Public Key Cryptology
Diffie-Hellman Key Exchange

Motivation

- Until early 70s, cryptography was mostly owned by government and military
  - Key distribution is more manageable and better funded
- Symmetric cryptography not ideal for commercialization
  - Enormous key distribution problem; most parties may never meet physically
  - Must ensure authentication, to avoid impersonation, fabrication
- Few researchers (Diffie, Hellman, Merkle), in addition to the IBM group, started exploring Cryptography because they realized it is critical to the forthcoming digital world
  - Privacy
  - Effective commercial relations
  - Payment
  - Voting

Public-Key Schemes

- Diffie-Hellman Key Exchange
- El-Gamal Encryption
- El-Gamal Signature
- Digital Signature Standard
- RSA encryption and signature

Groups

- A set *G* and a binary operation + that is:
  - Closed: If a and b are in *G* then a + b is in *G*.
  - Associative: (a + b) + c = a + (b + c).
  - An identity element 0 : a + 0 = 0 + a = a.
  - Inverses: -a + a = a + -a = 0.
- Examples:
  - Integers under addition.
  - Reals except 0 under multiplication.
  - Right multiplication of nonsingular matrices.

Modular Groups

- $Z_n = (\{0,1,\ldots,n-1\}, + \bmod n)$, $n > 0$
  - Additive group of order (size) n.
  - Identity element is 0.
  - Inverse of a is -a mod n.
- $Z_p^* = (\{1,2,\ldots,p-1\}, \times \bmod p)$, p prime.
  - Multiplicative group of order p – 1.
  - Identity element is 1.
  - Inverses can be found using extended Euclid's algorithm.

Modular Groups

- Z_p^* is a special group, known as a cyclic group
- All elements of Z_p^* can be written as powers of a single element g, called the primitive element of the group
- E.g., In Z_7^*, take g = 3.

  Using modulo 7 arithmetic:

  $3^1 = 3$ $3^2 = 2$ $3^3 = 6$

  $3^4 = 4$ $3^5 = 5$ $3^6 = 1$

  Thus, g = 3 is a primitive element

Generators

Generally, A single element $g \in Z_p^*$ is called a generator of the group if every element of Z_p^* can be expressed as a finite power of {g}.

The **order** (ord(*g*)) , sometimes **period**, of an element *g* of the group is the smallest positive integer *m* such that

$$g^m = 1.$$

Clearly, *g* is a generator if and only if ord(*g*)=*p-1*.

Additionally, the set $\{1, g, g^2, \ldots, g^{ord(g)-1}\}$ is a subgroup of Z_p^*.

Generators

Theorem: Let g be an element of Z_p^*. The order of g divides p-1.

So, in order to check if an element g is a generator we need to verify only the set

$$\{g^a , a|(p-1): a \text{ divides } p-1\}.$$

Theorem: There are p-1 generators of the group $(Z_p^*, \times)$.

Hence, for a given generator g all the generators are represented as

$$\{g^a , \gcd(a,p-1)=1\}.$$

Order of an Element : example

Table shows the result of $a^i \equiv x \pmod 7$ for the group $G = \langle Z_7^*, \times \rangle$. In this group, $\theta(7) = 6$.

		$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
	$a = 1$	x: 1	x: 1	x: 1	x: 1	x: 1	x: 1
	$a = 2$	x: 2	x: 4	x: 1	x: 2	x: 4	x: 1
Primitive root →	$a = 3$	x: 3	x: 2	x: 6	x: 4	x: 5	x: 1
	$a = 4$	x: 4	x: 2	x: 1	x: 4	x: 2	x: 1
Primitive root →	$a = 5$	x: 5	x: 4	x: 6	x: 2	x: 3	x: 1
	$a = 6$	x: 6	x: 1	x: 6	x: 1	x: 6	x: 1

Order of an Element: example

- Z_7^* has 6 elements {1, 2, 3, 4, 5, 6}
- Elements {1, 2, 4} form a subgroup with $g = 2$ as the generator
- Elements {1, 6} form a subgroup with $g = 6$ as the generator
- Presence of subgroups like these pose

  a problem with the Diffie-Hellman algorithm

Public-Key Cryptography

- Idea: use separate keys to encrypt and decrypt
  - First proposed by Diffie and Hellman
  - Independently proposed by Merkle (1976)

- Pair of keys for each user
  - generated by the user himself
  - Public key is advertised
  - Private key is kept secret, and is computationally infeasible to discover from the public key and ciphertexts
  - Each key can decrypt messages encrypted using the other key

- Applications:
  - Encryption
  - Authentication (Digital Signature)
  - Key Exchange (to establish Session Key)

Key Exchange

All the ciphers mentioned previously require keys known

a-priori to all the users, before they can encrypt and decrypt.

To communicate securely, they need to have a common key, known only to them.

Possible key exchanges:

1. An a-priori meeting to exchange keys.

2. Sending the keys by a special courier.

3. Using an already existing common key, to encrypt additional keys.

Key Exchange

The problem: A meeting is required between the users (or their couriers), in order to be able to communicate securely.

Conclusion: Two users who have never met cannot communicate securely!

Key Exchange

- The model: A network of N users.
- In order to let any user to communicate securely to any other user, the keys should be distributed in advance in a secure way.
- Possible solution: Trusted center.
- Each user sets in advance a secret key with the trusted center. When user A wishes to communicate with user B, he chooses a new common key and sends it (encrypted under A’s key) to the center, who forwards it (encrypted under B’s key) to B.

Key Exchange

Drawbacks:

1. There should be somebody that all the users trust.

2. All the users should have a common key with the trusted center in advance.

3. The center can always understand all the users' messages (and can fake messages).

Requirements for Key Exchange

We prefer a solution that allows two users who

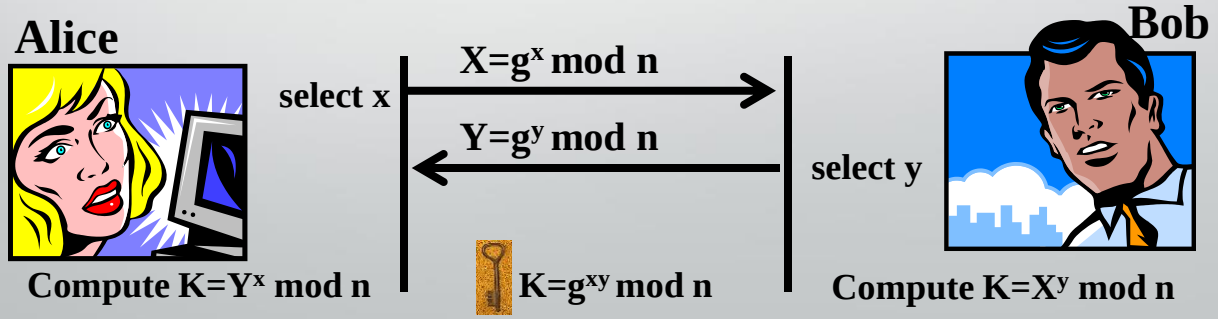
- have never met, and
- do not have any a-priori common information,

to decide on a common key

- known only to them, and
- unknown to anybody else, even to passive eavesdroppers who listen to their communication (but cannot modify it).

Diffie-Hellman Key Exchange

- First public-key algorithm, based on the difficulty of computing discrete logarithms modulo n
- Protocol:
  - Use key exchange protocol to establish session key
  - Use session key to encrypt actual communication
- Algorithm:
  - Choose a large prime n, and a primitive root g



Diffie-Hellman (D-H) Key Exchange Protocol

D-H model's primary contribution:

- Take a large prime p ($p > 2^{400}$) and a generator g;
- Public both g and p;
- Alice chooses some $x \in Z_p^*$ and sends the value of g^x to Bob
- Bob chooses some $y \in Z_p^*$ and sends the value of g^y to Alice
- Eve can see both g^x and g^y but she cannot calculate x or y. This is the basis for D-H.

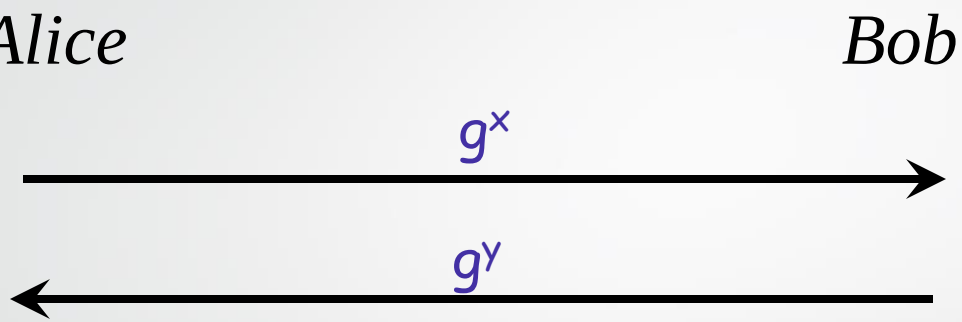
Diffie-Hellman Key Exchange Protocol

- In D-H model, Alice calculates the key

$$k = (g^y)^x \bmod p;$$

- And Bob similarly calculates the same key $k = (g^x)^y$;
- Since Eve does not know *x* or *y*, she cannot calculate the key k;
- Diffie and Hellman developed this method to share a key using some publicly available information;
- Diffie and Hellman did not develop the concept of private key.

Diffie-Hellman Key Exchange Protocol

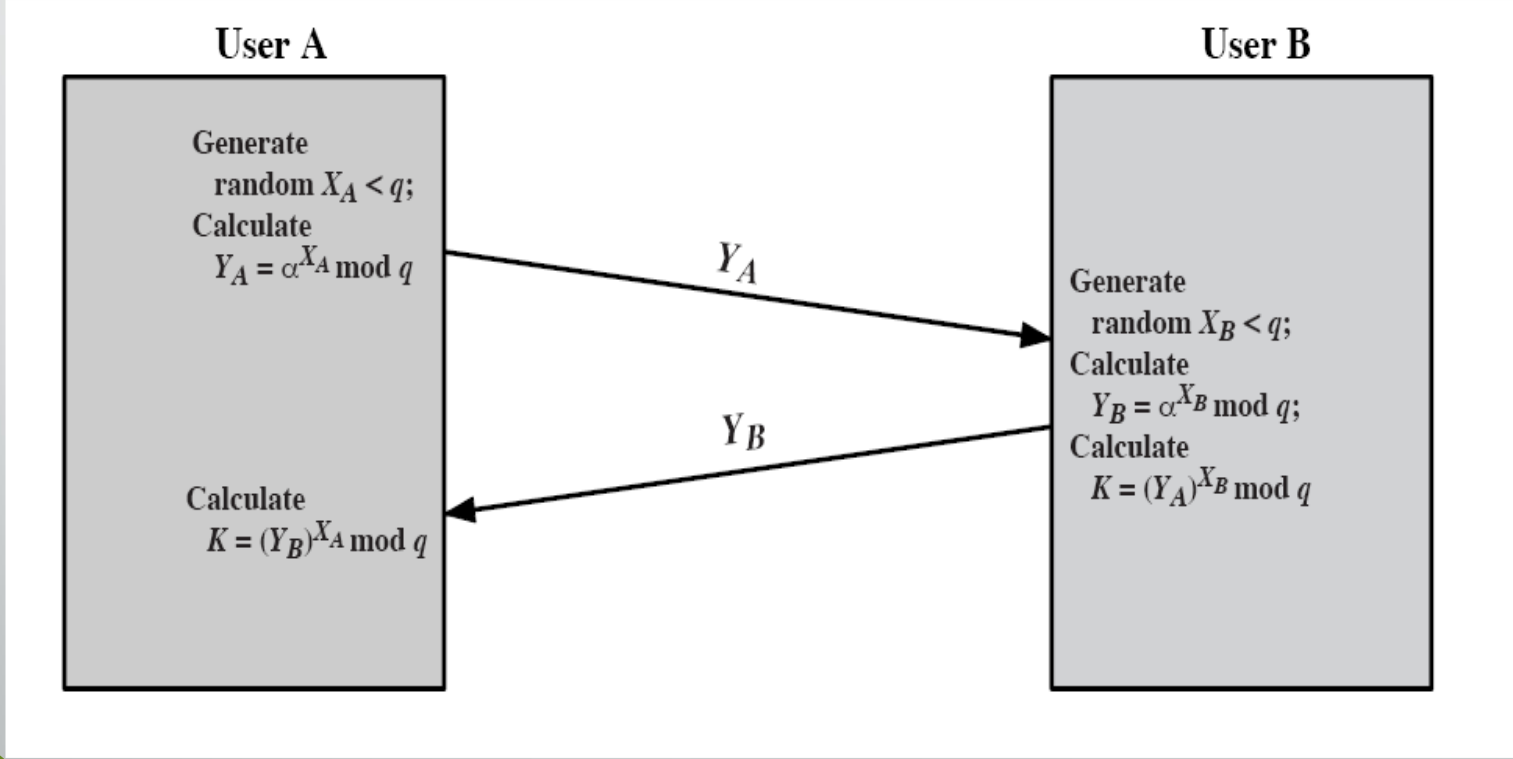


- both parties compute the secret key

$$k=g^{xy}=(g^x)^y=(g^y)^x$$

- assumes authenticated channels open to m-i-t-m in a realistic unauthenticated setting

Diffie-Hellman Key Exchange Protocol



Diffie-Hellman Key Exchange Protocol

- DH does not offer authentication
- Trudy can use a man-in-the-middle attack
  - Impersonating Alice to Bob and vice versa
  - Using his own key (or different keys) with each

- Solution: establish a public directory
  - Each person publishes (g,n,$\mathbf{g^x}$**)** – this is the public key
  - Note: g,n may be different from one user to another

- Make sure not to select x=0/1 mod n

Diffie-Hellman (D-H) Key Exchange Protocol

- In real number arithmetic, one can find x knowing g and g^x using logarithm functions;
- In finite field, to find x knowing g and g^x is known as the Discrete Logarithm problem

Discrete Logarithm Problem

The Problem: Solve for x if $y = g^x \pmod{p}$,

given you know y, g and p.

- Solution 2: Shank's Algorithm.
- This algorithm is more efficient, but it still requires $O(p^{1/2} \log(p))$ steps.
- This is not practical for large p.
- For example, if

p = 170141183460469231731687303715884105727, then it would take roughly $1.14824 \cdot 10^{21}$ steps to solve. (Each step requires many calculations).

- Even using Google's computers which are estimated to perform 300 trillion calculations per second, it would take roughly 5 years to solve.

Diffie-Hellman Key Exchange

Example:

- Suppose Alice and Bob agree to use $p = 47$ and $g = 5$.
- Alice chooses a number between 0 and 46, say $a = 18$.
- Bob chooses a number between 0 and 46, say $b = 22$.

- Alice publishes $g^a \pmod{p}$, i.e. u = 5^18 (mod 47) = 2.
- Bob publishes $g^b \pmod{p}$, i.e. v = 5^22 (mod 47) = 28.

Diffie-Hellman Key Exchange

- If Alice wants to know the secret key *k*, she takes Bob's public number, $v = 28$, and raises it to her private number, $a = 18$ (taking the result mod 47). This gives her: $28^{18} \pmod{47} = 24$.
- If Bob wants to know the secret key, he takes Alice's public number, $u = 2$, and raises it to his private number, $b = 22$ (taking the result mod 47). This gives him: $2^{22} \pmod{47} = 24$.
- Thus, Alice and Bob have agreed upon a secret key, $k = 24$.

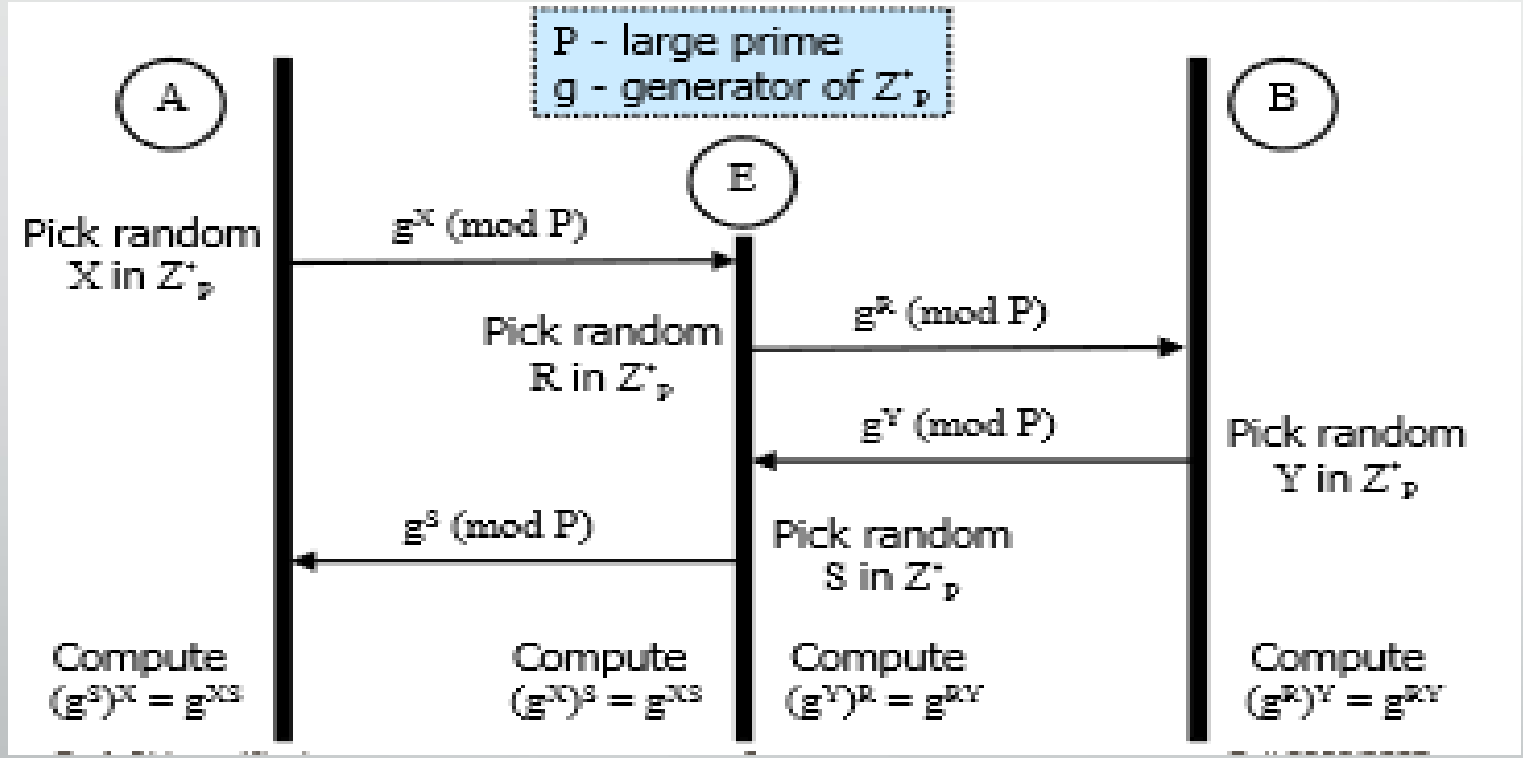
D-H Advantages

- Any user can choose a random x and publish g^x in a public database such as a phone book.
- Phone book must be maintained.
- Other users can look up the database and get the public key for the individual and use it to encrypt the message.
- Ideal for use with emails.

D-H Drawbacks

- Does not protect against man-in-the-middle attacks.
- Eve can intercept all traffic between Alice and Bob and generate separate keys for communication with them. In this scenario Eve originates the communications with Alice and Bob separately. It is lot of work but not too much of a problem.
- If Alice sends an encrypted message for Bob with his public key, Eve simply forwards it.

D-H – Man-in the-middle



D-H Drawbacks

- A major drawback is when Eve replaces Alice's g^x and Bob's g^y with the value 1. In this case the key $k = 1$ and so Eve will know the key
- One countermeasure is for Bob to check for the value 1 coming as the public key
- Eve could overcome this problem by using a value g that generates a small subgroup (say with only two elements)
- For large prime p, $(p-1)$ is an even number and so Z_p^* will have a subgroup of order 2

D-H Drawbacks

- If *g* is not a primitive element, then Eve can do an exhaustive search to find the key *k* since *g* would generate only a subgroup of Z_p^*
- One solution for the above problem is to choose the prime *p* as a safe prime,

  i.e., $p = 2 * q + 1$ where *q* is a prime
- E.g., 7 = 2 * 3 + 1. Thus, 7 is a safe prime.

D-H Drawbacks

- Does safe primes solve the problem?
- Answer: No
- Reason: For any prime *p*, among the numbers *1, 2, 3, ..., (p-1)*, exactly half are perfect squares and others are not
- E.g., Take $p = 7$. This gives us 1,2,3,4,5,6

  $1 = 6^2 \pmod 7$ $2 = 3^2 \pmod 7$

  $4 = 2^2 \pmod 7$

D-H Drawbacks

- Fact: If x is even, then g^x is a square

  If x is odd, then g^x is a non-square

- Eve can easily determine if x is even or odd by knowing g^x
- It makes Eve's job a bit easier in finding out x
- To avoid the problem use only the squares modulo p