Internet Research Task Force (IRTF)
Request for Comments: 8125
Category: Informational
ISSN: 2070-1721

J. Schmidt
secunet Security Networks

April 2017

Requirements for Password-Authenticated Key Agreement (PAKE) Schemes

Abstract

Password-Authenticated Key Agreement (PAKE) schemes are interactive protocols that allow the participants to authenticate each other and derive shared cryptographic keys using a (weaker) shared password. This document reviews different types of PAKE schemes. Furthermore, it presents requirements and gives recommendations to designers of new schemes. It is a product of the Crypto Forum Research Group (CFRG).

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Research Task Force (IRTF). The IRTF publishes the results of Internet-related research and development activities. These results might not be suitable for deployment. This RFC represents the consensus of the Crypto Forum Research Group of the Internet Research Task Force (IRTF). Documents approved for publication by the IRSG are not a candidate for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc8125.

Table of Contents

1. Introduction

Passwords are the predominant method of accessing the Internet today due, in large part, to their intuitiveness and ease of use. Since a user needs to enter passwords repeatedly in many connections and applications, these passwords tend to be easy to remember and can be entered repeatedly with a low probability of error. They tend to be low-grade and not-so-random secrets that are susceptible to brute-force guessing attacks.

A Password-Authenticated Key Exchange (PAKE) attempts to address this issue by constructing a cryptographic key exchange that does not result in the password, or password-derived data, being transmitted across an unsecured channel. Two parties in the exchange prove possession of the shared password without revealing it. Such exchanges are therefore resistant to offline, brute-force dictionary attacks. The idea was initially described by Bellovin and Merritt in [BM92] and has received considerable cryptographic attention since then. PAKEs are especially interesting due to the fact that they can achieve mutual authentication without requiring any Public Key Infrastructure (PKI).

Different types of PAKE schemes are reviewed in this document. It defines requirements for new schemes and gives additional recommendations for designers of PAKE schemes. The specific recommendations are discussed throughout Sections 3-7. Section 8 summarizes the requirements.

The requirements mentioned in this document have been discussed with active members and represent the consensus of the Crypto Forum Research Group (CFRG).

2. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. PAKE Taxonomy

Broadly speaking, different PAKEs satisfy their goals in a number of common ways. This leads to various design choices: how public keys are transmitted (encrypted or not), whether both parties possess the same representation of the password (balanced versus augmented), and the number of parties (two party versus multiparty).

3.1. Storage of the Password

When both sides of a PAKE store the same representation of the password, the PAKE is said to be "balanced". In a balanced PAKE, the password can be stored directly in a salted state by hashing it with a random salt or by representing the credential as an element in a finite field (by, for instance, multiplying a generator from a finite field and the password represented as a number to produce a "password element"). The benefits of such PAKEs are that they are applicable to situations where either party can initiate the exchange or both parties can initiate simultaneously, i.e., where they both believe themselves to be the "initiator". This sort of PAKE can be useful for mesh networking (see, for example, [DOT11]) or Internet of Things applications.

When one side maintains a transform of the password and the other maintains the raw password, the PAKE is said to be "augmented". Typically, a client will maintain the raw password (or some representation of it as in the balanced case), and a server will maintain a transformed element generated with a one-way function. The benefit of an augmented PAKE is that it provides some protection for the server's password in a way that is not possible with a balanced PAKE. In particular, an adversary that has successfully obtained the server's PAKE credentials cannot directly use them to

impersonate the users to other servers. The adversary has to learn the individual passwords first, e.g., by performing an (offline) dictionary attack. This sort of PAKE is useful for strict client-server protocols such as the one discussed in [RFC5246].

3.2. Transmission of Public Keys

All known PAKEs use public key cryptography. A fundamental difference in PAKEs is how the public key is communicated in the exchange.

One class of PAKEs uses symmetric key cryptography, with a key derived from the password, to encrypt an ephemeral public key. The ability of the peer to demonstrate that it has successfully decrypted the public key proves knowledge of the shared password. Examples of this exchange include the first PAKE, called the "Encrypted Key Exchange (EKE)", which was introduced in [BM92].

Another class of PAKEs transmits unencrypted public keys, like the J-PAKE (Password Authenticated Key Exchange by Juggling) protocol [JPAKE]. During key agreement, ephemeral public keys and values derived using the shared password are exchanged. If the passwords match, both parties can compute a common secret by combining password, public keys, and private keys. The SPEKE (Strong Password-Only Authenticated Key Exchange) [SPEKE] scheme also exchanges public keys, namely Diffie-Hellman values. Here, the generator for the public keys is derived from the shared secret. Afterwards, only the public Diffie-Hellman values are exchanged; the generator is kept secret. In both cases, the values that are transmitted across the unsecured medium are elements in a finite field and not a random blob.

A combination of EKE and SPEKE is used in PACE as described in [BFK09], which is, e.g., used in international travel documents. In this method, a nonce is encrypted rather than a key. This nonce is used to generate a common base for the key agreement. Without knowing the password, the nonce cannot be determined; hence, the subsequent key agreement will fail.

3.3. Two Party versus Multiparty

The majority of PAKE protocols allow two parties to agree on a shared key based on a shared password. Nevertheless, there exist proposals that allow key agreement for more than two parties. Those protocols allow key establishment for a group of parties and are hence called "Group PAKEs" or "GPAKEs". Examples of such protocols can be found in [ABCP06], while [ACGP11] and [HYCS15] propose a generic construction that allows the transformation of any two-party PAKE

into a GPAKE protocol. Another possibility of defining a multiparty PAKE protocol is to assume the existence of a trusted server with which each party shares a password. This server enables different parties to agree on a common secret key without the need to share a password among themselves. Each party has only a shared secret with the trusted server. For example, Abdalla et al. designed such a protocol as discussed in [AFP05].

4. Security of PAKEs

PAKE schemes are modeled on the scenario of two parties, typically Alice and Bob, who share a password (or perhaps Bob shares a function of the password) and would like to use it to establish a secure session key over an untrusted link. There is a powerful adversary, typically Eve, who would like to subvert the exchange. Eve has access to a dictionary that is likely to contain Alice and Bob's password, and Eve is capable of enumerating through the dictionary in a brute-force manner to try and discover Alice and Bob's password.

All PAKEs have a limitation. If Eve guesses the password, she can subvert the exchange. It is therefore necessary to model the likelihood that Eve will guess the password to access the security of a PAKE. If the probability of her discovering the password is a function of interaction with the protocol participants and not a function of computation, then the PAKE is secure (that is, Eve is unable to take information from a passive attack or from a single active attack). Thus, she cannot enumerate through her dictionary without interacting with Alice or Bob for each password guess, i.e., the only attack left is repeated guessing. Eve learns one thing from a single active attack: whether her single guess is correct or not.

In other words, the security of a PAKE scheme is based on the idea that Eve, who is trying to impersonate Alice, cannot efficiently verify a password guess without interacting with Bob (or Alice). If she were to interact with either, she would thereby be detected. Thus, it is important to balance restricting the number of allowed authentication attempts with the potential of a denial-of-service vulnerability. In order to judge and compare the security of PAKE schemes, security proofs in commonly accepted models SHOULD be used. Each proof and model, however, is based on assumptions. Often, security proofs show that if an adversary is able to break the scheme, the adversary is also able to solve a problem that is assumed to be hard, such as computing a discrete logarithm. By conversion, breaking the scheme is considered to be a hard problem as well.

A PAKE scheme SHOULD be accompanied with a security proof with clearly stated assumptions and models used. In particular, the proof MUST show that the probability is negligible that an active adversary

would be able to pass authentication, learn additional information about the password, or learn anything about the established key. Moreover, the authors MAY specify which underlying primitives are to be used with the scheme or MAY consider specific use cases or assumptions like resistance to quantum computers. A clear and comprehensive proof is the foundation for users to trust in the security of the scheme.

4.1. Implementation Aspects

Aside from the theoretical security of a scheme, practical implementation pitfalls have to be considered as well. If not carefully implemented, even a scheme that is secure in a well-defined mathematical model can leak information via side channels. The design of the scheme might allow or prevent easy protection against information leakage. In a network scenario, an adversary can measure the time that the computation of an answer takes and derive information about secret parameters of the scheme. If a device operates in a potentially hostile environment, such as a smart card, other side channels like power consumption and electromagnetic emanations or even active implementation attacks have to be taken into account as well.

The developers of a scheme SHOULD keep the implementation aspects in mind and show how to implement the protocol in constant time. Furthermore, adding a discussion about how to protect implementations of the scheme in potential hostile environments is encouraged.

4.2. Special Case: Elliptic Curves

Since Elliptic Curve Cryptography (ECC) allows for a smaller key length compared to traditional schemes based on the discrete logarithm problem in finite fields at similar security levels, using ECC for PAKE schemes is also of interest. In contrast to schemes that can use the finite field element directly, an additional challenge has to be considered for some schemes based on ECC, namely the mapping of a random string to an element that can be computed with, i.e., a point on the curve. In some cases, the opposite is also needed, i.e., the mapping of a curve point to a string that is not distinguishable from a random one. When choosing a mapping, it is crucial to consider the implementation aspects as well.

If the PAKE scheme is intended to be used with ECC, the authors SHOULD state whether there is a mapping function needed and, if so, discuss its requirements. Alternatively, the authors MAY define a mapping to be used with the scheme.

5. Protocol Considerations and Applications

In most cases, the PAKE scheme is a building block in a more complex protocol like IPsec or Transport Layer Security (TLS). This can influence the choice of a suitable PAKE scheme. For example, an augmented scheme can be beneficial for protocols that have a strict server-client relationship. If both parties can initiate a connection of a protocol, a balanced PAKE might be more appropriate.

A special variation of the network password problem, called "Password-Authenticated Key Distribution", is defined in [P1363] as password-authenticated key retrieval: "The retrieval of a key from a secure key repository or escrow requiring authentication derived in part from a password."

In addition to key retrieval from escrow, there is also the variant of two parties exchanging public keys using a PAKE in lieu of certificates. In this variant, public keys can be encrypted using a password. Authentication key distribution can be performed because each side knows the private key associated with its unencrypted public key and can also decrypt the peer's public key. This technique can be used to transform a short, one-time code into a long-term public key.

Another possible variant of a PAKE scheme allows combining authentication with certificates and the use of passwords. In this variant, the private key of the certificate is used to blind the password key agreement. For verification, the message is unblinded with the public key. A correct key establishment therefore implies the possession of the private key belonging to the certificate. This method enables one-sided authentication as well as mutual authentication when the password is used.

The authors of a PAKE scheme MAY discuss variations of their scheme and explain application scenarios where these variations are beneficial. In particular, techniques that allow long-term (public) key agreement are encouraged.

6. Privacy

In order to establish a connection, each party of the PAKE protocol needs to know the identity of its communication partner to identify the right password for the agreement. In cases where a user wants to establish a secure channel with a server, the user first has to let the server know which password to use by sending some kind of identifier to the server. If this identifier is not protected, everyone who is able to eavesdrop on the connection can identify the user. In order to prevent this and protect the privacy of the user,

the scheme might provide a way to protect the transmission of the user's identity. A simple way to protect the privacy of a user that communicates with a server is to use a public key provided by the server to encrypt the user's identity.

The PAKE scheme MAY discuss special ideas and solutions about how to protect the privacy of the users of the scheme.

7. Performance

The performance of a scheme can be judged along different lines depending on the optimization goals of the target application. Potential metrics include latency, code size/area, power consumption, or exchanged messages. In addition, there might be application scenarios in which a constrained client communicates with a powerful server. In such a case, the scheme has to require minimal efforts on the client side. Note that for some clients, the computations might even be carried out in a hardware implementation, which requires different optimizations compared to software.

Furthermore, the design of the scheme can influence the cost of protecting the implementation from adversaries exploiting its physical properties (see Section 4.1).

The authors of a PAKE scheme MAY discuss their design choices and the influence of these choices on the performance. In particular, the optimization goals could be stated.

8. Requirements

This section summarizes the requirements for PAKE schemes to be compliant with this document based on the previously discussed properties.

REQ1: A PAKE scheme MUST clearly state its features regarding balanced/augmented versions.

REQ2: A PAKE scheme SHOULD come with a security proof and clearly state its assumptions and models.

REQ3: The authors SHOULD show how to protect their PAKE scheme implementation in hostile environments, particularly, how to implement their scheme in constant time to prevent timing attacks.

REQ4: If the PAKE scheme is intended to be used with ECC, the authors SHOULD discuss their requirements for a potential mapping or define a mapping to be used with the scheme.

REQ5: The authors of a PAKE scheme MAY discuss its design choice with regard to performance, i.e., its optimization goals.

REQ6: The authors of a scheme MAY discuss variations of their scheme that allow the use in special application scenarios. In particular, techniques that facilitate long-term (public) key agreement are encouraged.

REQ7: Authors of a scheme MAY discuss special ideas and solutions on privacy protection of its users.

REQ8: The authors MUST follow the IRTF IPR policy <https://irtf.org/ipr>.

9. IANA Considerations

This document does not require any IANA actions.

10. Security Considerations

This document analyzes requirements for a cryptographic scheme. Security considerations are discussed throughout the document.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

11.2. Informative References

[ABCP06] Abdalla, M., Bresson, E., Chevassut, O., and D. Pointcheval, "Password-Based Group Key Exchange in a Constant Number of Rounds", PKC 2006, LNCS 3958, DOI 10.1007/11745853_28, 2006.

[ACGP11] Abdalla, M., Chevalier, C., Granboulan, L., and D. Pointcheval, "Contributory Password-Authenticated Group Key Exchange with Join Capability", CT-RSA 2011, LNCS 6558, DOI 10.1007/978-3-642-19074-2_11, 2011.

[AFP05] Abdalla, M., Fouque, P., and D. Pointcheval, "Password-Based Authenticated Key Exchange in the Three-Party Setting", PKC 2005, LNCS 3386, DOI 10.1007/978-3-540-30580-4_6, 2005.

[BFK09] Bender, J., Fischlin, M., and D. Kuegler, "Security Analysis of the PACE Key-Agreement Protocol", ISC 2009, LNCS 5735, DOI 10.1007/978-3-642-04474-8_3, 2009.

[BM92] Bellovin, S. and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure against Dictionary Attacks", Proc. of the Symposium on Security and Privacy, Oakland, DOI 10.1109/RISP.1992.213269, 1992.

[DOT11] IEEE, "IEEE Standard for Information technology-- Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE 802.11, DOI 10.1109/IEEESTD.2016.7786995.

[HYCS15] Hao, F., Yi, X., Chen, L., and S. Shahandashti, "The Fairy-Ring Dance: Password Authenticated Key Exchange in a Group", IoTPTS 2015, DOI 10.1145/2732209.2732212, 2015.

[JPAKE] Hao, F. and P. Ryan, "Password Authenticated Key Exchange by Juggling", SP 2008, LNCS 6615, DOI 10.1007/978-3-642-22137-8_23, 2008.

[P1363] IEEE Microprocessor Standards Committee, "Draft Standard Specifications for Password-Based Public Key Cryptographic Techniques", IEEE P1363.2, 2006.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[SPEKE] Jablon, D., "Strong Password-Only Authenticated Key Exchange", ACM SIGCOMM Computer Communications Review, Volume 26, Issue 5, DOI 10.1145/242896.242897, October 1996.

Author's Address

Joern-Marc Schmidt
secunet Security Networks
Mergenthaler Allee 77
65760 Eschborn
Germany

Email: joern-marc.schmidt@secunet.com