Hash Proof Systems and Password Protocols

II – Password-Authenticated Key Exchange

David Pointcheval

CNRS, Ecole normale supérieure/PSL & INRIA

8th BIU Winter School – Key Exchange
February 2018

Diffie-Hellman Key Exchange

Diffie-Hellman protocol: allows two parties to agree on a common session key:
In a finite cyclic group $\mathcal{G}$, of prime order p, with a generator g

$$\begin{array}{ccc} x \stackrel{\$}{\leftarrow} \mathbb{Z}_p, X \leftarrow g^x & \xrightarrow{\quad X \quad} & y \stackrel{\$}{\leftarrow} \mathbb{Z}_p, Y \leftarrow g^y \\ K \leftarrow Y^x = g^{xy} & \xleftarrow{\quad Y \quad} & K \leftarrow X^y = g^{xy} \end{array}$$

No authentication provided

Authenticated Key Exchange

Semantic security / Implicit Authentication:
the session key should be indistinguishable from a random string
to all except the expected players

Authentication Techniques

Asymmetric technique

- Assume the existence of a public-key infrastructure
- Each party holds a pair of secret and public keys

Symmetric technique

- Users share a random secret key

Password-based technique

- Users share a random *low-entropy* secret: **password**

Electronic Passport

Since 1998, some passports contain digital information on a chip
Standards specified by ICAO (International Civil Aviation Organization)

In 2004, security introduced:

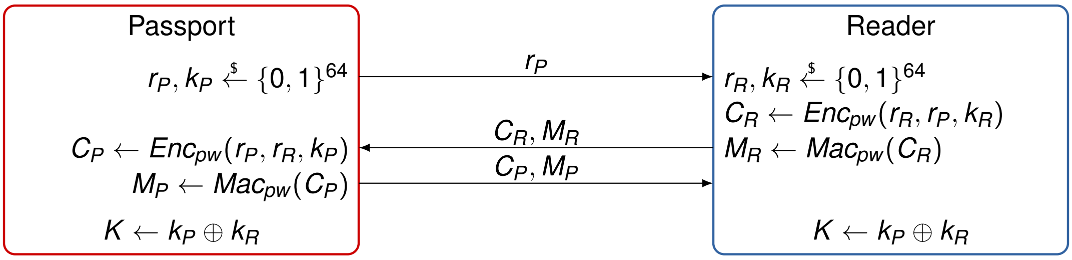
- encrypted communication between the chip and the reader
- access control: BAC (Basic Access Control)

The shared secret is on the MRZ
(Machine Readable Zone)
It has low entropy:
at most 72 bits,
but actually approx. 40

$\Longrightarrow$ low-entropy shared secret: a password *pw*

BAC: Basic Access Control

The symmetric encryption and MAC keys are deterministically derived from *pw*



From a pair (C_R, M_R), one can make an exhaustive search
on the password *pw* to check the validity of the Mac M_R
After a few eavesdroppings only : password recovery

What can we expect from a low-entropy secret?

Off-line Dictionary Attacks

As in the previous scenario, after having

- eavesdropped some (possibly many) transcripts
- interacted (quite a few times) with players

the adversary accumulates enough information
to take the real password apart from the dictionary

$\Longrightarrow$ Efficient password-recovery after off-line exhaustive search

For the BAC: quite a few passive eavesdroppings are enough to recover the password!
How many active interactions could one enforce?

On-line Dictionary Attacks

On-line Dictionary Attacks

- The adversary interacts with a player, trying a password
- In case of success: it has guessed the password
- In case of failure: it tries again with another password

In Practice

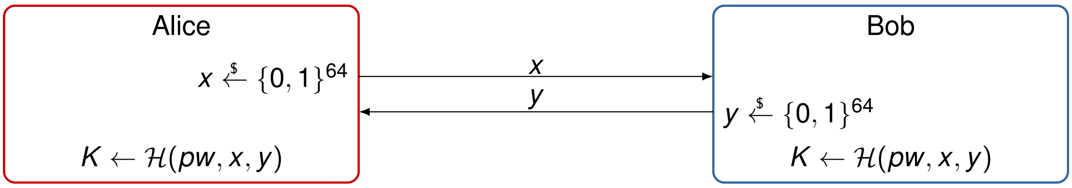
- This attack is unavoidable
- If the failures for a target user can be detected
  the impact can be limited by various techniques
- If the failures cannot be detected (anonymity, no check, . . .)
  the impact can be dramatic

Outline

Outline

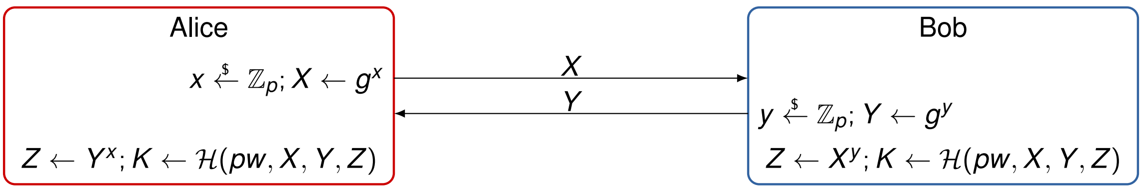
Outline

First Attempt



Seems better than BAC: no information leaks about K, so no leakage about pw either!
But K will be later used: $c = E_K(m)$
any information about m leaks about K, and leaks on pw...
$\Longrightarrow$ The security model has to deal with information leakage about K

Second Attempt



Passive eavesdropping, even with leakage of K: secure under **CDH**!
But the adversary can try to impersonate Bob, and know Z...
$\Longrightarrow$ The security model has to deal with active attacks

Security Models

- Game-based Security [Bellare-P.-Rogaway – Eurocrypt '00]
  - Find-then-Guess
  - Real-or-Random [Abdalla-Fouque-P. – PKC '05]
- Simulation-based Security [Boyko-MacKenzie-Patel – Eurocrypt '00]
- Universal Composability [Canetti-Halevi-Katz-Lindell-MacKenzie – Eurocrypt '05]

Where

- The adversary controls the network: it can create, alter, delete, duplicate messages
- Users can participate in concurrent executions of the protocol

On-line dictionary attack should be the best attack

$\Longrightarrow$ No adversary should win with probability greater than q_S/N
where $q_S = \#$Active Sessions and $N = \#$Dictionary

Outline

Game-based Security

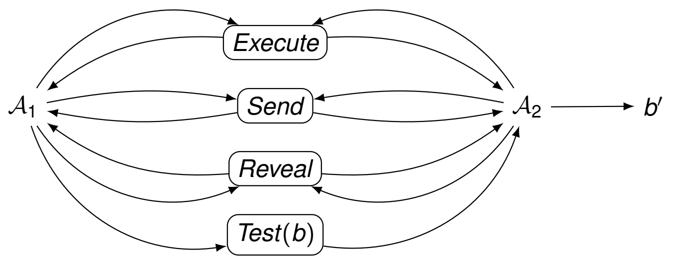
[Bellare-P.-Rogaway – Eurocrypt '00]

The adversary $\mathcal{A}$ interacts with oracles:

- *Execute*(A^i, B^j)
  $\mathcal{A}$ gets the transcript of an execution between A and B
  $\Longrightarrow$ Passive attacks (eavesdropping)
- *Send*(U^i, m)
  $\mathcal{A}$ sends the message m to the instance U^i
  $\Longrightarrow$ Active attacks against U^i (active sessions)
- *Reveal*(U^i)
  $\mathcal{A}$ gets the session key established by U^i and its partner
  $\Longrightarrow$ Leakage of the session key, due to a misuse
- *Test*(U^i) a random bit b is chosen
  - If $b = 0$, $\mathcal{A}$ gets the session key (*i.e.*, *Reveal*(U^i))
  - If $b = 1$, $\mathcal{A}$ gets a random key

Security Game: Find-then-Guess

Secrecy of the key: output b', the guess of the bit b involved in the Test-query
Is the obtained key real or random?
Constraint: no *Test*-query on a trivially known key
i.e., key already revealed through the instance or its partner



$$Adv^{FtG}(\mathcal{A}) = 2 \times \Pr[b' = b] - 1 \leq \frac{q_S}{N} + negl()$$

Freshness and Partnering

- Partners
  Two players are partners if they share the same Session ID
  Where SID should model ideal executions:
  - two players with same SID's and same *pw*'s conclude with the same session key
  - two players with different SID's or different *pw*'s conclude with independent keys
- Freshness
  A key or a player is fresh if none of the key/player or the partner's key/player has been revealed/tested

Only fresh keys/players can be revealed/tested

Security Notions: Forward Secrecy

- Semantic Security
  The Find-then-Guess game models the **secrecy** of the key
  $\Longrightarrow$ the session key is unknown to the other players
  - What about this secrecy after the corruption of a player?
  - What about the knowledge of the two players?
- Forward Secrecy
  - An additional oracle: *Corrupt*(U) provides the password *pw* of the player U to the adversary
  - A new constraint: For any *Test*(U^i), player U was not corrupted when U^i was involved in its session

Outline

Encrypted Key Exchange

[Bellovin-Merritt – S&P '92]

Alice

$x \stackrel{\$}{\leftarrow} \mathbb{Z}_p; X \leftarrow g^x$

$X^* \leftarrow E_{pw}(X)$

$Z \leftarrow Y^x \qquad Y \leftarrow D_{pw}(Y^*)$

$K \leftarrow \mathcal{H}(A, B, X^*, Y^*, Z)$

Alice → Bob: X^*

Bob → Alice: Y^*

Bob

$y \stackrel{\$}{\leftarrow} \mathbb{Z}_p; Y \leftarrow g^y$

$X \leftarrow D_{pw}(X^*)$

$Y^* \leftarrow E_{pw}(Y) \qquad Z \leftarrow X^y$

$K \leftarrow \mathcal{H}(A, B, X^*, Y^*, Z)$

Semantically Secure with Forward Secrecy if
- **CDH** assumption holds
- (E, D) is an Ideal Cipher onto $\mathbb{G} = \langle g \rangle$
- $\mathcal{H}$ is a Random Oracle

[Bellare-P.-Rogaway – Eurocrypt '00]

Simple PAKE

[Abdalla-P. – CT-RSA '05]

Alice

$x \stackrel{\$}{\leftarrow} \mathbb{Z}_p; X \leftarrow g^x$

$X^* \leftarrow X \cdot M^{pw}$

$Z \leftarrow Y^x \qquad Y \leftarrow Y^*/N^{pw}$

$K \leftarrow \mathcal{H}(A, B, pw, X^*, Y^*, Z)$

Alice → Bob: X^*

Bob → Alice: Y^*

Bob

$y \stackrel{\$}{\leftarrow} \mathbb{Z}_p; Y \leftarrow g^y$

$X \leftarrow X^*/M^{pw}$

$Y^* \leftarrow Y \cdot N^{pw} \qquad Z \leftarrow X^y$

$K \leftarrow \mathcal{H}(A, B, pw, X^*, Y^*, Z)$

Semantically Secure if
- **CDH**(M, N) hard to break
- $\mathcal{H}$ is a Random Oracle

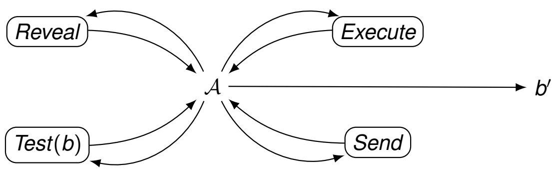
Outline

Security Game: Real-or-Random

[Abdalla-Fouque-P. – PKC '05]

Secrecy/independence of all the keys: many *Test*-queries with the same bit b

- If no key defined by the protocol yet: output $\perp$
- If dishonest/corrupted partner: output the real key
- If player/partner already tested (not fresh): output the same key
- If $b = 0$: output the real key
- If $b = 1$: output a random key



$$Adv^{RoR}(\mathcal{A}) = 2 \times \Pr[b' = b] - 1$$

Security Game: Real-or-Random

Semantic Security (Encryption)

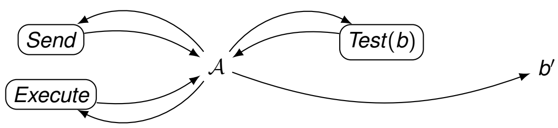
[Bellare-Desai-Jokipii-Rogaway – FOCS '97]

Find-then-Guess and Real-or-Random are polynomially equivalent

$$Adv^{RoR}(t, q_T) \leq q_T \times Adv^{FtG}(t)$$

where q_T is the number of Test-queries

- For Password-based Authenticated Key Exchange:
  $Adv^{FtG}(t) \leq \frac{q_S}{N} \not\Rightarrow Adv^{RoR}(t, q_T) \leq \frac{q_S}{N}$ $\Longrightarrow$ Stronger notion
- No need of Reveal-queries $\Longrightarrow$ Simpler security notion [Abdalla-Fouque-P – PKC '05]



Game-based Security: Limitations

- Proven bounds: $O(q_S)/N$, but almost never q_S/N
  $\Longrightarrow$ hard to get optimal bound!
  This means: a few passwords can be excluded by each active attack
  But q_S is sometimes the number of Send-queries
  which is more than the number of Active Sessions
- Passwords chosen from pre-determined, known distributions
- Different passwords are assumed to be independent
- No security guarantees under arbitrary compositions

$\Longrightarrow$ Universal Composability more appropriate [Canetti – FOCS '01]
[Canetti-Halevi-Katz-Lindell-MacKenzie – Eurocrypt '05]

Outline

Outline

Definition

Real Protocol

The real protocol $\mathcal{P}$ is run by players $P_1, \ldots, P_n$,
with their own private inputs $x_1, \ldots, x_n$.
After interactions, they get outputs $y_1, \ldots, y_n$

Ideal Functionality

An ideal function $\mathcal{F}$ is defined:

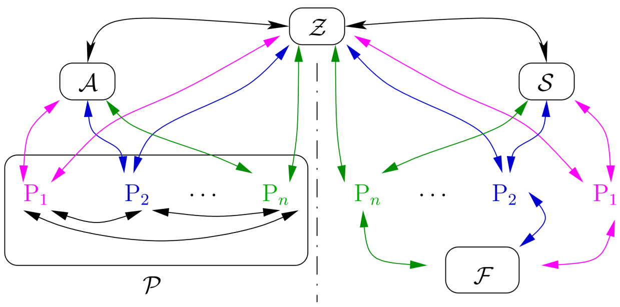
- it takes as input $x_1, \ldots, x_n$,
  the private information of each player,
- and outputs $y_1, \ldots, y_n$, given privately to each player

The players get their results, without interacting:
this is a "by definition" secure primitive

Simulator

$\mathcal{P}$ emulates $\mathcal{F}$ if, for any environment $\mathcal{Z}$, for any adversary $\mathcal{A}$,
there exists a simulator $\mathcal{S}$ so that, the view of $\mathcal{Z}$ is the same for

- $\mathcal{A}$ attacking the real protocol $\mathcal{P}$
- $\mathcal{S}$ attacking the ideal functionality $\mathcal{F}$



Outline

- Introduction
- 1 Security Notions
  - Intuition
  - Find-then-Guess Security
  - Examples
  - Real-or-Random Security
- 2 **Universal Composability**
  - Definition
  - Password-based Authenticated Key Exchange
  - Advanced Security Notions
  - Examples
- Conclusion

PAKE Ideal Functionality

[Canetti-Halevi-Katz-Lindell-MacKenzie – Eurocrypt '05]

Queries

- `NewSession` = a player joins the system with a password
- `TestPwd` = $\mathcal{A}$ attempts to guess a password (**one** per session)
  The adversary learns whether the guess was correct or not
- `NewKey` = $\mathcal{A}$ asks for the session key to be computed and delivered to the player

Corruption-Query

- $\mathcal{A}$ gets the long-term secrets (*pw*) and the internal state
- $\mathcal{A}$ takes the entire control on the player and plays on its behalf

Corruptions can occur **before the execution**: Static Corruptions
Corruptions can occur **at any moment**: Adaptive Corruptions

PAKE Ideal Functionality

[Canetti-Halevi-Katz-Lindell-MacKenzie – Eurocrypt '05]

Session Key

- No corrupted players, same passwords
  $\Longrightarrow$ same key, randomly chosen
- No corrupted players, different passwords
  $\Longrightarrow$ independent keys, randomly chosen
- A corrupted player
  $\Longrightarrow$ key chosen by the adversary
- Correct password guess (`TestPwd`-query)
  $\Longrightarrow$ key chosen by the adversary
- Incorrect password guess (`TestPwd`-query)
  $\Longrightarrow$ independent keys, randomly chosen

PAKE Ideal Functionality

Properties

- The `TestPwd`-query models the on-line dictionary attacks
- The `Corruption`-query includes forward secrecy

Advantages wrt Game-based Security

- No assumption on the distribution of passwords (chosen by the environment)
- Passwords can be related (it models mistyping)
- Security under arbitrary compositions $\Longrightarrow$ secure channels

Game-based Security vs. Universal Composability

Game-based Security

In the reduction, the simulator has to emulate the protocol execution
only up to an evidence the adversary has won ($pw \Longrightarrow$ not negl.)
In the global system, the simulation fails when the adversary breaks one sub-protocol
whereas other parts could provide protection ($pw \Longrightarrow$ weak proof!)

UC Security

Simulation handles compositions, but proofs are more complex:
the simulator must have an indistinguishable behavior, even when the adversary wins!

In the case of password-based cryptography:
the adversary can win with non-negligible probability!

Outline

Properties of the `NewKey`-Query

Session Key: `NewKey`-Query

...
- A corrupted player $\Longrightarrow$ key chosen by the adversary
- Correct password guess $\Longrightarrow$ key chosen by the adversary

...

The `NewKey`-query models possible Key Distribution:
$\Longrightarrow$ the session key can be controlled by one of the players
The contributiveness property models Key Agreement [Adalla-Catalano-Chevalier-P. – CT-RSA '09]
$\Longrightarrow$ no player can decide on the key

Properties of the `TestPwd`-Query

Dictionary Attack: `TestPwd`-Query

- Correct password guess $\Longrightarrow$ key chosen by the adversary
- Incorrect password guess $\Longrightarrow$ random key

And adversary informed of correct/incorrect guess

The `TestPwd`-query models Explicit Authentication:
$\Longrightarrow$ the players are informed of success/failure

Implicit-Only PAKE models Implicit Authentication [Dupont-Hesse-P.-Reyzin-Yakoubov – Eurocrypt '18]
$\Longrightarrow$ the keys have to be used to test success/failure

Outline

UC-Secure PAKE

With a random oracle and an ideal cipher: EKE [Abdalla-Catalano-Chevalier-P. – CT-RSA '08]
$\Longrightarrow$ First efficient scheme secure against Adaptive Corruptions

In the standard model, based on GL (abstraction of KOY)
$\Longrightarrow$ BPR-security using SPHFs [Gennaro-Lindell – Eurocrypt '03]

- with *SS-ZK* $\Longrightarrow$ Static corruptions [Canetti-Halevi-Katz-Lindell-MacKenzie – Eurocrypt '05]
- with an *equivocable/extractable commitment*
  $\Longrightarrow$ Adaptive corruptions [Abdalla-Chevalier-P. – Crypto '09]
- with *KV-SPHF and SS-NIZK* $\Longrightarrow$ One-round only [Katz-Vaikuntanathan – TCC '11]
- with *Explainable SPHFs*
  $\Longrightarrow$ Adaptive corruptions without erasures [Abdalla-Benhamouda-P. – PKC '17]

assuming a CRS (proven impossible in the plain model)

Outline

Conclusion

EKE is a secure PAKE in the ROM+ICM:

- BPR secure
- UC secure
- Withstands adaptive corruptions
- Provides forward secrecy
- Can guarantee Explicit or Implicit-Only authentication

All the constructions in the standard model exploit SPHFs:

- based on the KOY protocol [Katz-Ostrovsky-Yung – Crypto '01]
- extend the GL protocol [Gennaro-Lindell – Eurocrypt '03]

Let us see SPHF-based PAKE Protocols