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

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

No need for public key infrastructure (TA/CA for public keys)

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

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

First Protocol: EKE (Bellovin and Merrit 1992)

Pick prim. root $\alpha \in \mathbb{Z}_p$ for n-bit prime p

Alice		Bob
randomly choose $x_a \in \mathbb{Z}_p^*$	$\xrightarrow{[\alpha^{x_a}]_\pi}$	
	$\xleftarrow{[\alpha^{x_b}]_\pi}$	randomly choose $x_b \in \mathbb{Z}_p^*$

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

First Protocol: EKE (Bellovin and Merrit 1992)

Pick prim. root $\alpha \in \mathbb{Z}_p$ for n-bit prime p

Alice		Bob
randomly choose $x_a \in \mathbb{Z}_p^*$	$\xrightarrow{[\alpha^{x_a}]_\pi}$	
	$\xleftarrow{[\alpha^{x_b}]_\pi}$	randomly choose $x_b \in \mathbb{Z}_p^*$

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

Uses password directly $\implies$ many insecurities found

First Protocol: EKE (Bellovin and Merrit 1992)

Pick prim. root $\alpha \in \mathbb{Z}_p$ for n-bit prime p

Alice		Bob
randomly choose $x_a \in \mathbb{Z}_p^*$	$\xrightarrow{[\alpha^{x_a}]_\pi}$	
	$\xleftarrow{[\alpha^{x_b}]_\pi}$	randomly choose $x_b \in \mathbb{Z}_p^*$

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

Uses password directly $\implies$ many insecurities found

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

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

J-PAKE (Hao and Ryan 2010)

Setup Let $G = <g>$ 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 $x_2 s$
Bob sends $B = g^{(x_1+x_2+x_3)x_4 \cdot s}$ and ZKP of $x_4 s$

They share

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

J-PAKE (Hao and Ryan 2010)

Alice		Bob
Pick $x_1 \in_R \mathbb{Z}_q$ and $x_2 \in_R \mathbb{Z}_q^*$		Pick $x_3 \in_R \mathbb{Z}_q$ and $x_4 \in_R \mathbb{Z}_q^*$
	$\xrightarrow{g^{x_1}, g^{x_2}, ZKP\{x_1\}, ZKP\{x_2\}}$	
	$\xleftarrow{g^{x_3}, g^{x_4}, ZKP\{x_3\}, ZKP\{x_4\}}$	
Verify $ZKP\{x_3\}, ZKP\{x_4\}$ and $g^{x_4} \neq 1$		Verify $ZKP\{x_1\}, ZKP\{x_2\}$ and $g^{x_2} \neq 1$
	$\xrightarrow{A = g^{(x_1+x_3+x_4)x_2 \cdot s} \text{ and } ZKP\{x_2 s\}}$	
	$\xleftarrow{B = g^{(x_1+x_2+x_3)x_4 \cdot s} \text{ and } ZKP\{x_4 s\}}$	
Verify $ZKP\{x_4 s\}$		Verify $ZKP\{x_2 s\}$
$K = (B/g^{x_2 x_4 s})^{x_2}$		$K = (A/g^{x_2 x_4 s})^{x_4}$

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 PAKEs, via hashing password with random elements and powering

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

Satisfies all desired properties (and more)

Only two/three rounds of communication

PAK has explicit key confirmation
PPK has implicit key confirmation

Dragonfly (Harkins 2012)

Another PAKE, using discrete log / CDH problem as basis (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

		Assumptions[a]				
	Rounds / Flows	ROM	ICM	AAM		Time[c]
J-PAKE with Schnorr [24]	2 / 4 or 3 / 3	✗		✗	DSDH or (CSDH + DDH)	28 exp (12 exp + 8 mexp)
EKE [5], [7]	1 / 2		✗		CDH	4 exp + 2 memb + 2 enc
SPEKE [29], [35]	1 / 2	✗			DIDH[d]	4 exp + 2 memb
PPK [10]	2 / 2	✗			DDH	6 exp + 2 memb
SPAKE2 [3]	1 / 2	✗			CDH	4 exp + 2 memb

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

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

BPR = model of Bellare, Pointcheval, and Rogaway from EUROCRYPT 2000

PART 2
Extension to Group Setting

The Fairy-Ring Dance

Group members establish pairwise keys (for trust / authentication) and a group key simultaneously.

The Fairy-Ring Dance

Group members establish pairwise keys (for trust / authentication) and a group key simultaneously.

For pairwise key:
Use plain two-party PAKE protocols

The Fairy-Ring Dance

Group members establish pairwise keys (for trust / authentication) and a group key simultaneously.

For pairwise 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 pairwise keys (for trust / authentication) and a group key simultaneously.

For pairwise 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 (Burmester-Desmedt group key agreement protocol):

$$
\begin{align}
K_i &= (g^{y_{i-1}})^{n y_i} \cdot (y^{z_i y_i})^{n-1} \cdot (y^{z_{i+1} y_{i+1}})^{n-2} \cdots (g^{z_{i-2} y_{i-2}}) \tag{1} \\
&= g^{y_1 y_2 + y_2 y_3 + \cdots + y_n y_1} \tag{2}
\end{align}
$$

JPAKE+

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

JPAKE+

Setup Let $G = <g>$ 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

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

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 $G = <g>$ 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

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

After, P_i checks ZKPs and

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

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

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

JPAKE+

Rd 3 Every P_i broadcasts

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

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

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

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

- Members compute

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

JPAKE+

Rd 3 Every P_i broadcasts

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

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

- Members compute

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

- Members broadcast (and then verify)

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

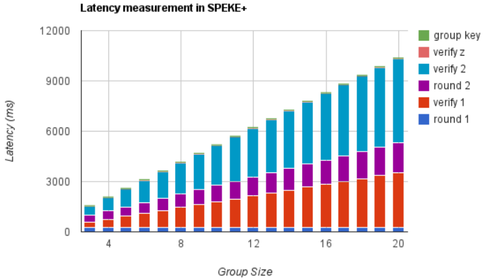
All members share $K = g^{y_1\cdot y_2 + y_2\cdot y_3 + \cdots + y_n\cdot y_1}$

PART 3
Timings

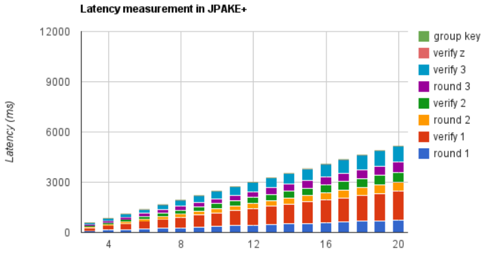
Specifications

- 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

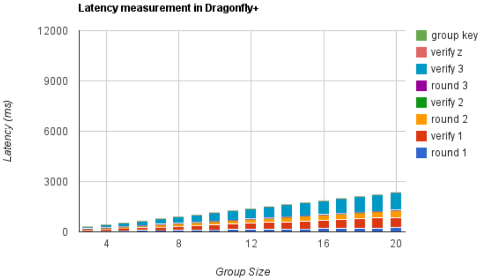


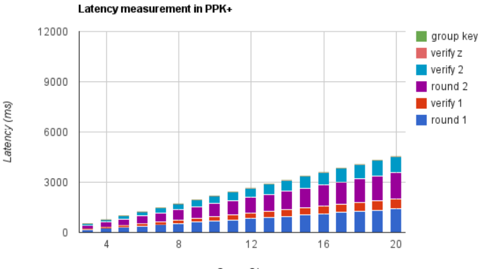
Latency measurement in SPEKE+
12000
9000
6000
3000
0
Latency (ms)
4
8
12
16
20
Group Size
group key
verify z
verify 2
round 2
verify 1
round 1



Latency measurement in JPAKE+
12000
9000
6000
3000
0
Latency (ms)
4
8
12
16
20
group key
verify z
verify 3
round 3
verify 2
round 2
verify 1
round 1

Results


Latency measurement in Dragonfly+
12000
9000
6000
3000
0
Latency (ms)
4
8
12
16
20
Group Size
group key
verify z
verify 3
round 3
verify 2
round 2
verify 1
round 1


Latency measurement in PPK+
12000
9000
6000
3000
0
Latency (ms)
4
8
12
16
20
group key
verify z
verify 2
round 2
verify 1
round 1

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 $\pi \in Q$.

Alice		Bob
repeat: randomly choose $r_A, m_A \in \mathbb{Z}_q^*$		repeat: randomly choose $r_B, m_B \in \mathbb{Z}_q^*$
$s_A = r_A + m_A \mod q$ until $s_A \geq 2$		$s_B = r_B + m_B \mod q$ until $s_B \geq 2$
$E_A = \pi^{-m_A} \mod p$	$\xrightarrow{s_A, E_A}$	$E_B = \pi^{-m_B} \mod p$
	$\xleftarrow{s_B, E_B}$	
Verify $E_A \neq E_B$ or $s_A \neq s_B$		Verify $E_A \neq E_B$ or $s_A \neq s_B$
$ss = (\pi^{s_B} E_B)^{r_A} = \pi^{r_A r_B}$		$ss = (\pi^{s_A} E_A)^{r_B} = \pi^{r_A r_B}$
$A = H(ss \vert E_A \vert s_A \vert E_B \vert s_B)$	$\xrightarrow{A}$	$B = H(ss \vert E_B \vert s_B \vert E_A \vert s_A)$
	$\xleftarrow{B}$	
Verify B		Verify A
$K = H(ss \vert E_A \times E_B \vert (s_A + s_B) \mod q)$		$K = H(ss \vert E_A \times E_B \vert (s_A + s_B) \mod q)$

Dragonfly+

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

Every P_i chooses

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

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

Dragonfly+

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

Every P_i chooses

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

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

Rd 1 Every P_i broadcasts

$$s_{ij} := r_{ij} + m_{ij} \mod q$$
$$E_{ij} := \pi^{-m_{ij}} \mod p$$
$$g^{y_i} \mod p$$
$$\text{ZKP}\{y_i\}$$

[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

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

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

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

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

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

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

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

Members broadcast

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

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

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

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

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

Members broadcast

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

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

PAK

Setup Let $G = <g>$ 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 $G = <g>$ 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
$x \in_R Z_q$		
$m = g^x \cdot (H_1(A, B, \pi))^r$	$\xrightarrow{\quad m \quad}$	Test $m \stackrel{?}{\not\equiv} 0 \bmod p$
		$y \in_R Z_q$
		$\mu = g^y$
		$\sigma = \left(\frac{m}{(H_1(A,B,\pi))^r}\right)^y$
$\sigma = \mu^x$	$\xleftarrow{\quad \mu, k \quad}$	$k = H_{2a}(A, B, m, \mu, \sigma, \pi)$
Test $k \stackrel{?}{=} H_{2a}(A, B, m, \mu, \sigma, \pi)$		
$k' = H_{2b}(A, B, m, \mu, \sigma, \pi)$		
$K = H_3(A, B, m, \mu, \sigma, \pi)$	$\xrightarrow{\quad k' \quad}$	Test $k' \stackrel{?}{=} H_{2b}(A, B, m, \mu, \sigma, \pi)$
		$K = H_3(A, B, m, \mu, \sigma, \pi)$

PPK

Setup Let $G = <g>$ 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 $G = <g>$ 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
$x \in_R Z_q$		
$m = g^x \cdot (H_1(A, B, \pi))^r$	$\xrightarrow{m}$	Test $m \overset{?}{\not\equiv} 0 \bmod p$
		$y \in_R Z_q$
		$\mu = g^y \cdot (H_1(A, B, \pi))^r$
		$\sigma = (\frac{m}{(H_1(A,B,\pi))^r})^y$
Test $\mu \overset{?}{\not\equiv} 0 \bmod p$	$\xleftarrow{\mu}$	$K = H_3(A, B, m, \mu, \sigma, \pi)$
$\sigma = (\frac{\mu}{(H_1(A,B,\pi))^r})^x$		
$K = H_3(A, B, m, \mu, \sigma, \pi)$		