One-Time Verifier-Based Encrypted Key Exchange

Michel Abdalla  
ENS – France

Olivier Chevassut  
LBNL – DOE - USA

David Pointcheval  
CNRS-ENS – France

PKC ’05  
Les Diablerets, Switzerland  
January 24th 2005

Summary

- Authenticated Key Exchange
- Password-Based Authentication
  - EKE and OKE
  - Security Results
- Enhanced Security against Corruption

Authenticated Key Exchange

Two parties (Alice and Bob) agree on a common secret key \( sk \), in order to establish a secret channel

- Basic security notion: semantic security
  - only the intended partners can compute the session key \( sk \)
- Formally:
  - the session key \( sk \) is indistinguishable from a random string \( r \), to anybody else

Further Properties

- Mutual authentication
  - They are both sure to actually share the secret with the people they think they do
- Forward-secrecy
  - Even if a long-term secret data is corrupted, previously shared secrets are still protected
Passive/Active Adversaries

- **Passive adversary**: history built using
  - the execute-queries → transcripts
  - the reveal-queries → session keys
- **Active adversary**: entire control of the network
  - the send-queries
    - active, adaptive adversary on concurrent executions
      - to send message to Alice or Bob (in place of Bob or Alice respectively)
      - to intercept, forward and/or modify messages

Semantic Security

As many execute, send and reveal queries as the adversary wants

0/1 try to guess b
- if b=0, answer the actual secret data sk
- if b=1, answer a random string r

Foward Secrecy: Corrupt-Query

**Forward Secrecy**: corruption of long term keys
- the corrupt-queries → long-term key

**FS-Freshness**:
- the instance has accepted (holds a key!)
- neither the instance nor its partner has been asked for a reveal query
- (neither the instance) nor its partner has been asked for a corrupt query

⇒ Diffie-Hellman provides the Forward Secrecy
**Diffie-Hellman Key Exchange**

\[ G = \langle g \rangle, \text{ cyclic group of prime order } q \]

- Alice chooses a random \( x \in \mathbb{Z}_q \), computes and sends \( X = g^x \)
- Bob chooses a random \( y \in \mathbb{Z}_q \), computes and sends \( Y = g^y \)
- They can both compute the value \( K = Y^x = X^y \)

**Properties**

- Without any authentication, no security is possible: man-in-the-middle attack
  \( \Rightarrow \) some authentication is required
- If flows are authenticated (MAC or Signature), it provides the forward secrecy under the DDH Problem
- If one derives the session key as \( sk = H(K, ...) \), in the random oracle model, the forward secrecy is relative to the CDH Problem

**Password-based Authentication**

Password (short – low-entropy secret – say 20 bits)

- exhaustive search is possible
- basic attack: on-line exhaustive search
  - the adversary guesses a password
  - tries to play the protocol with this guess
  - failure \( \Rightarrow \) it erases the password from the list
  - and restarts...
- after \( 2^{20} \) attempts, the adversary wins

**Dictionary Attack**

- The on-line exhaustive search
  - cannot be prevented
  - can be made less serious (delay, limitations, ...)
  We want it to be the best attack...
- The off-line exhaustive search
  - a few passive or active attacks
  - transcripts \( \Rightarrow \) password, by an off-line check
  - this is called dictionary attack
  \( \Rightarrow \) our GOAL: prevent dictionary attacks
**Example: EKE**

The most famous scheme: Encrypted Key Exchange
Either one or two flows are encrypted with the password

- $E_x$: ideal cipher
- $E_x(X) = H(\pi).X$ in ROM

**EKE - OKE**

- **OKE**: Open Key Exchange
  - first flow sent in clear (open)
  - forward secrecy = CDH

- **EKE**: Encrypted Key Exchange
  - both flows encrypted
  - semantic security = CDH

**EKE: Forward secrecy = open problem**

**EKE: Security Results**

- **Assumptions**
  - two different masks with $H_1$ and $H_2$
  - random-oracle model for $H, H_1,$ and $H_2$

**Semantic security of EKE**:
advantage $\leq 2q_s/N + 3q_h^2$ $\text{Succ}^\text{CDH}(t') + \epsilon$

**Forward Secrecy of EKE**:
advantage $\leq 2q_s/N + 4$ $\text{Succ}^\text{GDH}(t,q_h) + \epsilon$

$\text{Succ}^\text{GDH}(t,q) = \text{Probability to solve the CDH problem, within time } t,$ after $q$ calls to a DDH oracle
Increased Security

- **Protecting against server corruptions:** verifier-based authentication
  - Alice knows a password $\pi$,
  - Bob just knows a verifier of the password $\nu = f(\pi)$,
    - $\nu$ is the *actual* password,
    - then Alice proves her knowledge of $\pi = f^{-1}(\nu)$, in ZK

Improved Security (Con'd)

- **Protecting against client corruptions:** one-time password authentication
  - the *actual* password is $\nu_n = f^n(\pi)$
  - at the end the client sends,
    encrypted under the new session key, $\nu_{n-1} = f^{n-1}(\pi)$,
    which validity can be easily checked
  - the next password will be $\nu_{n-1}$