REVIEW:

Stanislav asked me to focus on security issues (evaluation of security assessment, verification of the security proofs). Aspects considered in other reviews, such as computational efficiency, implementation challenges, suitability for TLS and IPSec, etc., are out of scope of my review.

Björn Tackmann's review had a similar onbjective and covered already many aspects. Given that we have a rather small team looking at the proposals, I refrained from repeating his work. I also think that he is much more capable of understanding the subtleties of UC-like models than I am. The same holds for Julia Hesse, I understand that she also provided input to authors of proposals which, to my knowledge, have not been made public yet, but were taken into account in the revision of some proposals.

My review covers all aPAKE proposals, and puts particular emphasis on the following additional aspects:

- BSPAKE has not been considered in detail, since it does not have a security proof. However, BSPAKE is an extension of the classical Abdalla-Pointcheval construction, and I would like to understand whether their proof applies, too. This was suggested in the BSPAKE proposal, but it seems not obvious to me that the security analysis carries over.

- The proposals use different security models. I am currently trying to understand their relation to each other and hope that I can say something meaningful soon.

- I read the revised version of the OPAQUE paper that was recently posted on the ePrint archive. The paper was posted only a few days ago.

My review is based on the following documents:

AuCPace:
(1) The most recent (at time of review) version 20190922:200043 of the ePrint paper at
https://eprint.iacr.org/2018/286.pdf
(2) CPace and AuCPace - corrigendum.pdf from
https://github.com/cfrg/pake-selection/tree/master/Candidates

BSPAKE:
(3) bspake-explicit.txt, as provided at
https://gist.github.com/Sc00bz/ef0951ab98e8e1bac4810f65a42eab1a
(4) The comments on requirements from
https://github.com/cfrg/pake-selection/blob/master/Candidates/BSPAKE.md
(5) The original paper by Abdalla and Pointcheval from
https://www.di.ens.fr/~mabdalla/papers/AbPo05a-letter.pdf

OPAQUE:

(6) The most recent OPAQUE draft at
https://tools.ietf.org/html/draft-krawczyk-cfrg-opaque-03
(7) The paper at https://eprint.iacr.org/2018/163.pdf, currently most recent version (20191021:232825)
(8) The comments on requirements from
https://github.com/cfrg/pake-selection/blob/master/Candidates/OPAQUE.md

VTBPEKE:
(9) The paper at
https://www.di.ens.fr/david.pointcheval/Documents/Papers/2017_asiaccsB.pdf
(10) The comments on requirements from
https://github.com/cfrg/pake-selection/blob/master/Candidates/VTBPEKE.pdf

AuCPace:

The security analysis of this protocol is conducted in the UC framework, based on the classical ideal aPAKE functionality originally described by [Gentry et al.; CRYPTO 2006].

I am not sure which model exactly is used for the security proof. Figure 8 describes a slightly modified variant of the Gentry et al. '06 model, so I assume this one. However, §5.1.1 discusses the F_apwKE functionality and first states "for our real world protocol, we could not use it as-is", but then a few sentences later "but we finally decided to stick with it". This is confusing and should be clarified.
(Btw., in the same paragraph: I but did not understand the comment on "pepper", even though I know rainbow tables, salting and "peppering". What do you mean by "would allow for rainbow tables" in this context?)

The construction comes with a full security proof, but I found it very sketchy. More precisely, the proof in §5 describes a sequence of games, but the difference between two consecutive games is never bounded. For games G1 and G2 this seems fine to me, since the argument is rather obvious (but still it appears a bit sloppy that the exact probability of abort events is not provided). In G2, it is not clear what exactly is meant by "in case the adversary manages to guess". I can guess what the authors mean, but a proof should not force the reader to guess what makes sense here, but rather define such abort events explicitly.

Games 3 and 4 are the core of the proof of Thm. 2. The description of G3 mixes the description of the experiment with a brief analysis, I was not able to verify that the changes introduced in this game indeed provide a proper simulation that is indistinguishale from G2. I would suggest to split up the different arguments made in this step into several game hops, and then to analyze each hop individually, this would improve readability and verifiability of the proof significantly. The same holds for G4.

In summary, I see no reason to doubt that the scheme is secure, and the paper contains a proof sketch that overall seems plausible, but it is lacking clarity and rigor and I have not been able to verify it in detail. However, I think that the proposal is an interesting candidate and should still be considered, the issues with the proofs appear fixable, as far as I can tell at this point.

BSPAKE:

The proposal does not clearly specify the security model and assumptions. The comment on REQ2 in (4) merely describes the changes to the protocol, without stating clearly what the desired goals are and how the proposed changes achieve them exactly. The model form (5) does not allows for adaptive corruptions and does not cover forward security, while the game-based model from the VTBPEKE paper does.

I read the original Abdalla and Pointcheval '05 paper (5), but do not see how the security analysis carries over, it seems not obvious. I am not claiming that BSPAKE is insecure, but I also do not think that a proper formal security proof is as trivial as suggested and would require a significant amount of additional work and deeper analysis. It is also mentioned in (4) that parts of OPAQUE's security proof should carry over, but I consider this as even less obvious. At this point, the proposal should be considered as having no security proof at all (at least currently), but it might be possible to give a proof.

OPAQUE:

The F_saPAKE+ model considered in this paper is the first to rule out precomputation-based dictionary attacks on the server's password database. It extends the classical model by [Gentry et al.; CRYPTO 2006]. That is, in previous security models, and attacker might perform precomputations, based on the password dictionary and possibly the salt used for individul user's passwords. As soon as a server is compromised, the attacker might then *immediately* determine the user's password, by using the precomputed data. The new model considered here prevents this. Even after obtaining the server's password database, the attacker has to "pay" for each password guess when interacting with the idealized functionality. A protocol that securely realizes this ideal functionality is thus secure against this type of attacks.

The model w. r. t. which the proposed generic protocol construction is proven secure slightly differs from F_saPAKE+. The so-called F_saPAKE model additionally allows for GuessPassword-queries even before the server's database is compromised, but where the attacker receives the response only after the server is compromised. It seems to me that this is required only to make a step in the simulation go through. Even though it appears weird, I do not see how it could make any difference in practice.

The approach of this proposal looks very plausible to me. The proof is very carefully written and analyzed. The model considered by OPAQUE seems to be the strongest security model among all submissions.

So, in summary, my opinion about the security analysis of OPAQUE is very positive, its design approach is very clear and the level of details in the paper is exceptional (e.g., the security proof of the generic construction of an saPAKE from AKE+OPRF spans over 15 pages; OPAQUE is an instantiation of this construction). But I have to admit that I was probably not able to grasp all subtleties before the review deadline. This is due to my lack of experience with UC-based security models, I hope that people more familiar with UC will also have a deeper look into this proposal, since it is a very promising candidate.

VTBPEKE:

The security model considered in (9) is game based, as in (5), but it is significantly stronger, as it allows for adaptive corruptions and covers forward security.
The security proof is based on "gap" assumptions ("gap Diffie-Hellman" and "gap simultaneous Diffie-Hellman"). Those assumptions can be proven in idealized models, such as the generic group model, they are rather strong, but seem acceptable to me. The security analysis is thorough and clear. All assumptions are precisely specified and the security model is very clearly defined. The security proof is compact and occasionally sketchy, but overall appears sound and correct.

A comment on game-based vs. UC-based security models:

As far as I see, the main difference between game-based and classical UC-based security models is that the former usually assume a uniform distribution of passwords over a "small" password space. In contrast, UC-based models are able to consider arbitrary password distributions.

I got the impression that the assumption of uniformly distributed passwords is not really a weakness of the considered schemes, but rather a compromise that is necessary to formally define the advantage of a "trivial" adversary in a simple way in a game-based model. In contrast, the simulation-based formulation of UC allows to capture arbitrary password distributions more easily. While this is theoretically more general, I cannot think of a convincing example where this would make an actual difference in practice.

The VTBPEKE paper also states that its analysis can be extended to considering the min-entropy of passwords instead of a uniform distribution, or to consider only most likely passwords as in [Bresson e.a., PKC 2004]. This all appears plausible.

Hence, in theory, UC-based models appear stronger, but I currently do not see any convincing arguments that would make it really necessary to prefer UC-based security proofs over game-based ones when choosing a PAKE for standardization. Still, not being able to cover arbitrary password distributions is clearly a limitation of current game-based models.