PAKE standardization

Current ongoing activities at IRTF / CFRG regarding the PAKE selection process

Outline of this talk

- Short history on PAKE / prior to CFRG PAKE selection process
  - Patent pitfalls and circumvention strategies (e.g. PACE, SRP)
  - Implementation pitfalls (DragonFly, DragonBlood)
- Activities preceding CFRG PAKE selection process (RFC8125 by Jörn-Marc Schmidt)
- CFRG PAKE selection process
  - Process
  - Remaining candidates in December 2019: SPAKE2 / CPace and OPAQUE/AuCPace
  - Pre-computation attack resistance
  - Quantum “annoying” PAKE vs. Quantum-Prepared PAKE

- Current work at Endress + Hauser:
  V-PAKE as one component of TFA for users, Centralized Offline Authentication (COA)

History of PAKE: Use hampered by patents

Earliest proposals:

- “Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks” (Steven M. Bellovin; Michael Merritt, 1992)
  U.S. Patents 5,241,599 and 5,440,635 (expired in August 2013)

- “Strong Password-Only Authenticated Key Exchange” (David Jablon, 1996)
  U.S. Patent 6,226,383 (Expired in March 2017)

- “Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellmann” (Victor Boyko; Patrick MacKenzie; Savar Patel, 2000)
  U.S. Patent 7,047,408
  (Expiration as assessed in August 2019: 2020-03-10)

~~History of~~ PAKE: Use hampered by patents

Earliest proposals:

- “Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks” (Steven M. Bellovin; Michael Merritt, 1992)
  U.S. Patents 5,241,599 and 5,440,635 (expired in August 2013)

- “Strong Password-Only Authenticated Key Exchange” (David Jablon, 1996)
  U.S. Patent 6,226,383 (Expired in March 2017)

- “Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellmann” (Victor Boyko; Patrick MacKenzie; Savar Patel, 2000)
  U.S. Patent 7,047,408
  (~~Expiration as assessed in August 2019: 2020-03-10.~~ November 2019: Adjusted expiration 2023-03-10)

Important sub-component in many PAKE protocols: Map2Point primitive

- The most efficient PAKE protocols operating on ECC require an operation that hashes a string onto a group element as pre-requisite.
- Easy task for variable-time algorithms, difficult to do in constant time.
- Important patents held by Gemalto (Inventors J.-S. Coron and T. Icart)
- Circumvention approaches result in significant increase of computational complexity (J-PAKE, PACE with „generic mapping“)

Consequences of Patent trouble: 1.) Circumvention approaches

Patent circumvention approaches. Examples:

- SRP (Thomas D. Wu, 1998)
  - Added logical complexity in the protocol. No formal proof.
  - Needs full field structure for arithmetics. Could not be implemented with ECC.
  - Situation again complicated by Patents held by Stanford University.
- J-PAKE (Feng Hao, Peter Ryan, 2010)
  - Elegant design from a cryptographic proof perspective.
  - Significant increase in computational complexity due to the ZKP steps
- PACE (Bender, Kügler, Fischlin, 2009)
  - Added logical complexity in the protocol
  - Significantly larger computational complexity
  - excellent Side-Channel features

Consequences of Patent trouble: 2.) Insecure designs

Recent example:

- Dragonfly (Author Dan Harkins) as used, e.g. in WPA3 standard
- Uses a vulnerable non-constant-time “Hunt and peck” Map2Point primitive

- Vulnerabilities predicted already very early in CFRG discussions
  (Scott Fluhrer, Trevor Perrin, Watson Ladd, Bodo Möller, Rene Struik and others)
  https://mailarchive.ietf.org/arch/msg/tls/M9Wrwd0iDEAk-PztgmrqIPEXvao
- Standardization body WiFi-Alliance did not consider CFRG recommendations

- 2019 “Dragonblood” attack by Mathy Vanhoef and Eyal Ronen on present WPA3 standard: ia.cr/2019/383

PAKE: Pre-selection process discussions on CFRG mailing list

- Personal assessment B. Haase:

  Often emotional, unreasonable and inadequate discussions by some participants, including important promoters of Dragonfly in 2013

- In CFRG the common understanding seems to have grown that a formalized and structured process is mandatory.

- First step: Setting up consensus on the requirements and objectives

- Important achievement: RFC8125 (Jörn-Marc Schmidt) (April 2017)

Achievements of RFC8125 (Jörn-Marc Schmidt, April 2017)

- PAKE Taxonomy
  - Augmented vs. Balanced PAKE
  - Two-Party vs. Multi-Party
- Definitions regarding the security model specification for PAKE
- Explicitly considers implementation aspects and (timing) side-channels

Augmented vs. Balanced PAKE

- “Balanced” or “Symmetric” PAKE
  - Both sides share the same low-entropy secret

- “Augmented” or “Asymmetric” or “Verifier-based” PAKE
  - Client party has access to clear-text password
  - Server party has access to Password-verifier

Requirements in RFC8125 (Jörn-Marc Schmidt, April 2017)

- REQ1: A PAKE scheme MUST clearly state its features regarding balanced/augmented versions.
- REQ2: A PAKE scheme SHOULD come with a security proof and clearly state its assumptions and models.
- REQ3: The authors SHOULD show how to protect their PAKE scheme implementation in hostile environments, particularly, how to implement their scheme in constant time to prevent timing attacks.
- REQ4: If 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.
- REQ5: The authors of a PAKE scheme MAY discuss its design choice with regard to performance, i.e., its optimization goals.

Requirements in RFC8125 (Jörn-Marc Schmidt, April 2017)

- REQ6: The authors of a scheme MAY discuss variations of their scheme that allow the use in special application scenarios. In particular, techniques that facilitate long-term (public) key agreement are encouraged.
- REQ7: Authors of a scheme MAY discuss special ideas and solutions on privacy protection of its users.

History of PAKE at CFRG since IETF 103 (Bangkok)

- Information given in this part of the talk largely obtained by “copy and paste” from official CFRG documentation from IETF 106 conference

  https://datatracker.ietf.org/meeting/106/materials/slides-106-cfrg-pake-selection-update

IETF 103, Bangkok (November 2018)

- After receiving several PAKE proposals and seeing documents complete, the chairs want to announce a PAKE selection process
- The aim is to select one or more (“zero or more”) PAKEs to recommend to the wider IETF community
- Submissions to satisfy RFC8125 (Jörn-Marc Schmidt)
- Both balanced and augmented PAKEs are considered
- Better to select one without a variety of options
- Involving Crypto Review Panel to come up with recommendations
- Support of the process at the CFRG session (“and please do it soon”) and later at the TLS and IPSECME sessions

Text derived from slides from Stanislav V. Smyshlyaev, CFRG Secretary
https://datatracker.ietf.org/meeting/106/materials/slides-106-cfrg-pake-selection-update

Organization of the CFRG PAKE selection process

- Stage 1 (01.06.2019-30.06.2019)
  - Call for candidate protocols
  - Discussion on the list for questions to be asked
- Stage 2 (01.07.2019 – 19.07.2019)
  - The designers of the protocols prepare papers with responses for
  - … all positions of RFC 8125
  - … all additional questions selected at Stage 1

Text derived from slides from Stanislav V. Smyshlyaev, CFRG Secretary
https://datatracker.ietf.org/meeting/106/materials/slides-106-cfrg-pake-selection-update

Organization of the CFRG PAKE selection process

After Stage 2 the CFRG had the candidates, the additional questions and the responses (RFC8125 and the collected questions) for all candidates

- Balanced
  - SPAKE2 (nominated by Watson Ladd and Ben Kaduk)
  - J-PAKE (nominated by Feng Hao)
  - SPEKE (nominated by Dan Harkins)
  - CPace (nominated by B. Haase)
- Augmented
  - OPAQUE (nominated by Hugo Krawczyk)
  - AuCPace (nominated by B. Haase)
  - VTBPEKE (nominated by Guilin Wang)
  - BSPAKE (nominated by Steve Thomas)

Text derived from slides from Stanislav V. Smyshlyaev, CFRG Secretary
https://datatracker.ietf.org/meeting/106/materials/slides-106-cfrg-pake-selection-update

Organization of the CFRG PAKE selection process

Stage 3, 01.08.2019-15.08.2019

- Call for reviewers for the enumerated questions
- Crypto Review Panel members start their security analysis

Stage 4, 16.08.2019-15.09.2019

- The reviewers who volunteered at Stage 3 prepare their analysis
- Crypto Review Panel members prepare their security reviews
- “After the end of Stage 4 the CFRG obtained 14 great reviews, deeply studying various aspects of PAKEs, all of them collected at https://github.com/cfrg/pake-selection”
  (CFRG Secretary, S. V. Smyshlyaev)

Text derived from slides from Stanislav V. Smyshlyaev, CFRG Secretary
https://datatracker.ietf.org/meeting/106/materials/slides-106-cfrg-pake-selection-update

Organization of the CFRG PAKE selection process

Stage 5, 16.09.2019-30.10.2019

- Crypto Review Panel members review all gathered materials and write overall reviews for all candidate PAKEs.
- At the end of Stage 5, CFRG obtained 4 overall reviews of the Crypto Review Panel members:
  - Björn Tackmann
  - Russ Housley
  - Yaron Sheffer
  - Stanislav Shmyshlyaev

  (collected at https://github.com/cfrg/pake-selection)

Text derived from slides from Stanislav V. Smyshlyaev, CFRG Secretary
https://datatracker.ietf.org/meeting/106/materials/slides-106-cfrg-pake-selection-update

Organization of the CFRG PAKE selection process

Stage 6, 01.11.2019-16.11.2019

- CFRG chairs discuss the reviews and make recommendations.

- Decision of the CFRG:
  "Since the opinions of the reviewers were not unanimous and since some new questions were raised during the final stages of the first round of the PAKE selection proess, we move to the Round 2 of the selection process."
- There are 4 candidates left for Round 2:
  - SPAKE2 (balanced) – nominated by Watson Ladd and Ben Kaduk
  - CPace (balanced) – nominated by Björn Haase
  - OPAQUE (augmented) – nominated by Hugo Krawczyk
  - AuCPace (augmented) – nominated by Björn Haase

Text derived from slides from Stanislav V. Smyshlyaev, CFRG Secretary
https://datatracker.ietf.org/meeting/106/materials/slides-106-cfrg-pake-selection-update

Next Section

- Remaining candidates after IETF 106

- 2 balanced protocols SPAKE2, CPace
- 2 augmented protocols OPAQUE, AuCPace

Properties of balanced PAKE protocols

Balanced PAKE

- Both sides share the same low-entropy secret “pw”
- Both sides establish an ephemeral session key based on “pw”
- Communication of two honest parties does not reveal any information on “pw”
- Information obtained in active attacks does not reveal any information on “pw”

=> Offline dictionary attacks fended off

After IETF 106: Remaining balanced candidates. 1.) SPAKE2

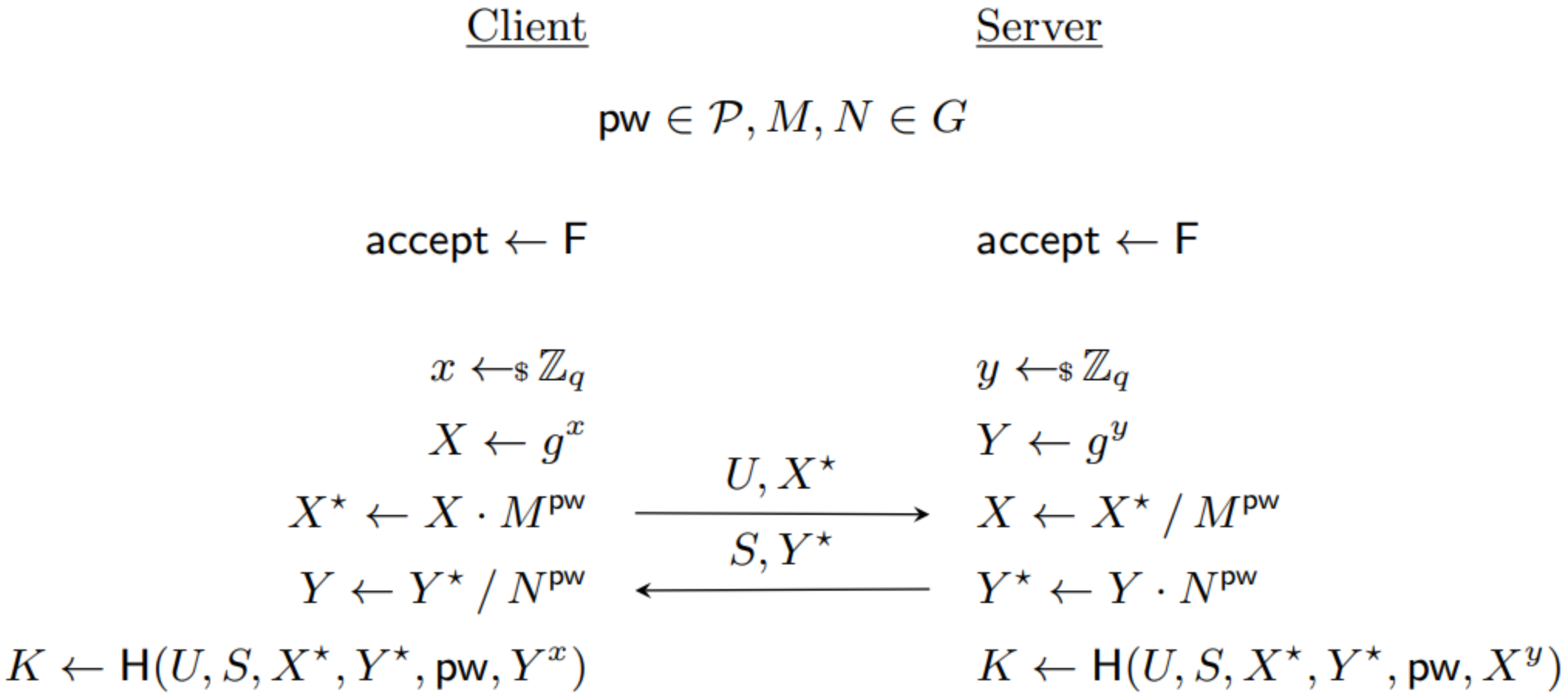


Image from taken from M. Abdalla and M. Barbosa https://eprint.iacr.org/2019/1194.pdf

After IETF 106: Remaining balanced candidates. 1.) SPAKE2

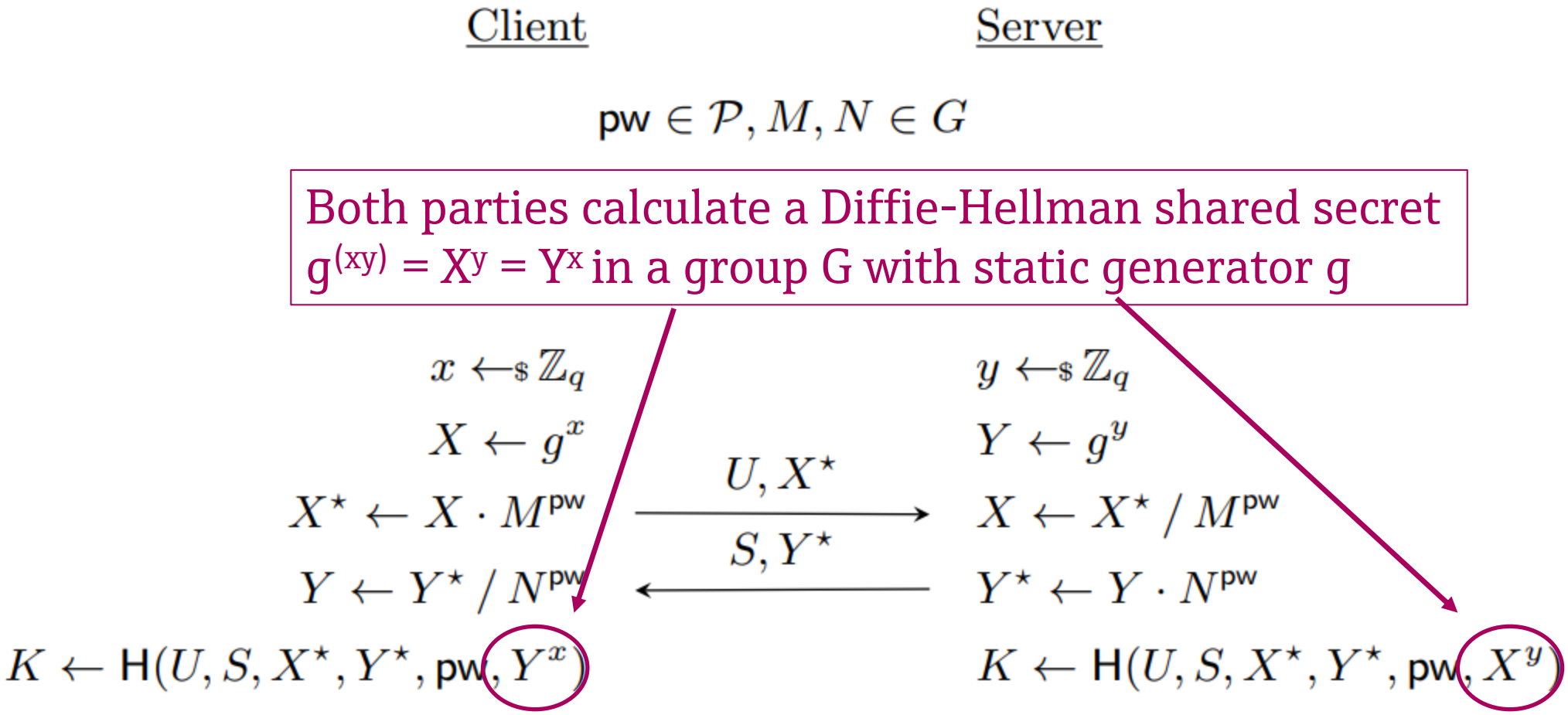


Image from taken from M. Abdalla and M. Barbosa https://eprint.iacr.org/2019/1194.pdf

After IETF 106: Remaining balanced candidates. 1.) SPAKE2

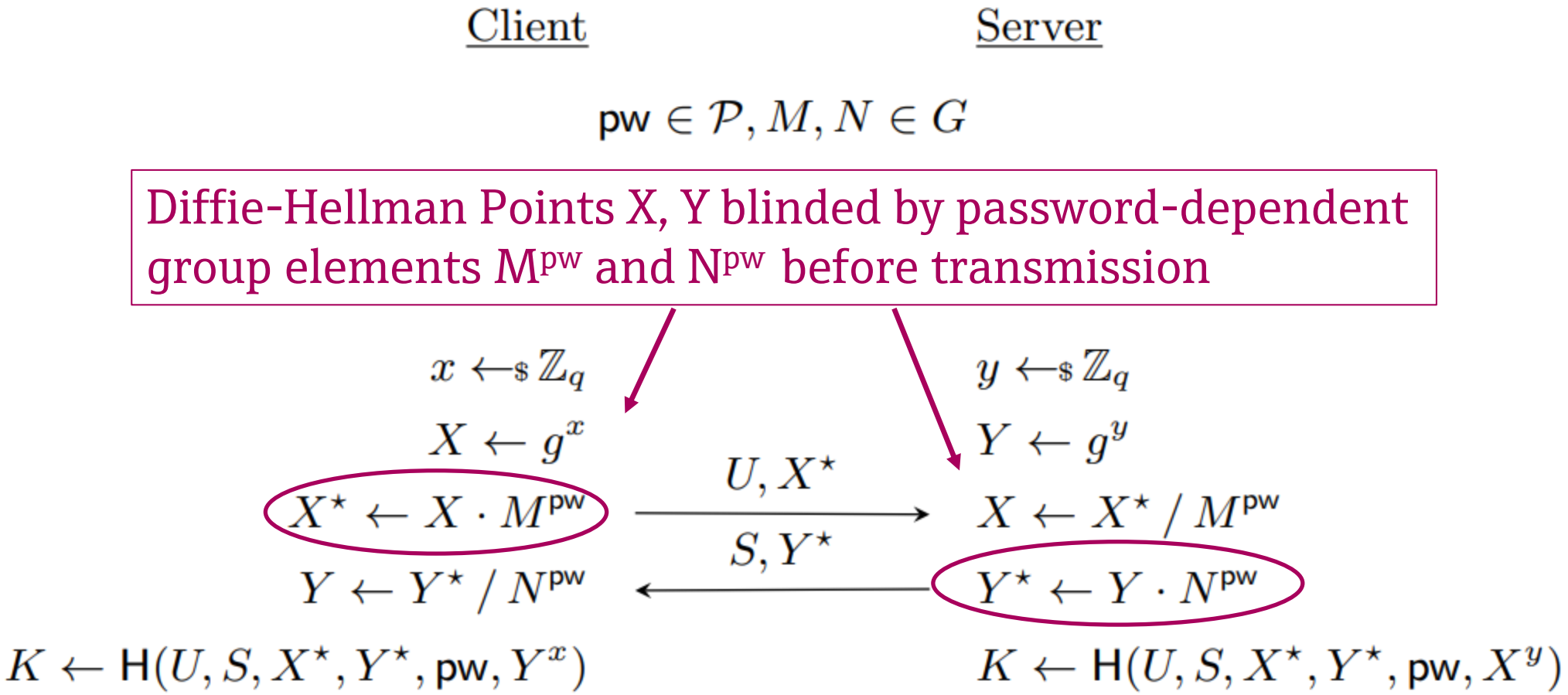


Image from taken from M. Abdalla and M. Barbosa https://eprint.iacr.org/2019/1194.pdf

After IETF 106: Remaining balanced candidates. 1.) SPAKE2

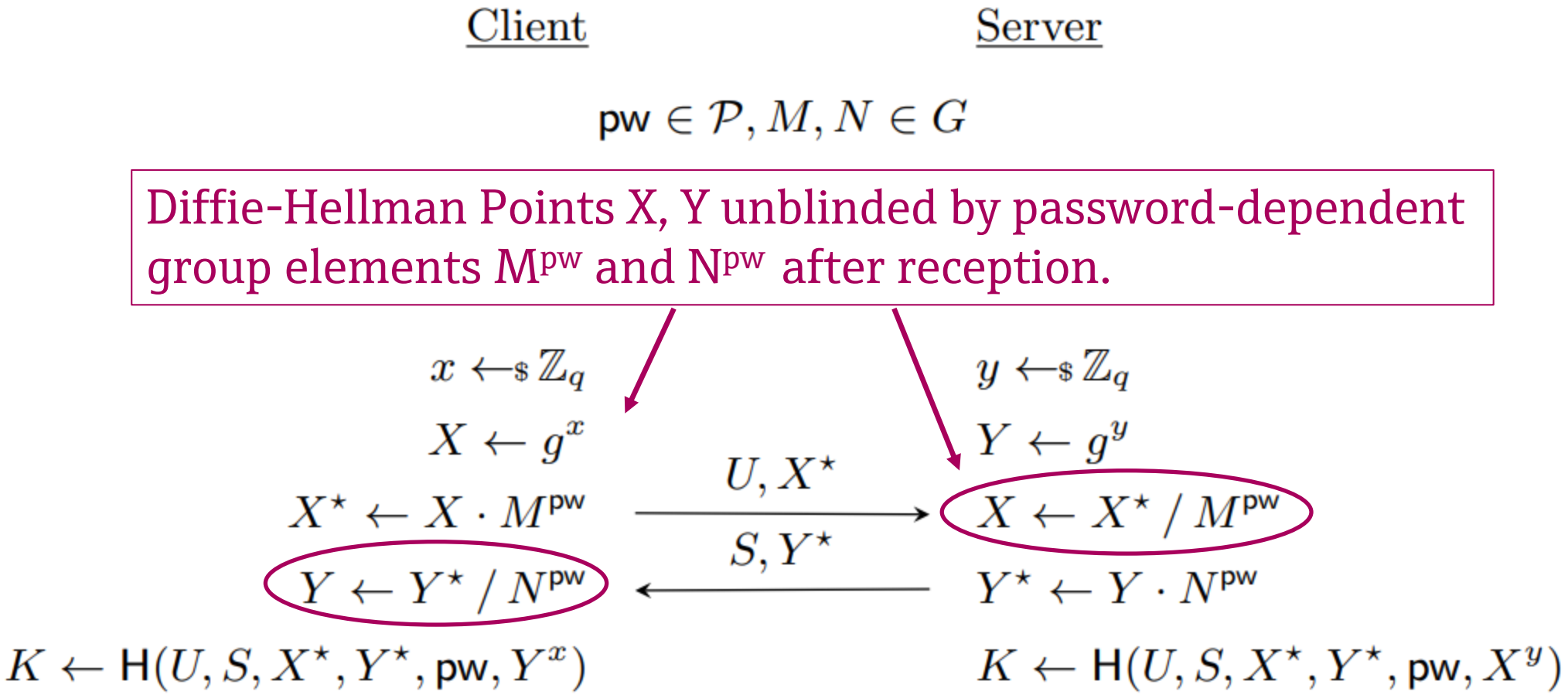


Image from taken from M. Abdalla and M. Barbosa https://eprint.iacr.org/2019/1194.pdf

After IETF 106: Remaining balanced candidates. 1.) SPAKE2

Advantages:

- No special primitive for Map2Point operation required.
- Asymmetric operation scalar multiplication is commonly available as constant-time operation in established cryptographic libraries.

Disadvantages:

- Requires trusted setup for parameters N, M. If discrete log relation between protocol constants M, N becomes known, protocol is vulnerable to offline attacks. => “Not Quantum-Annoying” (see following slides)
- Requires 4 scalar multiplications for each party for each protocol run.
- Proof considers only prime-order groups
- Assessment B. Haase: covered by patent US 7,047,408 until march 2023

After IETF 106: Remaining balanced candidates. 2.) CPace

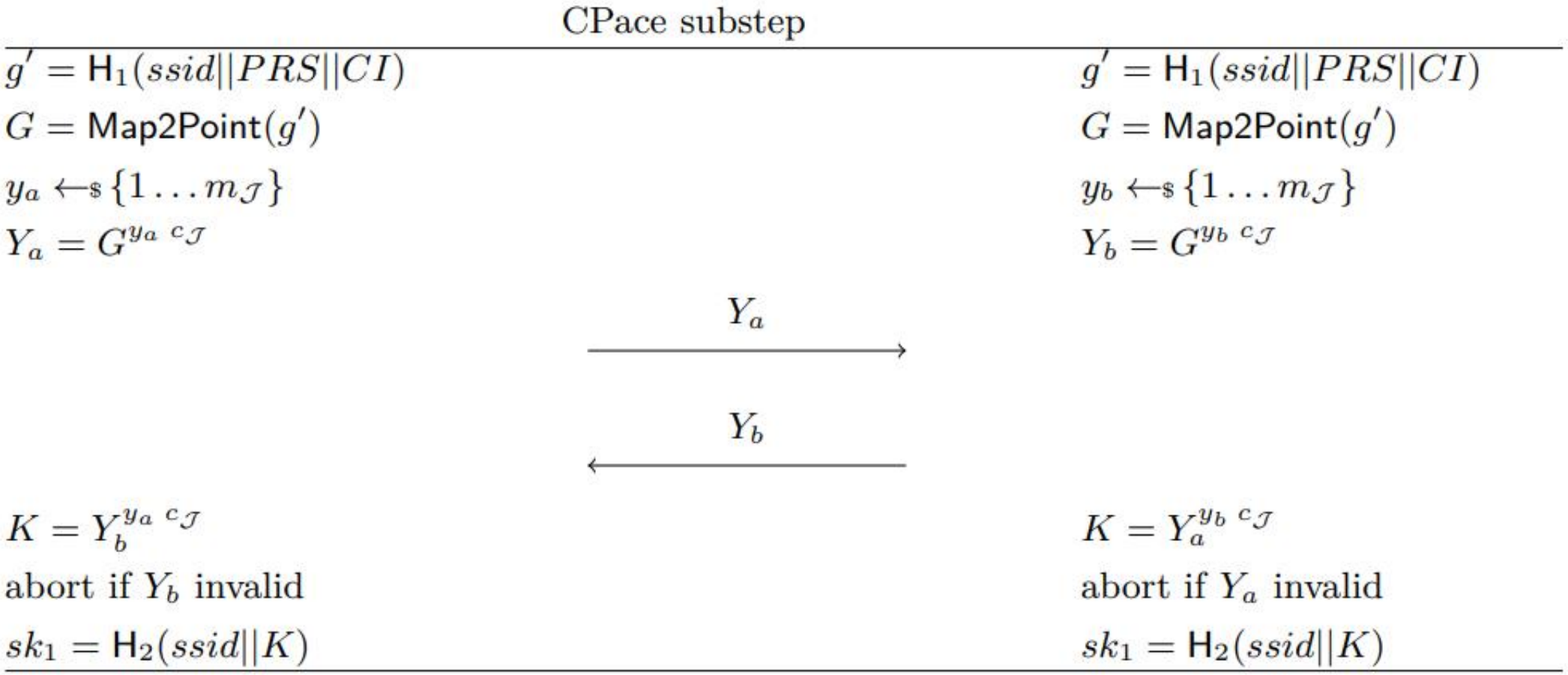
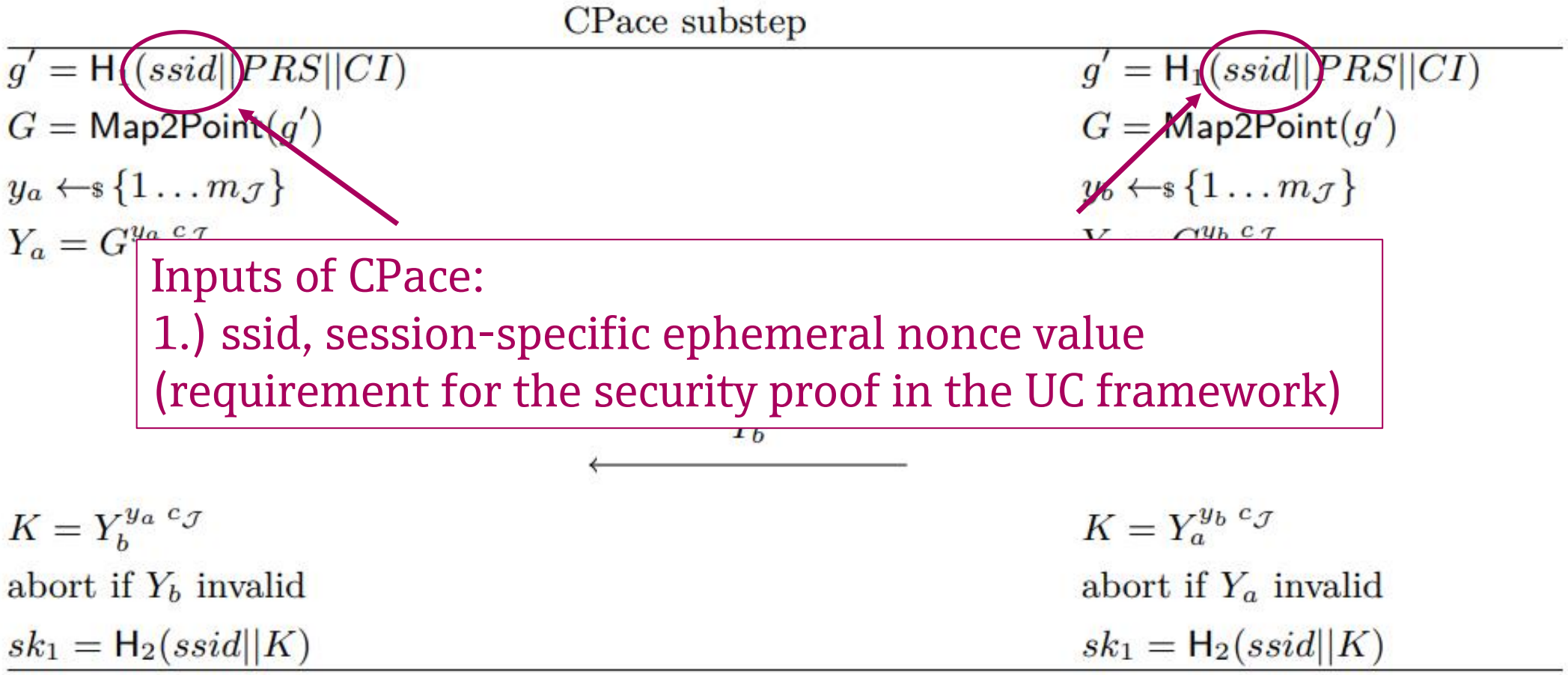


Image taken from B. Haase, B. Labrique https://eprint.iacr.org/2018/286.pdf

After IETF 106: Remaining balanced candidates. 2.) CPace



Inputs of CPace:
1.) ssid, session-specific ephemeral nonce value
(requirement for the security proof in the UC framework)

Image taken from B. Haase, B. Labrique https://eprint.iacr.org/2018/286.pdf

After IETF 106: Remaining balanced candidates. 2.) CPace

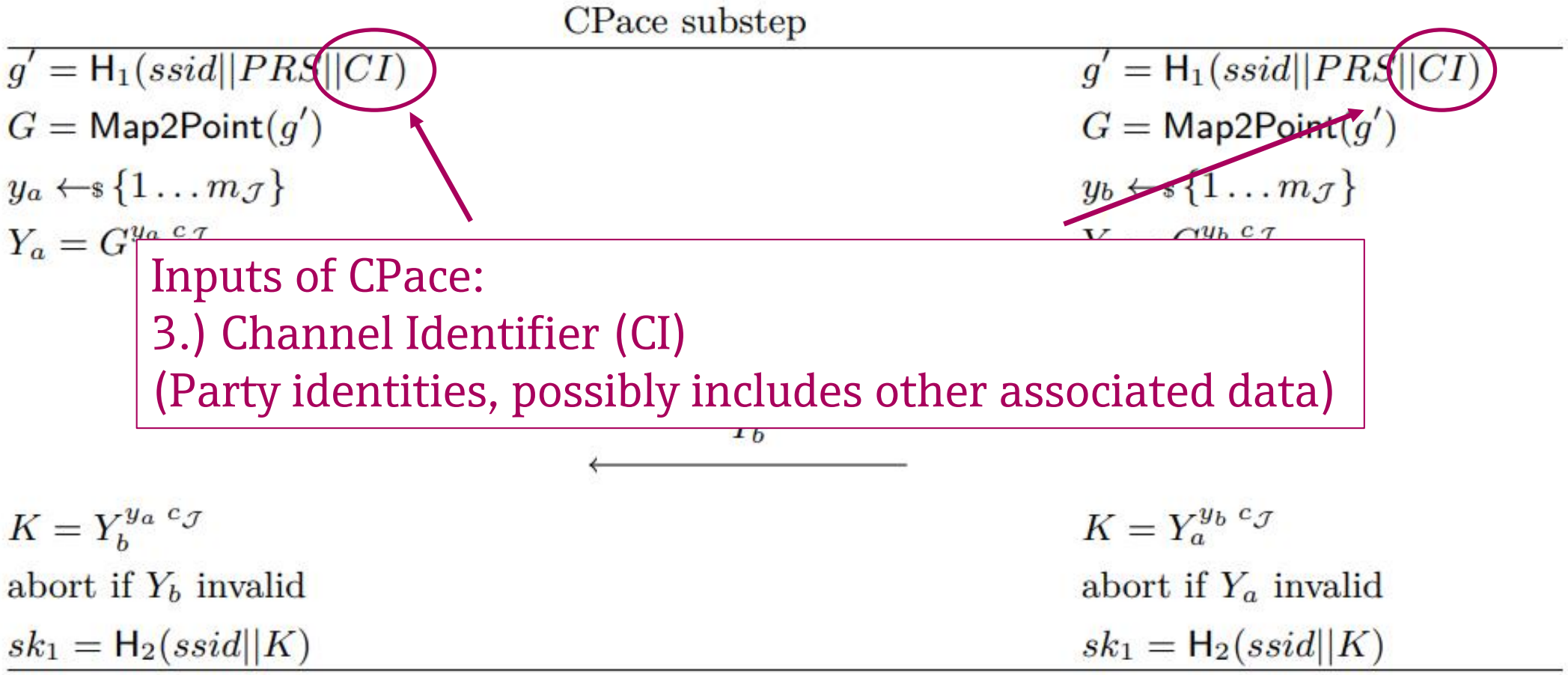
CPace substep

$g' = \mathsf{H}_1(ssid\|\|PRS\|\|CI)$	$g' = \mathsf{H}_1(ssid\|\|PRS\|\|CI)$
$G = \mathsf{Map2Point}(g')$	$G = \mathsf{Map2Point}(g')$
$y_a \leftarrow\$ \{1 \ldots m_{\mathcal{J}}\}$	$y_b \leftarrow\$ \{1 \ldots m_{\mathcal{J}}\}$
$Y_a = G^{y_a \, c_{\mathcal{J}}}$	$Y_b = G^{y_b \, c_{\mathcal{J}}}$
$K = Y_b^{y_a \, c_{\mathcal{J}}}$	$K = Y_a^{y_b \, c_{\mathcal{J}}}$
abort if Y_b invalid	abort if Y_a invalid
$sk_1 = \mathsf{H}_2(ssid\|\|K)$	$sk_1 = \mathsf{H}_2(ssid\|\|K)$

Inputs of CPace:
2.) Password-Related-String (PRS)
(e.g. calculated from memory-hard password-Hash and optional augmentation layer)

Image taken from B. Haase, B. Labrique https://eprint.iacr.org/2018/286.pdf

After IETF 106: Remaining balanced candidates. 2.) CPace



Inputs of CPace:
3.) Channel Identifier (CI)
(Party identities, possibly includes other associated data)

Image taken from B. Haase, B. Labrique https://eprint.iacr.org/2018/286.pdf

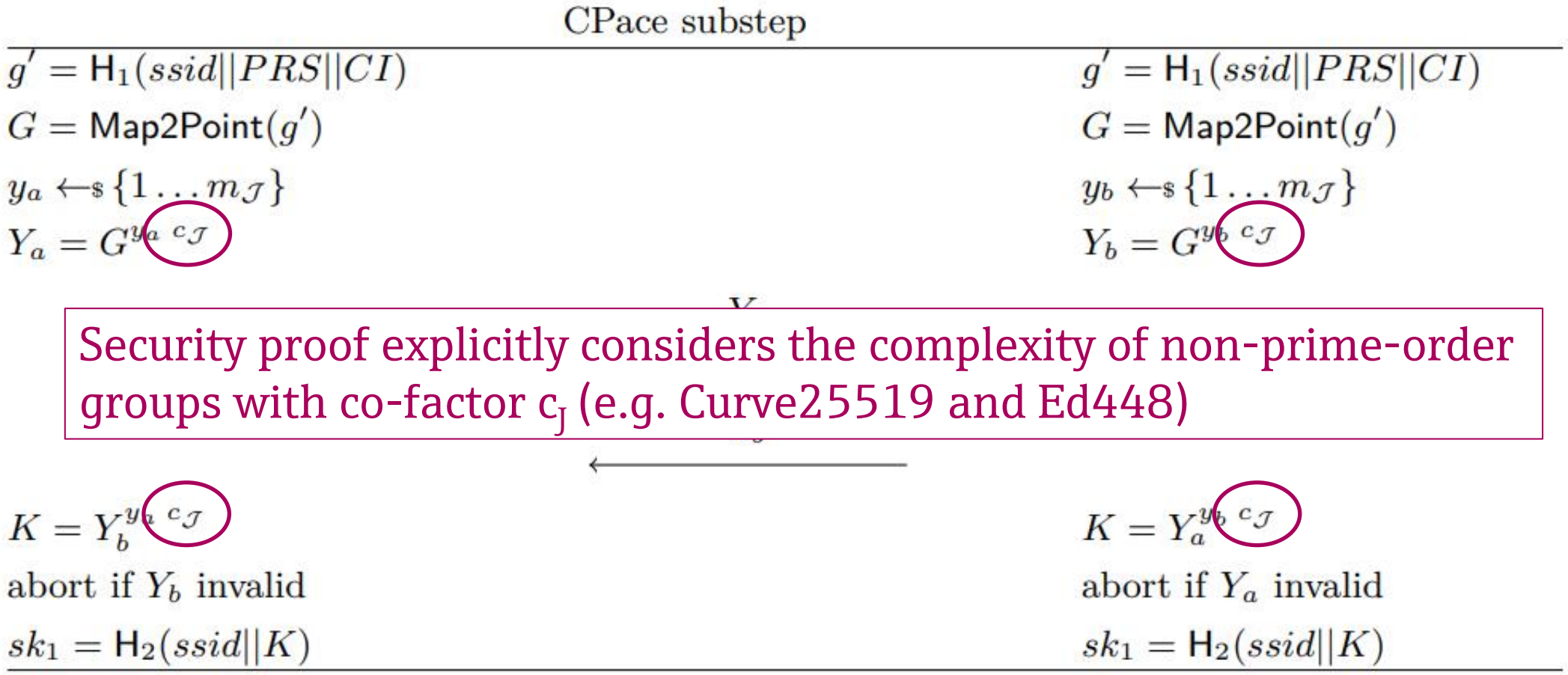
After IET

Both parties carry out a Diffie-Hellman key generation substep using a session-specific ephemeral generator G. Map2Point operation required

$g' = \mathsf{H}_1(ssid\|\|PRS\|\|CI)$	$g' = \mathsf{H}_1(ssid\|\|PRS\|\|CI)$
$G = \mathsf{Map2Point}(g')$	$G = \mathsf{Map2Point}(g')$
$y_a \leftarrow\$ \{1 \ldots m_{\mathcal{J}}\}$	$y_b \leftarrow\$ \{1 \ldots m_{\mathcal{J}}\}$
$Y_a = G^{y_a \, c_{\mathcal{J}}}$	$Y_b = G^{y_b \, c_{\mathcal{J}}}$
$\xrightarrow{\quad Y_a \quad}$	
$\xleftarrow{\quad Y_b \quad}$	
$K = Y_b^{y_a \, c_{\mathcal{J}}}$	$K = Y_a^{y_b \, c_{\mathcal{J}}}$
abort if Y_b invalid	abort if Y_a invalid
$sk_1 = \mathsf{H}_2(ssid\|\|K)$	$sk_1 = \mathsf{H}_2(ssid\|\|K)$

Image taken from B. Haase, B. Labrique https://eprint.iacr.org/2018/286.pdf

After IETF 106: Remaining balanced candidates. 2.) CPace



Security proof explicitly considers the complexity of non-prime-order groups with co-factor c_J (e.g. Curve25519 and Ed448)

Image taken from B. Haase, B. Labrique https://eprint.iacr.org/2018/286.pdf

After IETF 106: Remaining balanced candidates. 2.) CPace

Advantages:

- Requires only 2 scalar multiplications for each party for each protocol run.
- Requires no trusted setup.
- Ephemeral session id and generator make the protocol “Quantum annoying”
- Proof considers prime-order and non-prime-order groups
- Protocol design considers circumvention of all known patents.

Disadvantages:

- Special primitive for Map2Point operation required.
- Constant-time Map2Point might have to be added to some cryptographic libraries.

“Post quantum resistant” vs. “Quantum annoying” vs. “Post quantum ready”

- “Post quantum” protocol
  - Protocol or primitive based on a problem that is hard even in case that large-scale quantum-computing becomes practical.

- “Quantum annoying” protocol (definition suggested by Steve Tobtu)
  - “This property means that quantum computers need to solve a DLP for each password guess. For most […] PAKEs one would only need to solve 1 or 2 DLPs and then get classical computer offline password cracking”
    (Steve Tobtu, https://mailarchive.ietf.org/arch/msg/cfrg/dtf91cmavpzT47U3AVxrVGNB5UM)

- “Post quantum ready” protocol (term coined by H. Krawczyk)
  - Protocol does not provide any protection against quantum computers, but the modular structure with a proof in the UC security model might allow for replacing building blocks with future post-quantum building blocks, if such building blocks become available.

Properties of balanced PAKE protocols

Balanced PAKE

- Both sides share the same low-entropy secret “pw”
- Both sides establish an ephemeral session key based on “pw”
- Communication of two honest party does not reveal any information on “pw”
- Information obtained in active attacks does not reveal any information on “pw”

Properties of augmented PAKE protocols

Augmented PAKE

- Both sides share *different* secrets, “pw” and “PV(pw)”
  - Client has access to “pw”
  - Server has access to password-verifier “PV” = PV(pw)
- Both sides establish an ephemeral session key based on “pw” and “PV”
- Communication of two honest party does not reveal information on “pw” and “PV”
- Information obtained in active attacks does not reveal information on “pw”, “PV”
- When stealing the password database from the server, the adversary needs first to mount a dictionary search on PV(p’) in order to find the correct password pw

Augmented PAKE protocols and password hashing.

Security benefit of “Augmentation”, assessment B. Haase:

- In order to provide any meaningful additional protection in comparison to balanced PAKE, augmenting PAKE is only useful if calculating the password verifier PV(pw) involves memory-hard iterated hashing (scrypt, Argon2) with a random “salt” value

Memory-Hard password hashing designed for high computational complexity:

- Choose a “salt” value
- Calculate x = scrypt(salt, password)

- Since each password uses a different high-entropy value for “salt”, it is not practical to calculate “rainbow tables” for all password/salt combinations.

Pre-Computation attack resistance

Augmented PAKE

- "salt" value is publicly known for eavesdropper or active adversary
- The adversary may setup a pre-calculated "rainbow-table" with entries
  {PV(salt, pw1), PV(salt, pw2), PV(salt, pw3), ...}
  already *before* stealing the password verifier database from the server.
- Example: "AuCPace"

"Pre-computation attack resistant" augmented PAKE (Krawczyk, Jarecki, Wu, 2018)

- The "salt" value is kept secret
- An offline password test, can be mounted only *after* successfully stealing the password verifier database together with information on "salt" from the server.
- Examples: "OPAQUE", "*Strong* AuCPace"

Pre-Computation attack resistance

Augmented PAKE

- "sa
- Th
  {PV(salt, pw1), PV(salt, pw2), PV(salt, pw3), ...}
  already *before* stealing the password verifier database from the server.
- Example: "AuCPace"

Migration of a legacy-style password file to a conventional augmented PAKE is possible. (Cleartext password not required for migration).

"Pre-computation attack resistant" augmented PAKE (Krawczyk, Jarecki, Wu, 2018)

- The "salt" value is kept secret
- An c
  pass
- Examples: "OPAQUE", "*Strong* AuCPace"

Migration of a legacy-style password file to a pre-computation resistant PAKE not possible. (Cleartext passwords required for migration).

After IETF 106: Remaining augmented candidates. 1.) OPAQUE

OPAQUE: Two interleaved subcomponents

- Use “Olibvious Pseudo-Random Function” (OPRF) for keeping the salt value secret
- Authenticated KEM

- OPAQUE binds successful mutual authentication in KEM on knowledge of the “salt” value and the password hash.

After IETF 106: Remaining augmented candidates. 2.) AuCPace

AuCPace: Sequential scheme with three sub-components

- AuCPace “Augmentation layer”:
  Calculate an ephemeral password-related string (PRS) by use of El-Gamal scheme
- CPace: Derive an ephemeral high-entropy session key SK1 from PRS
- Mutual explicit key confirmation for SK1 and key refreshing

Pre-computation attack resistant “Strong AuCPace” PAKE

- uses OPAQUE’s OPRF scheme for keeping the “salt” value secret :

Conventional PAKE “AuCPace”:

- publicly transmits the “salt” value for the password hash when calculating PRS.

After IETF 106: Pre-Computation-Attack Resistance by Difie-Hellman OPRF

Essence of “pre-computation attack resistance”:
Keep the “salt” value secret for the adversary!

The following slides show the OPRF method of OPAQUE as re-used in *Strong* AuCPace.

Two steps

1. Password registration for OPRF fixes salt
2. Operations during password-based login sessions reconstruct “salt” for the client

Pre-computation attack resistance: How to keep the “salt” value secret

1. Password registration for OPRF

Server

Client

$\mathrm{q} \leftarrow_{\$} \{0,1\}^{l}$

$Z = \mathsf{Map2Point}(\mathsf{H}_1(\text{username} \,||\, pw))$

$\text{salt} = Z^{(\mathrm{q}\, c_{\mathcal{J}})}$

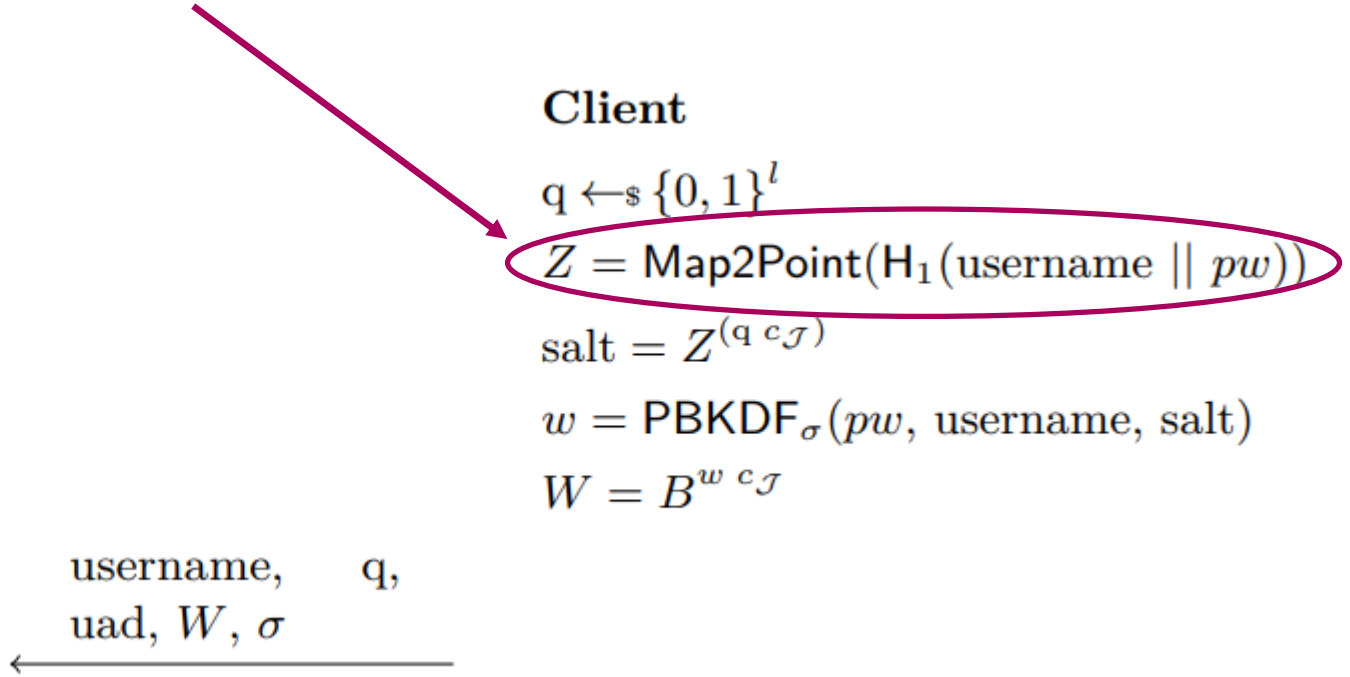
$w = \mathsf{PBKDF}_{\sigma}(pw, \text{username}, \text{salt})$

$W = B^{w\, c_{\mathcal{J}}}$

$\text{username}, \quad \mathrm{q},$
uad, W, σ ←

Pre-computation attack resistance: How to keep the “salt” value secret

A group element Z is calculated from the password „pw“ and the user name



After IETF 106: Remaining balanced candidates.

A secret exponent q is chosen during password registration

Server

Client

$\mathrm{q} \leftarrow_{\$} \{0,1\}^{l}$

$Z = \mathsf{Map2Point}(\mathsf{H}_1(\text{username} \mid\mid pw))$

$\text{salt} = Z^{(\mathrm{q}\, c_{\mathcal{J}})}$

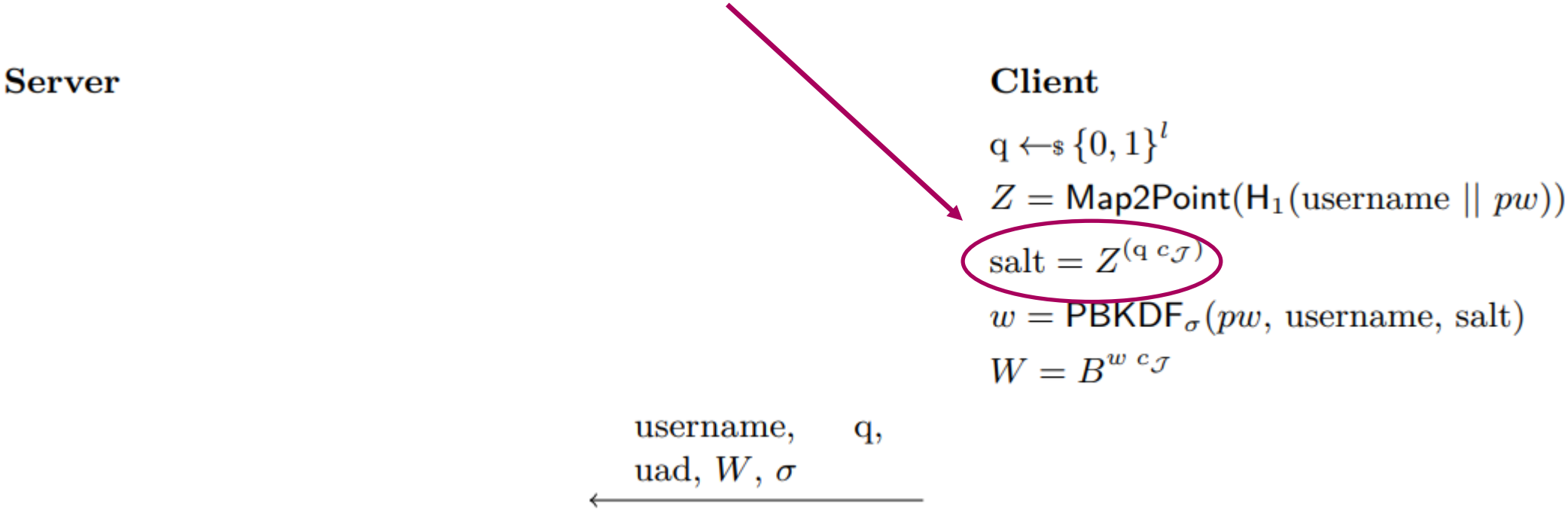
$w = \mathsf{PBKDF}_{\sigma}(pw, \text{username}, \text{salt})$

$W = B^{w\, c_{\mathcal{J}}}$

username, q,
uad, W, σ ←

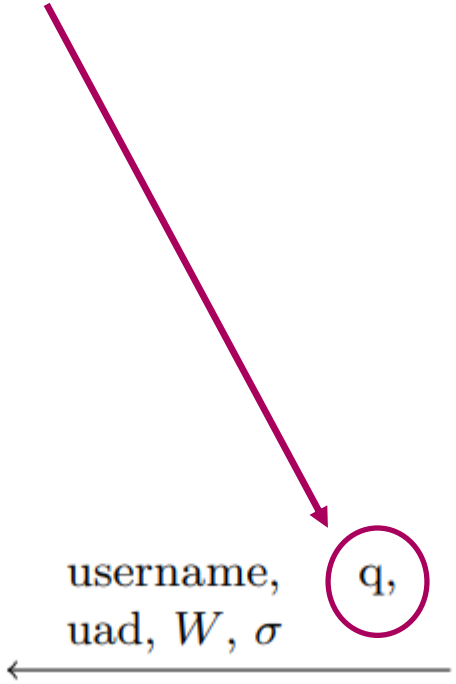
Pre-computation attack resistance: How to keep the “salt” value secret

The salt value is calculated as the result of a scalar multiplication of Z and q



Pre-computation attack resistance: How to keep the “salt” value secret

The server stores only q and not Z!



Pre-computation attack resistance: How to keep the “salt” value secret

Client calculates Z just as during password registration.

Server		**Client**
$x \leftarrow_\$ \{1 \dots m_{\mathcal{J}}\}$		$r \leftarrow_\$ \{1 \dots m_{\mathcal{J}}\}$
$X = B^{x\ c_{\mathcal{J}}}$		$Z = \mathsf{Map2Point}(\mathsf{H}_1(\text{username} \mid\mid pw)); V = Z^{r\ c_{\mathcal{J}}}$
	$\text{username}, U \longleftarrow$	
$W, \text{q} = \text{lookup}W(\text{user})$		
$UQ = U^{q\ c_{\mathcal{J}}}$		
abort if UQ invalid		
	$\mathcal{J}, X, UQ, \sigma \longrightarrow$	
		$\text{salt} = UQ^{\frac{1}{r\ (c_{\mathcal{J}})^2}\ c_{\mathcal{J}}}$

Pre-computation attack resistance: How to keep the “salt” value secret

Client blinds Z with an ephemeral value (* r).

Server		**Client**
$x \leftarrow_\$ \{1 \dots m_{\mathcal{J}}\}$		$r \leftarrow_\$ \{1 \dots m_{\mathcal{J}}\}$
$X = B^{x\ c_{\mathcal{J}}}$		$Z = \mathsf{Map2Point}(\mathsf{H}_1(\text{username} \mid\mid pw)); U = Z^{r\ c_{\mathcal{J}}}$
	$\text{username}, U \longleftarrow$	
$W, \text{q} = \text{lookup}W(\text{user})$		
$UQ = U^{q\ c_{\mathcal{J}}}$		
abort if UQ invalid		
	$\mathcal{J}, X, UQ, \sigma \longrightarrow$	
		$\text{salt} = UQ^{\frac{1}{r\ (c_{\mathcal{J}})^2}\ c_{\mathcal{J}}}$

Pre-computation attack resistance: How to keep the “salt” value secret

Server fetches secret exponent q from password registration from database.

Server		**Client**
$x \leftarrow_\$ \{1 \dots m_{\mathcal{J}}\}$		$r \leftarrow_\$ \{1 \dots m_{\mathcal{J}}\}$
$X = B^{x\ c_{\mathcal{J}}}$		$Z = \mathsf{Map2Point}(\mathsf{H}_1(\text{username} \mid\mid pw)); U = Z^{r\ c_{\mathcal{J}}}$
	$\text{username}, U \longleftarrow$	
$W,\mathrm{q} = \text{lookup}W(\text{user})$		
$UQ = U^{q\ c_{\mathcal{J}}}$		
abort if UQ invalid		
	$\mathcal{J}, X, UQ, \sigma \longrightarrow$	
		$\text{salt} = UQ^{\frac{1}{r\ (c_{\mathcal{J}})^2}\ c_{\mathcal{J}}}$

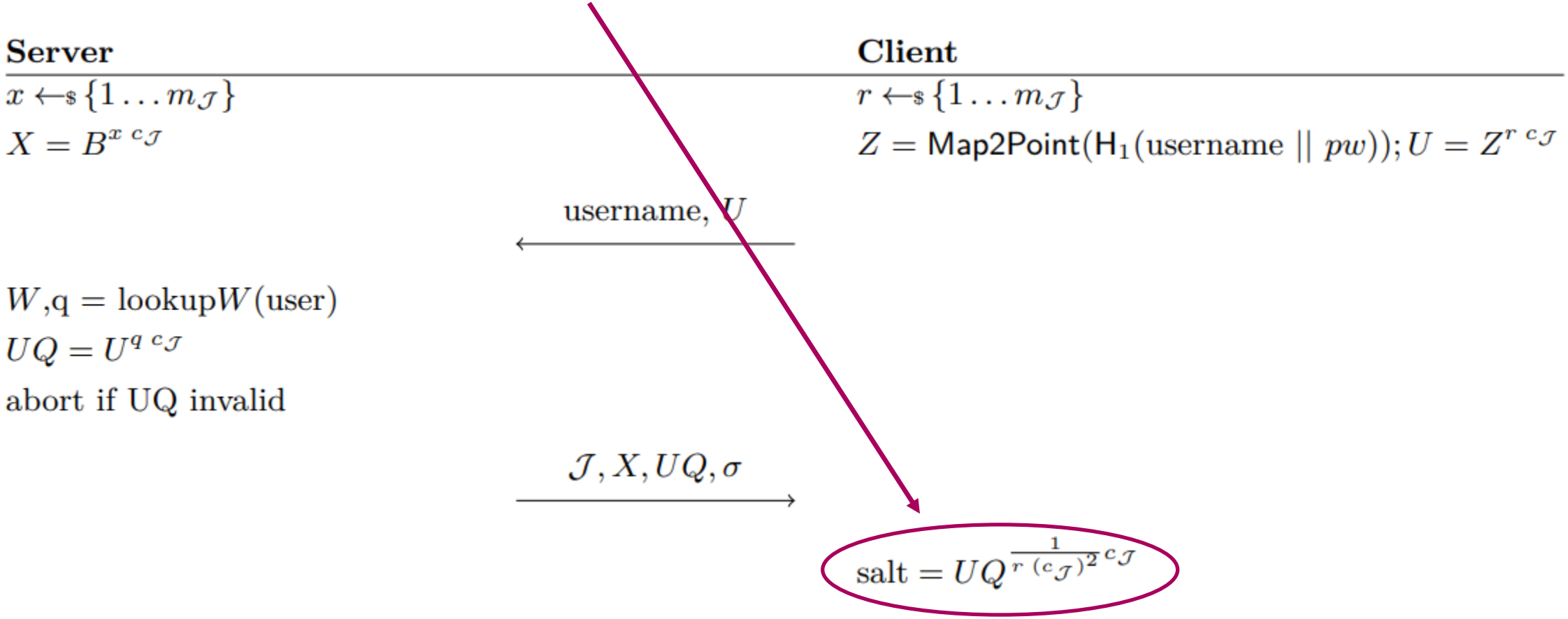
Pre-computation attack resistance: How to keep the “salt” value secret

Server calculates blinded point U as a power of q.

Server		**Client**
$x \leftarrow_\$ \{1 \dots m_{\mathcal{J}}\}$		$r \leftarrow_\$ \{1 \dots m_{\mathcal{J}}\}$
$X = B^{x\ c_{\mathcal{J}}}$		$Z = \mathsf{Map2Point}(\mathsf{H}_1(\text{username} \mid\mid pw)); U = Z^{r\ c_{\mathcal{J}}}$
	$\text{username}, U \longleftarrow$	
W,q = lookup$W$(user)		
$UQ = U^{q\ c_{\mathcal{J}}}$		
abort if UQ invalid		
	$\mathcal{J}, X, UQ, \sigma \longrightarrow$	
		$\text{salt} = UQ^{\frac{1}{r\ (c_{\mathcal{J}})^2}\ c_{\mathcal{J}}}$

Pre-computation attack resistance: How to keep the “salt” value secret

Client un-blinds the point *(1/r) in order to recover the salt value.



Pre-computation attack resistance: How to keep the “salt” value secret

Comparison: The OPAQUE approach is very similar but not identical.

Main difference in practice: In AuCPace if a (side-channel) attack on the server’s exponent q succeeds, AuCPace still provides conventional augmentation guarantees.
In contrast this operation is a single point of failure for OPAQUE.

$c_{\mathcal{J}}$

$W, \mathrm{q} = \mathrm{lookup}W(\mathrm{user})$

$UQ = U^{q\ c_{\mathcal{J}}}$

abort if UQ invalid

$$\xrightarrow{\mathcal{J}, X, UQ, \sigma}$$

$$\mathrm{salt} = UQ^{\frac{1}{r\ (c_{\mathcal{J}})^2}\ c_{\mathcal{J}}}$$

After IETF 106: Remaining augmented candidates. 1.) OPAQUE

Advantages:

- One message round less than AuCPace, fits better into TLS 1.3 handshake
- With HMQV as KEM faster than strong AuCPace

Disadvantages:

- Large password verifiers (~280 bytes)
- Augmentation relies on security of OPRF alone
  (CFRG discussions: “side-channel fragility” of OPAQUE OPRF)
- No protection against quantum computers
- Hugo Krawczyk argues that OPAQUE is “Post-Quantum prepared”
  (Currently no Post-Quantum OPRF function known).
- Most efficient version uses HMQV as KEM. (Patent by IBM, EP1847062B1).

After IETF 106: Remaining augmented candidates. 2.) AuCPace

Advantages:

- Very compact and efficient on constrained targets
- Pre-Computation attack resistance is *optional* feature
- UC Security proof of AuCPace considers adaptive adversaries
- Allows for x-coordinate-only Diffie-Hellman (e.g. X25519)
- Particularly small password verifiers (64 Byte)
- Protocol is “Quantum-Annoying”

Disadvantages:

- One more message round than OPAQUE
- Pre-computation attack resistant version is more complex than OPAQUE with HMQV

Security review summary of CFRG Crypto Review Panel members

Previous stages

TL;DR's of the Crypto Review Panel reviews

1. Bjoern Tackmann: „As balanced scheme, CPACE seems best, with SPAKE2 coming in somewhat close second. As augmented scheme, I think that OPAQUE should be considered for its possible seamless integration with TLS. As a general aPAKE, I have a slight preference for the strong AuCPace variant.“
2. Russ Housley: „OPAQUE“
3. Yaron Sheffer: „I think the Research Group should recommend one balanced and one augmented algorithm. [...] Of the balanced algorithms, I would recommend CPace. 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.“
4. Stanislav Smyshlyaev: „I would recommend selecting two PAKEs (one balanced and one augmented): SPAKE2 and OPAQUE. No strong objections against: CPace, AuCPace, VTBPEKE“

CFRG 9 / 18

https://datatracker.ietf.org/meeting/106/materials/slides-106-cfrg-pake-selection-update

Map2Point primitives

- Three of the four remaining CFRG candidates for PAKE require a Map2Point operation

- Input: String
- Output: Group element with unpredictable discrete logarithm

Map2Point primitives

- Elligator2
  - Available for Curve25519 and Ed448 (needs odd-characteristic and point of order two)
- Elligator
  - Not relevant for established curves.
- SWU
  - Gives three candidate x-coordinates. One candidate x-coordinate can be proven to be on the curve.
  - Difficult to implement in constant time. Many square-roots and constant-time CMOV pitfalls
- Simplified SWU
  - Guarantee that only two candidate x-Coordinates of SWU could be on the curve
  - Some variants of Simplified SWU covered by patent (Icart, Coron).
    2019: New Approach of Riad Wahby provides un-encumbered alternative for Brainpool and NIST curves.
- Icart's algorithm (for p = 2 mod 3)
  - could be used for P384, not applicable for all of the other NIST curves and Brainpool

PAKE selection process – personal impression of B. Haase

- Highly competent and constructive feedback from CFRG crypto review panel members and independent researchers.
- Discussions reach largely higher standards in comparison to the peer-review process for IACR journals.
- Both, AuCPace proof and OPAQUE proof needed major rework and were significantly improved

- Discussions sometimes still emotional but generally serious and fair

- Special thanks to Julia Hesse, Björn Tackmann and Stanislav Jarecki for their careful review!

PAKE selection process: Next steps at IETF

- December 5th 2019: Deadline for filing additional questions for stage 2 of the selection process
- Nominators are requested to prepare responses and documentation regarding the additional question set for Stage 2 of the process until February 10th 2020.
- Final decision is to be prepared for IETF 107 at Vancouver, March 21th 2020

PAKE selection process: Additional questions for round 2

- 1) (to SPAKE2):
  Can you propose a modification of SPAKE2 (preserving all existing good properties of PAKE2) with a correspondingly updated security proof, addressing the issue of a single discrete log relationship necessary for the security of all sessions (e.g., solution based on using M=hash2curve(A|B), N=hash2curve(B|A))?

- 2) (to CPace and AuCPace):
  Can you propose a modification of CPace and AuCPace (preserving all existing good properties of these PAKEs) with a correspondingly updated security proof (maybe, in some other security models), addressing the issue of requiring the establishment of a session identifier (sid) during each call of the protocol for the cost of one additional message?

PAKE selection process: Additional questions for round 2

- 3) (to all 4 remaining PAKEs) :
  Can the nominators/developers of the protocols please re-evaluate possible IPR conflicts between their candidates protocols and own and foreign patents? Specifically, can you discuss the impact of U.S. Patent 7,047,408 (expected expiration 10th of march 2023) on free use of SPAKE2 and the impact of EP1847062B1 (HMQV, expected expiration October 2026) on the free use of the RFC-drafts for OPAQUE?

- 4) (to all 4 remaining PAKEs) What can be said about the property of "quantum annoyance" (an attacker with a quantum computer needs to solve [one or more] DLP per password guess) of the PAKE?

- 5) (to all 4 remaining PAKEs) What can be said about "post-quantum preparedness" of the PAKE?

PAKE for critical industrial installations
Our Suggestion: Centralized Online and Offline authentication (COA)

Filling the gap regarding secure and practical human operator authentication for industrial control

Endress+Hauser
People for Process Automation

PAKE in equipment for critical infrastructure

- Threat analysis at Endress + Hauser for the “E+H BlueConnect” service app:
  Security on Human-Machine-Interfaces and remote-login / Emergency access interfaces should be considered at least as critical as Machine-Machine interfaces.

- Passwords are expected to remain a major authentication component on HMI
- Reliability / Availability of network infrastructure should mostly not be considered adequate for safety applications.
- DOS-attacks on network connections to centralized network-based authentication server infrastructure should be considered practical

- De-centralized storage of (emergency-)access credentials considered mandatory.
- Augmented PAKE in conjunction with memory-hard password hashing should be used for password authentication for such local “offline” device databases

PAKE in equipment for critical infrastructure

- For synchronizing local/de-centralized “offline” user credential database, a centralized user-credential server infrastructure is considered mandatory.
- Systems need to be inter-operable among different suppliers. Our activity in the CFRG standardization aims also at paving the way for such standardization for control equipment.
- Important need: Consistent, secure and inter-operable format for password-verifiers and authentication protocols.
- Should be both, based on client-side memory-hard hashing and prepared for server-side secure element.

- Current development project at Endress + Hauser:
  “Centralized Online and Offline Authentication” (COA)
  - Field-device login interface secured by AuCPace
  - COA-Proxy to off-the-shelf user-administration tools (LDAP, ActiveDirectory) for distributing password verifiers to de-centralized field device databases.

PAKE in equipment for critical infrastructure

- Two-Factor authentication should be considered more adequate in many settings

- Our Proposal:
  1) Use AuCPace based username/password authentication
     COA: User based Authentication (UBA)
  2) Additionally require hardware tokens / badges
     COA: Device based authentication (DBA)

- COA also aims at paving the way for centralized “online” management infrastructure for managing both, UBA and DBA information.

Summary

- IRTF CFRG has setup a process for selection of PAKE protocols with 2+2 remaining candidates
  Decision scheduled for IETF 107 (March 2020)
- Advantages of our nominations CPace and AuCPace
  - Resilience with respect to quantum computers (“quantum annoying” property)
  - Optimized for constrained targets
  - To our best knowledge not covered by patents
- Properties of SPAKE2
  - Less efficient than CPace; Does not require Map2Point
  - Assessment B. Haase: Covered by US patent 7,047,408 until march 2023
  - No resilience with respect to quantum computers
- Properties of OPAQUE
  - Message flows compatible with TLS1.3 handshake; Large password verifiers.
  - No resilience with respect to quantum computers

Thank you very much for your attention

More details on our proposals CPace and AuCPace

https://eprint.iacr.org/2018/286
https://datatracker.ietf.org/doc/draft-haase-cpace/
https://datatracker.ietf.org/doc/draft-haase-aucpace/