

Password Authenticated Key Exchange: From Two Party Methods to Group Schemes

Stephen Melczer, Taras Mychaskiw, and Yi Zhang

Based on: The Fairy-Ring Dance – Password Authenticated Key
Exchange in a Group by Hao, Yi, Chen, and Shahandashti



Introduction

1. Classical Two Party PAKEs
 - 1.1 Background and Security Properties
 - 1.2 J-PAKE
 - 1.3 Dragonfly and PPK
2. Extension to Group Setting (GPAKEs)
 - 2.1 Fairy-Ring Dance
 - 2.2 J-PAKE+
3. Timings
4. Conclusion

PART 1

Classical Two Party PAKEs

Password Authenticated Key Exchange (PAKE)

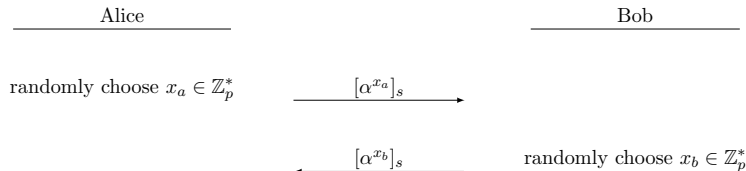
PAKEs allow two parties sharing a *short/weak* password to establish a shared key

Cannot broadcast password directly – would need to be protected (expensive)

Instead, modern PAKEs use *zero-knowledge proof*, *exponentiation*, or *hash* of expression w/ password

First Protocol: EKE (Bellare and Merritt 1992)

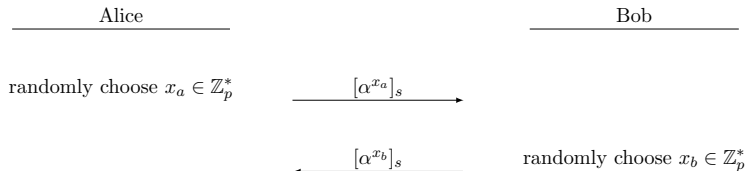
Pick prim. root $\alpha \in \mathbb{Z}_p$



Alice and Bob share $K = \alpha^{x_a \cdot x_b}$.

First Protocol: EKE (Bellare and Merritt 1992)

Pick prim. root $\alpha \in \mathbb{Z}_p$

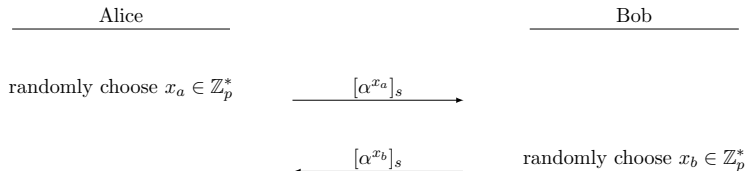


Alice and Bob share $K = \alpha^{x_a \cdot x_b}$.

Uses password directly \implies many insecurities found

First Protocol: EKE (Bellare and Merritt 1992)

Pick prim. root $\alpha \in \mathbb{Z}_p$



Alice and Bob share $K = \alpha^{x_a \cdot x_b}$.

Uses password directly \implies many insecurities found

Ex: Decypher $[\alpha^{x_a}]_{s'}$ – rule out s' if output in $[p, 2^n - 1]$

Desired Security Properties

Offline dictionary attack resistance

Don't leak info which can be used in brute force search

Forward secrecy for established keys

Past keys secure if password disclosed

Implies passive attacker w/ password cannot compute key

Known session security

All secrets of one session reveals nothing about others

Online dictionary attack resistance

Attacker can only test one password per protocol execution

J-PAKE (Hao and Ryan 2010)

Setup Let $G = \langle g \rangle$ be an order q subgroup of \mathbb{Z}_p^*

Alice picks $x_1 \in_R [0, q-1]$ and $x_2 \in_R [1, q-1]$

Bob picks $x_3 \in_R [0, q-1]$ and $x_4 \in_R [1, q-1]$

J-PAKE (Hao and Ryan 2010)

Setup Let $G = \langle g \rangle$ be an order q subgroup of \mathbb{Z}_p^*

Alice picks $x_1 \in_R [0, q - 1]$ and $x_2 \in_R [1, q - 1]$

Bob picks $x_3 \in_R [0, q - 1]$ and $x_4 \in_R [1, q - 1]$

Rd 1 Alice sends g^{x_1}, g^{x_2} and ZKPs of x_1 and x_2 to Bob

Bob sends g^{x_3}, g^{x_4} and ZKPs of x_3 and x_4 to Alice

[Alice and Bob verify ZKPs and $g^{x_2}, g^{x_4} \neq 1$]

J-PAKE (Hao and Ryan 2010)

Setup Let $G = \langle g \rangle$ be an order q subgroup of \mathbb{Z}_p^*

Alice picks $x_1 \in_R [0, q-1]$ and $x_2 \in_R [1, q-1]$

Bob picks $x_3 \in_R [0, q-1]$ and $x_4 \in_R [1, q-1]$

Rd 1 Alice sends g^{x_1}, g^{x_2} and ZKPs of x_1 and x_2 to Bob

Bob sends g^{x_3}, g^{x_4} and ZKPs of x_3 and x_4 to Alice

[Alice and Bob verify ZKPs and $g^{x_2}, g^{x_4} \neq 1$]

Rd 2 Alice sends $A = g^{(x_1+x_3+x_4)x_2 \cdot s}$ and ZKP of x_2s

Bob sends $B = g^{(x_1+x_2+x_3)x_4 \cdot s}$ and ZKP of x_4s

J-PAKE (Hao and Ryan 2010)

Setup Let $G = \langle g \rangle$ be an order q subgroup of \mathbb{Z}_p^*

Alice picks $x_1 \in_R [0, q-1]$ and $x_2 \in_R [1, q-1]$

Bob picks $x_3 \in_R [0, q-1]$ and $x_4 \in_R [1, q-1]$

Rd 1 Alice sends g^{x_1}, g^{x_2} and ZKPs of x_1 and x_2 to Bob

Bob sends g^{x_3}, g^{x_4} and ZKPs of x_3 and x_4 to Alice

[Alice and Bob verify ZKPs and $g^{x_2}, g^{x_4} \neq 1$]

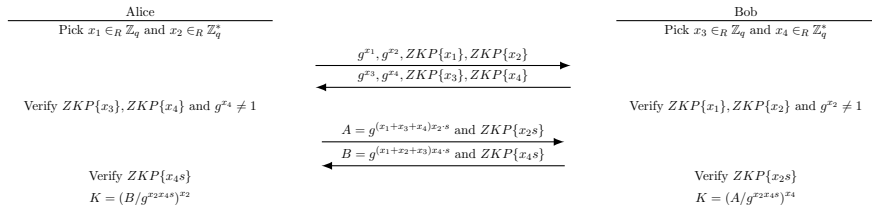
Rd 2 Alice sends $A = g^{(x_1+x_3+x_4)x_2 \cdot s}$ and ZKP of x_2s

Bob sends $B = g^{(x_1+x_2+x_3)x_4 \cdot s}$ and ZKP of x_4s

They share

$$K = \underbrace{(B/g^{x_2x_4s})^{x_2}}_{\text{Computable by Alice}} = g^{(x_1+x_3)x_2x_4s} = \underbrace{(A/g^{x_2x_4s})^{x_4}}_{\text{Computable by Bob}}.$$

J-PAKE (Hao and Ryan 2010)



J-PAKE (Hao and Ryan 2010)

Satisfies all 4 desired properties (under DSDH assumption)
More robust security proof in 2015

Only two rounds of communication

No *explicit* key confirmation (only implicit)

Not patented (ISO/IEC 11770-4 standard)

PAK/PPK (Boyko, MacKenzie, and Patel 2000)

Alternative PAKE, via hashing password with random elements and powering

Proposes ‘simulation model’ to prove security (under DDH & random oracle)

Satisfies all 4 desired properties

Only two/three rounds of communication

PAK has explicit key confirmation

PPK has implicit key confirmation

Dragonfly (Harkins 2012)

Alternative PAKE, using discrete log / CDH assumption
(IEEE Std 802.11-2012)

No formal security proofs, but claims resistance to ‘active attacks, passive attacks, and off-line dictionary attacks’
(previously attacked, but upgraded)

Only two rounds of communication

Very fast compared to other protocols

Comparisons

COMPARISON OF PRACTICAL DIFFIE-HELLMAN-BASED PAKE PROTOCOLS PROVEN SECURE IN THE BPR MODEL [5]

	Rounds / Flows	Assumptions ^a				Complexity	
		CRS	ROM	ICM	AAM	Communication ^b	Time ^c
J-PAKE with Schnorr [24]	2 / 4 or 3 / 3		✗		✗ (DSDH or CSDH + DDH)	$12 \times G + 6 \times \mathbb{Z}_p$	28 exp (12 exp + 8 mexp)
EKE [5], [7]	1 / 2			✗	CDH	$2 \times G$	4 exp + 2 memb + 2 enc
SPEKE [29], [35]	1 / 2		✗		DIDH ^d	$2 \times G$	4 exp + 2 memb
PPK [10]	2 / 2		✗		DDH	$2 \times G$	6 exp + 2 memb
SPAKE2 [3]	1 / 2		✗		CDH	$2 \times G$	4 exp + 2 memb
GK-SPOKE [1], [21], [30]	2 / 2	✗			DDH + PRG ^e	$6 \times G$	17 exp (4 exp + 7 mexp) + 6 memb
GL-SPOKE [1], [18], [32]	2 / 2	✗			DDH	$7 \times G$	21 exp (4 exp + 7 mexp) + 7 memb
KV-SPOKE [1], [33]	1 / 2	✗			DDH	$10 \times G$	30 exp (2 exp + 12 mexp) + 10 memb

^a CRS: common reference string, ROM: random-oracle model, ICM: ideal-cipher model, AAM: algebraic-adversary model;

^b G : group elements, \mathbb{Z}_p : scalars;

^c exp: number of exponentiations; mexp: number of multi-exponentiations; memb: verification of the membership of a group element to the cyclic group G . For elliptic curve with small co-factor, this only costs a small number of additions on the curve, but for subgroups of \mathbb{Z}_q (q being a prime larger than p , the order of the group G), this costs an exponentiation (with exponent $p - 1$); enc: encryption with the ideal cipher; multiplications, hash evaluations, and PRG evaluations are omitted;

^d DIDH: decision inverted-additive Diffie-Hellman assumption [35] (see Fig. 2 and the Appendix);

^e PRG: pseudo-random generator.

Security of the J-PAKE Password-Authenticated Key Exchange Protocol.

M. Abdalla, F. Benhamouda and P. MacKenzie, SP'2015.

PART 2

Extension to Group Setting

The Fairy-Ring Dance

Group members establish the group key and the pair-wise keys simultaneously.

The Fairy-Ring Dance

Group members establish the group key and the pair-wise keys simultaneously.

For pair-wise key:

Use plain two-party PAKE protocols

The Fairy-Ring Dance

Group members establish the group key and the pair-wise keys simultaneously.

For pair-wise key:

Use plain two-party PAKE protocols

For group key:

- ▶ Everyone additionally choose another random $y_i \in_R \mathbb{Z}_q$ and broadcast g^{y_i} w/ ZKP
- ▶ Everyone can calculate $g^{z_i} := g^{y_{i+1} - y_{i-1}} = g^{y_{i+1}} / g^{y_{i-1}}$

The Fairy-Ring Dance

Group members establish the group key and the pair-wise keys simultaneously.

For pair-wise key:

Use plain two-party PAKE protocols

For group key:

- ▶ Everyone additionally choose another random $y_i \in_R \mathbb{Z}_q$ and broadcast g^{y_i} w/ ZKP
- ▶ Everyone can calculate $g^{z_i} := g^{y_{i+1}-y_{i-1}} = g^{y_{i+1}} / g^{y_{i-1}}$

Group key:

$$K_i = (g^{y_{i-1}})^{ny_i} \cdot (g^{z_i y_i})^{n-1} \cdot (g^{z_{i+1} y_{i+1}})^{n-2} \dots (g^{z_{i-2} y_{i-2}}) \quad (1)$$

$$= g^{y_1 y_2 + y_2 y_3 + \dots + y_n y_1} \quad (2)$$

JPAKE+

Setup Let $G = \langle g \rangle$ be an order q subgroup of \mathbb{Z}_p^*

JPAKE+

Setup Let $G = \langle g \rangle$ be an order q subgroup of \mathbb{Z}_p^*

Rd 1 P_i chooses, for all $j \neq i$

$$a_{ij} \in_R \mathbb{Z}_q \qquad b_{ij} \in_R \mathbb{Z}_q^* \qquad y_i \in_R \mathbb{Z}_q,$$

and broadcasts

$$g^{a_{ij}} \quad g^{b_{ij}} \quad g^{y_i} \quad \text{ZKP}(a_{ij}) \quad \text{ZKP}(b_{ij}) \quad \text{ZKP}(y_i).$$

After, P_i checks ZKPs and

$$g^{z_i} = g^{y_{i+1}} / g^{y_{i-1}} \neq 1, \qquad g^{b_{ji}} \neq 1$$

JPAKE+

Setup Let $G = \langle g \rangle$ be an order q subgroup of \mathbb{Z}_p^*

Rd 1 P_i chooses, for all $j \neq i$

$$a_{ij} \in_R \mathbb{Z}_q \quad b_{ij} \in_R \mathbb{Z}_q^* \quad y_i \in_R \mathbb{Z}_q,$$

and broadcasts

$$g^{a_{ij}} \quad g^{b_{ij}} \quad g^{y_i} \quad \text{ZKP}(a_{ij}) \quad \text{ZKP}(b_{ij}) \quad \text{ZKP}(y_i).$$

After, P_i checks ZKPs and

$$g^{z_i} = g^{y_{i+1}} / g^{y_{i-1}} \neq 1, \quad g^{b_{ji}} \neq 1$$

Rd 2 P_i computes and broadcasts, for $j \neq i$

$$\beta_{ij} := \left(g^{a_{ij} + a_{ji} + b_{ji}} \right)^{b_{ij} \cdot s} \quad \text{ZKP}(b_{ij} \cdot s)$$

JPAKE+

Rd 3 Every P_i broadcasts

$$(g^{z_i})^{y_i} \text{ and } \text{ZKP}\{\tilde{y}_i\}.$$

Let $K_{ij} := (\beta_{ji}/g^{b_{ij} \cdot b_{ji} \cdot s})^{b_{ij}} =$ pairwise JPAKE key

JPAKE+

Rd 3 Every P_i broadcasts

$$(g^{z_i})^{y_i} \text{ and } \text{ZKP}\{\tilde{y}_i\}.$$

Let $K_{ij} := (\beta_{ji}/g^{b_{ij} \cdot b_{ji} \cdot s})^{b_{ij}}$ = pairwise JPAKE key

► Members compute

$$\kappa_{ij}^{MAC} = H(K_{ij} || 'MAC') \quad \kappa_{ij}^{KC} = H(K_{ij} || 'KC')$$

JPAKE+

Rd 3 Every P_i broadcasts

$$(g^{z_i})^{y_i} \text{ and } \text{ZKP}\{\tilde{y}_i\}.$$

Let $K_{ij} := (\beta_{ji}/g^{b_{ij} \cdot b_{ji} \cdot s})^{b_{ij}} =$ pairwise JPAKE key

- Members compute

$$\kappa_{ij}^{MAC} = H(K_{ij} || 'MAC') \quad \kappa_{ij}^{KC} = H(K_{ij} || 'KC')$$

- Members broadcast

$$t_{ij}^{MAC} = HMAC(\kappa_{ij}^{MAC}, g^{y_i} || \text{ZKP}\{y_i\} || (g^{z_i})^{y_i} || \text{ZKP}\{\tilde{y}_i\})$$
$$t_{ij}^{KC} = HMAC(\kappa_{ij}^{KC}, 'KC' || i || j || g^{a_{ij}} || g^{b_{ij}} || g^{a_{ji}} || g^{b_{ji}})$$

All members share $K = g^{y_1 \cdot y_2 + y_2 \cdot y_3 + \dots + y_n \cdot y_1}$

PART 3

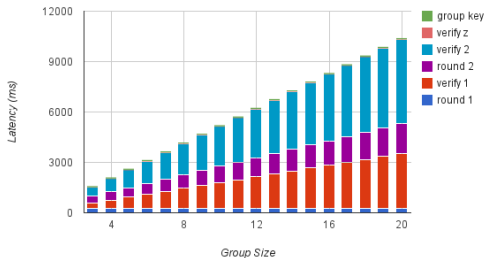
Timings

Specifications

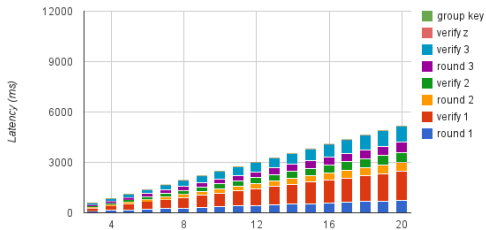
- ▶ All protocol benchmarks were implemented in Java 1.6 and run on a server (3GHz AMD processor, 6GB of RAM) running Ubuntu 12.04.
- ▶ Benchmarks measured latency, the amount of work each device would have to do in the group excluding communication.

Results

Latency measurement in SPEKE+

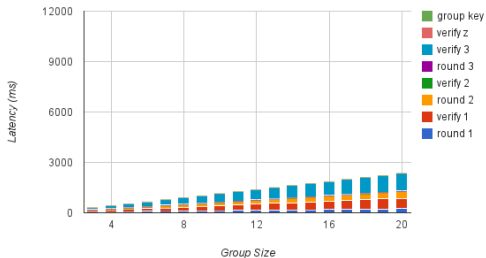


Latency measurement in JPAKE+

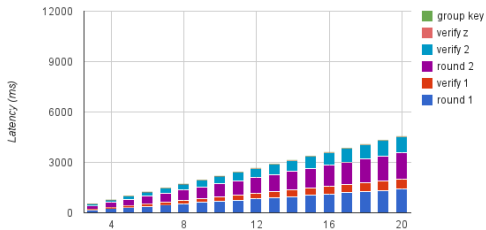


Results

Latency measurement in Dragonfly+



Latency measurement in PPK+



Conclusion

Conclusion

It is possible to transfer PAKEs into GPAKEs while preserving round efficiency

SPEKE+ is very slow

J-PAKE+ is a bit slow, but ‘proven’ secure (under DSDH)

PPK is faster

Dragonfly is fastest, but no security proof
(despite IEEE 802.11-2012 standard)

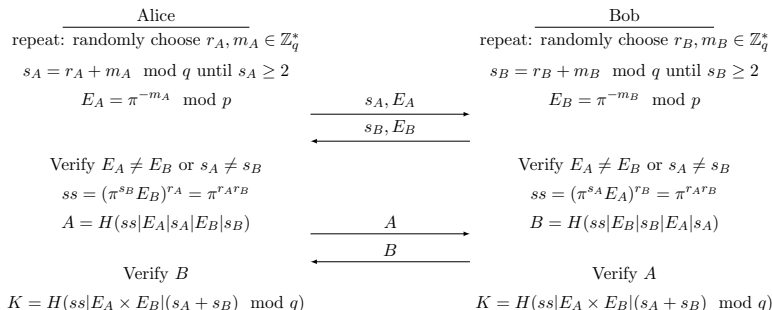
THANK YOU

Dragonfly

Setup Let Q be a cyclic subgroup of \mathbb{Z}_p^* with prime order q . Both members map the password to an element $\pi \in Q$.

Dragonfly

Setup Let Q be a cyclic subgroup of \mathbb{Z}_p^* with prime order q . Both members map the password to an element $\pi \in Q$.



Dragonfly+

Setup Let $Q = \langle g \rangle \subset \mathbb{Z}_p^*$ w/ order q and $\pi \in Q$

Every P_i chooses

$$r_{ij}, m_{ij} \in_R \mathbb{Z}_q^* \quad \forall j \neq i$$

$$y_i \in_R \mathbb{Z}_q$$

and computes $g^{y_i} \bmod p$

Dragonfly+

Setup Let $Q = \langle g \rangle \subset \mathbb{Z}_p^*$ w/ order q and $\pi \in Q$

Every P_i chooses

$$\begin{aligned} r_{ij}, m_{ij} &\in_R \mathbb{Z}_q^* & \forall j \neq i \\ y_i &\in_R \mathbb{Z}_q \end{aligned}$$

and computes $g^{y_i} \bmod p$

Rd 1 Every P_i broadcasts

$$\begin{aligned} s_{ij} &:= r_{ij} + m_{ij} \bmod q \\ E_{ij} &:= \pi^{-m_{ij}} \bmod p \\ g^{y_i} &\bmod p \\ \text{ZKP}\{y_i\} \end{aligned}$$

[All verify ZKP, $g^{z_i} \neq 1$ and check for reflection attacks]

Dragonfly+

Rd 2 Each member computes pairwise shared secrets:

$$ss_{ij} = (\pi^{s_{ji}} E_{ji})^{r_{ij}}$$

Each member broadcasts

$$H(ss_{ij} || E_{ij} || s_{ij} || E_{ji} || s_{ji})$$

Dragonfly+

Rd 2 Each member computes pairwise shared secrets:

$$ss_{ij} = (\pi^{s_{ji}} E_{ji})^{r_{ij}}$$

Each member broadcasts

$$H(ss_{ij} || E_{ij} || s_{ij} || E_{ji} || s_{ji})$$

Rd 3 Every member broadcasts

$$(g^{z_i})^{y_i} \text{ and ZKP}\{\tilde{y}_i\}.$$

Dragonfly+

Rd 2 Each member computes pairwise shared secrets:

$$ss_{ij} = (\pi^{s_{ji}} E_{ji})^{r_{ij}}$$

Each member broadcasts

$$H(ss_{ij} || E_{ij} || s_{ij} || E_{ji} || s_{ji})$$

Rd 3 Every member broadcasts

$$(g^{z_i})^{y_i} \text{ and ZKP}\{\tilde{y}_i\}.$$

Let $K_{ij} :=$ pairwise Dragonfly key. Members compute

$$\kappa_{ij}^{MAC} = H(K_{ij} || 'MAC') \quad \kappa_{ij}^{KC} = H(K_{ij} || 'KC')$$

Dragonfly+

Rd 2 Each member computes pairwise shared secrets:

$$ss_{ij} = (\pi^{s_{ji}} E_{ji})^{r_{ij}}$$

Each member broadcasts

$$H(ss_{ij} || E_{ij} || s_{ij} || E_{ji} || s_{ji})$$

Rd 3 Every member broadcasts

$$(g^{z_i})^{y_i} \text{ and } \text{ZKP}\{\tilde{y}_i\}.$$

Let $K_{ij} :=$ pairwise Dragonfly key. Members compute

$$\kappa_{ij}^{MAC} = H(K_{ij} || 'MAC') \quad \kappa_{ij}^{KC} = H(K_{ij} || 'KC')$$

Members broadcast

$$\begin{aligned} t_{ij}^{MAC} &= HMAC(\kappa_{ij}^{MAC}, g^{y_i} || \text{ZKP}\{y_i\} || (g^{z_i})^{y_i} || \text{ZKP}\{\tilde{y}_i\}) \\ t_{ij}^{KC} &= HMAC(\kappa_{ij}^{KC}, 'KC' || i || j || E_{ij} || E_{ji}) \end{aligned}$$

Dragonfly+

Rd 2 Each member computes pairwise shared secrets:

$$ss_{ij} = (\pi^{s_{ji}} E_{ji})^{r_{ij}}$$

Each member broadcasts

$$H(ss_{ij} || E_{ij} || s_{ij} || E_{ji} || s_{ji})$$

Rd 3 Every member broadcasts

$$(g^{z_i})^{y_i} \text{ and } \text{ZKP}\{\tilde{y}_i\}.$$

Let $K_{ij} :=$ pairwise Dragonfly key. Members compute

$$\kappa_{ij}^{MAC} = H(K_{ij} || 'MAC') \quad \kappa_{ij}^{KC} = H(K_{ij} || 'KC')$$

Members broadcast

$$\begin{aligned} t_{ij}^{MAC} &= HMAC(\kappa_{ij}^{MAC}, g^{y_i} || \text{ZKP}\{y_i\} || (g^{z_i})^{y_i} || \text{ZKP}\{\tilde{y}_i\}) \\ t_{ij}^{KC} &= HMAC(\kappa_{ij}^{KC}, 'KC' || i || j || E_{ij} || E_{ji}) \end{aligned}$$

All members share $K = g^{y_1 \cdot y_2 + y_2 \cdot y_3 + \dots + y_n \cdot y_1}$

PAK

Setup Let $G = \langle g \rangle$ be an order q subgroup of \mathbb{Z}_p^*

Let $p = rq + 1$ where r, q relatively prime.

Let π be the password and H_1, H_{2a}, H_{2b}, H_3 be random, independent hash functions.

PAK

Setup Let $G = \langle g \rangle$ be an order q subgroup of \mathbb{Z}_p^*

Let $p = rq + 1$ where r, q relatively prime.

Let π be the password and H_1, H_{2a}, H_{2b}, H_3 be random, independent hash functions.

PAK

A

B

$$x \in_R \mathbb{Z}_q$$

$$m = g^x \cdot (H_1(A, B, \pi))^r$$

$$\xrightarrow{m}$$

$$\text{Test } m \stackrel{?}{\neq} 0 \pmod{p}$$

$$y \in_R \mathbb{Z}_q$$

$$\mu = g^y$$

$$\sigma = \left(\frac{m}{(H_1(A, B, \pi))^r} \right)^y$$

$$\sigma = \mu^x$$

$$\xleftarrow{\mu, k}$$

$$k = H_{2a}(A, B, m, \mu, \sigma, \pi)$$

$$\text{Test } k \stackrel{?}{=} H_{2a}(A, B, m, \mu, \sigma, \pi)$$

$$k' = H_{2b}(A, B, m, \mu, \sigma, \pi)$$

$$K = H_3(A, B, m, \mu, \sigma, \pi)$$

$$\xrightarrow{k'}$$

$$\text{Test } k' \stackrel{?}{=} H_{2b}(A, B, m, \mu, \sigma, \pi)$$

$$K = H_3(A, B, m, \mu, \sigma, \pi)$$

PPK

Setup Let $G = \langle g \rangle$ be an order q subgroup of \mathbb{Z}_p^*

Let $p = rq + 1$ where r, q relatively prime.

Let π be the password and H_1, H_3 be random, independent hash functions.

PPK

Setup Let $G = \langle g \rangle$ be an order q subgroup of \mathbb{Z}_p^*

Let $p = rq + 1$ where r, q relatively prime.

Let π be the password and H_1, H_3 be random, independent hash functions.

PPK

A

B

$x \in_R \mathbb{Z}_q$

$m = g^x \cdot (H_1(A, B, \pi))^r$

\xrightarrow{m}

Test $m \stackrel{?}{\neq} 0 \bmod p$

$y \in_R \mathbb{Z}_q$

$\mu = g^y \cdot (H_1(A, B, \pi))^r$

$\sigma = (\frac{m}{(H_1(A, B, \pi))^r})^y$

Test $\mu \stackrel{?}{\neq} 0 \bmod p$

$\sigma = (\frac{\mu}{(H_1(A, B, \pi))^r})^x$

$K = H_3(A, B, m, \mu, \sigma, \pi)$

$\xleftarrow{\mu}$

$K = H_3(A, B, m, \mu, \sigma, \pi)$