The Diffie-Hellman Key Exchange

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
Email: gianluca.dini@.unipi.it

Version: 2021-04-12

Preliminaries

UNIVERSITÀ DI PISA

- Whitfield Diffie and Martin Hellman, New directions in cryptography, IEEE Transactions of Information Theory, 22(6), pp. 644-654, Nov. 1976
- One-way function
  - f(x): discrete exponentiation is computationally "easy"
  - $f^{-1}(x)$: discrete logarithm it is computationally "difficult"
- CRYPTOSYSTEM FOR KEY EXCHANGE

apr. '21 Diffie-Hellman Key Exchange 2

Preliminaries

UNIVERSITÀ DI PISA

- Mathematical foundation
  - Abstract algebra: groups, sub-groups, finite groups and cyclic groups
- We operate in $\mathbb{Z}_p^*$ with addition and multiplication modulo p, with p prime
  - $\mathbb{Z}_p^*$ is the set of integers i = 0, 1, …, p – 1, s.t. gcd(i, p) = 1
    - Ex. Z^*_{11} = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

DATO CHE p È PRIMO, Z_p^* CONTIENE TUTTI I NUMERI PIÙ PICCOLI DI P.

Facts on modular arithmetic

UNIVERSITÀ DI PISA

- Multiplication is commutative
  - $(a \times b) \equiv (b \times a) \bmod n$
- Exponentiation is commutative
  - $(a^x)^y \equiv (a^y)^x \bmod n$
- Power of power is commutative
  - $(a^b)^c \equiv a^{bc} \equiv a^{cb} \equiv (a^c)^b \bmod n$

apr. '21 Diffie-Hellman Key Exchange 4

$X \equiv y \pmod n$ means that $x \bmod n \equiv y \bmod n$

$(a \bmod n) \times (b \bmod n) \equiv a \times b \pmod n$ that is $((a \bmod n) \times (b \bmod n) \bmod n) \equiv ab \bmod n$

Facts on modular arithmetic

UNIVERSITÀ DI PISA

- Parameters → GRANDE 1024 bit
  - Let p be prime and $g \in \mathbb{Z}_p^*$ be a primitive element (or generator), i.e., for each y = 1, 2, …, p – 1, there is x s.t. y = $\equiv g^x$ mod p
- Discrete Exponentiation
  - Given $x \in \mathbb{Z}_p^*$, compute $y \in \mathbb{Z}_p^*$ s.t. y = g^x mod p
- Discrete Logarithm Problem (DLP)
  - Given $y \in \mathbb{Z}_p^*$, determine $x \in \mathbb{Z}_p^*$ s.t. y = g^x mod p
    - Notation x = $\log_g$ y mod p DIFFICILE

Z_p^*: 1 2 3 ---- p-1

g^i mod p = TUTTI QUESTI VALORI

g^0 mod p = 1

g^1 mod p = 8 (TIPO)

g^2 mod p = 3 ---

apr. '21 Diffie-Hellman Key Exchange 5

A few facts about discrete logarithm in prime fields.
Generator or primitive root it is a number g such that for each number y belonging to Zp* there exists x such that g^x = y mod p
It can be proven that if p is prime then there exists a primitive root g and there is a way to compute it efficiently.
For a general n it is not guaranteed that discrete log exists. But if we choose a prime number p and a generator g then the discrete log exists.
Taken from a different standpoint, g^x defines a permutation of Z_p. So, computing the discrete log of y consists in determine the position (x) of y in the permutation g^x.
There is no proof that DL is hard. The proof is that a lot of smart people has tried to solve it and failed. They only found (sub-)exponential algorithms.

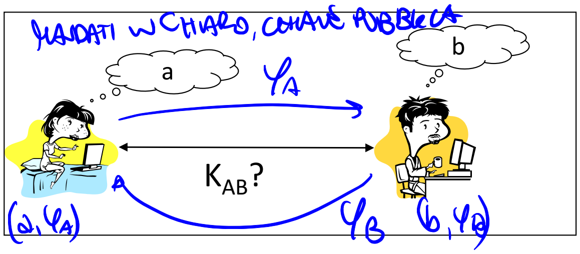
Properties of discrete log

UNIVERSITÀ DI PISA

- $\log_g(\beta\gamma) \equiv (\log_g\beta + \log_g \gamma) \bmod p$
- $\log_g(\beta)^s \equiv s\,(\log_g\beta) \bmod p$

The Diffie-Hellman Protocol

a, b devono essere grandi altrimenti → bruteforce



SETUP

- Let p be a large prime (600 digits, 2000 bits)
- Let 1< g < p a generator
- Let p e g publicly known

- THE DIFFIE-HELLMAN KEY EXCHANGE (DHKE)
  - Alice chooses a random secret number a (private key)
  - Bob chooses a random secret number b (~~public~~ PRIVATE key)

  NESSUN SEGRETO FINORA

  - M1: A → B: A, $Y_A \equiv g^a \bmod p$ (public key)
  - M2: B → A: B, $Y_B \equiv g^b \bmod p$ (public key)
  - Alice computes $K_{AB} \equiv (Y_B)^a \equiv g^{ab} \bmod p = (g^b)^a$
  - Bob computes $K_{AB} \equiv (Y_A)^b \equiv g^{ab} \bmod p = (g^a)^b$

  SHARED KEY

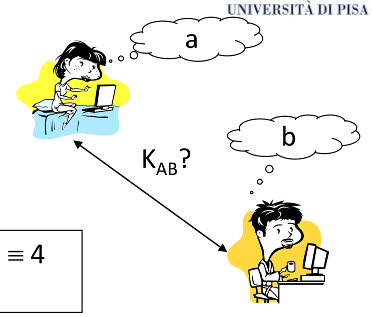
2 esponenziali e 2 messaggi ↓ molto semplice

apr. '21 Diffie-Hellman Key Exchange 7

DHKE with small numbers

POSSO FARE UN BRUTEFORCE

Let $p = 11$, $g = 7$ → IN PRATICA NON SI POSSONO USARE, PIU GRANDI



Alice chooses $a = 3$ and computes $Y_A \equiv g^a \equiv 7^3 \equiv 343 \equiv 2 \bmod 11$

Bob chooses $b = 6$ and computes $Y_B \equiv g^b \equiv 7^6 \equiv 117649 \equiv 4 \bmod 11$

A → B: 2
B → A: 4

Alice receives 4 and computes $K_{AB} = (Y_B)^a \equiv 4^3 \equiv 9 \bmod 11$

Bob receives 2 and computes $K_{AB} = (Y_A)^b \equiv 2^6 \equiv 9 \bmod 11$

apr. '21 Diffie-Hellman Key Exchange 8

DHKE computational aspects

UNIVERSITÀ DI PISA

- Large prime p can be computed as for RSA
- Exponentiation can be computed by square-and-multiply
  - The trick of using small exponents is non applicable here
- $\mathbb{Z}_p^*$ is cyclic
  - g is a generator, g^i mod p defines a permutation
    - p = 11, g = 2

– $2^1 \equiv 2 \bmod 11$	$2^5 \equiv 10 \bmod 11$	$2^9 \equiv 6 \bmod 11$
– $2^2 \equiv 4 \bmod 11$	$2^6 \equiv 9 \bmod 11$	$2^{10} \equiv 1 \bmod 11$
– $2^3 \equiv 8 \bmod 11$	$2^7 \equiv 7 \bmod 11$	*repeat cyclically*
– $2^4 \equiv 5 \bmod 11$	$2^8 \equiv 3 \bmod 11$	

Security of DHKE

UNIVERSITÀ DI PISA

- Intuition
  - Eavesdropper sees p, g, Y_A and Y_B and wants to compute K_{AB}
- Diffie-Hellman Problem (DHP)
  - Given p, g, $\boxed{Y_A \equiv g^a \bmod p}$ and $Y_B \equiv g^b \bmod p$, compute $g^{ab} \bmod p$

    POSSO FARLO FACILMENTE!
- How hard is this problem?

  DISCRETE LOGARITHM PROBLEM (DLP)

  POSSO OTTENERE A DA Y_A -

apr. '21 Diffie-Hellman Key Exchange 10

Security of DHKE

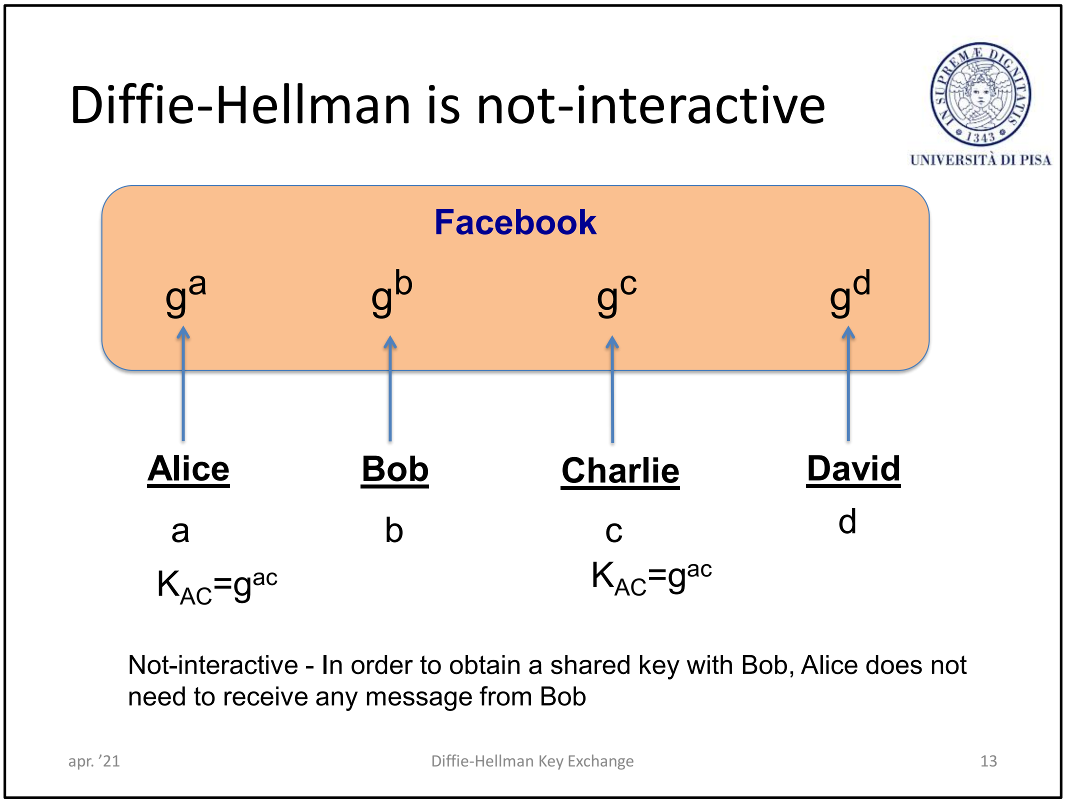
NESSUNO HA PROVATO CHE SE DLP E' DIFFICILE ALLORA DHP E' DIFFICILE

- DHP $\leq_P$ DLP
  - If DLP can be easily solved, then DHP can be easily solved
  - There is no proof of the converse, i.e., if DLP is difficult then DHP is difficult
  - At the moment, we don't see any way to compute K_{AB} from Y_A and Y_B without first obtaining either a or b

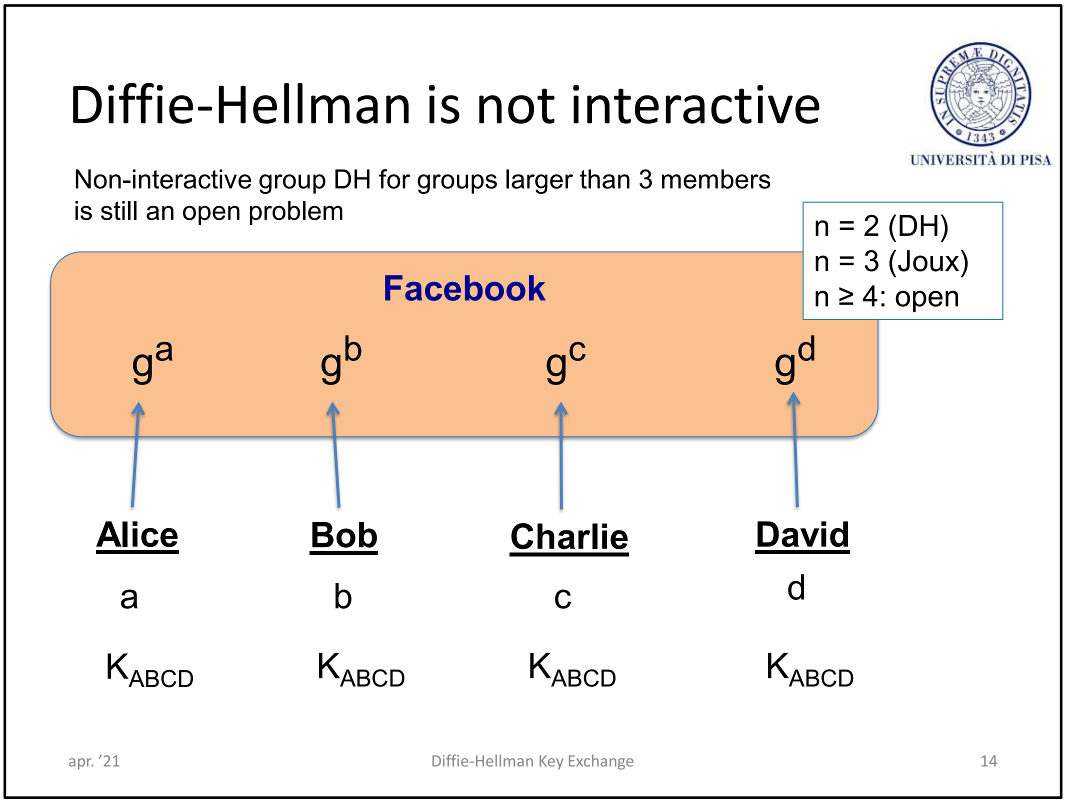
- If DLP is easy then it is easy to compute a from Y_A. Then, $K_{AB} = (Y_B)^a \pmod p$ which is easy because exp (mod p) is easy.
- If it were possible to prove the converse, than breaking DH would make it possible to solve DLP.

Diffie-Hellman Key Exchange

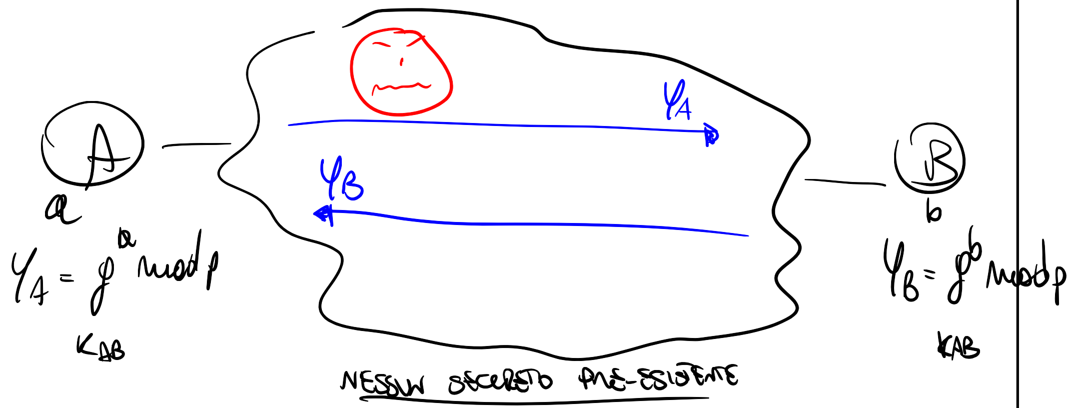
NOT-INTERACTIVITY



DH is not-interactive. This means that If Alice wants to communicate with Bob then Alice goes to Bob's profile, reads g^b mod p and generates K_{AB}. In other words, Alice does not need to receive any message from Bob



The problem here is to compute a group key shared by K_{ABCD} from g^a, g^b, g^c, and g^d in a non interactive way. Non-interactive Group DH for $n > 3$ is an open problem. The Joux algorithm is very complex and contains "fancy" mathematics.

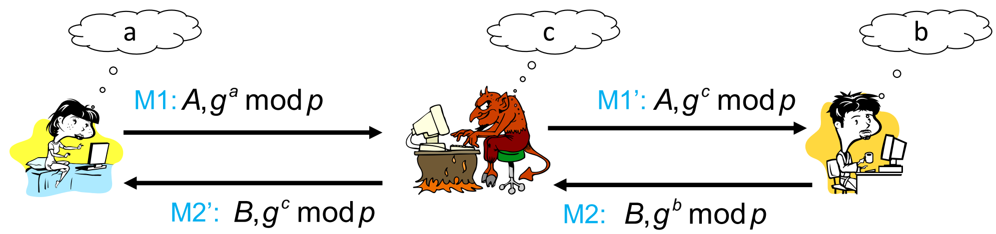


Diffie-Hellman Key Exchange

THE MAN-IN-THE-MIDDLE ATTACK

Man-in-the-Middle Attack

UNIVERSITÀ DI PISA



$K_{AM} = g^{ac} \bmod p$

$K_{AM} = g^{ac} \bmod p$, e
$K_{BM} = g^{bc} \bmod p$

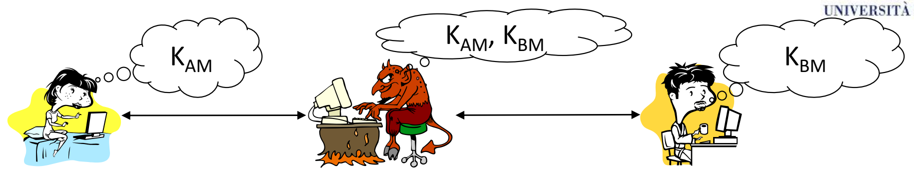
$K_{BM} = g^{bc} \bmod p$

NON HAI UN PROBLEMA DI CONFIDENTIALITY MA UN PROBLEMA DI KEY AUTHENTICATION → CERTIFICATI

apr. '21 Diffie-Hellman Key Exchange 16

In the man-in-the-middle attack, the adversary replaces both g^a and g^b with g^c. Alice believes to share key K_{AM} with Bob whereas she shares it with the adversary. So does Bob with K_{BM}. The problem is that messages M1' and M2' carry no proof that g^c is actually Alice's (Bob's) public key. In other words, there is nothing in message M1 that indissolubly links the identifier «Alice» to public key g^a. We already know the answer to this problem: certificates!

Man-in-the-Middle Attack



- Beliefs
  - Alice believes to communicate with Bob by means of K_{AM}
  - Bob believes to communicate with Alice by means of K_{BM}
- The adversary can
  - read messages between Alice and Bob
  - impersonate Alice or Bob
- DHKE is insecure against MIM (active) attack

Diffie-Hellman Key Exchange

THE GENERALIZED DLP AND ATTACKS AGAINST DLP

See Section 8.3 The Discrete Logarithm Problem of Paar's book.

The Generalized DLP

UNIVERSITÀ DI PISA

- DLP can be defined on any cyclic group
- GDLP (def)
  - Given a finite cyclic group G with group operation • and cardinality n, i.e., |G| = n. We consider a primitive element $\alpha \in G$ and another element $\beta \in G$. The discrete logarithm problem is finding the integer x, where $1 \leq x \leq n$, such that

$$\beta = \underbrace{\alpha \bullet \alpha \bullet \alpha \bullet \ldots \bullet \alpha}_{\text{x times}} = \alpha^x$$

apr. '21 Diffie-Hellman Key Exchange 19

DLP is not restricted only to the multiplicative group Z^*_p but can be extended in any cyclic group. It is important to notice that there are cyclic groups in which DLP is not "difficult". For example, consider $(Z_{11}, +)$ and $\alpha = 2$ and $\beta = 3$. We have to compute 2 + 2 + … 2 (x times) = $2 \cdot x \equiv 3 \bmod 11$. It follows that $x \equiv 2^{-1} \cdot 3 \equiv 6 \cdot 3 \equiv 7 \bmod 11$ (notice that gcd(2, 11 = 1). The reason why DLP is "easy" here is because we have operations, namely multiplication and inversion, that are not in the additive group.

DLP for cryptography

UNIVERSITÀ DI PISA

- Multiplicative prime group $\mathbb{Z}_p^*$
  - DHKE, ElGamal encryption, Digital Signature Algorithm (DSA)
- Cyclic group formed by Elliptic curves → RICHIEDONO CHIAVI PIÙ PICCOLE A STESSO LIVELLO DI SICUREZZA
- Galois field GF(2^m) → NON SICURI!
  - Equivalent to $\mathbb{Z}_p^*$
  - Attacks against GF(2^m) are more powerful than DLP in $\mathbb{Z}_p^*$ so we need "higher" bit lengths than $\mathbb{Z}_p^*$
- Hyperelliptic curves or algebraic varieties

Algorithms for DLP

- Generic Algorithms work in any cyclic group:
  - Brute-force Search
  - Shank's Baby-Step Giant-Step Method
  - Pollard's Rho Method
  - Pohlig-Hellman Algorithm
- Nongeneric algorithms exploit inherent structure of certain groups
- Fact – Difficulty of DLP is independent of the generator

The security of many asymmetric primitives is based on the difficulty of the DLP in cyclic groups. We still don't know the exact difficulty of computing the DLP in any given actual group. What we mean by this is that even though some attacks are known, one does not know whether there are any better, more powerful algorithms for solving the DLP. This situation is similar to the hardness of integer factorization, which is the one-way function underlying RSA. Nobody really knows what the best possible factorization method is. For the DLP some interesting general results exist regarding its computational hardness. We give an overview of algorithms for computing discrete logarithms which can be classified into generic algorithms and nongeneric algorithms and which will be discussed in a little more detail.

Algorithms for DLP

UNIVERSITÀ DI PISA

CARDINALITÀ DELL'INSIEME

DA TROVARE!

$y = g^{x} \bmod p$

- Generic algorithms
  - Brute-force Search
    - Running time: $O(|G|)$ multiplications
  - Shank's Baby-Step Giant-Step Method
    - Running time: $O\left(\sqrt{|G|}\right)$ multiplications
    - Storage: $\underbrace{O\left(\sqrt{|G|}\right)}$

SUB-EXPONENTIAL

p 1024

$|G| \simeq 2^{1024}$

%

$\sqrt{2^{1024}}$

Algorithms for DLP

- Generic Algorithms MIGLIORAMENTO DEL PRECEDENTE!
  - Pollard's Rho Method
    - Based on the Birthday Paradox
    - Running time: $O\left(\sqrt{|G|}\right)$ multiplications SUB-EXPONENTIAL
    - Storage: negligible

Algorithms for DLP

UNIVERSITÀ DI PISA

- Generic Algorithms
  - Pohlig-Hellman Algorithm
    - Based on CRT, exploits factorization of $\boxed{|G| = \prod_{i=1}^{r} (p_i)^{e_i}}$
      - Reduces DLP to DLP in (smaller) groups of order ${p_i}^{e_i}$
      - In the EC, computing |G| is not easy
    - Running time: $\mathcal{O}\left(\sum_{i=1}^{r} e_i \cdot \left(lg|G| + \sqrt{p_i}\right)\right)$ multiplications
      - Efficient if each p_i is «small»
      - To prevent the attack the *smallest factor* of |G| must be in the range 2^{160}

INVECE DI ESEGUIRLO IN $|\mathbb{Z}_p^*|$ CHE PUO' ESSERE MOLTO GRANDE SPEZZO.

FATTORI PRIMI PICCOLI LO RENDONO MOLTO EFFICIENTE.

$|\mathbb{Z}_p^*| = p-1$, $|\mathbb{Z}_{11}^*| = 10 \rightarrow 2 \times 5$ FATTORIZZAZIONE DELLA CARDINALITÀ

$\mathbb{Z}_2^*$ e $\mathbb{Z}_5^*$

Algorithms for DLP

UNIVERSITÀ DI PISA

- Nongeneric algorithms
  - Exploit inherent structure of certain groups
  - The Index-Calculus Method
    - Very efficient algorithm to compute DLP in $\mathbb{Z}_p^*$ and GF(2^m)
    - Sub-exponential running time 2^{80}
      - In $\mathbb{Z}_p^*$, in order to achieve 80-bit security, the prime p must be at list 1024 bit long
      - It is even more efficient in GF(2^m) ➔ For this reason, DLP in GF(2^m) are not used in practice

DLP – rule of thumb

- Let p be a prime on k bits ($p < 2^k$)
- Exponentiation takes at most $2 \cdot \log_2 p < 2k$ long integer multiplications (mod p)
  - Linear in the exponent size (k)
- Discrete logs require $p^{½} = 2^{k/2}$ multiplication
- Example n = 512
  - Exponentiation: #multiplications ≤ 1024
  - Discrete log: #multiplications $\approx 2^{256} = 10^{77}$

Diffie-Hellman Key Exchange

DLP IN SUBGROUPS

See Section 8.2 Some Algebra of Paar's book.

Cyclic groups

MODULAR OPERATION

mod(p)

- Theorem 8.2.2. For every prime p, ($\mathbb{Z}_p^*$, $\times$) is an abelian finite cyclic group
  - **Finite**: contains a finite number of elements
  - **Group**: closed, associative, identity element, inverse, commutative
  - **Cyclic**: contain an element α with maximum order ord(α) = $|\mathbb{Z}_p^*|$ = $p - 1$, where *order of* $a \in \mathbb{Z}_p^*$, ord(a) = k, is the smallest positive integer k such that $a^k \equiv 1 \bmod p$
  - α is called *generator* or *primitive element*
- The notion of finite cyclic group is generalizable to (G, •)

apr. '21 Diffie-Hellman Key Exchange 28

$\mathbb{Z}_p^*$

Cyclic groups

UNIVERSITÀ DI PISA

- Example: consider $\mathbb{Z}_{11}^{*}$ and a = 3
  - $a^1 = 3$
  - $a^2 = a \cdot a = 3 \cdot 3 = 9$
  - $a^3 = a^2 \cdot a = 9 \cdot 3 = 27 \equiv 5 \bmod 11$
  - $a^4 = a^3 \cdot a = 5 \cdot 3 = 15 \equiv 4 \bmod 11$
  - $a^5 = a^4 \cdot a = 4 \cdot 3 = 12 \equiv 1 \bmod 11$ ← ord(3) = 5
    ORDINE DELL'ELEMENTO
  - $a^6 = a^5 \cdot a \equiv 1 \cdot a \equiv 3 \bmod 11$
  - $a^7 = a^5 \cdot a^2 \equiv 1 \cdot a^2 \equiv 9 \bmod 11$
  - $a^8 = a^5 \cdot a^3 \equiv 1 \cdot a^3 \equiv 5 \bmod 11$
  - $a^9 = a^5 \cdot a^4 \equiv 1 \cdot a^4 \equiv 4 \bmod 11$
  - $a^{10} = a^5 \cdot a^5 \equiv 1 \cdot 1 \equiv 1 \bmod 11$ ← period
  - $a^{11} = a^{10} \cdot a \equiv 1 \cdot a \equiv 3 \bmod 11$
  - 3^i generates the periodic sequence {3, 9, 5, 4, 1}

apr. '21 Diffie-Hellman Key Exchange 29

For example, consider Z_{11}^{*} and a = 3. The 3^x mod 11 generates a periodic sequence {3, 9, 5, 4, 1}. The sequence contains a subset of Z11*. The period is 5.

Cyclic groups - PRIMITIVE ELEMENT %

- Example: consider $\mathbb{Z}_{11}^{*}$ and a = 2
  - $a = 2$ $a^6 \equiv 9 \bmod 11$
  - $a^2 = 4$ $a^7 \equiv 7 \bmod 11$
  - $a^3 = 8$ $a^8 \equiv 3 \bmod 11$
  - $a^4 \equiv 5 \bmod 11$ $a^9 \equiv 6 \bmod 11$
  - $a^5 \equiv 10 \bmod 11$ $a^{10} \equiv 1 \bmod 11 \leftarrow$ ord(2)
  - ord(2) = 10 = | $\mathbb{Z}_{11}^{*}$ | ➔ 2 is a primitive element

$\underbrace{1-2-3-4-5-6-7-8-9-10}$

$\mathbb{Z}_{11}^{*}$

GENERATOR

apr. '21 Diffie-Hellman Key Exchange 30

Consider now Z_{11}^{*} and a = 2. The 2^x mod 11 generates a periodic sequence {2, 4, 8, 5, 10, 9, 7, 3, 6, 1}. Now the sequence contains all elements of Z11*. Its length is 10.

Cyclic groups

UNIVERSITÀ DI PISA

Powers of a generator define a permutation of the elements of $\mathbb{Z}_p^*$

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

a ← CASUALE

g^a → SCELGO IN MODO CASUALE UNO DEI VALORI SOPRA

apr. '21 Diffie-Hellman Key Exchange 31

2x mod 11 defines a permutation.

ORD(5) = 5 , $5^5 = 1 \mod 11 \rightarrow 5 \cdot 5^2 \cdot 5^2 = 5 \cdot 3 \cdot 3 = 45 = 1 \mod 11$

$11 \cdot 4 + 1$

$\cdot 5 \mod 11 = 5$

$\cdot 25 \mod 11 = 3$

$\cdot 25 \mod 11 = 3$

Cyclic groups

- Order of elements of $\mathbb{Z}_{11}^*$
  - ord(1) = 1 ord(6) = 10
  - ord(2) = 10 ord(7) = 10
  - ord(3) = 5 ord(8) = 10
  - ord(4) = 5 ord(9) = 5
  - ord(5) = 5 ord(10) = 2
- Any order is a divisor of $|Z_{11}^*|$ = 10
- #(primitive elements) is $\Phi(10) = \Phi(|\mathbb{Z}_{11}^*|) = 4$
- Set of primitive elements = {2, 6, 7, 8}

$\Phi(n)$ = NUMERI PRIMI RISPETTO A n

$\Phi(10) = 1, 3, 7, 9$

apr. '21 Diffie-Hellman Key Exchange 32

Cyclic groups

UNIVERSITÀ DI PISA

- Theorem 8.2.3
  - Let G be a finite group. Then for every $a \in G$ it holds that:
  - 1. $a^{|G|} = 1$ (Generalization of Fermat's Little Theorem)
  - 2. ord(a) divides $|G|$
- Theorem 8.2.4
  - Let G be a finite cyclic group. Then it holds that
    1. The number of primitive elements of G is $\Phi(|G|)$.
    2. If $|G|$ is prime, then all elements $a \neq 1 \in G$ are primitive.

apr. '21 Diffie-Hellman Key Exchange 33

TEOREMA 8.2.5 CYCLIC SUBGROUP THEOREM;

Se G è un gruppo ciclico -

ALLORA OGNI ELEMENTO A ∈ G PUÒ CREARE UN SOTTO GRUPPO CHE MANTIENE LE CARATTERISTICHE DI UN GRUPPO CICLICO

Subgroups

- Theorem 8.2.6 (Lagrange's theorem)
  - Let H be a subgroup of G. Then $|H|$ divides $|G|$.
- Consider $\mathbb{Z}_{11}^*$, a = 3, ord(3) = 5 $|\mathbb{Z}_{11}^*| = 10 \rightarrow$ DIVISORI $= \{1, 2, 5\}$
  - H = {1, 3, 4, 5, 9}
  - H is a finite, cyclic subgroup of order 5 which divides 10
- Example

Subgroup	elements	primitive elements
H_1	{1} #1	α = 1
H_2	{1, 10} #2	α = 10
H_3	{1, 3, 4, 5, 9} #5	α= 3, 4, 5, 9

Subgroups

UNIVERSITÀ DI PISA

- Theorem 8.2.7
  - Let G be a finite cyclic group of order n and let α be a generator of G. Then for every integer k that divides n there exists exactly one cyclic subgroup H of G of order k. This subgroup is generated by $\alpha^{n/k}$. H consists exactly of the elements $a \in G$ which satisfy the condition $a^k = 1$. There are no other subgroups.

DATO Z_{11}^* e $\alpha = 8$ GENERATOR $\beta = 8^{10/2} = 10 \mod 11$

CHE E' UN GENERATORE PER H DI ORDINE $k=2$

SE USO Z_p^* → $p-1$ E' PARI ⇒ HA UN FATTORE PRIMO PICCOLO, 2!

Relevance to cryptography

- ON SOLVING DLP
- Pohlig-Hellman Algorithm
  - Exploit factorization of $|G| = p_1^{e1} \cdot p_2^{e2} \cdot \ldots \cdot p_\ell^{e\ell}$
  - Run time depends on the size of prime factors
    - The smallest prime factor must be in the range 2^{160}
- $|\mathbb{Z}_p^*| = p - 1$ is even ➔ 2 (small) is one of the divisors! → DOBBIAMO LAVORARE SU UN SOTTO-INSIEME, NON UNO QUALSIASI PERÒ
- It is advisable to work in a prime subgroup H
  - If |H| is prime, ∀a∈H, a is a generator (Theorem 8.2.4)

CARDINALITÀ = NUMERO PRIMO

apr. '21 Diffie-Hellman Key Exchange 36

$H \subseteq \mathbb{Z}_p^* \rightarrow |H| =$ NUMERO PRIMO E GRANDE

↓

COSÌ L'ALGORITMO PER DLP NON È EFFICIENTE

Relevance to cryptography [1/2]

UNIVERSITÀ DI PISA

- SMALL SUBGROUP CONFINEMENT ATTACK
  - Consider prime p, $\mathbb{Z}_p^*$, and generator α

UNA VARIAZIONE DEL MAN-IN-THE-MIDDLE

Alice **MIM** **Bob**

a ← random() b ← random()

$A \equiv \alpha^a \bmod p$ $B \equiv \alpha^b \bmod p$

------**A**------------**A'**-------- >

$K = (A')^b \bmod p$

<-------**B'**---------------------**B**-----

$K = (B')a \bmod p$

< ------------ E_K(*session*)--------------------- >

apr. '21 Diffie-Hellman Key Exchange 37

In cryptography, a subgroup confinement attack, or small subgroup confinement attack, on a cryptographic method that operates in a large finite group is where an attacker attempts to compromise the method by forcing a key to be confined to an unexpectedly small subgroup of the desired group. The attack exploits THEOREM 8.2.7. The adversary selects k that divides $|\mathbb{Z}_p^*| = p - 1$ then, (s)he computes

1) $A' \equiv A^{n/k} \equiv (\alpha^a)^{n/k} \equiv (\alpha^{n/k})^a \bmod p$
2) $B' \equiv B^{n/k} \equiv (\alpha^b)^{n/k} \equiv (\alpha^{n/k})^b \bmod p$

It follows that $\alpha^{n/k}$ is a generator of subgroup H of order k. It follows that DHKE gets confined in H_k and therefore a brute force attack becomes easier.

Relevance to cryptography [2/2]

UNIVERSITÀ DI PISA

- SMALL SUBGROUP CONFINEMENT ATTACK
- Given THEOREM 8.2.7
  - Consider k that divides $|\mathbb{Z}_p^*| = p - 1$ then
  - $A' \equiv A^{n/k} \equiv (\alpha^a)^{n/k} \equiv (\alpha^{n/k})^a \bmod p$ → GENERATORE β DI H DI ORDINE k
  - $B' \equiv B^{n/k} \equiv (\alpha^b)^{n/k} \equiv (\alpha^{n/k})^b \bmod p$
  - Session key $K = \beta^{ab} \bmod p$, with $\beta = \alpha^{n/k}$
  - $\beta = \alpha^{n/k}$ is a generator of subgroup H of order k ➔
  - DHKE gets confined in H_k and brute force becomes easier

→ NUMERO DI CHIAVI POSSIBILI DIMINUITO!

PER EVITARE QUESTO L'ORDINE n DEVE ESSERE GRANDE E PRIMO.

apr. '21 Diffie-Hellman Key Exchange 38