University of Kotli Azad Jammu & Kashmir
Faculty of Computing & Engineering
Department Software Engineering

Assignment: 02
Group: 03

Course: Information Security
Topic: Asymmetric cryptography

Submitted By:

Name	**Roll no**
Qura_Tul_Ain Saleem	04
Bilal Arif	08
Usman Arif	13
Hammad Haleem	15

Semester: 6th
Submitted to: Ms. Dania
Submission Date: 28-10-2024

Table of Contents

List of Figures

1. Asymmetric cryptography:

Asymmetric cryptography, also known as public key cryptography, is a process that uses a pair of related keys one public key and one private key to encrypt and decrypt a message and protect it from unauthorized access or use.

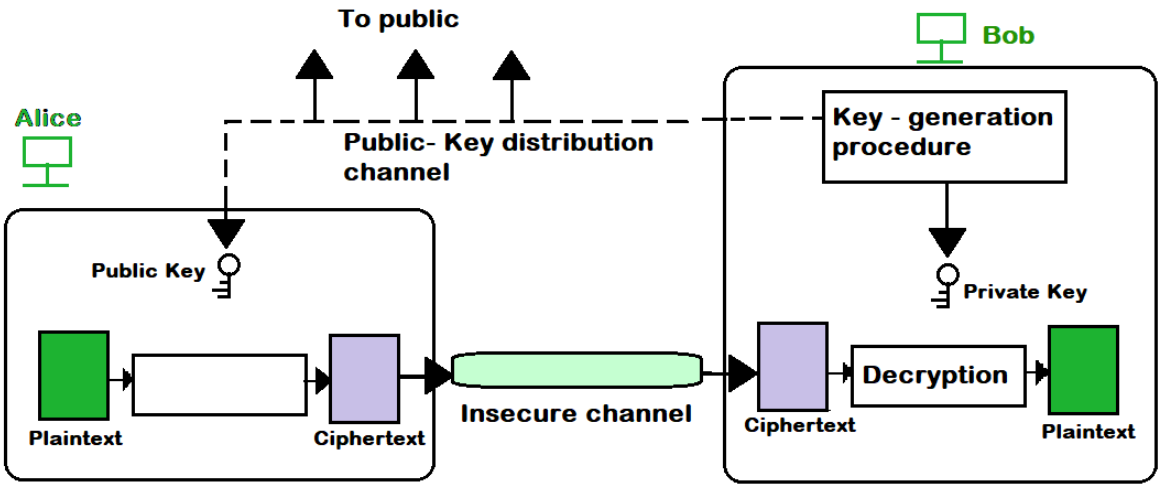

Figure 1

2. History of asymmetric cryptography

Whitfield Diffie and Martin Hellman, researchers at Stanford University, first publicly proposed asymmetric encryption in their 1977 paper, "New Directions in Cryptography."

The concept was independently and covertly proposed by James Ellis several years earlier, while he was working for the Government Communications Headquarters, the British intelligence and security organization. The asymmetric algorithm as outlined in the Diffie-Hellman paper uses numbers raised to specific powers to produce decryption keys. Diffie and Hellman initially teamed up in 1974 to solve the problem of key distribution.

3. How does asymmetric cryptography work?

Asymmetric encryption uses a mathematically related pair of keys for encryption and decryption: a public key and a private key. If the public key is used for encryption, then the related private key is used for decryption. If the private key is used for encryption, then the related public key is used for decryption.

The two participants in the asymmetric encryption workflow are the sender and the receiver. Each has its own pair of public and private keys. First, the sender obtains the receiver's public key. Next, the plaintext message is encrypted by the sender using the receiver's public key. This creates cipher text. The cipher text is sent to the receiver, who decrypts it with their private key, returning it to legible plaintext.

Because of the one-way nature of the encryption function, one sender is unable to read the messages of another sender, even though each has the public key of the receiver.

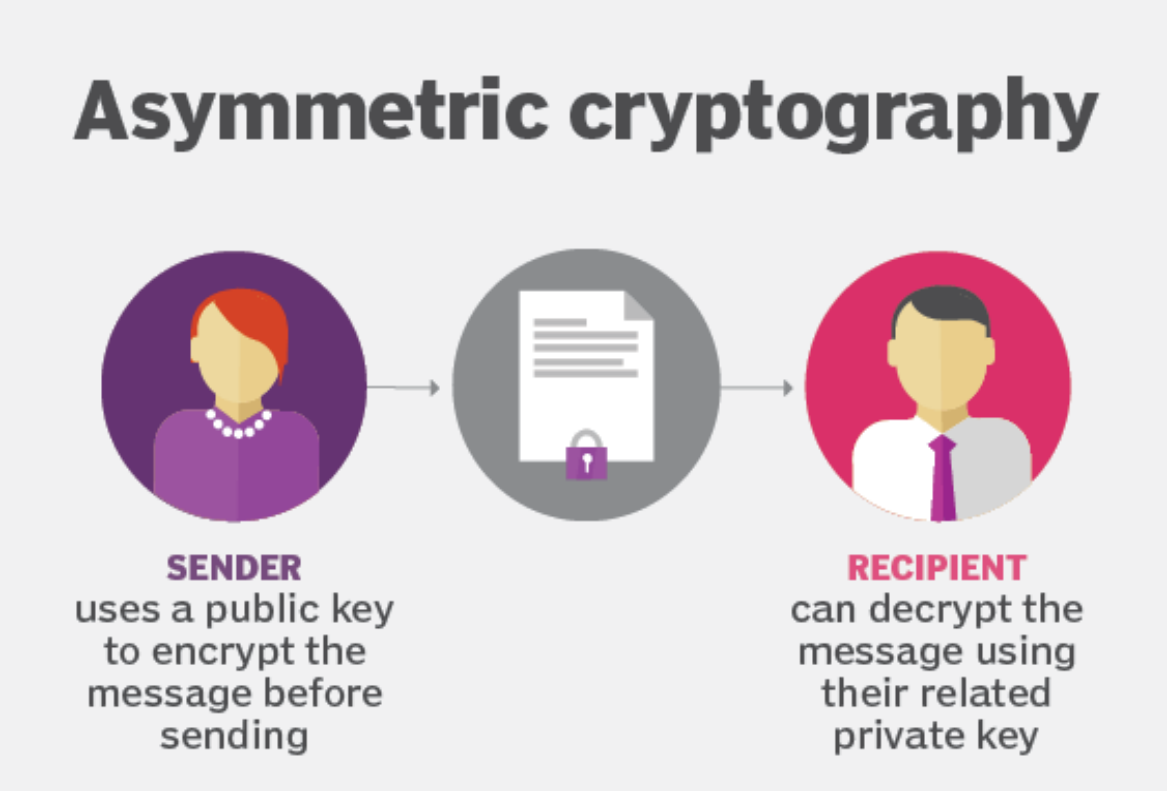

Figure 2

4. Uses of asymmetric cryptography:

Asymmetric cryptography is typically used to authenticate data using digital signatures. A digital signature is a mathematical technique that validates the authenticity and integrity of a message, software or digital document. It is the digital equivalent of a handwritten signature or stamped seal.

Based on asymmetric cryptography, digital signatures can provide assurances of evidence to the origin, identity and status of an electronic document, transaction or message, as well as acknowledge informed consent by the signer.

Asymmetric cryptography can also be applied to systems in which many users might need to encrypt and decrypt messages, including the following:

- **Encrypted email:** A public key can encrypt an email message, and a private key can decrypt it.
- **SSL/TLS:** Establishing encrypted links between websites and browsers also uses asymmetric encryption.

- **Cryptocurrencies:** Bitcoin and the other crypto currencies rely on asymmetric cryptography. Users have public keys that everyone can see and private keys that are kept secret. Bitcoin uses a cryptographic algorithm to ensure only legitimate owners can spend the funds.

In the case of the Bitcoin ledger, each unspent transaction output (UTXO) is typically associated with a public key. For example, if user X, who has an UTXO associated with their public key, wants to send the money to user Y, user X uses their private key to sign a transaction that spends the UTXO and creates a new UTXO that's associated with user Y's public key.

5. Benefits of asymmetric cryptography:

The benefits of asymmetric cryptography include the following:

- The key distribution problem is eliminated because there's no need for exchanging keys.
- Security is increased since the private keys don't ever have to be transmitted or revealed to anyone.
- The use of digital signatures is enabled so a recipient can verify that a message comes from a particular sender.
- It allows for nonrepudiation so the sender can't deny sending a message.

6. Disadvantages of asymmetric cryptography:

Disadvantages of asymmetric cryptography include the following:

- It's a slow process compared to symmetric cryptography. It's, therefore, not appropriate for decrypting bulk messages.
- If an individual loses their private key, they can't decrypt the messages they receive.
- Because public keys aren't authenticated, no one can ensure a public key belongs to the person specified. Consequently, users must verify their public keys belong to them.
- If a malicious actor identifies a person's private key, the attacker can read that individual's messages.

7. Difference between asymmetric & symmetric cryptography:

The main difference between asymmetric versus symmetric cryptography is that asymmetric encryption algorithms make use of two different but related keys. One key encrypts data and another key decrypts it. Symmetric encryption uses the same key to perform both encryption and decryption functions.

Another difference between asymmetric and symmetric encryption is the length of the keys. In symmetric cryptography, the length of the keys -- which is randomly selected is typically set at 128 bits or 256 bits, depending on the level of security needed.

In asymmetric encryption, there must be a mathematical relationship between the public and private keys. Since malicious actors can potentially exploit this pattern to crack the encryption, asymmetric keys need to be longer to offer the same level of security. The difference in the length of the keys is so pronounced that a 2,048-bit asymmetric key and a 128-bit symmetric key provide about an equivalent level of security.

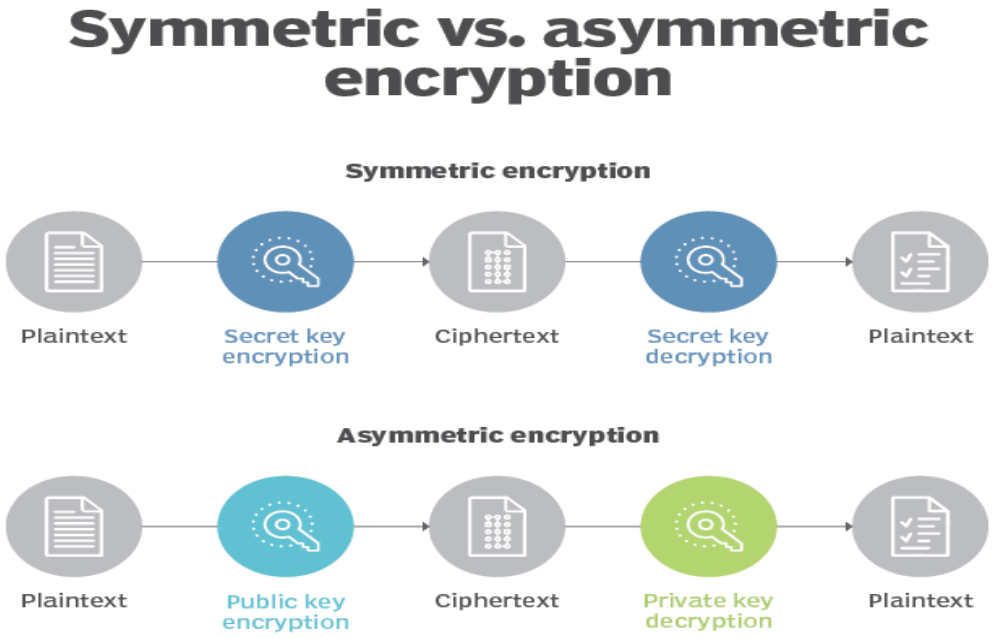

Figure 3

8. Limitations of asymmetric cryptography:

Some limitation of asymmetric cryptography:

8.1. Performance:

Asymmetric encryption is generally slower than symmetric encryption, making it less suitable for encrypting large amounts of data directly. Often, asymmetric encryption is used to exchange a symmetric key, which is then used for data encryption.

8.2. Key Management:

The security of the system depends on the protection of the private key. If it is compromised, the security of all communications associated with that key is also compromised.

9. Common Algorithms of asymmetric cryptography:

There are several algorithms used in asymmetric key cryptography, some of them are as follows:

- RSA (Rivest–Shamir–Adleman)
- Elliptic Curve Cryptography (ECC)
- Diffie-Hellman
- DSS (Digital Signature Standard)

9.1. RSA (Rivest–Shamir–Adleman)

It is commonly utilized to ensure secure communication and for creating digital signatures. It Uses large integer prime numbers for key generation. It Encrypts data with the public key and decrypts with the private key. It is Slower than some other algorithms but offers strong security.

Key Generation

- Choose/Select two large prime numbers, p and q.
- Calculate n=p∗q $n=p*q$
- Calculate φ(n) = (p-1)(q-1), where φ is Euler's totient function.
- Choose an integer e, that 1<e<φ(n)1<e<φ(n) and gcd(e,φ(n))=1gcd(e,φ(n))=1.
- Compute d, the modular multiplicative inverse of e modulo φ(n).
- Public key: (e,n)(e,n)
- Private key: (d,n)(d,n)

Encryption

- Convert plaintext message into an integer m.
- Compute ciphertext c=memodn $c=m^e mod n$.

Decryption

- Calculate the plaintext message

9.2. Elliptic Curve Cryptography (ECC)

It gives equal protection to RSA with shorter key sizes. The concept behind this is based on the mathematical properties of elliptic curves. It is Faster and more efficient for resource-constrained devices. It gaining popularity in mobile security and the Internet of Things (IoT).

Key Generation

- Select an elliptic curve over a finite field.
- Choose a base point on the curve and a large prime order.

- Select a private key, which is a random integer k.
- Now, using the elliptic curve scalar multiplication, you need to find the public key by multiplying the base point by the private key.

Encryption and Decryption

- ECC is primarily used for key exchange, such as in the Elliptic Curve Diffie-Hellman (ECDH) algorithm, rather than directly for encryption/decryption.

9.3. Diffie-Hellman Key Exchange

It doesn't directly encrypt data but establishes a shared secret key for secure communication. Two parties can generate a common secret key without ever exchanging it directly. It often used in conjunction with other algorithms like RSA for key exchange.

Key Exchange

- Sender and receiver agree on a large prime number p and a primitive root g modulo p.
- Each party selects a secret key: a and b.
- Party A computes public key A=gamodp $A=ga \bmod p$ and sends it to Party B.
- Party B computes public key B=gbmodp $B=gb \bmod p$ and sends it to Party A.
- Both parties compute the shared secret: s=Abmod

Security

- Diffie-Hellman does not provide proper authentication; it sets a shared secret between two parties.

9.4. Digital Signature Standard (DSS)

It uses a variant of the ElGamal encryption scheme. It is primarily for digital signatures, ensuring message authenticity and integrity. The sender signs a message with their private key, receiver verifies with the sender's public key. It is often used for secure emails and software signing.

Key Generation

- DSS uses the Digital Signature Algorithm (DSA), which relies on discrete logarithm problems in a finite field.
- Generate a prime number p and a prime divisor q of p-1.
- Choose a generator g such that gqmod p=1
- Generate private key x, a random integer between 1 and q-1.
- Calculate the public key y=gxmodp $y=gx \bmod p$.

Signing

- Calculate a hash of the message.

- Generate any random number k such that it lies between 1 and q-1.
- Calculate r=(gkmodp)
- Calculate s = k-1 * (hash + x * r) mod q.
- The signature is the pair (r, s).

Verification

- Recalculate the hash of the message.
- Compute w = s-1 mod q.
- Compute u1=(hash∗w)modq$u1=(hash*w)modq$ and u2=(r∗w)modq$u2=(r*w)modq$.
- Now, we need to calculate v as v = ((gu1 * yu2) mod p) mod q.
- If v is equal to r, it means that the signature is verified.

10. Use Cases of asymmetric cryptography:

Some use cases of asymmetric cryptography:

- **Secure Communications**: Used in SSL/TLS for secure web browsing.
- **Email Security**: Protocols like PGP (Pretty Good Privacy) use asymmetric cryptography for secure email communication.
- **Blockchain and Cryptocurrencies**: Cryptographic techniques used in these technologies often rely on asymmetric key pairs for secure transactions.

11. References:

1. https://www.geeksforgeeks.org/asymmetric-key-cryptography/?re
2. https://www.researchgate.net/profile/Mohammed-Al-f=gcseShabi/publication/332176079_A_Survey_on_Symmetric_and_Asymmetric_Cryptography_Algorithms_in_information_Security/links/5d6b8f4da6fdcc547d70434a/A-Survey-on-Symmetric-and-Asymmetric-Cryptography-Algorithms-in-information-Security.pdf
3. https://www.techtarget.com/searchsecurity/definition/asymmetric-cryptography
4. https://utimaco.com/service/knowledge-base/keys-secrets-management/what-asymmetric-cryptography