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

Abstract. In this paper we show that three 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 some 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 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. [8]. Subsequently many password-based key establishment schemes have been proposed (for example, those in [1–5]). Of course, this is by no means a complete list of existing protocols.

The password used in such protocols is often generated in one of the following two ways. Firstly, the password might be randomly selected from a known password set by a third party. In this case the need for users to be able to memorise the password will limit the size of the password set. As a result, the password will possess low entropy. Secondly, a user might be required to select his password from a known password set. In this case, the user is very likely to choose the password based on his personal preferences (such as name, birth date) again in order to memorise the password easily. As a result, even if the password set is large, the password will still possess low entropy. 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.

Because of this low password entropy, despite their implementation convenience, password-based key agreement schemes are potentially prone to password guessing attacks, including online dictionary attacks and offline dictionary attacks. In the password-based key agreement protocols described in the literature, much effort has been devoted to prevent such attacks. To restrict online dictionary attacks, the commonly used measure is to set a certain interval between two consecutive protocol executions, and at the same time to limit the number of consecutive unsuccessful executions of the protocol. It is clear that an adversary can easily mount a denial of service (DoS) attack against an honest user. However, means of preventing such attacks are beyond the scope of this paper.

In this paper, we first show that three potential security vulnerabilities exist in Jablon's strong password-only authenticated key agreement scheme [2]. The first password-based key agreement mechanism specified in a draft ISO standard [6] and the scheme BPKAS-SPEKE given in an IEEE standard draft [7], which are both based on Jablon's scheme, also suffer from some of these security vulnerabilities. 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 [6], and the scheme BPKAS-SPEKE (in the discrete logarithm setting) in [7].

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, where pw is assumed to be an integer. 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 [6] 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 [7], 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 [6] and the scheme BPKAS-SPEKE in [7] 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 three security vulnerabilities in the Jablon protocol, the third of which is of very general applicability. In addition, we show that the standardised password-based key agreement mechanisms in [6, 7] also suffer from certain of these vulnerabilities.

3.1 The first security vulnerability

We show that the Jablon protocol suffers from a partial offline dictionary attack, which means that an adversary can try several possible passwords by intervening in only one execution of the protocol.

To mount an attack, the adversary first guesses a possible password pw' and replaces the server's message with $m'_2 = (pw')^{2t'_2}$ in the second step of an ongoing protocol instance. The adversary then intercepts the authentication message C_1 in the third step of the same instance and mounts the attack as follows.

1. The adversary sets $i = 1$.
2. The adversary computes $pw'' = (pw')^i$, and checks whether pw'' falls into the password set. If the check succeeds, go to the third step. Otherwise, stop.
3. The adversary checks whether $C_1 = h(h(h((m_1)^{it'_2})))$. If the check succeeds, the adversary confirms that $pw = pw''$. Otherwise, set $i = i + 1$ and go to the second step.

It is straightforward to verify that this attack is valid. We now give a concrete example of how the attack works. Suppose that the password set contains all binary strings of length at most n, where the password pw is made into an integer by treating the string as the binary representation of an integer. Suppose that the adversary guesses a password $pw' = 2$; then he can try $n-1$ passwords $(pw')^i$ $(1 \leq i \leq n-1)$ by intervening in only one execution of the protocol.

However, note that the attack only works when the initial guessed password pw' satisfies $pw' < 2^{n/2}$.

3.2 The second 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 [6]. In fact, this attack also applies to the other key agreement mechanisms in [6]. 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 [7] is thus immune to this attack as long as the recommendation given in [7] to include this identifier in str is followed.

3.3 A generic vulnerability

We next discuss a general problem with key establishment protocols, which not only applies to the Jablon protocol, but also all those discussed in this paper.

In general this problem may apply if two different applications used by a client-server pair employ the same protocol and also the same keying material. Suppose the client and server start two concurrent sessions (A and B say), both of which need to execute a key establishment protocol.

Suppose also that the protocol instances are running simultaneously, and that an attacker can manipulate the messages exchanged between the client and server. The attacker then simply takes all the key establishment messages sent by the client in session A and inserts them in the corresponding places in the session B messages sent to the server; at the same time all the session B key establishment messages are used to replace the corresponding messages sent to the server in session A. Precisely the same switches are performed on the messages sent from the server to the client in both sessions. At the end of the execution of the two instances of the key establishment protocol, the keys that the client holds for sessions A and B will be the same as the keys held by the server for sessions B and A respectively. That is, an attacker can make the key establishment process give false results without it being detected by the participants.

This problem will arise in any protocol which does not include measures to securely bind a session identifier to the key established in the protocol. In particular, the first password-based key agreement mechanisms specified in FCD 11770-4 [6] and the scheme BPKAS-SPEKE in P1363.2 [7] suffer from this vulnerability, as will many other two-part key establishment schemes, including other schemes in these two draft standards.

4 Countermeasures

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

1. To prevent the first attack, which only applies to the Jablon protocol, one possible method is to require g to be computed as $g = h(pw||ID_U||ID_S||i) \bmod p$, where i ($i \geq 0$) is the smallest integer that makes g a generator of a multiplicative subgroup of order q in $GF(p)^*$.
   It is straightforward to verify that the proposed method can successfully prevent the first attack.
2. One possible method to prevent the second 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 [6], C_1 and C_2 would then be computed as follows:

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

   and

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

3. One possible means of addressing the generic attack described in section 3.3 is to include a unique session identifier in the computation of g in every protocol instance. For example, in the two standardised mechanisms [6, 7] the session identifier could be included in str.

5 Conclusions

In this paper we have shown that three potential security vulnerabilities exist in the strong password-only authenticated key exchange scheme due to Jablon [2] where one of these vulnerabilities is of very general applicability and by no means specific to the Jablon scheme. We have further shown that 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 certain of these security vulnerabilities.

References

1. Bellovin, S., Merritt, M.: Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. In: SP '92: Proceedings of the 1992 IEEE Symposium on Security and Privacy, IEEE Computer Society, (1992) 72–84
2. Jablon, D.: Strong Password-Only Authenticated Key Exchange. Computer Communication Review. **26** (1996) 5–26
3. Jablon, D.: Extended Password Key Exchange Protocols Immune to Dictionary Attack. Proceedings of the WETICE '97 Workshop on Enterprise Security. (1997) 248–255
4. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. In Preneel, B., ed.: Advances in Cryptology – EUROCRYPT '00. Volume 1807 of Lecture Notes in Computer Science, Springer-Verlag (2000) 139–155
5. Abdalla, M., Chevassut, O., Pointcheval, D.: One-Time Verifier-Based Encrypted Key Exchange. In Serge, V., ed: Proceedings of the 8th International Workshop on Theory and Practice in Public Key (PKC '05). Volume 3386 of Lecture Notes in Computer Science, Springer-Verlag (2005) 47–64
6. International Organization for Standardization. ISO/IEC FCD 11770–4, Information technology — Security techniques — Key management — Part 4: Mechanisms based on weak secrets. (2004)
7. Institute of Electrical and Electronics Engineers, Inc. IEEE P1363.2 draft D20, Standard Specifications for Password-Based Public-Key Cryptographic Techniques. (2005)
8. Lomas, T., Gong, L., Saltzer, J., Needham, R.: Reducing risks from poorly chosen keys. ACM SIGOPS Operating Systems Review. **23** (1989) 14–18