Proof of the Authentication Property of Secure WLAN Authentication Scheme (SWAS) using Protocol Composition Logic (PCL)

Rajeev Singh,
G.B.Pant University, Pantnagar
US Nagar, Uttarakhand (India)

Teek Parval Sharma,
National Institute of Technology
Hamirpur (H.P.), India

ABSTRACT

Authentication is one of the essential tools available for security in WLANs. Access control authentication mechanisms provides entity authentication, access into the network and key evolving for data frames protection. Secure WLAN Authentication Scheme (SWAS) is one such access control authentication mechanism. It provides entity authentication along with per frame authentication. All the participating entities in the scheme i.e. STA, AP and AS authenticate each other. The scheme makes use of cryptographic measure like delegation, key management, encryption and MIC for securing the scheme. The security properties of the scheme need to be validated for effectiveness. In this paper, a formal tool i.e. Protocol Composition Logic (PCL) is used for proving the authentication property of the scheme.

General Terms

Networks, Security, WLAN

Keywords

Authentication, Access control, Protocol Composition Logic (PCL).

1. INTRODUCTION

Several kinds of authentication exist in wireless Local Area Networks (WLANs) for ex. lightweight authentication, encryption based authentication, certificate based authentication, frame level authentication. In Lightweight authentication shared key is used as a seed value (input value) for Authentication Stream Generator (ASG). ASG generates authentication stream. Few bits are selected as authentication token and used for providing frame authentication by putting them in the frame [1-7]. The lightweight authentication schemes do not consider other aspects of security like privacy, integrity etc. and usually involve synchronization algorithms for synchronizing the two ASGs. Encryption based authentication is considered mainly by Wired Equivalent Privacy (WEP) protocol. Here, the sender encrypts the data using its shared key while the receiver decrypts the data using its shared key. Decrypting into something meaningful authenticates the sender. WEP has several weaknesses [8-11] and hence is deprecated. IEEE 802.11i [12] provides certificate based entity authentication. It depends upon 802.1X [13] for entity authentication. 802.1X [13] provides authentication along with the access control. For providing frame level authentication, 802.11i calculates message integrity code (MIC) for each frame using the shared key (PTK) [14].

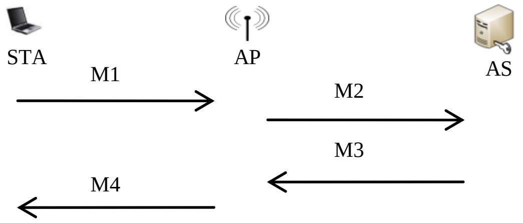
Secure WLAN Authentication Scheme (SWAS) proposed at [15] also provides entity authentication similar to 802.11i. It do not use certificate based authentication rather delegation based approach is used [16-17]. SWAS reduces the process length and complexity of the authentication process. This also decreases the network overload and communication latencies. The scheme ensures atleast the same effect and outcome as that produced jointly by authentication and four way handshake of IEEE 802.11i i.e. it provides STA authentication to AS, derives same number of session keys, maintains PTK freshness and maintains desired security properties. In addition, the scheme provides authentication to all the involved entities i.e. STA, AP, AS. Due to this, attacks like false AP attack where a rogue AP act like genuine AP is not possible. Denial of Service (DoS) attack effect is also reduced by the scheme. The scheme has 4 messages and all the messages are also authenticated by using cryptographic measures like delegation, encryption, MIC and digital signature. These cryptographic operations involve computations at the AP. It is shown that no extra latencies arise due to increase in computations.

The protocol's security properties need to be validated. Hence, a formal proof is provided for validating its authentication property. In proof, protocol composition logic (PCL) is utilized.

The rest of the paper is divided into 3 sections. Section2 presents an overview of the Secure WLAN Authentication Scheme (SWAS). Section 3 presents proof of the authentication property of SWAS using PCL. Section 4 provides conclusion. Appendix A contains existing rules and formulas used in formal proof. Appendix B provides statements of formal proof of authentication property.

2. REVIEW OF SECURE WLAN AUTHENTICATION SCHEME (SWAS)

Secure WLAN Authentication Scheme (SWAS) provides both entity authentication and per frame authentication. It has three entities namely wireless station (STA), access point (AP) and authentication server (AS). Four messages are used in the scheme namely M1, M2, M3 and M4.The entities and the messages used are shown in figure 1. The information not pertaining to the scope of the paper is removed from it. AS authenticates the STA entity and after the authentication is successful, AP provides access to STA into the network. For entity authentication secret key are used. Keys evolved in



M1: $\{IDAS,(s1,r2)_{MK},R,s,(r3,K_{AP-AS})_{\sigma},MIC_{MSK}\}$

M2: $\{IDSTA,(r3,K_{AP-AS})_{\sigma},MIC_{K_{AP-AS}},DigSig_{AP}\}$

M3: $\{(IDSTA,r3)_{K_{AP-AS}},(IDAP,r3)_{\sigma},DigSig_{AS}\}$

M4: $\{(s1,r3)_{PTK},(IDAP,r3)_{\sigma}\}$

Fig 1: Review of SWAS

SWAS are listed in table 1. Successful authentication finally leads to evolving Pairwise Transient Key (PTK) between STA and AP. PTK is used to encrypt the data packets in the following sessions [15].

Entity authentication in SWAS is ensured using delegation passcode, AS passcode, AP passcode and STA passcode (table 2). STA's Identity at the AP is verified by delegation verification. AS passcode is kept by STA in M1, it is forwarded by the AP to the AS in M2. Identity of AP is verified at AS using associated digital signature of AP in M2. AS passcode is checked on its receipt at AS. Only AS can verify AS passcode as only it has the shared key σ. AS creates AP passcode for the AP and STA passcode for the STA. AP passcode is verified by the AP using shared key K_{AP-AS}. This authenticates AS to AP. M3 also contains STA passcode. STA passcode kept in M3 is forwarded by the AP to the STA. It is verified at the STA using shared key σ. Its verification authenticates AS to STA. Thus, all the entity verifies the identity of each other in SWAS communication.

SWAS provides message authentication and integrity to all the four message used i.e. M1, M2, M3 and M4. M1 has associated MIC calculated using MSK evolved from the shared master key (MK). M2 and M3 are authenticated using digital signatures of AP and AS respectively. The digital signatures are verified using corresponding public keys of AP and STA. M4 is having two parts (table 3). First contains serial number s1 & random number r3 encrypted using PTK. This is authenticated by matching s1 and r3 with the stored s1 and r3 at the STA. This ensures integrity of first part of M4. Second part contains IDAP with whom STA is associating & r3 encrypted using σ. Both the parts contain same r3. Only AS can encrypt and put these in M4, hence AS is authenticated along with ensuring the integrity of second part of M4. Thus, all the messages used maintains integrity and are authenticated.

Table1: SWAS Key summary

Key	Shared between	Calculated as
MK	STA and AP	Calculated using Elliptic Curve Diffie Hellman key generation algorithm
MSK	STA and AP	PRF{ r1, MK }
K_{AP-AS}	STA, AP, and AS	PRF{ r2, MK }
PMK	STA and AP	PRF{ r3, MSK }
PTK	STA and AP	PRF{s1, PMK }

Table 2. SWAS Entity authentication

Passcode	Contents	Shared between	Entity authentication
Delegation passcode	R and s	STA and AP	Provides STA authentication at AP
AS passcode	$(r3,K_{AP-AS})_{\sigma}$	STA and AS	Provides STA authentication at AS
AP passcode	$(IDSTA,r3)_{K_{AP-AS}}$	AS and AP	Provides AS authentication at AP
STA passcode	$(IDAP,r3)_{\sigma}$	AS and STA	Provides AS authentication at STA

Table 3: SWAS message authentication

Message	Contents	Description
M1	MIC_{MSK}	Verified at AP using MSK evolved from MK
M2	$DigSig_{AP}$	Verified at AS using AP public key
M3	$DigSig_{AS}$	Verified at AP using AS public key
M4	$(s1,r3)_{PTK},(IDAP,r3)_{\sigma}$	r3 matching at STA after decrypting first part i.e. $(s1,r3)_{PTK}$ will indicate that this part is authentic. r3 matching after decrypting second part i.e. $(IDAP, r3)_{\sigma}$ will indicate that this part is authentic.

Table 4. Swas Program

SWAS: SUPPL = $(Y, \hat{X}, MK)$ [derieve $MSK, r1, MK$; derieve $K_{AP\text{-}AS}, r2, MK$; send $\hat{Y}, \hat{X}, "msg1", ENC_{MK}(s1, r2), R, s, ENC_{\sigma}(r3, K_{AP-AS}), Hash_{MSK}(Msg1)$ send $\hat{X}, \hat{Y}, "msg4", ENC_{PTK}, ENC_{\sigma}$; match $ENC_{PTK} / (IDSTA, r3)_{PTK}$; match $ENC_{\sigma} / (IDAP, r3)_{\sigma}$; $]_{Y}$
SWAS: AUTH = $(X, \hat{Y}, MK)$ [receive $\hat{Y}, \hat{X}, "msg1", ENC_{MK}, x$; match x/R,y; match y/$s$, ENC_{σ}, MIC_{MSK}; match $ENC_{MK} / ENC_{MK}(s1, r2)$; match $MIC_{MSK} / Hash_{MSK}(Msg1)$; derieve $K_{AP\text{-}AS}, r2, MK$; send $\hat{X}, \hat{Z}, "msg2", ENC_{\sigma}(r3, K_{AP-AS}), Hash_{K_{AP-AS}}(Msg2), digsig_{AP}$ receive $\hat{Z}, \hat{X}, ENC_{K_{AP-AS}}, ENC_{\sigma}, digsig_{AS}$; match $ENC_{K_{AP-AS}} / (IDSTA, r3)_{K_{AP-AS}}$; send $\hat{X}, \hat{Y}, ENC_{PTK}(s1, r3), (IDAP, r3)_{\sigma}$; $]_{X}$
SWAS: AS = $(Z, \hat{X})$ [receive $\hat{X}, \hat{Z}, "msg2", ENC_{\sigma}, MIC_{K_{AP-AS}}, digsig_{AP}$ match $ENC_{\sigma} / ENC_{\sigma}(r3, K_{AP-AS})$; match $MIC_{K_{AP-AS}} / Hash_{K_{AP-AS}}(Msg2)$ send $\hat{Z}, \hat{X}, (IDSTA, r3)_{K_{AP-AS}}, (IDAP, r3)_{\sigma}, digsig_{AS}$; $]_{Z}$

Table 5. SWAS invariants used in proving authentication property

$\Gamma_{SWAS,1}$:= Computes $(\hat{X}, PRF(s1, PMK)) \supset \neg$ (Send(X,M) $\wedge$ Contains (M, $(PRF(s1, PMK))$))
$\Gamma_{SWAS,2}$:= (Honest $(\hat{X})$ $\wedge$ Send(X,Msg1) $\supset \neg$ (Send(X, Msg2) $\wedge$ Send(X, Msg3) $\wedge$ Send(X, Msg4)))
$\Gamma_{SWAS,3}$:= (Honest $(\hat{X})$ $\wedge$ Receive(X,Msg2) $\supset \neg$ (Send(X, Msg1) $\wedge$ Send(X, Msg4)))
$\Gamma_{SWAS,4}$:= (Honest $(\hat{X})$ $\wedge$ Has(X,K_{AS}) $\supset$ $\hat{X} = \hat{Z}_0$)
$\Gamma_{SWAS,5}$:= (Honest $(\hat{X})$ $\wedge$ Has(X,K_{AP}) $\supset$ $\hat{X} = \hat{X}_0$)
$\Gamma_{SWAS,6}$:= (Honest $(\hat{X})$ $\wedge$ Has(X,σ) $\supset$ $\hat{X} = \hat{Y}_0 \vee \hat{X} = \hat{Z}_0$)
SPMK := Honest $(\hat{X})$ $\wedge$ Honest $(\hat{Y})$ $\supset$ Has $(\hat{Z}, PMK) \supset \hat{Z} = \hat{X} \vee \hat{Y}$

3. FORMAL PROOF OF THE SWAS AUTHENTICATION PROPERTY

The proof utilizes the Protocol Composition Logic (PCL) [18-21]. Appendix A contains existing rules and formulas used in formal proof. PCL is used to represent a protocol by a set of roles (such as "Initiator", "Responder" or "Server"). SWAS has three roles, one for each participants – STA, AP and AS. A role specifies a sequence of actions to be executed by an honest participant (or principal). A principal executing a particular instance of role is referred as a thread. It has a unique session id for each session.

Protocol proofs in PCL use formulas of the form ψ $[P]_X$ φ (modal formula). This means that if X starts from state where ψ is true then, in the resulting state security property φ will hold, irrespective of the actions of other participants, including attacker.The SWAS program is modelled in table 4. For simplicity, it is assumed that STA and AP posses MK. Msg1/2/3/4 are same as M1/2/3/4 of figure 1. Parts of messages not under consideration at a given moment are represented as msg1/2/3/4.

The session authentication is represented as matching conversations. Invariants for proving the session authentication are listed in table 5. $\Gamma_{SWAS,1}$ states that STA and AP derives PTK locally and do not reveals it. $\Gamma_{SWAS,2}$ states that honest supplicant do not act as authenticator or AS. It is trivial to follow that honest authenticator do not act as

For authenticator, authentication property is expressed as:

$$\theta_{SWAS}\ [SWAS : AUTH]_X\ \phi_{SWAS,auth}$$

$$\phi_{SWAS,auth} ::= \quad Honest(\hat{X}) \wedge Honest(\hat{Y}) \wedge Honest(\hat{Z}) \supset$$
$$\exists X.ActionsInOrder($$
$$Send(Y, \hat{Y}, \hat{X}, "msg1", ENC_{MK}(s1, r2), R, s, ENC_{\sigma}(r3, K_{AP-AS}), Hash_{MSK}(Msg1)) <$$
$$Receive(X, \hat{Y}, \hat{X}, "msg1", ENC_{MK}(s1, r2), R, s, ENC_{\sigma}(r3, K_{AP-AS}), Hash_{MSK}(Msg1,)) <$$
$$Send(X, \hat{X}, \hat{Z}, "msg2", Hash_{K_{AP-AS}}(Msg2), digsig_{AP}) <$$
$$Receive(Z, \hat{X}, \hat{Z}, "msg2", Hash_{K_{AP-AS}}(Msg2), digsig_{AP}) <$$
$$Send(Z, \hat{Z}, \hat{X}, "msg3", digsig_{AS}) <$$
$$Receive(X, \hat{Z}, \hat{X}, "msg3", digsig_{AS}) <$$
$$Send(X, \hat{X}, \hat{Y}, "msg4"))$$

For supplicant, it is expressed as:

$$\theta_{SWAS}\ [SWAS : SUPPL]_Y\ \phi_{SWAS,auth}$$

$$\phi_{SWAS,auth} ::= \quad Honest(\hat{X}) \wedge Honest(\hat{Y}) \supset$$
$$\exists Y.ActionsInOrder($$
$$Send(Y, \hat{Y}, \hat{X}, "msg1", ENC_{MK}(s1, r2), R, s, ENC_{\sigma}(r3, K_{AP-AS}), Hash_{MSK}(Msg1)) <$$
$$Receive(X, \hat{Y}, \hat{X}, "msg1", ENC_{MK}(s1, r2), R, s, ENC_{\sigma}(r3, K_{AP-AS}), Hash_{MSK}(Msg1,)) <$$
$$Send(X, \hat{X}, \hat{Y}, "msg4") <$$
$$Receive(Y, \hat{X}, \hat{Y}, "msg4"))$$

and for AS, it is expressed as:

$$\theta_{SWAS}\ [SWAS : AS]_Z\ \phi_{SWAS,auth}$$

$$\phi_{SWAS,auth} ::= \quad Honest(\hat{X}) \wedge Honest(\hat{Z}) \supset$$
$$\exists Z.ActionsInOrder($$
$$Send(X, \hat{X}, \hat{Z}, "msg2", Hash_{K_{AP-AS}}(Msg2), digsig_{AP}) <$$
$$Receive(Z, \hat{X}, \hat{Z}, "msg2", Hash_{K_{AP-AS}}(Msg2), digsig_{AP}) <$$
$$Send(Z, \hat{Z}, \hat{X}, "msg3", digsig_{AS}) <$$
$$Receive(X, \hat{Z}, \hat{X}, "msg3", digsig_{AS}))$$

supplicant (it can only receive Msg1 but do not send it). Thus, no principal performs the role of both authenticator and supplicant. $\Gamma_{SWAS,3}$ states that an honest AS do not send Msg1 and Msg4 and hence not act as supplicant or AS. $\Gamma_{SWAS,4}$ and $\Gamma_{SWAS,5}$ are used for stating that only an entity can own its own private key while $\Gamma_{SWAS,6}$ states that σ is shared only by STA and AS. SPMK asserts that only authenticator and supplicant has the PMK.

In this paper, the authentication property (Appendix B) for authenticator (AP) and supplicant (STA) is proved. Proof of authentication property for AS is not shown due to space consideration.

On execution of SWAS, the authenticator can reason as follows:

1. Since authenticator is honest ($\phi_{HONESTY,AP}$), it has its own sequence of receive and send i.e. it first receives Message 1, then sends Message 2. On receipt of Message 3, authenticator verifies it and then sends Message 4 to supplicant as shown in line (1) of the proof (Appendix B).
2. Since authenticator received and verified Message 3, there must be some entity $\hat{W}$, who puts its digital signature and sends out Message 3 (line2). This implies that $\hat{W}$ must know the signing key of AS (K_{AS}). Only AS can have its own signing key which means that $\hat{W}$ is AS (line3) i.e. Message 3 is indeed sent by AS. Further, this Message 3 is sent before it is received by authenticator (line4).

3. Similarly, for an honest AS ($\phi_{HONESTY,AS}$) it is proved that Message 2 is indeed sent by authenticator (digital signature using K_{AP}) and that this Message 2 is sent by authenticator before it is received by AS (line5-8).

4. Again, from $\phi_{HONESTY,AP}$ it is known (line 9) that authenticator receives Message1 containing R,s and MIC of the message (using MSK as key). Since authenticator received and verified Message 1, there must be some entity $\hat{W}$, who puts R,s and sends out Message 1 after calculating MIC (line10). As $\hat{W}$ calculates MIC using MSK, it may be supplicant or authenticator (line11). An authenticator cannot send Message3 to itself (line12).

5. As $\hat{W}$ also calculated R,s it can be either supplicant or AS. Since AS is honest and receives Message 2, $\hat{W}$ cannot be AS (line 13-14). This combined with point(4) above means Message3 is indeed sent by supplicant and this happens before authenticator receives it (line 15-16).

Based Upon these arguments, all the actions are matched in line (17). Hence, the authenticator can conclude that the security properties of session authentication are met.

On execution of SWAS, the supplicant can reason as follows:

1. Since supplicant is honest, it knows that it receive Message 4 only after it has sent Message 1 previously, represented in line (18) of the proof.

2. Since supplicant received and verified Message 4, there must be some entity $\hat{W}$, who computes and sends out Message 4. This implies that $\hat{W}$ must know the PTK used to encrypt (s1.r3) which is sent as part of Message 4. For this, $\hat{W}$ must have (s1,r3) and evaluated PTK utilizing MK and PMK, indicated in lines (19)-(22).

3. $\hat{W}$ must be either the supplicant Y itself or authenticator X, as these are the only two parties who have the PTK in the system, shown in line (23).

4. The supplicant, who knows the PTK and is honest, does not send Message 4 to itself. Thus, it must be the authenticator who have computed and sent Message 4. This occurs before the supplicant receives this Message 4; described in lines (24)-(26).

Based upon matching actions in line (26) the supplicant can conclude that the security properties of session authentication are met.

4. CONCLUSION

Secure WLAN Authentication Scheme (SWAS) achieves secure entity authentication and key generations for achieving secure communication. It utilizes cryptographic measures for improving the security in WLANs. The scheme strives to achieve the desired security properties like authentication, integrity, secrecy, availability, key freshness etc. All its packets are authenticated properly, thus reducing the attack chances. In this paper the authentication property of the SWAS using protocol composition logic (PCL) is proved. As future work, validation of scheme's other properties like secrecy and resistivity to DoS (Denial of Service) attacks will be done.

5. REFERENCES

[1] Johnson, H., Nilsson, A., Fu, J., Wu, S.F., Chen A., Huang, H. 2002. SOLA: A One bit Identity Authentication Protocol for Access Control in IEEE 802.11. In Proc. IEEE Global Telecommns Conference, GLOBECOM'02, pp. 768 – 772.

[2] Wu, F., Jonson, H., Nilson, A., 2004. SOLA: Lightweight Security for Access Control in IEEE 802.11, Wireless Security, May/June,10-16

[3] Wang, H., Velayutham, A., Guan, Y. 2003. A Lightweight Authentication Protocol for Access Control in IEEE 802.11. In Proc. IEEE Global Telecommunications Conference, GLOBECOM'03, pp. 1384 – 1388.

[4] Wang, H., Cardo, J., Guan, Y. 2005. Shepherd: A lightweight statistical authentication protocol for access control in wireless LANs. Computer Communications, 28, 1618-1630.

[5] Ren, K., Lee, H., Park, J., Kim, K. 2004. An Enhanced Lightweight Authentication Protocol for Access Control in Wireless LANs. In Proc. 4th International Conference on Networks, ICON'04, Daejeon, South Korea, pp. 444 – 450.

[6] Lee, Y-S., Chien, H-T., Tsai, W-N. 2009. Using Random Bit Authentication to defend IEEE 802.11 DoS attacks. Journal of Information Science and Engineering, 25, 1485-1500.

[7] Pepyne, D.L., Ho, Y-C., Zheng, Q. 2003. SPRiNG: Synchronized Random Numbers for Wireless Security. In Proc. IEEE Wireless Communications and Networking, WCNC'03, pp. 2027-2032.

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

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

[10] 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.

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

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

[13] IEEE Standard 802.1X-2001. IEEE Standard for Local and metropolitan area networks – Port-Based Network Access Control. June, 2001.

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

[15] Singh, R. Sharma, T.P. 2013. A Secure WLAN Authentication Scheme, in IEEK Transaction of Smart Processing and Computing, vol 2, no. 3.

[16] Lee, W-B., Yeh, C-K. 2005. A New Delegation-Based Authentication Protocol for Use in Portable

Communication Systems, IEEE Transaction on Wireless Commn. 4:1, 57-64.

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

[18] He, C. 2005. Analysis of Security Protocols for Wireless Networks, Ph.D. Dissertation.

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

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

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

Appendix A

Fragment of the PCL Proof System

Only the axioms used in proving the SWAS properties are mentioned here. A comprehensive list of axioms is present in appendix A at [18].

Honesty Rule 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.

A. Axioms for proving SWAS authentication property

AA1 $\phi\ [a]_X$ a

AA4 ϕ [$a_{1;}a_{2;...;}$ a_k]$_x a_1 \wedge ... \wedge a_{k-1} < a_k$

ARP Receive (X, $p(x)$) [match q(x) as q(t)]$_X$ Receive (X, $p(t)$)

HASH1 Computes (X, $HASH_K(x)$) $\supset$
Has (X,x) $\wedge$ Has(X,K)

HASH4 Has (X, $HASH_K(x)$) $\supset$
Computes (X, $HASH_K(x)$) $\vee$
$\exists Y,m$ .Computes (Y, $HASH_K(x)$)
$\wedge$Send (Y,m) $\wedge$ Contains (m, $HASH_K(x)$)

ENCO [m' : = symenc $m_,$ k;]$_x$ SymEnc (X,m,k)

ENC3 SymEnc(X,m,k) $\supset$ Has (X,k) $\wedge$ Has (X,m)

ENC4 SymDec(X,E[k] (m),k) $\supset$ $\exists Y$.SymEnc(Y,m,k)

Appendix B

Proof of authentication property, SWAS:

$\phi_{HONESTY,AP}$

$$\begin{aligned}&\theta_{SWAS}\\&[SWAS : AUTH]_X\\&Honest(\hat{X}) \wedge Honest(\hat{Y}) \supset\\&Receive(X,\hat{Y},\hat{X},"msg1",Hash_{MSK}(Msg1)) <\\&Send(X,\hat{X},\hat{Z},"msg2",Hash_{K_{AP-AS}}(Msg2),digsig_{AP}) <\\&Receive(X,\hat{Z},\hat{X},"msg3",digsig_{AS}) <\\&Send(X,\hat{X},\hat{Y},"msg4")\end{aligned}\tag{D.1}$$

(1)

$$\begin{aligned}&\theta_{SWAS}\\&[SWAS : AUTH]_X\\&Receive(X,\hat{Z},\hat{X},"msg3",digsig_{AS}) \supset\\&\exists W.Computes(W,digsig_{AS}) \wedge\\&Send(W,digsig_{AS}) \wedge\\&(Send(W,digsig_{AS}) <\\&Receive(X,\hat{Z},\hat{X},"msg3",digsig_{AS}))\end{aligned}\tag{D.2}$$

$\Gamma_{SWAS,4}$

$$\begin{aligned}&\theta_{SWAS}\\&[SWAS : AUTH]_X\\&Computes(W,digsig_{AS}) \wedge Send(W,digsig_{AS}) \supset\\&Has(\hat{W},K_{AS}) \supset \hat{W} = \hat{Z}\end{aligned}\tag{D.3}$$

(2),(3)

$$\theta_{SWAS}$$

$[SWAS : AUTH]_X$

$Honest(\hat{X}) \wedge Honest(\hat{Z}) \supset$

$(Send(Z, \hat{Z}, \hat{X}, "msg3", digsig_{AS}) <$

$Receive(X, \hat{Z}, \hat{X}, "msg3", digsig_{AS}))$ (D.4)

$\phi_{HONESTY,AS}$ θ_{SWAS}

$[SWAS : AUTH]_X$

$Honest(\hat{X}) \wedge Honest(\hat{Z}) \supset$

$Receive(Z, \hat{X}, \hat{Z}, "msg2") <$

$Send(Z, \hat{Z}, \hat{X}, "msg3")$ (D.5)

(5) θ_{SWAS}

$[SWAS : AUTH]_X$

$Receive(Z, \hat{X}, \hat{Z}, "msg2", digsig_{AP}) \supset$

$\exists W.Computes(W, digsig_{AP}) \wedge$

$Send(W, digsig_{AP}) \wedge$

$(Send(W, digsig_{AP}) <$

$Receive(Z, \hat{X}, \hat{Z}, "msg2", digsig_{AP}))$ (D.6)

$(6), \Gamma_{SWAS,5}$ θ_{SWAS}

$[SWAS : AUTH]_X$

$Computes(W, digsig_{AP}) \wedge Send(W, digsig_{AP}) \supset$

$Has(\hat{W}, K_{AP}) \supset \hat{W} = \hat{X}$ (D.7)

(6),(7) θ_{SWAS}

$[SWAS : AUTH]_X$

$Honest(\hat{X}) \wedge Honest(\hat{Z}) \supset$

$(Send(X, \hat{X}, \hat{Z}, "msg2", digsig_{AP}) <$

$Receive(Z, \hat{X}, \hat{Z}, "msg2", digsig_{AP}))$ (D.8)

$\phi_{HONESTY,AP}$ θ_{SWAS}

$[SWAS : AUTH]_X$

$Receive(X, \hat{Y}, \hat{X}, "msg1", R, s, MIC_{MSK}(Msg1))$

$Send(X, \hat{X}, \hat{Y}, "msg2", Hash_{K_{AP-AS}}(Msg2), digsig_{AP}) <$

$Receive(X, \hat{X}, \hat{Y}, "msg3", digsig_{AS})) <$

$Send(X, \hat{X}, \hat{Y}, "msg4")$ (D.9)

(9) θ_{SWAS}

$[SWAS : AUTH]_X$

$Receive(X, \hat{Y}, \hat{X}, "msg1", R, s, Hash_{MSK}(Msg1)) \supset$

$\exists W.Computes(W, R, s) \wedge$

$Computes(W, Hash_{MSK}(Msg1)) \wedge$

$Send(W, "msg1", R, s, Hash_{MSK}(Msg1)) \wedge$

$(Send(W, "msg1", R, s, Hash_{MSK}(Msg1)) <$

$Receive(X, \hat{Y}, \hat{X}, R, s, Hash_{MSK}(Msg1))$ (D.10)

(10) θ_{SWAS}

$[SWAS : AUTH]_X$

$$Computes(W, Hash_{MSK}(msg1, R, s)) \supset$$

$$Has(\hat{W}, MSK) \supset Has(\hat{W}, MK) \supset \hat{W} = \hat{Y} \vee \hat{W} = \hat{X}$$

(D.11)

(11)

$$\theta_{SWAS}$$
$$[SWAS : AUTH]_{X}$$

$$Honest(\hat{X}) \wedge Receive(X, \hat{Y}, \hat{X}, "msg1", R, s, Hash_{MSK}(msg1,$$

$$R, s)) \supset \hat{W} \neq \hat{X}$$

(D.12)

$(10), \Gamma_{SWAS,6}$

$$\theta_{SWAS}$$
$$[SWAS : AUTH]_{X}$$
$$Computes(W, R, s) \supset$$

$$Has(\hat{W}, \sigma) \supset \hat{W} = \hat{Y} \vee \hat{W} = \hat{Z}$$

(D.13)

$(13), \Gamma_{SWAS,3}$

$$\theta_{SWAS}$$
$$[SWAS : AUTH]_{X}$$

$$Honest(\hat{Z}) \wedge Receive(Z, \hat{X}, \hat{Z}, "msg2") \supset \hat{W} \neq \hat{Z}$$

(D.14)

(12), (14)

$$\theta_{SWAS}$$
$$[SWAS : AUTH]_{X}$$

$$Honest(\hat{X}) \wedge Honest(\hat{Y}) \wedge Honest(\hat{Z}) \supset$$
$$\exists W.Computes(W, "msg1", R, s, Hash_{MSK}(msg1, R, s)) \wedge$$

$$Send(W, "msg1", R, s, Hash_{MSK}(msg1, R, s)) \wedge \hat{W} = \hat{Y}$$

(D.15)

(10), (15)

$$\theta_{SWAS}$$
$$[SWAS : AUTH]_{X}$$

$$Honest(\hat{X}) \wedge Honest(\hat{Y}) \wedge Honest(\hat{Z}) \supset$$

$$Send(Y, \hat{Y}, \hat{X}, "msg1", R, s, Hash_{MSK}(Msg1)) <$$

$$Receive(X, \hat{Y}, \hat{X}, "msg1", R, s, Hash_{MSK}(Msg1))$$

(D.16)

(1), (4), (8), (16)

$$\theta_{SWAS}$$
$$[SWAS : AUTH]_{X}$$

$$Honest(\hat{X}) \wedge Honest(\hat{Y}) \wedge Honest(\hat{Z}) \supset$$

$$Send(Y, \hat{Y}, \hat{X}, "msg1", R, s, Hash_{MSK}(Msg1)) <$$

$$Receive(X, \hat{Y}, \hat{X}, "msg1", R, s, Hash_{MSK}(Msg1)) <$$

$$Send(X, \hat{X}, \hat{Z}, "msg2", Hash_{K_{AP-AS}}(Msg2), digsig_{AP}) <$$

$$Receive(Z, \hat{X}, \hat{Z}, "msg2", digsig_{AP}) <$$

$$Send(Z, \hat{Z}, \hat{X}, "msg3", digsig_{AS}) <$$

$$Receive(X, \hat{Z}, \hat{X}, "msg3", digsig_{AS}) <$$

$$Send(X, \hat{X}, \hat{Y}, "msg4")$$

(D.17)

AA1, AA4

$$\theta_{SWAS}$$
$$[SWAS : SUPPL]_{Y}$$

$$Send(Y, \hat{Y}, \hat{X}, "msg1") <$$

$$Receive(Y, \hat{X}, \hat{Y}, "msg4")$$

(D.18)

ARP,AA4

$$\begin{array}{l}\theta_{SWAS}\\ [receive(Y,\hat{X},\hat{Y},w);\\ match\ \ w/"msg4"]_{Y}\\ Receive(Y,\hat{X},\hat{Y},"msg4")\supset\\ \exists W.Computes(W,"msg4")\wedge\\ (Send(W,"msg4")<\\ Receive(Y,\hat{X},\hat{Y},"msg4"))\end{array}$$

(D.19)

ENC3

$$\begin{array}{l}Computes(W,"msg4")\equiv Computes(W,ENC_{PTK}(s1,r3))\equiv\\ Has(\hat{W},PTK)\wedge Has(\hat{W},"s1,r3")\end{array}$$

(D.20)

HASH4

$$\begin{array}{l}Has(\hat{W},PTK)\equiv Has(\hat{W},PRF(s1,PMK))\supset\\ Computes(W,PRF(s1,PMK))\vee\\ \exists X,m.Computes(X,PRF(s1,PMK))\wedge\\ Send(X,m)\wedge Contains(m,ENC_{PTK}(s1,r3))\end{array}$$

(D.21)

$(20),(21),\Gamma_{SWAS,1}$

$$\begin{array}{l}\theta_{SWAS}\\ [receive(Y,\hat{X},\hat{Y},w);\\ match\ \ w/"msg4"]_{Y}\\ Has(\hat{W},PTK)\equiv Has(\hat{W},ENC_{PTK}(s1,r3))\end{array}$$

(D.22)

(22),SPMK

$$\begin{array}{l}\theta_{SWAS}\\ [receive(Y,\hat{X},\hat{Y},w);\\ match\ \ w/"msg4"]_{Y}\\ Honest(X)\wedge Honest(Y)\supset Computes(W,ENC_{PTK}(s1,r3))\supset\\ Has(\hat{W},PTK)\supset Has(\hat{W},PMK)\supset Has(\hat{W},MSK)\supset Has(\hat{W},MK)\supset\ \ \hat{W}=\hat{X}\vee\hat{W}=\hat{Y}\end{array}$$

(D.23)

$AA1,\Gamma_{SWAS,2}$

$$\begin{array}{l}\theta_{SWAS}\\ [send(Y,\hat{Y},\hat{X},"msg1")]_{Y}\\ Honest(\hat{Y})\wedge Send(Y,\hat{Y},\hat{X},"msg1",Hash_{MSK}(Msg1))\\ \supset\hat{W}\neq\hat{Y}\end{array}$$

(D.24)

(23),(24)

$$\begin{array}{l}\theta_{SWAS}\\ [SWAS:SUPPL]_{Y}\\ Honest(\hat{X})\wedge Honest(\hat{Y})\supset\\ \exists W,m.Computes(W,ENC_{PTK}(s1,r3))\wedge\\ Contains(m,ENC_{PTK}(s1,r3))\wedge\\ Send(W,m)\wedge\hat{W}=\hat{X}\end{array}$$

(D.25)

(19),(25)

$$\begin{array}{l}\theta_{SWAS}\\ [SWAS:SUPPL]_{Y}\\ Honest(\hat{X})\wedge Honest(\hat{Y})\supset\\ Send(X,\hat{X},\hat{Y},"msg4")<\\ Receive(Y,\hat{X},\hat{Y},"msg4")\end{array}$$

(D.26)