                        Requirements for PAKE schemes
                         draft-irtf-cfrg-pake-reqs-00

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 and discusses
   their requirements.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on December 31, 2015.

Copyright Notice

Table of Contents

1. 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].

2. Introduction

Passwords are the predominant method of accessing the Internet today
due largely to their intuitiveness and ease of use. Since a user
needs to enter her password repeatedly over the course of many
connections to the Internet, these passwords tend to be easy to
remember and able to 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. In other words,
they are horrible credentials to use for authentication.

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 to the exchange prove
possession of the shared password without revealing it. Such
exchanges are therefore resistant to an off-line, brute-force
dictionary attack. PAKEs are especially interesting due to the fact
that they can achieve mutual authentication without requiring any
Public Key Infrastructure (PKI).

The problem was initially described by Bellovin and Merritt in [BM92] and has received considerable cryptographic attention since then.

3. PAKE Taxonomy

Broadly speaking, different PAKEs satisfy their goals in a number of common ways. This lead to various design choices - how the password is stored (balanced versus augmented), how public keys are transmitted (encrypted or not), and the number of parties (two party versus multiparty).

3.1. Storage of the Password

When both sides of a PAKE store their representation of the password identically, 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 the password represented as a number to produce a "password element"). The benefits of such PAKE are that it is applicable to situations where either party can initiate the exchange or both parties can initiate simultaneously (where they both believe themselves to be the "initiator"). This sort of PAKE can be useful for mesh networking (e.g. [DOT11]) or Internet-of-Things applications.

When one side maintains are uninvertable 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 and a server will maintain a transformed element generated with a one-way function. The benefit of an augmented PAKE is that the server's password database is protected in a way that is not possible with a balanced PAKE. Augmented PAKEs are resistant to Key Compromise Impersonation (KCI) where an adversary who has successfully attacked Bob can impersonate Bob to everyone, but it is not possible to impersonate everyone back to Bob. An adversary that has successfully obtained the server's PAKE credentials is still required to perform a dictionary attack in order to learn an individual password. This sort of PAKE is useful for strict client-server protocols, such as [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 it has successfully decrypted the
public key proves knowledge of the shared password. Examples of this
exchange include the first PAKE presented by [BM92], the Encrypted
Key Exchange (EKE). A variant of this method, as it is e.g. used in
European travel documents by PACE [TR-03110], is to encrypt a nonce
instead of a key, which is later used for the derivation of the
shared key.

The other class of PAKEs transmit unencrypted public keys. These
public keys may be blinded by some function of the shared password,
but the public key that is transmitted across the unsecured medium is
an element in a finite field, not a random blob. The ability of the
peer to successfully use that public key (for example, possibly
unblinding it) proves knowledge of the shared password. Examples of
this exchange include [SPEKE].

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, hence are called
Group PAKEs or GPAKEs. Examples of such protocols include [ABCP06]
and [HYCS15], while the second one proposes a generic construction
that allows transferring any two-party PAKE into a GPAKE protocol.
Another possibility to define a multi-party PAKE protocol is to
assume the existence of a trusted server each party shares a password
with. This server enables different parties to agree on a common
secret key without the need to share a password among each other.
Each party has only a shared secret with the trusted server. For
example Abdalla et al. designed such a protocol [AFP05].

4. Security of PAKEs

PAKE schemes are modelled 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 flaw: if Eve guesses the password she can subvert
the exchange. Therefore to consider security of a PAKE it is
necessary to model the difficulty of that happening. If the

probability of 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, if Eve is unable to take
information from a passive attack or a single active attack and
enumerate through her dictionary then the only attack left is
repeated guessing attacks. 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) and
hence is detected. In order to judge and compare the security of
PAKE schemes, security proofs in commonly accepted models should be
used. However, each proof and model is based on assumptions: Often,
a security proof shows that in case an adversary is able to break the
scheme, she is also able to solve a problem that is assumed to be
hard, like computing a discrete logarithm. By conversion, breaking
the scheme is considered as a hard problem, too. In addition, proofs
sometimes rely on idealized versions of hash functions and/or block
ciphers, called random oracles and ideal ciphers.

A PAKE scheme should come with a security proof and also clearly
state its assumptions and used models.

4.1. Implementation Aspects

Besides the theoretical security of a scheme, pitfalls when
implementing it in practice have to be considered as well. Even a
scheme that is secure in a well-defined mathematical model can leak
information via side-channels, if it is not carefully implemented.
The design of the scheme may allow or prevent an easy protection
against information leakage. In a network scenario, an adversary may
measure the time the computation of an answer takes and derive
information about secret parameters of the scheme. If a device
operates in a potential hostile environment, e.g. in case it is
implemented on a smart card, also 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 how to protect implementations of
their 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: The
mapping of a random string to an element that can be computed with,
i.e. a point on the curve. In some cases, also the opposite is
required, 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.

In case the PAKE scheme is intended to be used with ECC, the authors
should state whether there is a mapping function required and if so,
discuss its requirements. Alternatively, the authors may define a
mapping to be used with their scheme.

5. Protocol Considerations and Applications

In most cases, the PAKE scheme is a building block in a more complex
protocol like IPSEC or TLS. This can influence the choice of a
suited PAKE scheme. For example, an augmented scheme can be
beneficial for protocols that have a strict server-client
relationship. In case both parties may initiate a connection of a
protocol, a balanced PAKE may 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 retrieval of a key from escrow, there is the variant
of two parties exchanging public keys using a PAKE in lieu of
certificates-- public keys can be encrypted using a password and the
ability of each side to both know the private key associated with its
unencrypted public key and also decrypt the peer's public key
performs authenticated key distribution. This technique can be used
to parlay a short one-time code, into a long-lived 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 authentication of one side as well as mutual
authentication in addition to the authentication using the password.

The authors of a PAKE scheme MAY discuss variations of their scheme
and explain application scenarios, where these variations are
beneficial.

6. Performance

The performance of a scheme can be judged along different lines,
depending on what is the scarcest resource in the application field.
Potential metrics are 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, i.e., a scheme that requires minimal efforts on client side
is most suited. Note that for some clients the computations might
even be carried out in a hardware implementation, asking for
different optimizations compared to software.

Furthermore, the design of the scheme may also influence the cost of
protecting its 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 choice on the performance. In particular, the
optimization goals could be stated.

7. Requirements

This section formulates the requirements for PAKE schemes based on
the previous discussed properties.

R1: A PAKE scheme MUST clearly state its features regarding
balanced/augmented versions.

R2: A PAKE scheme SHOULD come with a security proof and clearly
state its assumptions and models.

R3: It SHOULD be possible to implement the PAKE scheme in constant
time.

R4: The authors MAY show how to protect an implementation of their
PAKE scheme in hostile environments.

R5: In case 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.

R6: A PAKE scheme MAY discuss its design choice with regard to performance, i.e., its optimization goals.

R7: The authors of a scheme MAY discuss variations of their scheme that allows the use in special application scenarios.

R8: The authors MUST declare the status of their scheme with respect to patents.

8. IANA Considerations

This document makes no request of IANA.

9. Security Considerations

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

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

10.2. 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, 2006.

[AFP05]   Abdalla, M., Fouque, P., and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting", PKC 2005, LNCS 3386, 2005.

[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, 1992.

[DOT11]   IEEE Computer Society, "Telecommunications and information exchange between systems Local and metropolitan area networks", Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications IEEE Std 802.11-2012, 2012.

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

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

   [RFC5246]   Dierks, T. and E. Rescorla, "The Transport Layer Security
               (TLS) Protocol Version 1.2", RFC 5246, August 2008.

   [SPEKE]   Jablon, D., "Strong Password-Only Authenticated Key
               Exchange", ACM Computer Communications Review October
               1996, 1996.

   [TR-03110]
               Bender, J., Fischlin, M., and D. Kuegler, "Security
               Analysis of the PACE Key-Agreement Protocol", ISC 2009,
               LNCS 5735, 2009.

Author's Address

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

   Phone: +49 201 5454 3694
   Email: joern-marc.schmidt@secunet.com