Lecture 6. Key exchange

Passive adversaries, agreeing on a secret

Bruno Grenet

https://membres-ljk.imag.fr/Bruno.Grenet/IntroCrypto.html

Introduction to cryptology
Université Grenoble Alpes – IM²AG
M1 INFO, MOSIG & AM

Introduction

Up to now: Symmetric cryptography

- Symmetric encryption *confidentiality*
- Message authentication codes *authenticity/integrity*

$\rightarrow$ Both require parties to share a common secret

How do two parties agree on a common secret?

Solutions

- Each pair of parties meets once (in person!) to agree on a key
  - Not always possible to meet
  - Each party retain a key for each other party N parties $\rightarrow \Theta(N^2)$ keys
- Key distribution centers:
  - Each party shares a (secret) key with the KDC, responsible to *distribute* session keys
  - Pros: Each party needs to meet only the KDC once and retain one key
  - Cons: the KDC is the central security point, and parties must meet the KDC

Public-key cryptography

Key-exchange protocols

- Two parties discuss publicly
- At the end, both parties know a same secret k
- External observers do not learn the secret, even after reading all exchanged messages

Public-key encryption and signatures

- Direct protocols to ensure confidentiality, authenticity and/or integrity
- Based on a pair (public key, private key) $\to$ no common secret

In this course

- Lecture 6: Key-exchange protocols
- Lecture 7: Public-key encryption
- Lecture 8: Signatures *public-key equivalent to MACs*

Contents

The goal of a key exchange

Allow two parties to agree on a key, remotely

Objective

- Alice and Bob *publicly* exchange messages
- At the end of the exchange, they both know the same key k
- An attacker who sees all the messages has no information about k

Is this possible?

- The attacker sees as much as Alice and Bob ‽
- No information $\rightarrow$ computational security

New Directions in Cryptography

644 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-22, NO. 6, NOVEMBER 1976

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

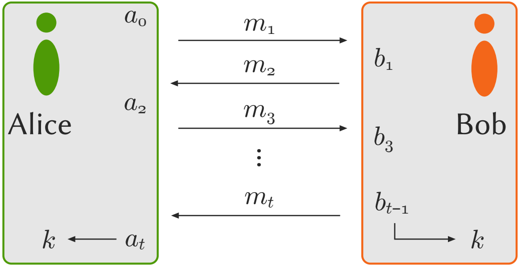
***Abstract*—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.**

I. INTRODUCTION

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of me-

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Definition of a protocol



Key exchange protocol

- Public : messages $m_1, \ldots, m_t$; key space $\mathcal{K}$
- Private : the a_is known only to Alice, the b_is only to Bob
- The protocol is correct if Alice and Bob compute the same key $k \in \mathcal{K}$

Vocabulary

- $m_1, \ldots, m_t$: the *transcript*

Security of a protocol

Secure key exchange protocol: given $m_1, \ldots, m_t$, it is difficult to compute k

Game: key exchange indistinguishability

Challenger simulates the protocol $\rightsquigarrow$ transcript $m_1, \ldots, m_t$ and key $k \in \mathcal{K}$
draws $b \twoheadleftarrow \{0, 1\}$ and returns $\hat{k} = k$ if $b = 1$ and $\hat{k} \twoheadleftarrow \mathcal{K}$ otherwise

Adversary sees the transcript and $\hat{k}$, and returns a bit b'

Advantages

- For a specific adversary $\mathcal{A}$:

$$\mathrm{Adv}_{\mathrm{KE}}^{\mathrm{IND-EAV}}(\mathcal{A}) = \left| \Pr\left[b' = 1 \middle| b = 1 \right] - \Pr\left[b' = 1 \middle| b = 0 \right] \right| = 2 \Pr\left[\textsc{success}\right] - 1$$

- For the protocol:

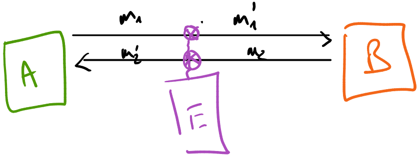
$$\mathrm{Adv}_{\mathrm{KE}}^{\mathrm{IND-EAV}}(t) = \max_{\mathcal{A}_t} \mathrm{Adv}_{\mathrm{KE}}^{\mathrm{IND-EAV}}(\mathcal{A}_t)$$

where $\mathcal{A}_t$ denotes an algorithm with running time $\leq t$

Eavesdropper security and *person-in-the-middle* attack

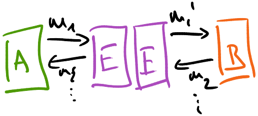
Indistinguishability in the presence of an eavesdropper

- Security definition assumes an authenticated channel between Alice and Bob
- The adversary is only *passive*



Person-in-the-middle attack

- Eve intercepts messages between Alice and Bob
- She impersonates both Alice and Bob
- She creates a common secret with Alice, and another one with Bob
- Alice and Bob incorrectly think they share a common secret



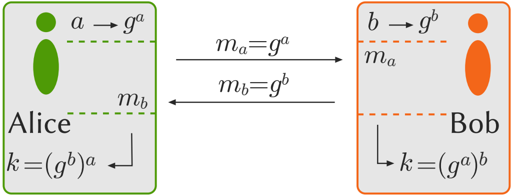
Key exchange is not enough

Combine with authentication

signatures

- *Authenticated* key exchange

A glimpse of Diffie-Hellman protocol



Protocol sketch

Public:	a *number* g
Alice	chooses a *random* a and computes g^a
Bob	chooses a *random* b and computes g^b
Alice	sends $m_a = g^a$ to Bob
Bob	sends $m_b = g^b$ to Alice
Alice	computes $k = m_b^a = g^{ab}$
Bob	computes $k = m_a^b = g^{ab}$

Correction:

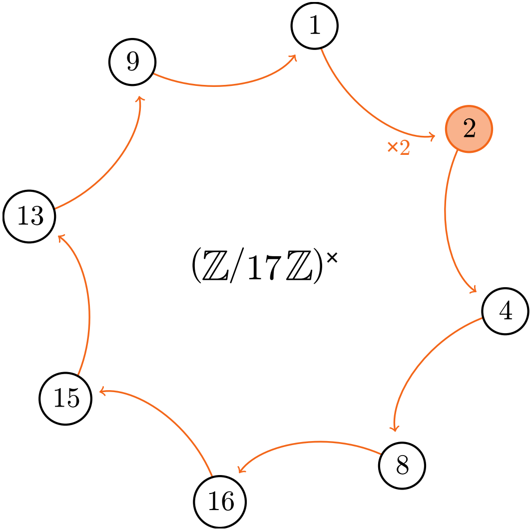
$$m_b^a = (g^b)^a = g^{ab} = (g^a)^b = m_a^b$$

Remaining questions

- What is this *number* g?
- How can this scheme be secure while Eve sees g^a and g^b?

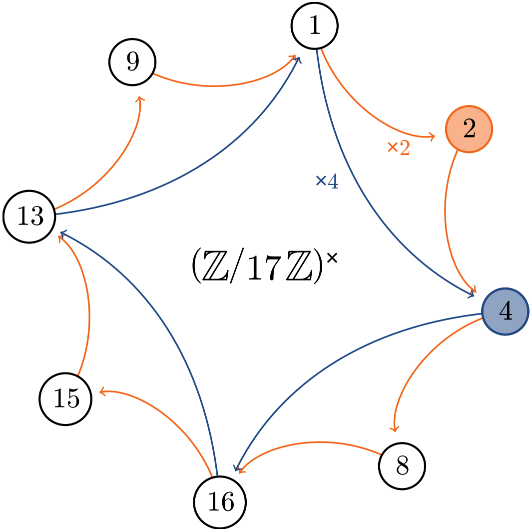
Contents

The multiplicative group of $\mathbb{Z}/p\mathbb{Z}$



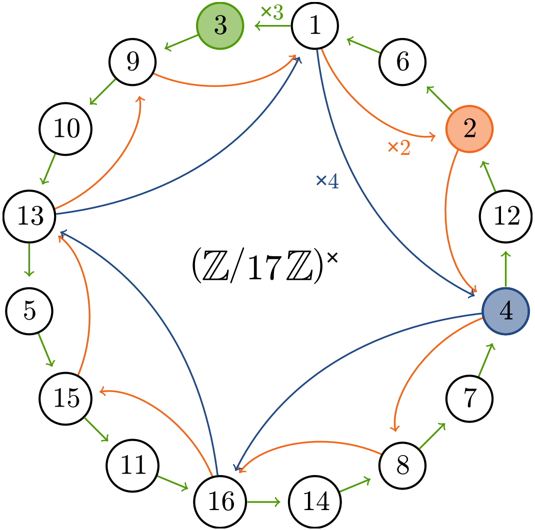
- 2 has order 8

The multiplicative group of $\mathbb{Z}/p\mathbb{Z}$



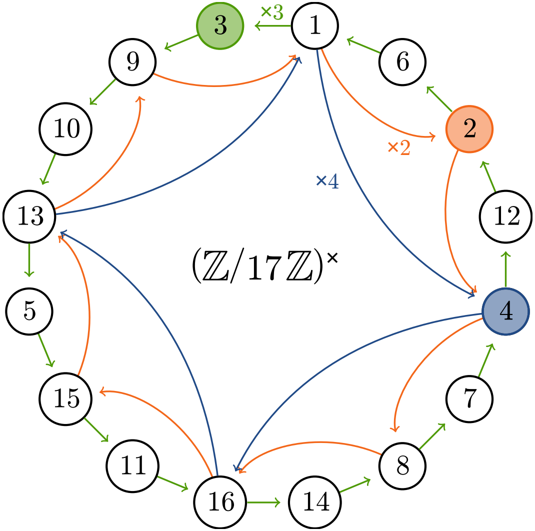
- 2 has order 8
- 4 has order 4

The multiplicative group of $\mathbb{Z}/p\mathbb{Z}$



- 2 has order 8
- 4 has order 4
- 3 has order 16 $\rightarrow$ generator

The multiplicative group of $\mathbb{Z}/p\mathbb{Z}$



- 2 has order 8
- 4 has order 4
- 3 has order 16 $\rightarrow$ generator

Theorem

For every p, $(\mathbb{Z}/p\mathbb{Z})^\times$ is a cyclic group: there exists a generator $g \in (\mathbb{Z}/p\mathbb{Z})^\times$ such that

$$(\mathbb{Z}/p\mathbb{Z})^\times = \{g^n : 0 \leq n < p-1\}$$

Remarks

- The generator is *not* unique (ex.: 3, 5, 6, 7, 10, 11, 12, 14)
- $g^n = g^{n \bmod p}$ since $g^p = 1$
  $(\mathbb{Z}/p\mathbb{Z})^\times = \{g^n : n \in \mathbb{Z}\}$

Cyclic groups

Definitions

A (multiplicative) cyclic group of order n with generator g is a set $\langle g \rangle = \{g^i : 0 \leq i < n\}$ of size n with $g^n = g^0$ and *standard* rules for exponents: $g^{a+b} = g^a \cdot g^b$, $(g^a)^b = g^{ab}$, …

Remarks

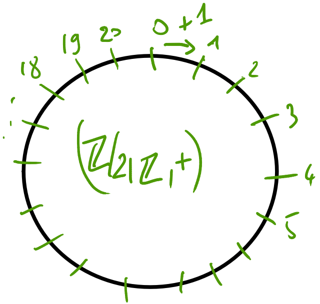
- The generator is not unique
- For all $m \in \mathbb{Z}$, $g^m = g^{m \bmod n}$ $\qquad \langle g \rangle = \{g^t : t \in \mathbb{Z}\}$
- Each element $x \in \langle g \rangle$ *generates a subgroup* $\langle x \rangle = \{x^t : t \in \mathbb{Z}\} \subset \langle g \rangle$
  - order of x = order of $\langle x \rangle$ = number of elements in $\langle x \rangle$

More general definitions

- Additive cyclic group with generator g: $G = \{i \cdot g : 0 \leq i < n\}$
  - Other binary operations
- Non-cyclic groups $\qquad$ Lecture 9
- Infinite (cyclic or non-cyclic) groups

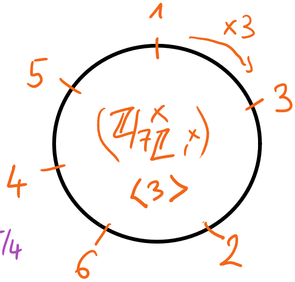
Examples – graphical representations

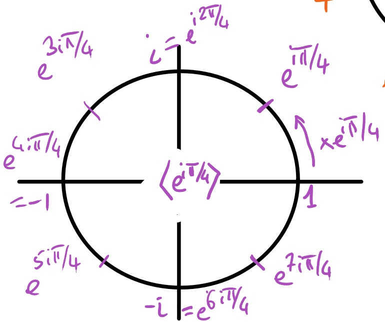
Additive



generator : 1

Multiplicative





Discrete logarithm problem

Definitions

Given a cyclic group $G = \langle g \rangle$,

- the discrete logarithm of x in base g is the unique $0 \leq t < \#G$ such that $x = g^t$
- the discrete logarithm problem is: given x, compute t

The naive algorithm

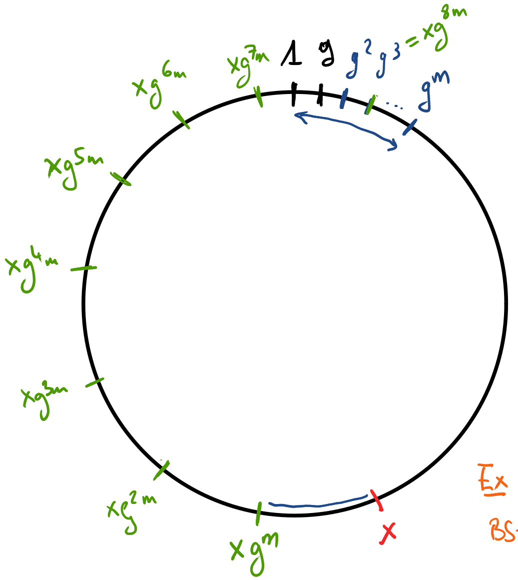
- Compute $g^0, g^1, g^2, \ldots$ until we get $g^t = x$
- Complexity $O(t) = O(\#G)$ operations in G

Easy case: $(\mathbb{Z}/n\mathbb{Z}, +)$

- Generators: 1 or any g such that $\textsc{gcd}(g, n) = 1$
- Discrete logarithm of x: t s.t. $x = t \cdot g \bmod n$
- Case $g = 1$: nothing to do!
- General case:
  1. Compute u, v s.t. $u \cdot g + v \cdot n = 1$ — Extended Euclidean Algorithm
  2. Return $t = u \cdot x \bmod n$ — $t \cdot g = u \cdot x \cdot g = x \bmod n$

Baby step-giant step: A picture is worth a thousand words

(Shanks, 1971)



$g^3 = x g^{8m}$

$\Updownarrow$

$x = g^{(3-8m) \bmod n}$

Complexity: m baby steps

$\leq n/m$ giant steps

↳ take $m \simeq \sqrt{n}$ to get complexity $\Theta(\sqrt{n})$.

Ex $\mathbb{Z}/17\mathbb{Z}^\times$ $g = 3$ $x = 7$ $(m = 4)$

$BS = [1, 3, 9, 10, 13]$ $GS = [7, 6, 10]$

$3^3 = x\, 3^{2m} \Leftrightarrow x = 3^{3-8} = 3^{11}$

Baby step-giant step: the algorithm

(Shanks, 1971)

Input: a cyclic group $G = \langle g \rangle$ of order n and $x \in G$

Output: the discrete logarithm t of x in base g

1. $m \leftarrow \lceil \sqrt{n} \rceil$
2. $B \leftarrow [1, g, g^2, \ldots, g^{m-1}]$ — *Baby steps*
3. $(h, y, j) \leftarrow (g^m, x, 0)$
4. while $y \notin B$: $(y, j) \leftarrow (y \cdot h, j + 1)$ — *Giant steps*: $y = x \cdot g^{m \cdot j}$
5. $i \leftarrow$ index such that $y = g^i$ — *Match found*: $x \cdot g^{m \cdot j} = g^i$
6. return $(i - m \cdot j) \bmod n$

Analysis

Correction: by Euclidean division, there exist $i, j < m$ such that $t = i - mj \bmod n$

Complexity: $O(\sqrt{n})$ baby steps + $O(\sqrt{n})$ giant steps

Test for a match at each (giant) step: $O(1)$ using hash tables

$\Rightarrow O(\sqrt{n})$ (same in space)

DLP hardness

Theorem (baby-step giant-step)

In any cyclic group G, the discrete logarithm problem can be computed in time $O(\sqrt{\#G})$

- Easily parallelizable
- Variants with better space complexity: Pollard's ρ or kangaroos (*a.k.a.* λ) algorithms
- Pohlig-Hellman (1978): $O(\sqrt{p})$ *largest prime divisor of* $\#G$

Choice of G

- $(\mathbb{Z}/n\mathbb{Z}, +)$ or $\#G$ small: easy DLP!
- $(\mathbb{Z}/p\mathbb{Z}^{\times}, \times)$: usually hard, though not *maximally* hard $\ll \sqrt{p}$
  - record: p of 795 bits, 3100 core-year Boudot *et al.*, 2019
- Points of an elliptic curve over a finite field: maximally hard $O(\sqrt{n})$
  - record: group of 114 bits, 13 days on GPU Zieniewicz & Pons, 2020

Additional remarks

- One should use prime order groups
- Algorithm polynomial in $\log n$ on a quantum computer Shor, 1997

Final words on the discrete logarithm

Example of a (conjecturally) *one-way* function

Informal definition

A function $f : E \to F$ is *one-way* if

- it is efficiently computable
- its inverse is hard to compute — given $y \in F$, find x s.t. $f(x) = y$

Exponentiation in a cyclic group

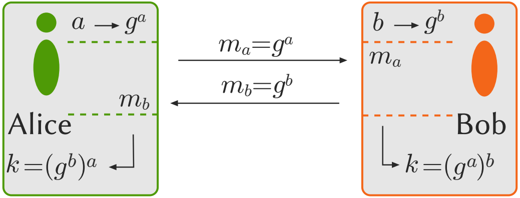
$\textsc{pow} : \mathbb{Z} \to \langle g \rangle$ defined by $\textsc{pow}(n) = g^n$

- Efficiently computable using *binary powering*
  - $\textsc{pow}(n) = \begin{cases} \textsc{pow}(\lfloor n/2 \rfloor)^2 & \text{if } n \text{ is even} \\ g \cdot \textsc{pow}(\lfloor n/2 \rfloor)^2 & \text{otherwise} \end{cases}$ — $O(\log n)$ operations in $\langle g \rangle$

  $g^n = \begin{cases} (g^{n/2})^2 & \text{for even } n \\ g \cdot (g^{\lfloor n/2 \rfloor})^2 & \text{for odd } n \end{cases}$
- Inverse of $\textsc{pow}$ is the DLP

Contents

The protocol



Diffie-Hellman protocol

Input : Group $G = \langle g \rangle$ of order n

Alice draws $a \twoheadleftarrow \{0, \ldots, n-1\}$, computes $m_a = g^a$ and sends m_a to Bob

Bob draws $b \twoheadleftarrow \{0, \ldots, n-1\}$, computes $m_b = g^b$ and sends m_b to Alice

Alice computes $k_a = m_b^a$

Bob computes $k_b = h_a^b$.

Correctness

The protocol is correct: $k_a = (g^b)^a = g^{ab} = (g^a)^b = k_b$

Use in practice

Where do the secret lives?

- Shared secret $k \in G$, while usually one needs it in $\{0,1\}^*$
- *Key derivation function* $\mathrm{KDF} : G \to \{0,1\}^*$ $\simeq$ hash function

Person-in-the-middle attack *a.k.a Man-in-the-middle*

- Requires an authentication between Alice and Bob
- Out-of-scope of this lecture $\to$ *c.f.* signatures Lecture 8

Cost of the protocol

- Requires two exponentiations in G
  - $O(\log \# G)$ operations in G *binary powering*
  - For instance: $O(\log^2 p \log\log p)$ bit-operations for $\mathbb{Z}/p\mathbb{Z}$

Three difficulty assumptions

G: fixed cyclic group with known generator g

Discrete-Log-Problem Game

Challenger samples $t \twoheadleftarrow \{0, \ldots, \#G - 1\}$ and computes $h = g^t$

Adversary is given h and tries to compute t

Computational Diffie-Hellman Game

Challenger simulates the DH protocol $\rightarrow$ transcript $m_a = g^a$, $m_b = g^b$ and key $k = g^{ab}$

Adversary is given the transcript and tries to compute k

Decisional Diffie-Hellman Game

Challenger simulates the DH protocol $\rightarrow$ transcript $m_a = g^a$, $m_b = g^b$ and key $k = g^{ab}$
draws $\lambda \twoheadleftarrow \{0, 1\}$ and sets $\hat{k} \leftarrow k$ if $\lambda = 1$, $\hat{k} \twoheadleftarrow G$ if $\lambda = 0$

Adversary is given the transcript (m_a, m_b) and $\hat{k}$, and tries to guess λ

DLP/CDH/DDH assumptions

- It is *hard* for an adversary with *reasonable* running-time to win the DLP game
- It is *hard* for an adversary with *reasonable* running-time to win the CDH game
- It is *hard* for an adversary with *reasonable* running-time to win the DDH game

Relating the three assumptions

Lemma

For any cyclic group $G = \langle g \rangle$, DDH is hard $\Rightarrow$ CDH is hard $\Rightarrow$ DLP is hard

Remark

Each game can be won by an adversary running in time $O(\sqrt{\#G})$

Proof of lemma Contrapositive : DLP easy $\overset{(1)}{\Rightarrow}$ CDH easy $\overset{(2)}{\Rightarrow}$ DDH easy

(1) . Assume that DLP is easy : I have $\mathcal{A}_{DLP}$ that solves DLP : given $h = g^t$, computes t

. Build $\mathcal{A}_{CDH}$: given g^a, g^b, computes g^{ab}.

- Compute a using $\mathcal{A}_{DLP}$
- Compute $(g^b)^a$ using binary powering.

(2) . We are given $\mathcal{A}_{CDH}$ that given g^a and g^b, computes g^{ab}.

. Build $\mathcal{A}_{DDH}$: given g^a, g^b and $\hat{k}$, compute $k = g^{ab}$ using $\mathcal{A}_{CDH}$ and test $k \overset{?}{=} \hat{k}$.

Security of the Diffie-Hellman protocol

Theorem

The Diffie-Hellman protocol in the group G is IND-EAV secure under the DDH assumption

Proof

The DDH game is exactly the IND-EAV game specialized with the Diffie-Hellman protocol.

Conclusion

50 shades of Diffie-Hellman

- The DH protocol is essentially the only key exchange protocol
- But many choices of cyclic group G: $(\mathbb{Z}/p\mathbb{Z})^\times$, elliptic curve, isogenies, …
- Key derivation function to go from G to $\{0,1\}^*$

Security of the protocol

- Three hardness assumptions: DLP, CDH, DDH
  - DDH $\iff$ IND-EAV security
  - DDH $\Rightarrow$ CDH $\Rightarrow$ DLP
- Generic attack in $O(\sqrt{\#G})$ operations *baby steps – giant steps*

Inherent vulnerability: *person-in-the-middle*

- Charlie stands between Alice and Bob, and intercepts, modifies, etc. all messages between Alice and Bob
- Requires *authentication* between Alice and Bob
  - Use for instance signatures
  - *Authenticated Key Exchange*