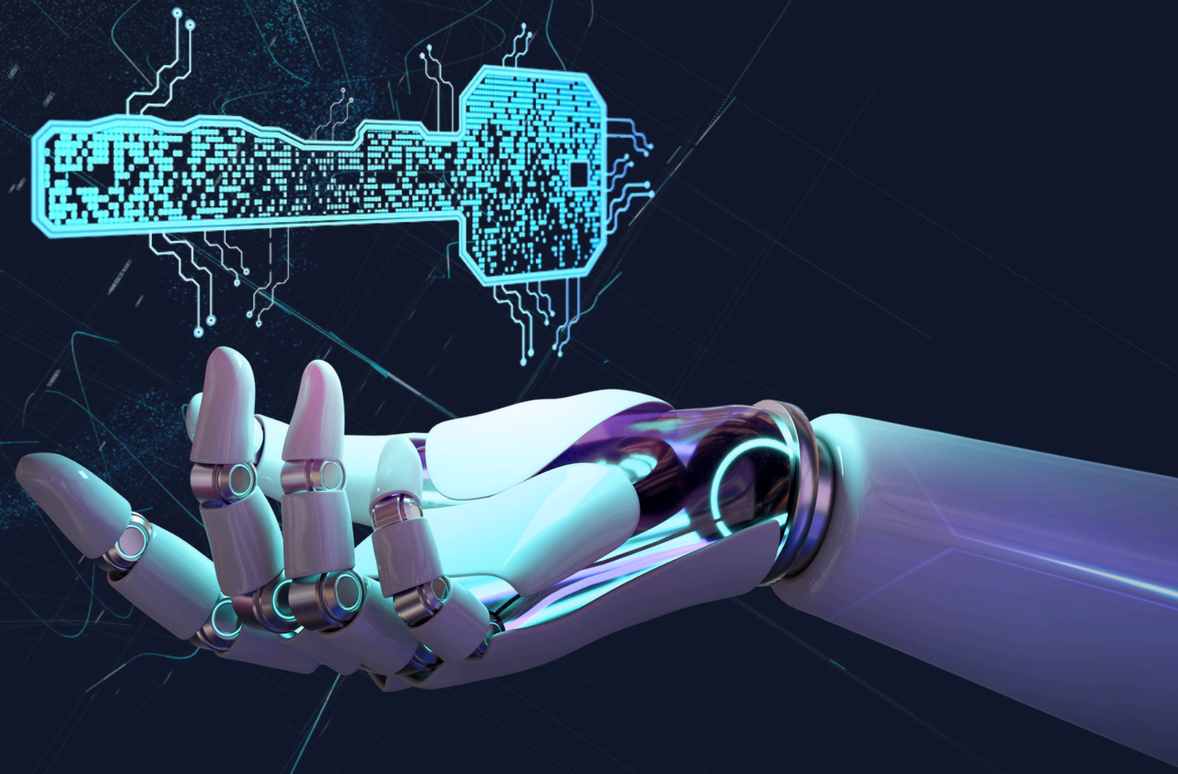
Data Security and Cryptology
Final Project



Group 31

Kfir Hemo

Ron Shahar

Lior Jigalo

Nitsan Maman

Welcome to our project presentation on secure Email communication!

Our discussion topics for today:

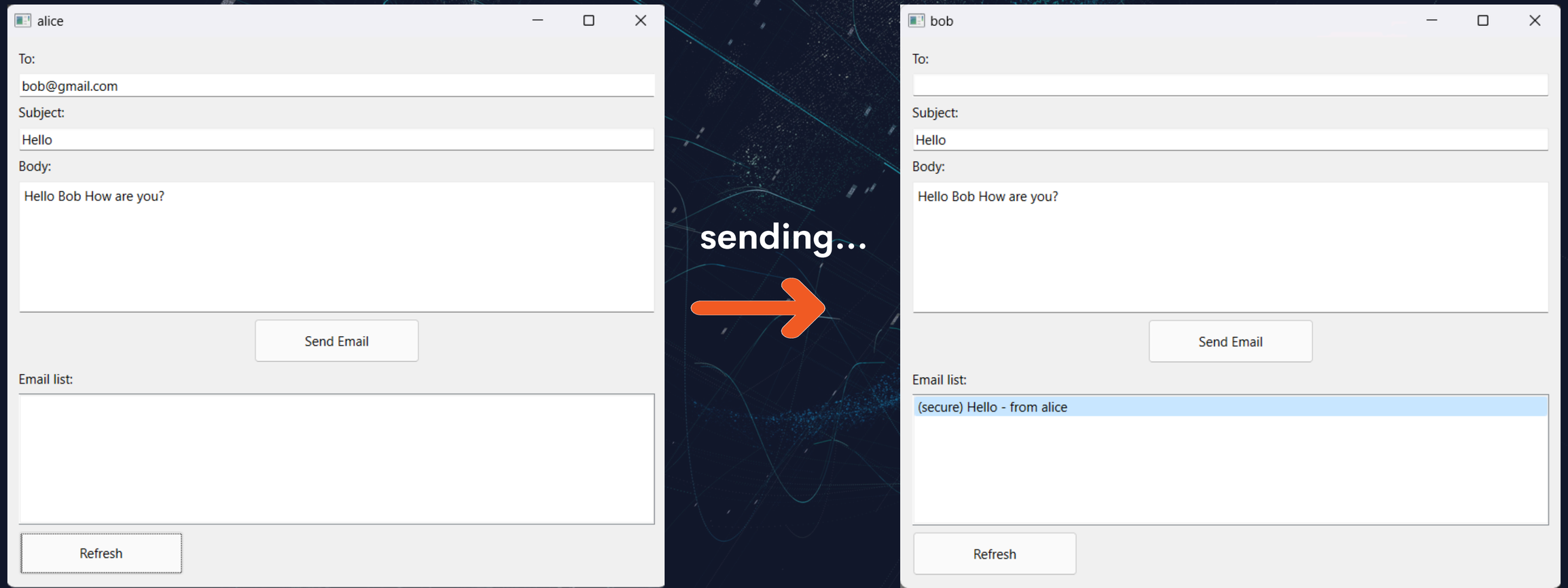
- Introduce Our Application

- The accompanying encryption and digital signature algorithms:
  - BlowFish Cypher (In OFB mode)
  - EC EL-GAMAL
  - Rabin Signature

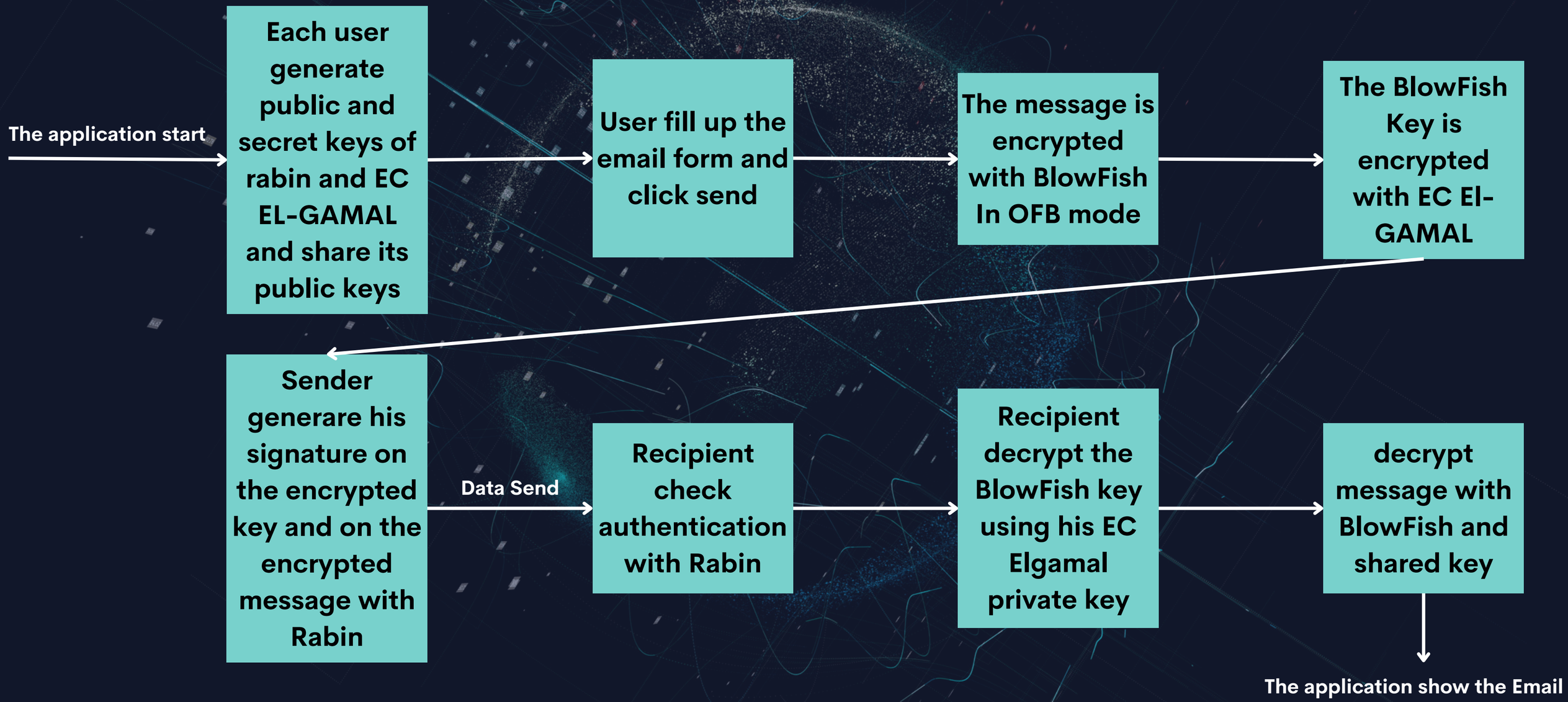
- Demonstrate how we integrate the functionalities of all algorithms into a cohesive system.

Introduction:

An application employs the BLOWFISH Cipher in OFB mode for secure email encryption and decryption, ensuring the confidentiality of communications. To securely deliver the secret key, it utilizes EC EL-GAMAL for encryption coupled with Rabin signatures for authentication and integrity checks.



Application Event Flow



BlowFish Cypher

Blowfish is a symmetric-key block cipher designed by Bruce Schneier in 1993.
as improvement to “DES” that started to age and become slow and insecure.

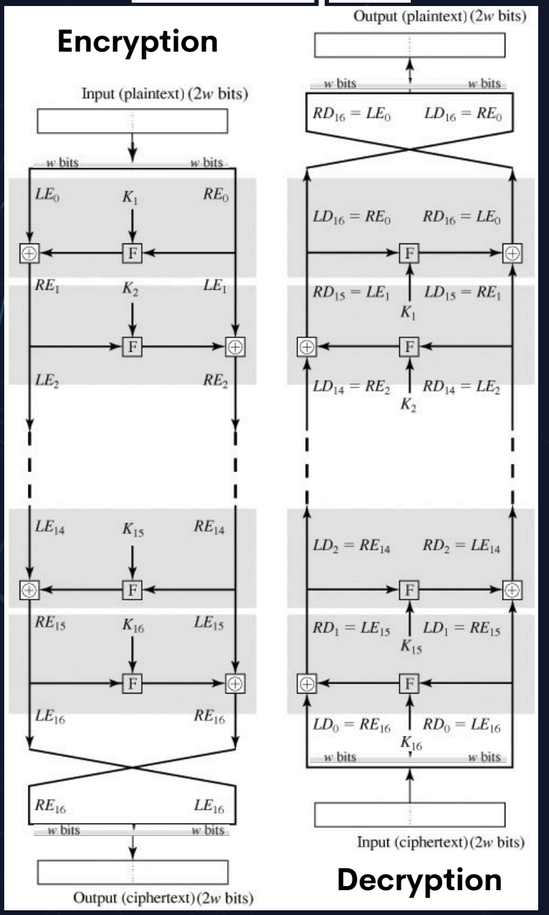
It is a 16-round Feistel cipher, It operates on 64-bit blocks and supports key sizes ranging from 32 bits to 448 bits.

Using Feistel network structure, making it fast and relatively simple.

First round of decryption is the same as the input to the last stage of encryption:
LD1 = RE15 and RD1 = LE15 , the F function doesn’t need to be reversible.

Blowfish consists of two main components:
a **key-expansion** part and a **data encryption** part.

Feistel cipher



BlowFish Cypher

Generating the subkeys - key-expansion:

In Blowfish, the P-array consists of 18 32-bit subkeys and four 32-bit S-boxes with 256 entries each 4KB.

The subkeys are calculated as follows:

1. The P-array and S-boxes are initialized with a fixed string of hexadecimal digits of pi.
2. XOR P1 with the first 32 bits of the key, XOR P2 with the second 32 bits of the key and so on until you have XORed all the P array. The key is just cycled through.
3. Encrypt all zero string with the blowfish algorithm, using the P-array subkeys derived in steps 1 and 2.
4. Replace P1 and P2 with the output of step 3.
5. Encrypt the output of step 3 using blowfish algorithm.
6. Replace P3 and P4 with the output from step 5.
7. Reapeat 512 times until all the P-arrays and S-boxes are modified.

BlowFish Cypher

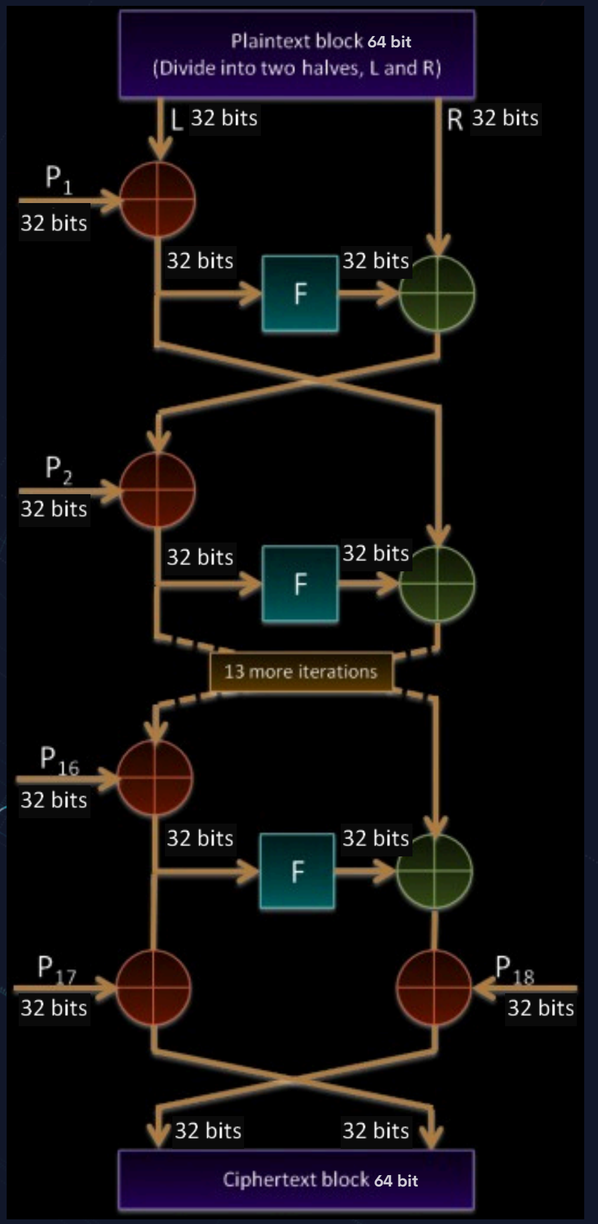
Feistel cipher – data encryption:

The Feistel cipher for blowfish algorithm barely differs from the original one.

Instead of key (K) you'll have to XOR the left side of data with Pi before running it through Feistel function.

Even though blowfish is 16 round Feistel network,
It has 18 P values, where at the very end both data sides are XORed with P17 and P18 – output whitening

Output whitening is a technique used to add an extra layer of security to the encryption process by XORing the output data with a key (or keys) one more time. This makes it harder for attackers to exploit patterns



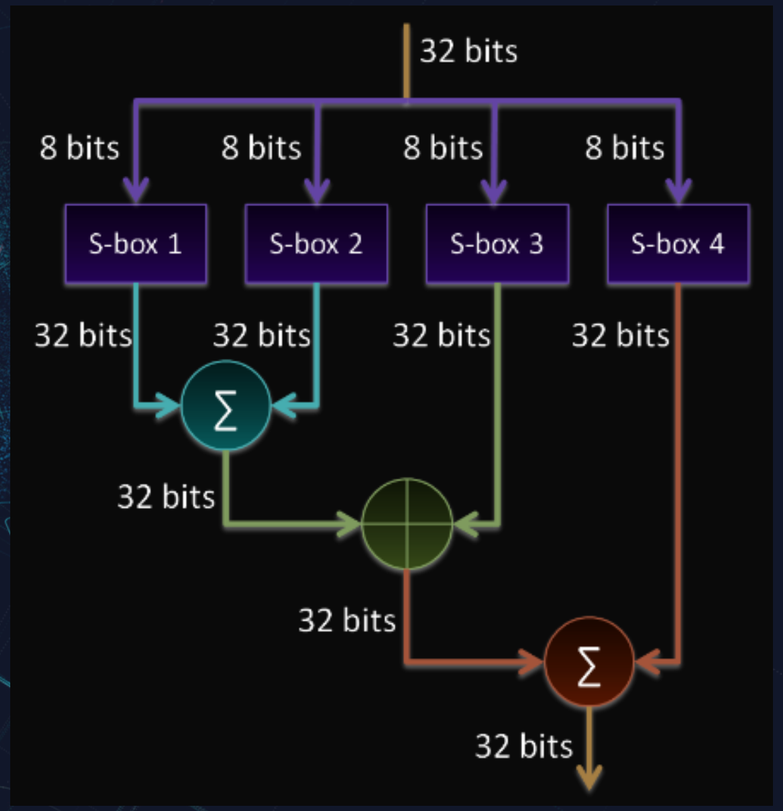
BlowFish Cypher

Feistel function:

One final thing we have to implement for any Feistel network is the Feistel function.

in case of Blowfish the Feistel function:
Divides the input of 32 bits into 4 8bit chunks
which will be used to lookup the corresponding S-box value.

Also in the meantime it will use those S-box values to further scramble the data.

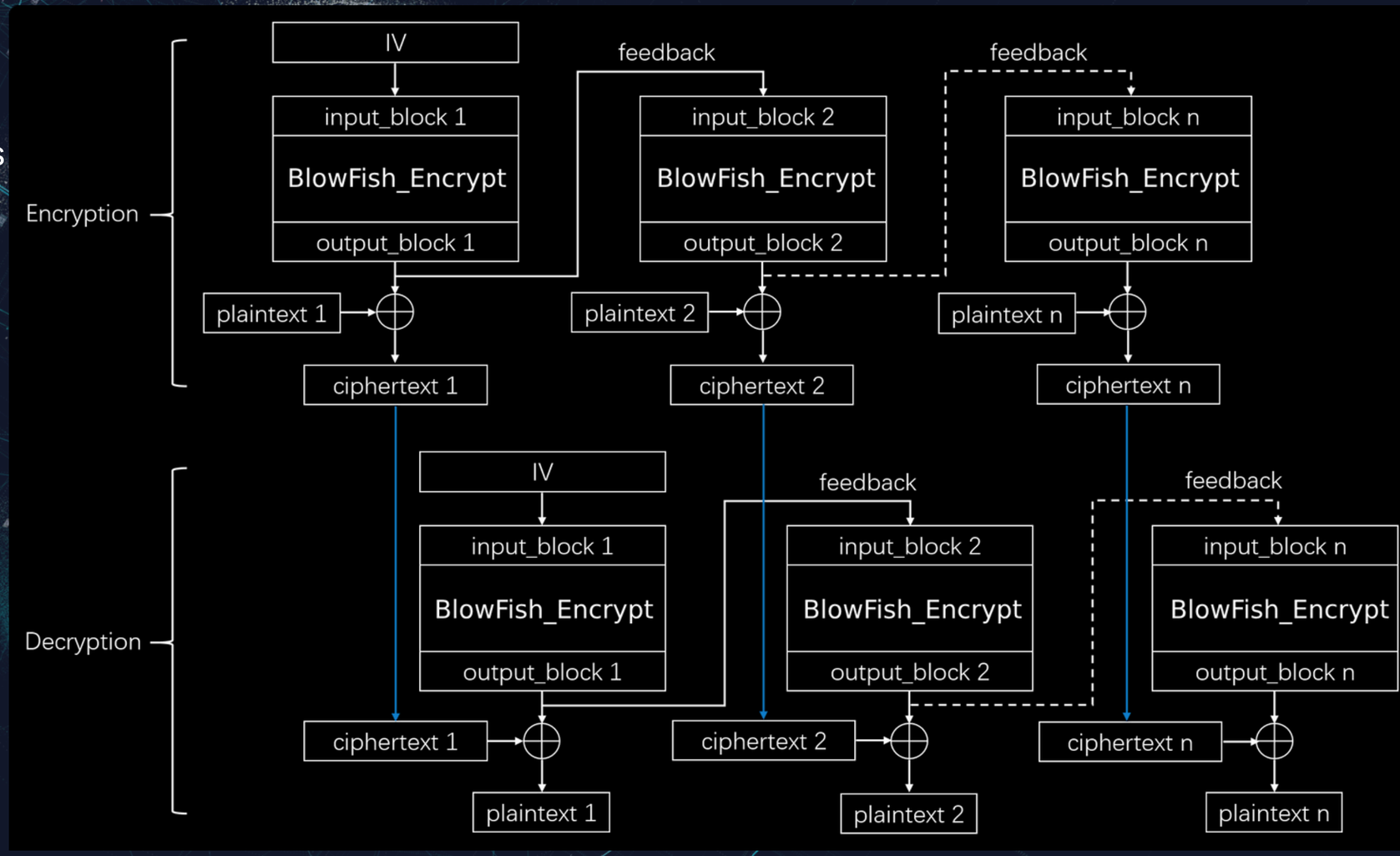


OFB MODE

The Output Feedback mode relies on XOR-ing plaintext and ciphertext blocks with expanded versions of random initialization vector.

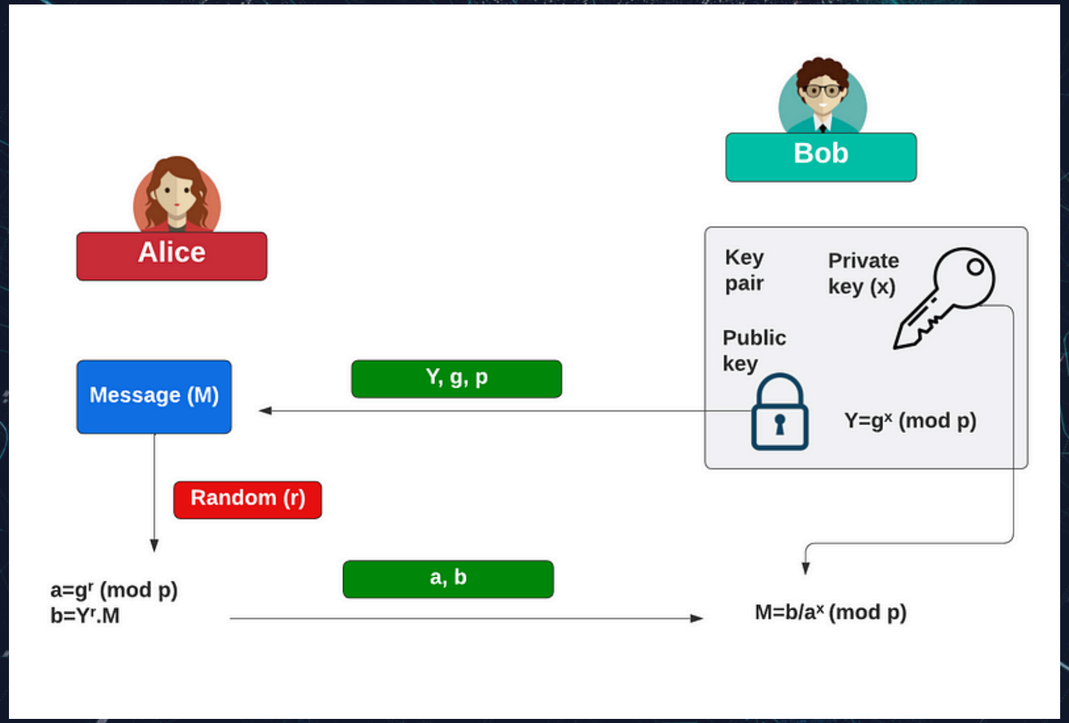
It holds great resistance towards bit transmission errors.

It also decreases the dependency or relationship of the cipher on the plaintext.



El-Gamal

An asymmetric key encryption algorithm for public-key cryptography, based on the Diffie–Hellman key exchange, by Taher Elgamal 1985.



At the core of the ElGamal public key methods is the Discrete Logarithm Problem,

Y= g^x (mod p)

and where it is difficult to determine x,

even if we have Y, g and p (as long as p is a large enough prime number).

Elliptic-Curve Cryptography

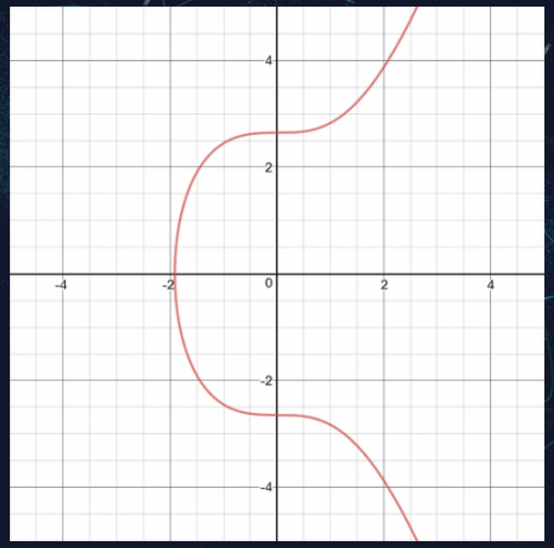
Elliptic curve cryptography (ECC) is a public key cryptographic algorithm used in encryption, authentication, and digital signatures.

ECC is based on the elliptic curve theory, which generates keys through the properties of the elliptic curve equation, compared to the traditional method of factoring very large prime numbers.

General Mathematical Form:

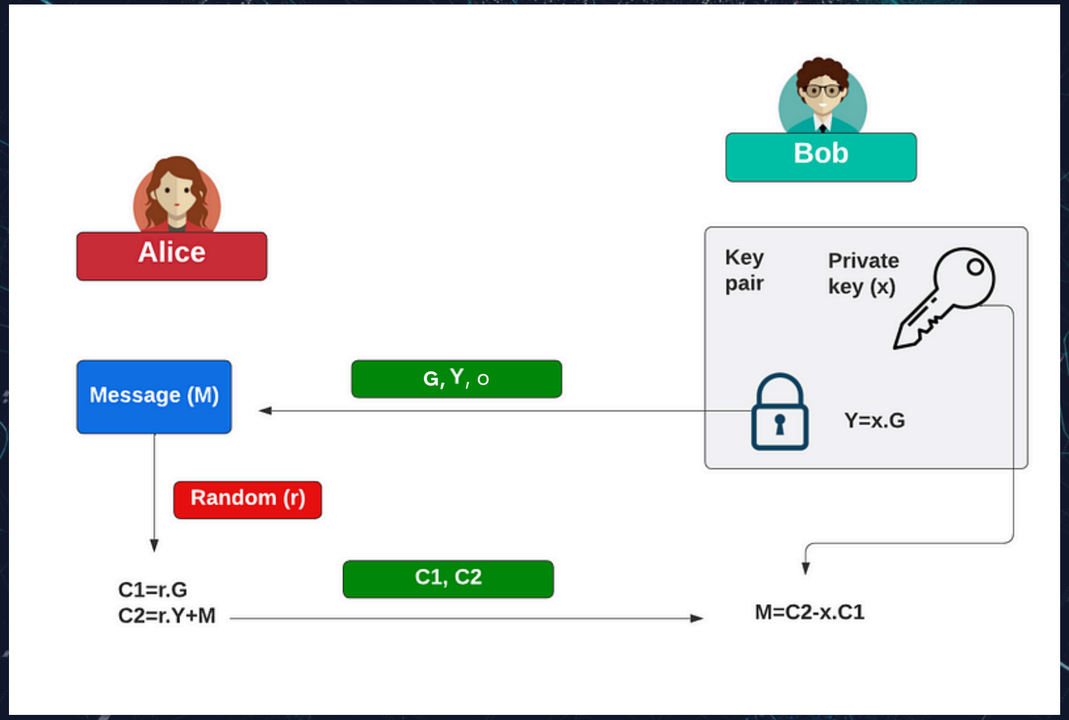
$$y^2 = x^3 + ax + b$$

This is a graph of secp256k1's elliptic curve y^2 = x^3 + 7



EC ElGamal

So now we can convert the ElGamal method into ECC



private key (x) - which is a random scalar value

public key (G, Y, o) -

- G - Start point on the elliptic curve
- Y - end point on the elliptic curve
- o - Finite field (modulo)

Rabin Signature

Theoretical Background:

- The Rabin Signature Algorithm, proposed by Michael O. Rabin in 1978, is a digital signature method.
- It's based on the challenge of computing modular square roots and leverages the hardness of factoring large prime numbers for its security.
- The algorithm utilizes two large prime numbers to generate a private and a public key, where the public key is the product of these primes.

Why Rabin Signature?

- Offers strong security based on the difficulty of factoring large numbers, a well-researched area in number theory.
- Simpler and less computationally intensive compared to other signature schemes, making it efficient and practical for various applications.

Rabin Signature Algorithm Overview:

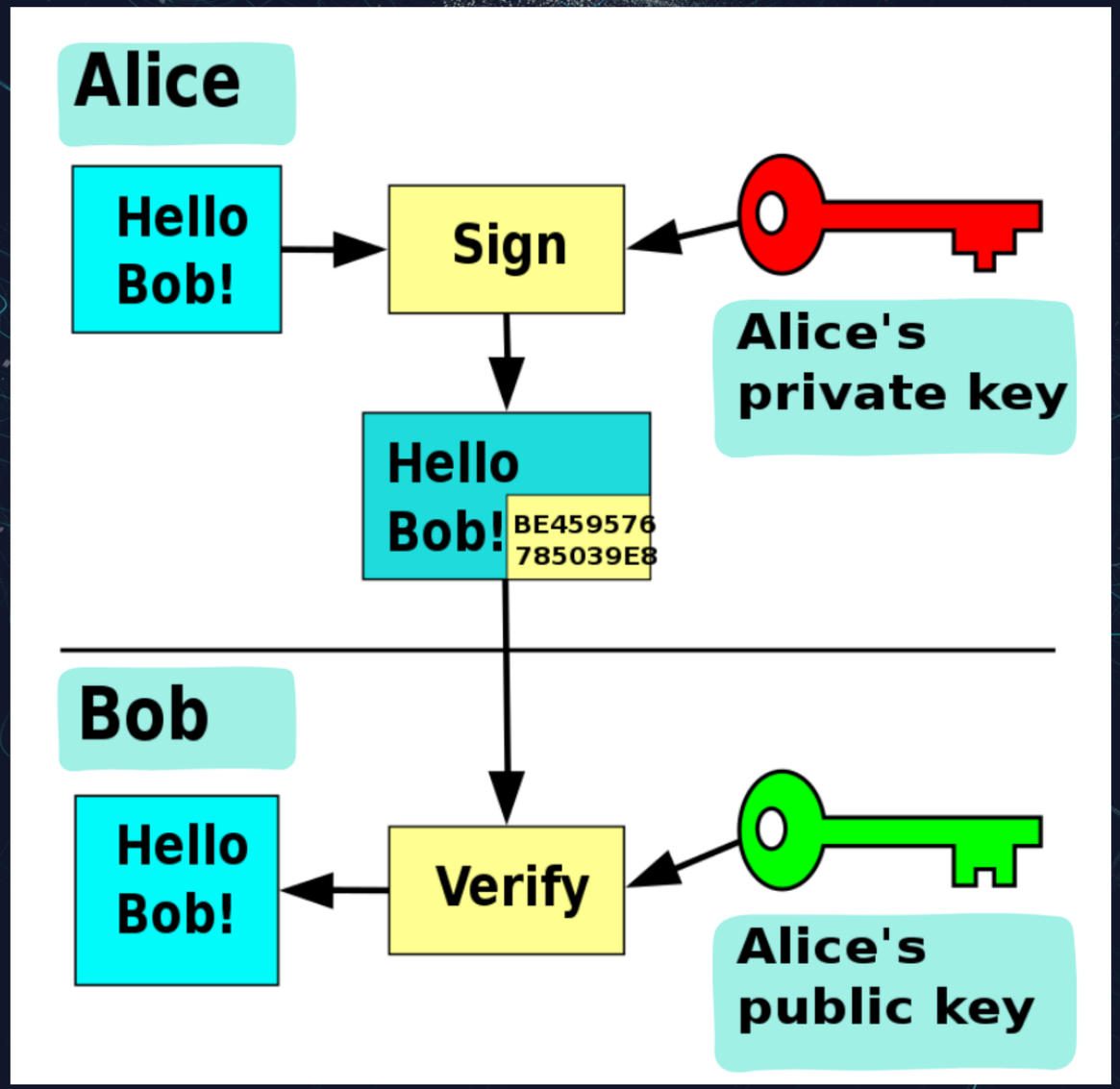
Creating a Signature:

1. Start with two selected odd primes (p and q), ensuring they are of the form 4k+3.
2. Compute n = p * q; (n, b) becomes the public key, while p and q remain private.
3. Use a hash function, H, to process the message string combined with a random string U, producing an integer c, ensuring c < n.
4. Find a solution x to a specified congruence that involves n, c, and the random string U. If a solution isn't feasible, a new U is chosen, and the process is repeated.
5. The signature is the pair (U, x) derived from this process.

Rabin Signature - Verifying a Signature:

- To confirm a message's authenticity, calculate c = H(M+U) and check if the signature satisfies the necessary congruence with x and b.
- If the conditions are met, the message is authentic; otherwise, it's not.

Rabin Signature



The combined Approach:

Alice

Lets Mix All Algorithms Together

Bob

Phase 0

Alice and **Bob** both generate a public and secret keys for the digital signature for future use and exchage the public keys

Digital Signature Keys

Private key = (p,q) such that p,q are large primes

Public key = (J,d) such that

$$J = (p \times q)$$

and

$$d = \left((10^9 + 7) \times 2^{(p-1)(q-1)-1}\right) mod J$$

Alice

Bob

Digital Signature Keys

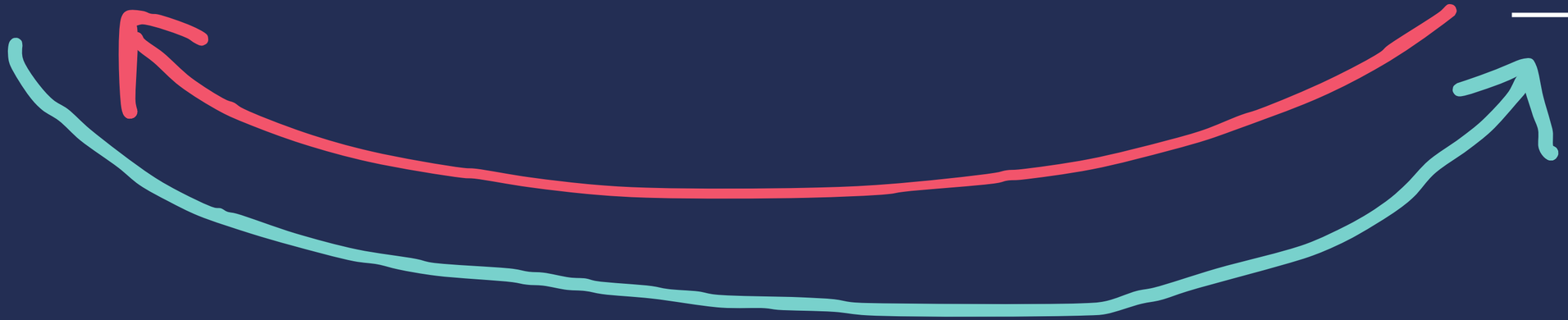
Private key = (u,t) such that p,q are large primes

Public key = (F,b) such that

$$F = (u \times t)$$

and

$$b = \left((10^9 + 7) \times 2^{(u-1)(t-1)-1}\right) mod F$$



Phase 1

Alice and Bob both generate a Point A and B on the curve as a public key.
Then they choose a secret private key n and s and both calculate the second part of the public key Z and H respectively.
Finnaly, each generate the size of the finite field y and o respectively as part of the public key and then send the public keys to each other

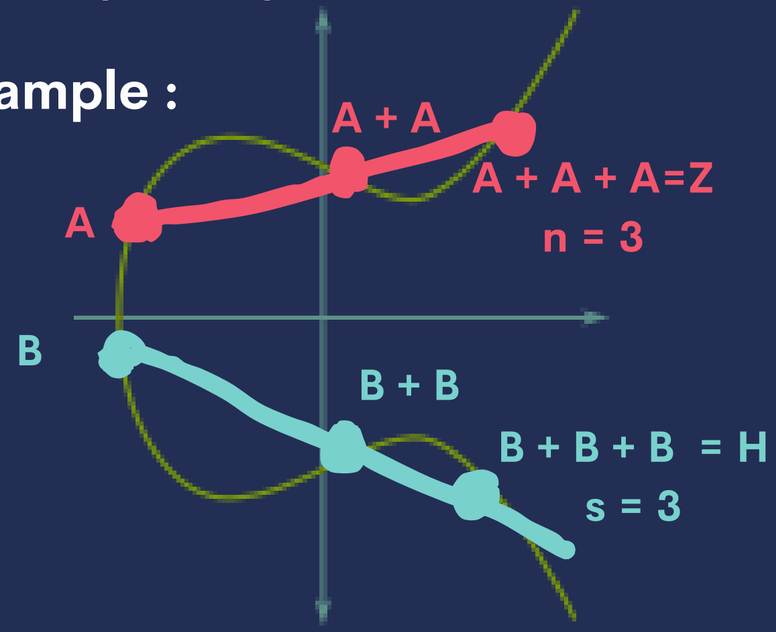
Digital Signature Keys
Private key = (p,q)
Public key = (J,d)
Public key = (F,b)

EC Elgamal Keys
Private key = s
Public key = (B,H,o)
Such that H :

$$H = (s \times B)$$

Alice

Example :



Bob

Digital Signature Keys
Private key = (u,t)
Public key = (F,b)
Public key = (J,d)

EC Elgamal Keys
Private key = n
Public key = (A,Z,y)
Such that Z :

$$Z = (n \times A)$$

Phase 2

Alice decides to send a Mail message m to Bob

Digital Signature Keys

Private key = (p,q)

Public key = (J,d)

Public key = (F,b)

EC Elgamal Keys

Private key = s

Public key = (B,H,o)

Public key = (A,Z,y)

Plaintext : m

Alice

Bob

Digital Signature Keys

Private key = (u,t)

Public key = (F,b)

Public key = (J,d)

EC Elgamal Keys

Private key = n

Public key = (A,Z,y)

Public key = (B,H,o)

Phase 3

Alice Generates a new Secret Key k to encrypt Plaintext m using Blowfish symmetric algorithm (In OFB mode)

Digital Signature Keys
Private key = (p,q)
Public key = (J,d)
Public key = (F,b)

EC Elgamal Keys
Private key = s
Public key = (B,H,o)
Public key = (A,Z,y)

Plaintext : m
Blowfish secret key: k

Alice

Bob

Digital Signature Keys
Private key = (u,t)
Public key = (F,b)
Public key = (J,d)

EC Elgamal Keys
Private key = n
Public key = (A,Z,y)
Public key = (B,H,o)

Phase 4

Alice encrypt the Plaintext **m** using **k** and the Blowfish algorithm (In OFB mode) and get Ciphertext **c** (with IV inside)

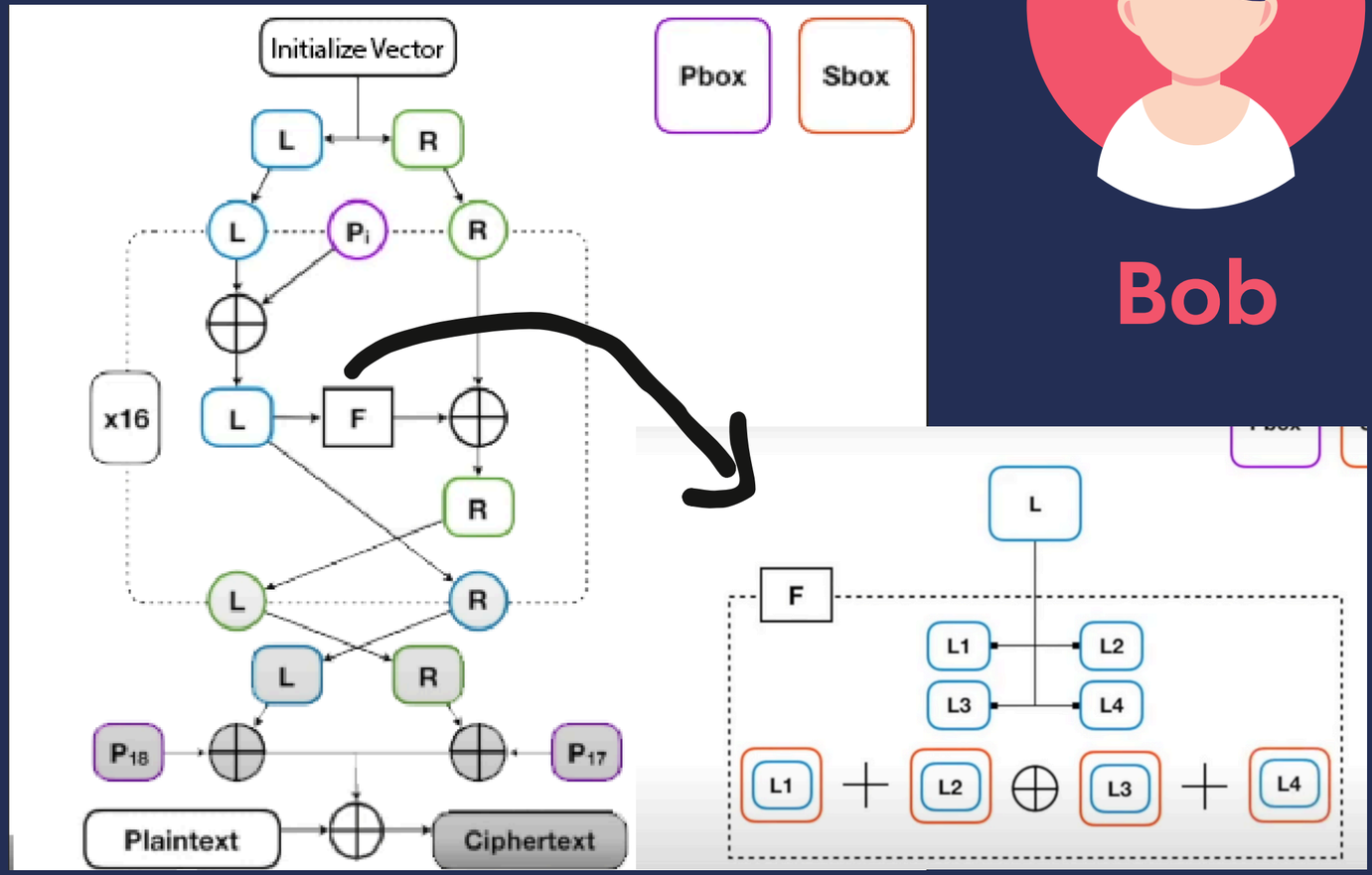
Digital Signature Keys
Private key = (p,q)
Public key = (J,d)
Public key = (F,b)

EC Elgamal Keys
Private key = s
Public key = (B,H,o)
Public key = (A,Z,y)

Plaintext : m

Blowfish
Blowfish secret key: k
Ciphertext : c

Alice



Bob

Digital Signature Keys
Private key = (u,t)
Public key = (F,b)
Public key = (J,d)

EC Elgamal Keys
Private key = n
Public key = (A,Z,y)
Public key = (B,H,o)

```
encrypted_data_with_iv = iv + encrypted_data
```

Phase 5

Alice encrypt the Blowfish secret key: k Using the a-symmetrical algorithm EC Elgamal and Bob's EC Elgamal Public key = (A,Z,y)

Process :

Alice generate a random i < y

Using A,Z and i, Alice calculate W and G :

$$W = i \times A$$
$$G = i \times Z + k$$

Such that k is the secret key for the Blowfish algorithm

Digital Signature Keys

Private key = (p,q)

Public key = (J,d)

Public key = (F,b)

EC Elgamal Keys

Private key = s

Public key = (B,H,o)

Public key = (A,Z,y)

Plaintext : m

Blowfish

Blowfish secret key: k

Ciphertext : c

Alice

Bob

Digital Signature Keys

Private key = (u,t)

Public key = (F,b)

Public key = (J,d)

EC Elgamal Keys

Private key = n

Public key = (A,Z,y)

Public key = (B,H,o)

Phase 6

Alice generate a signature for each :
W,G (EC Elgamal output) and
Ciphertext : c
Using Rabin signature algorithm.
For each message W,G,c Alice will generate R = (U, x)
For Example W :

$$R_W = (U_{W,x_W})$$

Such that

$U_W =$ Random String of 60 bits

And

$$x_W = (V_1 + V_2 - d) mod J$$

Such that

$$V_1 = \left(r^{roundDown\left(\frac{p+1}{4}\right)} \times q \times q^{p-2} \right) mod p$$

$$V_2 = \left(r^{roundDown\left(\frac{q+1}{4}\right)} \times p \times p^{q-2} \right) mod q$$

$$r = Hashval\,(W + U_W) + d^2$$

We will check if r is a quadratic residue of mod q and mod p. If not we will change Uw until we get the right r

Digital Signature Keys
Private key = (p,q)
Public key = (J,d)
Public key = (F,b)

EC Elgamal Keys
Private key = s
Public key = (B,H,o)
Public key = (A,Z,y)

Plaintext : m

Blowfish
Blowfish secret key: k
Ciphertext : c

EC Elgamal output

W
G

Alice

Bob

Digital Signature Keys
Private key = (u,t)
Public key = (F,b)
Public key = (J,d)

EC Elgamal Keys
Private key = n
Public key = (A,Z,y)
Public key = (B,H,o)

Phase 7

Alice send to Bob the following :

$$(W, R_W)$$
$$(G, R_G)$$
$$(c, R_c)$$

Digital Signature
Keys
Private key = (p,q)
Public key = (J,d)
Public key = (F,b)

EC Elgamal
Keys
Private key = s
Public key = (B,H,o)
Public key = (A,Z,y)

Alice

Plaintext : m
Blowfish
Blowfish secret key: k
Ciphertext : c

EC Elgamal output+ Rabin Signature

$$(W, R_W)$$
$$(G, R_G)$$
$$(c, R_c)$$

Bob

Digital Signature
Keys
Private key = (u,t)
Public key = (F,b)
Public key = (J,d)

EC Elgamal
Keys
Private key = n
Public key = (A,Z,y)
Public key = (B,H,o)

Phase 8

Bob Verify using the Rabin signature algorithm that the message was sent by Alice and that the message was not tampered with.

For Each W,G,c Bob calculates :

For example : (W, R_W)

Alice

Digital Signature Keys

Private key = (p,q)

Public key = (J,d)

Public key = (F,b)

EC Elgamal Keys

Private key = s

Public key = (B,H,o)

Public key = (A,Z,y)

Plaintext : m

Blowfish

Blowfish secret key: k

Ciphertext : c

EC Elgamal output+ Rabin Signature

(W, R_W)

(G, R_G)

(c, R_c)

Bob

Digital Signature Keys

Private key = (u,t)

Public key = (F,b)

Public key = (J,d)

EC Elgamal Keys

Private key = n

Public key = (A,Z,y)

Public key = (B,H,o)

Received from Alice

(W, R_W)

(G, R_G)

(c, R_c)

rightSide = Hashval(W+ U_W) + d^2

leftSide = $X_W \times (X_W + 10^9 + 7)$

If :

(rightSide)Mod J == (leftSide)Mod J

It means that the message is authentic, from Alice and no one tampered with it.

Phase 9

Bob decrypt W and G using his EC Elgamal private key n
we expect to get the
Blowfish secret key: k
that will help us later decrypt
Ciphertext : c using Blowfish

Let L be :

$$L = n \times W$$

Now lets calculate Alice's Blowfish secret key: k

$$k = G - L$$

Digital Signature Keys
Private key = (p,q)
Public key = (J,d)
Public key = (F,b)

EC Elgamal Keys
Private key = s
Public key = (B,H,o)
Public key = (A,Z,y)

Alice

Plaintext : m

Blowfish
Blowfish secret key: k
Ciphertext : c

EC Elgamal output+ Rabin Signature

(W, R_W)
(G, R_G)
(c, R_c)

Bob

Digital Signature Keys
Private key = (u,t)
Public key = (F,b)
Public key = (J,d)

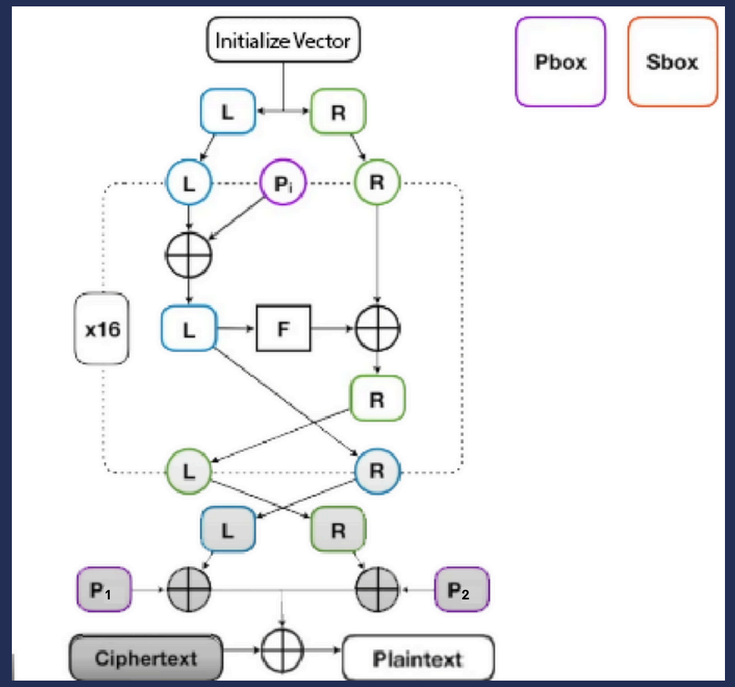
EC Elgamal Keys
Private key = n
Public key = (A,Z,y)
Public key = (B,H,o)

Received from Alice

(W, R_W) ✓
(G, R_G) ✓
(c, R_c) ✓

Phase 10

Bob decrypt **Ciphertext : c**
Using **Blowfish secret key: k** and using Blowfish symmetric algorithm (In OFB mode, The IV is inside the Ciphertext)



This time we start at P18 and go down to P3.
This will get us **Plaintext : m**

Alice

Digital Signature Keys
Private key = (p,q)
Public key = (J,d)
Public key = (F,b)

EC Elgamal Keys
Private key = s
Public key = (B,H,o)
Public key = (A,Z,y)

Plaintext : m

Blowfish
Blowfish secret key: k
Ciphertext : c

EC Elgamal output+ Rabin Signature

(W, R_W)
(G, R_G)
(c, R_c)

Bob

Digital Signature Keys
Private key = (u,t)
Public key = (F,b)
Public key = (J,d)

EC Elgamal Keys
Private key = n
Public key = (A,Z,y)
Public key = (B,H,o)

Received from Alice

(W, R_W) ✓
(G, R_G) ✓
(c, R_c) ✓

Decrypted
Blowfish secret key: k

Digital Signature

Keys

Private key = (p,q)

Public key = (J,d)

Public key = (F,b)

EC Elgamal

Keys

Private key = s

Public key = (B,H,o)

Public key = (A,Z,y)

Plaintext : m

Blowfish

~~Blowfish secret key: k~~

~~Ciphertext : c~~

EC Elgamal output+ Rabin Signature

~~(W, R_W)~~

~~(G, R_G)~~

~~(c, R_c)~~

Alice

Welcome..

Thanks Alice!

Bob

Digital Signature

Keys

Private key = (u,t)

Public key = (F,b)

Public key = (J,d)

EC Elgamal

Keys

Private key = n

Public key = (A,Z,y)

Public key = (B,H,o)

Received from Alice

(W, R_W) ✓

(G, R_G) ✓

(c, R_c) ✓

Decrypted

Blowfish secret key: k ✓

Plaintext : m

Note that Bob can also send a message to Alice, he just need to start from Phase 2, no more public key exchange is needed!


Thank
you!