) \quad (14)$$

case (ii) w.r.t. KAP-AS

$$NET1\ SafeNet\ (K_{AP\text{-}AS}, \mathcal{K})[\ receive\ IDAS.enc_{CA}.tc.\ MIC_{MSK}]_{A'}\ SafeMsg(tc, K_{AP\text{-}AS}, \mathcal{K}) \quad (15)$$

$$(15)[\ AP]_{A'}\ Derieve(A', K_{AP\text{-}AS}, r2, MK) \quad (16)$$

$$NET4, SAF1, (15), (16)\ SafeNet\ (K_{AP\text{-}AS}, \mathcal{K})[\ AP]_{A'}\ SafeMsg(tc.\ MIC_{K_{AP-AS}}.DigSig_{AP}, K_{AP\text{-}AS}, \mathcal{K}) \quad (17)$$

$$NET0,(17)\ SafeNet\ (K_{AP\text{-}AS}, \mathcal{K})[\ AP]_{A'}\ SendsSafeMsg(A', K_{AP\text{-}AS}, \mathcal{K}) \quad (18)$$

Let$[AS]_{S'}$: [receive tc. $MIC_{K_{AP-AS}}$. $DigSig_{AP}$;
$text_{tc}$: = symdec tc, σ;
match $text_{tc}$ as r3. $K_{AP\text{-}AS}$;
enc_{SA}: = symenc IDSTA.r3, $K_{AP\text{-}AS}$;
ts : = symenc IDAP.r3, σ;
send enc_{SA}.ts.$DigSig_{AS}$;] $_{S'}$

NET1, SAF1 SafeNet ($K_{AP\text{-}AS}$, $\mathcal{K}$)[receive tc. $MIC_{K_{AP-AS}}$.$DigSig_{AP}$] $_{S'}$ SafeMsg

$$(E_{sym}[\sigma]\ (r3.K_{AP\text{-}AS}),\ K_{AP\text{-}AS},\ \mathcal{K}) \tag{19}$$

SAF*,(19) SafeNet ($K_{AP\text{-}AS}$, $\mathcal{K}$)[receive tc. $MIC_{K_{AP-AS}}$.$DigSig_{AP}$] $_{S'}$ SafeMsg

$$(r3.K_{AP\text{-}AS}, K_{AP\text{-}AS},\ \mathcal{K}) \vee (\sigma \in \mathcal{K}) \tag{20}$$

SAF2,(20) SafeNet ($K_{AP\text{-}AS}$, $\mathcal{K}$)[receive tc. $MIC_{K_{AP-AS}}$.$DigSig_{AP}$] $_{S'}$ SafeMsg

$$(E_{sym}[K_{AP-AS}](IDSTA.r3),\ K_{AP\text{-}AS},\ \mathcal{K}) \tag{21}$$

SAF2,(20) SafeNet ($K_{AP\text{-}AS}$, $\mathcal{K}$)[receive tc. $MIC_{K_{AP-AS}}$.$DigSig_{AP}$] $_{S'}$

$$SafeMsg(E_{sym}[\sigma](IDAP.r3),\ K_{AP\text{-}AS},\ \mathcal{K}) \tag{22}$$

SAF1,(21),(22) SafeNet ($K_{AP\text{-}AS}$, $\mathcal{K}$)[receive tc. $MIC_{K_{AP-AS}}$.$DigSig_{AP}$] $_{S'}$

$$SafeMsg(enc_{SA}.ts.DigSig_{AS},\ K_{AP\text{-}AS},\ \mathcal{K}) \tag{23}$$

$$NET0,\ 23\ SafeNet\ (K_{AP\text{-}AS},\ \mathcal{K})[\ AS]_{S'}\ SendsSafeMsg(S',\ K_{AP\text{-}AS},\ \mathcal{K}) \tag{24}$$

Precondition θ_1 : Receive(A', E_{sym}[MK](s1,r2))

Let[AP] $_{A'}$: [receive enc_{SA}.ts.$DigSig_{AS}$;
$text_{SA}$: = symdec enc_{SA}, $K_{AP\text{-}AS}$;
match $text_{SA}$ as IDSTA.r3;
enc_{AC} : =symenc s1.r3, PTK;
send enc_{AC}.ts;] $_{A'}$

case (i) w.r.t. KAP-AS

$$NET1\ SafeNet\ (K_{AP\text{-}AS},\ \mathcal{K})[\ receive\ enc_{SA}.\ ts.DigSig_{AS}]_{A'}\ SafeMsg(E_{sym}[K_{AP\text{-}AS}](IDSTA.r3).ts,\ K_{AP\text{-}AS},\ \mathcal{K}) \tag{25}$$

$$(25)\ [AP]_{A'}\ Derieve(A',\ PMK,\ r3,\ MSK) \tag{26}$$

$$(26)\ [AP]_{A'}\ Derieve(A',\ PTK,\ s1,\ PMK) \tag{27}$$

$$\theta_1\ (25),\ (27)\ [AP]_{A'}\ SymEnc(A',\ s1.r3,\ PTK) \wedge K_{AP\text{-}AS} \neq PTK \tag{28}$$

$$(26),\ (27),\ (28)\ SafeNet\ (K_{AP\text{-}AS},\ \mathcal{K})[AP]_{A'}\ SafeMsg(enc_{AC}.ts,\ K_{AP\text{-}AS},\ \mathcal{K}) \tag{29}$$

$$NET0,\ (29)\ SafeNet\ (AKey,\ \mathcal{K})[AP]_{A'}\ SendsSafeMsg(A',\ K_{AP\text{-}AS},\ \mathcal{K}) \tag{30}$$

case (ii) w.r.t. PTK

$$NET1 SafeNet\ (MSK,\ \mathcal{K}) \wedge \theta_1\ [\ receive\ enc_{SA}.ts.DigSig_{AS}]_{A'}\ SafeMsg(E_{sym}[K_{AP\text{-}AS}](IDSTA.r3),\ MSK,\ \mathcal{K}) \tag{31}$$

$$(26),\ (27),\ (31)\ SafeNet\ (PTK,\ \mathcal{K}) \wedge \theta_1\ [AP]_{A'}\ SafeMsg(E_{sym}[K_{AP\text{-}AS}](IDSTA.r3),\ PTK,\ \mathcal{K}) \tag{32}$$

$$NET1\ SafeNet\ (PTK,\ \mathcal{K}) \wedge \theta_1\ [\ receive\ enc_{SA}.ts.DigSig_{AS}]_{A'}\ SafeMsg(ts,\ PTK,\ \mathcal{K}) \tag{33}$$

$$(32),\ SAF1\ SafeNet\ (PTK,\ \mathcal{K}) \wedge \theta_1\ [AP]_{A'}\ SafeMsg(r3,\ PTK,\ \mathcal{K}) \tag{34}$$

$$SAF2,(27),(34)\ SafeNet\ (PTK,\ \mathcal{K}) \wedge \theta_1\ [AP]_{A'}\ SafeMsg(E_{sym}[PTK](s1.r3),\ PTK,\ \mathcal{K}) \tag{35}$$

$$SAF1,(33),(35)\ SafeNet\ (PTK,\ \mathcal{K})[AP]_{A'}\ SafeMsg(E_{sym}[PTK](s1,r3).ts,\ PTK,\ \mathcal{K}) \tag{36}$$

$$NET0,\ (36)\ SafeNet\ (PTK,\ \mathcal{K})[AP]_{A'}\ SendsSafeMsg(A',\ PTK,\ \mathcal{K}) \tag{37}$$

Let[STA] $_{C'}$:[receive enc_{AC}.ts ;
$text_{AC}$: = symdec enc_{AC}, PTK;
match $text_{AC}$ as s1.r3;
$text_{ts}$: = symdec ts,σ;
match $text_{ts}$ as IDAP.r3;] $_{C'}$

$$[STA]_{C'}\ Derieve\ (C',\ PTK,\ s1,\ PMK) \tag{38}$$

$$SAF*,\ (38)\ SafeNet\ (PTK,)[receive(encAC.ts)]_{C'}\ SafeMsg(E_{sym}[PTK](s1.r3)\ .\ E_{sym}[\sigma](IDAP.r3),\ PTK,) \tag{39}$$

Φ_1, (39) SafeNet (PTK, $\mathcal{K}$)[STA] $_{C'}$ SafeMsg(C', PTK, $\mathcal{K}$) (40)

From (6), (14)

$$\Phi \wedge \text{Hon}\ (\hat{C}, \hat{A}) \supset \text{SafeNet(MSK,}\mathcal{K})$$

POS, $$\Phi \wedge \text{Hon}\ (\hat{C}, \hat{A}) \supset \text{Has(X, MSK)} \supset\ (\hat{X} = \hat{C} \vee \hat{X} = \hat{A})$$

$$\text{SWAS} \vdash \text{SEC}^{\text{STA}}_{\text{MSK}},\ \text{SEC}^{\text{AP}}_{\text{MSK}}$$

From (11), (18), (24), (30)

$$\Phi \wedge \text{Hon}\ (\hat{C}, \hat{A}, \hat{S}) \supset \text{SafeNet}(K_{\text{AP-AS}}, \mathcal{K})$$

POS, $$\Phi \wedge \text{Hon}\ (\hat{C}, \hat{A}, \hat{S}) \supset \text{Has(X, } K_{\text{AP-AS}}) \supset\ \hat{X} = \hat{C} \vee \hat{X} = \hat{A} \vee \hat{X} = \hat{S}$$

$$\text{SWAS} \vdash \text{SEC}^{\text{STA}}_{K_{\text{AP-AS}}},\ \text{SEC}^{\text{AP}}_{K_{\text{AP-AS}}},\ \text{SEC}^{\text{AS}}_{K_{\text{AP-AS}}}$$

From (37), (40)

$$\Phi \wedge \text{Hon}\ (\hat{C}, \hat{A}) \supset \text{SafeNet(PTK,}\mathcal{K})$$

POS, $$\Phi \wedge \text{Hon}\ (\hat{C}, \hat{A}) \supset \text{Has(X, PTK)} \supset\ (\hat{X} = \hat{C} \vee \hat{X} = \hat{A})$$

$$\text{SWAS} \vdash \text{SEC}^{\text{STA}}_{\text{PTK}},\ \text{SEC}^{\text{AP}}_{\text{PTK}}$$

REFERENCES

[1] IEEE 802.11i, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Security Enhancements, IEEE Standard, 2004.

[2] A. Holt and CY Huang, 802.11 Wireless Networks: Security and Analysis, Springer-Verlag 2010.

[3] W. A. Arbaugh, N. Shankar, J. Wang, K. Zhang, Your 802.11 network has no clothes, IEEE Wireless Commn. Magazine, 9 (2002) 44 - 51.

[4] A. Bittau, M. Handley, J. Lackey, The Final Nail in WEP's Coffin, in: Proc. of the IEEE Symp. on Security and Privacy (S&P' 06), 2006, pp. 386-400.

[5] E. Tews, R. Weinmann, A. Pyshkin, Breaking 104 bit WEP in less than 60 seconds, in: Proc. Int. Conf. on Info. Sec. Appl. WISA, 2007, pp.188-202.

[6] C. He and J. C. Mitchell, Security Analysis and Improvements for IEEE 802.11i, in: Proc. of the Annual Network and Distr. Syst. Sec. Symp. (NDSS'05), 2005, pp. 90-110.

[7] Wei-Bin Lee, Chang-Kuo Yeh, A New Delegation-Based Authentication Protocol for Use in Portable Communication Systems, IEEE Transaction on Wirel. Commn. 4:1 (2005) 57-64.

[8] C. Tang and D. O. Wu, An Efficient Mobile Authentication for Wireless Networks, IEEE Transactions on Wirel. Commn. 7:4 (2008)1408-1416.

[9] Rajeev Singh, Teek Parval Sharma, A Secure WLAN Authentication Scheme, IEEK Transaction of Smart Processing and Computing, Vol. 2, No. 3, June, 2013.

[10] Changhua He, Analysis of Security Protocols for Wireless Networks, Ph.D. Dissertation, December 2005.

[11] C. He, M. Sundararajan, A. Datta, A. Derek, J. C. Mitchell, A Modular Correctness Proof of IEEE 802.11i and TLS, 12th ACM conference on Computer and communications sec. (CCS'05), Pages 2 – 15, November 2005.

[12] A. Datta, J.C. Mitchell, A. Roy, S-H Stiller, Protocol Composition Logic (PCL) book chapter in V. Cortier and S. Kremer (Editors), Formal Models and Techniques for Analyzing Security Protocols, IOS Press, March 2011.

[13] A. Datta, A. Derek, J.C. Mitchell, A. Roy, Protocol Composition Logic (PCL) Electronic Notes in Theoretical Computer Science 172 (2007) 311–358.

[14] A. Roy, A. Datta, A. Derek, J.C. Mitchell, J-P. Seifert, Secrecy Analysis in Protocol Composition Logic, in: book chapter in O. Grumberg, T. Nipkow and C. Pfaller (Editors), Formal Logical Methods for System Security and Correctness, Volume 14 NATO Science for Peace and Security Series - D: Info. and Commn. Security, IOS Press, March 2008.