Prudent Practices in Security Standardization

Feng Hao
Department of Computer Science
University of Warwick, UK
feng.hao@warwick.ac.uk

Abstract—**From June 2019 to March 2020, IETF conducted a selection process to choose password authenticated key exchange (PAKE) protocols for standardization. Similar standardization efforts were conducted before by IEEE (P1362.2) and ISO/IEC (11770-4). An important hallmark for this IETF selection process is its openness: anyone can nominate any candidate; all reviews are public; all email discussions on the IETF mailing lists are archived and publicly readable. However, despite the openness, it is unclear whether this IETF selection process has presented a successful model. Several important questions that were raised during the selection process had remained unaddressed even after the two winners (CPace and OPAQUE) were announced. We reflect on the IETF PAKE selection process as a case study, and summarize lessons in a set of principles with the hope to improve security standardization in the future.**

I. Introduction

On 1 June 2019, Internet Engineering Task Force (IETF) started an open process to select password authenticated key exchange (PAKE) protocols for standardization. This process was coordinated by the Crypto Forum Research Group (CFRG), which is a research group under the Internet Research Task Force (IRTF) with the task to provide advice on cryptography to many activities in IETF. After ten months of public reviews and discussions, the selection process was concluded on 20 March 2020. Two winners were announced: CPace [6] and OPAQUE [11] for balanced and augmented PAKEs respectively.

An important hallmark for this IETF PAKE selection process is its openness. Anyone could nominate any candidate. Nominators needed to answer a set of pre-defined questions to address how the nominated candidates would meet the expected criteria. Researchers from both academia and industry were invited to review the nominated candidates and to provide feedback. All the collected reviews were openly accessible on a public Github repository (https://github.com/cfrg/pake-selection). Follow-up discussions on the CFRG mail list were archived and publicly readable. A CFRG review panel was formed to oversee the whole selection process. Memberships in the panel were voluntary (subject to the approval by the CFRG chairs).

The selection process was split into two rounds. In the first round, reviewers were invited to openly comment on the nominated candidates. Based on the collected comments, four members in the CFRG review panel (Tackmann, Smyshlyaev, Housley and Sheffer) wrote their own reviews, and decided which candidates should be chosen to proceed to the next round. No panel summary was provided apart from the four separate reviews, but the selection result was clearly based on the majority view among the four panel members. Nominators of the remaining candidates were asked to answer additional questions raised during the first-round review. Their answers were examined by additional reviewers in the second round. Finally, a slightly different review panel (Tackmann, Hesse, Housley, Fluhrer) provided four final reviews, based on which the winners were chosen. Again, no panel summary was provided for this round, but the final decision was clearly based on majority voting among the four panel members.

This selection process can be compared with previous similar standardisation efforts by IEEE (P1362.2) and ISO/IEC (11770-4). During the IEEE P1363.2 PAKE standardization project (2000-2008), access to documents and discussions was restricted to registered members only. As for ISO/IEC, standard documents are not freely available and the meetings are usually attended by national body delegates only. However, despite the openness of the IETF selection process, it is unclear whether it has presented a successful model. Neither of the two winning protocols (CPace and OPAQUE) was fully specified when they were chosen as winners. Several important questions that were raised during the selection process had remained unaddressed even after the winners were announced.

In this paper, we present a retrospective review of the IETF selection process, and summarize lessons in the hope that similar standardization activities can be improved in the future. The rest of the paper is organized as follows. Section II describes the background of PAKE research. Section III gives an overview of the IETF PAKE selection process with the focus to explain details of the two winners. In Section IV, we draw lessons from the IETF selection process, and present a set of principles as prudent practices in security standardization. Section VI provides further discussions on why these principles were sidestepped in the IETF selection process and the latest developments. Section VII concludes the paper.

II. Background on PAKE

A password authenticated key exchange (PAKE) protocol is a technique that allows two parties to establish a high-entropy session key based on a shared low-entropy secret (e.g., a memorable password) without requiring any trusted third parties. The first PAKE protocol, called Encrypted Key Exchange (EKE), was proposed by Bellovin and Merritt in 1992 [3]. In the next 30 years, many other protocols were proposed [2], [6], [8], [9], [11], [13].

PAKE protocols can be categorized into two types: balanced and augmented PAKE. A balanced PAKE assumes two parties share a common password. When designing J-PAKE, Hao and Ryan summarized four main requirements for a balanced PAKE in 2008, including "offline dictionary resistance", "forward secrecy", "known-session security" and "online dictionary attack resistance" [8]. The same requirements were later formalized in a model in 2015 due to Abdalla et al. [14]. The real-world deployments of J-PAKE in the past decade (secure sync and IoT commissioning) suggest that these requirements are realistic and meet the real-world needs.

An augmented PAKE differs from a balanced PAKE by adding an extra requirement called "server compromise resistance": namely, when the server is compromised, the attacker should not be able to use the stolen credential to impersonate a client to the server without first performing an offline dictionary attack. However, when the server is compromised, an offline dictionary attack is inevitable. Hence, all passwords need to be updated regardless what augmented PAKE scheme is used. OPAQUE [11] further adds another requirement called "pre-computation resistance". Under this requirement, an attacker should perform a (standard) offline dictionary attack without being able to use pre-computed tables to speed up the search. This increases the attacker's effort, but cannot prevent the password in plaintext to be uncovered. From this perspective, the real assurance for "pre-computation resistance" is limited.

As a trade-off for the (limited) resistance when the server is compromised, an augmented PAKE requires a registration phase that must be done within a pre-existing secure channel. In the example of OPAQUE, if the registration data is sent over a network and is eavesdropped by a passive attacker, the password will be trivially broken by an offline dictionary attack. By comparison, in a balanced PAKE, the shared secret can be communicated using a low-bandwidth out-of-band channel, e.g., reading a short code displayed on one device and entering it to another device or reading it out to another person over the phone. In OPAQUE, the registration phase involves exchanging long strings of data between the client and the server, which cannot be done over an out-of-band channel. The OPAQUE authors justify this design choice [11], "In this way the server never sees the user's password, a major benefit, for example to avoid accidental storage of plaintext passwords that has affected also security conscious companies". OPAQUE recommends using SSL/TLS to perform registration, but this will reduce arguably the biggest benefit of PAKE: namely, being free from any Public Key Infrastructure (PKI). We will explain more details on OPAQUE in Section III.

III. IETF PAKE Selection

A. Overview

The call for an IETF PAKE selection process followed after the completion of the TLS 1.3 specification. TLS 1.3 is primarily designed for enabling secure communication between client browsers and remote servers based on a pre-existing PKI. However, there are many applications, where secure communication is strongly demanded but a PKI does not exist, e.g., Internet of Things (IoT). Creating a new PKI for IoT will be tremendously difficult. Although TLS 1.3 has a pre-shared key (PSK) mode that allows two parties to establish secure communication based on a shared secret, in a typical IoT application, the two remote devices can only share a low-entropy secret (e.g., a short memorable code). When a low-entropy secret is used, the PSK mode is insecure, being vulnerable to offline dictionary attacks. Hence, PAKE naturally rises as a compelling solution when a PKI is unavailable (or not securely operational).

Eight PAKE protocols were submitted to this IETF selection process. Table I summarizes the main features of these protocols. Here, we use published papers as primary sources for the analysis (except BSPAKE which has no formal publication). During the selection process, IETF Internet Drafts (ID) documents were created with the aim to fully specify CPace and OPAQUE, but these ID documents kept evolving and had remained unpublished even when the selection process was finished. Note that the content of an ID document is not fixed unless it is published. Therefore, we will refer to these ID documents only when additional clarification is needed.

We should highlight one particular ID document entitled "Usage of PAKE with TLS 1.3" (current version 4, dated 16 July, 2018) by Barnes and Friel. This ID document was frequently cited during the IETF PAKE selection process as an authoritative reference for the usage of PAKE in TLS 1.3. In this document, the authors state: "*It must be possible to execute in one round-trip, with the client speaking first*" (draft-barnes-tls-pake-04). The goal of minimizing communication rounds, and hence latency, is totally reasonable, however, the question is whether the word "must" is overly assertive in a general context. According to this requirement, certain candidates such as J-PAKE (3 flows, hence 1.5 round-trips) had been disqualified automatically even before the selection began. In the following, we will explain technical features of the submitted candidates in terms of round-efficiency, dependence on trusted setup (TS) and dependence on hash-to-curve (H2C).

Round efficiency. In a two/multi-party computation system, a "round" is defined as a step in which all participants can complete operations without depending on each other. Among all candidates, only SPEKE, in both the original [9] and the revised [7] versions, is one-round (which can be implemented in 2 flows). The difference between the two versions is that the revised SPEKE patches all known attacks reported on the original SPEKE without changing round efficiency. (The revised version has been included into the ISO/IEC 11770-4:2016 standard.) SPAKE2 is a 2-round protocol, which can be implemented in two flows (note that it is not 1-round since the order of the user identities in the input to the key derivation function implies that the two flows cannot be completed in one round). Similarly, OPAQUE is a 2-round protocol, which can be implemented in 2 flows. J-PAKE is a 2-round protocol, which can be implemented in 3 flows. From the perspective of round efficiency, SPAKE2, SPEKE and OPAQUE require the minimum number of flows and satisfy the "one round-trip" requirement stated by Barnes and Friel. CPace claims to be "one round" (two flows) during the selection process, but this claim is based on incorrect assumptions. It actually requires 3 rounds (3 flows) based on the description in the published paper [6]. We will explain CPace in more detail later.

Trusted setup. Several protocols including SPAKE2, VTBPEKE and BSPAKE depend on a trusted setup. More specifically, the security of these protocols critically depends on the discrete logarithm (DL) relationship between two base generators in the system setup being unknown. SPAKE2 [2] uses three base generators (denoted g, M and N in the paper); the DL relationship between any two of them must remain unknown to anyone. If a DL relationship is known, the protocol is completely broken. Note that if an attacker manages to discover the DL relationship, they will keep it secret and hence gain exclusive power to break key exchange sessions without anyone else knowing.

Hash-to-curve. Several protocols including CPace, OPAQUE, AuCPace and BSPAKE, critically rely on a hash-to-curve (H2C) function to securely map an arbitrary string to a random point on a designated elliptic curve. It is further assumed that this mapping is a constant-time operation and incurs little cost. To date there is no general solution to perform such secure mapping for elliptic curves (one that was proposed in Dragonfly and implemented in WPA3 has been shown insecure by Vanhoef and Ronen in IEEE S&P'20). Instead, an Internet Draft (ID) document (draft-irtf-cfrg-hash-to-curve) was created trying to define a custom-built H2C solution for each of the ten selected elliptic curves. Consequently, any protocol that relies on this ID document has the specification and the implementation detail inevitably mixed together. Each of the 10 custom-built H2C functions is essentially a new security primitive on its own, and their security needs to be reviewed and established separately. During the selection process, the H2C ID document kept evolving and had remained unpublished when the PAKE selection process was finished. We note that it is not possible to establish properties of a new security primitive unless its specification is *fully defined* and *fixed*. From this perspective, the H2C function remained uninstantiated throughout the selection process.

The critical reliance on an unpublished H2C ID document creates two major problems. First, as H2C was not instantiated, the specifications of CPace and OPAQUE remained incomplete when they had been chosen for standardization. The second problem is causing ambiguity in the protocol specification. In the original CPace [6] and OPAQUE [11] papers, both protocols are described using notations for a multiplicative group over a finite field (FF), e.g., using modular exponentiations with respect to a large prime modulus. This is common practice in the cryptographic literature. The implicit assumption is that the same protocol can be implemented in an elliptic curve (EC) setting, e.g., by changing modular exponentiations to scalar multiplications. However, because of the critical reliance on H2C, this means both protocols were *undefined* in the FF setting (Table I). As for the EC setting, the specifications of both protocols were *incomplete* due to the H2C primitive not concretely instantiated. During the PAKE selection process, several reviewers simply counted the number of modular exponentiations as described in the CPace and OPAQUE papers without realizing that the two protocols were undefined in this FF setting.

We note that it is possible to define CPace and OPAQUE in an FF setting, e.g., by following SPEKE [7], [9] which uses a safe prime as the modulus and a square operation to map a hashed string to a group element (this achieves a similar effect as H2C in an FF setting). If the designers had done so, it will be explicitly clear that both protocols are in fact rather inefficient compared to others – e.g., for a typical 2048-bit modulus p, one modular exponentiation $(\bmod p)$ in CPace and OPAQUE will be about 9 times costly as that in J-PAKE due to the use of long exponents [8]. This shows the importance of having a complete specification to give a full picture. Simply counting the number of modular exponentiation without the underlying group fully specified can be misleading. We will describe CPace and OPAQUE in more detail below.

B. CPace

The CPace protocol due to Haase and Labrique was published in 2019 [6]. Figure 1 summarizes the protocol. CPace is essentially a variant of SPEKE with two notable modifications. First, while SPEKE hashes a password to derive a base generator, CPace adds more inputs to the hash function including a sub-session ID (ssid) and a channel identifier (CI). The ssid is defined to be a random string jointly agreed by Alice and Bob, and the CI is defined as a concatenation of the identities of the two parties. A consequence of this change is that each user needs to know ssid and CI even before any communication starts. This is clearly unrealistic. When additional communication is considered to transmit these values as part of the key exchange process, the actual round efficiency for CPace is 3 rounds (3 flows). The second modification is to use a new H2C function (called Map2Point in Figure 1) to map the output of a hash function to a random point on a designated elliptic curve. No H2C function is used in SPEKE, but the protocol [7] is only defined in the FF setting (see Table I).

C. OPAQUE

The OPAQUE protocol due to Jarecki, Krawczyk and Xu was published in 2018 [11]. Figure 2 shows a schematic representation of the OPAQUE protocol. The protocol has two stages: registration and login. The registration should be done over a pre-existing secure channel (clearly, if k and ρ_s are captured by an eavesdropper, the password pw will be trivially broken through offline dictionary search). The login is a 2-round protocol which can be implemented in 2 flows.

While a high-level description of OPAQUE is presented in the original paper [11], a complete specification of the protocol is lacking in two aspects. First of all, the protocol uses Oblivious Pseudorandom Function (OPRF) as a basic

Name	Type	Pub.	Original Designer(s)	Rnd (flow)	Exp (FF)	Mul (EC)	Rely on H2C	Rely on TS	Submitter(s)
SPAKE2	Bal.	2005	Abdalla, Pointcheval [2]	2(2)	3	2	No	Yes	Kaduk, Ladd
J-PAKE	Bal.	2008	Hao, Ryan [8]	2(3)	14	11	No	No	Hao
SPEKE	Bal.	1996	Jablon [9] (revised [7])	1(2)	2 (×9)	U/D	No	No	Harkins
CPace (*)	Bal.	2019	Haase, Labrique [6]	3(3)	U/D	2 + H2C	Yes	No	Haase
OPAQUE (*)	Aug.	2018	Jarecki, Krawczyk, Xu [11]	2(2)	U/D	C: 2 + H2C + AKE S: 2 + AKE	Yes	No	Krawczyk
AuCPace	Aug.	2019	Haase, Labrique [6]	4(4)	U/D	C: 3 + H2C S: 4 + H2C	Yes	No	Haase
VTBPEKE	Aug.	2017	Pointcheval, Wang [13]	3(3)	4	4	No	Yes	Wang
BSPAKE	Aug.	2019	Thomas (Email 30/06/19)	3(3)	U/D	N/A	Yes	Yes	Thomas

TABLE I

OVERVIEW OF CANDIDATES IN THE 2019 IETF PAKE SELECTION. TEXT IN RED REPRESENTS POTENTIAL ISSUES OR DRAWBACKS. (*) MARKS THE WINNERS OF THE SELECTION PROCESS. IN ALL PROTOCOLS, ONLY IMPLICIT KEY CONFIRM IS CONSIDERED; EXPLICIT KEY CONFIRMATION CAN BE ACHIEVED BY ADDING ONE MORE ROUND OR FLOW. FOR SPEKE, WE CONSIDER A 2048-BIT SAFE PRIME p AS THE MODULUS, AND THE COST OF ONE EXPONENTIATION (2047-BIT EXPONENT) $\bmod p$ IS ABOUT 9 TIMES COSTLY AS AN EXPONENTIATION (224-BIT EXPONENT) IN J-PAKE AND SPAKE2. FOR OPAQUE AND AUCPACE, WE CONSIDER THE COST FOR LOGIN ONLY (REGISTRATION COST EXCLUDED). U/D REFERS TO "UNDEFINED". N/A REFERS TO "NOT AVAILABLE" (DUE TO INCOMPLETE SPECIFICATION).

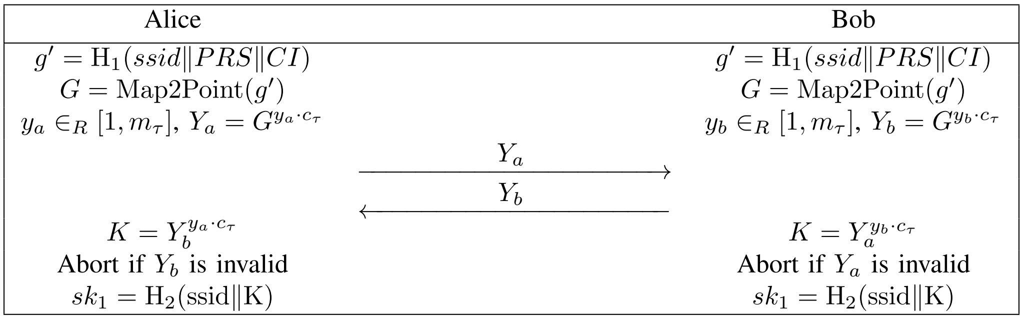
Alice
Bob
$g' = \mathrm{H}_1(ssid\|PRS\|CI)$
$G = \mathrm{Map2Point}(g')$
$y_a \in_R [1, m_\tau]$, $Y_a = G^{y_a \cdot c_\tau}$
$g' = \mathrm{H}_1(ssid\|PRS\|CI)$
$G = \mathrm{Map2Point}(g')$
$y_b \in_R [1, m_\tau]$, $Y_b = G^{y_b \cdot c_\tau}$
Y_a
Y_b
$K = Y_b^{y_a \cdot c_\tau}$
Abort if Y_b is invalid
$sk_1 = \mathrm{H}_2(\mathrm{ssid}\|\mathrm{K})$
$K = Y_a^{y_b \cdot c_\tau}$
Abort if Y_a is invalid
$sk_1 = \mathrm{H}_2(\mathrm{ssid}\|\mathrm{K})$

Fig. 1. CPace protocol [6]. H_1, H_2: two independent hash functions. *ssid*: sub-session ID. *PRS*: Password Related String. *CI*: Channel Identifier. Map2Point: map a hash output to a group element on a designated elliptic curve. m_τ: order of the group. c_τ: co-factor.

building block. A critical element in OPRF is a special hash function H' (see Figure 2), which should map a password to a random group element in *constant time*. However, no construction of H' is provided in the OPAQUE paper [11]. To fully specify OPAQUE, an ID document (initially called draft-krawczyk-cfrg-opaque, and later replaced by draft-irtf-cfrg-opaque) was created for the PAKE selection process. This OPAQUE ID document relies on another H2C ID document (draft-irtf-cfrg-hash-to-curve) for the definition of H'. At the end of the selection process when OPAQUE was announced as a winner, none of these ID documents was published. The specification of OPAQUE remained incomplete and not fixed.

A second building block in OPAQUE is an authenticated key exchange scheme called *KE* (see Figure 2). The OPAQUE paper proposes to use HMQV as the "most efficient" way to instantiate *KE* without giving full details. HMQV, as an AKE primitive, has the issue of being ambiguous on whether the public key validation is required. The original HMQV paper states that it is sufficient for each party to check if a received (static or ephemeral) public key "is not 0 or 1". However, this is not the same as the public key validation, and as a result, the protocol is vulnerable to a small-subgroup confinement attack [12] as shown by Menezes. The use of HMQV to instantiate the AKE in OPAQUE was described in the draft-krawczyk-cfrg-opaque ID document during the PAKE reviewing process. After the selection was concluded, another ID document draft-irtf-cfrg-opaque was created to replace draft-krawczyk-cfrg-opaque with the aim to fully specify OPAQUE in the post-selection time. A Github repository was also created as a working area (https://github.com/cfrg/draft-irtf-cfrg-opaque) to continue to specify OPAQUE. In some of the versions (e.g., draft-krawczyk-cfrg-opaque-3), public key validation is explicitly mandated in HMQV, but in later versions, such an explicit statement is removed. This is potentially confusing. Whether or not public key validation is required is an important detail in a protocol specification, which should be consistently and unambiguously stated.

As of October 2020, the use of HMQV was no longer recommended for OPAQUE (citing patent as a reason); instead, different AKE protocols, namely, 3DH and SIGMA, were proposed to replace HMQV. Krawczyk clarified this issue on 28 October 2020 by stating: "*It has to be understood that these AKEs are informational/illustrative, not intended as full specifications. Having the unspecified/optional info fields makes it very clear that we are not providing a full specification*" (https://github.com/cfrg/draft-irtf-cfrg-opaque/issues/77). However, the lack of a full and stable specification has also made it difficult to precisely analyze the security and efficiency properties of OPAQUE.

Init registration		
U		S
$pw, \rho_u \in_R Z_q, P_u = g^{\rho_u}$	$\xleftarrow{k, P_s = g^{\rho_s}}$	$k \in_R Z_q,\ \rho_s \in_R Z_q$
$rw = H(pw, H'(pw)^k)$		
$c = AuthEnc_{rw}(\rho_u,P_u,P_s)$	$\xrightarrow{c}$	Stores k, ρ_s, c
Login		
U (pw)		S (k, ρ_s, c)
$r, x \in_R Z_q$		
	$\xrightarrow{\alpha = H'(pw)^r, X = g^x}$	
	$\xleftarrow{\beta = \alpha^k, c, Y = g^y}$	$y \in_R Z_q$
$rw = H(pw, \beta^{1/r})$		
$\rho_u, P_u, P_s \leftarrow AuthDec_{rw}(c)$		
$K = KE(\rho_u, x, P_s, Y)$		$K = KE(\rho_s, y, P_u, X)$

Fig. 2. Schematic representation of OPAQUE [11]. H: a standard one-way hash. H': a special hash function that maps an arbitrary string to a random group element. pw: password. $AuthEnc$: authenticated encryption. $AuthDec$: authenticated decryption. KE: (authenticated) key exchange.

IV. Recommendations

Reflecting on the IETF PAKE selection process, we present a set of principles as guidelines to improve security standardization in the future. Although these principles are drawn from the IETF selection case study, we believe they are generally applicable, e.g., for the standardization activities in IEEE, ISO/IEC and NIST as well.

A. Selecting standardization candidates

First of all, we start with seemingly the obvious: the completeness principle.

> **Principle 1 (*completeness*):** Be complete with the system specification. All details should be fully specified. There should be no dependency on any external functions that are undefined or incompletely defined.

In the IETF PAKE selection process, the two winning protocols, CPace and OPAQUE, critically depended on a H2C function, which was not appropriately instantiated throughout the selection process. Nominators of CPace and OPAQUE assumed that details of this function would be completed by other people in the future. So did the reviewers. OPAQUE had a further issue that the AKE instantiation was not fully defined during the selection process. The full specification was left as follow-up work to complete in the post-selection time. We stress that in security, every detail matters. The decision of choosing a security technique for standardization should be based on what it is rather than what it might be in the future.

> **Principle 2 (*explicitness*):** Be explicit with any properties related to security analysis and implementation, such as assumptions, system parameters, the precise sequence of operations etc.

The explictness principle is one of the most important engineering principles in designing robust public key protocols. As stated by Anderson and Needham in 1995: “Robust security is about explicitness” [1]. Here, we extend this principle to the context of security standardization.

One of the prudent practices in the previous PAKE standardization efforts by IEEE P1363.2 and ISO/IEC 11770-4 is to explicitly define a PAKE protocol in two separate settings: FF and EC. Doing so would have avoided the confusion of mixing these two settings during the evaluation.

Another prudent practice, adopted in the NIST AES selection process, is to ask nominators to provide reference implementations. Doing so would have compelled the designers to fully specify all details in their proposed systems.

CPace explicitly claims to be “one round” (draft-irtf-cfrg-cpace-1), but makes two implicit assumptions. First, it defines a random sub-session ID (ssid) and implicitly assumes it is known by both parties before the communication starts. In fact this assumption is an artefact from the way the CPace protocol is defined. In the original paper, there is no explicit specification of CPace. CPace is derived from AUCPace by taking a portion of the key exchange flows. The use of ssid is actually inherited from the earlier flows in AUCPace. But this context is lost when CPace is extracted as a standalone protocol. Second, CPace implicitly assumes each party knows (or “remembers”) the exact identity to be used by the other party even before the communication starts. If a party “misremembers” the other party’s exact identity, the key exchange will fail even if the two parties have used the same password. This deviates from the established practice in the field of PAKE research that user identities are established and verified as part of the key exchange process rather than being “remembered” by the other party beforehand.

> **Principle 3:** Be sensible with security proofs. Security proofs are meaningful only when the system is fully specified. Also, security proofs need to be checked, which takes times.

First of all, we stress that security proofs are meaningful only when the system is fully specified. We use EKE as an example to illustrate this. EKE is the first PAKE protocol proposed in 1992 [3]. It works by using a password as a symmetric key to encrypt the Diffie-Hellman key exchange items. However, in 1996, Jaspon reported an offline dictionary attack against EKE [10]. This attack is caused by the fact that a low-entropy password is simply too weak to be used as a key for any symmetric cipher (e.g., AES). At Eurocrypt'00, Bellare, Pointcheval and Rogaway proposed a formal model for EKE and applied this formal model to prove that "the two-flow protocol at the core of EKE is a secure AKE [Authenticated Key Exchange]" [5]. However, the formal security proofs in the BPR model require an "ideal cipher". It is assumed that this "ideal cipher" does not leak any information even when a weak key is used (hence avoiding the attack reported by Jaspon). It is further assumed that this cipher works like a random function in encryption, but must map fixed-sized strings to group elements in decryption [4]. Clearly, no such cipher existed. Nonetheless, during the IEEE P1363.2 standardization project (2000-2008), many people believed that the core problem in EKE had been solved with formal security proofs, hence the remaining (simpler) problem was only to instantiate an "ideal cipher". Unfortunately, this proved much harder than what many had hoped. No secure instantiation of this "ideal cipher" had been found in the next 8 years. In the end, EKE was not included into IEEE P1363.2 when the project was concluded in 2008. Even today, this "ideal cipher" in EKE remains uninstantiated.

Second, we emphasize the importance of the fact that security proofs need to be checked, which unfortunately has been neglected by many. We use SPEKE as an example to illustrate this. SPEKE is one of the most well-known PAKE protocols, and is included in IEEE P1363.2 and ISO/IEC 11770-4 standards. The original SPEKE protocol [9] was proposed in 1996 with no security proofs. Concerns on a lack of security proofs were raised during the IEEE P1363.2 standardization process. To address this issue, in 2001, MacKenzie published a paper on IACR ePrint (2001/057) with a formal model and security proofs to show that SPEKE is "provably secure". Although IACR ePrint is not a formal publication, this paper was included in IEEE P1363.2 (2008) and ISO/IEC 11770-4 (2006) to support that SPEKE had formal security proofs. However, in 2014, Hao and Shahandashti pointed out two attacks on the "standardized" version of SPEKE in IEEE P1363.2 and ISO/IEC 11770-4:2006. Apparently, what was proved in MacKenzie's paper was actually a modified version of SPEKE (e.g., details in the key flows are different and key confirmation is mandatory rather than optional). Hence, the formal security proofs were simply not applicable to the versions defined in the IEEE and ISO/IEC standards. This apparent discrepancy had not be found for 13 years.

Both the CPace and OPAQUE protocols claim to have formal security proofs in a universally composable (UC) model. However, neither protocol was fully specified. The underlying assumption in the security proofs for the "1-round CPace" is that each party knows the ssid and the other party's exact identity to be used before key exchange. This assumption is clearly unrealistic. In OPAQUE, details of the AKE instantiation are not explicitly defined and fixed. Both protocols also critically depended on H2C, which remained uninstantiated throughout the selection process. In the absence of these security-critical details in a full system specification, the security proofs have limited value.

> **Principle 4:** Do not sidestep the scrutiny of time. Time is needed for any new security system to mature before it can be considered for standardization.

A prudent practice in the ISO/IEC JTC 1/SC 27 standardization of security techniques is that any technique nominated for standardization should meet the requirement of having received sufficient time of public scrutiny, typically at least 3 years from the date of the first publication. It is reasonable to expect that the specification of a security technique be complete, stable and published in public domain for sufficient time before it can be considered for standardization.

We note that in the IETF selection process, both the CPace and OPAQUE protocols had only about 1 year public scrutiny time since their first publication. The short time of public scrutiny was further compounded by the fact that neither protocol was fully specified in the original publication. The specifications kept evolving and were not finalized even when the selection process was concluded.

V. Managing Selection Process

> **Principle 5:** Be careful with modifying requirements. Do not underestimate the difficulty of formulating a requirement rigorously.

As part of the nomination for BSPAKE, Thomas added a new requirement for PAKE, namely, "quantum annoying". From Thomas, "This property means that quantum computers need to solve a DLP [Discrete Logarithm Problem] for each password guess." This requirement was never considered before in the PAKE research field, but became one of the main considerations for the PAKE selection. Although questions on the rigor of this requirement were raised, the CFRG review panel still listed "quantum annoying" as one of the questions to be considered, hence, implicitly endorsing this requirement. CPace claims to fulfill this requirement. OPAQUE also claims to fulfill this requirement with simple changes (using quantum-resistant primitives for OPRF and AKE).

However, the so-called "quantum annoying" requirement is not rigorously defined at all. First of all, it assumes a quantum computer, which by definition can solve a DLP efficiently in polynomial time. Under this assumption, all the submitted PAKE protocols are trivially broken. Suppose we have two protocols: A and B. An attacker is required to solve 1 DLP to break the password in Protocol A, but is required to solve 2 DLPs (or a polynomial number) to break the password in Protocol B. Which protocol is more secure? One might be tempted to think Protocol B is more secure as it requires more effort from the attacker. But actually neither protocol is secure in a quantum setting. This is similar to arguing whether a polyalphabetic cipher is more secure than a mono-alphabetic cipher when the attacker has access to a computer. Both ciphers are trivially broken. Arguing which one is more secure between two broken ciphers is not meaningful.

> **Principle 6:** Be careful with modifying a security technique. A small change is never small, and it can cause profound effects.

Questions on the ssid and round-efficiency problems of CPace were raised during the first round of the selection process. However, instead of addressing these questions, the CFRG review panel chose CPace into the second round with a request to CPace designers to modify the protocol "without negotiation of sid". This request had remained unfulfilled til the end of the second round. Nonetheless, the CPace protocol was still chosen as a winner. The issues on ssid, user identities, and H2C were simply left as follow-up works to complete in the post-selection time.

> **Principle 7:** Be careful with majority voting. Efforts should be made to converge disagreements into a broad consensus, and ensure all issues are fully addressed before any decisions are made.

In each of the two rounds during the PAKE selection process, the CFRG review panel published four reviews from the four voluntary members based on the comments collected from other people. These four reviews were written separately, and were not consistent. For example, among the final four reviews, although 3 out of 4 members preferred CPace, one member (Tackmann) stated that "*it would be prudent for CFRG to re-evaluate the proof support after the now-discussed modification to be chosen protocol (e.g., choice of session id for CPACE, ...) are finalized.*" Rather than addressing the disparity, the panel simply announced winners based on majority voting. Any unaddressed technical issues were left as follow-up works.

Here, we do not question the use of majority voting as a decision method, but we stress that this method should be exercised with care based on appropriate conditions. In fact, all the issues discussed in the paper were raised during the PAKE selection process. It should be the panel's responsibility to spend efforts to converge disagreements into a broad consensus, and to ensure all raised issues are fully addressed before any final decision is made. In the absence of such efforts, the outcome of majority voting is not meaningful. Furthermore, when majority voting is exercised, the voting population should be reasonably sized. In the IETF PAKE selection, there were only four voters. (Recall in the final round of the AES selection process, there were 212 votes.)

Finally, we emphasize the importance of having a report from the organization committee when the selection process is concluded. In the previous AES selection process, a report from NIST was published for each of the two rounds to summarize the findings and to justify the decisions. However, during the IETF selection process, no panel report was provided for any of the two rounds during the selection. The absence of a coherent report from the panel to explain the rationale of the decision is a missed opportunity in the IETF PAKE selection process to reflect on the questions raised and make sure they have been all appropriately addressed before committing to the decision.

VI. Further discussions

Why sound principles were bypassed? Having a complete, stable and unambiguous specification is a sound requirement for the standardization of any security technique. As an example, in the "Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process" document published by NIST in 2016, it states: "*A complete written specification of the algorithms shall be included, consisting of all necessary mathematical operations, equations, tables, and diagrams that are needed to implement the algorithms.*" The same requirement is stated in the lightweight cryptography standardization process announced by NIST in 2018. However, why was this prudent practice somehow sidestepped during the IETF selection process? Both CPace and OPAQUE claim to be provably secure in a UC model, which is argued by many to be the strongest theoretical model for key agreement. To a layman, the notion of provable security implies security with certainty. One might think if a protocol has been mathematically proven secure, it should be instantly ready for standardization; any remaining issues should be minor and can be left to implementers to address. However, in the example of CPace and OPAQUE, the fact that provable security in a strong UC model can be claimed based on abstract building blocks without fully specifying all details has significantly complicated the matter (see the earlier discussion on the ideal-cipher model in the formal proof of EKE as another example). During the PAKE selection, while extensive efforts were spent on reviewing the UC formal model and the proofs for CPace and OPAQUE, little attention was paid to more fundamental questions such as whether the assumptions made in the model are actually reasonable and whether the protocol is fully specified. When people overly rely on security proofs (see Principle 3), they also tend to neglect other principles. We refer the reader to Herley and Oorschot [15] for the important separation of inductive and deductive statements, which explains the root cause for many misunderstandings on provable security from the perspective of the philosophy of science.

Why CPace and OPAQUE were chosen as winners? As the panel did not write a report, we cannot give the full picture. Still, the following elements were brought up during the relevant discussions. First of all, the reliance on a trusted setup (a potential backdoor) was commonly considered by CFRG members to be inadequate for a large-scale deployment of PAKE, which removes SPAKE2, VTBPEKE and BSPAKE. The "one round-trip" requirement stated by Barnes and Friel removes J-PAKE. When nominating SPEKE, Harkins did not specify which exact SPEKE variant was nominated. The nomination contains a reference to Hao-Shahandashti's revised SPEKE in 2014 [7] (standardized in ISO/IEC 11770-4:2016) but also refers to MacKenzie's 2001 IACR paper (2001/057) for the security proofs. However, the SPEKE protocol specifications in these papers are different. This ambiguity was never clarified by the nominator throughout the selection process, which caused confusion among the reviewers and eventually the SPEKE candidate falling out of favor. This leaves CPace as the only balanced PAKE, and AuCPace and OPAQUE as two remaining augmented PAKEs. A crucial reason for CPace to be chosen as a winner is that it claims to be one-round (but as explained earlier, this is based on incorrect assumptions). Between AuCPace and OPAQUE, the latter was preferred by panel members. OPAQUE's pre-computation resistance property was perceived by some as an advantage over alternative schemes.

Current status of CPace and OPAQUE. As of June 2021 (more than one year after the selection process was finished), the H2C ID remains a draft (draft-irtf-cfrg-hash-to-curve). Both CPace and OPAQUE assume H2C as an idealized random function that returns a non-identity point (a generator) in the prime-order subgroup on an elliptic curve given a password (together with auxiliary public data) as input. In the latest H2C ID (v11), the H2C functions defined in the draft do not preclude small subgroup points in the output by design. The chance of falling into a small subgroup is negligible, but not zero. This shows a subtle but clear mismatch between the idealized assumption of H2C and what is provided by the actual construction. CPace continues to be specified in an ID (draft-irtf-cfrg-cpace). In the latest version (v1), the designers have changed the user identities (CI; see Figure 1) to be "optional" rather than "mandatory" and the ssid to be "unilateral" rather than "bidirectional". These are material changes to the protocol specification from its original form [6]. The security proof in the original CPace paper claims to be based on a computational Diffie-Hellman (CDH) assumption. However, this claim was disputed during the PAKE selection process. As a result, the assumption has been changed to "Computational Simultaneous Diffie-Hellman" in draft-irtf-cfrg-cpace-01 after the selection process was finished, and subsequently further changed to "Strong simultaneous non-uniform CDH" (and additionally, "Strong twist CDH" for certain elliptic curves) in the updated version of CPace in IACR ePrint 2021/114. OPAQUE continues to be specified in an ID (draft-irtf-cfrg-opaque). In the latest version (v5), the designers have changed to use 3DH to instantiate AKE, which is different from what was originally proposed [11], nominated and reviewed during the PAKE selection process. The use of HMQV is now included in the appendix of the ID; still, it remains unclear whether public key validation in HMQV should be mandatory or not. It has also been identified that the transmission of c during the login phase (see Figure 2) leaks information about whether a password has been recently changed, which can cause security concerns in certain applications. It is possible to stop this leakage by running OPAQUE over SSL/TLS, but that will contradict the basic goal of PAKE: removing dependence on a PKI.

VII. Conclusion

In this paper, we presented a retrospective review of a recent IETF PAKE selection process. Although the process had finished with the two winners announced, we highlighted a number of technical issues with the two winning protocols which had remained unaddressed even after the selection process was concluded. Aspects concerning the management of the selection process were also reviewed. The unaddressed issues added risks and uncertainties to the success of the standardization process, which could have been avoided in the first place. Based on this review, we presented a set of principles with the hope to help improve the standardization of security techniques in the future.

Acknowledgement

We thank anonymous reviewers for many helpful comments. We thank Paul van Oorschot, Ross Anderson, Liqun Chen, and Rene Struik for invaluable feedback and discussions.

References

[1] R.J. Anderson, R. Needham, "Robustness principles for public key protocols," Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology, LNCS 963, pp. 236–247, 1995.

[2] M. Abdalla, D. Pointcheval, "Simple password-based encrypted key exchange protocols," Proceedings of Topics in Cryptology – CT-RSA, pp. 191–20, 2005.

[3] S. Bellovin and M. Merritt, "Encrypted Key Exchange: password-based protocols secure against dictionary attacks," Proceedings of the IEEE Symposium on Research in Security and Privacy, May 1992.

[4] C. Boyd, A. Mathuria, *Protocols for authentication and key establishment*, Springer-Verlag, 2003.

[5] M. Bellare, D. Pointcheval, P. Rogaway, "Authenticated key exchange secure against dictionary attacks," Eurocrypt'00, LNCS 1807, pp. 139–155, 2000.

[6] B. Haase, B. Labrique, "AuCPace: Efficient verifier-based PAKE protocol tailored for the IIoT," IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 1–48, Vol. 2019, No. 2, 2019.

[7] F. Hao, R. Metere, S. Shahandashti and C. Dong, "Analysing and Patching SPEKE in ISO/IEC," *IEEE Transactions on Information Forensics and Security*, Vol. 13, No. 11, pp. 2844-2855, 2018.

[8] F. Hao, and P. Ryan, "Password authenticated key exchange by juggling," International Workshop on Security Protocols (SPW), 2008.

[9] D. Jablon, "Strong password-only authenticated key exchange," *ACM Computer Communications Review*, Vol. 26, No. 5, pp. 5–26, October 1996.

[10] B. Jaspan, "Dual-workfactor Encrypted Key Exchange: efficiently preventing password chaining and dictionary attacks," Proceedings of the Sixth Annual USENIX Security Conference, pp. 43-50, July 1996.

[11] S. Jarecki, H. Krawczyk, and J. Xu, "OPAQUE: an asymmetric PAKE protocol secure against pre-computation attacks," In Advances in Cryptology - EUROCRYPT 2018, pp. 456–486, Springer, 2018.

[12] A. Menezes, "Another Look At HMQV," *Journal of Mathematical Cryptology*, Vol. 1, No 1, pp. 47–64, 2007.

[13] D. Pointcheval and G. Wang, "VTBPEKE: Verifier-based Two-Basis Password Exponential Key Exchange," Proceedings of the 2017 ACM Symposium on Information, computer and communications security (AsiaCCS), 2017.

[14] M. Abdalla, F. Benhamouda, P. MacKenzie, "Security of the J-PAKE Password-Authenticated Key Exchange Protocol," Proceedings of IEEE Symposium on Security and Privacy, 2015.

[15] C. Herley, P.C. van Oorschot, "SoK: Science, Security, and the Elusive Goal of Security as a Scientific Pursuit," Proceedings of IEEE Symposium on Security and Privacy, 2017.