SoK: Password-Authenticated Key Exchange -- Theory, Practice, Standardization and Real-World Lessons

Feng Hao    Paul C. van Oorschot
Motivation for PAKE (1992, Bellovin and Merrit)

- Create a high-entropy session key based on a low-entropy password without PKI
- Not considered possible until 1992 (16 years after 1976 Diffie-Hellman protocol)
Landscape view of PAKE

- 1992 - 2000: Explosive research on PAKE
- 2000 - 2008: IEEE P1363.2 standardization
- 2008 - 2018: ISO/IEC standardization
- 2018 - Present: IETF PAKE standardization

- Many arguments on use cases of PAKE in the past
- Today, PAKE has been widely deployed, e.g., iCloud, e-passports, WPA3, Thread IoT, BBM etc
- Wi-fi, e-passports, IoT were ahead of time in 1992!

Take-away 1: uses cases of new protocols may emerge and evolve over time
1. Ideal cipher
- EKE (1992)
- EKE2 (2000)
- OEKE (2003)
- KHAPE (2021)

2. Hash-to-group
- SPEKE (1996) + ★
- B-SPEKE (1997)
- PAK (2000) +
- SAE (2008) + ★
- P-SPEKE (2014)
- OPAQUE (2018)
- CPace (2019) Selected by IETF in 2020
- AuCPace (2019)

3. Trusted setup
- SPAKE2 (2005)
- KOY (2001)
- Kobara-Imai (2002)
- SESPAKE (2017)
- TBPEKE (2017)
- VTBPEKE (2017)
- KC-SPAKE2+ (2020)

4. ZKP
- J-PAKE (2008) + ★

5. Password as exponent
- SRP-3 (1998)
- AMP (2001)
- SRP-6 (2002) + ★
- Revised AMP (2005) +
- SRP-6a (2009) ★
- AugPAKE (2010) +

★ Included in standards
+ Used in real-world apps
Class 1: EKE (Bellovin, Merritt, IEEE S&P’92)

<table>
<thead>
<tr>
<th>Alice (A)</th>
<th>Bob (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \in_R [0, p - 1]$</td>
<td>$A, E_w(g^x \mod p)$</td>
</tr>
<tr>
<td></td>
<td>$B, E_w(g^y \mod p)$</td>
</tr>
<tr>
<td>Compute $K$</td>
<td>$y \in_R [0, p - 1]$</td>
</tr>
</tbody>
</table>

- Use password (w) to encrypt Diffie-Hellman items
- But $E_w(g^x), E_w(g^y)$ may decrypt to a value $> p$, hence leaks info (Jaspan, USENIX Security’96)
Provable security of EKE

● “We prove (in an ideal-cipher model) that the two-flow protocol at the core of EKE is a secure AKE.” (Bellare, Pointcheval, Rogaway, Eurocrypt’00)

● But how does this result reconcile with the information leakage problem pointed out by Jaspon in 1996?
The assumption of an ideal cipher

- By definition, an ideal doesn’t leak content even when a low-entropy key is used, but no explicit ideal ciphers were specified.
- Several constructions of an ideal cipher were proposed (Bellare-Rogaway, submission to IEEE P1362.2 in 2000)
- But none of the proposed constructions was secure (Zhao et al, TCS’06)
- EKE not included into IEEE P1363.2 (2000-2008)

Take-away 2: a PAKE protocol should be completely specified.
Class 2: SPEKE (Jablon, 1996)

<table>
<thead>
<tr>
<th>Alice (A)</th>
<th>Bob (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \in_R \mathbb{Z}_q$</td>
<td>$A, f(w)^x \mod p$</td>
</tr>
<tr>
<td>Validate key</td>
<td>Validate key</td>
</tr>
<tr>
<td>$K = H(f(w)^{xy})$</td>
<td>$y \in_R \mathbb{Z}_q$</td>
</tr>
<tr>
<td>$B, f(w)^y \mod p$</td>
<td>$K = H(f(w)^{xy})$</td>
</tr>
</tbody>
</table>

- $p = 2q + 1$ is a safe prime; $w$ denotes the password
- $f(w)$: a **hash-to-group** function that maps a password $w$ to a generator
- Only two exponents - looks optimally efficient (compare with plain DH)
- However, be careful when something sounds too good to be true
Hash-to-group function in SPEKE

- In MODP: \( f(w) = H(w)^2 \mod p \) where \( p=2q+1 \) is a safe prime
- However, for 3072-bit \( p \), the exponent \( x \) on \( f(w)^x \) is 3071-bit
- 12 times more costly than an exponentiation in 3072-DSA (256-bit exp)

- In the EC: \( f(w) \) is called hash-to-curve
- However, a complex problem on its own
- Hash-to-curve in IEEE 1363.2 not constant time
- IETF is working on a hash-to-curve internet draft (2018-present)
Class 3: SPAKE2 (Abdalla, Pointcheval, RSA’05)

<table>
<thead>
<tr>
<th>Alice (A)</th>
<th>Bob (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x \in_R Z_q)</td>
<td>Validate key</td>
</tr>
<tr>
<td>Validate key</td>
<td></td>
</tr>
<tr>
<td>(K = H(A, B, g^x, g^y, w, g^{xy}))</td>
<td>(K = H(A, B, g^x, g^y, w, g^{xy}))</td>
</tr>
</tbody>
</table>

- \(\{g, M, N\}\) is a **trusted setup**
  - Knowing the DL relation between the generators forever breaks the system
  - Same issue as Dual-EC random number generator
- **Cyclic motivation/assumptions for trusted setup**
  - Remove random oracle (RO) \(\rightarrow\) common reference string (CRS) \(\rightarrow\) RO + CRS

---

Take-away 3: assumptions in a security model need to match reality
A dilemma

Researchers often had to make a difficult choice between the two

<table>
<thead>
<tr>
<th>Trusted setup (e.g., SPAKE2)</th>
<th>Hash-to-group/curve (e.g., SPEKE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Breaking one DL instance forever breaks all sessions</td>
<td>● Well-defined but costly operation in MODP</td>
</tr>
<tr>
<td></td>
<td>● Yet uninstantiated in EC</td>
</tr>
</tbody>
</table>

Take-away 4: PAKE protocols are rarely directly comparable
Class 4: J-PAKE (Hao, Ryan, SPW’08)

<table>
<thead>
<tr>
<th>Alice (A)</th>
<th>Bob (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1, x_2 \in_R \mathbb{Z}_q$</td>
<td>$A, g^{x_1}, g^{x_2}, \text{ZKP}{x_1, x_2}$</td>
</tr>
<tr>
<td>Validate ZKPs</td>
<td>$B, g^{y_1}, g^{y_2}, \text{ZKP}{x_3, x_4}$</td>
</tr>
<tr>
<td>Validate ZKP</td>
<td>$\leftarrow B, \beta y_2 \cdot w, \text{ZKP}{y_2 \cdot w}$</td>
</tr>
<tr>
<td>$K = H \left( g^{(x_1+x_3) \cdot x_2 \cdot x_4 \cdot w} \right)$</td>
<td>$\leftarrow A, \alpha^{x_2 \cdot w}, \text{ZKP}{x_2 \cdot w}$</td>
</tr>
</tbody>
</table>

- Use **Schnorr zero-knowledge proof** to enforce honest behavior
- Comparable efficiency to SPEKE in MODP (because of short exponents)
- Require only primitive operations: mul/exp in MODP (or add/mul in EC), hence flexible to implement in MODP or elliptic curve
Class 5: SRP (Wu, 1998 - 2009)

- SRP-6a after several revisions [http://srp.stanford.edu/design.html](http://srp.stanford.edu/design.html)
- Costly exponentiation in MODP due to mandatory use of a safe-prime modulus
- Also, no EC version of SRP-6a (distinct protocol SRP-5 supports EC but not MODP)
A note on standardization

- **IEEE P1363.2 (2000-2008)**
  - No clear winner
  - All the selected protocols have subtle security flaws
  - New flaws continued to be found after 2008
  - 2019, IEEE 1363.2 officially withdrawn

- **ISO/IEC 11770-4 (active)**
  - Include new schemes and patch existing schemes through revisions

- **IETF (2019-2020)**
  - Two protocols selected: CPace, OPAQUE
  - But specs were incomplete when they were selected
  - Both protocols were modified after the IETF selection (not yet finalized …)

**Take-away 5:** PAKE standardization should not be a one-off process; it needs to be regularly revisited.
1. Ideal cipher

- EKE (1992)
- EKE2 (2000)
- OEKE (2003)
- KHAPE (2021)

2. Hash-to-group

- SPEKE (1996)
- B-SPEKE (1997)
- PAK (2000)
- SAE (2008)
- P-SPEKE (2014)
- OPAQUE (2018)
- CPace (2019)
- AuCPace (2019)

3. Trusted setup

- SPAKE2 (2005)
- KHAPE (2003)
- SEPAKE (2017)
- TBPEKE (2017)
- VTBPEKE (2017)
- KC-SPAKE2+ (2020)

4. ZKP

- J-PAKE (2008)

5. Password as exponent

- SRP-3 (1998)
- AMP (2001)
- SRP-6 (2002)
- Revised AMP (2005)
- SRP-6a (2009)
- AugPAKE (2010)

Many PAKEs in Class 1 and 3 are provably secure, but they are least used in practice.

Selected by IETF in 2020

- OPAQUE
- CPace

+ Included in standards
★ Used in real-world apps