2005

Applying Unbalanced RSA to Authentication and Key Distribution in 802.11

Zhong Zheng

Kemal Tepe
University of Windsor

Huapeng Wu

Applying Unbalanced RSA to Authentication and Key Distribution in 802.11

Zhong Zheng
Dept of ECE
University of Windsor
Windsor, Ontario, Canada
Email: zhengf@uwindsor.ca

Kemal E.Tepe
Dept of ECE
University of Windsor
Windsor, Ontario, Canada
Email: ktepe@uwindsor.ca

Huapeng Wu
Dept of ECE
University of Windsor
Windsor, Ontario, Canada
Email: hwu@uwindsor.ca

Keywords: Security, 802.11, WLAN, RSA, authentication, key distribution.

I. INTRODUCTION

It is well known that the data confidentiality algorithm, called Wired Equivalent Privacy (WEP), offerred by the original IEEE 802.11 is not secure mainly due to its improper implementation of RC4 algorithm [3], [4]. The IEEE 802.11 Task Group 'I' (TGi) has designed two options to address this problem. One is called Temporal Key Integrity Protocol (TKIP), intended to be used as a short-term patch for currently deployed equipment. The other one uses Advanced Encryption Standard (AES), a powerful block cipher recommended by NIST to replace DES in 2000, as a long-term solution [1], [2].

TKIP has adopted IEEE 802.1X to provide both authentication and key distribution for WLAN, intending to solve the problems in original 802.11 [2]. Among many options provided by 802.1X, TLS handshake protocol is probably the most secure choice, because it can achieve both mutual authentication and key distribution.

In a wireless environment such as WLAN, the computation power of a mobile device is usually very limited, compared to that of a server. In the TLS handshake protocol, which involves time-consuming public key algorithms, the computation speed due to the client side could become a bottleneck of the system performance. It is thus of great importance to speedup the computation of authentication and key distribution for a power and size constrained device at the client side.

On the other hand, Shamir has proposed a variant of the RSA system [8] that allows for use of large moduli while the actual operations are performed using a much smaller modulus, compared to the operations required for a standard RSA encryption/decryption using Chinese remainder theorem (CRT). This method is reffered to as RSA for paranoids in [8].

In this extended abstract, Shamir's unbalanced RSA method is applied to authentication and key distribution in 802.11 which results in a very simple new protocol. We show that by using the proposed protocol not only the amount of computation can be greatly reduced but also the protocol can be significantly simplified. We also show that our scheme is immune to the attacks to Shamir's 'RSA for paranoids' method proposed by Gilbert, et al [6].

II. PROPOSED PROTOCOL

Although TLS is a well designed protocol for authentication and key distribution, it has disadvantages when we apply it to WLAN. One problem is its complexity. TLS is originally designed for transport layer, it is compatible for most of the systems. For instance, in the phase 1 of TLS handshake protocol, the client and server negotiate a certain version of SSL in order to finish the following conversation. This is because different systems may support different SSL versions. But in WLAN, because we only combine TLS into 802.1X, there is only one version of SSL used for authentication and key distribution. Therefore, negotiating SSL version is not necessary. In our case, we still use WEP as the cipher algorithm for data privacy, and for some efficiency concern (which will be explained later), we will use RSA key-exchange algorithm to distribute keys. Thus, negotiating these algorithms is not necessary. In the second phase of TLS handshake, certificate must be sent to each other for authentication. Server_key_exchange is only necessary for a couple of certain key exchange algorithms, such as Diffie-Hellman key exchange. But in our case, since RSA key exchange algorithm doesn't require any previous parameters shared by both parties. Thus mutual authentication is a requirement in WLAN in order to avoid man-in-the-middle attack, the server doesn't need to send the certificate_request to ask client for a certificate. In the third phase, because of the same reason in phase 2, client_key_exchange can be eliminated, too. During the last phase, the purpose of sending change_cipher_spec is to indicate each other that right after the authentication and key distribution, the new negotiated algorithm and parameters will be used for data privacy. In our case, we can combine the whole 4-phase TLS handshake protocol into several simple steps, in order to make it suitable for 802.11.

The proposed protocol is a simplified version of the existing TLS protocol, which uses the unbalanced RSA algorithm for authentication and key distribution. The proposed protocol can be viewed as one more option for the existing suite of protocols for 802.11.

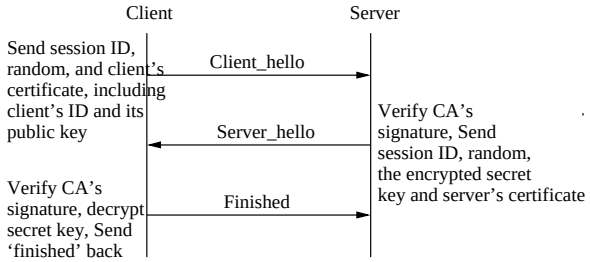

Fig. 1. Proposed protocol

The protocol consists of three steps, which is shown in Figure 1. In Step 1, the client sends a client_hello message with the following parameters:

- *Random:* A client-generated random structure, consisting of a timestamp and a sequence of random number. These values are used during key exchange to prevent replay attacks.
- *Session ID:* A variable-length session identifier.
- Certificate(s): One or a chain of X.509 certificates. It is sent for being authenticated by the server.

In Step 2, the server sends the server_hello message with a session ID and his own random. The random is generated by the server and is independent of the client's random. Following the session ID and the random, the server also sends its own certificate(s), in order to be authenticated by the client. At last, the ciphertext which contains the encrypted secret key will be sent. In Step 3, the client first verifies the server's certificate which he just received. If it is approved, the client then decrypts the ciphertext to get the secret key. So far, both client and server have authenticated each other; and the secret key has been distributed to both parties, too. The client finally sends a finished message to indicate that the whole authentication and key-distribution process is done, and from now on, they can use the shared secret key to create lower level secret keys, in order to transmit data by using WEP algorithm.

Another disadvantage for using TLS is that the client side could become a bottleneck of the system performance due to the fact that a mobile device has limited power and resources to carry out the highly complex public-key cryptographic operations required by authentication and key distribution. However, in WLAN, the clients' devices are not usually computationally powerful, like PDAs, laptop cards and cellphones. This is a very serious problem in reality. Disregarding other factors which may delay the time for handshake, when a user tries to roam from one access point to another in large infrastructure deployments, the time used for a full re-authentication and key-distribution, is too slow to support real-time applications such as audios and videos. A variant of the conventional RSA algorithm proposed by Shamir [8], will be used to achieve this goal.

III. Simulation Results

The protocol was simulated by Java programs on application layer. Since the default packages in Java does not include the class to generate X.509 standard certificates, several other providers such as BouncyCastle and Cryptix32 were added into the Java extension file. The Java program was tested on a laptop computer with an Intel Celeron 1.33GHz CPU and 240 MB RAM. The operation system is Windows XP. We estimate the time consumption for RSA decryption by taking an average of 100 times handshakes. It can be seen from Table I that the speed up factor is about 12.1.

	Conventional RSA	Unbalanced RSA
n (in bits)	1024	1024
p (in bits)	256	1024
q (in bits)	768	1024
Time consumption for decryption (ms) $M \equiv c^d \bmod n$	20.73 (using CRT)	1.72

TABLE I

TIME CONSUMPTION FOR RSA DECRYPTION

IV. Security Against the Attacks

Note that the attacks proposed by Gilbert et al [6] work for decryption using the unbalanced RSA, so we only need to consider the security of key distribution in the proposed protocol against the attacks. The secret key transmitted by key distribution has a known-in-advance fixed size, and it is assumed that p is of size larger than the size of the secret key. If the attacker increases the size of secret key, the client would ask for retransmission no matter whether the unbanlanced RSA or the conventional RSA is used by the client.

V. Conclusions

In this paper, we apply Shamir's 'RSA for paranoids' to authentication and key distribution in 802.11. We show that by using the proposed scheme not only the amount of computation can be greatly reduced but also the protocol can be signifficantly simplified. We also show that our scheme is secure against the attacks to Shamir's 'RSA for paranoids' proposed by Gilbert, et al [6].

References

[1] *IEEE Std.802.11, Standards for local and metropolitan area networks: wireless LAN medium access control (MAC) and physical layer (PHY) specification*, IEEE Standard 802.11, 1999 Edition, 1999.

[2] *IEEE standard for local and metropolitan area networks, port-based network access control*, IEEE Standard 802.1X, June 2001.

[3] N. Borisov, I. Goldberg, D. Wagner, "Intercepting mobile communications: The insecurity of 802.1", *Proc.International Conference on Mobile Computing and Networking*, ACM, July 2001, pp 180-189.

[4] S. Fluhrer, I. Mantin, A. Shamir, "Weaknesses in the key scheduling algorithm of RC", *Proc.4th Annual Workshop on Selected Areas of Cryptography*, August 2001.

[5] M. Gast, *802.11 Wireless Networks: The Definitive Guide*, O'REILLY, April 2002.

[6] H. Gilbert, D. Gupta, A. M. Odlyzko, and J.-J. Quisquater, "Attacks on Shamir's RSA for paranoids", *Information Processing Letters* 68 (1998), pp. 197-199.

[7] A. Menezes, P. Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[8] A. Shamir, "RSA for paranoids", *RSA laboratories' CryptoBytes*, Volume 1, Number 3, Fall 1995.