HAL open science

Cryptanalysis of a SIP Authentication Scheme

Fuwen Liu, Hartmut Koenig

To cite this version:

Fuwen Liu, Hartmut Koenig. Cryptanalysis of a SIP Authentication Scheme. 12th Communications and Multimedia Security (CMS), Oct 2011, Ghent, Belgium. pp.134-143, 10.1007/978-3-642-24712-5_11 . hal-01596196

HAL Id: hal-01596196

https://inria.hal.science/hal-01596196

Submitted on 27 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Cryptanalysis of a SIP Authentication Scheme

Fuwen Liu, Hartmut Koenig

Brandenburg University of Technology Cottbus,
Department of Computer Science
PF 10 33 44, 03013 Cottbus, Germany
{lfw, Koenig}@informatik.tu-cottbus.de

Abstract. SIP (Session Initiation Protocol) is becoming the mostly deployed signaling protocol for VoIP (Voice over IP). Security is of utmost importance for its usage due to the open architecture of the Internet. Recently, Yoon et al. proposed a SIP authentication scheme based on elliptic curve cryptography (ECC) that claimed to provide higher security than other schemes. However, as demonstrated in this paper, it is still vulnerable to off-line dictionary and partition attacks.

Keywords: SIP, VoIP, ECC, authentication, cryptanalysis.

1 Introduction

VoIP that delivers voice and multimedia over the Internet has gained large popularity nowadays. SIP is the dominant standard signaling protocol used for VoIP. It is defined by IETF (*Internet Engineering Task Force*) for the establishment, handling, and release of multimedia sessions among participants over the Internet [1]. Because of its flexibility and extensibility, SIP has been also adopted by 3GPP (*Third Generation Partnership Project)* as the signaling protocol for multimedia applications in 3G mobile networks as well [2]. Thus, SIP has become a key technology to support multimedia communications spanning wired and wireless networks.

Like any other services or protocols running in the hostile Internet environment, the SIP protocol is exposed to a wide range of security threats and attacks. Therefore, appropriate security measures have to be taken to protect SIP. The SIP protocol is based on request/response communication model like HTTP. This implies that a SIP client initiates a request on a SIP server and then waits for a response from the server. Mutual authentication between the SIP client and server is required to ensure the communication partner's identity is legitimate. SIP applies HTTP digest authentication [3] by default to performing authentication. Unfortunately, it is a weak authentication that provides only one-way authentication (i.e. the client is authenticated to the server). This makes server spoofing attacks possible. Moreover, the scheme is vulnerable to the off-line dictionary attacks.

Several SIP authentication schemes have been proposed in order to overcome the security weaknesses of the original SIP authentication scheme. Yang et al. developed a secure SIP authentication scheme whose security relies on the difficulty of the

discrete logarithm problem (DLP) [4]. However, this scheme is not well suited to wireless network settings, where the computational power of devices and the bandwidth of the wireless links are limited. Durlanik [5] et al. presented an efficient SIP authentication scheme using elliptic curve cryptography (ECC) technologies, whose security is based on the elliptic curve discrete logarithm problem (ECDLP). Durlanik's scheme requires less computing power and lower network bandwidth than Yang's scheme, since ECC can achieve the same security level as the conventional cryptosystems by using significantly smaller keys. In 2009, Wu et al. [6] introduced another SIP authentication scheme based on ECC, and proved its security using the Canetti-Krawczyk (CK) security model [7]. The authors stated that the scheme is resilient to various known attacks, such as off-line dictionary attacks, man-in-the-middle attacks, and server spoofing attacks. However, Durlanik's and Wu's schemes have been broken meanwhile. Yoon et al. showed that both schemes are prone to off-line dictionary attacks, Denning-Sacco attacks, and stolen-verifier attacks [8]. They proposed another SIP authentication scheme that is supposed to withstand the aforementioned attacks. In this paper we reveal serious security vulnerabilities of Yoon's scheme, and demonstrate that it is actually insecure against off-line dictionary and partition attacks.

The remainder of the paper is organized as follows. Section 2 forms the preliminary that gives a short overview on the original SIP authentication procedure and the basic knowledge of elliptic curve cryptography (ECC). Next, Yoon's scheme is briefly presented in Section 3. A detailed cryptanalysis of Yoon's scheme is performed in Section 4. Some final remarks conclude the paper.

2 Preliminaries

This section first introduces the SIP authentication procedure. Then the elliptic curve cryptography (ECC) is briefly reviewed.

2.1 SIP Authentication Procedure

SIP uses a challenge-response based mechanism for authentication that is identical to the HTTP digest authentication. Figure 1 illustrates the SIP authentication mechanism. Prerequisite for the SIP authentication is that password and username of each client in the domain have been configured securely in the SIP server in advance, i.e. the SIP server knows the password and the username of each client. Once a SIP server receives a request, it challenges the client with a nonce to determine the client's identity. The client acknowledges the SIP server with a response which is computed by using his/her password and the nonce. The server validates the legitimacy of the client by verifying the received response. The SIP authentication procedure is detailed as follows.

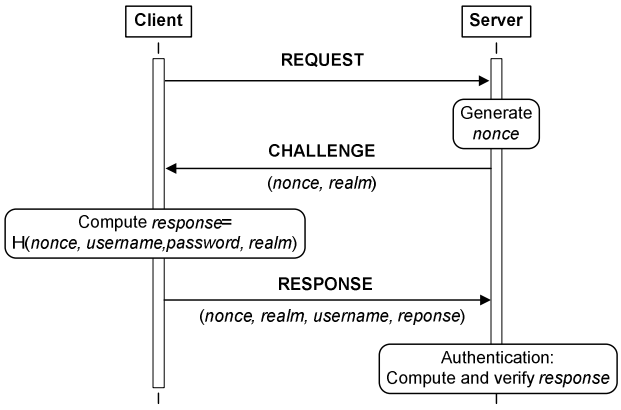

Fig. 1. SIP Authentication Procedure

1. Client → Server: REQUEST
   The client invokes a REQUEST (e.g. SIP REGISTER message) and sends it to the server (e.g. SIP Registrar)

2. Server → Client: CHALLENGE (*nonce*, *realm*)
   The CHALLENGE generated by the server contains a *nonce* and *realm*. The former offers replay protection, since it is fresh for each session. The latter is the host or domain name of the server, and used to remind the client which username and password should be applied.

3. Client → Server: RESPONSE (*nonce, realm, username, response*)
   The client acknowledges the server with the message RESPONSE containing *nonce, realm, username, and response*. The *response* is computed as follows: *response* = *H*(*nonce, realm, username, password*), where *H*(·) is a one-way hash function.

4. After receiving the RESPONSE message, the server retrieves the *password* of the client by referring to the *username.* The server computes *H*(*nonce, realm, username, password*) and compares it with the received *response*. If they match the client is authenticated.

In the SIP authentication scheme the password is never sent across the network in clear text. So an attacker cannot receive the password directly from the captured messages. But the scheme is insecure against off-line dictionary attacks, where an attacker searches for a password matching the recorded message from the password dictionary. This kind of attacks is difficult to detect and to foil because the adversary only needs to eavesdrop on the protocol messages. With a network sniffer tool, such as Wireshark [9], an attacker may easily capture the message RESPONSE (*nonce, realm, username, response*), when the protocol is running over the Internet. He/she can randomly select a password *pw*` from the password dictionary and calculate the

hash function $H(nonce, realm, username, pw`)$. If the hash value does not match the response the attacker can select another password and repeat the procedure till the correct password is found. This attack can be completed in a reasonable time because a human-chosen password has low entropy. For an 8-character password, its entropy is less than 30 bits (i.e. 2^{30} possible passwords) [10].

In the SIP authentication scheme the client is authenticated to the server, whereas the client does not validate the identity of the server. Thus, the scheme is prone to server spoofing attacks. An attacker can forge the identity of the server and sends a CHALLENGE to the client. The honest client always acknowledges the server with a correct message RESPONSE when receiving the message CHALLENGE. After receiving the RESPONSE message, the attacker can launch an off-line dictionary attack as stated above to crack the password of the client.

2.2 Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) [11] uses elliptic curves over a finite field F_p to construct the cryptosystem, where p is a large prime that is usually larger than 224 bits considering security requirements. Let E be an elliptic curve over F_p, so E can be expressed using the Weierstrass equation as follows:

$$y^2 = x^3 + ax + b \quad \textbf{(1)}$$

where $a, b \in F_p$, and $4a^3 + 27b^2 \neq 0 \pmod p$. A pair (x, y), where $x, y \in F_p$, is a point on the curve if (x, y) satisfies equation (1). The point at infinity denoted by $\mathcal{O}$ is on the curve. The set of points in E over finite field F_p, denoted as $E(F_p)$, can be expressed as follows:

$$E(F_p)=\{(x,y) \in F_p^2 : y^2=x^3 + ax + b\} \cup \{\mathcal{O}\} \quad \textbf{(2)}$$

The number of points in $E(F_p)$ is defined as the order of the curve E over F_p. It is denoted by $\#E(F_p)$, which can be estimated by the following formula using Hasse's theorem.

$$P + 1 - 2\sqrt{p} \leq \#E(\mathrm{F_p}) \leq p + 1 + 2\sqrt{p} \quad \textbf{(3)}$$

Key generation in ECC

The elliptic curve domain parameters (p,a,b,P,n,h) are public, where p is a large prime number, a and b specifying an elliptic curve over F_p , P is a base point in $E(F_p)$ and has prime order n, h is the cofactor which is equivalent to $\#E(\mathrm{F_p})/n$. A private key is an integer k which is randomly selected from the interval $[1, n\text{-}1]$, and the corresponding public key is $Q = kP$. Given domain parameters and Q, it is hard to determine k. This is called elliptic curve discrete logarithm problem (*ECDLP*). The ECC systems usually employ the standardized domain parameters which can be found in [13].

Encoding elliptic curve points

The public key $Q = kP$ is a point on the curve E which satisfies equation (1). The point Q can be represented by a pair of field elements (x_Q, y_Q), where x_Q is the x-coordinate and y_Q is the y-coordinate. To reduce the bandwidth required by transmitting point Q can be encoded in a compressed format which is composed of the x-coordinate x_Q and an additional bit β to uniquely determine the y-coordinate y_Q, i.e. (x_Q, β), where β=0 or 1. Therefore one x-coordinate x_Q exactly corresponds to two y-coordinates y_Q, when the compressed point format is used. The number of x_Q denoted as $\#x_Q$ can be determined by the following equation when using Hasse theorem:

$$(p+1)/2-\sqrt{p} \leq \#x_Q \leq (p+1)/2+\sqrt{p} \quad \textbf{(4)}$$

3 Yoon's Authentication Scheme

This section briefly reviews Yoon's scheme used for SIP authentication. It consists of three phases: system setup phase, registration phase, and authentication phase.

A) System Setup Phase

In set up phase client and server determine the common elliptic curve domain parameters ($p,a,b,P\ n,h$) to be used for the system.

B) Registration Phase

Client and server execute the following steps over a secure channel to complete the registration procedure.

1. Client → Server: *username*, $H(pw)$
   The client hashes the password *pw*. The result $H(pw)$ together with the *username* are sent to the server over a secure channel.

2. The server computes $V= H(pw)\oplus H(username,se)$, where *se* is a secret key of the server. Then *username* and *V* are stored in the verification database table. Here, the purpose of *V* is to prevent stolen verifier attacks.

C) Authentication Phase

Fig. 2 illustrates Yoon's SIP authentication scheme. It proceeds as follows.

1. Client → Server: REQUEST (*username*, $cP\oplus H(pw)$)
   The client chooses a random integer *c* from the interval [1, *n*-1], computes the public key *cP*, and encrypts it with the hash value of the password by using the bit-wise exclusive-or (XOR) operation ⊕. The result $cP\oplus H(pw)$ and *username* are sent in message REQUEST to the server.

2. Server → Client: CHALLENGE (*realm*, *sP*, $H(username,sk)$)

After receiving the REQUEST message the server derives the public key cP of the client by computing $cP \oplus H(pw) \oplus H(pw)$. Then it generates a random integer $s \in [1, n\text{-}1]$, and computes a common secret session key $sk = scP$ and a message authentication code $H(username,sk)$. Finally the server responds to the client with the CHALLENGE message ($realm$, sP, $H(username,sk)$).

3. Client→Server: RESPONSE ($username$, $realm$, $H(username\ realm,sk)$)
   After receiving the CHALLENGE message client computes the secret session key $sk = scP$. Thereafter it calculates the hash value $H(username,sk)$ and veryfies it with the received one. If both values are unequal, the client aborts the protocol. Otherwise, the server is authenticated to the client. The client computes the message authentication code $H(username,realm,sk)$, and sends it in the RESPONSE message to the server.

4. Server: Client authentication
   When receiving the RESPONSE message the server computes the message authentication code $H(username,realm,sk)$ and verifies whether it is equal to the received one. If they are not equal the server rejects the request. Otherwise, the client is authenticated to the server. The server accepts his/her request.

Yoon's scheme provides protection against server spoofing attacks due to the mutual authentication between client and server. The authors claim that the proposed scheme is secure against a variety of attacks, including replay attacks, off-line dictionary attacks, man-in-the middle attacks, modification attacks, Denning-Sacco attacks, and stolen-verifier attacks [8]. It should be noted that the prevention of off-line dictionary attacks is the central security requirement for a password-based authentication scheme. If the scheme is vulnerable to off-line dictionary attacks this implies that the password of a client is disclosed to attackers and the whole scheme is broken.

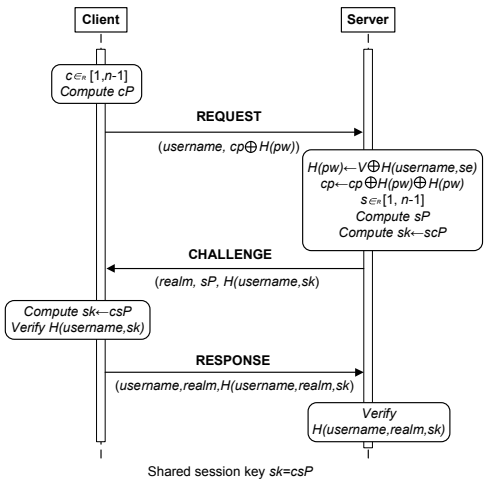

Fig. 2. Yoon's SIP authentication scheme

4 Security Vulnerabilities of Yoon's Scheme

This section shows that an adversary can launch off-line dictionary attacks or partition attacks on Yoon's scheme by using the captured REQUEST message. The partition attack [12] is a special variant of the off-line dictionary attack in which an attacker can partition the password space into a valid and an invalid part by analyzing the eavesdropped protocol messages. Yoon's scheme is vulnerable to off-line dictionary attacks when the public key cP is encoded in an uncompressed format and to partition attacks when the public key cP is encoded in a compressed format. We analyze them separately in the following.

4.1 Off-line Dictionary Attacks

It is trivial for an attacker to capture the REQUEST message which contains the *username* of the client and the masked public key $cP \oplus H(pw)$. We assume that the public key cP is encoded in an uncompressed format (x,y). Accordingly, the masked public key is represented by the format $(x \oplus H(pw), y \oplus H(pw))$. An attacker can perform off-line dictionary attacks as follows.

1. The attacker chooses a candidate password pw_i from the password dictionary D and computes $H(pw_i)$. The password space of the dictionary D is expressed as $|D|$.

2. The attacker computes $x_i = x \oplus H(pw) \oplus H(pw_i)$ and $y_i = y \oplus H(pw) \oplus H(pw_i)$ in order to get a guessed point (x_i,y_i).

3. The attacker examines whether the point (x_i,y_i) is on the elliptic curve E. In other words: he/she checks whether a pair (x_i,y_i) satisfies equation (1) as described in Section 2.2. If the point (x_i,y_i) is not on the elliptic curve E the attacker can eliminate the guessed password pw_i from the dictionary D and return to step 1. He/she repeats the above steps until a point on curve E is found. Finally, the attacker outputs the correct password pw.

Note that the attacker can always verify whether a guessed public key is on the curve because the elliptic curve domain parameters (p,a,b,P,n,h) is public available for their standardizations. The program to perform an off-line dictionary attack can be described as follows.

```
Program off-line dictionary attack
  Input: D,(x⊕H(pw), y⊕H(pw));
  Output: pw;
begin
 1: for i=0 to |D|
 2: {
 3:   pw_i←D;   // select password pw_i from D
 4:   Compute H(pw_i);
 5:   x_i= x⊕H(pw)⊕H(pw_i);
```

```
 6:   y_i= y⊕H(pw)⊕H(pw_i);
 7:   if (y_i^2!=x_i^3+ax_i+b) then goto 3;
 8:       else output pw=pw_i;
 9:  }
end.
```

An attacker can identify the correct password pw in a reasonable time by running the program because the password space $|D|$ is usually less than 2^{30} as stated in [10].

4.2 Partition Attacks

Alternatively, the public key cP can be encoded in a compressed format (x,β), where β is 0 or 1. Accordingly the masked public key $cP \oplus H(pw)$ is represented by $(x \oplus H(pw),\beta)$. Note that the masked public key can not be interpreted as $(x, \beta \oplus H(pw))$ because β is just one bit lang. An adversary can launch a partition attack as follows:

1. Choose a candidate password pw_i from the password dictionary D and compute $H(pw_i)$.

2. Compute $x_i = x \oplus H(pw) \oplus H(pw_i)$ and $\alpha = x_i^3 + ax_i + b$ in order to check whether x_i is a valid x-coordinator in the curve E in the next step.

3. Check whether α is a square in F_p. If so x_i is a valid x-coordinator in the curve E accordingly, and put pw_i into the valid password set VP. Otherwise, put pw_i into the invalid password set UVP.

4. Return to step 1.

The program to run a partition attack can be described as follows.

```
Program partition attack
  Input: D, (x⊕H(pw),β);
  Output: VP, UVP;
begin
 1: for i=0 to |D|
 2: {
 3:   pw_i←D; // select password pw_i from D
 4:   compute H(pw_i);
 5:   x_i= x⊕H(pw)⊕H(pw_i);
 6:   α=x_i^3+ax_i+b;
 7:   if (α is a square) VP←pw_i;
 8:       else UVP←pw_i;
 9:  }
end.
```

As stated in equation (4) of Section 2.2, the number of x_i is in the interval $[(p+1)/2-\sqrt{p}, (p+1)/2+\sqrt{p}]$. So the possibility that x_i is a valid x-coordinate in the curve E over finite field F_p is in the range $[1/2-O(1/\sqrt{p}), 1/2+O(1/\sqrt{p})]$ because the total number of

points in $E(F_p)$ is nearly p. As a result, the adversary can reduce the possible password space by roughly half by running the above program. This means $|VP| \approx ½|D|$, where $|\cdot|$ denotes the password space. After capturing $\log_2|D|$ SIP communication sessions and executing the above program, the adversary can recover the correct password. Such an attack is feasible in practice. For example, it needs only 30 SIP communication sessions to crack the password for an 8-character password whose password space is $|D| \approx 2^{30}$.

4.3 Lessons learned

The aforementioned cryptanalysis demonstrates that Yoon's authentication scheme is insecure no matter how the public key is encoded (uncompressed or compressed format). The main reason is that ECC public key is encrypted directly with the hash value of the password. This inherently provides an attacker the chance to rule the invalid passwords out. He/she can use a guessed password to decrypt the masked public key, and check whether the decrypted result satisfies the elliptic curve equation. Thus, the public key cannot be directly encrypted with the password for a secure password based authentication scheme in the ECC domain. There is a need to revise Yoon's scheme to eliminate the security vulnerabilities. Certainly this means developing a new SIP authentication scheme. Designing a password-based authentication protocol is a challenging task since such protocols are easy vulnerable to off-line dictionary attacks. After many years' research, IEEE has standardized several password-based authentication protocols [14]. Although they can not be directly applied to SIP authentication, their authentication framework can be adapted to the SIP authentication.

5 Final Remarks

User authentication is an essential function in the SIP framework. Several schemes have been proposed to overcome the shortcomings of the original SIP authentication scheme. Yoon's authentication scheme is the newest that claims to be secure against various attacks. In this paper, we have demonstrated that Yoon's scheme is vulnerable to off-line dictionary and partition attacks. An attacker can recover the correct password in a reasonable time. Further research efforts are needed for developing a secure and efficient SIP authentication scheme.

References

1. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler: SIP - Session Initiation Protocol. IETF RFC3261, June 2002.
2. M. Garcia-Martin, E. Henrikson, and D. Mills: Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd Generation Partnership Project (3GPP), IETF RFC3455, 2003.

3. J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart: HTTP Authentication: Basic and Digest Access Authentication. IETF RFC 2617, June 1999.
4. C.C. Yang, R.C. Wang, W.T. Liu: Secure Authentication Scheme for Session Initiation Protocol. Computers and Security 24 (2005): 381–386.
5. A. Durlanik, I. Sogukpinar: SIP Authentication Scheme Using ECDH. World Informatika Society Transaction on Engineering Computing and Technology 8 (2005): 350–353.
6. L. Wu, Y. Zhang, F. Wang, A New Provably Secure Authentication and Key Agreement Protocol for SIP Using ECC, Computer Standards and Interfaces 31 (2009) 2: 286–291.
7. R. Canetti, H. Krawczyk: Analysis of Key-exchange Protocols and Their Use for Building Secure Channels. In: Proc. Eurocrypt 2001, LNCS 2045, Springer, pp. 453–474, 2001.
8. E. J. Yoon, K.Y. Yoo, C. Kim, Y. S. Hong, M. Jo, H. H. Chen: A Secure and Efficient SIP Authentication Scheme for Converged VOIP Networks. Computer Communications 33 (2010): 1674-1681.
9. Wireshark, http://www.wireshark.org/.
10. W. E. Burr, D. F. Dodson, W. T. Polk: Electronic Authentication Guideline. NIST Special Publication 800-63, April 2006.
11. D. Hankerson, A. Menezes, S. Vanstone: Guide to Elliptic Curve Cryptography. Springer, 2003.
12. S. Bellovin, M. Merritt: Encrypted Key Exchange: Password-based Protocols Secure against Dictionary Attacks. Proceedings of the IEEE Symposium on Research in Security and Privacy, May 1992.
13. Certicom Research: SEC 2: Recommended Elliptic Curve Domain Parameters. http://www.secg.org/collateral/sec2_final.pdf
14. IEEE P1363.2: Password-Based Public-Key Cryptography. September 2008.