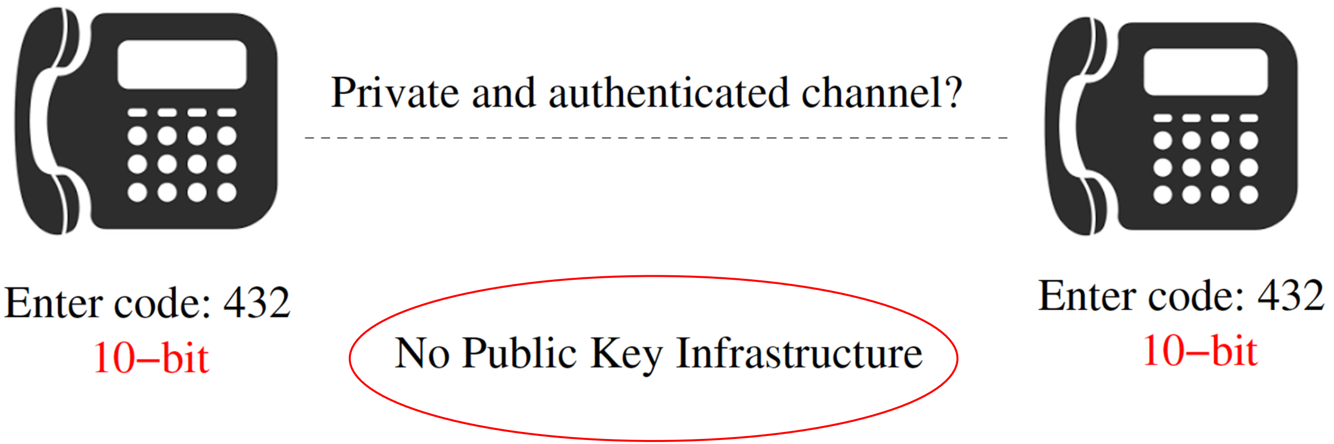
AsiaCCS'22

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)



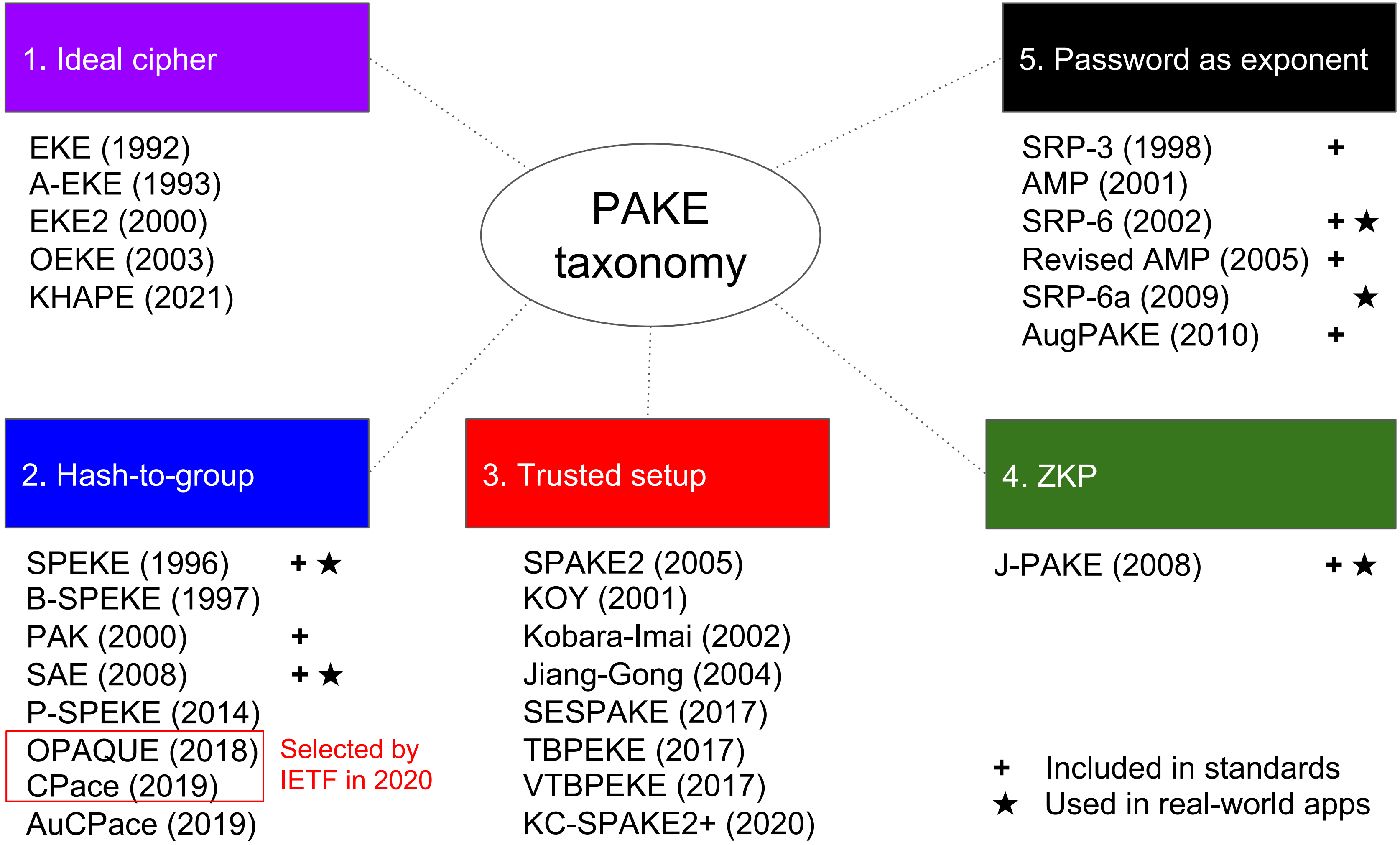
- Create a high-entropy session key based on a low-entropy password without PKI
- Not considered possible until 1992 (16 years after 1976 Diffie-Hellman protocol)

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



PAKE taxonomy
1. Ideal cipher
EKE (1992)
A-EKE (1993)
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)
Jiang-Gong (2004)
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)

Alice (A)		Bob (B)
$x \in_R [0, p-1]$	$A, \mathcal{E}_w(g^x \bmod p)$ $\longrightarrow$	
	$\longleftarrow$ $B, \mathcal{E}_w(g^y \bmod p)$	$y \in_R [0, p-1]$
Compute K		Compute K

- 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, Eucrocrypt’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)

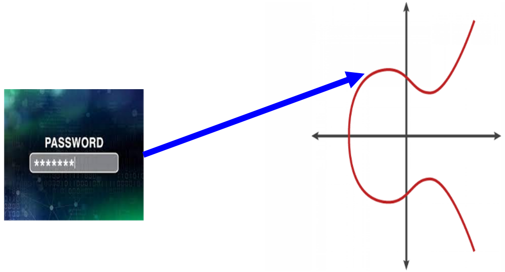
Alice (A)		Bob (B)
$x \in_R Z_q$	$A, f(w)^x \bmod p$ $\longrightarrow$	Validate key
Validate key	$B, f(w)^y \bmod p$ $\longleftarrow$	$y \in_R Z_q$
$K = H\left(f(w)^{xy}\right)$		$K = H\left(f(w)^{xy}\right)$

- 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 exps - 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: $f(w) = H(w)^2 \bmod p$ 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)

Alice (A)		Bob (B)
$x \in_R Z_q$	$\xrightarrow{A,\, g^x M^w \bmod p}$	Validate key
Validate key	$\xleftarrow{B,\, g^y N^w \bmod p}$	$y \in_R Z_q$
$K = H(A, B, g^x, g^y, w, g^{xy})$		$K = H(A, B, g^x, g^y, w, g^{xy})$

- {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) → common reference string (CRS) → 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

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

Take-away 4: PAKE protocols are rarely directly comparable

Class 4: J-PAKE (Hao, Ryan, SPW'08)

Alice (A)		Bob (B)
$x_1, x_2 \in_R Z_q$	$\xrightarrow{A, g^{x_1}, g^{x_2}, \text{ZKP}\{x_1, x_2\}}$	Validate ZKPs
Validate ZKPs	$\xleftarrow{B, g^{y_1}, g^{y_2}, \text{ZKP}\{x_3, x_4\}}$	$y \in_R Z_q$
Validate ZKP	$\xleftarrow{B, \beta^{y_2 \cdot w}, \text{ZKP}\{y_2 \cdot w\}}$	
	$\xrightarrow{A, \alpha^{x_2 \cdot w}, \text{ZKP}\{x_2 \cdot w\}}$	Validate ZKP
$K = H\left(g^{(x_1+x_3)\cdot x_2 \cdot x_4 \cdot w}\right)$		$K = H\left(g^{(x_1+x_3)x_2 x_4 \cdot w}\right)$

- 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)

Client (C)		Server
$a \in_R [2, p-1], A = g^a$	$\xrightarrow{C, A}$	Look up s, v
		$b \in_R [2, p-1]$
$x = H(s, w), u = H(A, B)$	$\xleftarrow{s, B}$	$B = k \cdot v + g^b$
$S = (B - k \cdot g^x)^{a+u \cdot x}$		$u = H(A, B)$
$K = H(S)$		$S = (Av^u)^b$
$M_1 = H\Big(H(p) \oplus H(g),$		$K = H(S)$
$H(C), s, A, B, K\Big)$	$\xrightarrow{M_1}$	Check M_1
Check M_2	$\xleftarrow{M_2}$	$M_2 = H(A, M_1, K)$

- SRP-6a after several revisions 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.

PAKE taxonomy

1. Ideal cipher

EKE (1992)
A-EKE (1993)
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)

Selected by IETF in 2020 (OPAQUE, CPace)

3. Trusted setup

SPAKE2 (2005)
KOY (2001)
Kobara-Imai (2002)
Jiang-Gong (2004)
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) +

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

CPace and OPAQUE in Class 2

\+ Included in standards
★ Used in real-world apps