

New Security Results on Encrypted Key Exchange

Emmanuel Bresson
Olivier Chevassut
David Pointcheval

CELAR – France
LBNL – DOE – USA
CNRS-ENS – France

Summary

- Contributions of this talk
 - Encrypted Key Exchange – example
 - Security Results
- One-Mask Diffie-Hellman Scheme
 - Password-based Authentication
 - Security Model
 - Analysis of the Protocol
 - Properties – Denial of service
- Conclusion

Summary

- Contributions of this talk
 - Encrypted Key Exchange – example
 - Security Results
- One-Mask Diffie-Hellman Scheme
 - Password-based Authentication
 - Security Model
 - Analysis of the Protocol
 - Properties – Denial of service
- Conclusion

Key Exchange Schemes

- Alice and Bob agree on a common secret key sk , in order to establish a secret channel
- Intuitively: implicit authentication
 - only the intended partners can compute the session key
- Formally: semantic security
 - the session key sk is indistinguishable from a random string r , to anybody else

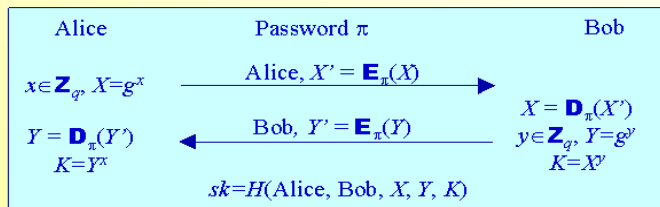
Example: Diffie-Hellman

- Diffie-Hellman Key Exchange
 - $G = \langle g \rangle$, cyclic group of prime order p
 - Alice chooses $x \in \mathbb{Z}_p$ and sends $X = g^x$
 - Bob chooses $y \in \mathbb{Z}_p$ and sends $Y = g^y$
 - Both can compute $K = g^{xy}$
- (Passive) Security under DDH Assumption
- No security against active adversaries
 - Authentication is needed

How Authentication is Done

- Asymmetric: (sk_A, pk_A) and possibly (sk_B, pk_B)
 - they authenticate to each other using the knowledge of the private key associated to the certified public key
- Symmetric: common (long / high-entropy) secret
 - they use the long term secret to derive a secure and authenticated ephemeral key sk
- Password: common (short / low-entropy) secret
 - let us assume a 20-bit password

EKE - AuthA

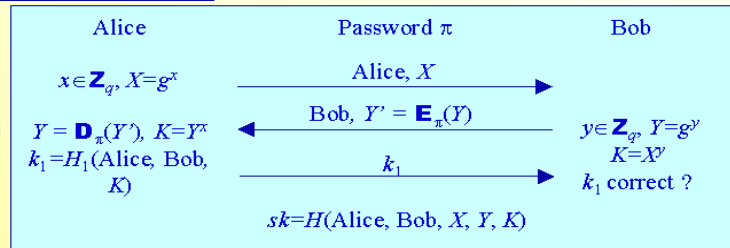


EKE

Bellovin-Merritt 1992
*Two-flow Encrypted
Key Exchange*

AuthA

Bellare-Rogaway 2000
*OEKE = One-flow
Encrypted
Key Exchange*



- Both schemes used an ``ideal cipher``

New Results

- Provable security is achieved for both EKE and AuthA
 - In the random oracle model only
 - Based on CDH assumption
- Which means...
 - Security against dictionary attacks
 - Semantic security of the session key
- Add Denial-of-Service protection

Summary

- Contributions of this talk
 - Encrypted Key Exchange – example
 - Security Results
- One-Mask Diffie-Hellman Scheme
 - Password-based Authentication
 - Security Model
 - Analysis of the Protocol
 - Properties – Denial of service
- Conclusion

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...

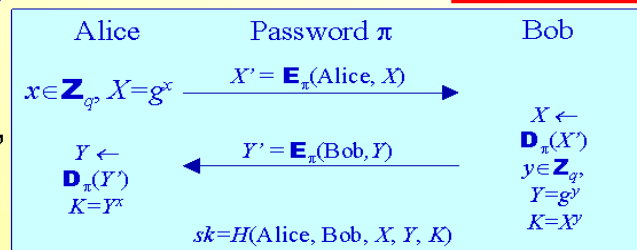
- The off-line exhaustive search
 - a few passive or active attacks
 - failure \Rightarrow erasure of MANY passwords from the list
 - this is called **dictionary attack**

Example: EKE

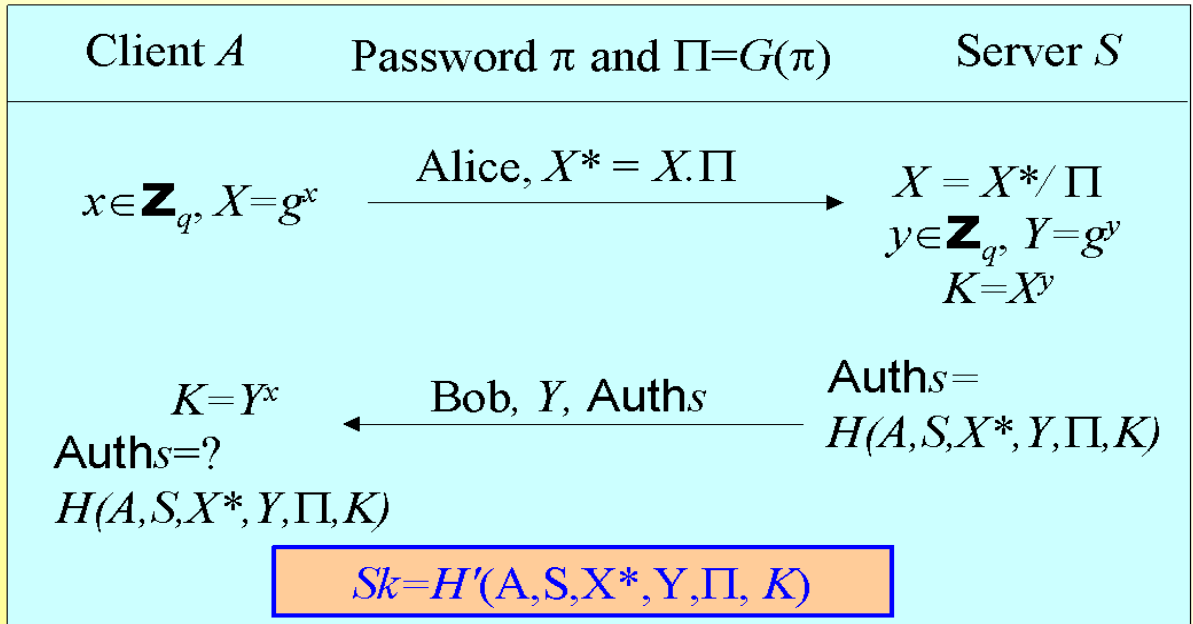
- The most famous scheme EKE:
 - Encrypted Key Exchange
- 2 flows are encrypted with the password.
- Must be done carefully: no redundancy
- For each password π

- decrypt X'
- check whether it begins with "Alice"

bad one !

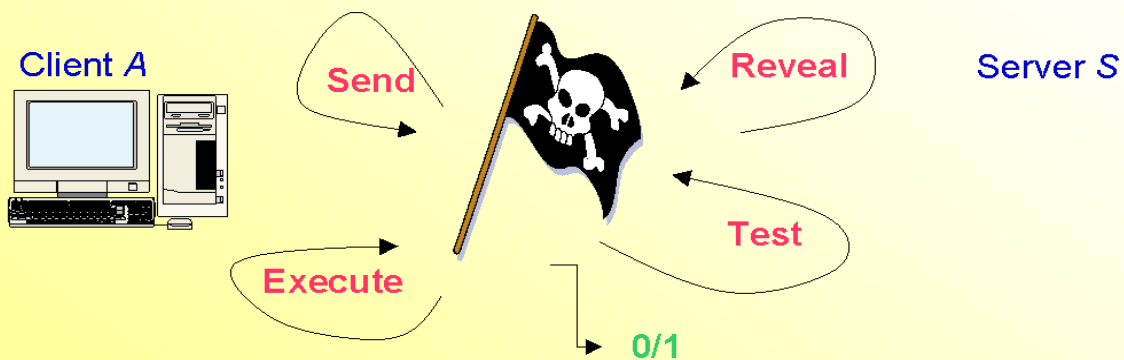


One-Mask Diffie-Hellman KE



Security Model

As many **Execute**, **Send** and **Reveal** queries as the adversary wants



But one **Test**-query, with b to be guessed...

Passive/Active Adversaries

- Passive adversary: history built using
 - the **Execute**-queries \Rightarrow transcripts
 - the **Reveal**-queries \Rightarrow session keys
 - must learn *no information* about password
- Active adversary: entire control of the network
 - the **Send**-queries \Rightarrow send arbitrary messages
 - a **Send**-query allows to erase at most one password from the list

Semantic Security

- For breaking the semantic security, the adversary asks one **Test**-query which is answered, according to a random bit b , by
 - the actual secret data sk (if $b=0$)
 - a random string r (if $b=1$)
- \Rightarrow the adversary has to guess this bit b

OMDHKE: New Security Result

Assumptions

- the random-oracle model – for G, H and H1

Notations

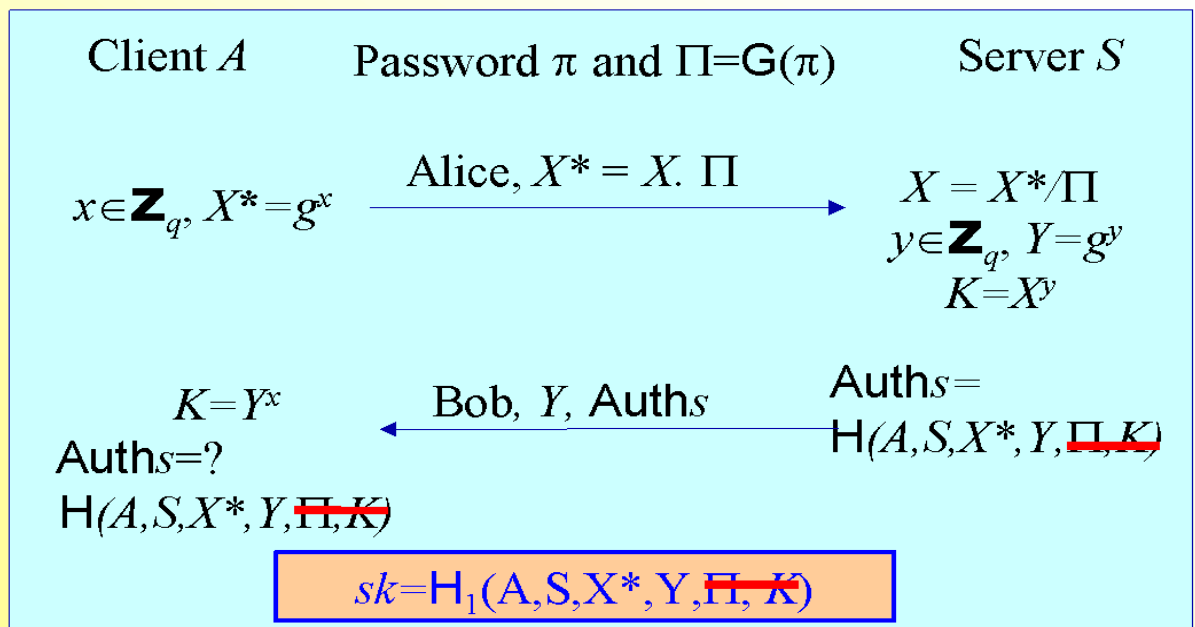
- q_s , the number of **Send**-queries (active and adaptive)
- q_h , the number of **Hash**-queries to G, H and H1
- N , the number of passwords

Semantic security of DHKE :

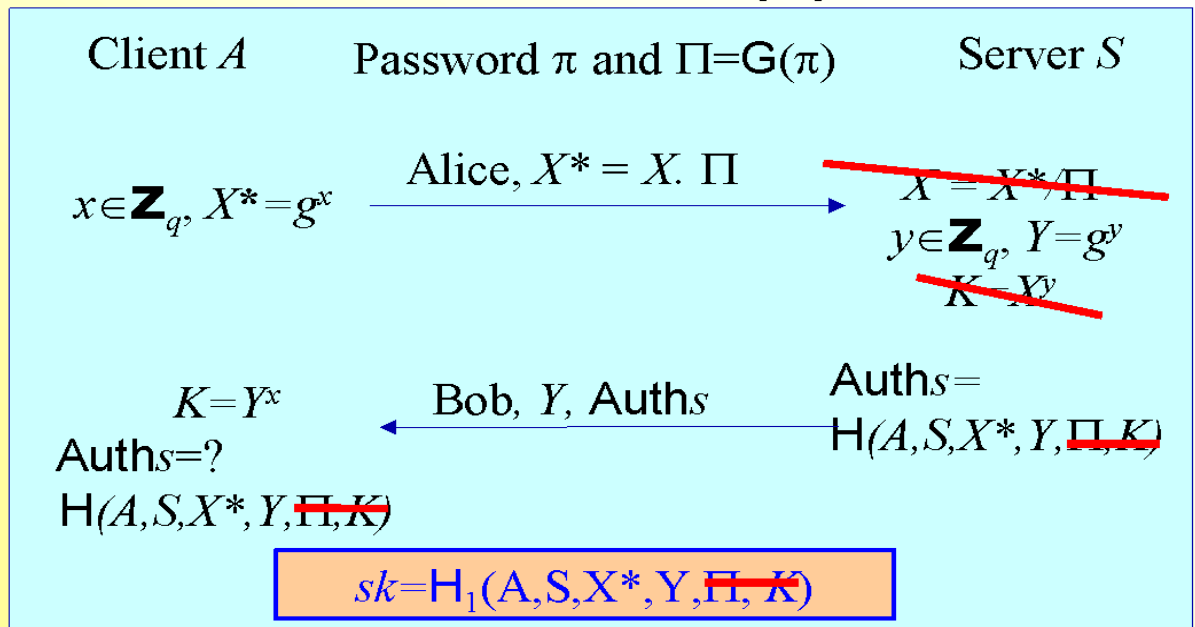
advantage $\geq 12q_s/N + \varepsilon$,

\Rightarrow CDH problem : probability $\geq \varepsilon/qh^2$
(within almost the same time)

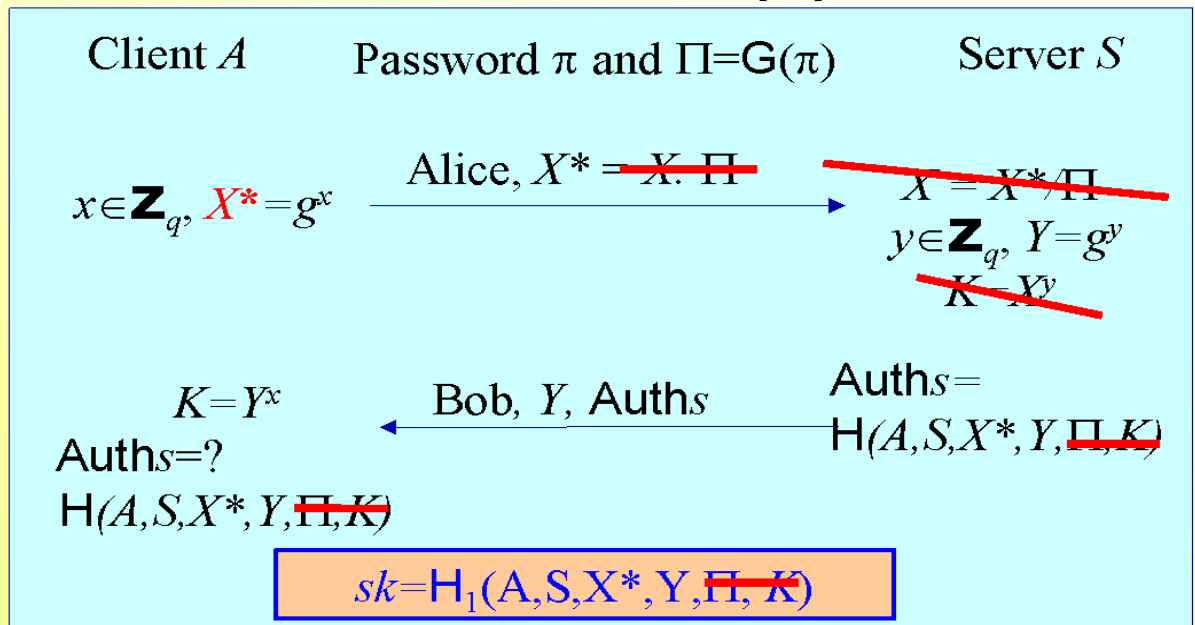
One-Mask DHKE: the Proof



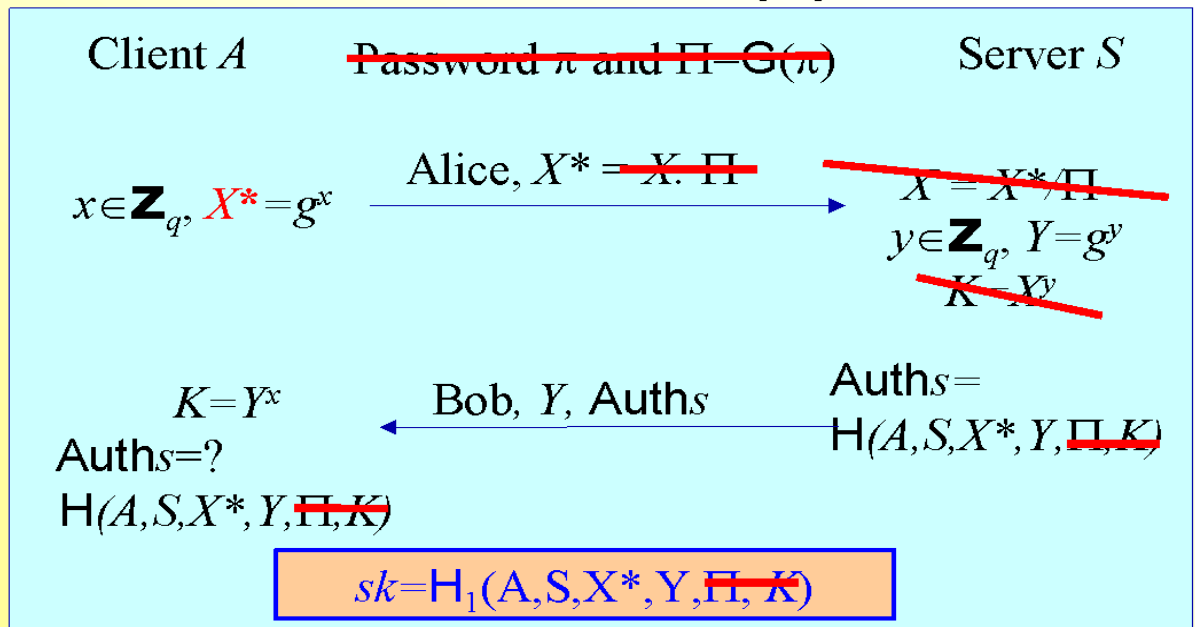
The Proof (2)



The Proof (3)



The Proof (4)



One-Mask DH Key Exchange

- The simulated execution is indistinguishable from the real one, unless:
 - adversary asks the random oracle on values such as (A, S, X^*, Y, Π, K)
 - if both X^* and Y are simulated from an instance of the DH problem, the adversary has solved it (when submitting K)
 - if one of these values is built by the adversary, it corresponds to an active attempt => at most q_s
 - adversary has guessed the password by pure chance: proba $\leq q_s / N$ since, the password is information-theoretically hidden in the simulation

Conclusion

- **One-Mask and Two-Mask EKE variants are**
 - provably secure in the random oracle model
 - semantic security
 - unilateral or mutual authentication
- **More efficient than EKE**
 - only one flow is encrypted
- **More suitable for client-server schemes**
 - the server can first send a generic flow not encrypted, and thus independent of the client