Documents: 8 PAKEs, nominated to the PAKE selection process; partial reviews provided at Stage 4 (see https://github.com/cfrg/pake-selection).
Reviewer: Stanislav Smyshlyaev
Review Date: 2019-10-23
Summary: **I would recommend selecting two PAKEs (one balanced and one augmented): SPAKE2 and OPAQUE. No strong objections against: CPace, AuCPace, VTBPEKE**

1. Balanced
1.1. SPAKE2
The main issue with SPAKE2 is potential existence of a backdoor in case when the parameters M and N have not been selected in a way that their joint discrete logarithm is guaranteed to be unknown. A variant of the protocol has been proposed, which is using a hash-to-curve function – but such a change would lead to a different protocol, which requires a separate security analysis. Another possible issue is that the protocol is not “quantum annoying”, since one needs to calculate only one discrete logarithm to break any instance of the protocol.
In my opinion, this is not an important issue for the current PAKE selection process. From the security point of view (regarding “classical” attacks on key exchange protocols), SPAKE2 has such an advantage as absence of known attacks exploiting small subgroups. Nevertheless, the checks related to cofactors are mentioned in the draft, which is good. The experts do not see major issues with integrating SPAKE2 into TLS 1.3, while there is a note about minor issues with mixing-in a password value into KDF (but it seems to be possible to mix it as ePSK). There does not seem to be any major issues with integrating into IKEv2 also or IoT applications also.
1.2, 1.3. CPace and SPEKE
SPEKE and CPace are based on the same basic scheme, but SPEKE has been initially defined for the finite fields with the proof only for that case. Therefore, it seems that it is worth considering CPace, since it is defined in the general case.
The main issue with CPace seems to be about the stage of negotiating “sid” parameter. Such a stage turns the CPace into a 2-RTT protocol, which eliminates its main advantage, efficiency. Such a sid is needed to provide a proof in UC-framework. The existence of sid for UC-framework may be more a technical issue for the approach, so CPace without negotiating the sid could be considered. The important part of the protocol is a Map2Point function, which impacts the overall security of the protocol, hence a careful choice of such a primitive is required. The CPace without negotiating sid seems to be easily integrated into TLS 1.3, IKEv2 and IoT protocols.
Nevertheless, CPace should be separately defined and described (not only as a part of AuCPace) and carefully studied for the case without sid. In my opinion, if CPace is selected as recommended PAKE, these actions can be done during the further steps of writing a CFRG RFC on recommendations for PAKEs.
1.4. J-PAKE
The main advantage of J-PAKE seems to be that it does not use any hash-to-curve functions, that can lead to some vulnerabilities or backdoors. At the same time, it has significant problems with efficiency. Therefore, it seems to be much more problematic to integrate it into TLS and IKEv2. Moreover, since IKEv2 and IoT protocols are very sensitive

to the message sizes, long messages (with up to three points in a single message) in J-PAKE look like a real problem for practical usage.

There are no major problems with the security of the protocol, although some improvements of the proofs could be made (SE-NIZK-proofs, but “none of them would be nearly as practical”).

1.5. Balanced: overall

Two ideas compete: DH on password-based points as generators (CPace and SPEKE) and DH on points, which are masked with password-based points (SPAKE2).

In my opinion, only CPace and SPAKE2 can be considered in the current selection process.

For CPace the security without pre-negotiation of sid should be studied.

Since the only issue with SPAKE2 seems to be eliminating the known discrete logarithm (between M and N) problem and since it can be done (in my opinion) during the further steps of writing a CFRG RFC on recommendations for PAKEs, I would recommend SPAKE2 as a balanced PAKE.

2. Augmented

2.1. OPAQUE

OPAQUE is more a “converter” of AKEs to PAKEs using a secure OPRF. The main advantage of OPAQUE is security against precomputations, which is desirable for applications, for which augmented PAKEs are preferred.

OPAQUE can be integrated into TLS 1.3 (the method of this integration has already been specified) without any changes in the protocol.

The authors have recently updated the security proof, addressing the raised concerns about it; nevertheless, in my opinion, the security assessment is already mature enough and sufficient for considering it secure.

The protocol is also not “quantum annoying”, but, in my opinion, that cannot be treated as a major disadvantage of the protocol.

2.2. AuCPace

AuCPace is an augmented version of CPace. AuCPace itself is not secure against precomputations, but preventing precomputation is a minor change – a strong version of AuCPace is called strong AuCPace.

There are some questions to the security proof of AuCPace (one of the reviewers treats the initially submitted version of it as “rather sketchy”), but, as well as OPAQUE, the security assessment seems to be already mature enough and sufficient for considering it secure.

AuCPace is a «quantum annoying» PAKE.

Integrating AuCPace into TLS 1.3 is deeply studied in the materials – there exist some issues, but none of them seems to be critical.

2.3. BSPAKE

BSPAKE – is an augmented Elligator-version of SPAKE2. The main disadvantage of it is absence of a complete security proof (the authors just say that the security follows from the security of the underlying elements of the construction).

The blind salt mechanism is similar to the one used in OPAQUE (OPRF); the mechanism of using blind salt in AuCPace is different: in AuCPace the salt is chosen by the client during registration phase.

BSPAKE is «quantum annoying».

BSPAKE is 2-RTT, so it needs certain efforts to be integrated into TLS 1.3. It seems that a separate work of modifying the PAKE in a way similar to OPAQUE for TLS 1.3.
BSPAKE does not seem to be a solid construction with detailed security analysis, in my opinion it should not be considered to be recommended as a selected PAKE.

2.4. VTBEKE

VTBEKE – is an augmented version of TBEKE (a modified SPEKE). VTBEKE is not secure against precomputations, but it can be modified to be such by adding blind salt.
The game-based security proof is sufficient to consider the protocol secure. The situation with integrating AuCPace into TLS 1.3 is similar to the one with AuCPace, several issues have to be resolved.

2.5. Augmented: overall

In my opinion, only AuCPace, VTBEKE and OPAQUE can be considered in the current selection process. Currently only OPAQUE provides security against precomputations – and in my opinion, it is important for an augmented PAKE (otherwise, balanced PAKEs are not much less convenient for the same client-server applications).
Blind-salt versions of AuCPace и VTBEKE should be considered instead of the "plain" versions of them, but the corresponding detailed security proofs should be obtained to do so.
In addition, since integration of OPAQUE into TLS 1.3 also seems to be studied more deeply, I would recommend OPAQUE as an augmented PAKE, if no patent issues occur to be preventing it.

3. Remarks

To be considered in the future for the selected PAKEs: while integrating a PAKE into protocol, it is important to decide, on which step to negotiate PAKE parameters (e.g., elliptic curve group); cross-cipher suite security must also be taken into account.

4. Overall recommendations

Overall recommendations about the anticipated results of the PAKE selection. If we are to use PAKEs for IKEv2 or other peer-to-peer protocols, a balanced PAKE is desirable. To address the remote access applications or other client-server scenarios, it is better to also have an augmented PAKE.
Therefore, I would recommend selecting one balanced PAKE and one augmented PAKE.
I would recommend selecting two PAKEs (one balanced and one augmented): SPAKE2 and OPAQUE. In my opinion, these protocols are mature enough and do not have any significant problems; all existing concerns can be addressed during the work on a CFRG RFC on recommendations for PAKEs. CPace, AuCPace and VTBPEKE are also strong candidates (I wouldn't have any strong objections against CFRG recommending any of them).