Interactive Diffie-Hellman Assumptions With Applications to Password-Based Authentication

Michel Abdalla and David Pointcheval

Ecole normale supérieure, Paris, France

Financial Cryptography – February 2005

Outline

Outline

Key Exchange

Goal

Two parties want to agree on a common secret key sk, in order to establish a private/authenticated channel.

Example (Diffie-Hellman)

- Alice sends $X = g^x$ to Bob
- Bob sends $Y = g^y$ to Alice
- They can both compute $sk = X^y = Y^x = g^{xy}$

Man-in-the-middle attack

- Charlie can sit in between Alice and Bob
- He impersonates Alice to Bob, and Bob to Alice

Authentication is required!

Authentication

Asymmetric Authentication

Flows can be signed

Symmetric Authentication

Entropy

- high-entropy secret: Message Authentication Codes
- low-entropy secret: Password

Shared secrets

- 2-party: the secret is shared by Alice and Bob
- 3-party: the secrets are shared between the users and an authentication server

Adversaries

Passive Adversary

Eavesdrops all the network: transcripts and bad uses of the keys

Model

- *Execute*-queries: transcript of an execution of the protocol
- *Reveal*-queries: key agreed on by the players

Active Adversary

Controls all the network: intercepts, forwards, forges messages

Model

- *Send[Client/Server]*-queries: it sends any message of its choice to any player, who answers according to the protocol

Semantic Security

Ability of the Adversary

The adversary is able to distinguishes the actual session key from a random one

Model

- *Test*-query: it tests one session key, and receives either the actual key sk if $(b = 0)$, or a random key if $(b = 1)$.
- The adversary ends the game by answering its guess b'
- It wins if $b' = b$

Security

$\mathrm{Adv}(A) = 2Pr[b' = b] - 1$ must be negligible.

Dictionary Attacks

Password: low-entropy

4 digits: exhaustive search is possible!

Basic attack: on-line exhaustive search

1. choose a password and try it
2. in case of failure, erase it from the list

$\Longrightarrow$ 5000 trials are enough: cannot be avoided!

Dictionary attack: off-line exhaustive search

1. play a few active attacks
2. eavesdrop many transcripts

$\Longrightarrow$ find the good password: should be prevented

Encrypted Key Exchange

Example

A Diffie-Hellman key exchange encrypted by the password

EKE

- Alice computes $X = g^x$ and sends $X' = \mathcal{E}_{\mathrm{pw}}(X)$ to Bob
- Bob computes $Y = g^y$ and sends $Y' = \mathcal{E}_{\mathrm{pw}}(Y)$ to Alice
- They can both compute $sk = H(K)$, where $X^y = Y^x = g^{xy}$

Security

- Security against passive attacks: under the **CDH** problem
- Security against active attacks:

$$\mathrm{Adv}(t) \leq \frac{2q_s}{N} + \mathcal{O}(\mathrm{Succ}^{\mathrm{cdh}}(t)) + \epsilon.$$

Outline

An Example

Scenario

- Alice shares a password pw_A with the server
- and Bob shares a password pw_B with the server
- Alice and Bob want to establish a secure channel

3-GPAKE-Weak

Alice		*Server*		*Bob*
sk_A	$\xleftrightarrow{EKE(\mathrm{pw}_A)}$	$sk_A \quad sk_B$	$\xleftrightarrow{EKE(\mathrm{pw}_B)}$	sk_B
sk	$\xleftarrow{\mathcal{E}_{sk_A}(sk)}$	sk	$\xrightarrow{\mathcal{E}_{sk_B}(sk)}$	sk

Key-Privacy

The server knows the key sk distributed to Alice and Bob

Key Privacy

Security Model

If Alice and Bob *indeed* agree on a key, it is hidden to the server

3-GPAKE

Alice		*Server*		*Bob*
sk_A	$\xleftrightarrow{EKE}$	$sk_A \quad sk_B$	$\xleftrightarrow{EKE}$	sk_B
sk	$\xleftarrow{\mathcal{E}_{sk_A}(sk)}$	sk	$\xrightarrow{\mathcal{E}_{sk_B}(sk)}$	sk
SK	$\xleftrightarrow{MAC-based \quad AKE}$			SK

Efficiency

This protocol requires 4 exponentiations per player

A First Scheme

Basic EKE

Alice		*Server*		*Bob*
$PW_A = G(pw_A)$ $x, X = g^x$		$PW_U = G(pw_U)$ r		$PW_B = G(pw_B)$ $y, Y = g^y$
$X' = X \cdot PW_A$	$\xrightarrow{X'}$	$X = X'/PW_A$		
		$X^\star = X \cdot PW_B$	$\xrightarrow{X^\star}$	
		$Y = Y'/PW_B$	$\xleftarrow{Y'}$	$Y' = Y \cdot PW_B$
	$\xleftarrow{Y^\star}$	$Y^\star = Y \cdot PW_A$		
$Y = Y^\star/PW_A$ $K_A = Y^x = g^{xy}$		$sk = H(g^{xy})$		$X = X^\star/PW_B$ $K_B = X^y = g^{xy}$

Efficiency

This protocol requires only 2 exponentiations per player

Insider Attack

Insider Adversary

Bob may try to learn Alice's password

Example

Alice		*Server*		*Bob*
		$Y = Y'/PW_B$	$\xleftarrow{Y'}$	$Y' = Y \cdot PW_B$
	$\xleftarrow{Y^\star}$	$Y^\star = Y \cdot PW_A$		

Attack

- From Y' and $Y^\star$: One immediately gets PW_A/PW_B
- From Y and $Y^\star$: Bob immediately gets PW_A

A Second Scheme

Randomized EKE

Alice		*Server*		*Bob*
$x, X = g^x$		r		$y, Y = g^y$
$X' = X \cdot PW_A$	$\xrightarrow{X'}$	$X = X'/PW_A$ $\bar{X} = X^r$		
		$X^\star = \bar{X} \cdot PW_B$	$\xrightarrow{X^\star}$	
		$Y = Y'/PW_B$ $\bar{Y} = Y^r$	$\xleftarrow{Y'}$	$Y' = Y \cdot PW_B$
	$\xleftarrow{Y^\star}$	$Y^\star = \bar{Y} \cdot PW_A$		
$\bar{Y} = Y^\star/PW_A$ $K_A = \bar{Y}^x = g^{xyr}$		$sk = H(g^{xyr})$		$\bar{X} = X^\star/PW_B$ $K_B = \bar{X}^y = g^{xyr}$

Security Proof Problem

With a fixed and unique password PW_A: no security proof

Outline

The New Scheme

Our Scheme

Randomized EKE with Variable Passwords

Alice		Server		Bob
$x, X = g^x$		r		$y, Y = g^y$
$X' = X \cdot \mathrm{PW}_A$	$\xrightarrow{X'}$		$\xleftarrow{Y'}$	$Y' = Y \cdot \mathrm{PW}_B$
		$X = X'/\mathrm{PW}_A$ $\bar{X} = X^r$ $X^\star = \bar{X} \cdot \mathrm{PW}_B(Y')$	$\xrightarrow{X^\star}$	
	$\xleftarrow{Y^\star}$	$Y = Y'/\mathrm{PW}_B$ $\bar{Y} = Y^r$ $Y^\star = \bar{Y} \cdot \mathrm{PW}_A(X')$		
$\bar{Y} = Y^\star/\mathrm{PW}_A(X')$ $K_A = \bar{Y}^x = g^{xyr}$		$sk = H(g^{xyr})$		$\bar{X} = X^\star/\mathrm{PW}_B(Y')$ $K_B = \bar{X}^y = g^{xyr}$

Example

$$\mathrm{PW}_A = G(\mathrm{pw}_A) \qquad \mathrm{PW}_A(X') = G(\mathrm{pw}_A, X')$$

The New Scheme

Properties

Efficiency

This protocol requires only two exponentiations per player

Scenario

In the three-party setting, but for non-concurrent executions

Security

In the random-oracle model:

$$\mathrm{Adv}(t) \leq \frac{2q_s}{N} + q_e \times \mathrm{Adv}^{\mathrm{ddh}}(t) + \mathrm{poly}(Q) \times \mathrm{Succ}^{\mathrm{cdh}}(t) + 2q_s \times \mathrm{Adv}^{\mathrm{pcddh}}(t) + \epsilon$$

Computational Assumptions

The Chosen-Basis Diffie-Hellman Problem

Formal Definition: CDDH(U, V)

- A outputs X and Y
- One chooses two random exponents r_0 and r_1, as well as two random bits b and b_0, and sets $b_1 = b \oplus b$
- One sets $Y' = Y^{r_0}$ and $X_0 = (X/U)^{r_{b_0}}$, $X_1 = (X/V)^{r_{b_1}}$
- A is given Y' and X_0, X_1, it outputs b' (its guess for b)

Idea

Given U and V, no adversary can find X and Y so that given Y^r, it can compute $\mathbf{CDH}_Y(X/U, Y^r)$ and $\mathbf{CDH}_Y(X/V, Y^r)$

Either he can compute the former, with $X = Y^\alpha U$, or the latter, with $X = Y^\alpha V$.

Summary

Summary

A new password-based key exchange protocol

- in the three-party setting
- twice as much efficient as the generic scheme
- provably secure in the random-oracle model

New computational assumptions

- Chosen-basis Diffie-Hellman problems
  - intuitively hard to solve
- Password-Based Chosen-basis Diffie-Hellman problems
  - formally related to the above ones
  - used in the security analysis