Group Key Exchange and Provable Security

joint work with E. Bresson and O. Chevassut

David Pointcheval
Département d'Informatique
ENS - CNRS

David.Pointcheval@ens.fr http://www.di.ens.fr/~pointche

Overview

- Provable Security
- Key Agreement
  and Mutual Authentication
  - Definitions
  - Security Model
  - Example
- Group Key Agreement
  - Security Model
  - Example (security result)
- Conclusion

Overview

- Provable Security
- Key Agreement
  and Mutual Authentication
- Group Key Agreement
- Conclusion

Provably Secure Scheme

To prove the security of a cryptographic scheme, one has to make precise

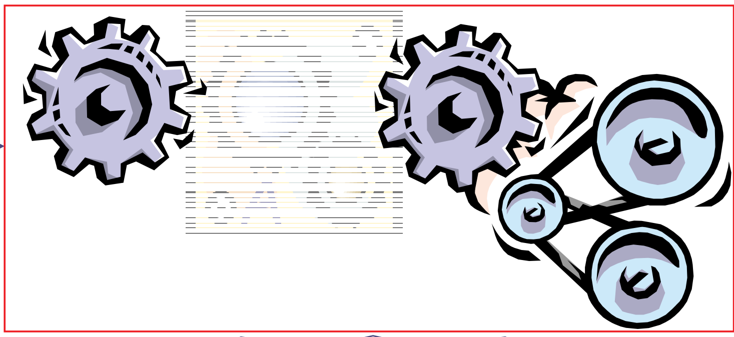
- the algorithmic assumptions
- the security notions to be guaranteed
- a reduction:
  an adversary can help
  to break the assumption

Proof by Reduction

Reduction of a problem **P** to an attack *Atk*:

Let *A* be an adversary that breaks the scheme then *A* can be used to solve **P**

Instance **I** of **P** →

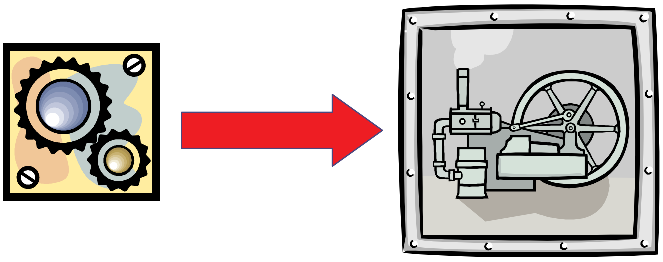


→ Solution of **I**

P intractable ⇒ scheme unbreakable

Practical Security

Adversary within t



Algorithm against **P** within $t' = T(t)$

- Complexity theory: T polynomial
- Exact Security: T explicit
- Practical Security: T small (linear)

Eg : $t' = 4t$

P intractable within less than 2^{80} operations

⇒ scheme unbreakable within less than 2^{78} operations

Security Notions

According to the needs, one defines

- the goals of an adversary
- the means of an adversary,
  i.e. the available information

Overview

- Provable Security
- Key Agreement
  and Mutual Authentication
  - Definitions
  - Security Model
  - Example
- Group Key Agreement
- Conclusion

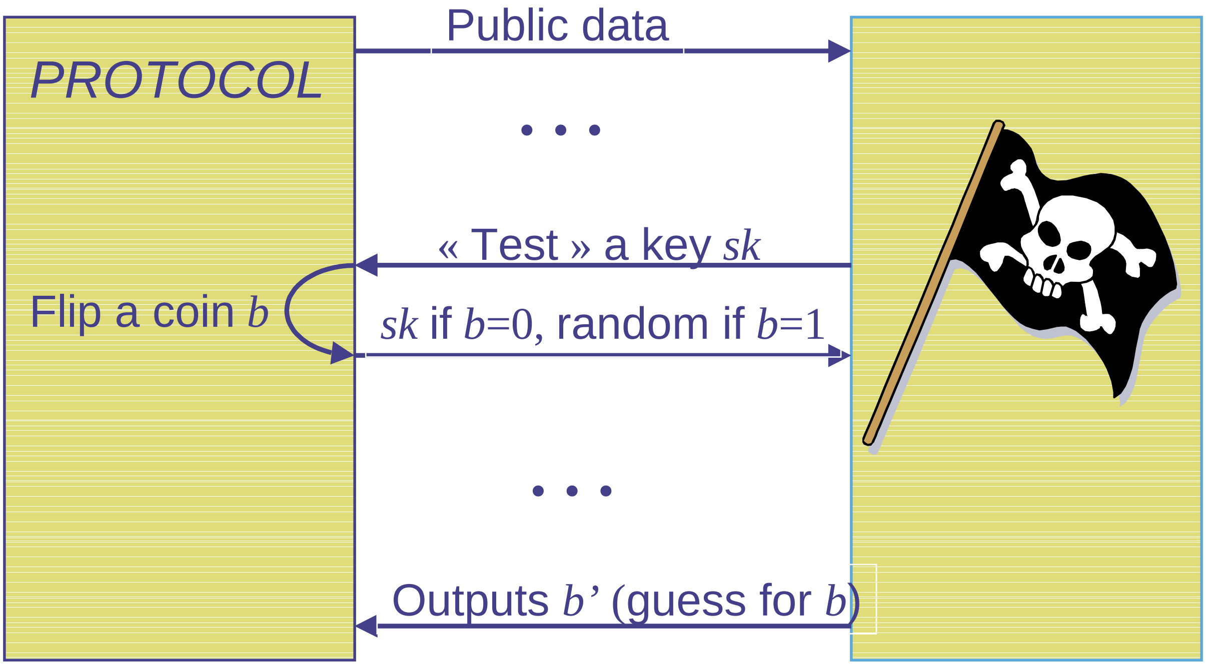
Overview

- Provable Security
- Key Agreement
  and Mutual Authentication
  - Definitions
  - Security Model
  - Example
- Group Key Agreement
- Conclusion

Authenticated Key Exchange

- Implicit authentication
  - only the intended partners can compute the session key
- Semantic security
  - the session key is indistinguishable from a random string
  - modeled via a **Test**-query

Security Definitions (AKE)



Further Properties

- Mutual authentication

  they are both sure to share the secret with the people they think they do

- Forward secrecy

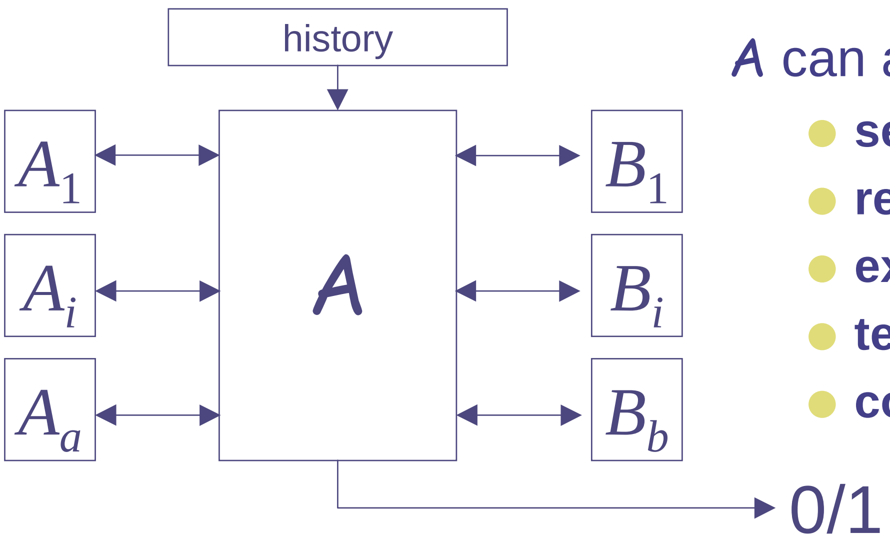
  even if a long-term secret data is corrupted, previous shared secrets are still semantically secure

Overview

- Provable Security
- Key Agreement
  and Mutual Authentication
  - Definitions
  - Security Model
  - Example
- Group Key Agreement
- Conclusion

Formal Model

Bellare-Rogaway model revisited by Shoup



$\mathcal{A}$ can ask

- **send**-queries
- **reveal**-queries
- **execute**-queries
- **test**-query
- **corrupt**-queries

Semantic Security

- A misuse of the secret data is modeled by the **reveal**-query, which is answered by this secret data
- For the semantic security, the adversary asks one **test**-query which is answered, according to a bit b, by
  - $b=0$: the actual secret data
  - $b=1$: a random string

$\Rightarrow$ the adversary has to guess this bit b

Passive/Active Adversaries

- Passive adversary: **history** built using the **execute**-queries $\rightarrow$ transcripts
- Active adversary: entire control of the network with **send**-queries:
  - to send message to Alice or Bob (in place of Bob or Alice respectively)
  - to intercept, forward and/or modify messages

Forward Secrecy

Forward secrecy means that the adversary cannot distinguish a session key established ***before*** any corruption of the long-term private keys:

- the **corrupt**-query is answered by the long-term private key of the corrupted party
- then the **test**-query must be asked on a session key established ***before*** any **corrupt**-query

Overview

- Provable Security
- Key Agreement and Mutual Authentication
  - Definitions
  - Security Model
  - Example
- Group Key Agreement
- Conclusion

Diffie-Hellman Key Exchange

The most classical key exchange scheme has been proposed by Diffie-Hellman:

$\mathbf{G} = \langle g \rangle$, cyclic group of prime order q

- Alice chooses a random $x \in \mathbf{Z}_q$, computes and sends $X = g^x$
- Bob chooses a random $y \in \mathbf{Z}_q$, computes and sends $Y = g^y$
- They each can compute the session key

$$K = Y^x = X^y$$

Properties

- If flows are authenticated, it is well-known to provide the semantic security of the session key under the **Decisional Diffie-Hellman Problem**
- If one derives the session key as $k = \mathrm{H}(K)$, where H is assumed to behave like a random oracle, semantic security is relative to the **Computational Diffie-Hellman Problem**

Further Features

- But there is no explicit authentication (Replay attacks)
- Adding key confirmation rounds: **mutual authentication** [BPR00]

Alice (S_a, P_a)		Bob (S_b, P_b)
$x \in \mathbf{Z}_q$, $X = g^x$	Bob, X, $\mathbf{S}(S_a, X)$ $\longrightarrow$	
		$y \in \mathbf{Z}_q$, $Y = g^y$
		$K = X^y$
	$\longleftarrow$ Alice, Y, $\mathbf{S}(S_b, X, Y)$, k_1	$k_1 = H(K \Vert 1)$
k_1 correct?		
$k_2 = H(K \Vert 2)$	k_2 $\longrightarrow$	k_2 correct?
	$k = H(K \Vert 0)$	

Overview

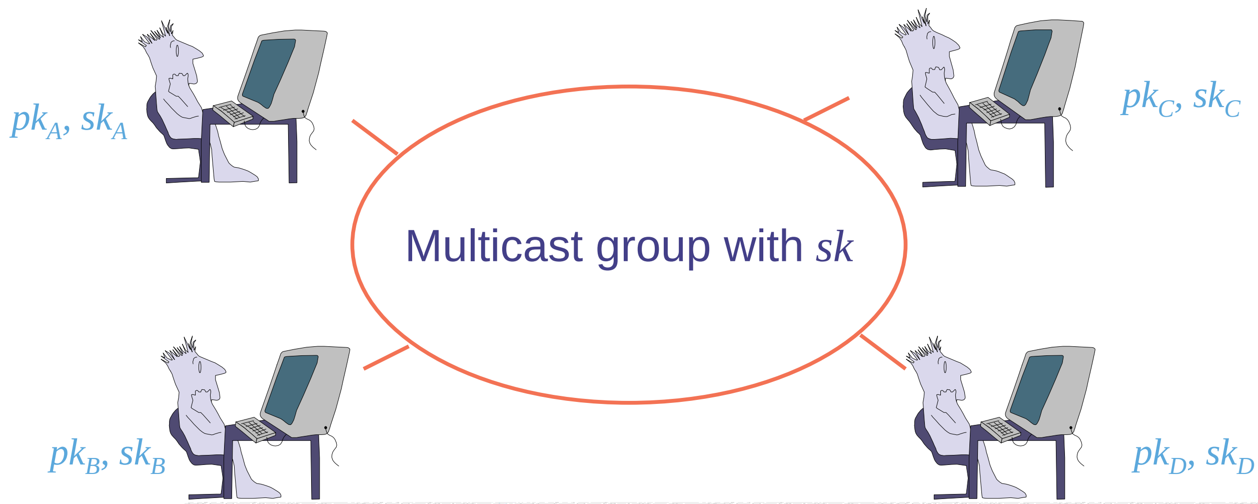
- Provable Security
- Key Agreement and Mutual Authentication
- Group Key Agreement
  - Security Model
  - Example (security result)
- Conclusion

Overview

- Provable Security
- Key Agreement
  and Mutual Authentication
- Group Key Agreement
  - Security Model
  - Example (security result)
- Conclusion

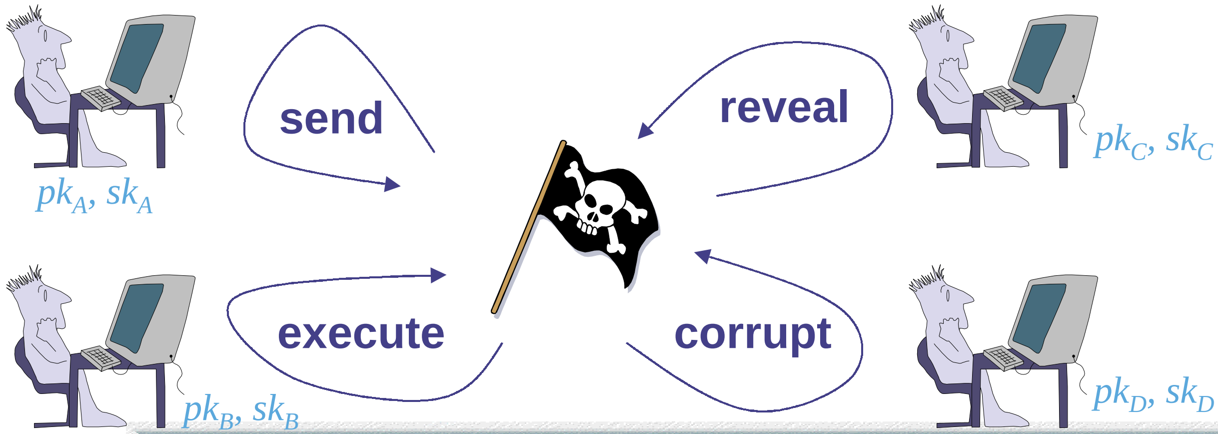
Model of Communication

- A set of *n* players, modeled by oracles
- A multicast group consisting of a set of players

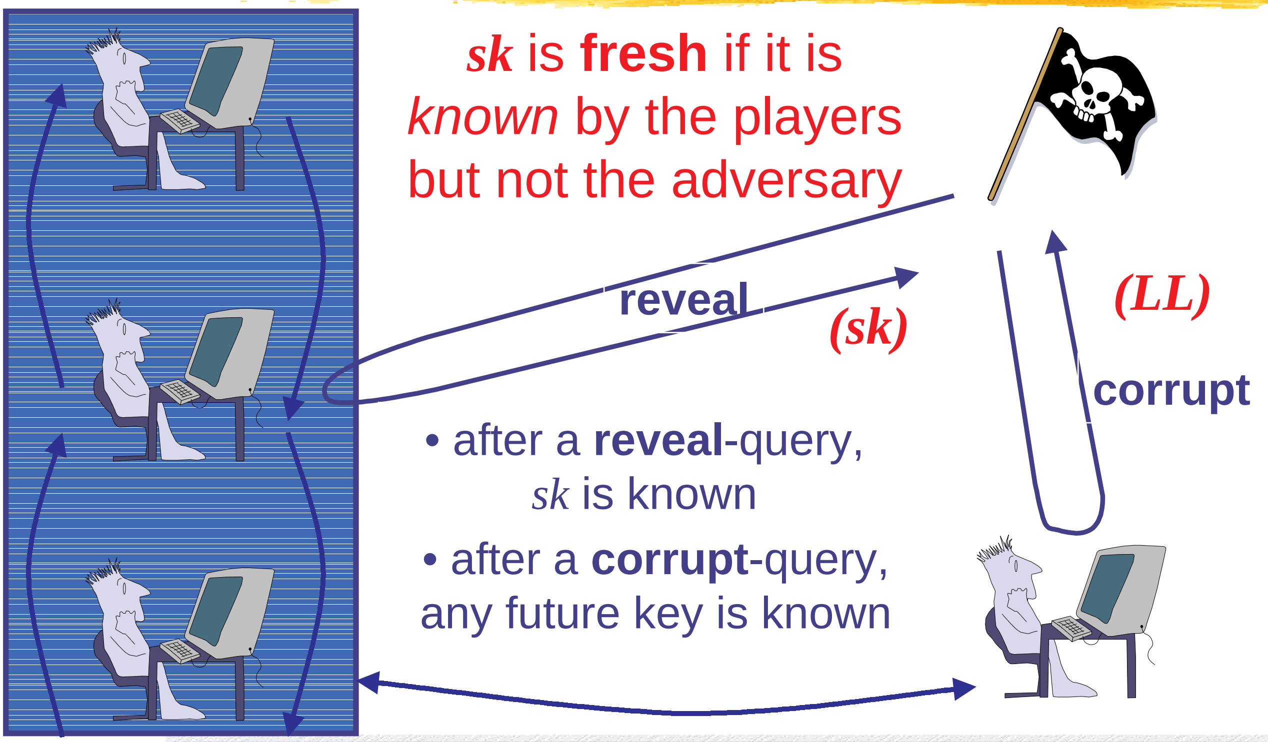


Modeling the Adversary

- **send**: send messages to instances
- **execute**: obtain honest executions of the protocol
- **reveal**: obtain an instance's session key
- **corrupt**: obtain the value of the password



Freshness



Overview

- Provable Security
- Key Agreement
  and Mutual Authentication
- Group Key Agreement
  - Security Model
  - Example (security result)
- Conclusion

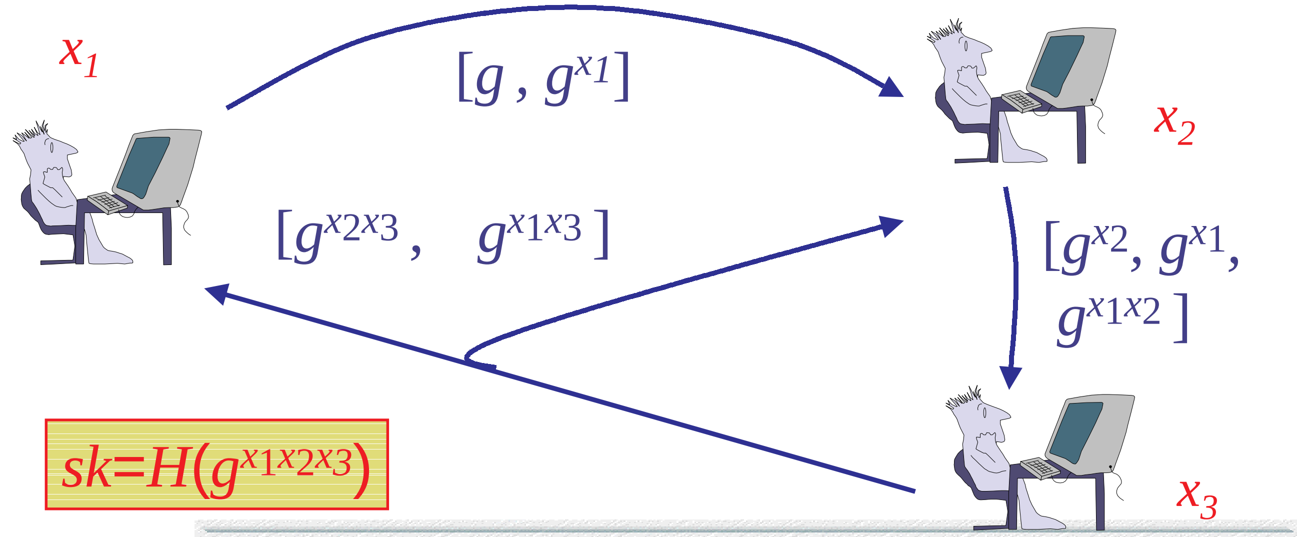
A Group Key Exchange

- Generalization of the 2-party DH,
  the session key is $sk=H(g^{x_1 x_2 \cdots x_n})$
- Ring-based algorithm
  - **up-flow**: the contributions of each instance are gathered
  - **down-flow**: the last instance broadcasts the result
  - **end**: instances compute the session key from the broadcast

The Algorithm

- **Up-flow**: U_i raises received values to the power x_i
- **Down-flow**: U_n broadcasts (except $g^{x_1 x_2 \dots x_n}$)

Everything is authenticated (Signature/MAC)



Group CDH

- The CDH generalized to the multi-party case
  - given the values $g^{\prod x_i}$ for some choice of proper subset of $\{1, \dots, n\}$
  - one has to compute the value $g^{x_1 \cdots x_n}$
- Example (n=3 and I={1,2,3})
  - given the set of the **blue** values g, g^{x_1}, g^{x_1}, g^{x_2}, $g^{x_1 x_2}$, $g^{x_1 x_3}$, $g^{x_2 x_3}$
  - compute the red value $g^{x_1 x_2 x_3}$
- The GCDH $\Leftrightarrow$ DDH and CDH [SAC '02]

Security Result

- Theorem (in the random oracle model)

$$\mathrm{Adv}^{\mathrm{ake}}(T,n,q_s,q_e) \leq 2q_s^{\,n}\, q_h \cdot \mathrm{Succ}^{\mathrm{gcdh}}(n,T) + 2n \cdot \mathrm{Succ}^{\mathrm{sign}}(q_s,T)$$

- Proof:
  - Game 0 : the adversary *A* plays against the oracles in order to defeat the AKE-security

$$\varepsilon = (\mathrm{Adv}(A)+1)/2 = \Pr[b' = b] = \Pr[S_0]$$

Security Result (2)

Game 1:

- Exclude games wherein a signature/MAC forgery is performed:

$$| \Pr[S_1] - \Pr[S_0] | < n \cdot \mathrm{Succ}^{\mathrm{sign}}(q_s,T)$$

Security Result (3)

Game 2:

- guess n indices between 1 and q_s
  (this defines a pool of n instances,
  involved in the n queries)
- cancel executions of the game such that this pool of instances does not correspond to the **Test**-query (in other cases, output a random b')

Remarks:

- The probability of a correct guess is exactly $1/q_s^n$
- Such a correct guess is independent with S_1

Security Result (4)

$$\begin{aligned} \Pr[S_2] &= \Pr[S_1 \wedge \text{guess}] + \Pr[S_1 \wedge \neg\text{guess}] \\ &= \Pr[S_1 \mid \text{guess}] \Pr[\text{guess}] \\ &\qquad + \Pr[S_1 \mid \neg\text{guess}] \Pr[\neg\text{guess}] \\ &= \Pr[S_1] / q_s^n + 1/2\,(1 - 1 / q_s^n) \\ &= 1/2 + (\Pr[S_1] - 1/2) / q_s^n \end{aligned}$$

$$\Pr[S_0] \leq \Pr[S_1] + n \cdot \mathrm{Succ}^{\mathrm{sign}}(q_s, T)$$

$$\begin{aligned} 2 \cdot \Pr[S_0] - 1 &\leq 2 \cdot \Pr[S_1] - 1 + 2n \cdot \mathrm{Succ}^{\mathrm{sign}}(q_s, T) \\ &\leq q_s^n\,(2 \cdot \Pr[S_2] - 1) + 2n \cdot \mathrm{Succ}^{\mathrm{sign}}(q_s, T) \end{aligned}$$

Security Result (5)

Game 3:

- Replace sk for this pool, by a random value

Remark:

- A problem may happen if $\mathcal{A}$ asks for $H(g^{x_1 x_2 \ldots x_n})$, which should be equal to sk: Event **AskH$_3$**

$$| \Pr[S_3] - \Pr[S_2] | \leq \Pr[\mathbf{AskH_3}]$$

- Since sk is random (independent to the view of the adversary)

$$\Pr[S_3] = 1/2$$

$$\mathrm{Adv}(\mathcal{A}) \leq 2 q_s^n \cdot \Pr[\mathbf{AskH_3}] + 2n \cdot \mathrm{Succ}^{\mathrm{sign}}(q_s, T)$$

Security Result (6)

Game 4:

- Inject the GCDH instance for simulating the selected oracle instances

$$\Pr[\mathbf{AskH_4}] = \Pr[\mathbf{AskH_3}]$$

Remark: event **AskH$_4$** means that

- $H(g^{x_1 x_2 \ldots x_n})$, has been asked
- $g^{x_1 x_2 \ldots x_n}$ is in the list of the queries asked to H
- With a random guess, one gets it:

$$\Pr[\mathbf{AskH_4}] \leq q_h \cdot \mathrm{Succ}^{\mathrm{gcdh}}(n, T)$$

Overview

- Provable Security
- Key Agreement
  and Mutual Authentication
- Group Key Agreement
- Conclusion

Improvements

- Security result: exponential in n [ACM CCS '01]
  - No guess of the tested pool
  - Use of the random self-reducibility
    of the CDH and GCDH problems
    $\Rightarrow$ reduction linear in n
  - Standard Model [Eurocrypt '02]
- Dynamic groups [Asiacrypt '01]
  - If one party leaves or joins the group,
    the protocol does not need to be restarted
    from scratch

Improvements: Result

- Group of n people
- Tested group of size s
- Number of dynamic modifications (setup, join, remove): Q
- Time: T

$$\mathrm{Adv}^{\mathrm{ake}}(A) \leq 2\, Q \cdot C_n^{\,s} \cdot q_h \cdot \mathrm{Succ}^{\mathrm{gcdh}}(s,T) + 2n \cdot \mathrm{Succ}^{\mathrm{sign}}(q_s,T)$$

Mutual Authentication

- Authentication of the parties:
  - Public Key Infrastructures (signatures)
  - Secret keys - MAC [Eurocrypt '02]
  - Passwords [Asiacrypt '02]

In the latter case, a new kind of attack has to be considered: dictionary attacks

Conclusion

- Formal model for (Group) AKE
- Provably secure schemes
  but still not « practical security »
- Various authentication modes