RSA

- To encrypt message M compute
  - $c = m^e \bmod n$
- To decrypt ciphertext c compute
  - $m = c^d \bmod n$
- Parameters to decide
  - e, d, n

Key Choice

- Let p and q be two large prime numbers
- Let n = pq
- Compute $\phi(n) = \phi(p)\phi(q) = (p - 1)(q - 1)$, where ϕ is Euler's totient function. This value is kept private.

Key Choice

- Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$; i.e., e and $\phi(n)$ are coprime.
  - e is released as the encryption key.
- Determine d as $d \equiv e^{-1} \pmod{\phi(n)}$; or solve for d given $d \cdot e \equiv 1 \pmod{\phi(n)}$
  - d is the decryption key
- (e, d) is the RSA key pair

Example

- Select primes p=11, q=3
- n = p* q = 11*3 = 33
- Compute $\phi(33) = \phi(11)\phi(3) = (11-1)*(3-1)=20$
- Choose e = 3

Example (cont.)

- Compute d such that

$$e * d \mod \phi(n) = 1$$

$$3 * d \mod 20 = 1$$

$$d = 7$$

Public key = (n, e) = (33, 3)

Private key = (d) = (7)

Example (cont.)

- Now say we want to encrypt message m = 5

- $c = m^e \bmod n = 5^3 \bmod 33 = 125 \bmod 33 = 26$
  - Hence the ciphertext c = 26

- To check decryption, we compute

$$m = c^d \bmod n = 26^7 \bmod 33 = 5$$

Lecture 10

Digital Signatures

CS 450/650

Fundamentals of

Integrated Computer Security

A Scenario

- A customer (Bob) made an online order from a seller (Alice) for a clothes which is under “no return” policy
- Bob does not like the clothes and claims that he has never made such order before
- How to address this dispute?

Digital Signatures

- A digital signature can be interpreted as indicating the signer's agreement with the contents of an electronic document
  - Similar to handwritten signatures on physical documents, but in digital format

Digital Signatures

- A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or documents.
- A valid digital signature gives a recipient reason to believe that the message was created by a known sender, that
  - **the sender cannot deny having sent the message, and**
  - **the message was not altered in transit.**

Digital Signature Properties

- **Unforgeable**: Only the signer can produce his/her signature
- **Non-Alterable**: A signed document cannot be altered without invalidating the signature
- **Non-Reusable**: A signature from one document cannot be moved to another document

Digital Signature Properties

- Signatures can be validated by other users
  - the signer cannot reasonably claim that he/she did not sign a document bearing his/her signature

Digital Signature Using RSA

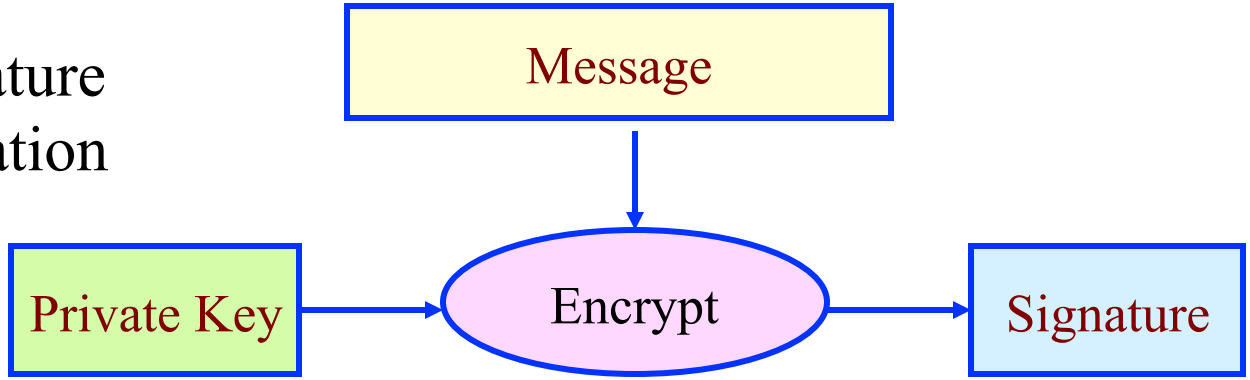
- The RSA public-key cryptosystem can be used to create a digital signature for a message m
  - Asymmetric Cryptographic techniques are well suited for creating digital signatures

- RSA cryptosystem
  - $c = M^e \bmod n$
  - $M = c^d \bmod n$

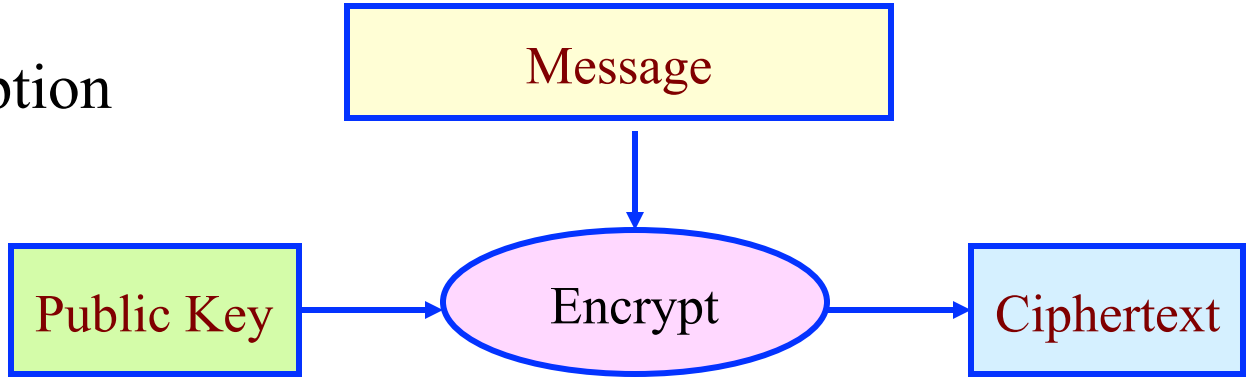
Digital Signature Entities

Signature Generation (Signer)

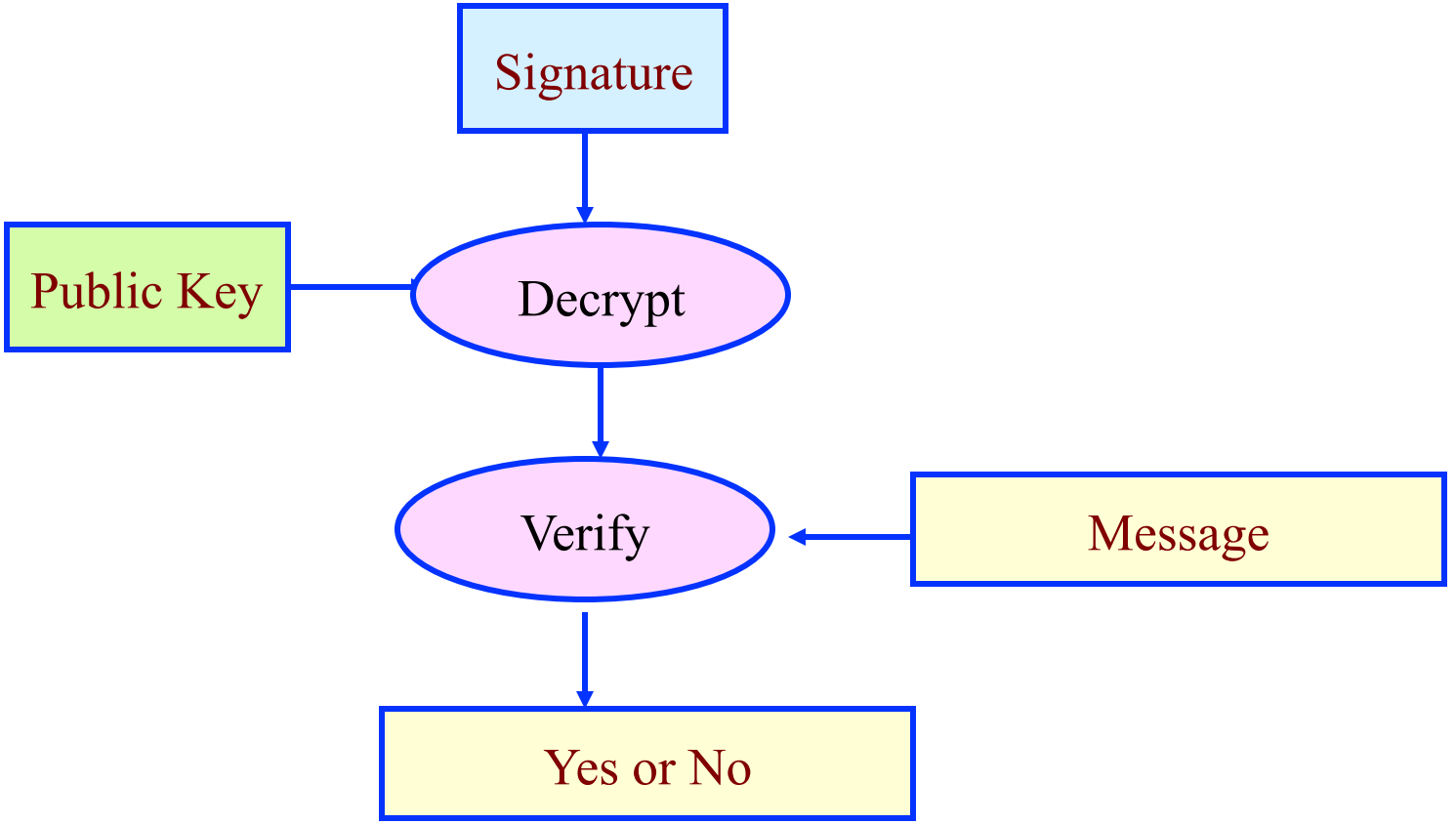
•Signature generation



•Data encryption



Signature Verification



Example

- Generate signature S
  - e = 53 (encryption key, or signing key, private!)
  - d = 413 (decryption key, public!)
  - n = 629
  - m = 250 (message)

- $S = (m)^e \bmod n$
  - $S = 250^{53} \bmod 629 = 411$

Example

- Verify signature with message recovery
  - Public key (d) = 413
  - n = 629
  - S = 411
- $m' = S^d \bmod n$
  - $m' = 411^{413} \bmod 629 = 250$
- Verifier checks if m'=m
  - If yes, then S is a valid signature over m

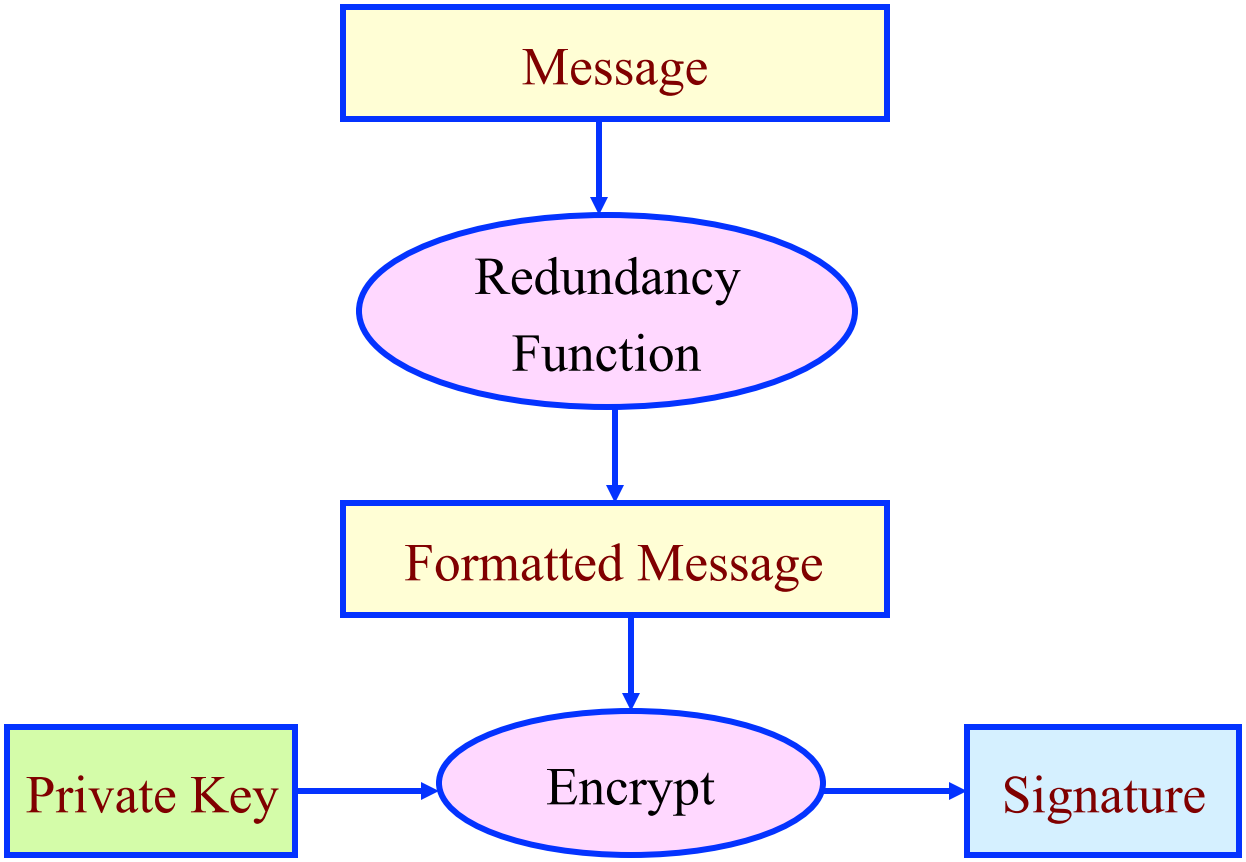
Creating a forged signature

- Choose a random number between 0 and n-1 for S (n=629)
  - S = 323
- Use the signer's public key to decrypt S
  - $x = S^d \bmod n = 323^{413} \bmod 629 = 85$
  - A valid signature (323) has been created for a random message (85)!!
  - unforgeable has been compromised
    - is it conflict with the RSA security claim?

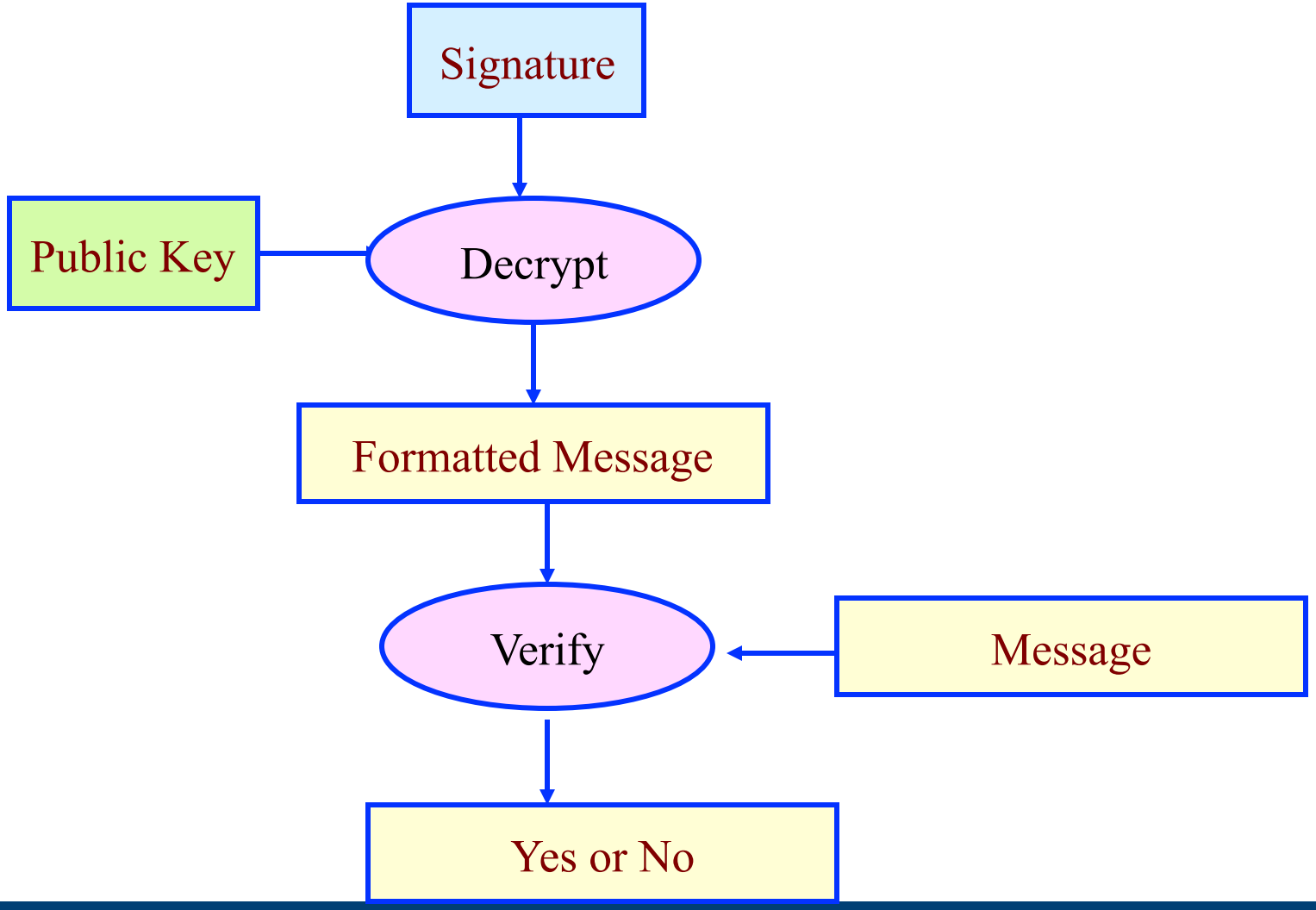
Redundancy Function

- Solution: redundancy function

Signature Generation (Signer)



Signature Verification



Example

- generate signature S
  - e = 53
  - d = 413
  - n = 629
  - m = 7
  - Assume that R(X) = XX
- $S = R(m)^e \bmod n$
  - $S = 77^{53} \bmod 629 = 25$

Example

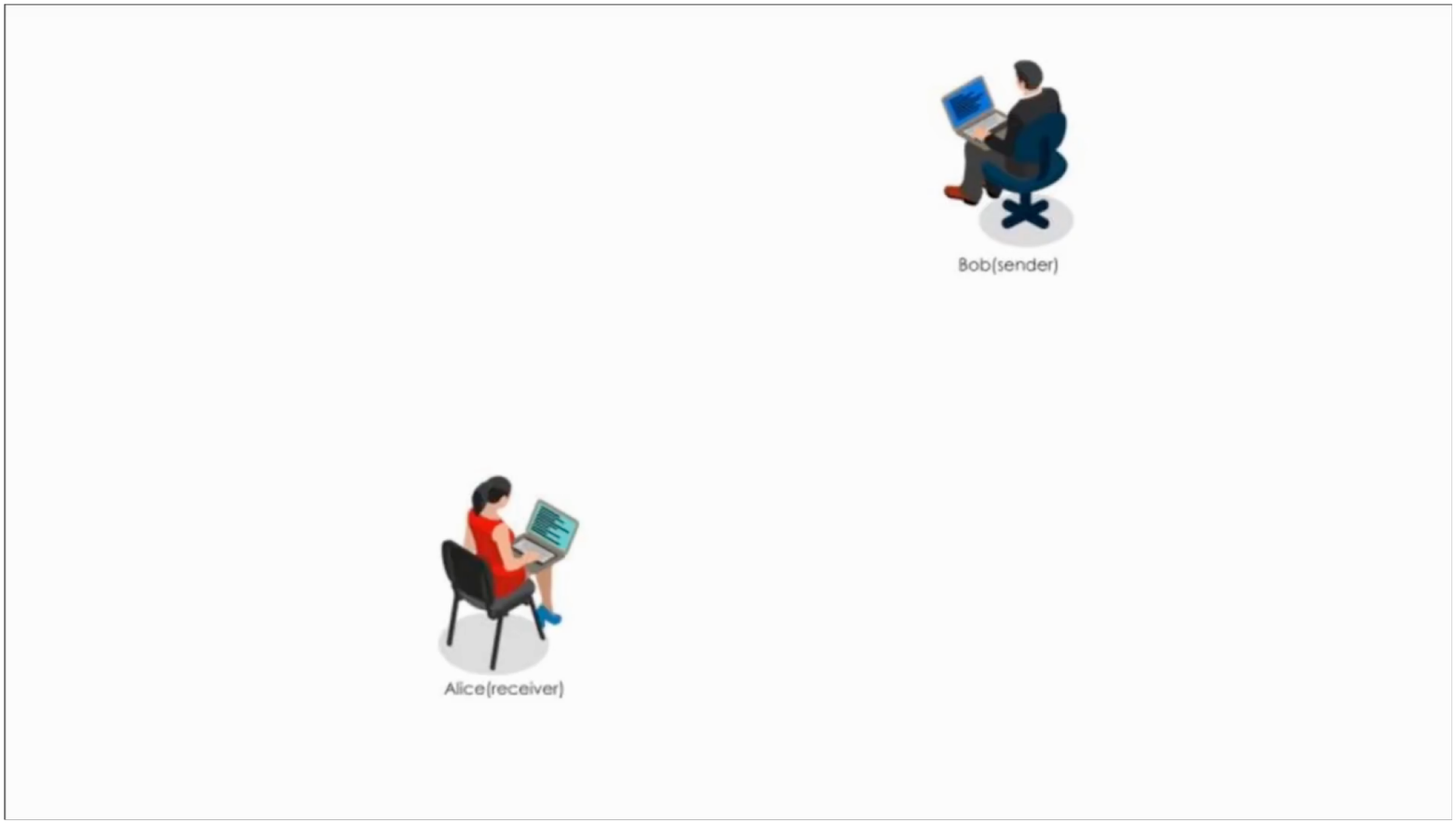
- verify signature with message recovery
  - Public key (d) = 413
  - n = 629
  - S = 25
- $R(m) = S^d \bmod n$
  - $R(m) = 25^{413} \bmod 629 = 77$
- The verifier then checks that R(m) is of the form XX for some message X
  - $m = R^{-1}(m) = 7$

Forging signature (revisited)

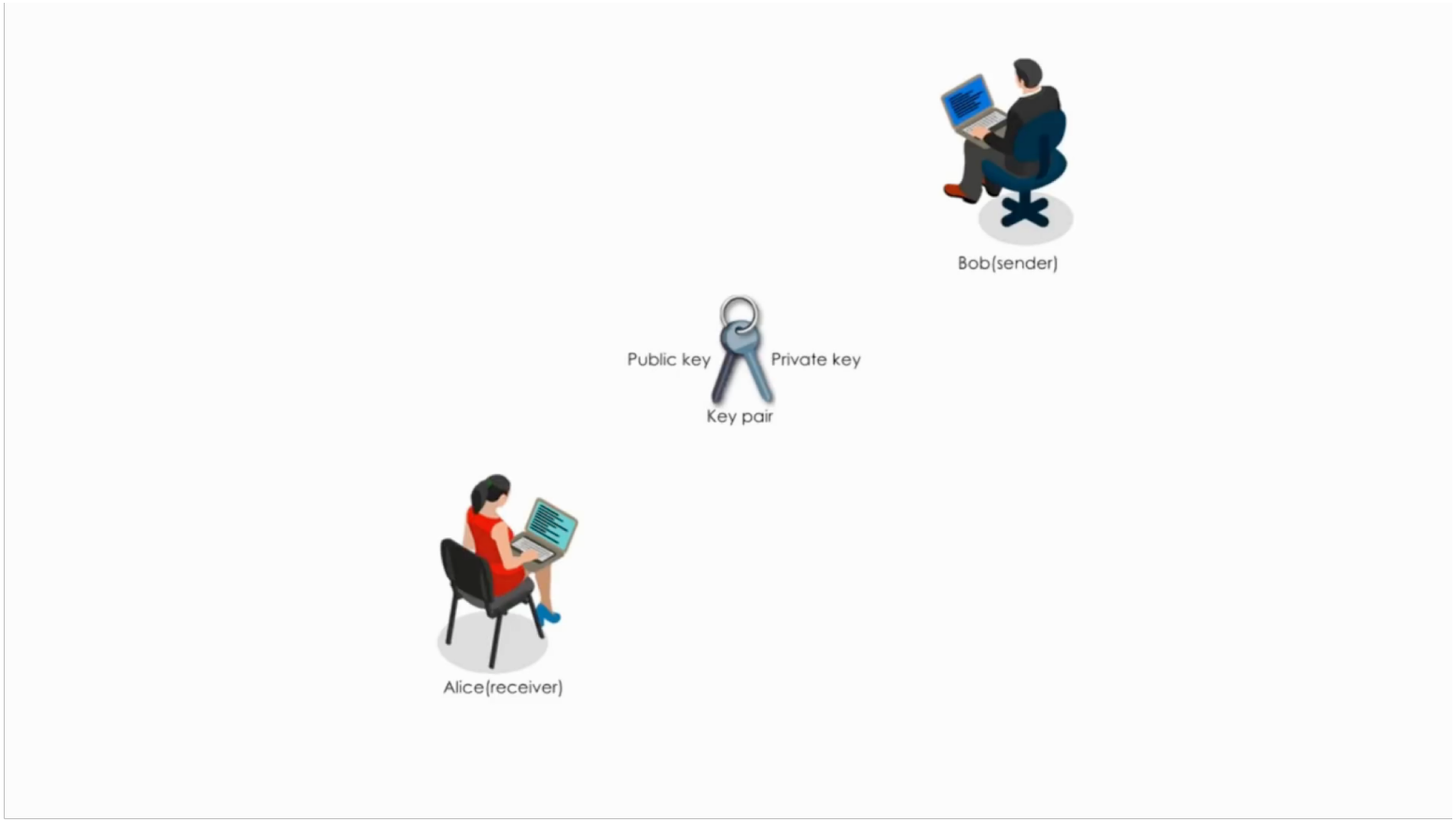
- Choose a random number between 0 and n-1 for S
  - S = 323
- Use the signer's public key to decrypt S
  - $R(m) = 323^{413} \bmod 629 = 85$
- However, 85 is not a legal value for R(m)
  - so S = 323 is not a valid signature

- Can we use symmetric crypto system for digital signature?

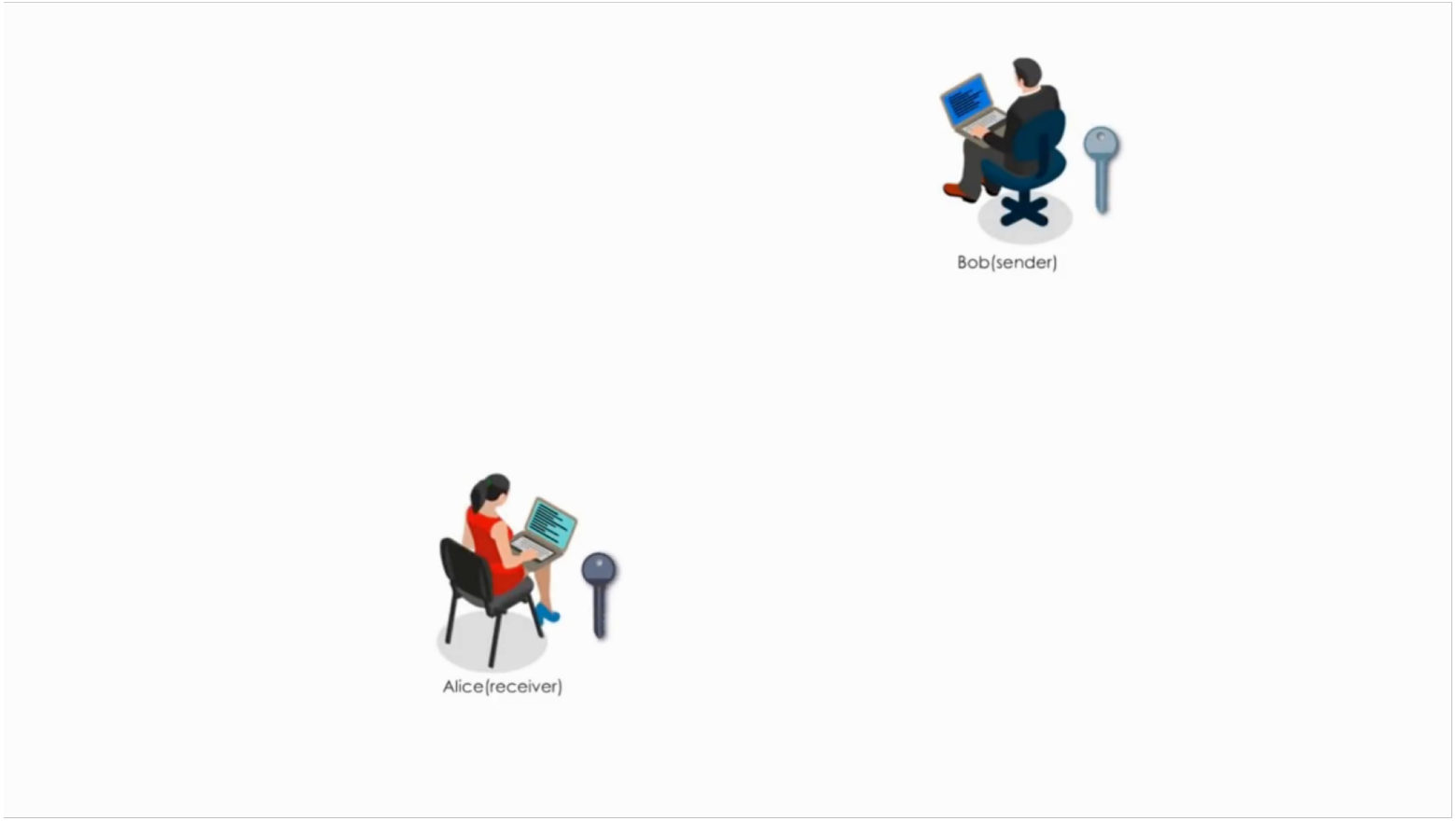
Simple Scenario of Digital Signature



- **Bob wants to send a memo to Alice with his DS**

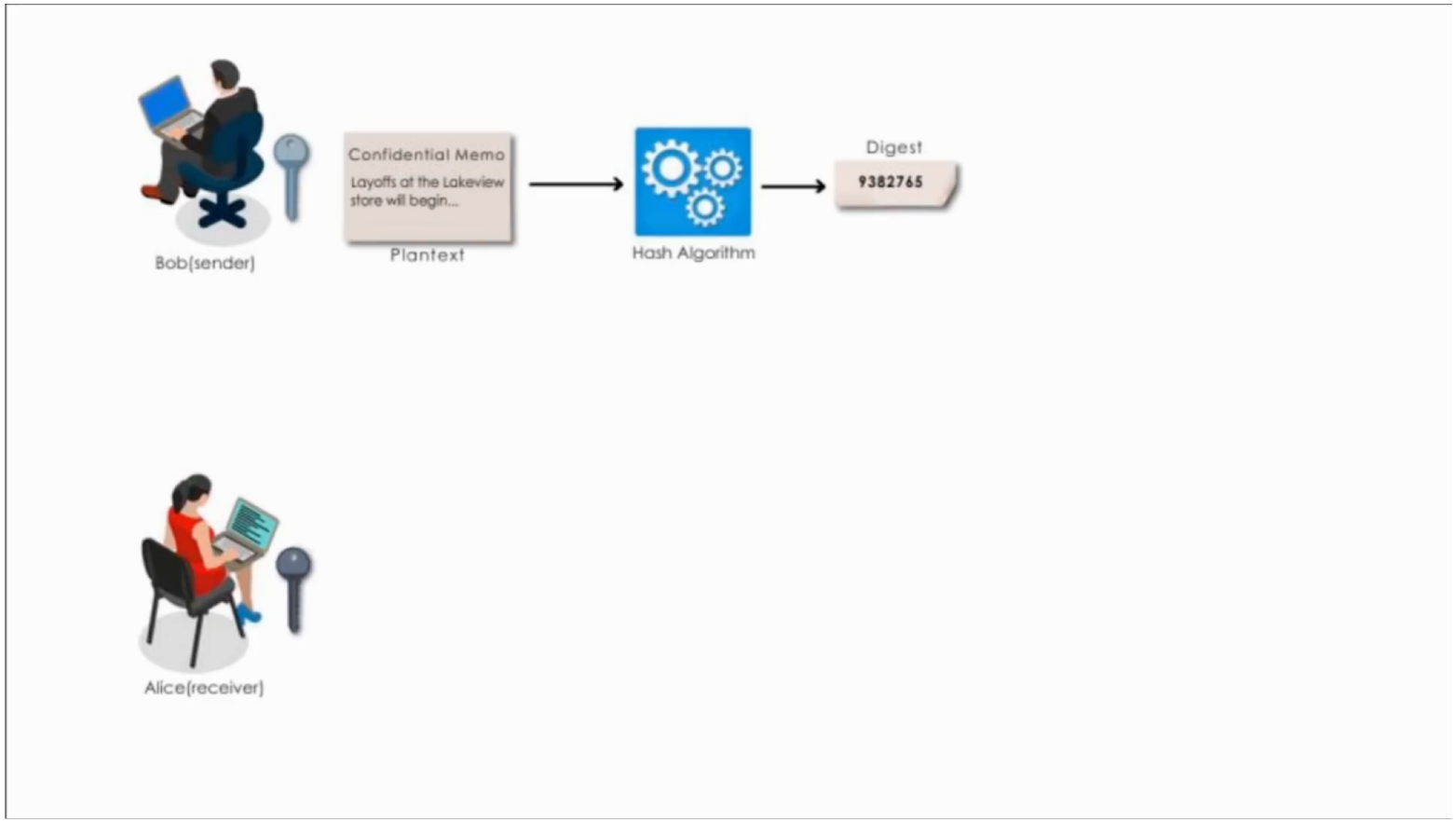


- **First, Bob generates Public and Private keys**

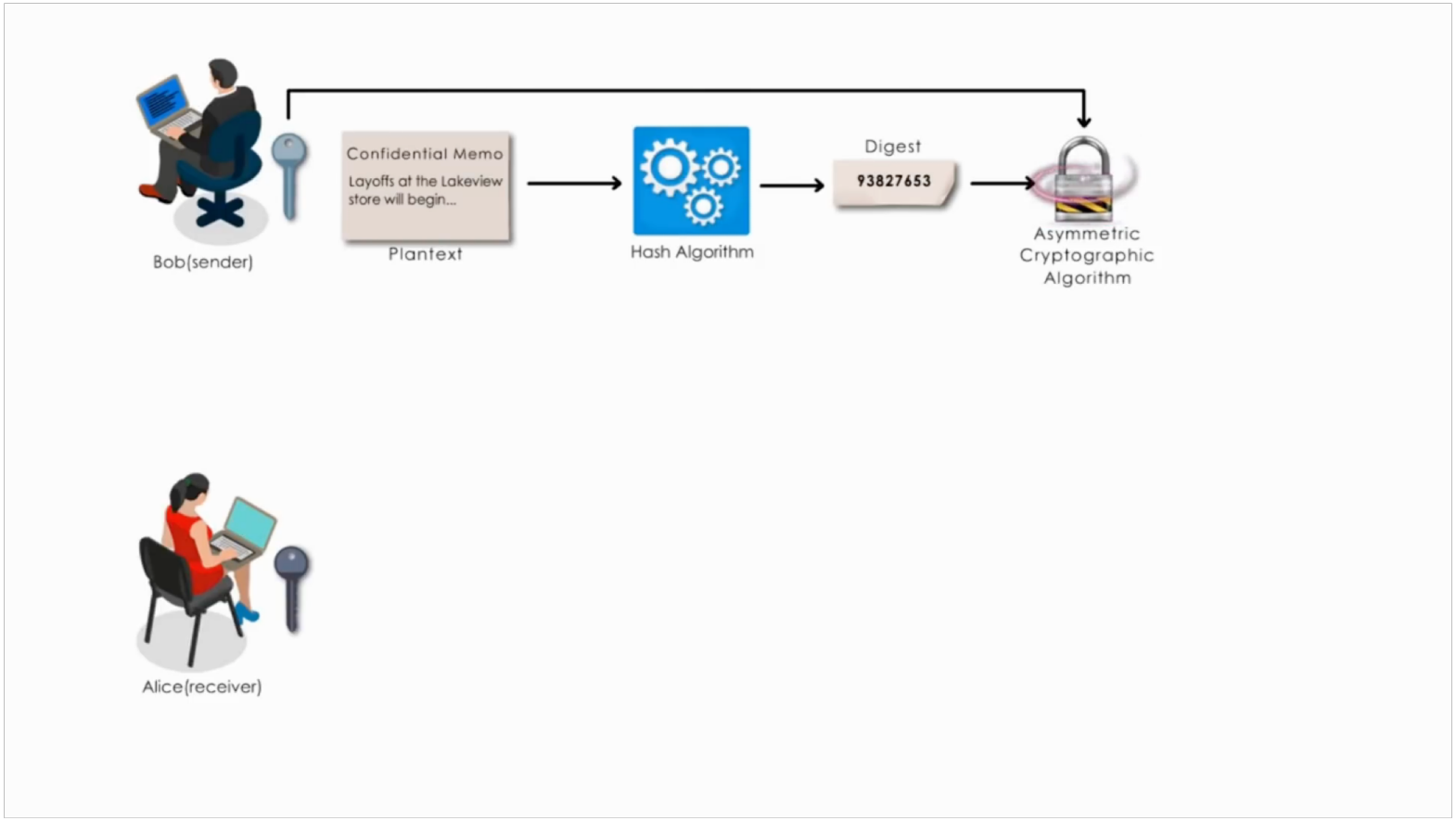


- **Bob keeps his private key and sends his public key to Alice**

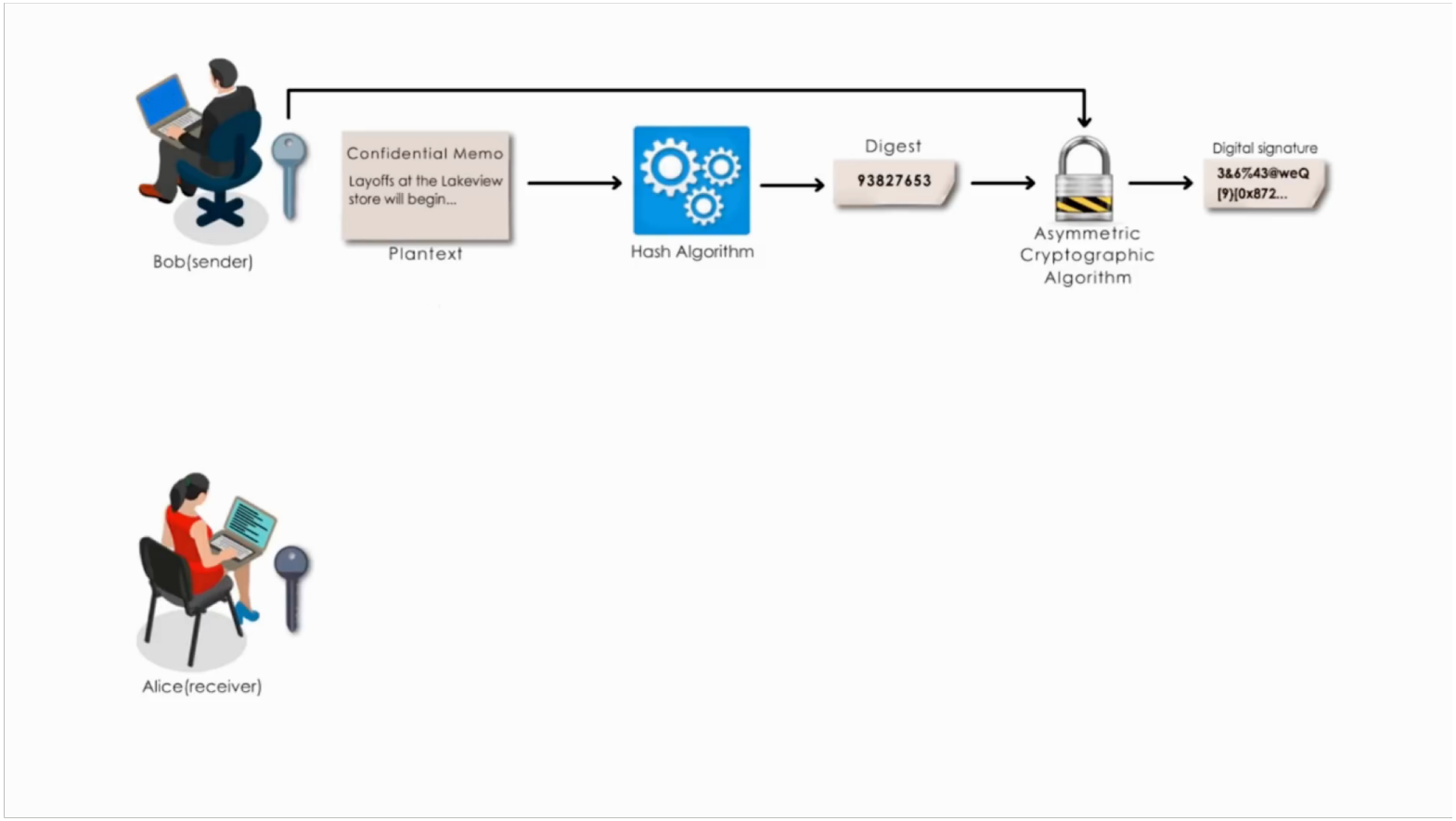
- **Then, Bob creates his memo which he wishes to send to Alice**



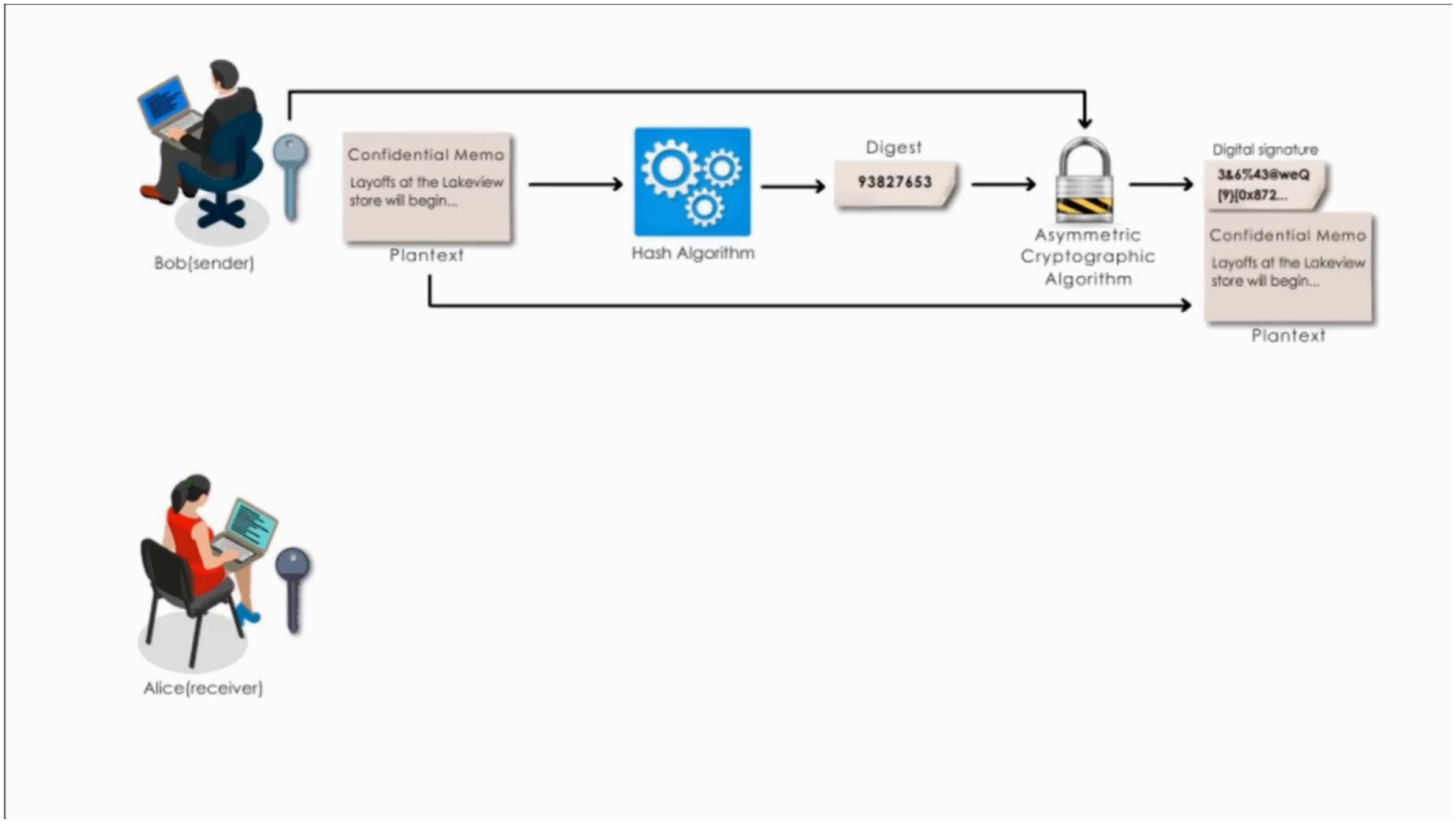
- **After creating the memo, Bob generates a ‘digest’ by using a hash algorithm**



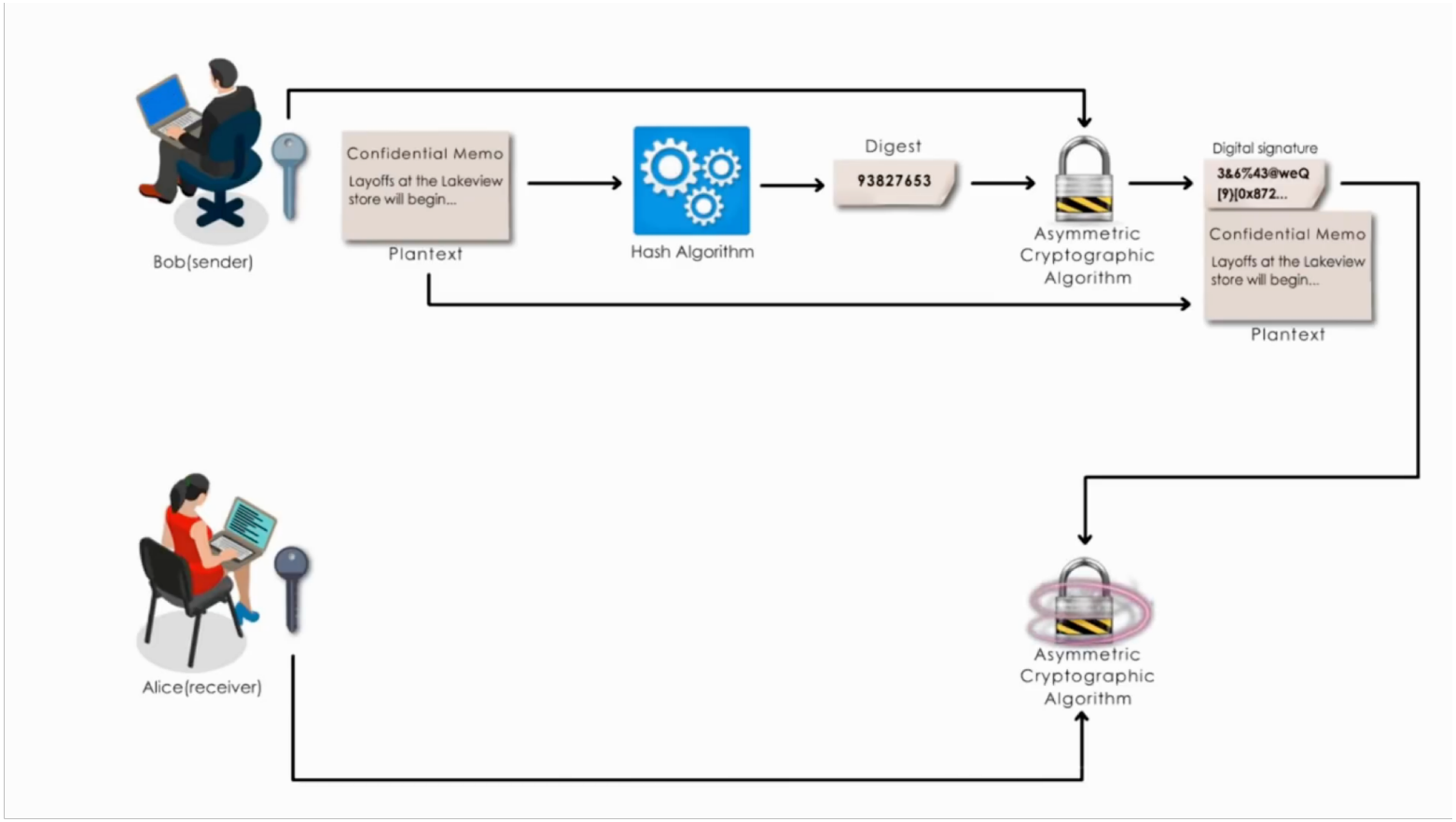
- **After generating the digest, Bob uses his private key to encrypt the digest**



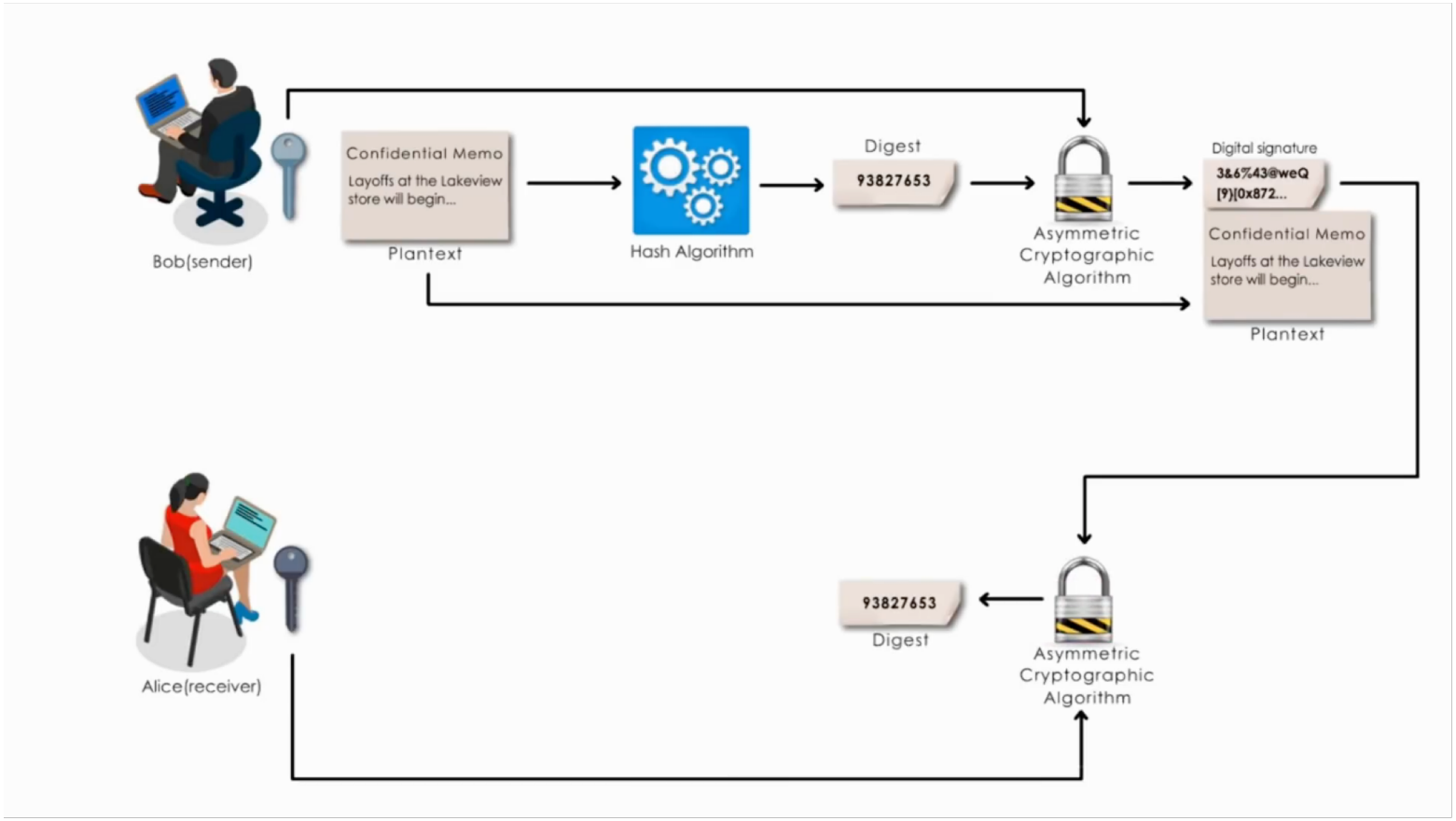
- **The encrypted digest is called the ‘Digital Signature’ for the memo**



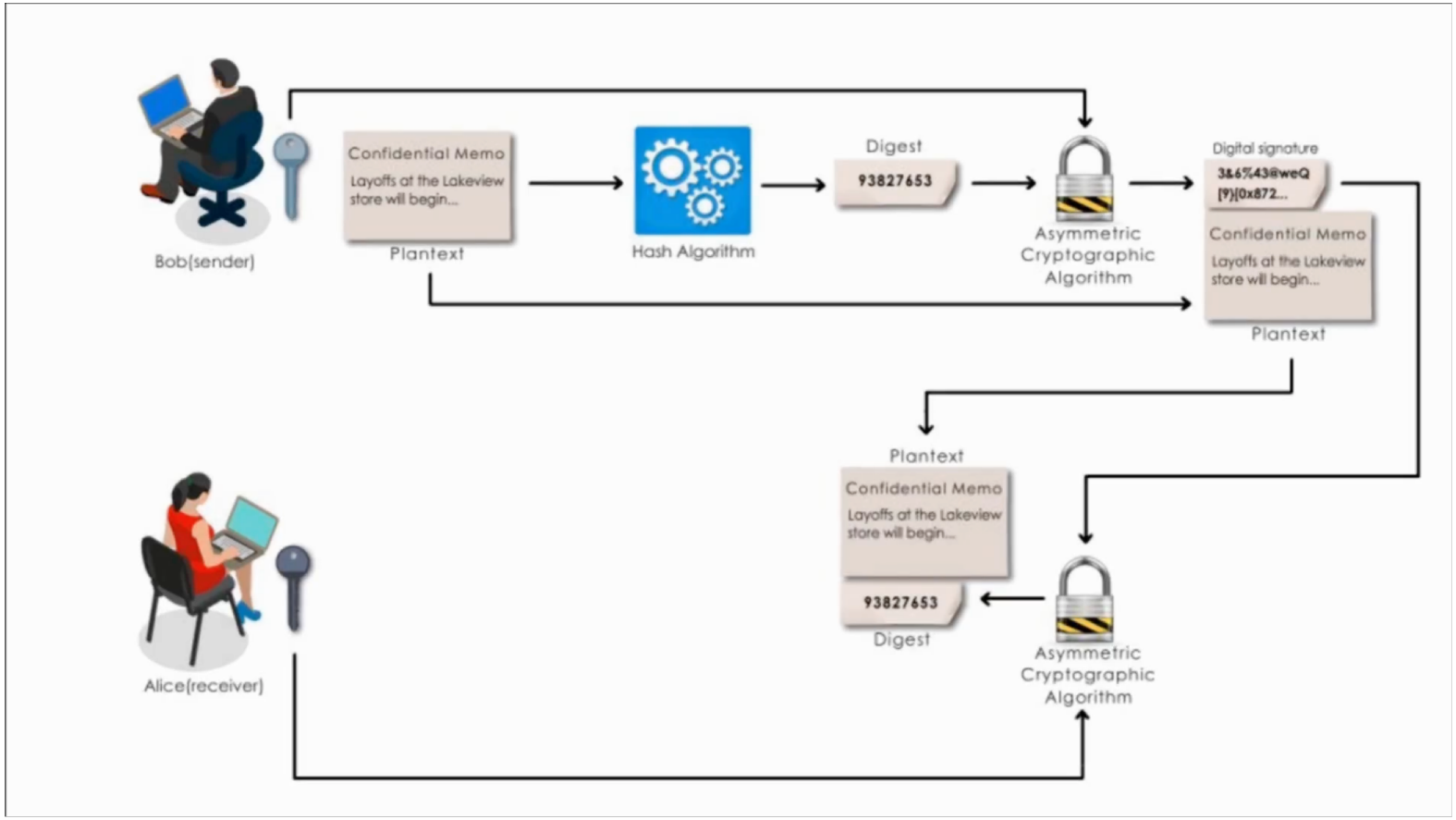
- **Bob sends both the memo and the DS to Alice**
- **In this example, the memo is not encrypted**



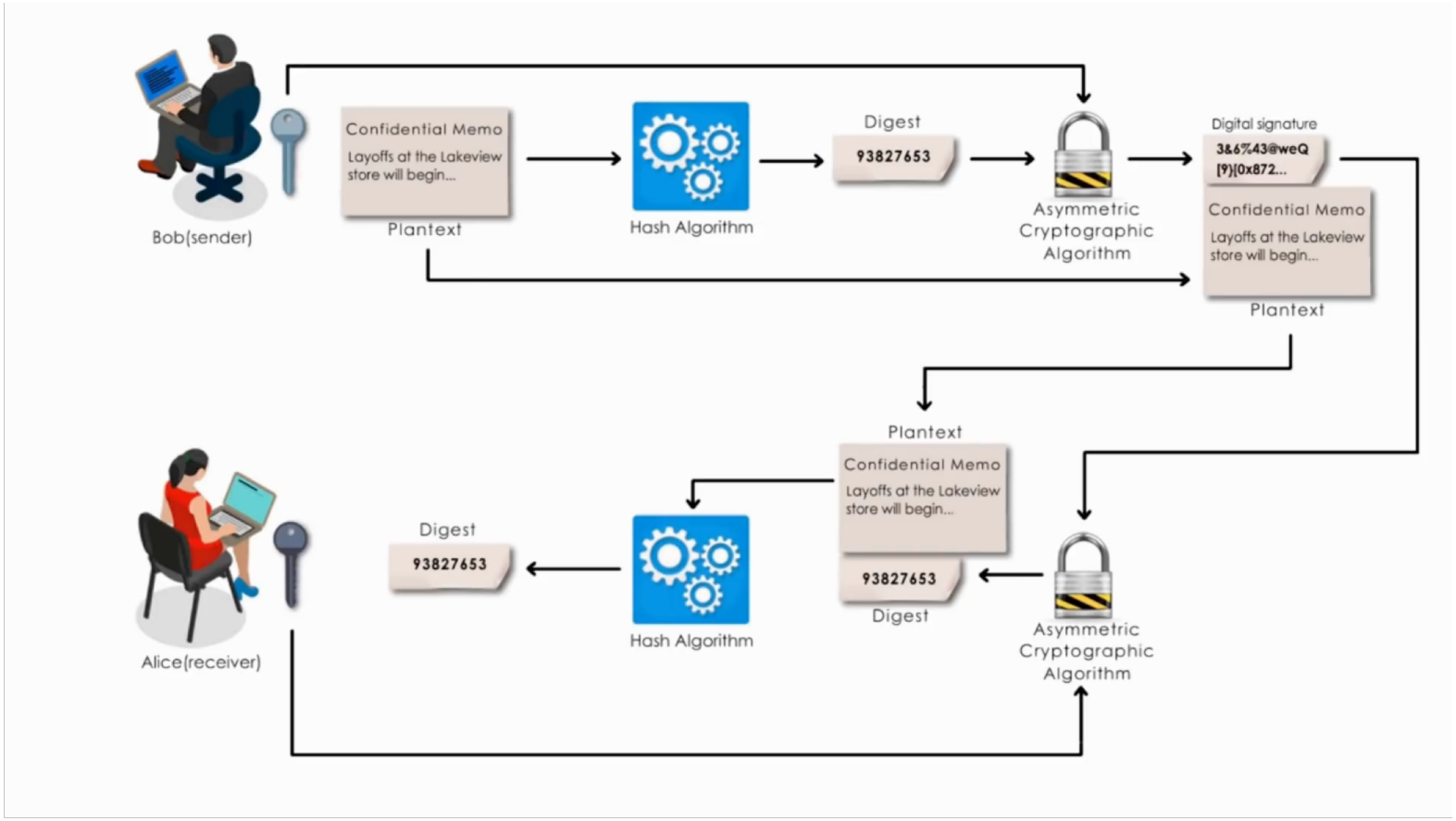
- **Alice decrypts the DS using Bob's public key**



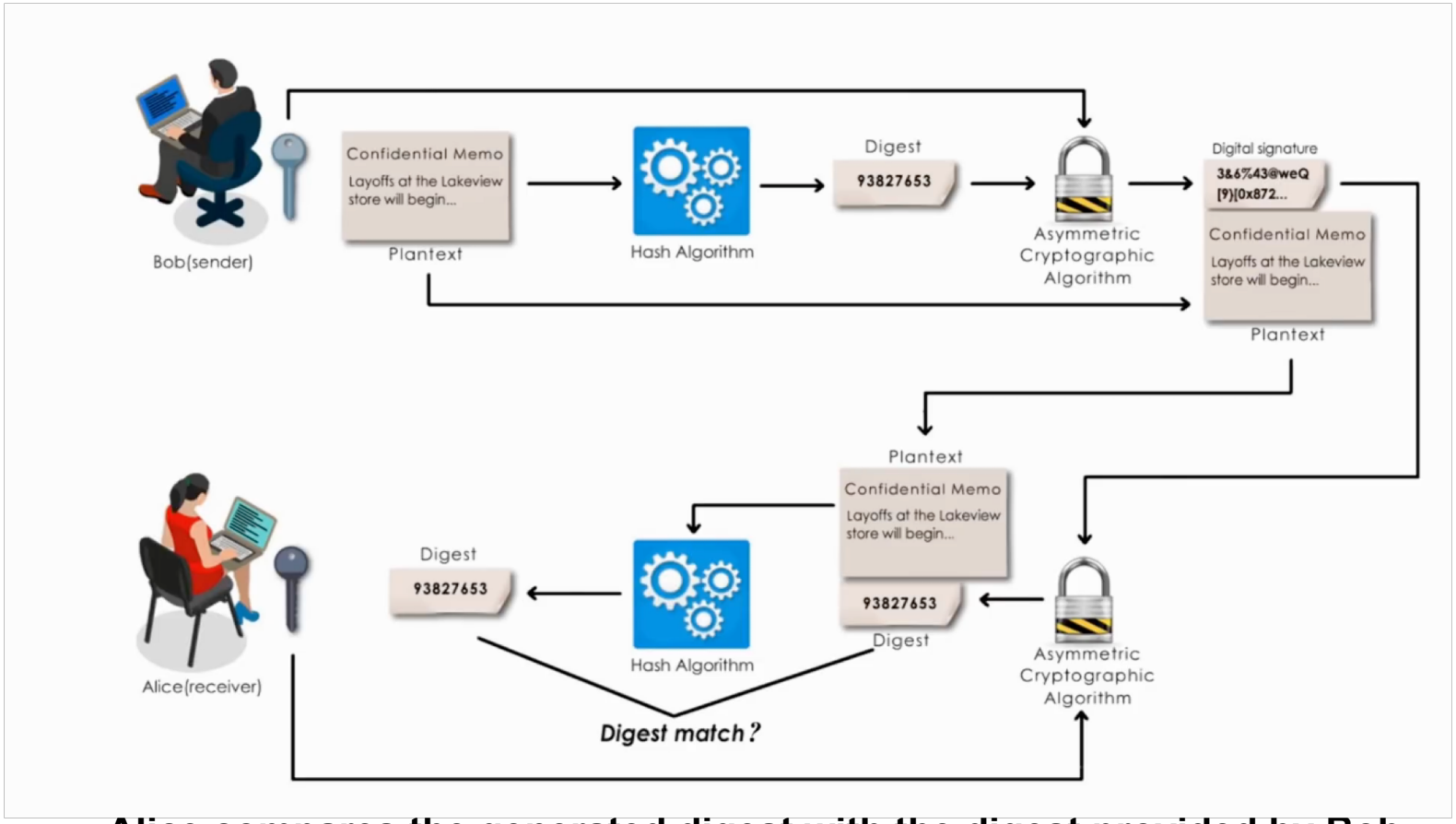
- **After decrypting, Alice gets the digest for the memo**



- **After finding the digest, Alice will check the integrity of the memo**



- **Alice uses the same hash algorithm Bob uses, in order to find the digest for the memo**



- Alice compares the generated digest with the digest provided by Bob
- If they are equal, Alice verifies two things
  - The memo has not been changed during transit
  - The memo was sent by Bob
- If the are not equal, Alice knows that the message was changed