New S/Key System against Dictionary Attack : A Case Study in Casper and CSP/FDR

Il-Gon Kim and Jin-Young Choi

Department of Computer Science and Engineering,
Korea University
SEOUL, 136-701 KOREA

Abstract. S/Key(One-Time Password) system has vulnerabilities such as dictionary attack. In this paper, we propose a corrected S/Key system mixed with EKE to solve the man-in-the-middle attack. In addition, we specify a new S/Key system with Casper, verify its secrecy and authentication properties using CSP/FDR.

1 Introduction

S/Key system has the vulnerability, which an intruder can guess a passphrase used in generating an one-time password, comparing the resulting 128 bit hash with variation lists to the dictionary files[2]. For the secure key exchange of passphrase, we compose EKE[7] protocol with S/Key system[6]. Second, for the robustness of passphrase, the new S/key system uses a large random passphrase. In this paper, we specify a new S/Key with Casper, translate its high level description into CSP[4] code, and finally verify its secrecy and authentication properties with FDR[3]. Finally we check a random passphrase's robustness using password cracking tool. In section 2, we give an overview of a new S/Key model. In section 3, we talk about analyzing and verifying a corrected S/Key system using Casper[5] and CSP/FDR. We sum up in section 4.

2 A new S/Key System

In this paper, we focus on the new S/Key system against the offline attack. As mentioned before, to prevent passphrase from dictionary attack, passphrase must be secret between client and server. In order to satisfy the secrecy requirements, we modify S/Key system with a secure EKE. Secondly, passphrase must be randomized, not guessable. We combine EKE with challenge & response protocol in S/Key, exception with applying other variants to it because other variants are

Kim I. and Choi J. (2004).
New S/Key System against Dictionary Attack : A Case Study in Casper and CSP/FDR.
In *Proceedings of the 2nd International Workshop on Verification and Validation of Enterprise Information Systems*, pages 75-77
DOI: 10.5220/0002677500750077

more complex, less effective in the tradeoff between security and traffic performance. Therefore, firstly we find the vulnerabilities of EKE protocol based on [1], then modify it. Example 1 shows a new S/Key authentication mechanism against dictionary attack.

Example 1. New S/key System
Message 1: A → B : A, $\{PK(a)\}K_{ab}$
Message 2: B → A : $\{B, Challenge\}PK(a)$
Message 3: A → B : $\{Na\}K_{ab}$
Message 4: B → A : $\{B, N_a, Passphrase_{ab}\}K_{ab}$
Message 5: A → B : Response

PK(a) is a public key of host A, K_{ab} is a secret password between A and B, *Challenge* is a random number including seed and sequence number, N_a is a random number generated by host A. The *Response* is a hash function of one-time password generated by host A. $Passphrase_{ab}$ is a random session key to generate one-time password used in authenticating a correct user. The most important characteristics in a new S/Key system, differently from that of the previous one, are the encryption of B identity with *PK(a)* and $Passphrase_{ab}$ with K_{ab}. Therefore, the intruder can't be disguised as a legitimate agent and get hold of the plaintext passphrase.

3 Analysis and Verification of new S/Key system

Based on message sequences shown in section 2, we write the new S/Key system in Casper script. A corrected S/Key system is as belows.

```
#Protocol description
0.    -> a : b
1.  a -> b : a, {ServerKey(a)}{passwd}
2.  b -> a : {b, challenge}{ServerKey(a)}
3.  a -> b : {na}{passwd}
4.  b -> a : {b,passphrase}{passwd}
5.  a -> b : F(otp)
```

The ♯*Specification* section is used to specify the requirements of the protocol. In order words, ♯*Specification* section defines the specification model that describes secrecy and authentication properties. We assume that the intruder is a user of the computer network, and so can take part in normal runs of the protocol, and other agents may initiate runs of the protocol with him.

```
#Specification
Secret(a, na, [b])
Secret(b, challenge, [a])
Agreement(a, b, [na, challenge, otp])
Agreement(b, a, [na, challenge, otp])
```

The lines starting *Secret* specify *secrecy property*. In CSP, a *secrecy property* can be formalized as *signal.Claim_Secret.a.b.s* event. This may be understood to mean; 'The secret value s used in the run between *a* and *b*, which was initiated by *a* should be secret for the entire protocol run'. If the secrecy property is satisfied in the model, then the intruder should not be able to obtain access to the secret value, *s*. *Secret(a, na, [b])* may be paraphrased as: 'a thinks na is a secret that can be known to only a and b'. In like manner, *Secret(b, challenge, [a])* represents that 'a thinks challenge is a secret that can be known only b and a'. The lines starting with *Agreement* define that *authentication property*. In CSP model, an *authentication property* can be observed from two viewpoints; one is the authentication of the initiator by the responder,while the other is the authentication of the responder by the initiator. The *Agreement(a, b, [na, challenge, otp])* means that 'a is authenticated to b based on the agreement with *na, challenge, otp* values between two hosts. Also the *Agreement(b, a, [na, challenge, otp])* implies that 'b is authenticated to a based on the agreement with *na, challenge, otp* values between two hosts. After running FDR tool, we confirm that new S/Key system satisfies security properties of secrecy and authentication.

4 Conclusion

S/Key is a very famous authentication system that uses a one-time password scheme to protect a malicious attacker from replay attack. But with the great development of PC's computation power, we unfortunately are confronted by a various threat factors. Moreover, vulnerabilities of S/Key were reported by many researchers. Among them, one of the most effective attacks is to sniff a seed and sequence number from client, then guess passphrase using dictionary attack tool. In this paper, we point out the weakness of an original EKE protocol and modify it. We combine a corrected EKE with S/Key and propose a new S/Key system against dictionary attack. And we specify and verify its secrecy and authentication properties using Casper and CSP/FDR.

References

1. J. Clark and J. Jacob, A survey of authentication protocol literature: Version 1.0, Available via http://www.win.tue.nl/ẽcss/downloads/clarkjacob.pdf, 1997.
2. L.Chen and C.J.Mitchell, Comments on the S/KEY user authentication scheme, ACM SIGOPS Operating Systems Review, Volume 30, Issue 4.,1996.
3. Formal Systems(Europe) Ltd, Failure Divergence Refinement-FDR2 User Manual, Aug. 1999.
4. C.A.R. Hoare, *Communicating Sequential Processes*. Prentice-Hall, 1985.
5. G. Lowe, Casper: A compiler for the analysis of security protocols, 10th IEEE Computer Security Foundations Workshop, 1997.
6. N. Haller, "The S/Key one-time password system," RFC 1760, 1995.
7. S. M. Bellovin, M. Merritt, Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks, AT&T Bell Laboratories. *Proceedings of the 1992 IEEE Computer Society Conference on Research in Security and Privacy*, May 1992.