Queensland University of Technology

Sun microsystems

Multi-Factor Password-Authenticated Key Exchange

Douglas Stebila, Poornaprajna Udupi, and Sheueling Chang

Information Security Institute
Queensland University of Technology, Brisbane, Australia

Sun Microsystems Laboratories
Menlo Park, California, USA

Wednesday, January 20, 2010

Two major security problems on today's Internet are **phishing** and **spyware**. They aim to extract valuable private information.

Existing security approaches, such as SSL, passwords, and one-time tokens, don't comprehensively protect private information in the face of these attacks.

Multi-factor password-authenticated key exchange provides

- strong, mutual multi-factor authentication (client-to-server and server-to-client) and
- confidentiality

even in the face of

- spyware and
- phishing

and has formal security arguments.

Private information is valuable

Prices on the black market (Symantec, April 2008)

bank accounts:	$ 10 - 1000
credit cards:	$ 1 - 20
identities:	$ 1 - 15
eBay accounts:	$ 1 - 8
email passwords:	$ 4 - 30

How do attackers get this information?

Attack the server

- hack into the server
- bribe an employee
- steal a backup tape

Attack the user

- steal a computer
- hack into a computer
- convince the user to tell you their password (phishing)
- install spyware on their computer

Two security goals

1. Confidentiality
   - establish a private channel using a shared secret key
   - use public key cryptography to get a shared secret key
2. Authentication
   - user and server must prove to each other that they are who they say they are
   - using multiple attributes, of different natures, can enhance authentication robustness

Confidentiality and authentication are intertwined.

It's no good having confidential communications with someone if it's the wrong someone.

Authentication

Client-to-server:
User can show that

- she knows her password

Server-to-client:
Server can show that

- it looks like PayPal
- it has the domain name “paypal.com”
- a lock icon shows up in the browser because
- it has an SSL certificate

Existing approaches

- SSL + basic passwords
- SSL + client certificates
- SSL + multi-layer authentication
- password-based key agreement

Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

SSL + basic passwords:

Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

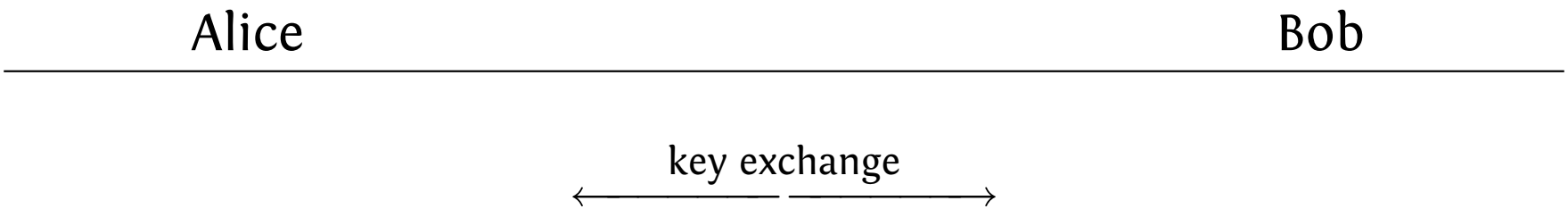
SSL + basic passwords:

Alice	Bob

Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

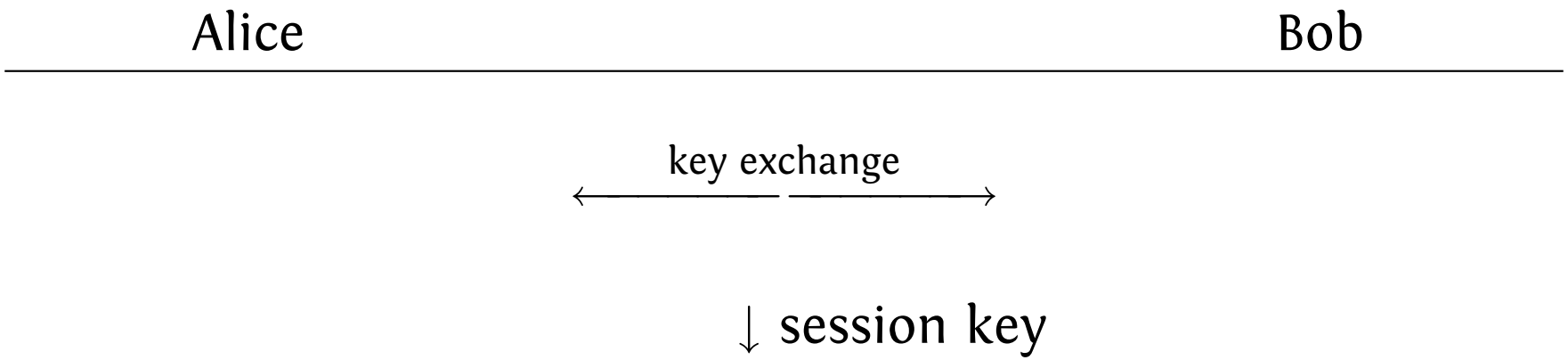
SSL + basic passwords:



Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

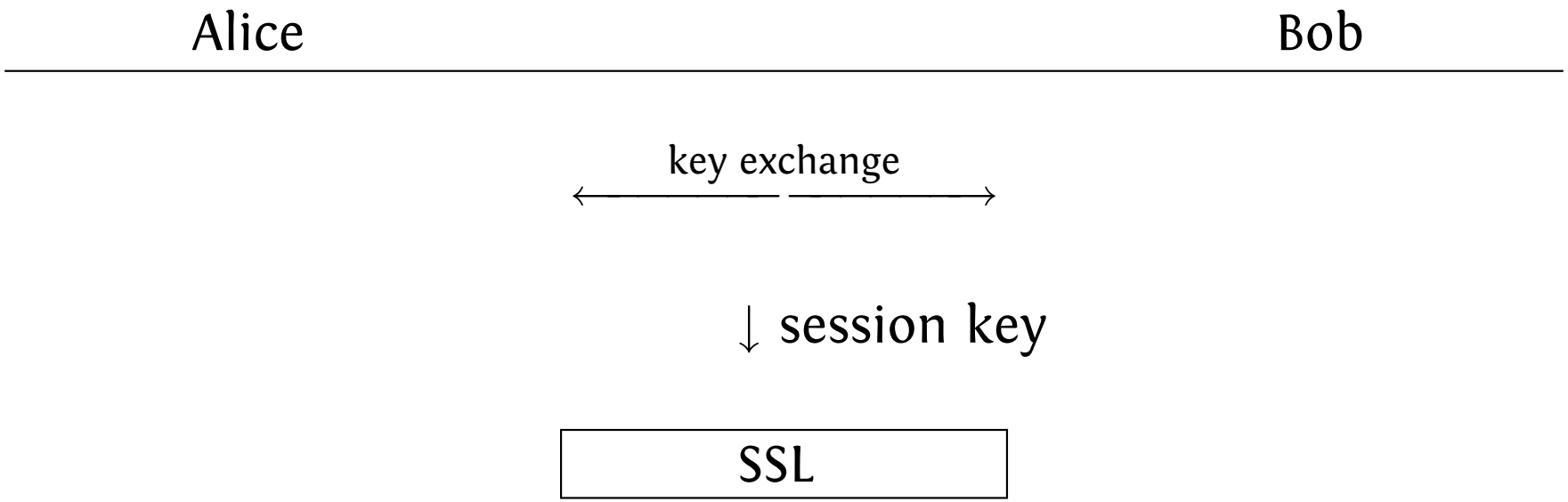
SSL + basic passwords:



Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

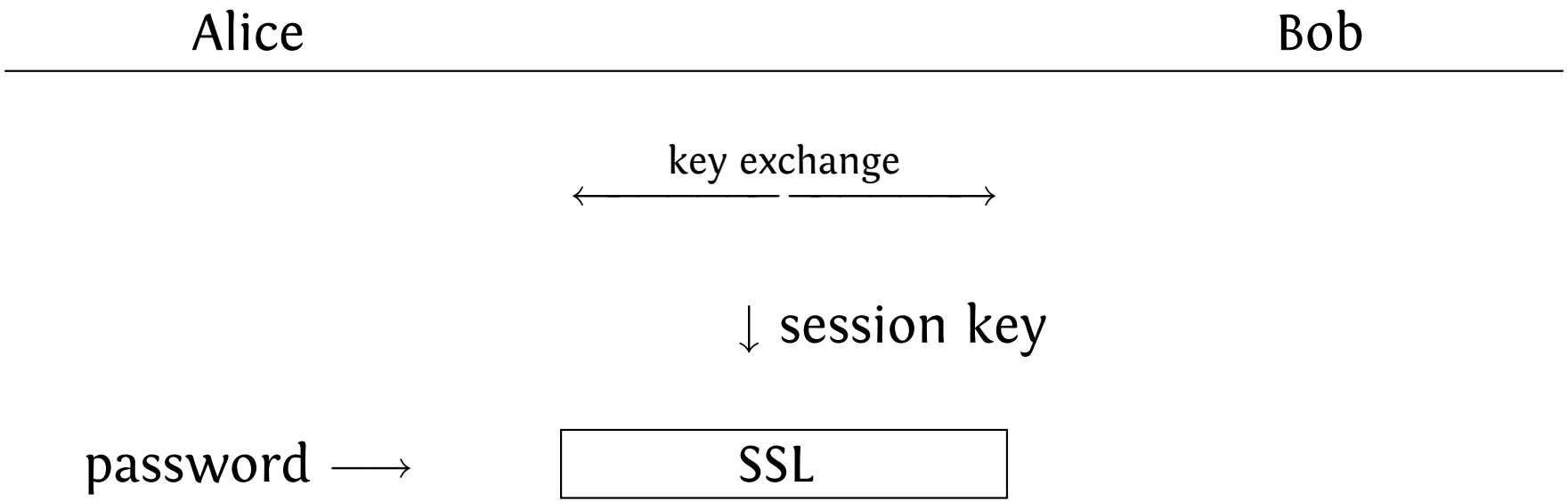
SSL + basic passwords:



Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

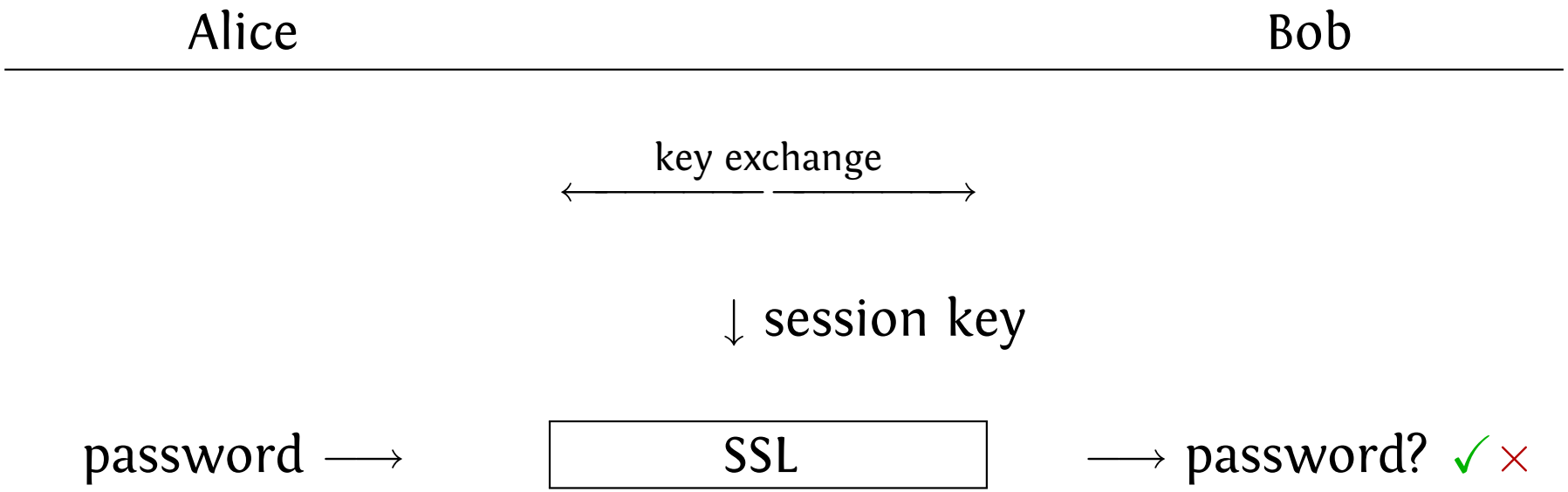
SSL + basic passwords:



Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

SSL + basic passwords:



Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

Password-authenticated key exchange:

Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

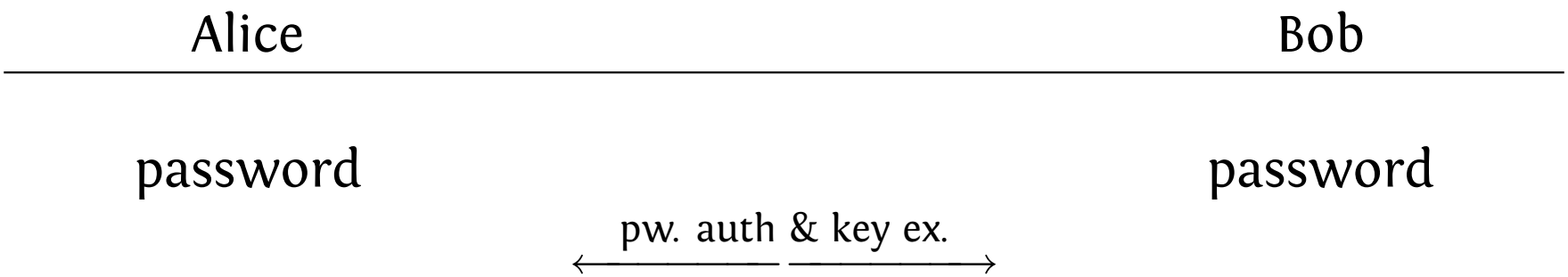
Password-authenticated key exchange:

Alice	Bob
password	password

Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

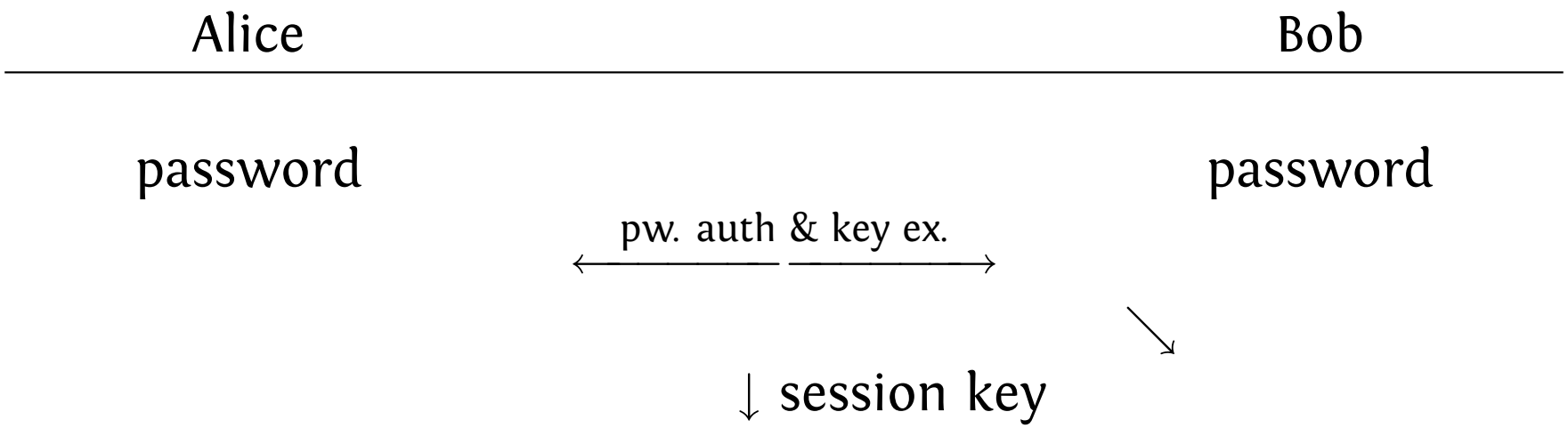
Password-authenticated key exchange:



Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

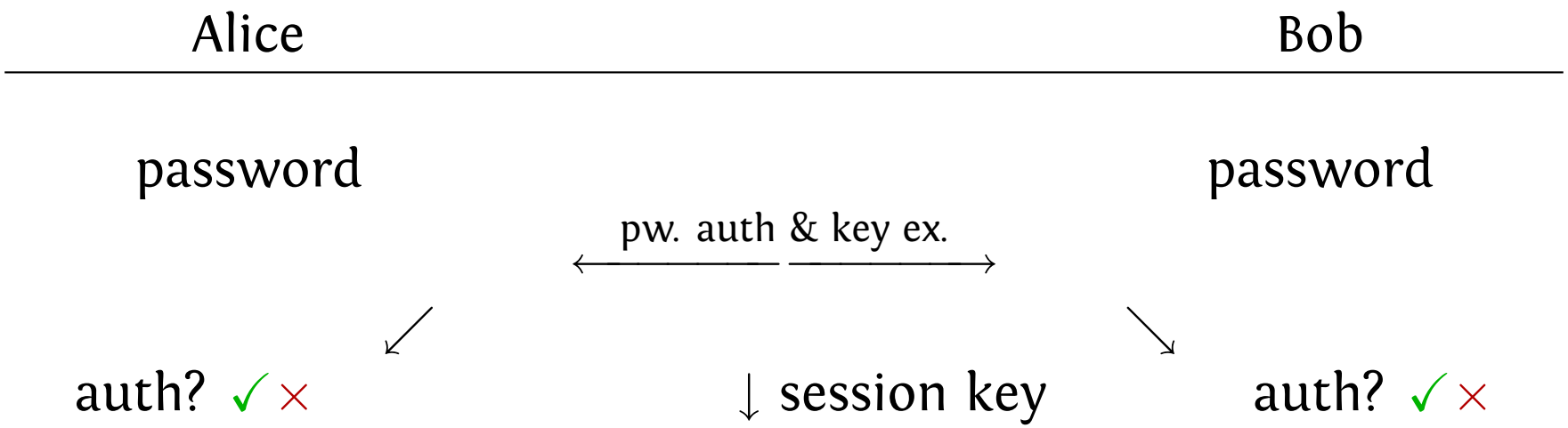
Password-authenticated key exchange:



Password-authenticated key exchange

Server and client prove to each other that they know the password without disclosing any useful information about the password; they also get a shared secret out at the end.

Password-authenticated key exchange:



Multi-factor authentication

Use two passwords:

1. long-term, unchanging password
2. short-term, changing password

This technique is being adopted by

- banks,
- corporations (for remote access),
- government

Multi-factor authentication

Use two passwords:

1. long-term, unchanging password: memorize
2. short-term, changing password: use an electronic password token or sheet of paper

Multi-factor authentication

People are already using multi-factor authentication but are using it insecurely.

Basic principles:

1. Strong client-to-server multi-factor authentication.
2. Strong server-to-client multi-factor authentication.
3. Authentication secrets should never be directly divulged.
4. Secure against offline dictionary attacks.
5. The protocol should remain secure as long as at least one of the factors is uncompromised.
6. Authentication and confidentiality should be tied intertwined.

Multi-factor authentication

People are already using multi-factor authentication but are using it insecurely.

Basic principles:

1. Strong client-to-server multi-factor authentication.
2. Strong server-to-client multi-factor authentication.
3. Authentication secrets should never be directly divulged.
4. Secure against offline dictionary attacks.
5. The protocol should remain secure as long as at least one of the factors is uncompromised.
6. Authentication and confidentiality should be tied intertwined.

Multi-factor authentication

People are already using multi-factor authentication but are using it insecurely.

Basic principles:

1. Strong client-to-server multi-factor authentication.
2. Strong server-to-client multi-factor authentication.
3. Authentication secrets should never be directly divulged.
4. Secure against offline dictionary attacks.
5. The protocol should remain secure as long as at least one of the factors is uncompromised.
6. Authentication and confidentiality should be tied intertwined.

MFPAK

We use techniques like in password-authenticated key exchange protocols to combine multiple factors securely and provide strong, multi-factor mutual authentication.

There are two separate stages to our protocol:

1. **User registration stage**: the user sets up her password with the server. This happens once, over a secure, authentic, out-of-band channel.
2. **Login stage**: a user attempts to login to a server.

Types of factors

Asymmetric / verifier-based:

- Server stores transformation of password called **verifier**.
- Provides protection against server database compromise – the compromised data can't immediately be used to impersonate a user.
- Can only be changed by the user – suitable for long-term passwords.
- Requires more computationally expensive protocols.

Symmetric / non-verifier-based:

- Client and server both store the password.
- Can be changed more easily – suitable for one-time passwords.

MFPAK User Registration		
	Client C	Server S
1.	$\mathsf{pw}_{C,S} \in_R$ Passwords	
2.	$\mathsf{re}_{C,S} \in_R$ Responses	
3.	$(V, W) \leftarrow \mathsf{Gen}(1^\kappa)$	
4.	$\gamma' = (H_1(C, S, \mathsf{pw}_{C,S}))^{-1}$	
5.	$V' = H_2(C, S, \mathsf{pw}_{C,S}) \oplus V$	
6.	$V'' = H_3(V)$	
7.	$\tau' = (H_4(C, S, \mathsf{re}_{C,S}))^{-1}$	
8.		$\xrightarrow{C,\gamma',W,V',V'',\tau'}$
9.		Store $\mathsf{pw}_S[C] = \langle \gamma', W, V', V'' \rangle$
10.		Store $\mathsf{re}_S[C] = \tau'$

MFPAK User Registration

Client C		Server S
1. $\mathsf{pw}_{C,S} \in_R$ Passwords		
2. $\mathsf{re}_{C,S} \in_R$ Responses		
3. $(W, V) = G_1(\mathsf{pw}_{C,S})$		
4. $\gamma' = (H_1(C, S, \mathsf{pw}_{C,S}))^{-1}$		
5. $V'' = H_2(C, S, \mathsf{pw}_{C,S}) \oplus V$		
6. $V' = H_3(V)$		
7. $\tau' = (H_4(C, S, \mathsf{re}_{C,S}))^{-1}$		
8.	$\xrightarrow{C,\gamma',W,V',V'',\tau'}$	
9.		Store $\mathsf{pw}_S[C] = \langle \gamma', W, V', V'' \rangle$
10.		Store $\mathsf{re}_S[C] = \tau'$

Client only stores long-term & short-term passwords

MFPAK User Registration

Client C		Server S
1. $\mathsf{pw}_{C,S} \in_R$ Passwords		
2. $\mathsf{re}_{C,S} \in_R$ Responses		
3. $(W, V) = \mathsf{Enc}(H_1(C, S, \mathsf{pw}_{C,S}))$		
4. $\gamma' = (H_1(C, S, \mathsf{pw}_{C,S}))^{-1}$		
5. $V' = H_2(C, S, \mathsf{pw}_{C,S}) \oplus V$		
6. $V'' = H_3(V)$		
7. $\tau' = (H_4(C, S, \mathsf{re}_{C,S}))^{-1}$		
8.	$\xrightarrow{C,\gamma',W,V',V'',\tau'}$	
9.		Store $\mathsf{pw}_S[C] = \langle \gamma', W, V', V'' \rangle$
10.		Store $\mathsf{re}_S[C] = \tau'$

Client only stores long-term & short-term passwords

Server never gets long-term passwords

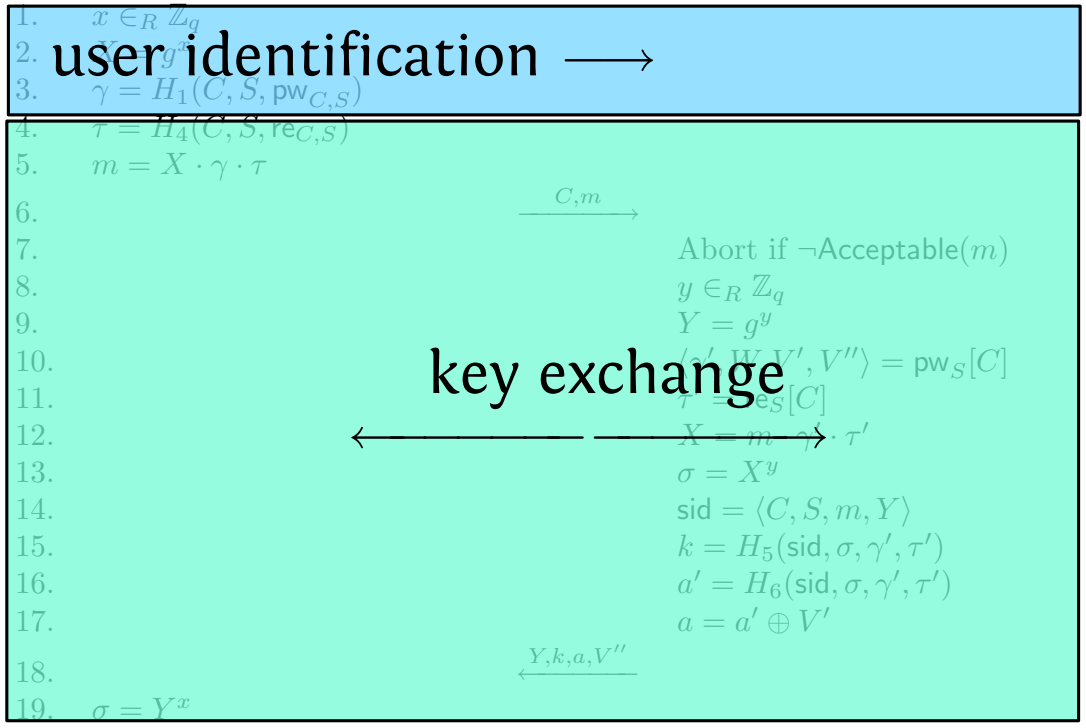
MFPAK Login			
	Client C		Server S
1.	$x \in_R \mathbb{Z}_q$		
2.	$X = g^x$		
3.	$\gamma = H_1(C, S, \mathsf{pw}_{C,S})$		
4.	$\tau = H_4(C, S, \mathsf{re}_{C,S})$		
5.	$m = X \cdot \gamma \cdot \tau$		
6.		$\xrightarrow{C,m}$	
7.			Abort if $\neg\mathsf{Acceptable}(m)$
8.			$y \in_R \mathbb{Z}_q$
9.			$Y = g^y$
10.			$\langle \gamma', W, V', V'' \rangle = \mathsf{pw}_S[C]$
11.			$\tau' = \mathsf{re}_S[C]$
12.			$X = m \cdot \gamma' \cdot \tau'$
13.			$\sigma = X^y$
14.			$\mathsf{sid} = \langle C, S, m, Y \rangle$
15.			$k = H_5(\mathsf{sid}, \sigma, \gamma', \tau')$
16.			$a' = H_6(\mathsf{sid}, \sigma, \gamma', \tau')$
17.			$a = a' \oplus V'$
18.		$\xleftarrow{Y,k,a,V''}$	
19.	$\sigma = Y^x$		
20.	$\gamma' = \gamma^{-1}$		
21.	$\tau' = \tau^{-1}$		
22.	$\mathsf{sid} = \langle C, S, m, Y \rangle$		
23.	Abort if $k \neq H_5(\mathsf{sid}, \sigma, \gamma', \tau')$		
24.	$k' = H_7(\mathsf{sid}, \sigma, \gamma', \tau')$		
25.	$a' = H_6(\mathsf{sid}, \sigma, \gamma', \tau')$		
26.	$V' = a' \oplus a$		
27.	$V = H_2(C, S, \mathsf{pw}_{C,S}) \oplus V'$		
28.	Abort if $V'' \neq H_3(V)$		
29.	$s = \mathsf{Sign}_V(\mathsf{sid})$		
30.		$\xrightarrow{k',s}$	
31.			Abort if $k' \neq H_7(\mathsf{sid}, \sigma, \gamma', \tau')$
32.			Abort if $\neg\mathsf{Verify}_W(\mathsf{sid}, s)$
33.	$\mathsf{sk} = H_8(\mathsf{sid}, \sigma, \gamma', \tau')$		$\mathsf{sk} = H_8(\mathsf{sid}, \sigma, \gamma', \tau')$

MFPAK Login

	Client C		Server S
1.	$x \in_R \mathbb{Z}_q$		
2.	$X = g^x$		
3.	$\gamma = H_1(C, S, \mathsf{pw}_{C,S})$		
4.	$\tau = H_4(C, S, \mathsf{re}_{C,S})$		
5.	$m = X \cdot \gamma \cdot \tau$		
6.		$\xrightarrow{C,m}$	
7.			Abort if $\neg\mathsf{Acceptable}(m)$
8.			$y \in_R \mathbb{Z}_q$
9.			$Y = g^y$
10.			$\langle \gamma', W, V', V'' \rangle = \mathsf{pw}_S[C]$
11.			$\tau' = \mathsf{re}_S[C]$
12.			$X = m \cdot \gamma' \cdot \tau'$
13.			$\sigma = X^y$
14.			$\mathsf{sid} = \langle C, S, m, Y \rangle$
15.			$k = H_5(\mathsf{sid}, \sigma, \gamma', \tau')$
16.			$a' = H_6(\mathsf{sid}, \sigma, \gamma', \tau')$
17.			$a = a' \oplus V'$
18.		$\xleftarrow{Y,k,a,V''}$	
19.	$\sigma = Y^x$		
20.	$\gamma' = \gamma^{-1}$		
21.	$\tau' = \tau^{-1}$		
22.	$\mathsf{sid} = \langle C, S, m, Y \rangle$		
23.	Abort if $k \neq H_5(\mathsf{sid}, \sigma, \gamma', \tau')$		
24.	$k' = H_7(\mathsf{sid}, \sigma, \gamma', \tau')$		
25.	$a' = H_6(\mathsf{sid}, \sigma, \gamma', \tau')$		
26.	$V' = a' \oplus a$		
27.	$V = H_2(C, S, \mathsf{pw}_{C,S}) \oplus V'$		
28.	Abort if $V'' \neq H_3(V)$		
29.	$s = \mathsf{Sign}_V(\mathsf{sid})$		
30.		$\xrightarrow{k',s}$	
31.			Abort if $k' \neq H_7(\mathsf{sid}, \sigma, \gamma', \tau')$
32.			Abort if $\neg\mathsf{Verify}_W(\mathsf{sid}, s)$
33.	$\mathsf{sk} = H_8(\mathsf{sid}, \sigma, \gamma', \tau')$		$\mathsf{sk} = H_8(\mathsf{sid}, \sigma, \gamma', \tau')$

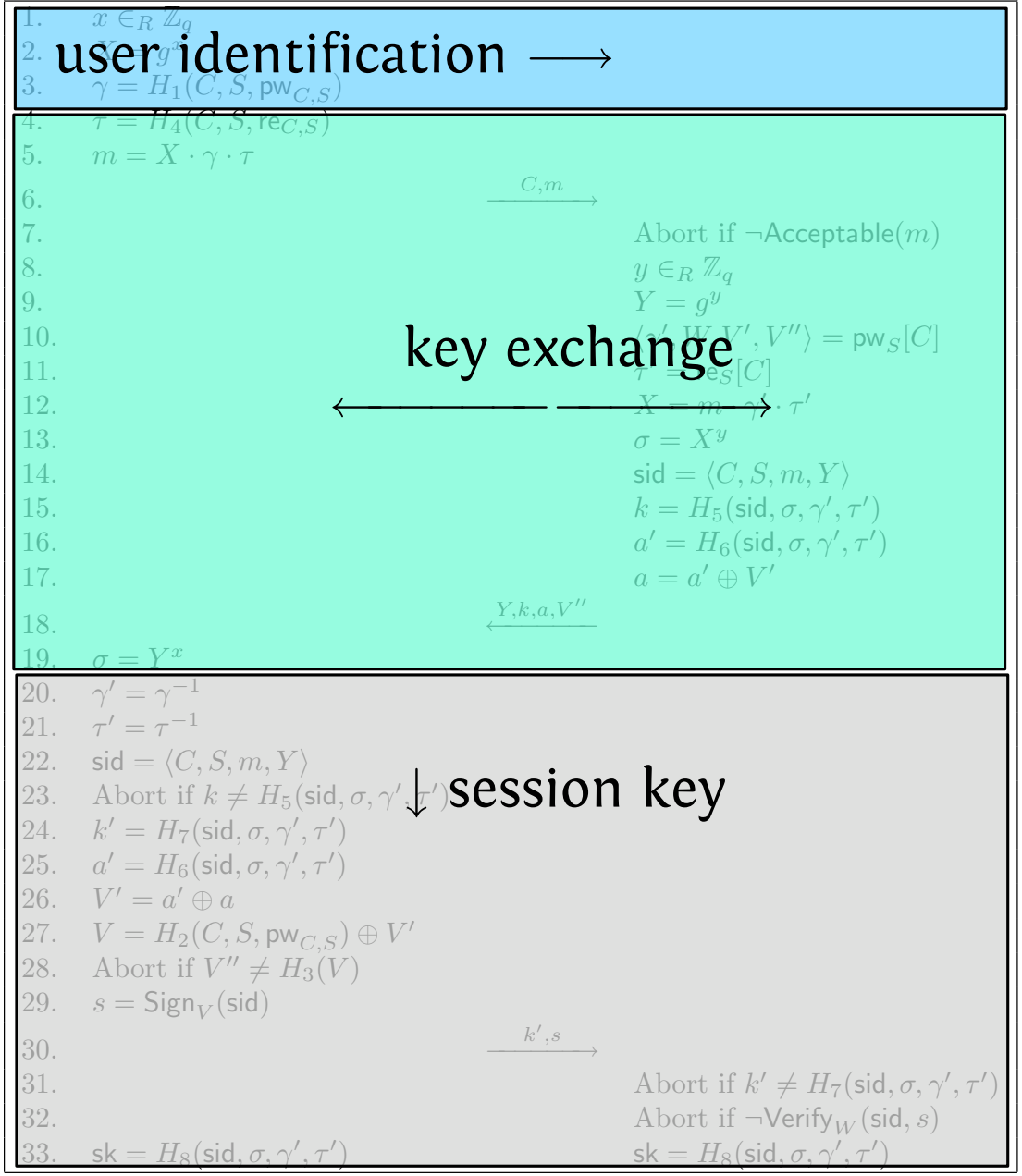
MFPAK Login

	Client C		Server S
1.	$x \in_R \mathbb{Z}_q$		
2.	$X = g^x$		
3.	$\gamma = H_1(C, S, \mathsf{pw}_{C,S})$		
4.	$\tau = H_4(C, S, \mathsf{re}_{C,S})$		
5.	$m = X \cdot \gamma \cdot \tau$		
6.		$\xrightarrow{C,m}$	
7.			Abort if $\neg\mathsf{Acceptable}(m)$
8.			$y \in_R \mathbb{Z}_q$
9.			$Y = g^y$
10.			$\langle \gamma', W, V', V'' \rangle = \mathsf{pw}_S[C]$
11.			$\tau' = \mathsf{re}_S[C]$
12.			$X = m \cdot \gamma' \cdot \tau'$
13.			$\sigma = X^y$
14.			$\mathsf{sid} = \langle C, S, m, Y \rangle$
15.			$k = H_5(\mathsf{sid}, \sigma, \gamma', \tau')$
16.			$a' = H_6(\mathsf{sid}, \sigma, \gamma', \tau')$
17.			$a = a' \oplus V'$
18.		$\xleftarrow{Y,k,a,V''}$	
19.	$\sigma = Y^x$		
20.	$\gamma' = \gamma^{-1}$		
21.	$\tau' = \tau^{-1}$		
22.	$\mathsf{sid} = \langle C, S, m, Y \rangle$		
23.	Abort if $k \neq H_5(\mathsf{sid}, \sigma, \gamma', \tau')$		
24.	$k' = H_7(\mathsf{sid}, \sigma, \gamma', \tau')$		
25.	$a' = H_6(\mathsf{sid}, \sigma, \gamma', \tau')$		
26.	$V' = a' \oplus a$		
27.	$V = H_2(C, S, \mathsf{pw}_{C,S}) \oplus V'$		
28.	Abort if $V'' \neq H_3(V)$		
29.	$s = \mathsf{Sign}_V(\mathsf{sid})$		
30.		$\xrightarrow{k',s}$	
31.			Abort if $k' \neq H_7(\mathsf{sid}, \sigma, \gamma', \tau')$
32.			Abort if $\neg\mathsf{Verify}_W(\mathsf{sid}, s)$
33.	$\mathsf{sk} = H_8(\mathsf{sid}, \sigma, \gamma', \tau')$		$\mathsf{sk} = H_8(\mathsf{sid}, \sigma, \gamma', \tau')$



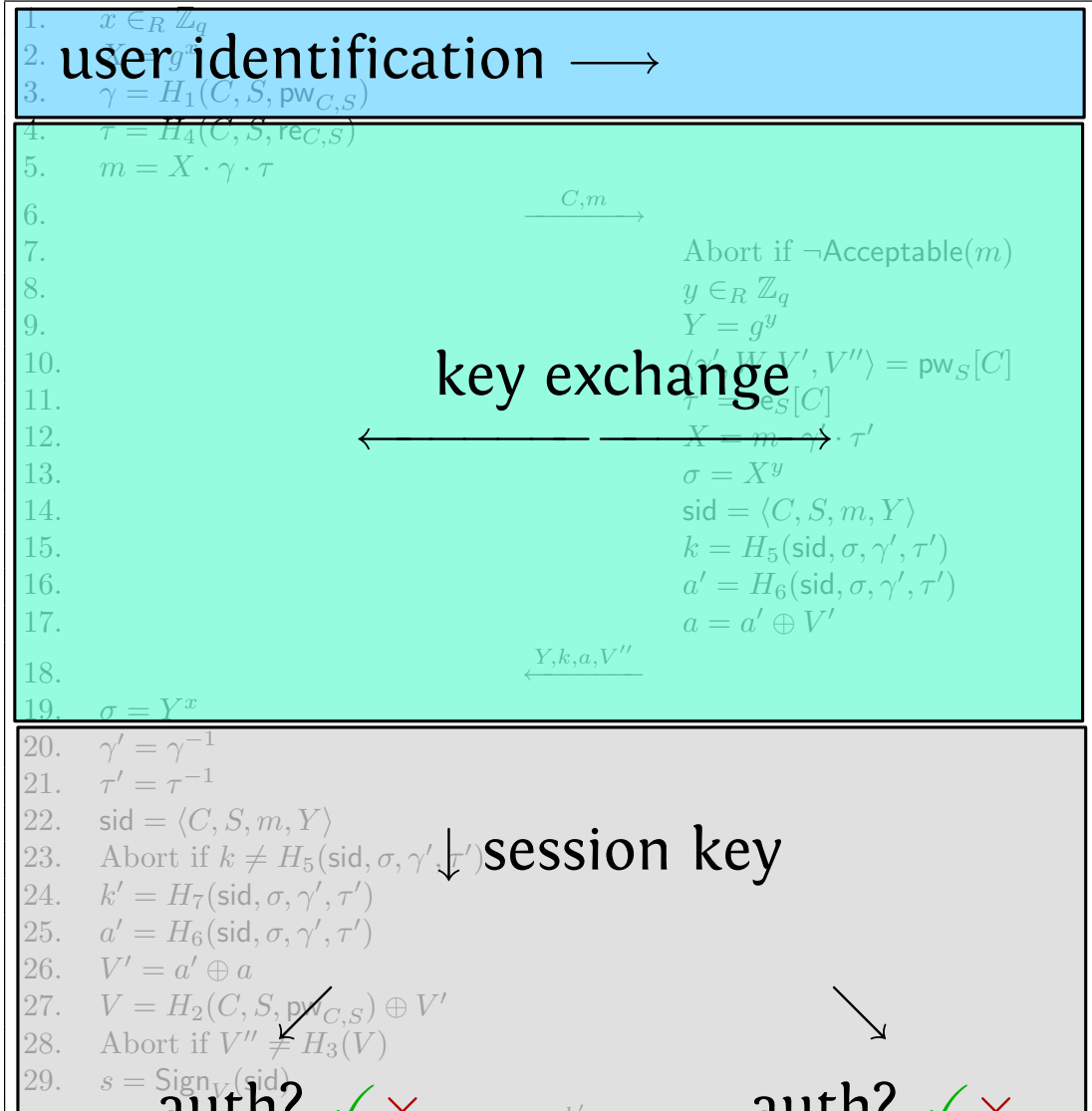
MFPAK Login

	Client C		Server S
1.	$x \in_R \mathbb{Z}_q$		
2.	$X = g^x$		
3.	$\gamma = H_1(C, S, \mathsf{pw}_{C,S})$		
4.	$\tau = H_4(C, S, \mathsf{re}_{C,S})$		
5.	$m = X \cdot \gamma \cdot \tau$		
6.		$\xrightarrow{C,m}$	
7.			Abort if $\neg\mathsf{Acceptable}(m)$
8.			$y \in_R \mathbb{Z}_q$
9.			$Y = g^y$
10.			$\langle \gamma', W, V', V'' \rangle = \mathsf{pw}_S[C]$
11.			$\tau' = \mathsf{re}_S[C]$
12.			$X = m \cdot \gamma' \cdot \tau'$
13.			$\sigma = X^y$
14.			$\mathsf{sid} = \langle C, S, m, Y \rangle$
15.			$k = H_5(\mathsf{sid}, \sigma, \gamma', \tau')$
16.			$a' = H_6(\mathsf{sid}, \sigma, \gamma', \tau')$
17.			$a = a' \oplus V'$
18.		$\xleftarrow{Y,k,a,V''}$	
19.	$\sigma = Y^x$		
20.	$\gamma' = \gamma^{-1}$		
21.	$\tau' = \tau^{-1}$		
22.	$\mathsf{sid} = \langle C, S, m, Y \rangle$		
23.	Abort if $k \neq H_5(\mathsf{sid}, \sigma, \gamma', \tau')$		
24.	$k' = H_7(\mathsf{sid}, \sigma, \gamma', \tau')$		
25.	$a' = H_6(\mathsf{sid}, \sigma, \gamma', \tau')$		
26.	$V' = a' \oplus a$		
27.	$V = H_2(C, S, \mathsf{pw}_{C,S}) \oplus V'$		
28.	Abort if $V'' \neq H_3(V)$		
29.	$s = \mathsf{Sign}_V(\mathsf{sid})$		
30.		$\xrightarrow{k',s}$	
31.			Abort if $k' \neq H_7(\mathsf{sid}, \sigma, \gamma', \tau')$
32.			Abort if $\neg\mathsf{Verify}_W(\mathsf{sid}, s)$
33.	$\mathsf{sk} = H_8(\mathsf{sid}, \sigma, \gamma', \tau')$		$\mathsf{sk} = H_8(\mathsf{sid}, \sigma, \gamma', \tau')$



MFPAK Login

Client C		Server S
$x \in_R \mathbb{Z}_q$		
$X = g^x$		
$\gamma = H_1(C, S, \mathsf{pw}_{C,S})$		
$\tau = H_4(C, S, \mathsf{re}_{C,S})$		
$m = X \cdot \gamma \cdot \tau$		
	$\xrightarrow{C,m}$	
		Abort if $\neg\mathsf{Acceptable}(m)$
		$y \in_R \mathbb{Z}_q$
		$Y = g^y$
		$\langle a', W, V', V'' \rangle = \mathsf{pw}_S[C]$
		$\tau' = \mathsf{re}_S[C]$
		$X = m \cdot \gamma' \cdot \tau'$
		$\sigma = X^y$
		$\mathsf{sid} = \langle C, S, m, Y \rangle$
		$k = H_5(\mathsf{sid}, \sigma, \gamma', \tau')$
		$a' = H_6(\mathsf{sid}, \sigma, \gamma', \tau')$
		$a = a' \oplus V'$
	$\xleftarrow{Y,k,a,V''}$	
$\sigma = Y^x$		
$\gamma' = \gamma^{-1}$		
$\tau' = \tau^{-1}$		
$\mathsf{sid} = \langle C, S, m, Y \rangle$		
Abort if $k \neq H_5(\mathsf{sid}, \sigma, \gamma', \tau')$		
$k' = H_7(\mathsf{sid}, \sigma, \gamma', \tau')$		
$a' = H_6(\mathsf{sid}, \sigma, \gamma', \tau')$		
$V' = a' \oplus a$		
$V = H_2(C, S, \mathsf{pw}_{C,S}) \oplus V'$		
Abort if $V'' \neq H_3(V)$		
$s = \mathsf{Sign}_V(\mathsf{sid})$		
	$\xrightarrow{k',s}$	
		Abort if $k' \neq H_7(\mathsf{sid}, \sigma, \gamma', \tau')$
		Abort if $\neg\mathsf{Verify}_W(\mathsf{sid}, s)$
$\mathsf{sk} = H_8(\mathsf{sid}, \sigma, \gamma', \tau')$		$\mathsf{sk} = H_8(\mathsf{sid}, \sigma, \gamma', \tau')$



Efficiency

Operation	PAK & PAK-Z+		MFPAK	
	Client	Server	Client	Server
exponentiations	$2\|I_s\| + 2\|I_a\|$	$2\|I_s\| + 2\|I_a\|$	2	2
signature generation	$\|I_a\|$	0	$\|I_a\|$	0
signature verification	0	$\|I_a\|$	0	$\|I_a\|$
total	$2\|I_s\| + 3\|I_a\|$	$2\|I_s\| + 3\|I_a\|$	$2 + \|I_a\|$	$2 + \|I_a\|$

$|I_s|$: # of symmetric factors
$|I_a|$: # of asymmetric factors

Formally modelling security

To show a protocol secure, we:

1. Model the powers of an adversary.
2. Define a game that the adversary has to win in order to break security.
3. Show upper bounds on the probability that an adversary can win the game (possibly related to hard computational problems).

Formal models: be suspicious!

Formal security arguments ("security proofs", "provable security") do not always mean a protocol is secure in practice.

- Does the model capture all possible forms of attack?
- Is the proof correct?
- Is the underlying "hard" computational problem actually hard?
- Are the parameter sizes appropriate given the proof?
- Does the implementation have flaws?

But formal security arguments can still be a good heuristic that the design of the protocol is sound.

Powers of the adversary

The adversary has complete control of the communication links and can direct participants to perform certain actions.

The adversary can:

- modify, reorder, or delete protocol messages
- send protocol messages
- direct participants to perform certain actions
- compromise certain secrets

Goal of the adversary

The adversary has two goals:

1. Break confidentiality:
   determine the session key of any “fresh” session.
2. Break authentication:
   impersonate one party in any “fresh” session.

Security ingredients

Our security is based on three assumptions:

1. group where Computational Diffie-Hellman is hard
   e.g., integers modulo a prime, elliptic curve groups
2. good hash functions (random oracle model)
3. secure digital signature scheme
   e.g., RSA-OAEP, DSA, ECDSA
   (but not long-term certificates / private keys)

Session key security

We show that, for an adversary $\mathcal{A}$ running in time t and making at most q queries,

$$\Pr(\mathcal{A} \text{ can break session key}) \leq O(q)\left(\frac{1}{\#\text{Passwords}}\right) + \epsilon$$

where

$$\epsilon \approx O\left(q^3 \Pr(\mathcal{A} \text{ can solve CDH})\right) \quad .$$

Under the CDH assumption, ϵ is small.

With a 450-bit elliptic curve and 9-character passwords, an adversary running in time 2^{80} can succeed with advantage at most 2^{-25}.

2^{80} operations: 1 million computers with 2 GHz CPUs running for 15,000 years

Authentication

The adversary's goal with respect to authentication is to cause one party $\mathcal{A}$ to accept authentication with $\mathcal{B}$ but for $\mathcal{B}$ to not have completed his session.

We can show similar bounds on the ability of an adversary to break authentication.

Security properties

By using a model like this, the results above imply a variety of desirable security properties:

- man-in-the-middle attacks are prevented
- offline dictionary attacks are prevented
- unknown key share attacks are prevented
- impersonation attacks are prevented
- forward secrecy

Future directions

Integration with SSL/TLS and other protocols

- This could provide stronger authentication in web browsers.
- Challenge: MFPAK doesn't fit within the message flow of TLS, so we need to find creative ways around that.

Testing usability of authentication protocols

Multi-factor password-authenticated key exchange

provides

- strong, mutual multi-factor authentication (client-to-server and server-to-client) and
- confidentiality

even in the face of

- spyware and
- phishing

and has formal security arguments.