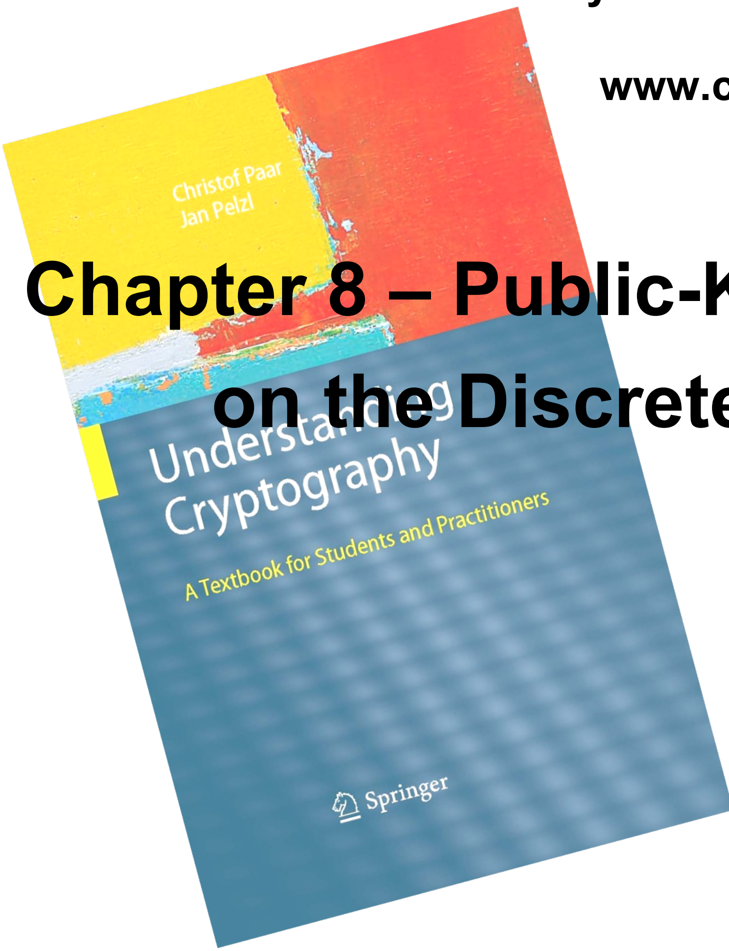
Understanding Cryptography

by Christof Paar and Jan Pelzl

www.crypto-textbook.com



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

These slides are a modified version of the slides prepared by Christof Paar and Jan Pelzl

Some legal stuff (sorry): Terms of Use

- The slides can used free of charge. All copyrights for the slides remain with Christof Paar and Jan Pelzl.
- The title of the accompanying book “Understanding Cryptography” by Springer and the author’s names must remain on each slide.
- If the slides are modified, appropriate credits to the book authors and the book title must remain within the slides.
- It is not permitted to reproduce parts or all of the slides in printed form whatsoever without written consent by the authors.

Content of this Chapter

- Diffie–Hellman Key Exchange
- The Discrete Logarithm Problem
- Security of the Diffie–Hellman Key Exchange
- The Elgamal Encryption Scheme

Groups

- To define (general) discrete logarithm problem, we need to define the notion of *cyclic groups*.
- Recall that an Abelian group is a set G with an operator °, satisfying:
  - The operation $\circ$ is **closed**: if $a, b \in G$ then $a \circ b \in G$.
  - **Associativity**: $a \circ (b \circ c) = (a \circ b) \circ c$
  - **Neutral element**: there is an element $\mathrm{e} \in G$ such that $a \circ e = a$ for all $a \in G$.
  - **Inverse**: for each $a \in G$ there is an inverse $a^{-1} \in G$ such that $a \circ a^{-1} = e$.
  - **Commutativity**: if $a \circ b \in G$ then $b \circ a \in G$.

Group Z_n^*

- Z_n^* be a set whose members are all integers i from 1 to $n-1$ for which $\gcd(i,n) = 1$ (i relative prime to n).
- Define ∘ to be the multiplication operator × modulo n.
- The neutral element is 1.
- Then Z_n^* is a group, with operator ∘ is a group.

Example

- $\mathbb{Z}_9^* = \{1,2,4,5,7,8\}$. All members are relative prime to 9.
- Multiplication table:

$\times \mathrm{mod}\ 9$	1	2	4	5	7	8
1	1	2	4	5	7	8
2	2	4	8	1	5	7
4	4	8	7	2	1	5
5	5	1	2	7	8	4
7	7	5	1	8	4	2
8	8	7	5	4	2	1

Example:

$5 \times 7 \bmod 9$

$= 35\ mod\ 9 = 8$

Order of an element

An order of an element a in a group Z_n^* is the smallest positive integer k such that

$$\underbrace{a \times a \times \cdots \times a}_{k \text{ times}} \equiv 1 \bmod n$$

Example: let $a = 3$ in the group Z_{11}^*

$$a^1 = 3$$
$$a^2 = 3 \times 3 = 9$$
$$a^3 = 9 \times 3 = 27 \equiv 5 \ (mod\ 11)$$
$$a^4 = 5 \times 3 = 15 \equiv 4 \ (mod\ 11)$$
$$a^5 = 4 \times 3 = 12 \equiv 1 \ (mod\ 11)$$

If we multiply a^5 further with a's, we'll cycle through the same numbers.

So the order of a is 5.

Cyclic group

- Let $|Z_n^*|$ denote the size of Z_n^*, that is, the number of elements in Z_n^*.
- A group Z_n^* which contains an element a with order $|Z_n^*|$ is called a **cyclic group**.
- Example: $Z_{11}^* = \{1,2,3,4,5,6,7,8,9,10\}$. Then $a = 2$ has order $|Z_{11}^*| = 10$. So $\mathbb{Z}_{11}^*$ is a cyclic group.

$a^1 = 2$	$a^6 \equiv 9 \ mod\ 11$
$a^2 = 4$	$a^7 \equiv 7 \ mod\ 11$
$a^3 = 8$	$a^8 \equiv 3 \ mod\ 11$
$a^4 \equiv 5 \ mod\ 11$	$a^9 \equiv 6 \ mod\ 11$
$a^5 \equiv 10 \ mod\ 11$	$a^{10} \equiv 1 \ mod\ 11$

Primitive element

- An element a of Z_n^* with order $|Z_n^*|$ is called a *primitive element*, or *a generator* of the group.
- Example: $a = 2$ in group Z_{11}^* is a generator of the group: all elements of Z_{11}^* can be generated by powers of a.
- **Fact**: for every prime p, Z_p^* is a cyclic group (i.e., it has a primitive element).

The Discrete Logarithm Problem

Discrete Logarithm Problem (DLP) in Z_p^*

- Given is the finite cyclic group Z_p^* of order $p-1$ and a primitive element $\alpha \in Z_p^*$ and another element $\beta \in Z_p^*$.
- The DLP is the problem of determining the integer $1 \le x \le p-1$ such that $\alpha^x \equiv \beta \bmod p$
- This computation is called the **discrete logarithm problem (DLP)**

$$x = \log_\alpha \beta \bmod p$$

Remark: For the coverage of groups and cyclic groups, we refer to Chapter 8 of *Understanding Cryptography*

Example

- $Z_{11}^* = \{1,2,3,4,5,6,7,8,9,10\}$ has a primitive element *a* = 2.

i	1	2	3	4	5	6	7	8	9	10
a^i	2	4	8	5	10	9	7	3	6	1

- Then, $5 = \log_2 10 \bmod 11$
- In general, for large *p*, it is infeasible to construct the table of exponents for all generators of Z_p^* like above.
- Encryption/digital signature based on discrete logarithm relies on the difficulty of finding the logarithm.

The Generalized Discrete Logarithm Problem

The following discrete logarithm problems have been proposed for use in cryptography

1. The multiplicative group of the prime field Z_p *or a subgroup of it.* For instance, the classical DHKE uses this group (cf. next slides), but also Elgamal encryption or the Digital Signature Algorithm (DSA).
2. The cyclic group formed by an elliptic curve (see Chapter 9)
3. The multiplicative group of a Galois field $GF(2^m)$ or a subgroup of it. Schemes such as the DHKE can be realized with them.
4. Hyperelliptic curves or algebraic varieties, which can be viewed as generalization of elliptic curves.

Remark: The groups 1. and 2. are most often used in practice.

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 p.
2. Choose an integer $\alpha \in \{2,3, . . . , p-2\}$. α must be a primitive element of the cyclic group Z_p^*
3. Publish p *and* α.

Diffie–Hellman Key Exchange

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 $k_{pubA} = A = \alpha^a \bmod p$	$\xrightarrow{A}$	
	$\xleftarrow{B}$	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 $k_{AB} = A^b = (\alpha^b)^a \bmod p$

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		Bob
Choose random private key $k_{prA}= a = 5$		Choose random private key $k_{prB}=b = 12$
Compute corresponding public key $k_{pubA}= A = 2^5 = 3 \bmod 29$	$\xrightarrow{\quad A \quad}$	
	$\xleftarrow{\quad B \quad}$	Compute correspondig public key $k_{pubB}= B = 2^{12} = 7 \bmod 29$
Compute common secret $k_{AB} = B^a = 7^5 = 16 \bmod 29$		Compute common secret $k_{AB} = A^b = 3^{12} = 16 \bmod 29$

Proof of correctness:

Alice computes: $B^a = (\alpha^b)^a \bmod p$
Bob computes: $A^b = (\alpha^a)^b \bmod p$

i.e., Alice and Bob compute the same key k_{AB}!

Attacks against the Discrete Logarithm Problem

- Security of many asymmetric primitives is based on the difficulty of computing the DLP in cyclic groups, i.e.,

  Compute x for a given α and β such that $\beta = \alpha \circ \alpha \circ \alpha \circ . . . \circ \alpha = \alpha^x$

- The following algorithms for computing discrete logarithms exist
  - 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
- Remark: Elliptic curves can only be attacked with generic algorithms which are weaker than non-generic algorithms. Hence, elliptic curves are secure with shorter key lengths than the DLP in prime fields Z_p

Attacks against the Discrete Logarithm Problem

Summary of records for computing discrete logarithms in Z_p^*

Decimal digits	Bit length	Date
58	193	1991
68	216	1996
85	282	1998
100	332	1999
120	399	2001
135	448	2006
160	532	2007

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^*

Security of the classical Diffie–Hellman Key Exchange

- Which information does Oscar have?
  - α, p
  - $k_{pubA} = A = \alpha^a \bmod p$
  - $k_{pubB} = B = \alpha^b \bmod p$
- Which information does Oscar want to have?
  - $k_{AB} = \alpha^{ba} = \alpha^{ab} = \bmod p$
  - This is kown as Diffie-Hellman Problem (DHP)

Security of the classical Diffie–Hellman Key Exchange

- 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 $p > 2^{1024}$

- However, DHKE is still vulnerable to impersonation attack. The above assumes k_{pub_A} and k_{pub_B} are authentic, i.e., they have not been changed by the attacker during transit.
  - This problem is solved using public key certificates – to be covered later.

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 $d = k_{prB} \in \{2,\ldots,p\text{-}2\}$
	$\xleftarrow{\ \beta\ }$	compute $\beta = k_{pubB} = \alpha^d \bmod p$
choose $i = k_{prA} \in \{2,\ldots,p\text{-}2\}$		
compute ephemeral key $k_E = k_{pubA} = \alpha^i \bmod p$	$\xrightarrow{\ k_E\ }$	
		compute $k_M = k_E^{\,d} \bmod p$
compute $k_M = \beta^i \bmod p$		
encrypt message $x \in Z_p^*$: $y = x \cdot k_M \bmod p$	$\xrightarrow{\ y\ }$	
		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
		choose large prime p
		choose primitive element $\alpha \in Z_p^*$ or in a subgroup of Z_p^*
		choose $d = k_{prB} \in \{2,\ldots,p-2\}$
		compute $\beta = k_{pubB} = \alpha^d \bmod p$
	$\xleftarrow{k_{pubB} = (p,\ \alpha,\ \beta)}$	
choose $i = k_{prA} \in \{2,\ldots,p-2\}$		
compute $k_E = k_{pubA} = \alpha^i \bmod p$		
compute masking key $k_M = \beta^i \bmod p$		
encrypt message $x \in Z_p^*$: $y = x \cdot k_M \bmod p$		
	$\xrightarrow{(k_E,\ y)}$	
		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 of size of at least 1024 bits
  - cf. Section 7.6 in *Understanding Cryptography* for prime-finding algorithms
- Encryption
  - Requires two modular exponentiations and a modular multiplication
  - All operands have a bitlength of $\log_2 p$
  - Efficient execution requires methods such as the square-and-multiply algorithm (cf. Chapter 7)
- 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 $p, \alpha, \beta = \alpha^d$, $k_E = \alpha^i$, $y = x \cdot \beta^i$ and wants to recover x
  - Problem relies on the DLP

- Active attacks
  - If the public keys are not authentic, an attacker could send an incorrect public key (cf. Chapter 13)
  - An Attack is also possible if the secret exponent i is being used more than once (cf. *Understanding Cryptography* for more details on the attack)

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.