Crypto Forum Research Group
Internet-Draft
Intended status: Informational
Expires: May 3, 2020

Y. Sheffer
Intuit

October 31, 2019

# Review of the CFRG PAKE Proposals
# draft-sheffer-cfrg-pake-review-00

Abstract

This draft consists of the author's review of the password-authenticated key exchange (PAKE) protocols, as submitted to the IRTF's CFRG. All opinions here are the author's alone.

Status of This Memo



Copyright Notice

Table of Contents



## 1. Introduction

The CFRG took upon itself to review multiple proposed PAKE algorithms and select zero or more of them as suitable for general use in IETF protocols. Eight protocols were submitted for consideration, and they are listed on the CFRG GitHub repository: https://github.com/cfrg/pake-selection.

Over the last few months multiple reviews were submitted to the CFRG, evaluating the protocols' cryptographic quality as well as their engineering properties. As the last stage of this process, members of the CFRG Crypto Review Panel were asked to provide summary reviews, and this document is the author's contribution as a Panel member.

### 1.1. Disclaimer

The author is not a cryptographer. Specifically, I do not have the skills to prove security of such protocols, nor even to evaluate the quality of such proofs. I do, however, possess a reasonable amount of experience in integrating cryptography into protocols, including PAKE-based algorithms [RFC6124] [RFC6631].

### 1.2. Conventions used in this document

This is essentially an opinion piece and does not employ any normative language.

## 2. Preliminaries

Before diving into the individual protocols, I would like to get two important points out of the way.

### 2.1. Protocol Completeness and Clarity

CFRG has published in the past some protocols in enough detail that they can be implemented by a non-expert developer. A good example is [RFC7748]. Of the eight PAKE submissions, in my opinion only one comes close to this level of rigor. Whatever protocols are selected, CFRG must make it clear that such selection is conditional on the algorithms being republished in a detailed format. CFRG must not leave this task to the IETF working groups, because that would both duplicate work and introduce a major risk of inadvertent errors that invariably manifest themselves as vulnerabilities.

Ironically, I can quote the abstract of one of the submissions to support this position: "We observe that the original SPEKE specification is subtly different from those defined in the ISO/IEC 11770-4 and IEEE 1363.2 standards. We show that those differences have critical security implications by presenting two new attacks on SPEKE: an impersonation attack and a key-malleability attack." In other words, an under-specified protocol resulted in two different standards, both of them vulnerable. This is ironic because the paper from which this is quoted is not itself a rigorous description of the protocol that it attempts to fix.

I would propose that each of the selected protocols be published as an RFC, containing:

- A detailed description of the protocol, to a level that can be implemented by developers who are not security experts.
- Test vectors to ensure interoperability.
- Recommendations on integrating with higher-level protocols: supported identity fields and recommendations on how they should be protected, session ID and "exporter" integration, secure capability and parameter negotiation, conditions on whether and how "optional" protocol exchanges can be eliminated.
- Mandated auxiliary primitives, such as hash-to-curve and memory-hard iterated hashing.

### 2.2. Integration into Existing Protocols

The IPsec/IKE community has always been interested in PAKE as a component, both for remote access and for peer-to-peer VPN deployments. This to me justifies the selection of both a balanced and an augmented PAKE, assuming good candidates exist. It also means that the integration of such protocols into IKEv2 is relatively straightforward.

On the other hand, the TLS community has been less receptive to PAKE solutions, and as a result, the properties required from the protocol for proper integration are not as clear. It is possible that the most common deployment will be a combination of TLS, PAKE and OAuth. Arguably we should take the combination into account when defining the PAKE portion of the protocol, and resist the temptation to only consider the narrow integration of a PAKE protocol into TLS 1.3.

## 3. Detailed Review

As mentioned above, I believe we should select one balanced and one augmented PAKE protocol.

### 3.1. Balanced Algorithms

#### 3.1.1. SPAKE2

This protocol is the best documented of all the candidates. On the down side, it relies on a set of parameters that present a high value target for factorization once a quantum computer is available to an adversary, and that would break all instances of this protocol.

#### 3.1.2. J-PAKE

This algorithm is an outlier in its complexity, which also implies a significant performance penalty. For this reason I don't think it would be a realistic selection.

#### 3.1.3. SPEKE

SPEKE has been around for a long time, which is an advantage. But the quoted paper describes several attacks on concrete specifications/implementations, and Karthik's review casts doubts about the validity of the security proof presented for this protocol. As far as I can tell, the mailing list discussion has not fully clarified which exact version of the protocol is proposed and which published security proof applies to it. Specifically, does [Hao2018] apply?

#### 3.1.4. CPace

CPace is not specified as a stand-alone protocol, but rather needs to be extracted out of the AuCPace specification. Moreover, it adds a mandatory (though trivial) message round to establish a session ID. This extra round may or may not be subsumed by the higher-level protocol. Having said that, it comes with an actual security proof and no magic parameters.

### 3.2. Augmented Algorithms

#### 3.2.1. OPAQUE

OPAQUE is described as a generic framework, with two instantiations, and will have to be narrowed down when standardized. The protocol is secure against pre-computation attacks. This is a good thing of course, however I am not sure how serious this attack is in practice: while servers are often breached with attackers gaining bulk access to hashed passwords, I don't think it is common for attackers to record multiple salt exchanges and use them in a follow-on attack. OPAQUE comes with a security proof. OPAQUE is well documented, with a separate draft [I-D.sullivan-tls-opaque] on its integration into TLS.

#### 3.2.2. AuCPace

The protocol has two versions, the main paper and Appendix C ("Strong AuCPace"), which is resistant to pre-computation attacks. It is unclear which one is nominated.

#### 3.2.3. VTBPEKE

This 2017 paper extends SPEKE into a balanced PEKE that can be proven even for elliptic curves, and then again into a verifier-based (i.e., augmented) PAKE named VTBPEKE. It has a few "magic" constants which are potentially of concern - I didn't see any mention of how they should be generated.

#### 3.2.4. BSPAKE

This protocol is somewhat loosely specified, with no security proof (or even security justification/intuition) provided. So it is hard to be convinced of its fit for purpose.

## 4. Conclusions

As noted, I think the Research Group should recommend one balanced and one augmented algorithm.

Before presenting my conclusions, I would like to clarify my view about quantum resistance in this context. Steve Thomas defines "quantum annoying" as: an attacker with a quantum computer needs to solve a DLP per password guess. IMO this is too high of a bar, and once we get to the point where this is a real risk we will need to migrate to PQC for these protocols. However I think that even now, a protocol where a single DLP solve would break _all_ instances of the protocol, is too risky to adopt.

Of the balanced algorithms, I would recommend CPace. I think the extra round trip is a reasonable price to pay for a formally proven protocol. To me the potential quantum vulnerability of the SPAKE2 parameters is a showstopper.

Of the augmented algorithms, I will follow the Mozilla report and recommend OPAQUE, which appears to be the best fit into TLS, and is also a good fit into IKEv2.

## 5. Informative References

[Hao2018] Hao, F., Metere, R., Shahandashti, S., and C. Dong, "Analyzing and Patching SPEKE in ISO/IEC", IEEE Transactions on Information Forensics and Security Vol. 13, pp. 2844-2855, DOI 10.1109/tifs.2018.2832984, November 2018.

[I-D.sullivan-tls-opaque] Sullivan, N., Krawczyk, H., Friel, O., and R. Barnes, "Usage of OPAQUE with TLS 1.3", draft-sullivan-tls-opaque-00 (work in progress), March 2019.

[RFC6124] Sheffer, Y., Zorn, G., Tschofenig, H., and S. Fluhrer, "An EAP Authentication Method Based on the Encrypted Key Exchange (EKE) Protocol", RFC 6124, DOI 10.17487/RFC6124, February 2011, <https://www.rfc-editor.org/info/rfc6124>.

[RFC6631] Kuegler, D. and Y. Sheffer, "Password Authenticated Connection Establishment with the Internet Key Exchange Protocol version 2 (IKEv2)", RFC 6631, DOI 10.17487/RFC6631, June 2012, <https://www.rfc-editor.org/info/rfc6631>.

[RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <https://www.rfc-editor.org/info/rfc7748>.

## Appendix A. Document History

### A.1. draft-sheffer-cfrg-pake-review-00

- o Initial version.

Author's Address

Yaron Sheffer
Intuit

EMail: yaronf.ietf@gmail.com