New attacks on ISO key establishment protocols

Anish Mathuria and G. Sriram
DA-IICT
Near Indroda Circle
Gandhinagar-382007
INDIA

anish_mathuria@daiict.ac.in, g.sriram.248@gmail.com

July 29, 2008

Abstract

Cheng and Comley demonstrated type flaw attacks against the key establishment mechanism 12 standardized in ISO/IEC 11770-2:1996. They also proposed enhancements to fix the security flaws in the mechanism. We show that the enhanced version proposed by Cheng and Comley is still vulnerable to type flaw attacks. As well we show that the key establishment mechanism 13 in the above standard is vulnerable to a type flaw attack.

1 Introduction

The ISO/IEC 11770-2:1996 standard provides 13 different key establishment mechanisms using symmetric techniques. One of the six server-based protocols in the standard, namely mechanism 12, is known to contain serious security flaws, which were found by Cheng and Comley [1, 2]. In this paper we present attacks against the enhanced version proposed by Cheng and Comley as well as an attack against the key establishment mechanism 13 in the above standard. We discovered the attacks with the aid of Scyther, an automated protocol analysis tool developed by Cremers [3]. The attacks can be classified as *type flaw attacks* [4], that is, attacks where a message field of one type

X	The distinguishing identifier of entity X
S	A trusted third party
$I(X)$	An intruder impersonating as X
$\{M\}_K$	The encryption of message M with key K
(X, Y)	Concatenation of strings X and Y
TVP	A time variant parameter, e.g. random number, timestamp or sequence number
TVP_X, TVP'_X	A time variant parameter generated by X
R_X, R'_X	A random number generated by entity X
T_X, T'_X	A timestamp generated by entity X
N_X, N'_X	A sequence number generated by entity X
K_{XY}	A symmetric key between entities X and Y

Table 1: Notations

is mistaken as a field of another type. The server-based protocols in the standard use optional text fields in the messages that are encrypted. For convenience, we ignore the optional text fields. The protocols use two types of symmetric key, a long-term key which is initially shared by a protocol user with a server, and a short-term key which is established by running the protocol. We will use the notations shown in Table 1.

The rest of the paper is organized as follows. Section 2 presents an attack on mechanism 13. Section 3 reviews the attacks identified by Cheng and Comely on mechanism 12. Section 4 presents attacks on the enhancement proposed by Cheng and Comley.

2 Attack on mechanism 13

The original description of mechanism 13 assumes that the initiator and responder are B and A, respectively. However, it is more conventional to assume that A is the initiator and B is the responder. The description shown in Fig. 1 is obtained by interchanging the roles of A and B in the original description.

Using Scyther, we discovered an attack on this protocol, which is shown in Fig. 2. The result of the attack is that A accepts a forged key to communicate with B; this key is generated by the

intruder I, instead of B. The attack begins with I intercepting the nonce R_A intended for B. I generates a random value X and starts a parallel run with B in which I masquerades as the principal (R_A, A, X) to B. In this run, B generates a key K_{IB} to communicate with the entity whose identity is (R_A, A, X) and sends message 2' to S. The reply from S is intercepted by I. I replays the message part encrypted in K_{BS} by S to masquerade as B in the first run. S interprets this message as a request from B to transfer the key X to A. Using S's reply I sends a forged message to A in the first run. Upon receiving this message, A interprets X as a session key with B. Thus I can successfully impersonate B to A.

The above attack makes the assumption that a concatenation be accepted as an entity name. Such assumptions have been used in type flaw attacks that have appeared in the literature previously; for example, see the attacks by Boreale and Buscemi [5] on the Needham-Schroeder-Lowe protocol [6] and by Gao, Boedei and Degano [7] on the amended Needham-Schroeder symmetric key protocol [8].

3 Attacks by Cheng and Comley on Mechanism 12

Mechanism 12 uses a time variant parameter TVP_A in the message sent from A to S. We call the implementation which uses a nonce issued by A as TVP_A the nonce version and which uses a timestamp issued by A as TVP_A the timestamp version. Fig. 4 shows the replay attack on the nonce version of the protocol, which was discovered by Cheng and Comley [1, 2].

In the attacking run, I tricks S into re-issuing an old key, which B accepts as new even though it

1. $A \rightarrow B: \quad R_A$
2. $B \rightarrow S: \quad \{R'_B, R_A, A, K_{AB}\}_{K_{BS}}$
3. $S \rightarrow B: \quad \{R'_B, A\}_{K_{BS}}, \{R_A, K_{AB}, B\}_{K_{AS}}$
4. $B \rightarrow A: \quad \{R_A, K_{AB}, B\}_{K_{AS}}, \{R_B, R_A\}_{K_{AB}}$
5. $A \rightarrow B: \quad \{R_A, R_B\}_{K_{AB}}$

Figure 1: Mechanism 13

$$
\begin{array}{lrl}
1. & A \rightarrow I(B): & R_A \\
1'. & I(R_A, A, X) \rightarrow B: & R_I \\
2'. & B \rightarrow S: & \{R'_B, R_I, (R_A, A, X), K_{IB}\}_{K_{BS}} \\
3'. & S \rightarrow I(B): & \{R'_B, (R_A, A, X)\}_{K_{BS}}, \{R_I, K_{IB}, B\}_{K_{(R_A, A, K_I)S}} \\
2. & I(B) \rightarrow S: & \{R'_B, R_A, A, X\}_{K_{BS}} \\
3. & S \rightarrow I(B): & \{R'_B, A\}_{K_{BS}}, \{R_A, X, B\}_{K_{AS}} \\
4. & I(B) \rightarrow A: & \{R_A, X, B\}_{K_{AS}}, \{R'_I, R_A\}_X \\
5. & A \rightarrow I(B): & \{R_A, R'_I\}_X
\end{array}
$$

Figure 2: Attack on mechanism 13

$$
\begin{array}{lll}
1. & A \rightarrow S: & \{TVP_A, B, K_{AB}\}_{K_{AS}} \\
2. & S \rightarrow A: & \{TVP_A, B\}_{K_{AS}}, \{T_S, K_{AB}, A\}_{K_{BS}} \\
3. & A \rightarrow B: & \{T_S, K_{AB}, A\}_{K_{BS}}, \{T_A, B\}_{K_{AB}} \\
4. & B \rightarrow A: & \{T_B, A\}_{K_{AB}}
\end{array}
$$

Figure 3: ISO Protocol 12

may possibly be an old compromised key. Such an attack is commonly called a known-key attack.

The replay attack shown above is not possible on the timestamp version of the protocol. However, Cheng and Comley found a type flaw attack against the timestamp version, which works on the nonce version of the protocol as well. The type flaw attack is shown in Fig. 5. The result of the attack is that A accepts a forged key for communication with B.

Using Scyther we were able to re-discover the attacks shown in Fig. 4 and Fig. 5; no other attacks were detected using Scyther.

4 Attacks on Cheng-Comley-ISO mechanism 12

Cheng and Comley proposed a modification to mechanism 12 to avoid the attacks described in the previous section. The enhanced version is called the fixed *universal protocol*; like its original

1. $I(A) \rightarrow S: \quad \{R_A, B, K_{AB}\}_{K_{AS}}$
2. $S \rightarrow I(A): \quad \{R_A, B\}_{K_{AS}}, \{T'_S, K_{AB}, A\}_{K_{BS}}$
3. $I(A) \rightarrow B: \quad \{T'_S, K_{AB}, A\}_{K_{BS}}, \{T_I, B\}_{K_{AB}}$
4. $B \rightarrow A: \quad \{T'_B, A\}_{K_{AB}}$

Figure 4: Replay attack on protocol using nonces

1. $I \rightarrow S: \quad \{T_I, B, A\}_{K_{IS}}$
2. $S \rightarrow I: \quad \{T_I, B\}_{K_{IS}}, \{T_S, A, I\}_{K_{BS}}$

$1'$. $I(B) \rightarrow S: \quad \{T_S, A, I\}_{K_{BS}}$

$2'$. $S \rightarrow I(B): \quad \{T_S, A\}_{K_{BS}}, \{T'_S, I, B\}_{K_{AS}}$

$3'$. $I(B) \rightarrow A: \quad \{T'_S, I, B\}_{K_{AS}}, \{T_I, A\}_I$

$4'$. $A \rightarrow I(B): \quad \{T_A, B\}_I$

Figure 5: Type attack on protocol using nonces

counterpart it can be viewed as a generalization of the nonce and timestamp versions. The fixed universal protocol is shown in Fig. 6. They also proposed an alternative fixed protocol using timestamps, which uses a different message structure than the protocol obtained from the fixed universal protocol by using a timestamp for TVP_A. Using Scyther, we discovered attacks on the three fixed protocols.

The fixed universal protocol using nonces avoids both the attacks that are possible against its original counterpart. In the fixed protocol the intruder cannot obtain the message intended for B

1. $A \rightarrow S: \quad \{TVP_A, B, K_{AB}\}_{K_{AS}}$
2. $S \rightarrow A: \quad \{TVP_A, B, \{T_S, K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
3. $A \rightarrow B: \quad \{T_S, K_{AB}, A\}_{K_{BS}}, \{T_A, B\}_{K_{AB}}$
4. $B \rightarrow A: \quad \{T_B, A\}_{K_{AB}}$

Figure 6: Fixed universal protocol

from the message sent by S, because of the double encryption in message 2. Fig. 7 shows an attack found by Scyther on the above protocol.

$$
\begin{array}{lll}
1. & A \rightarrow I(S): & \{R_A, B, K_{AB}\}_{K_{AS}} \\
2. & I(S) \rightarrow A: & \{R_A, B, K_{AB}\}_{K_{AS}} \\
3. & A \rightarrow I(B): & K_{AB}, \{T_A, B\}_{K_{AB}} \\
4. & I(B) \rightarrow A: & \{T_I, A\}_{K_{AB}}
\end{array}
$$

Figure 7: Attack on fixed protocol using nonces

The attack exploits the symmetry between the first two messages of the protocol. The intruder I captures the encrypted message sent from A to S, and masquerades as S by replaying the message to A. In the third message, A inadvertently transmits the key K_{AB} in clear. Thus I can successfully impersonate B to A.

The above attack assumes that A interprets the key K_{AB} as the encrypted message $\{T_S, K_{AB}, A\}_{K_{BS}}$. The attack found by Gao, Bodei and Degano and Brodo [9] on the Woo and Lam protocol π_1 [10] makes a similar assumption: it requires that a nonce be interpreted as an encrypted message.

An attack similar to the one against the fixed universal protocol using nonces applies to the timestamp version of the protocol. Using Scyther we were able to find another possible attack on the timestamp version of the protocol, which is shown in Fig. 8.

$$
\begin{array}{lll}
1. & A \rightarrow S: & \{T_A, B, K_{AB}\}_{K_{AS}} \\
2. & S \rightarrow A: & \{T_A, B, \{T_S, K_{AB}, A\}_{K_{BS}}\}_{K_{AS}} \\
3. & A \rightarrow B: & \{T_S, K_{AB}, A\}_{K_{BS}}, \{T'_A, B\}_{K_{AB}} \\
3'. & I(\{T_S, K_{AB}, A\}_{K_{BS}}) \rightarrow A: & \{T_A, B, \{T_S, K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}, \{T_I, A\}_B \\
4'. & A \rightarrow I(\{T_S, K_{AB}, A\}_{K_{BS}}): & \{T''_A, \{T_S, K_{AB}, A\}_{K_{BS}}\}_B
\end{array}
$$

Figure 8: Attack on fixed protocol using timestamps

The attack works as follows. The intruder I waits for A and B to exchange the first three messages. Then I starts a second run in which I masquerades as an initiator with identity equal to the

encrypted message $\{T_S, K_{AB}, A\}_{K_{BS}}$ and A plays the role of the responder. The first two messages of the second run are omitted. I spoofs the third message by replaying the message sent by S in the first run. Upon receiving this message, A interprets the name B as a session key with the principal $\{T_S, K_{AB}, A\}_{K_{BS}}$.

The alternative fixed protocol using timestamps, which was proposed by Cheng and Comley, is shown in Fig. 9. It modifies the original protocol by adding B's name in encrypted message sent to B. This minor change prevents the attack shown in Fig. 5.

1. $A \rightarrow S: \quad \{T_A, B, K_{AB}\}_{K_{AS}}$
2. $S \rightarrow A: \quad \{T_A, B\}_{K_{AS}}, \{T_S, B, K_{AB}, A\}_{K_{BS}}$
3. $A \rightarrow B: \quad \{T_S, B, K_{AB}, A\}_{K_{BS}}, \{T_A, B\}_{K_{AB}}$
4. $B \rightarrow A: \quad \{T_B, A\}_{K_{AB}}$

Figure 9: Alternative fixed protocol using timestamps

Fig. 10 shows an attack found by Scyther on the above protocol. It involves three runs. In the first run, I requests S to transfer a key K_{IB} to B. After receiving the reply from S, I starts a second run where I masquerades as B to S by replaying the message part encrypted in K_{BS} by S in the first run. S interprets this message as a request from B to transfer a key (K_{IB}, I) to B. I intercepts the reply from S and starts a third attacking run where I masquerades as the principal (I, B) to B by replaying the message part encrypted in K_{BS} by S in the second run. Upon receiving this message, B accepts K_{IB} as a key with the principal (I, B). Thus I can successfully impersonate as (I, B) to B.

It should be noted that above attack assumes that a principal can use the protocol to establish a key with itself. A similar assumption is used the attack proposed by Just and Vaudenay [11] on the MTI A class of protocols [12].

$$
\begin{array}{lll}
1. & I \rightarrow S: & \{T_I, B, K_{IB}\}_{K_{IS}} \\
2. & S \rightarrow I: & \{T_I, B\}_{K_{IS}}, \{T_S, B, K_{AB}, I\}_{K_{BS}} \\
1'. & I(B) \rightarrow S: & \{T_S, B, (K_{IB}, I)\}_{K_{BS}} \\
2'. & S \rightarrow I(B): & \{T_S, B\}_{K_{BS}}, \{T'_S, B, K_{IB}, I, B\}_{K_{BS}} \\
3''. & I(I,B) \rightarrow B: & \{T'_S, B, K_{IB}, (I,B)\}_{K_{BS}}, \{T_I, B\}_{K_{IB}} \\
4''. & B \rightarrow I(I,B): & \{T_B, (I,B)\}_{K_{IB}}
\end{array}
$$

Figure 10: Attack on alternative fixed protocol using timestamps

5 Conclusions

The mechanism 12 in ISO/IEC 11770-2:1996 was analyzed previously by Cheng and Comley, who found vulnerabilities in it. Subsequently, they proposed an enhanced version of the mechanism. Using Scyther, we were able to discover type flaw attacks on the enhanced mechanism. Scyther also discovered a type flaw attack on mechanism 13 in the above standard. Finally, we remark that type flaw attacks can be quite complex, thus they may not be easily found without the aid of automated tools that are capable of detecting such attacks.

References

[1] Z. Cheng and R. Comley, Attacks on an ISO/IEC 11770-2 key establishment protocol, Cryptology eprint archive: Report 2004/249. (http://eprint.iacr.org/2004/249)

[2] Z. Cheng and R. Comley, Attacks on an ISO/IEC 11770-2 key establishment protocol, International Journal of Network Security, Vol. 3, No. 3, 2006, 238–243.

[3] C. Cremers, Scyther - Semantics and verification of security protocols, PhD thesis, University Press, Eindhoven, 2006. (http://people.inf.ethz.ch/cremersc/scyther/index.html)

[4] U. Carlsen, Cryptographic protocol flaws: know your enemy, IEEE Computer Security Foundations Workshop VII, 1994, 192-200.

[5] M. Boreale and M. Buscemi, Experimenting with STA, a tool for automatic analysis of protocols, ACM Symposium on Applied Computing (SAC), 2002, 281-285.

[6] G. Lowe, An attack on the Needham-Schroeder public-key authentication protocol, Information Processing Letters, Vol. 56, No. 3, 1995, 131–133.

[7] H. Gao, C. Bodei and P. Degano, A formal analysis of complex type flaw attacks on security protocols, 12th International Conference on Algebraic Methodology and Software Technology (AMAST'08), 2008. (http://compass2.di.unipi.it/TR/Files/TR-08-03.pdf.gz)

[8] R. Needham and M. Schroeder, Authentication revisited, ACM Operating Systems Review, Vol. 21, No. 1, 1987, 7.

[9] H. Gao, C. Bodei, P. Degano and L. Bordo, Detecting and preventing type flaws: a control flow analysis with tags, Electronic Notes in Theoretical Computer Science, Vol. 194, No. 1, 2007, 3-22.

[10] T. Woo and S. Lam, A lesson in authentication protocol design, ACM Operating Systems Review, Vol. 28, No. 3, 1994, 24–37.

[11] M. Just and S. Vaudenay, Authenticated multi-party key agreement, ASIACRYPT'96, LNCS 1163, 1996, 36–49.

[12] T. Matsumoto, Y. Takashima and H. Imai, On seeking smart public-key-distribution systems, Transactions of the IECE of Japan, Vol. E68, No. 2, 1996, 99–106.