# The Twist-Augmented Approach for Diffie-Hellman Key Exchange
## Entropy Smoothing and Key Derivation


David Pointcheval

CNRS - Ecole normale supérieure, FRANCE


Workshop on Cryptography

CIRM-Luminy, France

*Joint work with Olivier Chevassut, Pierre-Alain Fouque and Pierrick Gaudry*

# Overview

- Authenticated Diffie-Hellman Key Exchange
- Security Model
- Usual Flaw in the Security Analysis
- The Twist-Augmented Approach

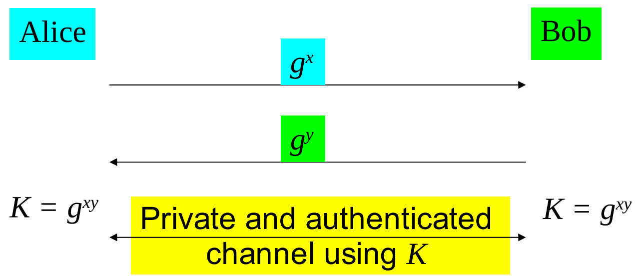


# Overview

- ➔ Authenticated Diffie-Hellman Key Exchange
- Security Model
- Usual Flaw in the Security Analysis
- The Twist-Augmented Approach

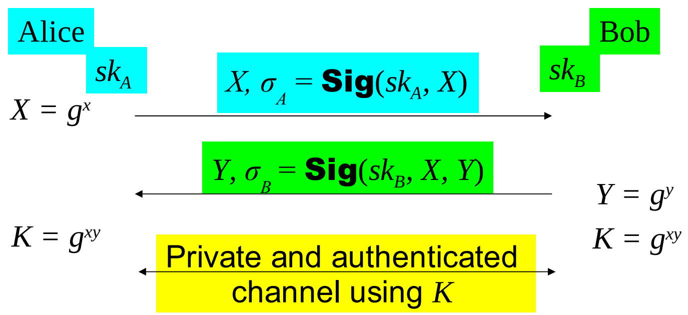


# Diffie-Hellman Key Exchange




- Semantic security:
  $K$ is indistinguishable from a random key
  $\Rightarrow$ a *random* ***bit-string***
- Man-in-the-middle attacks
  $\Rightarrow$ authentication

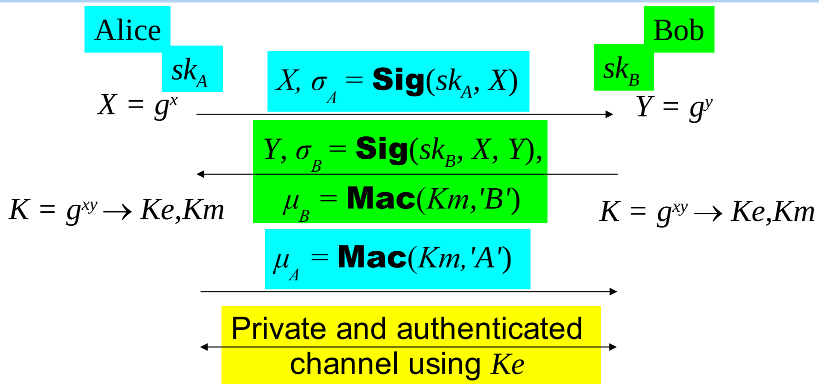
# Authenticated Diffie-Hellman Key Exchange




- Replay attacks are still possible
  $\Rightarrow$ explicit authentication: key confirmation rounds
  MACs using a key derived from $K$



# Explicit Authentication




- Two keys ($Ke$ and $Km$) have to be derived from the common secret $K$



# Key Derivation A Classical Technique

- The usual way for the key derivation
  $K \rightarrow Ke, Km$ is
  - $Ke = \text{PRF}_K(0)$
  - $Km = \text{PRF}_K(1)$
- $K = g^{xy}$ is a random element in the group,
  (under the Decisional Diffie-Hellman assumption),
  but not a random bit-string in $\{0,1\}^n$
  - While this is a requirement for the PRF security!



# Overview

- Authenticated Diffie-Hellman Key Exchange
- Security Model
- Usual Flaw in the Security Analysis
- The Twist-Augmented Approach

# Security Model

Two parties (Alice and Bob) agree on a ***common*** secret key $Ke$, in order to establish a secret channel

- Intuitive goal: ***implicit authentication***
  - only the intended partners can compute the session key
- Formally: ***semantic security***
  - the session key $Ke$ is indistinguishable from a random string $r$, to anybody else



# Semantic Security

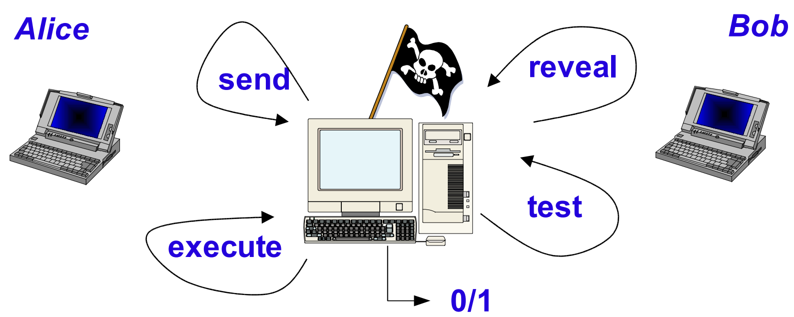
- For breaking the semantic security, the adversary asks one **test**-query which is answered, according to a random bit $b$, by
  - the actual secret key $Ke$ (if $b=0$)
  - a random bit-string $r$ (if $b=1$)

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



# Security Model

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



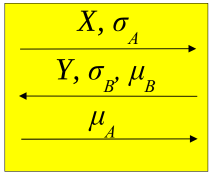

But one **test**-query, with $b$ to be guessed...



# Overview

- Authenticated Diffie-Hellman Key Exchange
- Security Model
- Usual Flaw in the Security Analysis
- The Twist-Augmented Approach

# Security Analysis




- Key derivation from $K=g^{xy}$
  - $Ke$ = PRF$_K(0)$
  - $Km$ = PRF$_K(1)$
- Usual security analysis [SigMa:Kr02]
  - REAL: $K=g^{xy}$ $Ke=$PRF$_K(0)$ $Km=$PRF$_K(1)$
  - RPRF: $K=rand$ $Ke=$PRF$_K(0)$ $Km=$PRF$_K(1)$
  - ALLR: $Ke=rand$ $Km=rand$
  - HYBR: $K=rand$ $Ke=rand$ $Km=$PRF$_K(1)$
  - RAND: $K=g^{xy}$ $Ke=rand$ $Km=$PRF$_K(1)$



# Security Analysis: Intuition

- **REAL**: $K=g^{xy}$ $Ke=$PRF$_K(0)$ $Km=$PRF$_K(1)$
  - This the real attack game
- **RPRF**: $K=rand$ $Ke=$PRF$_K(0)$ $Km=$PRF$_K(1)$
  - DDH assumption
- **ALLR**: $Ke=rand$ $Km=rand$
  - PRF property (2 queries), since $K=rand$
- **HYBR**: $K=rand$ $Ke=rand$ $Km=$PRF$_K(1)$
  - PRF property (1 query), since $K=rand$
- **RAND**: $K=g^{xy}$ $Ke=rand$ $Km=$PRF$_K(1)$
  - DDH assumption

$\Rightarrow$ Ideal attack: advantage = 0



# Security Analysis: Flaw

- **REAL**: $K=g^{xy}$ $Ke=$PRF$_K(0)$ $Km=$PRF$_K(1)$
  - This the real attack game
- **RPRF**: $K=rand$ $Ke=$PRF$_K(0)$ $Km=$PRF$_K(1)$
  - DDH assumption: $K$ random in the group
- **ALLR**: $Ke=rand$ $Km=rand$
  - PRF property (2 queries), since $K$ random bit-string
- Idem between **ALLR-HYBR** & **HYBR-RAND**

$\Rightarrow$ One more step is needed: derive a random bit-string from a random group element



# Random Group Element vs. Random Bit String

- The DDH assumption just says that $(g^x,g^y,g^{xy})$ and $(g^x,g^y,g^z)$ are indistinguishable
- But $(g^x,g^y,g^{xy})$ and $(g^x,g^y,R)$ (for a random bit string $R$) are not indistinguishable:
  - If the group is of even order, Legendre's symbol helps to distinguish them

## The Leftover Hash Lemma

- Family of Universal Hash Functions $(H_r)_r$
- Leftover Hash Lemma (LHL)
  - $(H_r(g^z),r) \approx (R,r)$ , statistically indistinguishable: the bias is bounded by $2^{-(e+1)}$
    - if $g^z$ has an entropy of $m$ bits
    - $H_r$: $\{0,1\}^n \rightarrow \{0,1\}^{m-2e}$, uniformly drawn from $(H_r)_r$
    - $R$ uniformly drawn from $\{0,1\}^{m-2e}$

E.g. One wants to extract 160 bits ($m$-2$e$ = 160), with bias $2^{-80}$ ($e$ = 80) $\Rightarrow m = 320$



## Improvements

- Main drawback of the LHL:
  - For practical requirements, the order of the group has to be quite large
- 1st Improvement: [GKR] – Eurocrypt '04
  - $(r, g, g^x, g^y, H(r,g^{xy})) \approx (r, g, g^x, g^y, H(r,g^z))$
  - Non-standard assumption: Hash-Diffie-Hellman Assumption
- 2nd Improvement: [DGHKR] – Crypto '04
  - Cascade methods (E.g. CBC, HMAC)
  - Non-standard assumption: Some primitives are ideal = random

$\Rightarrow$ Ideal-cipher/random-oracle model



## Overview

- Authenticated Diffie-Hellman Key Exchange
- Security Model
- Usual Flaw in the Security Analysis
- The Twist-Augmented Approach



## Elliptic Curve and Quadratic Twist

- Elliptic curve

$$\mathrm{E}_{a,b} = \{(x,y) \mid y^2 = x^3 + ax + b \bmod p\}$$

- Quadratic twist, for some $c \notin \mathrm{QR}(\mathbf{F}_p)$

$$\mathbf{E}_{a,b} = \{(x,y) \mid cy^2 = x^3 + ax + b \bmod p\}$$

- Let $x$ be an element in $\mathbf{F}_p$
  - If $x^3 + ax + b \in \mathrm{QR}(\mathbf{F}_p)$, there is $y \in \mathbf{F}_p$ such that $\mathrm{Q} = (x,y) \in \mathrm{E}_{a,b}$
  - Else, $c\,(x^3 + ax + b) \in \mathrm{QR}(\mathbf{F}_p)$, there is $y \in \mathbf{F}_p$ such that $\mathrm{Q} = (x,y) \in \mathbf{E}_{a,b}$

# Elliptic Curve and Quadratic Twist

$X = \{x \mid (x,y) \in E_{a,b}\}$ and $\mathbf{X} = \{x \mid (x,y) \in \mathbf{E}_{a,b}\}$

$\mathbf{F}_p = X \cup \mathbf{X}$

- Hasse's Theorem: $\#X \approx \#\mathbf{X} \approx p / 2$ (bias in $\sqrt{p}$)
- Random points $P$,$\mathbf{Q}$ $\rightarrow$ random scalar $x$
  - $P$ ($\mathbf{Q}$ resp.) a random point on $E_{a,b}$ ($\mathbf{E}_{a,b}$ resp.)
    - $x_P$ ($x_{\mathbf{Q}}$ resp.) is randomly distributed in $X$ ($\mathbf{X}$ resp.)
  - One flips a bit $b$: $b=0 \Rightarrow x=x_P$, else $x=x_{\mathbf{Q}}$
  - $x$ is "almost" uniformly distributed in $\mathbf{F}_p$
    the bias is bounded by $1/\sqrt{p}$



# Elliptic Curve and Quadratic Twist

- Random points $P$,$\mathbf{Q}$
  $\rightarrow$ random scalar $x$ in $\mathbf{F}_p$
  (bias bounded by $1/\sqrt{p}$)
- Random scalar $x$
  $\rightarrow$ random bit string $s$ in $\{0,1\}^k$
  - With a particular $p$: if $|2^k\text{-}p| \leq \sqrt{p}$
    (bias bounded by $1/\sqrt{p}$)

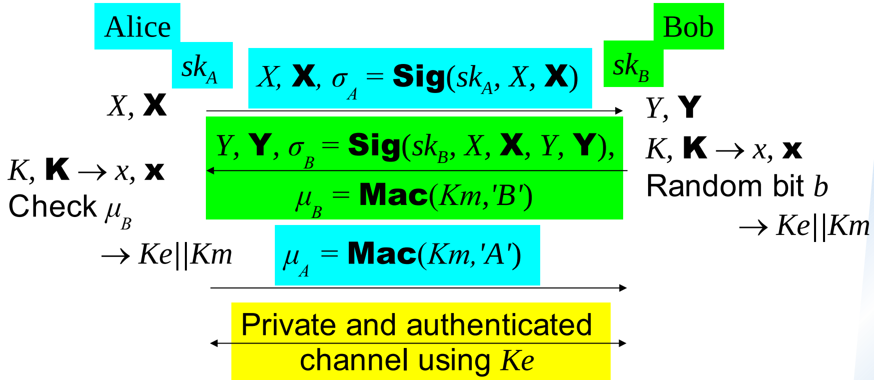


# TAU: Twist AUgmentation

- From any AKE scheme:
  - One runs 2 executions in parallel
    - One on the curve $E_{a,b} \rightarrow K$
    - One on the twist $\mathbf{E}_{a,b} \rightarrow \mathbf{K}$
  - One randomly chooses between $x_K$ and $x_{\mathbf{K}}$
  - One gets a random bit-string, a $k$-bit long string
    where $k$ is the bit-length of $p$
- With a 160-bit finite field,
  one gets a random 160-bit string
  (with a bias bounded by $2^{-80}$)



# Explicit Authentication




The two keys ($Ke$ and $Km$) are bit-strings
"almost" uniformly distributed,
under the DDH assumption only

# Conclusion

- Key derivation for AKE
  - Flaw in the usual technique
- New practical alternative to the LHL
  - Under the DDH assumption
  - In the standard model