

# Session I: Discrete Logs and Cryptography

Simon R. Blackburn

Royal Holloway, University of London

5th December 2008

# Overview

Public Key Crypto

The Discrete Log Problem

Diffie–Hellman, ElGamal

How hard is the DLP?

Further information

# Public key cryptography

- ▶ Standard (symmetric key) cryptography needs users to share a secret key.

# Public key cryptography

- ▶ Standard (symmetric key) cryptography needs users to share a secret key.
- ▶ Can users who have never met before exchange a key?

# Public key cryptography

- ▶ Standard (symmetric key) cryptography needs users to share a secret key.
- ▶ Can users who have never met before exchange a key?
- ▶ **Yes:** key exchange (or key agreement) protocols.

# Public key cryptography

- ▶ Standard (symmetric key) cryptography needs users to share a secret key.
- ▶ Can users who have never met before exchange a key?
- ▶ **Yes:** key exchange (or key agreement) protocols.
- ▶ Is it possible to send an encrypted message to a user without ever interacting with them?

# Public key cryptography

- ▶ Standard (symmetric key) cryptography needs users to share a secret key.
- ▶ Can users who have never met before exchange a key?
- ▶ **Yes:** key exchange (or key agreement) protocols.
- ▶ Is it possible to send an encrypted message to a user without ever interacting with them?
- ▶ **Yes:** public key cryptosystems.

# Public key cryptography

- ▶ Standard (symmetric key) cryptography needs users to share a secret key.
- ▶ Can users who have never met before exchange a key?
- ▶ **Yes:** key exchange (or key agreement) protocols.
- ▶ Is it possible to send an encrypted message to a user without ever interacting with them?
- ▶ **Yes:** public key cryptosystems.
- ▶ Same techniques allow digital signatures.

# Discrete logarithm problem (DLP)

Let  $p$  be a prime and let  $g, h \in \mathbb{Z}_p^*$ .

Find an integer  $x$  (if it exists) such that  $h \equiv g^x \pmod{p}$ .

# Discrete logarithm problem (DLP)

Let  $p$  be a prime and let  $g, h \in \mathbb{Z}_p^*$ .

Find an integer  $x$  (if it exists) such that  $h \equiv g^x \pmod{p}$ .

In general, this is a hard computational problem (for large  $p$ ).

**Example:** Let  $p = 11$ ,  $g = 2$  and  $h = 9$ . Solve the DLP.

# Discrete logarithm problem (DLP)

Let  $p$  be a prime and let  $g, h \in \mathbb{Z}_p^*$ .

Find an integer  $x$  (if it exists) such that  $h \equiv g^x \pmod{p}$ .

In general, this is a hard computational problem (for large  $p$ ).

**Example:** Let  $p = 11$ ,  $g = 2$  and  $h = 9$ . Solve the DLP.

$x$	0	1	2	3	4	5	6	7	8	9
$g^x$	1	2	4	8	5	10	9	7	3	6

# Exponentiation is easy

Given  $g, h \in \mathbb{Z}_p$ , it is hard to find  $x$  such that  $h = g^x$ .

The ‘inverse’ problem (given  $g$  and  $x$ , to find  $h$ ) is easy.

To solve the inverse problem (i.e. exponentiation) efficiently, use the square and multiply algorithm.

## Diffie–Hellman key exchange

Suppose Alice and Bob want to agree on a random key  $K$ . They decide upon a large prime  $p$  and some  $g \in \mathbb{Z}_p^*$  and perform the following protocol:

## Diffie-Hellman key exchange

Suppose Alice and Bob want to agree on a random key  $K$ . They decide upon a large prime  $p$  and some  $g \in \mathbb{Z}_p^*$  and perform the following protocol:

- ▶ Alice chooses a random integer  $1 \leq a < p - 1$  and sends  $c_1 = g^a \pmod p$  to Bob.

## Diffie-Hellman key exchange

Suppose Alice and Bob want to agree on a random key  $K$ . They decide upon a large prime  $p$  and some  $g \in \mathbb{Z}_p^*$  and perform the following protocol:

- ▶ Alice chooses a random integer  $1 \leq a < p - 1$  and sends  $c_1 = g^a \pmod p$  to Bob.
- ▶ Bob chooses a random integer  $1 \leq b < p - 1$  and sends  $c_2 = g^b \pmod p$  to Alice.

## Diffie–Hellman key exchange

Suppose Alice and Bob want to agree on a random key  $K$ . They decide upon a large prime  $p$  and some  $g \in \mathbb{Z}_p^*$  and perform the following protocol:

- ▶ Alice chooses a random integer  $1 \leq a < p - 1$  and sends  $c_1 = g^a \pmod p$  to Bob.
- ▶ Bob chooses a random integer  $1 \leq b < p - 1$  and sends  $c_2 = g^b \pmod p$  to Alice.

Alice and Bob both share the same key  $K = g^{ab} \pmod p$ .

## Diffie–Hellman key exchange

Suppose Alice and Bob want to agree on a random key  $K$ . They decide upon a large prime  $p$  and some  $g \in \mathbb{Z}_p^*$  and perform the following protocol:

- ▶ Alice chooses a random integer  $1 \leq a < p - 1$  and sends  $c_1 = g^a \pmod p$  to Bob.
- ▶ Bob chooses a random integer  $1 \leq b < p - 1$  and sends  $c_2 = g^b \pmod p$  to Alice.
- ▶ On receiving  $c_2$  Alice computes  $K = c_2^a \pmod p$ .

Alice and Bob both share the same key  $K = g^{ab} \pmod p$ .

## Diffie–Hellman key exchange

Suppose Alice and Bob want to agree on a random key  $K$ . They decide upon a large prime  $p$  and some  $g \in \mathbb{Z}_p^*$  and perform the following protocol:

- ▶ Alice chooses a random integer  $1 \leq a < p - 1$  and sends  $c_1 = g^a \pmod p$  to Bob.
- ▶ Bob chooses a random integer  $1 \leq b < p - 1$  and sends  $c_2 = g^b \pmod p$  to Alice.
- ▶ On receiving  $c_2$  Alice computes  $K = c_2^a \pmod p$ .
- ▶ On receiving  $c_1$  Bob computes  $K = c_1^b \pmod p$ .

Alice and Bob both share the same key  $K = g^{ab} \pmod p$ .

# The Diffie–Hellman problem

- ▶ A (passive) eavesdropper cannot determine  $K$  unless they can solve the following computational problem:

# The Diffie-Hellman problem

- ▶ A (passive) eavesdropper cannot determine  $K$  unless they can solve the following computational problem:
- ▶ **Diffie-Hellman problem (DHP)**: Given the triple  $(g, g^a, g^b)$  of elements of  $\mathbb{Z}_p^*$ , compute  $g^{ab} \pmod p$ .

# The Diffie-Hellman problem

- ▶ A (passive) eavesdropper cannot determine  $K$  unless they can solve the following computational problem:
- ▶ **Diffie-Hellman problem (DHP):** Given the triple  $(g, g^a, g^b)$  of elements of  $\mathbb{Z}_p^*$ , compute  $g^{ab} \pmod p$ .
- ▶ Let  $p = 11$ . What is the solution to the DHP  $(2, 4, 7)$ ?

# The Diffie-Hellman problem

- ▶ A (passive) eavesdropper cannot determine  $K$  unless they can solve the following computational problem:
- ▶ **Diffie-Hellman problem (DHP):** Given the triple  $(g, g^a, g^b)$  of elements of  $\mathbb{Z}_p^*$ , compute  $g^{ab} \pmod p$ .
- ▶ Let  $p = 11$ . What is the solution to the DHP  $(2, 4, 7)$ ?
- ▶ Since  $4 = 2^2 \pmod{11}$ , we have  $a = 2$ . So  $g^{ab} = (g^b)^2 = 7^2 = 5 \pmod{11}$ .

# The Diffie-Hellman problem

- ▶ A (passive) eavesdropper cannot determine  $K$  unless they can solve the following computational problem:
- ▶ **Diffie-Hellman problem (DHP):** Given the triple  $(g, g^a, g^b)$  of elements of  $\mathbb{Z}_p^*$ , compute  $g^{ab} \pmod p$ .
- ▶ Let  $p = 11$ . What is the solution to the DHP  $(2, 4, 7)$ ?
- ▶ Since  $4 = 2^2 \pmod{11}$ , we have  $a = 2$ . So  $g^{ab} = (g^b)^2 = 7^2 = 5 \pmod{11}$ .
- ▶ To solve the DHP this way, you have to solve one DLP. Is there another way, avoiding the DLP?

## ElGamal public key encryption

**Key Generation:** Alice generates a large prime  $p$  and an element  $g \in \mathbb{Z}_p^*$ .

She chooses a random integer  $0 < a < p - 1$  and sets  $h = g^a \bmod p$ .

She publishes  $(p, g, h)$  and keeps  $a$  secret.

## ElGamal public key encryption

**Key Generation:** Alice generates a large prime  $p$  and an element  $g \in \mathbb{Z}_p^*$ .

She chooses a random integer  $0 < a < p - 1$  and sets  $h = g^a \bmod p$ .

She publishes  $(p, g, h)$  and keeps  $a$  secret.

**To encrypt** a message  $m \in \mathbb{Z}_p$ : Bob chooses a random  $0 < b < (p - 1)$ . He sets  $c_1 = g^b \bmod p$  and  $c_2 = mh^b \bmod (p)$ . He sends  $(c_1, c_2)$  to Alice.

## ElGamal public key encryption

**Key Generation:** Alice generates a large prime  $p$  and an element  $g \in \mathbb{Z}_p^*$ .

She chooses a random integer  $0 < a < p - 1$  and sets  $h = g^a \bmod p$ .

She publishes  $(p, g, h)$  and keeps  $a$  secret.

**To encrypt** a message  $m \in \mathbb{Z}_p$ : Bob chooses a random  $0 < b < (p - 1)$ . He sets  $c_1 = g^b \bmod p$  and  $c_2 = mh^b \bmod (p)$ . He sends  $(c_1, c_2)$  to Alice.

**To decrypt:** Alice calculates  $c_2 c_1^{-a} = mh^b g^{-ab} = m$ .

# How hard is the DLP

- ▶ Naive search:  $p$  operations (a bad algorithm).

## How hard is the DLP

- ▶ Naive search:  $p$  operations (a bad algorithm).
- ▶ Silver–Pohlig–Hellman method: DLP is easy if  $p - 1$  has no large prime factors.

## How hard is the DLP

- ▶ Naive search:  $p$  operations (a bad algorithm).
- ▶ Silver–Pohlig–Hellman method: DLP is easy if  $p - 1$  has no large prime factors.
- ▶ Baby-step giant-step algorithm:  $\sqrt{p}$  operations and  $\sqrt{p}$  memory.

## How hard is the DLP

- ▶ Naive search:  $p$  operations (a bad algorithm).
- ▶ Silver–Pohlig–Hellman method: DLP is easy if  $p - 1$  has no large prime factors.
- ▶ Baby-step giant-step algorithm:  $\sqrt{p}$  operations and  $\sqrt{p}$  memory.
- ▶ Index calculus, and number field sieve: better (sub-exponential) complexity.

## How hard is the DLP

- ▶ Naive search:  $p$  operations (a bad algorithm).
- ▶ Silver–Pohlig–Hellman method: DLP is easy if  $p - 1$  has no large prime factors.
- ▶ Baby-step giant-step algorithm:  $\sqrt{p}$  operations and  $\sqrt{p}$  memory.
- ▶ Index calculus, and number field sieve: better (sub-exponential) complexity.
- ▶ All this works for general finite fields, not just  $\mathbb{Z}_p$ .

## How hard is the DLP

- ▶ Naive search:  $p$  operations (a bad algorithm).
- ▶ Silver–Pohlig–Hellman method: DLP is easy if  $p - 1$  has no large prime factors.
- ▶ Baby-step giant-step algorithm:  $\sqrt{p}$  operations and  $\sqrt{p}$  memory.
- ▶ Index calculus, and number field sieve: better (sub-exponential) complexity.
- ▶ All this works for general finite fields, not just  $\mathbb{Z}_p$ .
- ▶ Use the group of points of an elliptic curve: index calculus and NFS no longer apply.

## More information

This talk will appear soon on my home page:

<http://www.cs.rhbnc.ac.uk/~simonb/>

A good textbook in the area:

Douglas R. Stinson, *Cryptography: Theory and Practice* (3rd Edition), Chapman and Hall / CRC Press, 2006.