3.2 Digital Signature Scheme

1. ElGamal encryption is a public-key cryptosystem. It uses asymmetric key encryption for communicating between two parties and encrypting the message. This cryptosystem is based on the difficulty of finding discrete logarithm in a cyclic group that is even if we know ga and gk, it is extremely difficult to compute gak.

Idea of ElGamal cryptosystem:

EXAMPLE:

Original Message : encryption g used:

58606969545224177071889523715479440353333315907890 gas used:
4711309755639364289552454834506215144653958055252 g^k used :
12475188089503227615789015740709091911412567126782 g^ak used :
39448787632167136161153337226654906357756740068295

Decrypted Message : encryption

In this cryptosystem, the original message M is masked by multiplying gak to it. To remove the mask, a clue is given in form of gk. Unless someone knows a, he will not be able to retrieve M. This is because finding discrete log in a cyclic group is difficult and simplifying knowing ga and gk is not good enough to compute gak.

Advantages:

● Security: ElGamal is based on the discrete logarithm problem, which is considered to be a hard problem to solve. This makes it secure against attacks from hackers.

● Key distribution: The encryption and decryption keys are different, making it easier to distribute keys securely. This allows for secure communication between multiple parties.

● Digital signatures: ElGamal can also be used for digital signatures, which allows for secure authentication of messages.

Disadvantages:

● Slow processing: ElGamal is slower compared to other encryption algorithms, especially when used with long keys. This can make it impractical for certain applications that require fast processing speeds.

- Key size: ElGamal requires larger key sizes to achieve the same level of security as other algorithms. This can make it more difficult to use in some applications.
- Vulnerability to certain attacks: ElGamal is vulnerable to attacks based on the discrete logarithm problem, such as the index calculus algorithm. This can reduce the security of the algorithm in certain situations.

2.Schnorr Digital Signature:

In cryptography, a Schnorr signature is a digital signature produced by the Schnorr signature algorithm that was described by Claus Schnorr. It is a digital signature scheme known for its simplicity, is efficient and generates short signatures. It is one of the protocols used to implement “Proof Of Knowledge”. In cryptography, a proof of knowledge is an interactive proof in which the prover succeeds in ‘convincing’ a verifier that the prover knows something ‘X’. For a machine to know ‘X’ is defined in terms of computation. A machine knows ‘X’ if this ‘X’ can be computed. The Verifier either accepts or rejects the proof. The signature proof is supposed to convince the Verifier that they are communicating with a user who knows the private key corresponding to the public key. In other words, the Verifier should be convinced that they are communicating with the Prover without knowing the private key. Schnorr Digital Signature to implement Zero Knowledge Proof : Let’s take an example of two friends Sachin and Sanchita. Sanchita has announced to the world that she has a public key and can accept and receive information through it. Sachin thinks that Sanchita is lying. Sanchita wants to prove her honesty without showing her private keys. Here is where Schnorr’s protocol will help us.

3.RSA algorithm is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes that the Public Key is given to everyone and the Private key is kept private.

An example of asymmetric cryptography:

1.A client (for example browser) sends its public key to the server and requests some data.

2.The server encrypts the data using the client’s public key and sends the encrypted data.

3.The client receives this data and decrypts it.

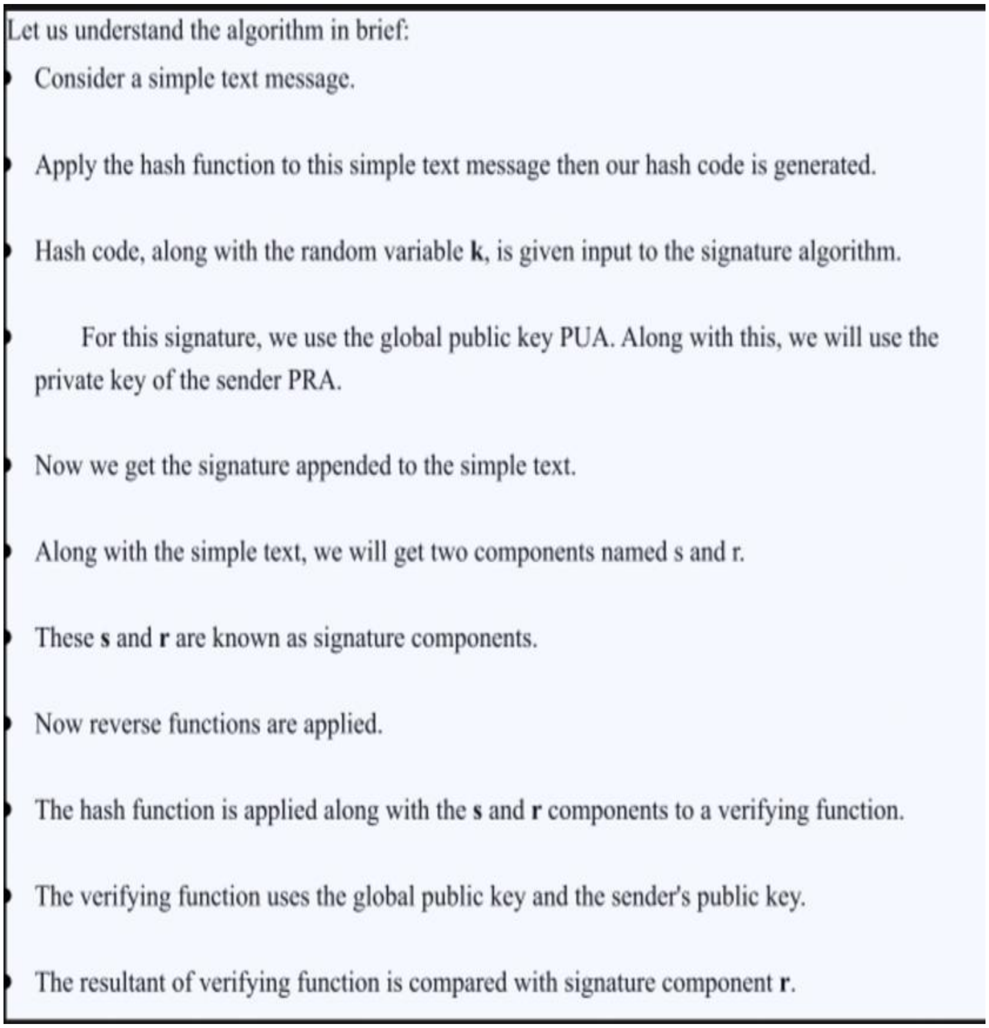
Since this is asymmetric, nobody else except the browser can decrypt the data even if a third party has the public key of the browser.

The idea! The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is a multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken in the near future. But till now it seems to be an infeasible task.

4.DSA(Digital Signature Algorithm)

We will be focussing on DSA. The NIST(National Institute of Standards and Technology) accepted the Digital Signature Algorithm as a standard in 1994. The digital signature algorithm is similar to asymmetric encryption in many ways but departs from it slightly.

In contrast to asymmetric encryption, the Digital Signature Algorithm generates a digital signature from two 160-bit values using mathematical functions.



Let us understand the algorithm in brief:

- Consider a simple text message.
- Apply the hash function to this simple text message then our hash code is generated.
- Hash code, along with the random variable **k**, is given input to the signature algorithm.
- For this signature, we use the global public key PUA. Along with this, we will use the private key of the sender PRA.
- Now we get the signature appended to the simple text.
- Along with the simple text, we will get two components named s and r.
- These **s** and **r** are known as signature components.
- Now reverse functions are applied.
- The hash function is applied along with the **s** and **r** components to a verifying function.
- The verifying function uses the global public key and the sender's public key.
- The resultant of verifying function is compared with signature component **r**.

5. Elliptic Curve Digital Signature Algorithm (ECDSA)

What is ECDSA?

The Elliptic Curve Digital Signature Algorithm is a Digital Signature Algorithm (DSA) that uses elliptic curve cryptography keys. It is a very efficient equation that is based on cryptography with public keys. ECDSA is utilized in many security systems, is popular in encrypted messaging apps, and is the foundation of Bitcoin security (with Bitcoin "addresses" serving as public keys). Elliptic Curve Digital Signature Algorithms (ECDSA) have recently received significant attention, particularly from standards developers, as alternativesto existing standard cryptosystems such as integer factorization cryptosystems and discrete logarithm problem cryptosystems. In security applications, crypto-algorithms are always the most significant fundamental tool.

Digital Signature of ECDSA

A digital signature is an electronic equivalent of a handwritten signature that allows a receiver to persuade a third party that the message was indeed sent by the sender. Handwritten signatures are substantially less secure than digital signatures. A digital signature cannot be forged in any way. Another advantage of digital signatures over handwritten signatures is that they apply to the entire message.

Every part of the digital message is affected by the signature key. On the bottom of a paper document, a handwritten signature is applied. Nothing prohibits the text displayed above the penned signature from being altered while the signature remains unaltered. Digital signatures do not allow for such changes. Today's digital signature methods can be categorized based on a mathematical issue that provides the foundation for their security:

●Integer Factorization (IF) Schemes: They rely their security on the integer factorization problem's intractability. RSA Signature Schemes are one example.

●Discrete Logarithm (DL) Schemes: Their security is based on the intractable nature of the discrete logarithm challenge in a finite field.

●Elliptic Curve (EC) Schemes: They rely their security on the elliptic curve discrete logarithm problem's intractability. The Elliptic Curve Digital Signature Algorithm, for example, is being used in this investigation and without a doubt the most recent of the many designs.

6. EdDSA- Edwards-curve Digital Signature Algorithm

In public-key cryptography, Edwards-curve Digital Signature Algorithm (EdDSA) is a digital signature scheme using a variant of Schnorr signature based on twisted Edwards curves. It is designed to be faster than existing digital signature schemes without sacrificing security.