# SECURITY ANALYSIS OF LWAPP

Zhaohui Cheng, Manos Nistazakis and Richard Comley
*School of Computing Science, Middlesex University,*
*White Hart Lane, London N17 8HR, UK*
{m.z.cheng,e.nistazakis,r.comley}@mdx.ac.uk

Last revised on 7 April 2004

**Abstract** Light Weight Access Point Protocol (LWAPP) is a new protocol being designed to make communications between access points and wireless switches automatic. This protocol allows a router or switch to interoperably control and manage a collection of wireless access points, so as to move some of the loading due to Wi-Fi processes and function complexity to the centralized wireless switches or routers. In this report we analyze the security design of the protocol, address some possible attacks and present some fix solutions. Moreover the proposed key-transport protocols have their own interest and can be used in other scenarios.

**Keywords:** LWAPP, Security, Denial of Service, Key Transport Protocol

## 1. Introduction

With the development of Wireless Local Area Network (WLAN) technology, more and more vendors are now providing Wi-Fi products. But there exists two philosophically opposite approaches to provide wireless access. One is using 'heavy' Access Points (AP) with a lot of intelligence but a hefty price tag. The benefit of having these access points is their ability to communicate directly with the existing routers, and thus support robust applications. This approach is supported by companies such as Cisco, due to their domination of the router market. By contrast, the 'lightweight' group believes that taking intelligence out of APs and putting it into switches or routers will allow networks to scale and will drive down the overall cost of deployment. The later method using cheaper APs but new types of wireless switches is preferred by vendors who want to spearhead into this huge market by making use of the low price advantage against Cisco's market influence. The problem with this

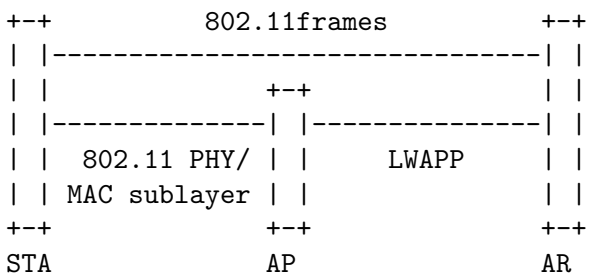


*Figure 1.* LWAPP Architecture[7]

still-emerging sector is the lack of interoperability. Unless vendors have completed a bilateral interoperability exercise, wireless switches from one vendor won't work with APs of another, and so on. This motivates the birth of LWAPP [7]. The standardization of LWAPP is led by two major vendors: Airespace and NTT Docomo. The latest memo draft 3 was recently published, although it still hasn't been accepted as an IETF official draft. For a wireless network standard, security is always a major concern. In this report we briefly analyze the security aspect of LWAPP and the paper is organized as follows. Section 2 is a general introduction of the architecture and principle of LWAPP. A security analysis of the protocol including some feasible attacks and fixing suggestions are presented in section 3 and we draw the conclusions in the final part.

## 2. LWAPP's Principle

LWAPP is a generic protocol defining how lightweight access points communicate with Access Routers (AR). It assumes a network configuration that consists of multiple APs connected either via layer 2 (Ethernet), or layer 3 (IP) to an AR. The APs can be considered as remote RF interfaces, being controlled by the AR. The AP forwards all 802.11 frames received from mobile stations (STA) to the AR via the LWAPP protocol, which processes the frames, if LWAPP works on layer 2. Similarly, packets from authorized mobiles are forwarded by the AP to the AR via this protocol, if the protocol works on layer 3. These forwarding operations between APs and ARs are accomplished according to a LWAPP transport layer specification which defines how to tunnel 802.11 frames in 802.3 frames or IP packets in UDP packets. Figure 1 presents the general architecture of LWAPP.

Because an AP could possibly connect to more than one AR, when the AP boots up it needs to find the available ARs which can provide LWAPP service. So, before it can really provide wireless access to mobile

stations, an AP begins with a discovery phase, whereby it sends a Discovery Request frame (when LWAPP works on layer 2), causing any AR receiving that frame to respond with a Discovery Reply. From the Discovery Replies received, the AP selects an AR with which to associate, using the Join Request and Join Reply. Once the AP and the AR have joined, a configuration exchange is accomplished to provide the necessary configurations to the AP. The configuration of an AP includes the typical name (802.11 Service Set Identifier, SSID), and security parameters, the data rates to be advertised as well as the radio channel (channels, if the AP is capable of operating more than one 802.11 MAC and PHY simultaneously) to be used. Finally, the AP is ready for operation.

In addition to the functions thus far described, LWAPP also provides for the delivery of commands from the AR to the AP for the management of 802.11 devices that are communicating with the AP. This may include the creation of local data structures in the AP for the 802.11 devices and the collection of statistical information about the communication between the AP and the 802.11 devices. Moreover an AR can send Add Mobile and Delete Mobile commands to enable the APs to accomplish access control. LWAPP also provides heartbeat detection using Echo Request and Echo Reply to detect the AR's active status and can be used to update the firmware on the APs too.

## 3. Security Analysis

One design goal of LWAPP is:

> Reduction of the amount of protocol code being executed at the light weight AP, to apply the computing resource of the AP to the application of wireless access, rather than bridge forwarding and filtering.

So the designers try to take the security services from APs and put them into a centralized router or switch in an interoperable fashion because they argue that this approach increases manageability and allows network operators to more tightly control their wireless network infrastructure. Furthermore, since the interface between the AP and the AR is point-to-point, it is now possible to centralize user or station authentication (such as 802.1x, see Figure 2) as well as policy enforcement functions, without the risk of 802.11 leakage into the network. In this architecture, an AP only forwards frames or packets to or from ARs. On APs there is no implementation and process cost of related authentication protocols such as EAP/RADIUS [5][16] which is processed in ARs. This is a good idea if the centralized switch is powerful enough to coper with the large volume cryptography processes (now more likely AES-CCM [11][19]) with linear speed. And the introduce of LWAPP between APs and ARs doesn't affect the user data's security. But LWAPP

provides one ability to pass wired equivalence privacy (WEP) keys to APs, so APs can help ARs to encrypt and decrypt users' data. On the other hand, designers argue that there is no need to provide security protection for users' data between APs and ARs because the two entities are connected by wired networks and if users are really concern about their data's security, they should adopt end-to-end security mechanics such as IPsec.

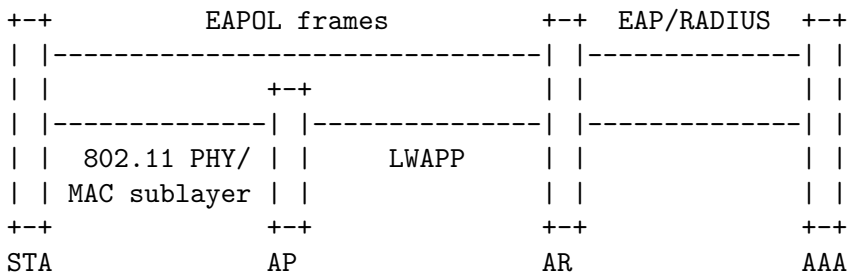


*Figure 2.* 802.1X Authentication in the AR[7]

As far as the security of LWAPP control messages is concerned, privacy and authentication are needed because of some sensitive data are exchanged using control messages between APs and ARs, e.g. firmware images. It is strange that the authors state that WEP keys for user sessions, passed from ARs to APs is one motivation of applying protection on LWAPP control messages but insist that there is no need to protect users' data between APs and ARs. If an adversary can eavesdrop users' plain data along the path between APs and ARs, we can't see the great interest to crack the used WEP keys by attacking the LWAPP control messages. Fortunately this problem can be avoided by simply not pushing the cryptography process to APs but by introducing more ARs to provide load balancing which is inherent in LWAPP or by protecting the user data in the same way as protecting LWAPP control messages.

In LWAPP, the privacy and authentication of control messages are provided by applying AES-CCM on the messages with random session keys. LWAPP adopts a key transport protocol (KTP) to distribute session keys randomly generated by ARs to APs and uses public key cryptography to provide entity authentication. But this protocol only enables APs to authenticate ARs and only provides implicit key confirmation (we call it semi-authenticated KTP). The key distribution can be accomplished in two phases: Join phase and Key Update phase.

During the Join phase, the AP sends a Join Request message including its identifier ($ID_A$), its certification and a randomly generated session

identifier ($Session_{ID}$) to the AR found in the Discovery phase (see Figure 3). Upon receiving Join Request message, the AR first verify the validity of the received certification and then generates a random session key, which is used to secure all future control messages, encrypts the session key with the AP's public key obtained from the AP's certification and computes the AR's digital signature over the hash result of the concatenation of the received session identifier and the encrypted key material. The AR sends its certification, the received session identifier, the encrypted key material and the digital signature in a Join Reply message to the AP. After receiving Join Reply message, the AP first checks the freshness of the session identifier, then verifies the AR's certification and the digital signature. If all the checks are passed, the AP uses its private key to decrypt the session key. The protocol is presented in Figure 4. The used notations are defined as follows:

| | |
|---|---|
| $M_1 \| M_2$: | denotes concatenating component $M_1$ and $M_2$ |
| $[M]$: | component $M$ in the message is an option |
| $SK$: | the material of an encryption key and an authentication hash key |
| $Session_{ID}$: | a random number uniquely identifying a session |
| $pk_X$: | the public key of party $X$ |
| $Cert_X$: | the certification of party $X$ |
| $ID_X$: | the identity of party $X$ |
| $Enc_{pk_X}(M)$: | encrypt message $M$ with asymmetric algorithm using $pk_X$ as the public key |
| $Enc_{SK}(M)$: | encrypt message $M$ with symmetric algorithm, i.e. AES with the encryption key in $SK$. In fact, it is better to used a derived key from $SK$[1] . |
| $Sig_X(M)$: | party $X$'s signature on message $M$ |
| $H_{SK}(M)$: | the authentication code on message $M$ generated by a keyed-hash function[13] using a derived key from $SK$ |
| $H(M)$: | applying hash function on message $M$ |

In order to maximize session key security, APs and ARs periodically update the session keys using Key Update messages. This ensures that a potentially previously compromised key does not affect the security of communication with new key material. The Key Update phase is essentially the same as the Join phase except that the messages are protected by the current session key as presented in Figure 5. The procedure in Figure 5 is slightly different from the one in [7]. The Key Update Request only includes the new session identifier and the Key Update Response only includes the session key in [7]. We think the certifications can expire, so if a certification expires, a new one should be included in the message. We have this field as an option. Moreover, in the draft the signatures are computed directly on the concatenation without applying hash operation.

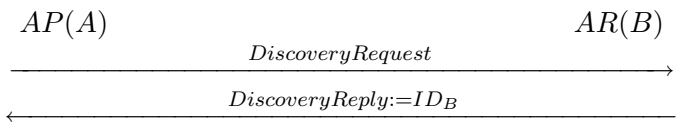


*Figure 3.* Discovery Phase in LWAPP

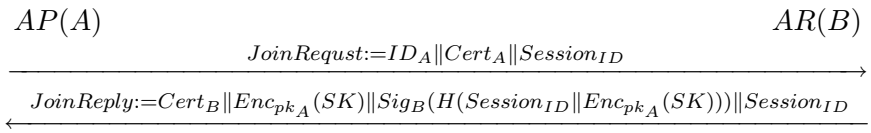


*Figure 4.* Join Phase in LWAPP

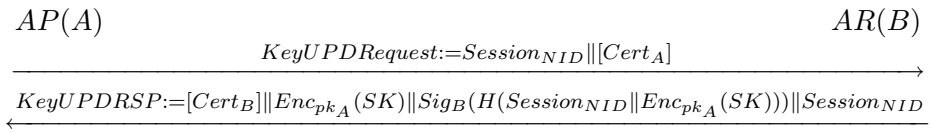


*Figure 5.* Key Update Phase in LWAPP

As we have addressed, the protocol is a KTP without explicit key confirmation and only ARs are authenticated. Based on the assumptions that (1) an AP knows the AR's identifier *a priori* and (2) an AP can verify the AR's certification effectively and (3) the AR can verify an AP's certification, the key transport protocol is secure, provided the used public key cryptosystems and the cryptographic hash function are secure. Here "secure" means nothing more than that only the principal that knows the certification $Cert_A$ corresponding private key can get the session key $SK$ and only the principal that knows the certification $Cert_B$ corresponding private key can generate the second message. Unfortunately, not all of the assumptions hold in the environment where LWAPP is applied.

For assumption (1), an AP possibly doesn't know the identifiers of ARs in advance except that they are configured manually. The identifiers are found in the Discovery phase. When an AP boots up, it broadcasts the Discovery Request message, causing any AR receiving that frame to respond with a Discovery Reply message with the AR's

identifier. A low security level AP with a legal certification which is compromised can send Discovery Reply with its own identifier. Obviously without extra check, this AP can launch the man-in-the-middle attack if the victim AP selects it as the AR candidate. This attack can be prevented by requiring a Certification Authority (CA) using different signature keys to produce certifications for APs and ARs separately. But if in the network ARs have several security levels, it is almost impossible to require a commercial CA to use different keys to produce certifications for different security levels. An alternative way is to use different identifier sets for different levels, for example, $*$.level1.AP.company.com, $*$.level2.AR.company.com. An AP is configured to only accept some specific identifier set. But this approach requires that the implementation of the protocol should support general identifier matching rules.

For assumption (2), it seems that the designers do not consider the certification revocation problem. If an AP needs to check the Certification Revocation List (CRL) [1], it should be able to pass through an AR as ARs are the only routes for APs to other parts of the network. There are two ways to solve this dilemma. One solution, which is commonly used by protocols such as [2], is to require AP to check the CRL immediately after instead of in the middle of establishing a security channel. The second method requires ARs to help APs query the CA center. LWAPP should add four new messages, i.e, Certification Request, Certification Reply, CRL Request and CRL Reply. Using the new messages, an AR can act as a proxy to accept the certification or CRL queries from APs, relay the queries to the Public Key Infrastructure (PKI) and forward the results to APs. The messages Certification Request and Reply enable the system to support cross-CA certifications which need to obtain other CAs' certifications. Both solutions have advantages and trade-off.

Now let's consider whether a semi-authenticated KTP without explicit key confirmation is enough for LWAPP. Firstly from the principle of LWAPP, with an AR there could be only one active session bound with a session key in an AP because if a new session is created, the old one should be abandoned in case the previous key is compromised. Now let's look at how easily an AP can be attacked by a type of Denial of Service (DoS) attack. What an adversary needs to do is to randomly select a session identifier and send a Join Request message along with the victim AP V's identifier and certification to the AR communicating with V. The AR will update V's session key directly because there is no explicit key confirmation message from V required. Hence V's following messages will be discarded because the session key has been changed, even more the state of the protocol in the AR for V has also changed. V has to restart the Join phases. By inserting a third message in the protocol

to provide the authentication of AP and explicit key confirmation, the attack can be prevented.

We present three approaches to fix the protocol (in fact only protocol 3 is secure). In **protocol 1** presented in Figure 6, the AP should return a message including the session identifier and its signature over the keyed-hashing result of its session identifier. Only after verifying the signature in the third message, the AR will update the session key and change the state. As an adversary can launch DoS attack (consuming computation power) on ARs concerned by the designers (ARs need to do one encryption, one signature and one signature verification) and the non-repudiation property provided by the signature in message 3 is not required in LWAPP, we present the second protocol with less computation cost (see Figure 7). In **protocol 2**, the third message includes the session identifier and the cipher text created using the received key over the keyed-hashing result of its session identifier. In fact, one has the option to only use the hash value without encryption in the third message. But applying encryption makes the attack on hash function much harder because the hash result is also unknown to an adversary. However message 2 in the above protocols can't prove that the AR knows the session key $SK$. One possible attack is that an adversary AR $C$ eavesdrops $Enc_{pk_A}(SK)$ from a session between AR $B$ to AP $A$ and uses it in the second message when $A$ later launches a session with $C$. After that, $C$ somehow induces $A$ to reveal $SK$ to it, then $C$ can decrypt the previous messages between $A$ and $B$. This attack requires a strong assumption. Although we have not yet known whether this can be used by attackers in LWAPP, we present the third variant of the protocol (**protocol 3**) which removes this doubt with little cost. The protocol also considers the problem that in message 2, there is no explicit information reflecting that the message is for the intended AP. Protocol 3 requires the AR to compute the signature over the keyed-hashing result of the concatenation of the intended AP's ID, the received session identifier and the encrypted key material. The third message is similar to protocol 2 except without encryption. Now protocol 3 in Figure 8 provides mutual authentication and explicit key confirmation which is similar to one mode of transport level security protocol [9]. The Key Update phase can be modified in the similar way.

From the above analysis, the DoS attack caused by lacking of explicit key confirmation can be fixed completely by using the new cryptographic protocol. Accessing CRLs to verify ARs' certifications can be supported by introducing four more messages in LWAPP. Some alternative ways are also feasible including using password-based schemes or Identify-Based Cryptosystems (IBC) [8]. The password-based anti off-line dictionary

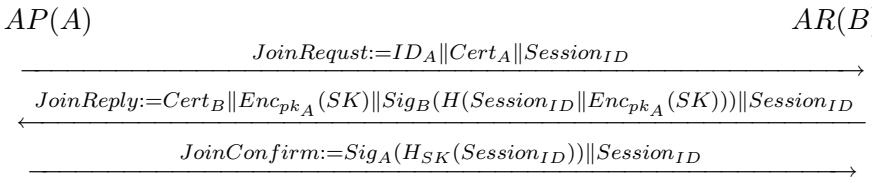


*Figure 6.* Join Phase with Explicit Key Confirmation 1

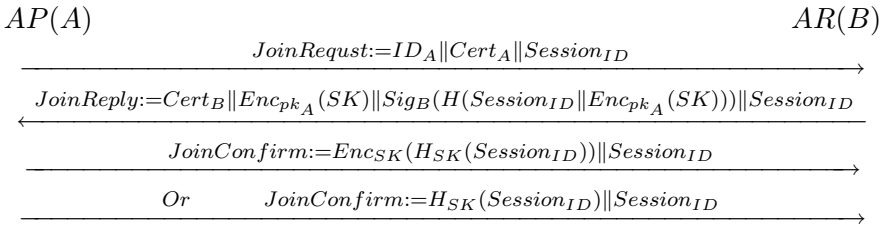


*Figure 7.* Join Phase with Explicit Key Confirmation 2

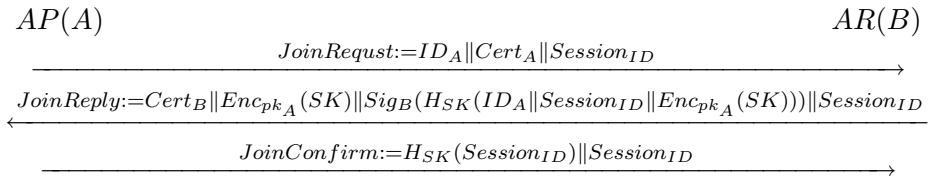


*Figure 8.* Join Phase with Explicit Key Confirmation 3

attack schemes are available [10] (in fact there is no essential need to consider this attack because APs can use human-unfriendly long passwords). Using IBC APs suffer from the same problem (identity revocation check). But it is simpler and more flexible to use the identity-based CAs to enable ARs to use identifiers with a short expiry date appended and update key pairs easily and frequently. These properties significantly reduce the need for revocation check in an IBC. But to prevent the attack introduced by the identifier discovery operation depends on the protocol implementation and the deployment configuration in the real environment.

## 4. Conclusion

LWAPP as a candidate[2] of a possible new wireless network standard for wireless switches is widely expected to be an effective tool to make wireless switching more competitive. But as the not long history of WLAN, the designers did not pay enough attention on the security issues. In this report we briefly analyze some security issues of LWAPP and find that some attacks are feasible. We present some fixing suggestions to the existing protocol to preclude some of the presented attacks. But it seems that some attacks can't be prevented only by designing a good cryptographic protocol, for example, a party's identifier problem which is inherent in a public key based protocol needs the cooperation of other operations, even including the protocol implementation and deployment configurations in various environments. The proposed protocols have their own interest and can be used in other environments.

## Notes

1. One possible derivation is: encryption key $SK_e = H_{SK}(ID_A \| ID_B \| Session_{ID} \| 0x01)$ and authentication key $SK_a = H_{SK}(ID_A \| ID_B \| Session_{ID} \| 0x11)$.

2. The new working group on control and provisioning of wireless access point standard is CAPWAP[12].

## References


[1] C. Adams and S. Farrell, "Internet X.509 Public Key Infrastructure," IETF RFC 2510, March 1999.

[2] B. Aboba and D. Simon, "PPP EAP TLS Authentication Protocol," IETF RFC 2716, October 1999

[3] M. Bumester, "On the risk of opening distributed keys," Crypto94, LNCS 839 1994.

[4] M. Bellare and P. Rogaway, "Entity Authentication and Key Distribution," CRYPTO '93, LNCS 773, pages 232-249. Springer-Verlag, Berlin, 1994.

[5] L. Blunk and J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)," IETF RFC 2284, March 1998.

[6] N. Borisov, I. Goldberg and D. Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11," In Proc. 7th ACM Conference on Mobile Computing and Networking (MOBICOM01), Rome, Italy, 2001.

[7] P. Calhoun, B. O'Hara, S. Kelly, R. Suri, D. Funato and M. Vakulenko, "Light Weight Access Point Protocol (LWAPP)," draft version 3, June 28, 2003.

[8] L. Chen, C. Kudla, "Identity Based Authenticated Key Agreement from Pairings," Cryptology ePrint Archive, Report 2002/184.

[9] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0," IETF RFC 2246, November 1998.

[10] R. Gennaro and Y. Lindell, "A Framework for Password-Based Authenticated Key Exchange," In Proceedings of EUROCRYPT'03, 2003.

[11] B. Gladman, "A Specification for Rijndael, the AES Algorithm," v3.3, May 1 2002.

[12] B. O'Hara and L. Yang, "Architecture for Control and Provisioning of Wireless Access Points(CAPWAP)," draft-ietf-capwap-arch-00.txt, February 2004

[13] H. Krawczyk, M. Bellare and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," IETF RFC 2104, February 1997.

[14] S. Kelly and D. Molnar, "LWAPP Security Requirements draft", draft-kelly-ietf-lwapp-sec-00.txt, August 2003.

[15] A. Menezes, P. van Oorschot and S. Vanstone, "Handbook of Applied Cryptography," CRC Press, 1996.

[16] C. Rigney, A. Rubens, W. Simpson and S. Willens, "Remote Authentication Dial In User Service (RADIUS)," IETF RFC 2058, January 1997.

[17] V. Shoup, "On Formal Models for Security Key Exchange," Theory of Cryptography Library, 1999.

[18] M. Smetannikov, "LWAPP standard props up wireless switching case," http://www.the451.com/, Jun 27 2003.

[19] D. Whiting, R. Housley, N. Ferguson, "Counter with CBC-MAC (CCM)," IETF RFC 3610, September 2003.