On the security of some password-based key agreement schemes

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

23rd May 2005

Abstract

In this paper we show that two potential security vulnerabilities exist in the strong password-only authenticated key exchange scheme due to Jablon. Two standardised schemes based on Jablon's scheme, namely the first password-based key agreement mechanism in ISO/IEC FCD 11770-4 and the scheme BPKAS-SPEKE in IEEE P1363.2 also suffer from one or both of these security vulnerabilities. We further show that other password-based key agreement mechanisms, including those in ISO/IEC FCD 11770-4 and IEEE P1363.2, also suffer from these two security vulnerabilities. Finally, we propose means to remove these security vulnerabilities.

1 Introduction

Password-based authenticated key agreement has recently received growing attention. In general, such schemes only require that a human memorable secret password is shared between the participants. In practice, password-based schemes are suitable for implementation in a wide range of environments, especially those where no device is capable of securely storing high-entropy long-term secret keys. Password-based key agreement schemes originate from the pioneering work of Lomas et al. [1]. Subsequently many password-based key establishment schemes have been proposed, including by Bellovin and Merritt [2, 3], Steiner et al. [4], Jablon [5, 6], Lucks [7], Wu [8], Goldreich and Lindell [9], Bellare et al. [10, 11], Bresson et al. [12], Kwon [13], Nguyen and Vadhan [14], Abdalla et al. [15, 16, 17], and Laih et

al. [18].

However, despite their implementation convenience, password-based key agreement schemes are prone to password guessing attacks, because the users often select their passwords so that they can be easily memorised, which means that they are likely to be much easier to guess than randomly selected passwords. Moreover, for convenience, many users select the same passwords with different partners. For example, in a client-server setting, the client might choose to use the same password with several different servers.

In this paper, we first show that two potential security vulnerabilities exist in Jablon's strong password-only authenticated key agreement scheme [5]. The first password-based key agreement mechanism in [19] and the scheme BPKAS-SPEKE in [20], which are both based on Jablon's scheme, also suffer from one or both of these security vulnerabilities. In fact, other password-based key agreement schemes also suffer from these vulnerabilities. Finally, we show how to remove these vulnerabilities.

2 Description of Jablon's scheme

In this section, we describe the Jablon scheme. At relevant points we also point out the differences between the Jablon scheme and the first password-based key agreement mechanism (in the discrete logarithm setting) in [19], and the scheme BPKAS-SPEKE (in the discrete logarithm setting) in [20].

In the Jablon protocol, the following parameters are made public. p and q are two large prime numbers, where $p = 2q + 1$. h is a strong one-way hash function. Suppose a user (U) with identity ID_U and a server (S) with identity ID_S share a secret password pw. When U and S want to negotiate a session key, they first compute $g = pw^2 \bmod p$.

Note that in the first mechanism of ISO/IEC FCD 11770-4 [19] g is instead computed as $h(pw||str)^2$, where str is an optional string. Also, in BPKAS-SPEKE in draft D20 of P1363.2 [20], g is instead computed as $h(salt||pw||str)^2$, where $salt$ is is a general term for data that supplements a password when input to a one-way function that generates password verification data. The purpose of the $salt$ is to make different instances of the function applied to the same input password produce different outputs. Finally, str is an optional string which it is recommended should include ID_S.

U and S 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 computes $z = g^{t_2 t_1} \bmod p$, and checks whether $z \geq 2$. If the check succeeds, S uses z as the shared key material, and computes $K = h(z)$ as the shared key.

3. After receiving m_2, U computes $z = g^{t_2 t_1} \bmod p$, and checks $z \geq 2$. If the check succeeds, U uses z as the shared key material, and computes $K = h(z)$ as the shared key. Then U constructs and sends the confirmation message $C_1 = h(h(h(z)))$ to S.

   Note that in both the ISO/IEC FCD 11770-4 and IEEE P1363.2 versions of the mechanism, C_1 is instead computed as:

$$C_1 = h(3||m_1||m_2||g^{t_1 t_2}||g),$$

4. After receiving C_1, S checks that the received message equals $h(h(h(z)))$. If the check fails, S terminates the protocol execution. Otherwise, S computes and sends the confirmation message $C_2 = h(h(z))$ to U.

   Note that in both the ISO/IEC FCD 11770-4 and IEEE P1363.2 versions of the mechanism, C_2 is instead computed as:

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

5. After receiving C_2, U checks that it equals $h(h(z))$. If the check fails, U terminates the protocol execution. Otherwise, U confirms that the protocol execution has successfully ended.

Finally, note that in the elliptic curve setting the first password-based key agreement mechanism in [19] and the scheme BPKAS-SPEKE in [20] are essentially the same as above, except that g is a generator of the group of points on an elliptic curve.

3 Security vulnerabilities

In this section we describe two security vulnerabilities in the Jablon protocol. In addition, we show that the standardised password-based key agreement mechanisms in [19, 20] also suffer from one or both of these vulnerabilities.

3.1 The first security vulnerability

This security vulnerability exists when one entity shares the same password with at least two other entities. This is likely to occur when a human user chooses the passwords it shares with a multiplicity of servers. Specifically

we suppose that a client, say U with identity ID_U, shares a password pw with two different servers, say S_1 with identity ID_{S_1} and S_2 with identity ID_{S_2}. A malicious third party can mount the attack as follows.

Suppose U initiates the protocol with an attacker which is impersonating server S_1. Meanwhile the attacker also initiates the protocol with server S_2, impersonating U. The attacker now forwards all messages sent by U (intended for S_1) to S_2. Also, all messages sent from S_2 to U are forwarded to U as if they come from S_1. At the end of the protocol, U will believe that he/she has authenticated S_1 and has established a secret key with S_1. However S_1 will not have exchanged any messages with U. In fact, the secret key will have been established with S_2.

The above attack demonstrates that, even if the server (S_1) is absent, the attacker can make the client believe that the server is present and that they have computed the same session key as each other. Of course, if U shares the same password with servers S_1 and S_2, then S_1 can always impersonate U to S_2 and also S_2 to U, regardless of the protocol design. However, the problem we have described in the Jablon scheme applies even when U, S_1 and S_2 all behave honestly, and this is not a property that is inevitable (we show below possible ways in which the problem might be avoided).

Based on the descriptions in Section 2, it is straightforward to mount this attack on the first password-based key agreement mechanism in [19]. In fact, this attack also applies to the other key agreement mechanisms in [19]. However, if the identifier of the server is used in computing g, e.g. if it is included in the string str, then this attack will fail. The scheme BPKAS-SPEKE in [20] is thus immune to this attack as long as the recommendation given in [20] to include this identifier in str is followed.

3.2 The second security vulnerability

This security vulnerability, which applies to a variety of password-based schemes, exists when two instances of the protocol are concurrently executed. One example of circumstances in which such concurrent execution may arise is when the same key agreement protocol and secret key are used by multiple applications. Without loss of generality, we show how the attack works on the Jablon scheme.

Suppose a client, say U with identity ID_U, shares a password pw with a server, say S with identity ID_S. If the client simultaneously initiates two instances of the protocol with S, then the attacker can mount the attack as follows:

1. In the first protocol instance, suppose that U generates random number $t_1 \in Z_q^*$, and sends $m_1 = g^{t_1} \bmod p$ to S. In the second protocol

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

The attacker then intercepts and swaps these two messages, i.e., it impersonates U to send m_1 and m_1' to S in the second and first protocol instances respectively.

2. In the first protocol instance, after receiving m_1', S generates random number $t_2 \in Z_q^*$, and sends $m_2 = g^{t_2} \bmod p$ to U. S also computes $z' = g^{t_2 t_1'} \bmod p$ as the shared key material with U, and computes $K' = h(z')$ as the shared key with U.

   In the second protocol instance, after receiving m_1, S generates random number $t_2' \in Z_q^*$, and sends $m_2' = g^{t_2'} \bmod p$ to U. S also computes $z = g^{t_2' t_1} \bmod p$ as the shared key material with U, and computes $K = h(z)$ as the shared key with U.

   The attacker then intercepts and swaps these two messages, i.e., it impersonates S to send m_2' and m_2 to U in the first and second protocol instances respectively.

3. In the first protocol instance, after receiving m_2', U computes $z = g^{t_2' t_1} \bmod p$ as the shared key material with S, and computes $K = h(z)$ as the shared key with S. Then U constructs and sends the confirmation message $C_1 = h(h(h(z)))$ to S.

   In the second protocol instance, after receiving m_2, U computes $z' = g^{t_2 t_1'} \bmod p$ as the shared key material with S, and computes $K' = h(z')$ as the shared key with S. Then U constructs and sends the confirmation message $C_1' = h(h(h(z')))$ to S.

   The attacker then intercepts and swaps these two messages, i.e., it impersonates U to send C_1 and C_1' to S in the second and first protocol instances respectively.

4. In the first protocol instance, after receiving C_1', S checks that the received message equals $h(h(h(z')))$. It is simple to verify that this check will succeed. S computes and sends the confirmation message $C_2' = h(h(z'))$ to U.

   In the second protocol instance, after receiving C_1, S checks that the received message equals $h(h(h(z)))$. It is simple to verify that this check will succeed. S computes and sends the confirmation message $C_2 = h(h(z))$ to U.

   The attacker then intercepts and swaps these two messages, i.e., impersonates S to send C_2 and C_2' to U in the first second protocol instances respectively.

5. In the first protocol instance, after receiving C_2, U checks that it equals $h(h(z))$. U has now confirmed that he shares the same key K with S in the first protocol instance, whereas S computes the shared key as K' in the first protocol instance.

   In the second protocol instance, after receiving C_2', U checks that it equals $h(h(z'))$. U has now confirmed that he shares the same key K' with S in the second protocol instance, whereas S computes the shared key as K in the second protocol instance.

The above attack demonstrates that in any pair of concurrent protocol instances, beliefs of the participants regarding shared session keys can be falsified. In other words, in concurrent executions of the protocol an attacker can make the key confirmation progress give false results without it being detected by the participants.

The first password-based key agreement mechanism in [19] and the scheme BPKAS-SPEKE in [20] also suffer from this security vulnerability. Finally, note that this attack applies to many other two-party key agreement protocols, including those in [19, 20].

4 Countermeasures

The following methods can be used to prevent the two security vulnerabilities discussed above.

1. One possible method to prevent the first attack is to include the identities of the participants in the authentication messages C_1 and C_2. In the Jablon scheme, C_1 and C_2 would then be computed as follows:

   $$C_1 = h(h(h(z||ID_U||ID_S))),\ C_2 = h(h(z||ID_S||ID_U))$$

   Correspondingly, in the first password-based key agreement mechanism in [19], C_1 and C_2 would then be computed as follows:

   $$C_1 = h(3||m_1||m_2||g^{t_1t_2}||g^{t_1}||ID_U||ID_S),$$

   and

   $$C_2 = h(4||m_1||m_2||g^{t_1t_2}||g^{t_1}||ID_S||ID_U),$$

2. One possible means preventing the second attack is including a unique session identifier in the computation of g in every protocol instance. For example, in the two standardised mechanisms [19, 20] the session identifier could be included in str.

5 Conclusions

In this paper we have shown that two potential security vulnerabilities exist in the strong password-only authenticated key exchange scheme due to Jablon [5]. The first password-based key agreement mechanism in [19], which is based on Jablon's scheme, suffers from both security vulnerabilities. Moreover, the other key agreement mechanisms in [19] also suffer from both security vulnerabilities. The scheme BPKAS-SPEKE in [20], which is also based on Jablon's scheme, suffers from the second security vulnerability. In fact, it might be possible to mount the second attack against many other two-party key agreement protocols.

References

[1] T. Lomas, L. Gong, J. Saltzer, and R. Needham. Reducing risks from poorly chosen keys. *ACM SIGOPS Operating Systems Review*, 23(5):14–18, 1989.

[2] S. M. Bellovin and M. Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *SP '92: Proceedings of the 1992 IEEE Symposium on Security and Privacy*, pages 72–84, Washington, DC, USA, 1992. IEEE Computer Society.

[3] S. M. Bellovin and M. Merritt. Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In *Proceedings of the First ACM Conference on Computer and Communications Security*, pages 244–250, 1993.

[4] M. Steiner, G. Tsudik, and M. Waidner. Refinement and extension of encrypted key exchange. *Operating Systems Review*, 29(3):22–30, 1995.

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

[6] D. P. Jablon. Extended password key exchange protocols immune to dictionary attack. In *Proceedings of the WETICE '97 Workshop on Enterprise Security*, pages 248–255, Cambridge, MA, USA, 1997.

[7] S. Lucks. Open key exchange: How to defeat dictionary attacks without encrypting public keys. In B. Christianson, B. Crispo, T. Lomas, and M. Roe, editors, *Proceedings of the 5th International Workshop on Security Protocols*, pages 79–90. Springer-Verlag, Berlin, 1997.

[8] T. Wu. The secure remote password protocol. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 1998*, pages 97–111, 1998.

[9] O. Goldreich and Y. Lindell. Session-key generation using human passwords only. In J. Kilian, editor, *Advances in Cryptology – CRYPTO '01*, volume 2139 of *Lecture Notes in Computer Science*, pages 408–432. Springer-Verlag, 2001.

[10] M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract). In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 419–428, 1998.

[11] M. Bellare and P. Rogaway. Authenticated key exchange secure against dictionary attacks. Unpublished manuscript (service contribution) submitted to IEEE P1363, 2000.

[12] E. Bresson, O. Chevassut, and D. Pointcheval. New security results on encrypted key exchange. In B. Feng, R. Deng, and J. Zhou, editors, *Proceedings of the 7th International Workshop on Theory and Practice in Public Key (PKC '04)*, volume 2947 of *Lecture Notes in Computer Science*, pages 145 –158. Springer-Verlag, 2004.

[13] T. Kwon. Practical authenticated key agreement using passwords. In K. Zhang and Y. Zheng, editors, *Proceedings of the 7th Information Security Conference*, volume 3225 of *Lecture Notes in Computer Science*, pages 1–12. Springer-Verlag, 2004.

[14] M. H. Nguyen and S. P. Vadhan. Simpler session-key generation from short random passwords. In M. Noar, editor, *Proceedings of the First Theory of Cryptography Conference (TCC '04)*, volume 2951 of *Lecture Notes in Computer Science*, pages 428–445. Springer-Verlag, 2004.

[15] M. Abdalla, O. Chevassut, and D. Pointcheval. One-time verifier-based encrypted key exchange. In V. Serge, editor, *Proceedings of the 8th International Workshop on Theory and Practice in Public Key (PKC '05)*, volume 3386 of *Lecture Notes in Computer Science*, pages 47–64. Springer-Verlag, 2005.

[16] M. Abdalla, P.-A. Fouque, and D. Pointcheval. Password-based authenticated key exchange in the three-party setting. In V. Serge, editor, *Proceedings of the 8th International Workshop on Theory and Practice in Public Key (PKC '05)*, volume 3386 of *Lecture Notes in Computer Science*, pages 65–84. Springer-Verlag, 2005.

[17] M. Abdalla and D. Pointcheval. Simple password-based encrypted key exchange protocols. In A. Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 191–208. Springer-Verlag, 2005.

[18] C. S. Laih, L. Ding, and Y. M. Huang. Password-only authenticated key establishment protocol without public key cryptography. *Electronics Letters*, 41(4):31–32, 2005.

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

[20] Institute of Electrical and Electronics Engineers, Inc. *IEEE P1363.2 draft D20, Standard Specifications for Password-Based Public-Key Cryptographic Techniques*, March 2005.