

Authentication Protocols with Time stamps: – Encryption Algorithm Dependent

Zhan Liu and Mi Lu

Department of Electrical and Computer Engineering

Texas A & M University

College Station, Texas 77840, U.S.A.

{liuzhan, mlu}@ee.tamu.edu

Tel: 979 845 9578

Fax: 979 845 2630

Key word: Authentication Protocol, Time Stamps, Encryption Algorithm, Attack, Security Flaws

Abstract

In this paper, we point out that authentication protocols with time stamps are encryption algorithm dependent. They are vulnerable to replay attack if stream ciphers are used. A detailed attack is given in this paper. Nodes of sensor networks and home networks are resource limited. One encryption algorithm instead of two should be used for sensor networks and home networks for saving memory space. An authentication protocol which is encryption algorithm dependent is needed for sensor networks and home networks.

1 Introduction

A number of authentication protocols [6] are designed with time stamps, in order to guarantee the freshness of messages. The authors only mentioned that symmetric encryptions should be used for their protocols, but did not specify which symmetric encryption. There are two kinds of symmetric encryptions, stream cipher and block cipher. In this paper, we point out that stream cipher should not be used for those authentication protocols with time stamps. We give the attack in details.

Nodes of sensor networks [1] - [3], home networks

[4] and duckling [5] have limited resources, such as very limited storage. For example, nodes of Smart Dust [1] has only 8 Kbytes instruction flash, 512 bytes RAM, and 512 bytes EEPROM. Nodes of sensor networks and home networks have very limited memory resource for data buffer. For many sensor network and home network applications, the data is relatively short. The data of Smart Dust is only 16 bytes. We quote here Menezes et al [7]'s statement about stream cipher, "They can also be used when the data must be processed one symbol at a time (e.g., if the equipment has no memory or buffering of data is limited)". Since sensor networks and home networks have limited memory space for buffering of data, stream cipher is good for those networks. In this paper, we point out that many authentication protocols, which use time stamps, are encryption algorithm dependent. Stream cipher can not be used for those protocols, while other cipher algorithms such as block cipher should be used. So we either have to use two cipher algorithms, one for data encryption and one for key update, or we should use block cipher for both data encryption and key update. If two algorithms are used, two different programs are needed. A program will take memory space. More programs mean more memory space. Since stream ciphers need less buffer than block ciphers, we would like to use stream cipher for both data encryption and key exchange to save memory space. Saving memory space is always attractive and important for resource limited networks, such as sensor networks and home networks.

In this paper, we give attack in details for authentication protocols with time stamps using stream cipher.

Notation used	
$A(B)$:	Participant
Z :	Intruder
Z_A :	Intruder pretends to be A
K_{ab} :	Key between A and B
K^{stream} :	Stream key used for stream cipher
$K_{T_a}^{stream}$:	Part of the stream key using to encrypt T_a
C_{T_a} :	Cipher text of T_a
T_a :	Time stamps
Δt :	Maximum clock error
$\Delta \tau$:	Maximum acceptable time difference
\oplus :	XOR operation

2 Attack on authentication protocol with time stamps using stream cipher

A time stamp is a four byte number, which represents the number of seconds elapsed since 00:00:00 GMT, Jan. 1, 1970.

Assume that A and B have accurately synchronized clocks and there is no delay in sending packets. Suppose there is an intruder Z . When Z intercept a packet, he can check his time immediately and get the current time T_z .

One example of authentication protocol using time stamps is the ISO Symmetric Key One-Pass Unilateral Authentication Protocol which is described as follows.

$$(1) A \rightarrow B : Text2, \{T_a | N_a, B, Text1\}_{K_{ab}}$$

For authentication protocols using time stamps, we generalize the protocol as follows.

$$A \rightarrow B : \{\dots, T_a, \dots\}_{K_{ab}}$$

Where “...” means any other information sent by A . Since we are only interested in time stamps, we omit all other information. We use $K_{T_a}^{stream}$ to represent the key stream which is corresponding to the message part of T_a . And C_{T_a} is the corresponding part of cipher text. So we have

$$C_{T_a} = T_a \oplus K_{T_a}^{stream}.$$

Z has C_{T_a} and T_z . Since we assume that there is no delay and the clocks are synchronized, we have $T_z = T_a$. Z performs $C_{T_a} \oplus T_a = K_{T_a}^{stream}$, and obtain the corresponding part of key stream. Therefore, he can forge any time stamps message later. Suppose at a later time T_{z_later} , Z wants to forge a message. He can operate as follows:

1. Read his current time: T_{z_later}
2. Forge the time stamps: $T_{z_later} \oplus K_{T_a}^{stream} = C_{T_{z_later}}$
3. Substitute C_{T_a} with $C_{T_{z_later}}$
4. Send it to B: $Z_A \rightarrow B : \{\dots, T_{z_later}, \dots\}_{K_{ab}}$

When B receives the message, he performs the following:

1. Decrypt the message: $\{\dots, T_{z_later}, \dots\}_{K_{ab}} \oplus K^{stream} = \{\dots, T_{z_later}, \dots\}$
2. Check the time stamps: T_{z_later} which, of course, is fresh. Accept the message

So B will be fooled by Z and thinks that Z is A . B will accept an old key and an old message. Replay is succeeded.

However, there is a packet delay for any network. Generally speaking, a practical network has a clock synchronization error Δt between A and B . Now, let's suppose the maximum clock synchronization error is Δt_{max} . Assume that a receiver will accept a packet if $T - T_a \leq \Delta \tau$. Z gets the information and checks his time T_z . He can be convinced that $|T_z - T_a| \leq \Delta \tau$ if Z is also a legitimate user. Suppose a time stamps has n bits, i.e. $T = b_{n-1}b_{n-2} \dots b_m b_{m-1} \dots b_0$, and $\Delta \tau$ has m bits, i.e. $\Delta \tau = b_{m-1} \dots b_0$.

$$T_a = b_{n-1}^a b_{n-2}^a \dots b_m^a b_{m-1}^a \dots b_0^a.$$

$$T_z = b_{n-1}^z b_{n-2}^z \dots b_m^z b_{m-1}^z \dots b_0^z.$$

Since $|T_z - T_a| \leq \Delta \tau$, we have either $T_z \geq T_a$ or $T_z < T_a$.

If $T_z \geq T_a$, then $T_z - T_a \leq \Delta \tau$. There are two cases for this situation.

(1.1) If $b_{m-1}^z \dots b_0^z \geq b_{m-1}^a \dots b_0^a$, we have $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z$ (See appendix for detailed proof).

(1.2) If $b_{m-1}^z \dots b_0^z < b_{m-1}^a \dots b_0^a$, we have $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z - 1$.

If $T_z < T_a$, then $T_a - T_z \leq \Delta\tau$. Since $T_z < T_a$ means $T_a > T_z$. The symmetric analysis will give the following results.

(2.1) If $b_{m-1}^a \dots b_0^a \geq b_{m-1}^z \dots b_0^z$, we have $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z$.

(2.2) If $b_{m-1}^a \dots b_0^a < b_{m-1}^z \dots b_0^z$, we have $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z + 1$.

So we have $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z$ or $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z - 1$ or $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z + 1$. Now, like the synchronization clock case, an intruder can calculate the keys for the three cases and fake the corresponding time stamps and information. One of the three cases will give the fresh and legitimate information.

The intruder performs the following:

For the case $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z$,

1. Calculate the key stream: $\{b_{n-1}^a b_{n-2}^a \dots b_m^a\}_{K_{n-1, \dots, m}^{stream}} \oplus (b_{n-1}^z b_{n-2}^z \dots b_m^z) = K_{n-1, \dots, m}^{stream}$.
2. Read current time $T_c = b_{n-1}^c b_{n-2}^c \dots b_m^c b_{m-1}^c \dots b_0^c$.
3. Calculate $\{b_{n-1}^c b_{n-2}^c \dots b_m^c\}_{K_{n-1, \dots, m}^{stream}}$.
4. Substitute $\{b_{n-1}^a b_{n-2}^a \dots b_m^a\}_{K_{n-1, \dots, m}^{stream}}$ with $\{b_{n-1}^c b_{n-2}^c \dots b_m^c\}_{K_{n-1, \dots, m}^{stream}}$.
5. $Z_A \rightarrow B$: $\{\dots, b_{n-1}^c b_{n-2}^c \dots b_m^c b_m^a b_{m-1}^a \dots b_0^a, \dots\}_{K^{stream}}$.

If the assumption is correct, the forged time stamps is fresh and the information is legitimate. B will accept it. Otherwise, Z will consider the case $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z - 1$.

1. Calculate the key stream: $\{b_{n-1}^a b_{n-2}^a \dots b_m^a\}_{K_{n-1, \dots, m}^{stream}} \oplus (b_{n-1}^z b_{n-2}^z \dots b_m^z - 1) = K_{n-1, \dots, m}^{stream}$.
2. Read current time $T_c = b_{n-1}^c b_{n-2}^c \dots b_m^c b_{m-1}^c \dots b_0^c$.
3. Calculate $b_{n-1}^c b_{n-2}^c \dots b_m^c - 1 = b_{n-1}^{c-} b_{n-2}^{c-} \dots b_m^{c-}$.
4. Calculate $\{b_{n-1}^{c-} b_{n-2}^{c-} \dots b_m^{c-}\}_{K_{n-1, \dots, m}^{stream}}$.
5. Substitute $\{b_{n-1}^a b_{n-2}^a \dots b_m^a\}_{K_{n-1, \dots, m}^{stream}}$ with $\{b_{n-1}^{c-} b_{n-2}^{c-} \dots b_m^{c-}\}_{K_{n-1, \dots, m}^{stream}}$.

$$6. Z_A \rightarrow B : \{\dots, b_{n-1}^{c-} b_{n-2}^{c-} \dots b_m^{c-} b_m^a b_{m-1}^a \dots b_0^a, \dots\}_{K^{stream}}.$$

If the assumption is correct, the forged time stamps is fresh and the information is legitimate. B will accept it. If not, then Z will exam the last case $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z + 1$.

1. Calculate the key stream: $\{b_{n-1}^a b_{n-2}^a \dots b_m^a\}_{K_{n-1, \dots, m}^{stream}} \oplus (b_{n-1}^z b_{n-2}^z \dots b_m^z + 1) = K_{n-1, \dots, m}^{stream}$.
2. Read current time $T_c = b_{n-1}^c b_{n-2}^c \dots b_m^c b_{m-1}^c \dots b_0^c$.
3. Calculate $b_{n-1}^c b_{n-2}^c \dots b_m^c - 1 = b_{n-1}^{c+} b_{n-2}^{c+} \dots b_m^{c+}$.
4. Calculate $\{b_{n-1}^{c+} b_{n-2}^{c+} \dots b_m^{c+}\}_{K_{n-1, \dots, m}^{stream}}$.
5. Substitute $\{b_{n-1}^a b_{n-2}^a \dots b_m^a\}_{K_{n-1, \dots, m}^{stream}}$ with $\{b_{n-1}^{c+} b_{n-2}^{c+} \dots b_m^{c+}\}_{K_{n-1, \dots, m}^{stream}}$.
6. $Z_A \rightarrow B$: $\{\dots, b_{n-1}^{c+} b_{n-2}^{c+} \dots b_m^{c+} b_m^a b_{m-1}^a \dots b_0^a, \dots\}_{K^{stream}}$.

This time Z should succeeds. From the above, one can see that the maximum number of cases he needs to consider is three.

Of course, the intruder has no way to know in which case B will accept the information. However, by trying three times, he can cheat B to accept an old message. Since replay means that an intruder can trick B to accept an old message, replay succeeds.

3 Conclusion

Authentication protocols with time stamps are encryption algorithm dependent. They are not against replay if stream ciphers are used. An intruder can launch a replay with 100% success rate. Those protocols can not be applied to sensor networks or home works if only stream ciphers are used.

A Appendix: Proof of

$$\begin{aligned}
b_{n-1}^a b_{n-2}^a \dots b_m^a &= b_{n-1}^z b_{n-2}^z \dots b_m^z \\
\text{or} \\
b_{n-1}^z b_{n-2}^z \dots b_m^z &= \\
b_{n-1}^a b_{n-2}^a \dots b_m^a - 1 &\text{ or} \\
b_{n-1}^z b_{n-2}^z \dots b_m^z &= \\
b_{n-1}^a b_{n-2}^a \dots b_m^a + 1
\end{aligned}$$

The condition for the following discussion is $|T_z - T_a| \leq \Delta\tau$.

A.1 $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z$

For this result, we have two cases.

Case 1: $T_z \geq T_a$ and $b_{m-1}^z \dots b_0^z \geq b_{m-1}^a \dots b_0^a$.

Suppose $b_{n-1}^a b_{n-2}^a \dots b_m^a \neq b_{n-1}^z b_{n-2}^z \dots b_m^z$.

Because $T_z \geq T_a$.

We have $b_{n-1}^z b_{n-2}^z \dots b_m^z > b_{n-1}^a b_{n-2}^a \dots b_m^a$.

Assume that $b_{n-1}^z b_{n-2}^z \dots b_m^z = b_{n-1}^a b_{n-2}^a \dots b_m^a + \delta$, where $\delta > 0$

Because δ is an integer number.

Therefore $\delta \geq 1$.

Now $T_z - T_a = \delta \times 2^m + (b_{m-1}^z \dots b_0^z - b_{m-1}^a \dots b_0^a)$.

Because $b_{m-1}^z \dots b_0^z - b_{m-1}^a \dots b_0^a \geq 0$ and $\Delta\tau = 2^m - 1$.

Therefore $\delta \times 2^m + (b_{m-1}^z \dots b_0^z - b_{m-1}^a \dots b_0^a) \geq 2^m > \Delta\tau$.

This conflicts with $|T_z - T_a| \leq \Delta\tau$. Therefore, we must have $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z$.

Case 2: $T_a > T_z$ and $b_{m-1}^a \dots b_0^a \geq b_{m-1}^z \dots b_0^z$.

This is a symmetric case of case 1 with the exchange of a and z . By symmetric analysis we have $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z$.

A.2 $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z - 1$

Because $T_z \geq T_a$ and $b_{m-1}^z \dots b_0^z < b_{m-1}^a \dots b_0^a$.

We have $b_{n-1}^z b_{n-2}^z \dots b_m^z > b_{n-1}^a b_{n-2}^a \dots b_m^a$.

Assume that $b_{n-1}^z b_{n-2}^z \dots b_m^z = b_{n-1}^a b_{n-2}^a \dots b_m^a + \delta$, where $\delta \geq 1$.

Now $T_z - T_a = \delta \times 2^m + (b_{m-1}^z \dots b_0^z - b_{m-1}^a \dots b_0^a)$ and $\Delta\tau = 2^m - 1$.

Because $|T_z - T_a| \leq \Delta\tau$,

Then $\delta \times 2^m + (b_{m-1}^z \dots b_0^z - b_{m-1}^a \dots b_0^a) \leq 2^m - 1$

Therefore $\delta \times 2^m \leq 2^m - 1 + (b_{m-1}^a \dots b_0^a - b_{m-1}^z \dots b_0^z)$.

Because $b_{m-1}^z \dots b_0^z - b_{m-1}^a \dots b_0^a < 0$.

Therefore $b_{m-1}^a \dots b_0^a - b_{m-1}^z \dots b_0^z > 0$.

When $b_{m-1}^a \dots b_0^a = 11\dots 1$ and $b_{m-1}^z \dots b_0^z = 00\dots 0$, we have the maximum value, i.e.

$\max(b_{m-1}^a \dots b_0^a - b_{m-1}^z \dots b_0^z) = 2^m - 1$.

Therefore $\delta \times 2^m \leq 2^m - 1 + 2^m - 1$.

Then $\delta \times 2^m \leq 2 \times 2^m - 2$.

Therefore $\delta < 2$.

And $1 \leq \delta < 2$.

Hence $\delta = 1$.

And $b_{n-1}^z b_{n-2}^z \dots b_m^z = b_{n-1}^a b_{n-2}^a \dots b_m^a + 1$.

A.3 $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z + 1$

$T_a > T_z$ and $b_{m-1}^a \dots b_0^a < b_{m-1}^z \dots b_0^z$. This is symmetric to the above case. By a similar symmetric analysis, we have $b_{n-1}^z b_{n-2}^z \dots b_m^z = b_{n-1}^a b_{n-2}^a \dots b_m^a - 1$, i.e. $b_{n-1}^a b_{n-2}^a \dots b_m^a = b_{n-1}^z b_{n-2}^z \dots b_m^z + 1$.

References

- [1] J. D. Tygar, et al, "SPINS: Security Protocols for Sensor Networks", *Wireless Networks*, v8, n5, Sept. 2002, pp. 521 - 534.
- [2] Drago Niculescu, "Communication Paradigms for Sensor Networks", *IEEE Communication Magazine*, March 2005, pp. 116 - 122.
- [3] Ian F. Akyildiz, et al, "A Survey on Sensor Networks", *IEEE Communication Magazine*, Aug. 2002, pp. 102 - 114.
- [4] Peter M. Corcoran, et al, "Wireless home network infrastructure for wearable appliances", *IEEE international Conference on Consumer Electronics*, 2002, pp. 104 - 105.
- [5] Frank Stajano, et al, "The resurrection duckling: security issues for ubiquitous computing", *Security & Privacy-2002*, pp. 22 - 26.
- [6] Colin Boyd and Anish Mathuria, "Protocols for Authentication and Key Establishment", *Springer-Verlag*, Chapter 3, 2003.
<http://www2.imm.dtu.dk/courses/02913/F05/papers/CJ97.pdf>.
- [7] A. Menezes, P.van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", *CRC Press*, 1996.