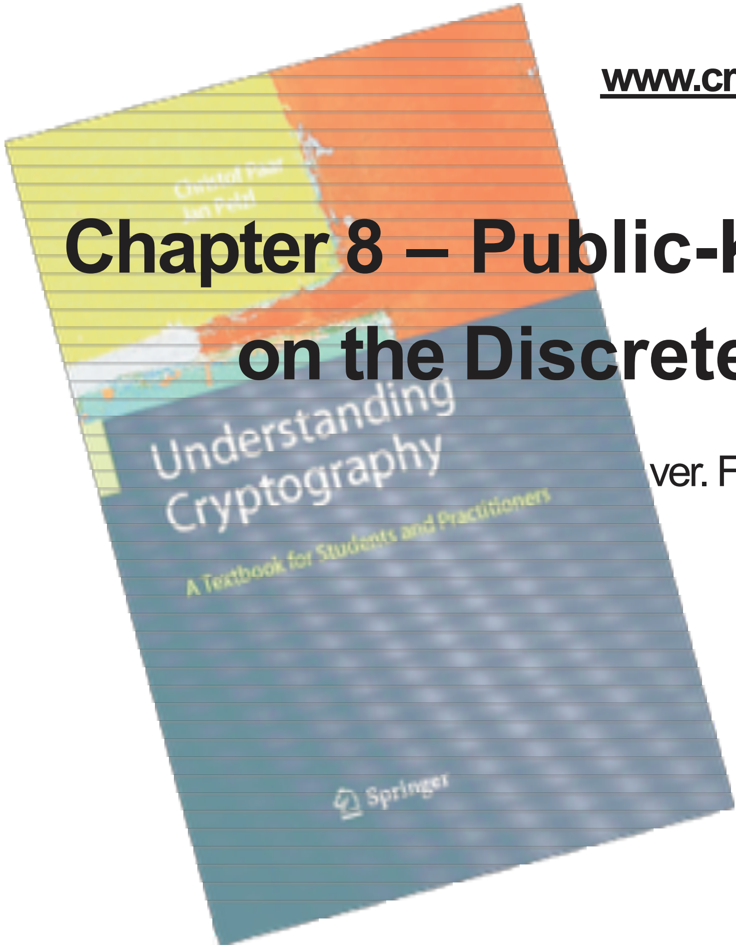
Understanding Cryptography

by Christof Paar and Jan Pelzl

www.crypto-textbook.com



Chapter 8 – Public-Key Cryptosystems Based on the Discrete Logarithm Problem

ver. February 22, 2010

First draft, published 10-23-17

These slides were prepared by Christof Paar and Jan Pelzl and modified by Sam Bowne

- **Some legal stuff (sorry): Terms of Use**

Contents of this Chapter

8.1 Diffie–Hellman Key Exchange

Diffie–Hellman Key Exchange: Overview

- Proposed in 1976 by **Whitfield Diffie and Martin Hellman**
- **Widely used**, e.g. in Secure Shell (SSH), Transport Layer Security (TLS), and Internet Protocol Security (IPSec)
- The Diffie–Hellman Key Exchange (DHKE) is a key exchange protocol and **not** used for encryption

  (For the purpose of encryption based on the DHKE, ElGamal can be used.)

Diffie–Hellman Key Exchange: Set-up

1. Choose a large prime $\boldsymbol{p}$
2. Choose an integer $\boldsymbol{\alpha} \in \{2,3, \ldots, p-2\}$
3. Publish $\boldsymbol{p}$ *and* $\boldsymbol{\alpha}$

Essential idea:

Choose two random secrets **a** and **b**

$(\alpha^a)^b \bmod p = (\alpha^b)^a \bmod p$

Both parties can calculate that value without sending secrets over the wire

Diffie–Hellman Key Exchange

Alice

Choose random private key
$k_{prA}=a \in \{1,2,\ldots,p-1\}$

Compute corresponding public key
$k_{pubA} = A = \alpha^a \bmod p$

$A \longrightarrow$

Bob

Choose random private key
$k_{prB}=b \in \{1,2,\ldots,p-1\}$

Compute correspondig public key
$k_{pubB} = B = \alpha^b \bmod p$

$\longleftarrow B$

Alice

Compute common secret
$k_{AB} = B^a = (\alpha^a)^b \bmod p$

Bob

Compute common secret
$k_{AB} = A^b = (\alpha^b)^a \bmod p$

We can now use the joint key k_{AB} for encryption, e.g., with AES

$y = AES_{kAB}(x)$ $\xrightarrow{y}$ $x = AES^{-1}_{kAB}(y)$

Diffie–Hellman Key Exchange: Example

Domain parameters
p=29, α=2

Alice

Choose random private key

a = 5

Compute public key

$A = \alpha^a = 2^5 = 3 \bmod 29$

$\xrightarrow{A}$

$\xleftarrow{B}$

Bob

Choose random private key

b = 12

Compute public key

$B = \alpha^b = 2^{12} = 7 \bmod 29$

Compute common secret

$k_{AB} = B^a = 7^5 = 16 \bmod 29$

Compute common secret

$k_{AB} = A^b = 3^{12} = 16 \bmod 29$

8.2 Some Algebra -- SKIP THIS SECTION

8.3 The Discrete Logarithm Problem

The Discrete Logarithm Problem

Definition 8.3.1 Discrete Logarithm Problem (DLP) in $\mathbb{Z}_p^*$
Given is the finite cyclic group $\mathbb{Z}_p^*$ of order $p-1$ and a primitive element $\alpha \in \mathbb{Z}_p^*$ and another element $\beta \in \mathbb{Z}_p^*$. The DLP is the problem of determining the integer $1 \leq x \leq p-1$ such that:

$$\alpha^x \equiv \beta \mod p$$

A **primitive element** is a number α

that can generate all the values in the group

when raised to all the powers from 1 to p-1

For prime p, all elements are primitive except 1

Example: mod 7

3 is a ***primitive element*** or ***generator*** under the ***multiplication*** operation

3^1 = 3 mod 7

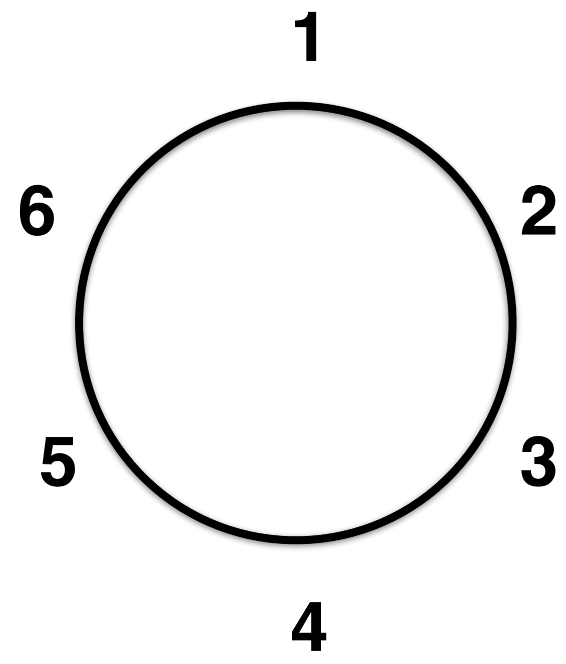
$3^2 = 9$ = 2 mod 7

$3^3 = 27$ = 6 mod 7

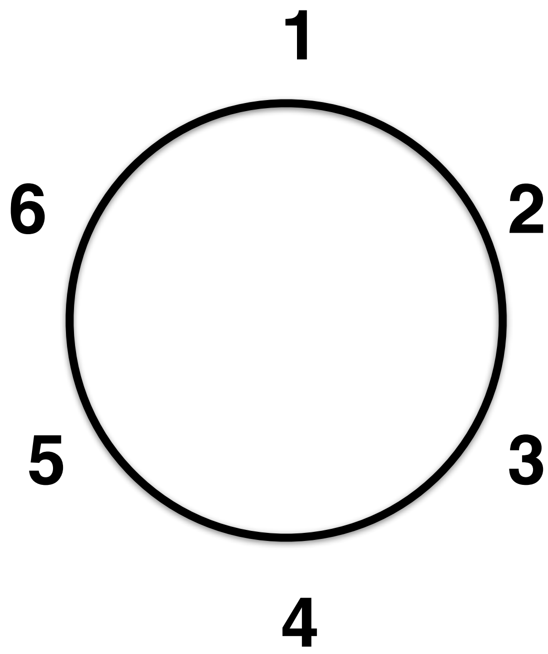
$3^4 = 81$ = 4 mod 7

$3^5 = 243$ = 5 mod 7

$3^6 = 729$ = 1 mod 7

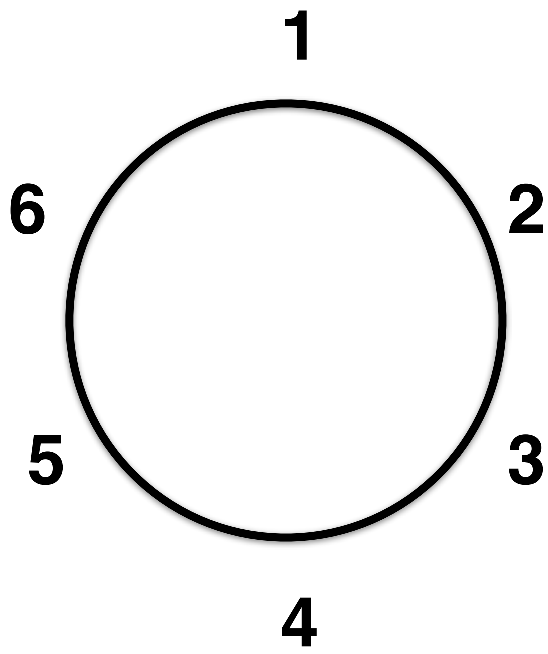


Example: mod 7




```
>>> for i in range(1,7):
...     print 3, "**", i, "= ", (3**i) % 7, "mod 7"
...
3 ** 1 =  3 mod 7
3 ** 2 =  2 mod 7
3 ** 3 =  6 mod 7
3 ** 4 =  4 mod 7
3 ** 5 =  5 mod 7
3 ** 6 =  1 mod 7
```

Example: mod 7




```
>>> for i in range(1,7):
...     print 3, "**", i, "= ", (3**i) % 7, "mod 7"
...
3 ** 1 =  3 mod 7
3 ** 2 =  2 mod 7
3 ** 3 =  6 mod 7
3 ** 4 =  4 mod 7
3 ** 5 =  5 mod 7
3 ** 6 =  1 mod 7
```

$\boldsymbol{\alpha} = 3$

DLP: $3^x = 4 \bmod 7$ $\quad$ $x = 4$

DLP: $3^x = 1 \bmod 7$ $\quad$ $x = 6$

Kahoot!

Discrete Logarithm Problems Used in Cryptography

1. The **multiplicative group of the prime field** Z_p *or a subgroup of it.*

   The classical **DHKE** uses this group, also **Elgamal** encryption and the Digital Signature Algorithm (**DSA**).

2. The cyclic group formed by an **elliptic curve**

3. The multiplicative group of a **Galois field** $GF(2^m)$ or a subgroup of it. *S*chemes such as the DHKE can be realized with them.

4. **Hyperelliptic curves** or **algebraic varieties**, which can be viewed as generalization of elliptic curves.

Groups 1 and 2 are most often used in practice.

Attacks against the Discrete Logarithm Problem

- Generic algorithms: Work in any cyclic group
  - Brute-Force Search
  - Shanks‘ Baby-Step-Giant-Step Method
  - Pollard‘s Rho Method
  - Pohlig-Hellman Method
- Non-generic Algorithms: Work only in specific groups, in particular in Z_p
  - The Index Calculus Method

Generic algorithms

- Brute-Force Search
- Shanks‘ Baby-Step-Giant-Step Method
  - Time-memory trade-off
- Pollard‘s Rho Method
  - Probabilistic, as fast as Shank's but doesn't require much memory, based on the Birthday paradox
  - Best known algorithm for elliptic curves

Generic algorithms

- Pohlig-Hellman Method
  - Based on the Chinese Remainder Theorem
  - Factor group order (size) into prime components
  - Attack each subgroup separately
  - To prevent this attack, the group order must have its largest prime factor in the range of 2^{160}

Chinese remainder theorem

From Wikipedia, the free encyclopedia

The **Chinese remainder theorem** is a theorem of number theory, which states that if one knows the remainders of the Euclidean division of an integer n by several integers, then one can determine uniquely the remainder of the division of n by the product of these integers, under the condition that the divisors are pairwise coprime.

The theorem was first discovered in the 3rd century AD by the Chinese mathematician Sunzi in *Sunzi Suanjing*.

The Chinese remainder theorem is widely used for computing with large integers, as it allows replacing a computation for which one knows a bound on the size of the result by several similar computations on small integers.

Nongeneric algorithm

- The **Index Calculus** Method
- Works only in specific groups, in particular in Z_p
- Works on many discrete logarithm encryption schemes
- Does **not work** on **elliptic curve** systems

Attacks against the Discrete Logarithm Problem

Summary of records for computing discrete logarithms in Z_p*

Integers modulo p with Number Sieve (Index Calculus)		
Year	# digits	# bits
2005	130	431
2007	160	530
2014	180	596
2016	232	768

In order to prevent attacks that compute the DLP, it is recommended to use primes with a length of at least 1024 bits for schemes such as Diffie-Hellman in Z_p*

Attacks against the Discrete Logarithm Problem

Elliptic curves challenges with key sizes of 108 and 109 bits have been solved

But no solutions are known for 131-bit keys

8.4 Security of the Diffie–Hellman Key Exchange

Active Attacker: Man-in-the-Middle

MITM can defeat the system, lying to both parties
Attacker chooses his own **a** and **b** and completes the DHE with both parties

Alice		Bob
Choose random private key $k_{prA}=a \in \{1,2,\ldots,p-1\}$		Choose random private key $k_{prB}=b \in \{1,2,\ldots,p-1\}$
Compute corresponding public key $k_{pubA}= A = \alpha^a \bmod p$	A →	
	← B	Compute correspondig public key $k_{pubB}= B = \alpha^b \bmod p$
Compute common secret $k_{AB} = B^a = (\alpha^a)^b \bmod p$		Compute common secret $k_{AB} = A^b = (\alpha^b)^a \bmod p$

Passive Attacker (Listening Only)

- Which information does Oscar have?
  - $\boldsymbol{\alpha}$, $\boldsymbol{p}$ (they are public)
  - $k_{pubA} = \boldsymbol{A} = \alpha^a \bmod p$
  - $k_{pubB} = \boldsymbol{B} = \alpha^b \bmod p$
- Which information does Oscar want to have?
  - $k_{AB} = \alpha^{ba} = \alpha^{ab} = \bmod\ p$
  - This is kown as Diffie-Hellman Problem (DHP)
- The only known way to solve the DHP is to solve the DLP, i.e.
  1. Compute $a = log_\alpha A \bmod p$
  2. *Compute* $k_{AB} = B^a = \alpha^{ba} = \bmod\ p$

  It is conjectured that the DHP and the DLP are equivalent, i.e., solving the DHP implies solving the DLP.
- To prevent attacks, i.e., to prevent that the DLP can be solved, choose $p > 2^{1024}$

8.5 The Elgamal Encryption Scheme

The Elgamal Encryption Scheme: Overview

- Proposed by Taher Elgamal in 1985
- Can be viewed as an extension of the DHKE protocol
- Based on the intractability of the discrete logarithm problem and the Diffie–Hellman problem

The Elgamal Encryption Scheme: Principle

Alice		Bob
		choose $d = k_{prB} \in \{2,\ldots,p-2\}$
		compute $\beta = k_{pubB} = \alpha^d \bmod p$
	$\xleftarrow{\quad\beta\quad}$	
choose i = $k_{prA} \in \{2,\ldots,p-2\}$		
compute ephemeral key $k_E = k_{pubA} = \alpha^i \bmod p$	$\xrightarrow{\quad k_E\quad}$	
		compute $k_M = k_E^d \bmod p$
compute $k_M = \beta^i \bmod p$		
encrypt message $x \in Z_p^*$: $y = x \cdot k_M \bmod p$	$\xrightarrow{\quad y\quad}$	
		decrypt $x = y \cdot k_M^{-1} \bmod p$

This looks very similar to the DHKE! The actual Elgamal protocol re-orders the computations which helps to save one communication (cf. next slide)

The Elgamal Encryption Protocol

Alice | **Bob**

Bob:

choose large prime p

choose primitive element $\alpha \in Z_p^*$
or in a subgroup of Z_p^*

choose $d = k_{prB} \in \{2,\dots,p\text{-}2\}$

compute $\beta = k_{pubB} = \alpha^d \bmod p$

$$k_{pubB} = (p, \alpha, \beta) \quad \longleftarrow$$

Alice:

choose i = $k_{prA} \in \{2,\dots,p\text{-}2\}$

compute $k_E = k_{pubA} = \alpha^i \bmod p$

compute masking key $k_M = \beta^i \bmod p$

encrypt message $x \in Z_p^*$:
$y = x \cdot k_M \bmod p$

$$(k_E, y) \quad \longrightarrow$$

Bob:

compute masking key $k_M = k_E^{\,d} \bmod p$
decrypt $x = y \cdot k_M^{-1} \bmod p$

Computational Aspects

- Key Generation
  - Generation of prime p
  - p has to be at least 1024 bits long
- Encryption
  - Requires two modular exponentiations and a modular multiplictation
  - All operands have a bit length of $\log_2 p$
  - Efficient execution requires methods such as the square-and-multiply algorithm
- Decryption
  - Requires one modular exponentiation and one modular inversion
  - As shown *in Understanding Cryptography*, the inversion can be computed from the ephemeral key

Security

- Passive attacks
  - Attacker eavesdrops p, α, $\beta = \alpha^d$, $k_E = \alpha^i$, $y = x \cdot \beta^i$ and wants to recover x
  - Problem relies on the DLP
  - Key must be at leasst **1024 bits** long
- Active attacks
  - MITM attack defeats it
  - An attack is also possible if the secret exponent i is being used more than once
  - If attacker can guess the plaintext of one message, it can be used to decrypt another message using the same key

Lessons Learned

- The Diffie–Hellman protocol is a widely used method for key exchange. It is based on cyclic groups.
- The discrete logarithm problem is one of the most important one-way functions in modern asymmetric cryptography. Many public-key algorithms are based on it.
- For the Diffie–Hellman protocol in Z_p*, *the prime p should be at least 1024 bits* long. This provides a security roughly equivalent to an 80-bit symmetric cipher.
- For a better long-term security, a prime of length 2048 bits should be chosen.
- The Elgamal scheme is an extension of the DHKE where the derived session key is used as a multiplicative masked to encrypt a message.
- Elgamal is a probabilistic encryption scheme, i.e., encrypting two identical messages does not yield two identical ciphertexts.

Kahoot!