1. Into

2. Symmetric crypto

Symmetric crypto relies on the fact, that two sides know the same key, which they obtain by another secure channel. The same key used for encryption and decryption.

Issues:
1. Key distribution problem
2. Number of keys on N users is n(n-1)/2
3. No protection against cheating between Alice and Bob

3. Asymmetric encryption.

Public key cryptography - is a method of encrypting, which allows individuals to securely communicate without sharing the secret key.

Trapdoor one way function - function that can be easily computed in one way, and hard in the inverse without special(secret) information(trapdoor). In this case forward -> encrypt. Inverse -> decrypt.

Private key - is key to decrypt message. Public key - is key to encrypt message. Or vice versa

For signatures private key enables to sign and public to verify this sign validity.

Arithmetic functions, that considered as a one way(not proofed)
1. Multiplication and factorization
2. x^y mod n = z => find y
3. x^y mod n = z => find x
4. x^2 mod n. n - not prime, jacobi(z/n)=1 => find x
5. g^ab mod p => find a

Also use cases:
1. Key exchange
2. Identification
3. Signature to cannot deny having sent/received a message

4. Prime numbers

Prime number an integer P which has exactly two positive divisors(1 and P).

5. Alternative problems

6. Group

Group is set of elements, that are related to each other according to certain well-defined rules.

Z_p^* - is a group with nonzero integers between 1 and p-1 modulo some prime nomber p.

6.1. Axioms

Operation for example is a multiplication.

1. Closure - $a, b \in G, a * b = c \to c \in G$
2. Associativity $a * (b * c) = (a * b) * c$
3. Identity existence $a * 1 = a$
4. Inverse existence $a * b = 1$

A group is commutative if $a * b = b * a$ A group is cyclic if there is generator g

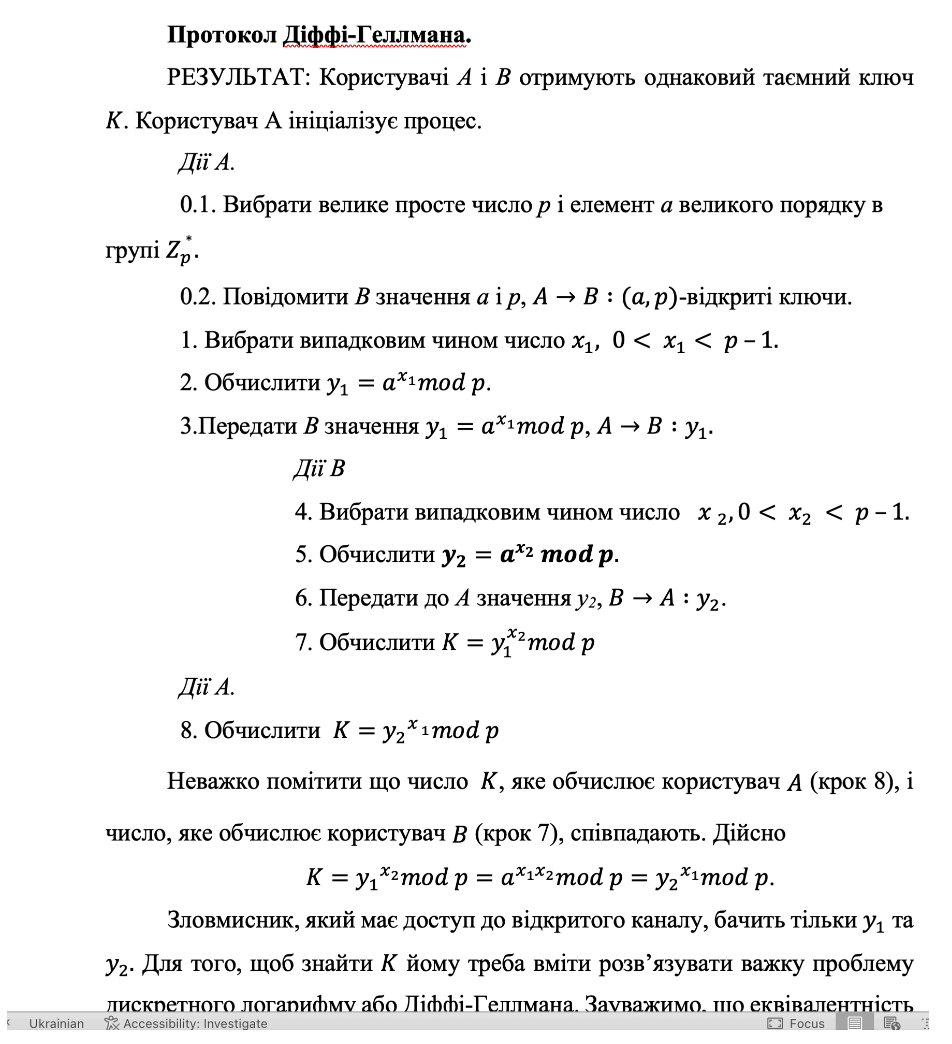
$$g \in G, \forall x \in G, \exists n : g^n = x$$

6.2. How to choose or check that a is a generator?

Theorem $g \in Z_p^*$ is a generator of Z_p^* if and only if $g^{\frac{p-1}{q}} \neg \equiv 1 \bmod p$ for all primes q such that $q|(p-1)$

7. Diffie-hellman

Protocol:


Протокол Діффі-Геллмана.

РЕЗУЛЬТАТ: Користувачі A і B отримують однаковий таємний ключ K. Користувач A ініціалізує процес.

Дії A.

0.1. Вибрати велике просте число p і елемент a великого порядку в групі Z_p^*.

0.2. Повідомити B значення a і p, $A \to B : (a, p)$-відкриті ключи.

1. Вибрати випадковим чином число x_1, $0 < x_1 < p - 1$.

2. Обчислити $y_1 = a^{x_1} mod\ p$.

3.Передати B значення $y_1 = a^{x_1} mod\ p$, $A \to B : y_1$.

Дії B

4. Вибрати випадковим чином число $x_2, 0 < x_2 < p - 1$.

5. Обчислити $\boldsymbol{y_2 = a^{x_2}\ mod\ p}$.

6. Передати до A значення y_2, $B \to A : y_2$.

7. Обчислити $K = y_1^{x_2} mod\ p$

Дії A.

8. Обчислити $K = {y_2}^{x_1} mod\ p$

Неважко помітити що число K, яке обчислює користувач A (крок 8), і число, яке обчислює користувач B (крок 7), співпадають. Дійсно

$$K = {y_1}^{x_2} mod\ p = a^{x_1 x_2} mod\ p = {y_2}^{x_1} mod\ p.$$

Зловмисник, який має доступ до відкритого каналу, бачить тільки y_1 та y_2. Для того, щоб знайти K йому треба вміти розв'язувати важку проблему дискретного логарифму або Діффі-Геллмана. Зауважимо, що еквівалентність

8. Rsa

Target: send information from A to B securely.

Public key can encrypt.

Private key can decrypt.

1. Choose 2 prime int p,q
2. Count n = p*q
3. Count Euler function $\varphi(n) = (p-1)(q-1)$
4. Count $e : \gcd(e, \varphi(n)) = 1$
5. Solve $ex = 1 \bmod \varphi(n)$, find x = d

X represents a number in Z_n^*, binary value of X must be less than n.(also n and x are coprime, can just check that $\neg x|e$)

Public key is a pair of (n, e)

Y is a ciphertext. $Y = (x^e \bmod n)$

Decryption: the private key is d.

$$y^d \bmod n = x$$

$x^{ed} \bmod n = x \bmod n$

$$x^{ed-1} \bmod n = 1 \bmod n$$

It's possible only when:

$ed - 1 = k\varphi(n)$

$$x^{k\varphi(n)+1} \bmod n = 1x \bmod n$$

First, choose p and q, count n=p*q, count $\varphi(n) = (p-1)(q-1)$

Second, choose e coprime to $\varphi(n)$. Solve equation $ed = 1 \bmod \varphi(n)$. $d = e^{-1} \bmod \varphi(n)$

x, y, n, d are large numbers(1024 bits or more)

e, d

8.1. Rsa also vulnerable to man in the middle attack

Example: https://security.stackexchange.com/questions/189468/why-can-a-man-in-the-middle-attack-not-happen-with-rsa

9. Elgamal encryption scheme

1. Perform DH
2. Use Key as a mask for the message modulo p

9.1. Elgamal protocol

1. Set up phase

Done only once. Bob makes p, α, b. Compute $B = \alpha^b \bmod p$

$$k_{\text{pub}} = (p, \alpha, B)$$

Bob publishes keys.

2. The encryption phase

Executed every time

Alice chooses ephemeral key $K_E == \alpha^a \bmod p$ Alice computes the "shared key"(private) $K = B^a \bmod p$

$$y - xK \bmod p$$

And sends K_E and y to Bob

3. The decryption phase

Executed every time

$$X = YK^{-1} \bmod p$$

$$m = K^{-1} Y \bmod p$$

$$K = g^{ab} \bmod p$$

By fundamental theorem of arithmetic

$$\gcd(g, p) = 1 \to \gcd(g^n, p) = 1$$

by fermat's little theorem:

$$K_E^{p-1} \equiv 1 \bmod p$$

$$(g^b)^{p-1} \equiv 1 \bmod p$$

$$(g^b)^{p-1} \equiv 1 \bmod p$$

$$(g^b)^{p-1} \equiv 1 \bmod p$$

$$(g^b)^{p-1} \equiv (g^b)^{p-1-a} g^{ab} \equiv 1 \bmod p$$

$$g^{ab} = K$$

$K_{-1} = (g^b)^{p-1-a} \bmod p$

10. Coin flipping

Using DLP Given two generators a, b

Bob chooses one(f.e. a) and pow it to x, sending a^x to alice. Alice chooses a or b and sends it to bob. Bob sends x to Alice. Alice checks a^x.

Note: there are possibility that $a^x == b^x$

11. Lampart algorithm