WILFRID LAURIER UNIVERSITY
LAURIER
Inspiring Lives.

CP 460 - Applied Cryptography

Public-Key Cryptosystems Based on the Discrete Logarithm Problem

Department of Physics and Computer Science
Faculty of Science, Waterloo

Abbas Yazdinejad, Ph.D.

Fall 2024

Content of this Chapter

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

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.)

It's important to note that DHKE is not an encryption algorithm itself but a way to securely exchange keys that can be used for encryption, such as with the ElGamal encryption scheme.

Diffie–Hellman Key Exchange: Set-up

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

In DHKE, two parties (usually called Alice and Bob) agree on two public parameters: a large prime number p and a base α, which is a primitive element in the group Zp*. These parameters are published for both parties to use in generating their public and private keys.

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 $k_{pubA} = A = \alpha^a \bmod p$		Compute correspondig public key $k_{pubB} = B = \alpha^b \bmod p$
	A →	
	← B	
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

$$y = AES_{kAB}(x)$$

y →

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

Alice and Bob generate their private keys and corresponding public keys. Each party computes the common secret kAB using the other's public key. The resulting shared key kAB can then be used for encryption (e.g., AES encryption), ensuring secure communication between the two parties without having to directly exchange the encryption key itself.

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 $k_{pubA} = A = 2^5 = 3 \bmod 29$	A →	
	← B	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} !

This slide provides a numerical example of the DHKE process with specific values. Alice and Bob generate their private keys, compute the public keys, and arrive at the same shared secret using the public keys and the Diffie-Hellman computation process. It demonstrates the correctness of the protocol by showing that both Alice and Bob compute the same shared key.

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 $1 \leq x \leq p-1$ such that $\alpha^x \equiv \beta \bmod p$
- This computation is called the **discrete logarithm problem (DLP)**

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

- Example: Compute x for $5^x \equiv 41 \bmod 47$

The discrete logarithm problem (DLP) is the foundation of the security of DHKE. The DLP asks to find an exponent x such that α^x≡βmod p, where α is a known base, and β is a known result. Solving the DLP is computationally hard, which underpins the security of Diffie-Hellman.

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

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. previous 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. *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.

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

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 $\beta = \alpha \circ \alpha \circ \alpha \circ \ldots \circ \alpha = \alpha^x$

1. Brute Force Attack
In a brute-force attack, the attacker simply tries all possible values of x until the correct one is found. This approach is inefficient because the complexity grows exponentially with the size of the group.

2. Shanks' Baby-Step Giant-Step Algorithm
This is a **meet-in-the-middle attack** that reduces the number of guesses an attacker needs to make by using memory to store intermediate values.

3. Pollard's Rho Method
Pollard's Rho method is a **probabilistic algorithm** that efficiently computes discrete logarithms in groups. It uses the concept of pseudorandom sequences and is much faster than brute force.

4. Pohlig-Hellman Algorithm
The **Pohlig-Hellman algorithm** is designed to speed up the computation of discrete logarithms when the order of the group is not prime. It works by breaking the problem into smaller subproblems using the prime factors of the group's order.

5. Index Calculus Method
The **Index Calculus** is a specialized algorithm for solving the DLP in specific groups, particularly in **prime fields Zp∗**. It's one of the most powerful methods for attacking the DLP in large fields and exploits the structure of the multiplicative group of a prime field.

- 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

Comparison of Attack Methods

Attack Method	Time Complexity	Memory Usage	Best Used For
Brute Force	$O(p)$	Low	Small groups (impractical for cryptographic sizes)
Baby-Step Giant-Step	$O(\sqrt{p})$	High	Cryptographic applications where memory is not an issue
Pollard's Rho	$O(\sqrt{p})$	Low	Practical use for large groups, less memory intensive
Pohlig-Hellman	$O(\sqrt{q})$ where q is largest prime factor	Moderate	Groups where $p-1$ has small prime factors
Index Calculus	Sub-exponential, $O(e^{c\sqrt{\log p \log\log p}})$	High	Prime fields Z_p^* with large p

Each of these attacks targets the discrete logarithm problem differently, depending on the group in which the DLP is defined and the properties of the group order. Modern cryptographic protocols avoid these attacks by choosing large primes and using group structures (like elliptic curves) that resist these algorithms, making the computation of discrete logarithms infeasible within practical time frames.

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

It offers both confidentiality and authenticity, which makes it an important cryptosystem in modern cryptography.

Key Components of the Elgamal Encryption Scheme

1. Key Generation
The key generation process involves the creation of a public-private key pair

2. Encryption
To encrypt a message x (where $x \in Zp*$) using ElGamal, the sender (Alice) uses the recipient's public key.

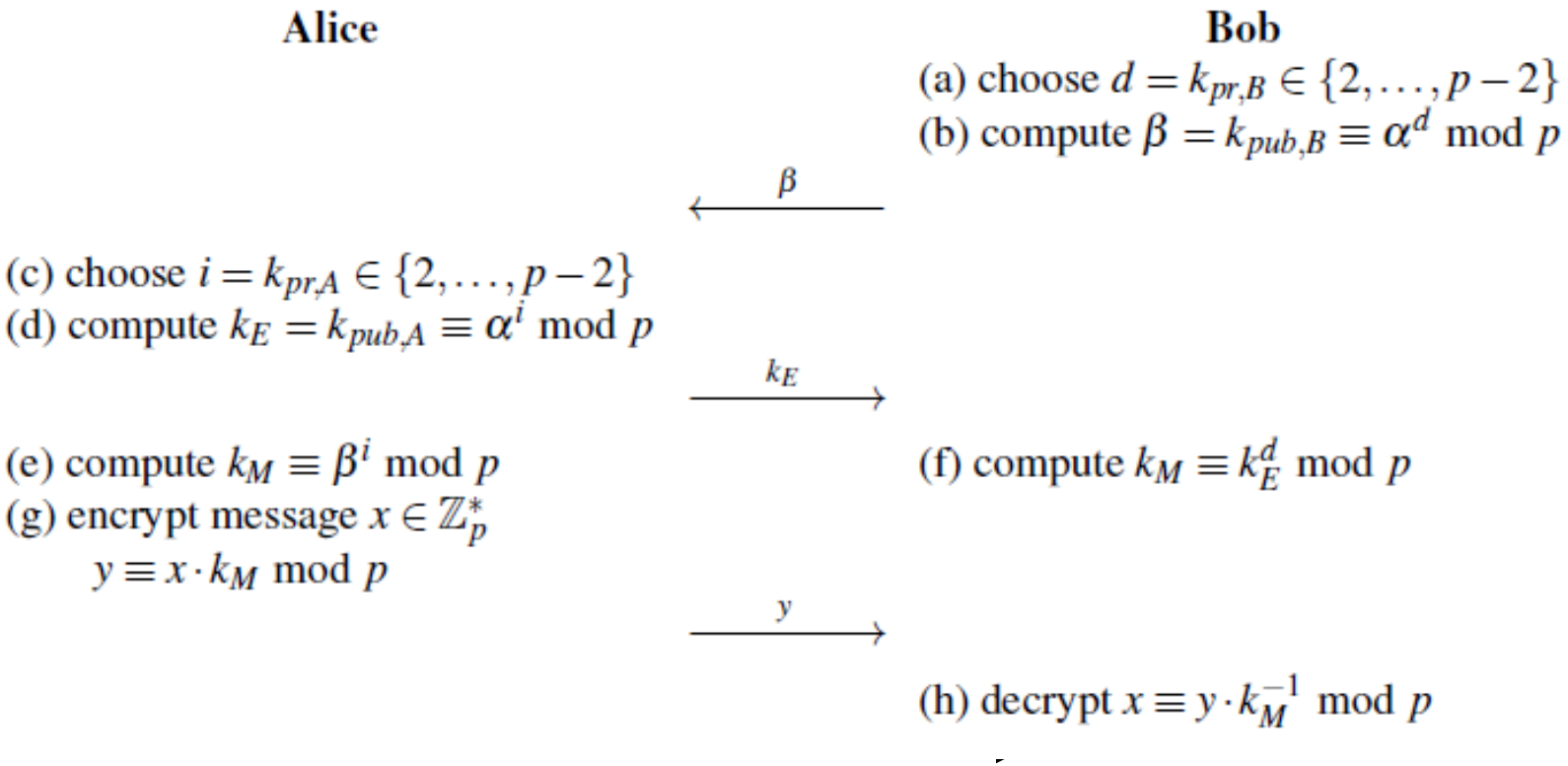
3. Decryption
When the recipient (Bob) receives the ciphertext (kE,y), he decrypts the message using his private key d

Security of ElGamal Encryption

The security of ElGamal encryption relies on two main problems:

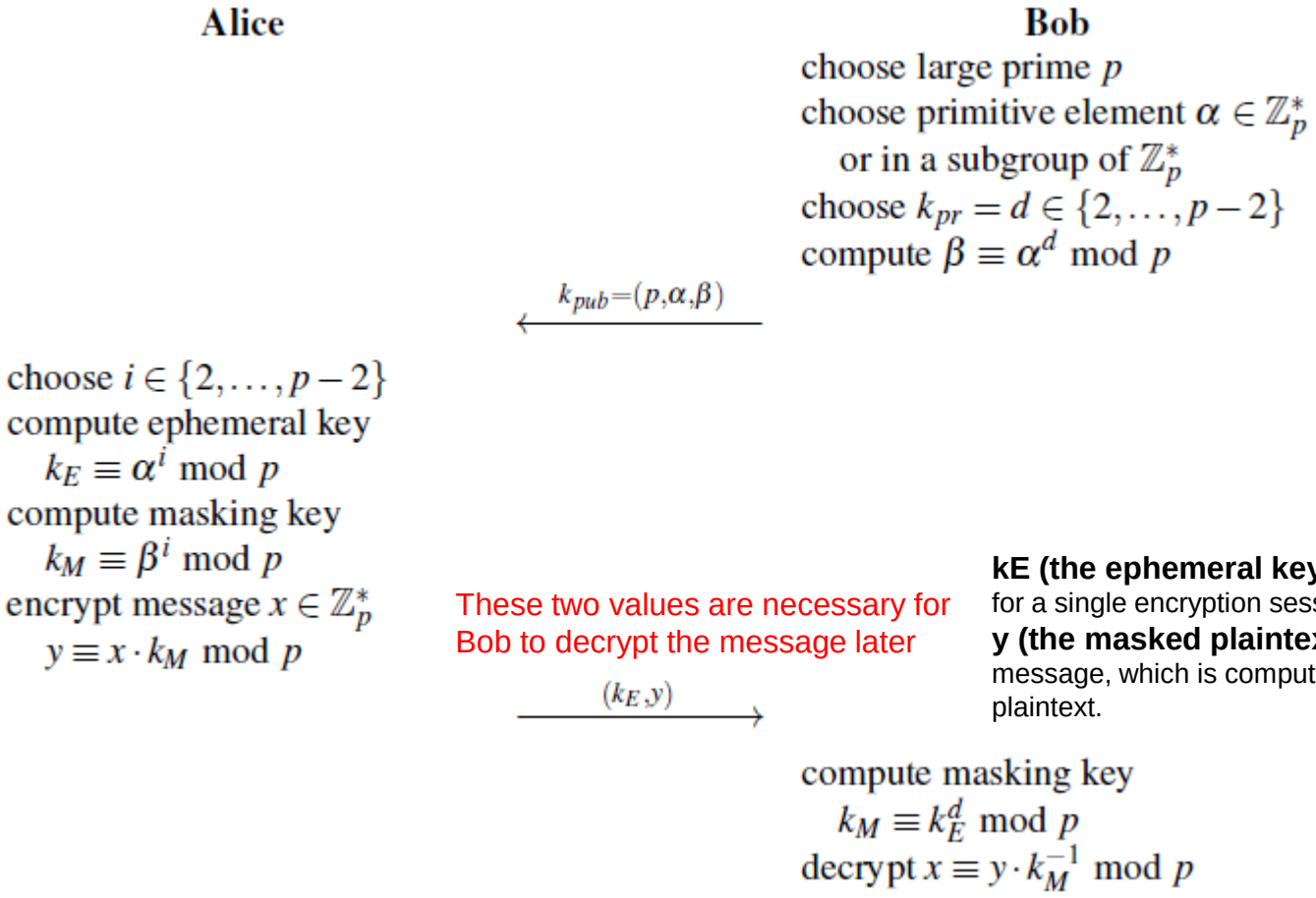
1. **Discrete Logarithm Problem (DLP)**
2. **Diffie-Hellman Problem (DHP)**

The Elgamal Encryption Scheme: Principle



The protocol consists of two phases, the classical DHKE (Steps a–f) followed by the message encryption and decryption (Steps g and h, respectively). Bob computes his private key d and public key b. This key pair does not change, i.e., it can be used to encrypt many messages. Alice, however, has to generate a new public/private-key pair for the encryption of every message. Her private key is denoted by i and her public key by k_E. The latter is an ephemeral, i.e, a temporary, key, hence the index "E". The joint key is denoted by k_M because it is used for masking the plaintext.

The Elgamal Encryption Protocol



These two values are necessary for Bob to decrypt the message later

kE (the ephemeral key): This is a temporary key used for a single encryption session.

y (the masked plaintext): This is the actual encrypted message, which is computed by masking the original plaintext.

The actual Elgamal encryption protocol rearranges the sequence of operations from the Diffie–Hellman–inspired approach we saw before. Now, Alice has to send only one message to Bob, as opposed to two messages in the earlier protocol. The ciphertext consists of two parts, the ephemeral key kE, and the masked plaintext y. In general, all parameters have a bit length of d log2 pe, hence the ciphertext (kE; y) is twice as long as the message. Thus, the message expansion factor of Elgamal encryption is two.

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 multiplictation
  - 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 modulare 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

- 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)

Advantages of ElGamal

- **Probabilistic Encryption:** ElGamal is a probabilistic encryption scheme, meaning that encrypting the same message twice with different random ephemeral keys will result in different ciphertexts. This makes it more resistant to certain cryptographic attacks like chosen ciphertext attacks.
- **Based on Hard Mathematical Problems:** The security of ElGamal depends on the DLP which is computationally hard problems.

Disadvantages of ElGamal

- **Ciphertext Expansion:** ElGamal produces ciphertexts that are twice as large as the plaintext, which can be inefficient in terms of storage and transmission.
- **Efficiency:** ElGamal requires more computational steps compared to some other cryptographic algorithms, making it slower, especially for large-scale systems.

Applications

- **Digital Signatures:** ElGamal is used as the basis for digital signature schemes, such as the **Digital Signature Algorithm (DSA)**.
- **Hybrid Encryption Schemes:** ElGamal is often used in combination with symmetric encryption in hybrid cryptosystems, where ElGamal is used to encrypt a symmetric key, which is then used to encrypt the actual message.


Practical Networking

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.


Thank
You