Enhanced password-based key establishment protocol

Qiang Tang and Chris J. Mitchell
Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
{qiang.tang,c.mitchell}@rhul.ac.uk

15th June 2005

Abstract

In this paper we analyse a password-based authenticated key establishment protocol due to Laih, Ding and Huang, which enables a user to authenticate himself to a server and negotiate a shared session key. This protocol is also designed to guarantee that a human being is actually involved in an ongoing protocol execution. However we show that the protocol suffers from offline dictionary attacks. We propose an enhanced password-based authenticated key establishment protocol which is secure against offline dictionary attacks, and that possesses an additional feature guaranteeing that a user is involved in each protocol execution.

1 Introduction

Recently Laih, Ding and Huang proposed a password-based authenticated key establishment protocol [1] (referred to as the LDH protocol) in which a user and a server can authenticate each other and negotiate a session key. In the LDH protocol, a special function, which is a combination of a picture function and a distortion function, is adopted to authenticate the user and protect the password from offline dictionary attacks. The CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) scheme [2] is an example of such a special function. Laih, Ding and Huang [1] analyse the security of the LDH protocol, and claim that it is secure because it resists known attacks.

However, despite these claims, we show that the LDH protocol suffers from

offline dictionary attacks, which implies that the attacker can collect the messages sent during protocol execution and use them as the basis for an exhaustive search for the password without initiating any new protocol instance. We further propose a new password-based protocol, based on the ideas used in the LDH protocol, which is secure against offline dictionary attacks. In the enhanced protocol, a CAPTCHA scheme is incorporated to guarantee that a human being is involved in every protocol execution.

The rest of this paper is organised as follows. In Section 2, we review the LDH protocol. In Section 3, we analyse the LDH protocol and describe certain security vulnerabilities. In Section 4, we describe the enhanced protocol. In Section 5, we conclude this paper.

2 Review of the LDH protocol

We first introduce some notation. The special function used in [1] is defined as $\varphi(r, s) = g(p(r, s))$, where g is a a distortion function and p is a picture function. Specifically, given inputs r and s, where r is a random string of characters or bits and s is a random number, p generates a random picture which depicts r in some way. Given an input $p(r, s)$ (a picture) the distortion function g generates a distorted version $R' = g(p(r, s))$ such that humans have the ability to recognise r from R' while a machine typically cannot.

Suppose $\{E_{pw}, D_{pw}\}$ denotes a pair of symmetric encryption/decryption functions, where pw is the secret key. h denotes a one-way hash function, n is a security parameter, and B_n denotes the set of all strings of length n, with elements drawn from some set of characters (e.g. all letters or all alphanumeric symbols). All these system parameters except pw are made known to all relevant parties. The secret key pw (a password) is only known to the user and the server.

2.1 Description of the LDH protocol

Suppose a user (U) with identity ID_U wishes to authenticate himself to the server (S) and negotiate a session key. U and S perform the following steps.

1. U generates a random number t, and sends $\{ID_U, t\}$ to S.
2. S first generates a random number s and randomly selects $r \in B_n$. Then S computes and sends $C_1 = E_{pw}(\varphi(r, s))$ and $C_2 = h(pw||r||t)$ to U, where, as throughout, $||$ represents the concatenation operator.
3. U first computes $D_{pw}(C_1)$, which should equal a distorted version of an image depicting r. U then recovers r' from the image, and checks

whether or not $C_2 = h(pw||r'||t)$. If the check succeeds (implying that $r = r'$), U computes and sends $C_3 = h(1||pw||r'||t)$ to S. Otherwise U terminates the protocol.

4. S checks whether $C_3 = h(1||pw||r||t)$ holds. If the check succeeds, S has confirmed that U is the valid user and is involved in the current protocol execution. Otherwise, S terminates the protocol.

If the protocol successfully ends, S and U compute their shared session key as $h(2||pw||r||t)$.

2.2 Claims of Laih, Ding and Huang

In their analysis of the LDH protocol, Laih, Ding and Huang claim that the protocol is secure under the condition $n \log_2 a + \log_2(|C_{pw}|) > 70$, where a is the size of the symbol set used to construct B_n, C_{pw} is the set of passwords, and $|C_{pw}|$ is the size of the password set. They also recommend that $|C_{pw}|$ equals 2^{23} which can be achieved by choosing $n = 4$ and $a = 62$, as results from allowing the symbols to be any lower or upper case letter or any digit from 0 – 9. Specifically they make the following two security claims.

1. Exhaustive search by a machine

   The machine first needs to compute ${C_1}' = E_{pw}(\varphi(r', s')$ by guessing the values of r' and pw', and then compares ${C_1}'$ and C_1 in order to verify this guess. There are a^n possible values for r', and $|C_{pw}|$ possible values for pw', i.e. the total search space is of size

   $$a^n |C_{pw}| = 2^{\log_2 (a^n) + \log_2(|C_{pw}|)} > 2^{70}$$

   So, based on the assumption that $n \log_2 a + \log_2(|C_{pw}|) > 70$, it is computationally infeasible for the machine to compute pw.

2. Exhaustive search by a human being and a machine

   If a valid message $C_1 = E_{pw}(\varphi(r, s)$ is obtained, the machine first guesses a password pw' and computes $A = D_{pw'}(C_1)$; then the human being decides whether or not A contains a string from B_n, which indicates whether or not pw' equals pw. This process is repeated until the correct password is found. This would require the human to check $|C_{pw}| = 2^{23}$ possible values for pw'. Based on this, Laih, Ding and Huang estimate that in this case it will take about 3.2 months for a human being and a machine to successfully search for the password.

3 Security vulnerabilities

In the LDH protocol, the protection of the password is based on the security of the function φ, i.e., the assumption that a machine (without a human being involved) cannot effectively recognise r from $\varphi(r, s)$. As Laih, Ding and Huang point out in [1], the string recognisation CAPTCHA schemes [2] are potentially suitable choices for the function φ. However, the security of these artificial intelligence (AI) problems is based on the state of the art in pattern recognisation research, and is thus essentially heuristic. Mori and Malik [3] have recently developed efficient methods based on shape context matching that can identify, with a high success rate (83%), the word in an ez-gimpy image, a type of CAPTCHA scheme currently in use. Thayananthan et al. [4] developed a program that can achieve a 93% correct recognition rate against ez-gimpy. Recently Moy et al. [5] developed a program that can achieve a 78% accuracy against gimpy-r, another type of CAPTCHA scheme.

Apart from the above problems, we now exhibit a number of security vulnerabilities in the LDH protocol which exist almost regardless of the choice of φ. These vulnerabilities are based on the following observations.

1. A human being must be able to easily recognise r from $D_{pw}(\varphi(r, s))$, which implies that $D_{pw}(\varphi(r, s))$ is very different from a completely random picture.
2. If $pw' \neq pw$ then $D_{pw'}(\varphi(r, s))$ will resemble a random image. This implies that it is possible to determine whether or not a guessed password pw' is correct merely by deciding whether $D_{pw'}(C_1)$ is a (distorted) image or a random pattern.
3. It is likely to be very simple to develop software to distinguish between a distorted image and a random pattern (for example, a compression algorithm should be able to compress an image whereas a random pattern will be incompressible). This is certainly a much simpler problem than automatic string recognition.
4. If humans choose passwords, then they are much more likely to choose some passwords than others; hence if users are free to choose 4-character passwords, then in practice $|C_{pw}|$ will be significantly less than 2^{23}.

Specifically, the following attacks might be mounted by a machine or a human being.

1. In some cases it might be feasible for a machine to mount an offline password guessing attack. The machine works through all possible

passwords and, for each guessed password pw', the machine computes $A = D_{pw'}(C_1)$. By some means (see fact 3 above) the machine then checks whether or not A resembles a distorted image rather than a random bit pattern. Because of fact 2 above, the correct password can be identified from the unique case where A is a distorted image rather than a random bit pattern. This attack only requires a machine-based search of size $|C_{pw}|$. If, for example, it takes a millisecond to check one value of A, then checking through a password space of size 2^{23} will take only 2.3 hours.

2. The above attack does not take into account fact 4 above. Hence the process can be made significantly faster by checking the most likely passwords first.

3. Even if the method of distinguishing random from genuine images is not perfect, i.e. the exhaustive search yields a small number of possible candidate values pw', then a human can be used to check the remaining candidate values A to eliminate all but the value corresponding to the correct password.

4. Distributed attacks are also possible. It may be possible to deploy a cooperative Internet-based attack, e.g. by distributing the pattern recognition problems to users across the Internet (see, for example, [6]).

The security issues in the LDH protocol arise from the fact that the image recognition problem (such as a CAPTCHA scheme) is being used to protect the secrecy of a password. This is not something that appears to have been attempted before, and seems inherently risky. It is probably better to restrict use of such techniques to guaranteeing the presence of a human during protocol execution, rather than to protect the secrecy of passwords or keys. We take this latter approach in the scheme described below.

4 Enhanced password-based key establishment protocol

As we have shown in the previous section, the LDH protocol suffers from a number of vulnerabilities, which are mainly caused by the use made of the distortion function φ. It would appear to be inherently dangerous to use hard pattern recognition problems to protect the secrecy of credentials such as passwords, since progress in solving some of these apparently hard problems has recently been achieved. The difficulty of these problems would appear to be much less well-established than that of problems on which

cryptographic protocols are typically based, such as the difficulty of computational Diffie-Hellman in the multiplicative group of a finite field or the group of a points of an elliptic curve.

In this section, we describe an enhanced password-based key establishment protocol, which can guarantee that a human being is actually involved in each protocol execution. The protocol is a variant of the first password-based key agreement mechanism in ISO/IEC FCD 11770-4 [7], itself based on a scheme originally due to Jablon [8].

In the enhanced protocol, we make the following assumptions. Suppose a user (U) with identity ID_U and a server (S) with identity ID_S share a secret password pw. We also suppose that p and q are two large prime numbers, where $p = 2q + 1$, and h is a secure one-way hash function. When U and S want to negotiate a session key, they first compute $g = h(pw||ID_U||ID_S||i) \bmod p$, where i ($i \geq 0$) is the smallest integer that makes g a generator of a multiplicative subgroup of order q in $GF(p)^*$. U and S then perform the following steps.

1. U generates a random number $t_1 \in Z_q^*$, and sends $m_1 = g^{t_1} \bmod p$ to S.

2. After receiving m_1, S generates a random number $t_2 \in Z_q^*$, and sends $m_2 = g^{t_2} \bmod p$ to U. S uses a CAPTCHA scheme to construct a distorted picture $\varphi(r)$, where r is a random string, and also sends $\varphi(r)$ to U. We suppose that the selected CAPTCHA scheme has not be broken.

   S computes $z = g^{t_2 t_1} \bmod p$ as the shared key material, and computes $K = h(z||1)$ as the shared key.

3. After receiving m_2, U recognises r from the distorted picture $\varphi(r)$, computes $z = g^{t_2 t_1} \bmod p$ as the shared key material, and computes $K = h(z||1)$ as the shared key. Then U constructs and sends the following confirmation message to S:

$$C_1 = h(\varphi(r)||r||3||m_1||m_2||g^{t_1 t_2}||g||ID_U||ID_S)$$

4. After receiving C_1, S checks that the received message equals

$$h(\varphi(r)||r||3||m_1||m_2||g^{t_1 t_2}||g||ID_U||ID_S)$$

   If the check fails, S terminates the protocol execution. Otherwise, S computes and sends the following confirmation message to U:

$$C_2 = h(4||m_1||m_2||g^{t_1 t_2}||g||ID_U||ID_S)$$

5. After receiving C_2, U checks that it equals:

$$C_2 = h(4||m_1||m_2||g^{t_1 t_2}||g||ID_U||ID_S)$$

If the check fails, U terminates the protocol execution. Otherwise U confirms that the protocol execution has successfully ended.

The enhanced key agreement protocol described above is clearly at least as secure as the first password-based key agreement mechanism in [7], which has been thoroughly evaluated. Our new enhanced key agreement protocol has two additional features: the identities of the participants are included in the authentication messages, and a CAPTCHA scheme is adopted to guarantee that a human being is involved in an ongoing protocol execution. Finally note that, even if the CAPTCHA scheme is broken, the security of the enhanced protocol will not be compromised.

5 Conclusions

We have shown that the LDH protocol suffers from serous security vulnerabilities – offline dictionary attacks. It would appear to be inherently dangerous to use hard pattern recognisation problems to protect the secrecy of credentials such as passwords, since progress in solving some of these apparently hard problems has recently been achieved. The difficulty of these problems would appear to be much less well-established than that of problems on which cryptographic protocols are typically based, such as the difficulty of computational Diffie-Hellman in the multiplicative group of a finite field or the group of a points of an elliptic curve. Indeed, recent advances have shown that some problems previously believed to be hard are actually quite simple. However, our enhanced password-based authenticated key establishment protocol shows that such pattern recognisation problems can be usefully incorporated into cryptographic protocol design to provide optional secure features.

References

[1] C. S. Laih, L. Ding, and Y. M. Huang. Password-only authenticated key establishment protocol without public key cryptography. *Electronics Letters*, 41(4):185–186, 2005.

[2] L. Ahn, M. Blum, N. Hopper, and J. Langford. Captcha: Using hard AI problems for security. In E. Biham, editor, *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 294–311. Springer-Verlag, 2003.

[3] G. Mori and J. Malik. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR '03)*, volume 1, pages 134–141. IEEE Computer Society, 2003.

[4] A. Thayananthan, B. Stenger, P. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 127–133. IEEE Computer Society, 2003.

[5] G. Moy, N. Jones, C. Harkless, and R. Potter. Distortion estimation techniques in solving visual CAPTCHAs. In *Proceedings of 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 23–28. IEEE Computer Society, 2004.

[6] G. Price. A general attack model on hash-based client puzzles. In K. G. Paterson, editor, *Cryptography and Coding, 9th IMA International Conference, Cirencester, UK, December 16-18, 2003, Proceedings*, volume 2898 of *Lecture Notes in Computer Science*, pages 319–331. Springer-Verlag, 2003.

[7] International Organization for Standardization. *ISO/IEC FCD 11770–4, Information technology — Security techniques — Key management — Part 4: Mechanisms based on weak secrets*, December 2004.

[8] D. P. Jablon. Strong password-only authenticated key exchange. *Computer Communication Review*, 26(5):5–26, 1996.