# Security Analysis of the Authentication Modules of Chinese WLAN Standard and Its Implementation Plan★

Xinghua Li[1,2], Jianfeng Ma[1], SangJae Moon[2]

[1] Key Laboratory of Computer Networks and Information Security(Ministry of Education)，
Xidian University, Xi'an 710071, China

[2] Mobile Network Security Technology Research Center, Kyungpook National University, Sankyuk-dong,
Buk-ku, Daegu 702-701, Korea

**Abstract.** With the Canetti-Krawczyk (CK) model, we analyze the authentication module WAIs in the Chinese WLAN national security standard WAPI and its implementation plan respectively. The security weaknesses of WAI in the original WAPI are presented; then WAI in the implementation plan is proved secure in the CK model; at last we point out how the implementation plan overcomes the security weaknesses in the original WAPI.

**Keywords:** WLAN; WAPI; Key-agreement protocol; Canetti-Krawczyk model

## 1 Introduction

The Chinese WLAN standard WAPI (GB 15629.11-2003) [1], the first issued Chinese standard in the field of WLAN, has been formally implemented since November 1, 2003. WAPI (WLAN Authentication and Privacy Infrastructure) is composed of two parts: WAI (Wireless Authentication Infrastructure) and WPI (Wireless Privacy Infrastructure). They realize the identity authentication and data encryption, respectively. In March of 2004, China IT Standardization Technical Committee drafted out a new version, WAPI implementation plan [2], which improves the original standard WAPI. Compared with the original standard, the greatest change the implementation plan made lies in the WAI module.

As a national standard which is about to be deployed and implemented in a big scale, its security is undoubtedly the focus. But as far as we know, up to now, there are no articles that systemically analyze the security of WAPI and its implementation plan, which is imperfect for a national standard. This contribution discusses the security of WAPI and its implementation plan with the Canetti-Krawczyk (CK) model [3]. It has three contributions. (1) The security weaknesses of WAI in WAPI are given. (2) The WAI module in the implementation plan is proved secure in the CK model. (3) How the

★ Research supported by the National Natural Science Foundation of China (Grant No.90204012, No. 60633020), the National "863" High-tech Project of China (Grant No. 2002AA143021), the Excellent Young Teachers Program of Chinese Ministry of Education, the Key Project of Chinese Ministry of Education, and the University IT Research Center Project of Korea.

implementation plan overcomes the security weaknesses of the original WAPI is pointed out. The analysis results can help us understand the necessity of the implementation plan and enhance the confidence of it. At the same time, as a case study, their analysis is helpful for the design of a secure key-agreement protocol.

The rest of the paper is organized as follows. In Section 2, we give an overview of the CK model. In Section 3, WAIs in the WAPI and its implementation plan are introduced. In Section 4, WAI in WAPI is analyzed and its security weaknesses are presented. We analyze WAI in the implementation plan in Section 5. In Section 6, we point out how the implementation plan overcomes the security weaknesses in the original national standard. This paper is concluded in Section 7.

## 2 The CK Model

At present, the CK model is a very popular formal methodology for the analysis of key-agreement protocols [4]. In this section, we give a brief description of the CK model.

A key-exchange (KE) protocol is run in a network of interconnected parties where each party can be activated to run an instance of the protocol called a session. A KE session is a quadruple (*A*, *B*, *X*, *Y*) where *A* is the identity of the holder of the session, *B* the peer, *X* the outgoing messages in the session, and *Y* the incoming messages. The session (*B*, *A*, *Y*, *X*) (if it exists) is said to be matching to the session (*A*, *B*, *X*, *Y*). Matching sessions play a fundamental role in the definition of security [3].

### 2.1 Attacker Model

The attacker is modeled to capture realistic attack capabilities in open networks, including the control of communication links and the access to some of the secret information used or generated in the protocol. The attacker, denoted $\mathcal{M}$, is an active “man-in-the-middle” adversary with full control of the communication links between parties. $\mathcal{M}$ can intercept and modify messages sent over these links, it can delay or prevent their delivery, inject its own messages, interleave messages from different sessions, etc. (Formally, it is $\mathcal{M}$ to whom parties hand their outgoing messages for delivery.) $\mathcal{M}$ also schedules all session activations and session-message delivery. In addition, in order to model potential disclosure of secret information, the attacker is allowed to access to secret information via session exposure attacks of three types: state-reveal queries, session-key queries, and party corruption [3].

### 2.2 Definition of Session-Key Security

In addition to the regular actions of the attacker $\mathcal{M}$ against a key-exchange protocol $\pi$, he can perform a *test session query.* That is, at any time during its run, $\mathcal{M}$ is able to choose, a *test-session* among the sessions that are completed, unexpired and unexposed at the time. Let $k$ be the value of the corresponding session key. We toss a coin $b$, $b \xleftarrow{R} \{0,1\}$. If $b$=0 we provide $\mathcal{M}$ with the value $k$. Otherwise we provide $\mathcal{M}$ with a value $r$ randomly chosen from the probability distribution of keys generated by protocol $\pi$. The attacker $\mathcal{M}$ is not allowed state-reveal queries, session-key queries, or party corruption on the test-session or its matching session. At the end of its run, $\mathcal{M}$ outputs a bit $b'$ (as its guess for $b$).

An attacker that is allowed test-session queries is referred to as a KE-adversary.

**Definition 1 Session-key Security:** *A key-exchange protocol $\pi$ is called Session-key secure (or SK-secure) if the following properties hold for any KE-adversary $\mathcal{M}$.*

1. *Protocol $\pi$ satisfies the property that if two uncorrupted parties complete matching sessions then they both output the same key; and*
2. *the probability that $\mathcal{M}$ guesses correctly the bit b(i.e., outputs b'=b) is no more than 1/2 plus a negligible fraction $\varepsilon$ in the security parameter. $\varepsilon$ is called "advantage".*

## 3 WAIs in WAPI and Its Implementation Plan

WAI adopts port-based authentication architecture that is identical with IEEE 802.1X. The whole system is composed of mobile guest STA, Access Point (AP), and Authentication Service Unit (ASU).

### 3.1 WAI in WAPI

The interaction procedure of WAI in the original national standard WAPI is shown in Fig.1. From this figure, we can see that WAI is composed of two parts: certificate authentication and key agreement.

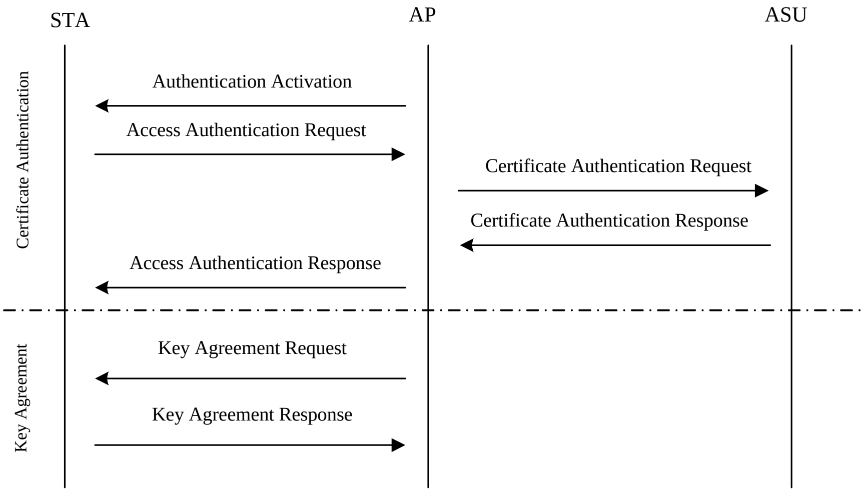

STA
AP
ASU
Certificate Authentication
Authentication Activation
Access Authentication Request
Certificate Authentication Request
Certificate Authentication Response
Access Authentication Response
Key Agreement
Key Agreement Request
Key Agreement Response

**Figure 1.** *WAI in WAPI*

### (1) Certificate authentication

In this process, STA sends its public key certificate and access request time to AP in the Access Authentication Request. AP sends its certificate, STA's certificate, STA's access request time and its signature on them to ASU in Certificate Authentication Request. After ASU validates AP's signature and the two certificates, it sends the certificates validation result, STA's access request time and ASU's signature on them to STA and AP.

### (2) Key agreement

First, STA and AP negotiate the cryptography algorithms. Then, they respectively generate one random value $r_1$ and $r_2$. These random values are encrypted with the peer's public key and sent to each other. Both parties decrypt the encrypted random values and derive the session key $K=r_1 \oplus r_2$. The key agreement process is shown in Fig.2, where $ENC(\ )$is the encryption function, $PK_{AP}$ and $PK_{STA}$ are AP and STA's public key respectively.

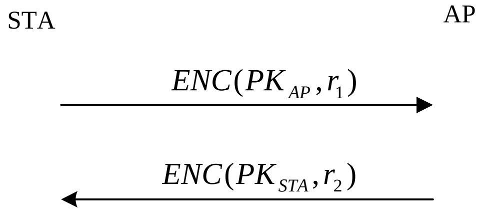


**Figure 2.** *The key agreement in the WAI of WAPI*

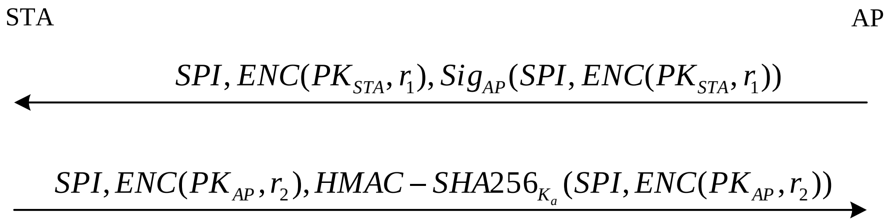
### 3.2 WAI in the Implementation Plan

In the framework, WAI in the implementation plan is same as that of the original WAPI, and it is also composed of certificate authentication and key agreement. Compared with the original standard WAPI, the implementation plan remains unchanged in the certificate authentication, but makes rather big improvement in the key agreement. The new key agreement protocol is shown in Fig.3. It is different from the original one in the following points:

(1) In the implementation plan, the Key Agreement Request has to be initiated by AP. At the same time, the secure parameter index SPI, AP's signature on the encrypted random value and SPI are included in this request. The signature algorithm is ECDSA.

(2) In the Key Agreement Response, SPI and the STA's message authentication code on encrypted random and SPI are included. The message authentication code is computed through HMAC-SHA256 algorithm.

(3) The keys derivation method is different. STA and AP first calculate the host key $k$= $r_1 \oplus r_2$, then extend $k$ with KD-HMAC-SHA256 algorithm to get the session key $k_d$, the authentication key $k_a$ and integration check key.


STA
AP
$SPI, ENC(PK_{STA}, r_1), Sig_{AP}(SPI, ENC(PK_{STA}, r_1))$
$SPI, ENC(PK_{AP}, r_2), HMAC-SHA256_{K_a}(SPI, ENC(PK_{AP}, r_2))$

*SPI=the MAC of the STA||the BSSID of the AP||the time of authentication request*

**Figure 3.** *The key agreement protocol in WAI of the implementation plan*

## 4 The Security Weaknesses of WAI in WAPI

The WAI module in the original WAPI has several security weaknesses as follows.

**(1)Its key agreement protocol can’t resist the unknown key-share (UKS) attack [5, 6].**

We assume that an attacker $E$ gets a certificate where his public key $PK_E$ is same as $PK_{STA}$. (In many practical settings, the Certificate Authority (CA) does not require a proof-of-possession of the corresponding private key from a registrant of a public key [7], so an attacker $E$ can get a certificate from the CA in which his public key is same as STA’s.) In addition, in the certificate authentication process, ASU just verifies the authenticity and validity of a certificate, so $E$ also can pass the certification authentication. Then he can launch the unknown-key share attack in the key agreement. When STA sends the first message $ENC(PK_{AP}, r_1)$, $E$ forwards this message to AP and claims that this message is from $E$. Then AP replies with $ENC(PK_E, r_2)$. $E$ forwards this message to STA. When the protocol completes, STA thinks that he agreed upon a key with AP, while AP thinks that he negotiated a key with $E$. And these two keys are same. So, the attacker $E$ succeeds in the unknown-key share attack.

Let’s analyze this attack in the CK model. In the attack above, the KE-adversary chooses the session in STA as the test session and expose the session in AP (because these two sessions are not matching sessions, the session in AP can be exposed). Because STA and AP get a same session key, the KE-adversary can completely get the session key of the test session. According to Definition 1, this protocol is not SK-secure. And [8] can be referred to for the consequences of this attack.

**(2)Its key agreement protocol can’t resist key-compromise impersonation (KCI) attack.**

Let’s analyze this attack in the CK model. First, we assume that STA’s private key is compromised and the attacker chooses the session in STA as the test session after STA complete the matching sessions with AP. The attacker can first corrupt another mobile guest STA' and impersonates him to send message $ENC(PK_{AP}, r_1)$ to AP. We denote the session between STA' and AP as SID'. When AP receives this message from STA', he chooses another random value $r_3$ and responds with $ENC(PK_{STA'}, r_3)$. AP computes its session key of SID' $k' = r_1 \oplus r_3$. The attacker can expose this session and get $k'$ (this session is not the matching session of the test session). In addition, the attacker can decrypt $ENC(PK_{STA'}, r_3)$ to get $r_3$. Thus he can get $r_1 = k' \oplus r_3$. In addition, the attacker can also decrypt $ENC(PK_{STA}, r_2)$ to get $r_2$ . Then he can get the session key of the test session: $k=r_1 \oplus r_2$. Thus the attacker can impersonate AP to STA. According to Definition 1, this protocol is not SK-secure.

**(3) It does not realize the explicit identity authentication of STA and perhaps lead to the faulty charge.**

From the WAI process, we can see that it does not realize the explicit identity authentication of STA to AP. An attacker can pass the certificate authentication and access the networks only if he gets a legal user's certificate, which will lead to the faulty charge if the networks charge the fee according to the access time.

## 5 The Security Analysis of WAI in WAPI Implementation Plan

In the certificate authentication, AP makes signature in the certificate authentication request, and ASU makes signature in the certificate authentication response. Both these signatures include STA's access request time which ensures the freshness of the signatures. Therefore ASU can authenticate AP's identity and STA can authenticate ASU's identity. In addition, STA trusts ASU. So STA can authenticate the identity of AP after the certificate authentication. At the same time, AP authenticates the certificate provided by STA.

The key agreement protocol in WAI of implementation plan is denoted by $\pi$ . In the following, we will prove that $\pi$ is SK-secure without PFS [9]. That is, the protocol is SK-secure, but does not provide perfect forward secrecy of the session keys. In order to prove that $\pi$ is SK-secure, we define a "game". The game designed is very similar to the one in [11], therefore authors can refer to [11] for details.

### 5.1 Security Analysis of Key agreement Protocol in WAI

According to Definition 1, in order to prove that $\pi$ is SK-secure, we have to argue that it can meet two requirements. The first one is that STA and AP can get a same session key after they complete matching sessions. The second one is that $\mathcal{B}$ cannot distinguish the session key $k_d$ from a random value with a non-negligible advantage. In the following, we will prove that $\pi$ can meet these two requirements.

**Lemma 1.** *If the encryption scheme ENC is secure against the CCA2 attack, then at the end of protocol* $\pi$*, STA and AP will complete matching sessions and get a same session key.*[11]

**Lemma 2.** *If the encryption scheme ENC is secure against the CCA2 attack, the attacker cannot distinguish the session key* $k_d$ *from a random value with a non-negligible advantage.* [11]

**Theorem 1.** *If the encryption scheme ENC adopted is secure against CCA2 attack, then* $\pi$ *is SK-secure without PFS.*

*Proof.* According to Lemma 1, Lemma 2 and Definition 1, we can get that the protocol $\pi$ is SK-secure.

In addition, if the private keys of STA and AP are compromised, the attacker can get the random values exchanged and can work out all the session keys that have been agreed about. Thus this protocol cannot provide PFS. So we can get that the key-agreement protocol is SK-secure without PFS. □

## 6 The Implementation Plan Overcomes the Weaknesses of the Original WAPI

We know that WAI in the original WAPI has some security weaknesses. But WAI in the implementation plan is secure in the CK model, and according to [13], we get that the WAI module of the implementation plan can resist KCI attack and UKS attack. In the following, we will analyze how the implementation plan overcomes the security weaknesses in the original WAPI.

**(1) The key agreement protocol in the implementation plan can resist UKS attack.**

In the implementation plan, even though the attacker $\mathcal{B}$ gets a certificate in which his public key is same as STA's or AP's, he cannot launch the UKS attack. Because the implementation plan requires that the key agreement request be sent by AP, STA just accepts the request from AP. So, $\mathcal{B}$ can just launch the UKS attack against the AP (i.e., AP thinks that he agrees upon a key with $\mathcal{B}$, but in fact he negotiates a key with STA, while STA correctly thinks that he negotiates a key with AP), that is, $\mathcal{B}$ just can forwards the key agreement request message for him to STA. But in this request, AP's signature includes SPI which includes the MAC address of the $\mathcal{B}$, so STA will not accept this request forwarded from $\mathcal{B}$. Therefore the key agreement protocol in WAI of implementation plan can resist the UKS attack.

From the analysis above, we can see that the essential reasons that WAI in the implementation plan can resist the UKS attack are that: (1) the implementation plan requires that the key agreement request be sent from AP; (2) AP's signature includes SPI which includes the destination entity's address.

**(2) The key agreement protocol in the WAI of the implementation plan can resist the KCI attack.**

KCI attacks for the protocol $\pi$ have two manners. The first one is that AP's private key is compromised and the attacker can impersonate STA to AP. The second one is that STA's private key is compromised and the attacker can impersonate AP to STA. In the following, we will discuss these two cases respectively.

If AP's private key is compromised, the attacker can decrypt $ENC(PK_{AP}, r_2)$ to get $r_2$. In order to get $r_1$, he just has two possible methods: (1) attacks the encryption algorithm *ENC*; (2) impersonates other entity to establish another session with STA, and sends $ENC(PK_{STA},r_1)$ to STA, then the attacker exposes this session and gets $r_1$ through some computations. But neither of these two methods is feasible. For the first method, we know that if the encryption algorithm *ENC* is CCA2 secure, the attacker cannot get $r_1$ from the attack of this algorithm directly. As for the second method, the implementation plan requires the key agreement request be sent by AP, and the attacker cannot forge AP's signature, so the attacker cannot impersonate other entity to establish another session with STA. Therefore the attacker cannot get $r_1$. Then he still cannot get the host key $k$ and session key $k_d$ .

If STA's private key is compromised, the attacker can decrypt $ENC(PK_{STA},r_1)$ to get $r_1$. In order to get session key $r_2$, he just has two possible methods: (1) attacks the encryption algorithm *ENC* directly to get $r_2$; (2) impersonates another mobile guest STA' to establish a new session with AP and sends it $ENC(PK_{AP}, r_2)$ in the key agreement acknowledgement. From the analysis above we get that the first method is infeasible. As for the second method, because $r_2$ and the host key $k$ are just the ephemeral values, we assume that they are not the session states of AP. Therefore, the session states of the new session in AP are just the session key $k_d^*$, the message authentication key $k_a^*$ and the message integration key. The attacker cannot get any information about $r_2$ from these session states because these three keys are the hash values of the host key $k^*$. Therefore the attacker cannot get $r_2$ either. (If the session key is not the hash value of $k^*$, the attacker can get $k^*$, futher can get $r_2$.) So the attacker still cannot get the host key $k$ and the session key $k_d$ .

As a whole, the essential reasons that the key agreement protocol can resist KCI attack are that: (1) the implementation plan requires that the key agreement request be sent by AP; (2) the session key in the implementation plan is derived through the hash function.

**(3) The WAI module in the implementation plan realizes the mutual explicit identity authentication between STA and AP, which can withstand faulty charge.**

For AP, $\pi$ is an explicit key authentication protocol [12]. So AP can authenticate the identity of STA at the end of WAI. At the same time, STA can authenticate the identity of AP in the certificate authentication. Therefore WAI in the implementation plan realizes the mutual explicit identity authentication between AP and STA. Therefore it can withstand faulty charge.

## 7 Conclusion

With the CK model, this paper analyzes the authentication module WAIs in the original WAPI and its implementation plan. The security weaknesses of WAI in the original WAPI are presented: its key agreement protocol cannot resist the UKS and KCI attacks; it does not realize the explicit identity authentication and can lead to the faulty charge. Then the WAI module in WAPI implementation plan is analyzed. We prove that if the encryption scheme *ENC* adopted is secure against the CCA2 attack, then the WAI module in the implementation plan is SK-secure without PFS. At last we analyze how the implementation plan overcomes the security weaknesses in the original WAPI. Compared with the original standard WAPI, the security of the implementation plan is improved greatly.

## References

1. National Standard of the People’s Republic of China. GB 15629.11-2003 "Information technology-Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". 2003.
2. National Standard of the People’s Republic of China. Guide for GB 15629.11-2003 "Information technology—Telecommunications and information exchange between systems-Local and metropolitan area networks- Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications." and GB 15629.1102-2003 "Information technology —Telecommunications and information exchange between systems —Local and metropolitan area networks—Specific requirements-Part 11: Wireless LAN Medium access control (MAC) and physical layer(PHY) Specifications: Higher-Speed Physical layer Extension in the 2.4 GHz Band". 2004.3
3. Canetti, R., Krawczyk, H. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channel. Eurocrypt 2001, *Lecture Notes in Computer Science*, Springer-Verlag, Vol. 2045. (2001) 453-474
4. Boyd, C., Mao, W., Paterson, K. Key Agreement using Statically Keyed Authenticators. Applied Cryptography and Network Security 2004. 2004: *Lecture Notes in Computer Science*, Springer-verlag, Vol. 3089 (2004) 248 -262
5. Burton S., Kaliski J R. An unknown key-share attack on the MQV key agreement protocol. *ACM transactions on Information and System Security*, 4(3):275-288, August 2001
6. Blake-Wilson S., Johnson D. and Menezes A. Key agreement protocols and their security analysis. *In Proceedings of the sixth IMA International*

*Conference on Cryptography and Coding, LNCS 1355,* pages 30-45, Springer-Verlag, 1997

7. Krawczyk H., HMQV: A High-Performance Secure Diffie-Hellman Protocol. *Advances in Cryptology–CRYPTO 2005: 25th Annual International Cryptology Conference. LNCS 3621*, pages 546-566, Springer-Verlag, 2005
8. Diffie W, Oorschot P V, Wiener M. Authentication and authenticated key exchanges, *Designs, Codes and Cryptography*, 2, 1992:107-125
9. G ü ther, C.G. An identity-based key-exchange protocol. *Advances in Cryptology-EUROCRYPT'89, Lecture Notes in Computer Science*, Springer-Verlag, Vol.434 (1990) 29-37
10. Mao, W. Modern Cryptography: Theory and Practice. Hewlett-Packard Company. Publisher: Prentice Hall PTR 2003
11. Li, X., Moon, S., Ma, J, On the Security of the Authentication Module of Chinese WLAN Standard Implementation Plan. *The 4th International Conference on Applied Cryptography and Network Security (ACNS06). Lecture Notes in Computer Science*. Springer-Verlag. Vol.3989(2006) 340-348
12. Menezes, A., van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, 1996
13. Li, X., Ma, J., Moon, S.: On the Security of Canetti-Krawczyk Model. *2005 International Conference on Computational Intelligence and Security, Lecture Notes in Artificial Intelligence,* Springer-Verlag, Vol.3802 (2005) 356-363