**[Cryptography](https://www.askpython.com/python-modules/cryptography-module" \t "_blank)**

[Cryptography](https://www.askpython.com/python-modules/cryptography-module) is the practice of securing communication by using codes and ciphers. It includes a variety of techniques for converting plaintext into ciphertext, enabling secure communication, and protecting the confidentiality and integrity of data. Banking, email, e-commerce, and other industries all employ cryptography extensively. In this article you will learn about asymmetric encryption and the RSA algorithm.

**Asymmetric Encryption**

Asymmetric encryption, commonly referred to as public-key cryptography, uses two distinct keys for encryption and decryption. The public key, which is extensively used to encrypt data and is known to all, is one type of key. The private key, on the other hand, is kept private i.e., only the receiver knows it and is used to decode data.

Both, the public key and the private key should be available at both the sender’s end and the receiver’s end for the asymmetric encryption to succeed.

The encryption algorithm receives the sender’s plain text message, encrypts it using the recipient’s public key, and generates a cipher. The recipient then receives this cipher via a transmission or communication channel. The decryption process on the receiver’s end uses the decryption algorithm and the receiver’s private key to recover the original plain text message.

Asymmetric encryption typically consists of three main components:

1. Key Generation: In this step, a user generates a public-private key pair. The public key is made freely available to anyone who wants to send a message to the user, while the private key is kept secret by the user.
2. Encryption: In this step, the sender uses the recipient’s public key to encrypt the message. This ensures that only the recipient, who has the corresponding private key, can decrypt and read the message.
3. Decryption: In this step, the recipient uses their private key to decrypt the message, which was encrypted using their public key. This ensures that only the recipient can read the original message.

Although there are mathematical connections between the public key and the private key, doing so computationally is not practical. This means that anyone can encrypt data using the public key, but only the owner of the private key can decode the data.

**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.

Now let us learn about the RSA Algorithm.

**RSA Algorithm**

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:

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

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

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 RSA algorithm is a widely used public-key encryption algorithm named after its inventors Ron Rivest, Adi Shamir, and Leonard Adleman. It is based on the mathematical concepts of prime factorization and modular arithmetic.

The algorithm for RSA is as follows:

1. Select 2 prime numbers, preferably large, p and q.
2. Calculate n = p\*q.
3. Calculate phi(n) = (p-1)\*(q-1)
4. Choose a value of e such that 1<e<phi(n) and gcd(phi(n), e) = 1.
5. Calculate d such that d = (e^-1) mod phi(n).

Here the public key is {e, n} and private key is {d, n}. If M is the plain text then the cipher text C = (M^e) mod n. This is how data is encrypted in RSA algorithm. Similarly, for decryption, the plain text M = (C^d) mod n.

**Example**: Let p=3 and q=11 (both are prime numbers).

* Now, n = p\*q = 3\*11 = 33
* phi(n) = (p-1)\*(q-1) = (3-1)\*(11-1) = 2\*10 = 20
* Value of e can be 7 since 1<7<20 and gcd(20, 7) = 1.
* Calculating d = 7^-1 mod 20 = 3.
* Therefore, public key = {7, 33} and private key = {3, 33}.

Suppose our message is M=31. You can encrypt and decrypt it using the RSA algorithm as follows:

**Encryption:** C = (M^e) mod n = 31^7 mod 33 = 4

**Decryption:** M = (C^d) mod n = 4^3 mod 33 = 31

Since we got the original message that is plain text back after decryption, we can say that the algorithm worked correctly.

Being able to do both encryption and digital signatures is one of the RSA algorithm’s key benefits. To confirm that the message has not been tampered with, digital signatures are made by encrypting a message hash with the sender’s private key. This encryption may then be validated by anybody with access to the sender’s public key.

**Introduction to Elliptic Curve Cryptography**

ECC, as the name implies, is an asymmetric encryption algorithm that employs the algebraic architecture of elliptic curves with finite fields.

Elliptic Curve Cryptography (ECC) is an encryption technology comparable to RSA that enables public-key encryption.

 While RSA’s security is dependent on huge prime numbers, ECC leverages the mathematical theory of elliptic curves to achieve the same level of security with considerably smaller keys.

Victor Miller and Neal Koblitz separately proposed elliptic curve ciphers in the mid-1980s. On a high level, they are analogs of actual public cryptosystems in which modular arithmetic is substituted by elliptic curve operations.

**History of Elliptic Curve Cryptography**

Neal Koblitz and Victor S. Miller independently proposed the use of elliptic curves in encryption in 1985.

Elliptic curve cryptography algorithms entered wide use from 2004 to 2005.

In the mid-1980s, researchers found that examining elliptic curves could lead to the discovery of new sources of difficult problems. Elliptic Curve Cryptography (ECC) introduced a new degree of security to public key cryptosystems, that provide combined encryption and digital signature services.

The security of elliptic curve cryptosystems, like that of all public-key cryptosystems, is based on tough mathematical issues at the core. Given two elliptic curve points G and Y, where Y = kG.

The term “elliptic curve” is derived from the ellipse. Elliptic curves were discovered in the form of the Diophantine equation for c, after the 17th century. Furthermore, while calculating the surface of the ellipse is simple, calculating the circumference of the ellipse is difficult. The equation can be simplified to an integral:

Components of Elliptic Curve Cryptography

Below are the components of elliptic curve cryptography:

**1. ECC keys:**

* Private key: ECC cryptography’s private key creation is as simple as safely producing a random integer in a specific range, making it highly quick. Any integer in the field represents a valid ECC private key.
* Public keys: Public keys within ECC are EC points, which are pairs of integer coordinates x, and y that lie on a curve. Because of its unique features, EC points can be compressed to a single coordinate + 1 bit (odd or even). As a result, the compressed public key corresponds to a 256-bit ECC.

**2. Generator Point:**

* ECC cryptosystems establish a special pre-defined EC point called generator point G (base point) for elliptic curves over finite fields, which can generate any other position in its subgroup over the elliptic curve by multiplying G from some integer in the range [0…r].
* The number r is referred to as the “ordering” of the cyclic subgroup.
* Elliptic curve subgroups typically contain numerous generator points, but cryptologists carefully select one of them to generate the entire group (or subgroup), and is excellent for performance optimizations in calculations. This is the “G” generator.

**Elliptic Curve Cryptography Algorithms**

Based on the arithmetic of elliptic curves over finite fields, Elliptic-Curve Cryptography (ECC) provides numerous sets of algorithms:

**Digital signature algorithms:**

* **Elliptic Curve Digital Signature Algorithm. (ECDSA):**ECDSA, or Elliptic Curve Digital Signature Algorithm, is a more highly complicated public-key cryptography encryption algorithm. Elliptic curve cryptography is a type of public key cryptography that uses the algebraic structure of elliptic curves with finite fields as its foundation. Elliptic curve cryptography is primarily used to generate pseudo-random numbers, digital signatures, and other data.
* **Edwards-curve Digital Signature Algorithm (EdDSA):**The Edwards-curve Digital Signature Algorithm (EdDSA) was proposed as a replacement for the Elliptic Curve Digital Signature Algorithm for performing fast public-key digital signatures (ECDSA). Its primary benefits for embedded devices are higher performance and simple, secure implementations. During a signature, no branch or lookup operations based on the secret values are performed. Many side-channel attacks are foiled by these properties.

**Encryption algorithms:**

* **Elliptic Curve Integrated Encryption Scheme (ECIES):**ECIES is a public-key authenticated encryption scheme that uses a KDF (key-derivation function) to generate a separate Medium Access Control key and symmetric encryption key from the ECDH shared secret. Because the ECIES algorithm incorporates a symmetric cipher, it can encrypt any amount of data. In practice, ECIES is used by standards such as Intelligent Transportation Systems.
* **EC-based ElGamal Elliptic Curve Cryptography:** ElGamal Elliptic Curve Cryptography is the public key cryptography equivalent of ElGamal encryption schemes that employ the Elliptic Curve Discrete Logarithm Problem. ElGamal is an asymmetric encryption algorithm that is used to send messages securely over long distances. Unfortunately, if the encrypted message is short enough, the algorithm is vulnerable to a Meet in the Middle attack.

**Key Agreement algorithm:**

* **Elliptic-curve Diffie–Hellman (ECDH):**Elliptic-curve Diffie-Hellman (ECDH) is a key agreement protocol that enables two parties to establish a shared secret over an insecure channel, each with an elliptic-curve public-private key pair. This shared secret can be used directly as a key or to generate another key. Following that, the key, or the derived key, can be used to encrypt subsequent communications with a symmetric-key cipher.
* **Fully Hashed Menezes-Qu-Vanstone(FHMQV):**Fully Hashed Menezes-Qu-Vanstone is an authenticated key agreement protocol based on the Diffie-Hellman scheme. MQV, like other authenticated Diffie-Hellman schemes, protects against an active attacker. The protocol can be adapted to work in any finite group, most notably elliptic curve groups, in which it is recognized as elliptic curve MQV (ECMQV).

**Application of Elliptic Curve Cryptography**

* **Diffie-Hellman:**The basic public-key cryptosystem suggested for secret key sharing is the Diffie-Hellman protocol. If A (Alice) and B (Bob) initially agree on a given curve, field size, and mathematical type. They then distribute the secret key in the following manner. We can see that all we need to build the Diffie-Hellman protocol is scalar multiplication.
* **Elliptic Curve Digital Signature Algorithm (ECDSA):**ECC is one of the most widely utilized digital signature implementation approaches in cryptocurrencies. In order to sign transactions, both Bitcoin and Ethereum use the field inverse multiplication, but also arithmetic multiplication, inverse function, and modular operation.
* **Online application:**Moreover, ECC is not limited to cryptocurrencies. It is an encryption standard that will be utilized by most online apps in the future due to its reduced key size and efficiency. Most commonly used in cryptocurrencies such as Bitcoin and Ethereum, along with single-way encryption of emails, data, and software.
* **Blockchain application:**The cryptocurrency Bitcoin employs elliptic curve cryptography.   Ethereum 2.0 makes heavy use of elliptic curve pairs with BLS signatures, as stated in the IETF proposed BLS specification, to cryptographically ensure that a specific Eth2 validator has really verified a specific transaction.

| **Parameters** | **ECC** | **RSA** |
| --- | --- | --- |
| **Working algorithm** | ECC is a cryptography technique that works just on a mathematical model of elliptic curves. | RSA cryptography algorithm is primarily based on the prime factorization approach. |
| **Bandwidth savings** | ECC gives significant bandwidth savings over RSA. | RSA provides much lesser bandwidth saving than ECC. |
| **Encryption process** | The encryption process takes less time in ECC. | The encryption process takes more time in RSA. |
| **Decryption process** | The decryption process takes more time. | Decryption is faster than ECC. |
| **Security** | ECC is much safer than RSA and is currently in the process of adapting. | RSA is heading toward the end of its tenure. |

**Types of Security Attacks**

* **Side-channel attack:**Side-channel attacks in elliptic curve cryptography are caused by unintended information leaking during processing. The computation of n\*P, where n is a positive number and P is a location on the elliptic curve E, is a critical operation.
* **Backdoor attack:**Concerns have been made by cryptographic specialists that the National Security Agency has installed a kleptographic backdoor into at least one elliptic curve-based pseudo-random generator. According to one investigation of the potential backdoor, an attacker in possession of the algorithm’s secret key might access encryption keys provided only 32 bytes of outputs.
* **Quantum computing attacks:**By calculating discrete logarithms on a hypothetical quantum computer, Shor’s technique can be used to break elliptic curve cryptography. The most recent quantum resource estimates are 2330 qubits and 126 billion Toffoli gates for cracking a curve with only a 256-bit modulus (128-bit security level).

**Benefits of Elliptic Curve Cryptography**

* **Fast key generation:**ECC cryptography’s key creation is as simple as securely producing a random integer in a specific range, making it highly quick. Any integer in the range represents a valid ECC secret key. The public keys in the ECC are EC points, which are pairs of integer coordinates x, and y that lie on a curve.
* **Smaller key size:**Cipher text, signatures, and Elliptic-curve cryptography (ECC) is a public-key encryption technique based on the algebraic structure of elliptic curves with finite fields. Compared to non-EC encryption (based on ordinary Galois fields), ECC allows for fewer keys to guarantee equal security**.**
* **Low latency:**Signatures can be computed in two stages, allowing latency much lower. By computing signatures in two stages, ECC achieves lower latency than the inverse throughout. ECC has robust protocols for authorized key exchange, and the technology has widespread adoption.
* **Less computation power:**Since the ECC key is shorter the computation power is also less computational power, ECC offers high security with faster, shorter keys compared to RSA and take more energy to factor than it does to calculate an elliptic curve objective function.
* **High security:**A 256-bit ECC public key ensures comparable security to a 3072-bit RSA public key. With ECC, you may obtain the same level of security with smaller keys. ECC provides strong security in a world where mobile phones must do more and more encryption with fewer computational resources.

**Limitations of Elliptic Curve Cryptography**

* **Large encryption size:**ECC increases the size of the encrypted message significantly more than RSA encryption. The default key length for ECC private keys is 256 bits, but many different ECC key sizes are conceivable depending on the curve.
* **A more complex:**The ECC algorithm is more complete and more difficult to implement than RSA. Algorithms cost have been computed from the computation of the elliptic curve operation and finite field operations that determine the running time of the scalar multiplication integer sub-decomposition (ISD) method.
* **Complex security:**Complicated and tricky to implement securely, mainly the standard curves. If the key size used is large enough, ECC is regarded to be highly secure. For internal communications, the US government needs ECC with a key size of either 256 or 384 bits, depending on the sensitivity level of the material being communicated.
* **Binary curves:**Processing of binary curves is costly. Elliptic curve cryptography (ECC) employs elliptic curves over finite fields Fp (where p is prime and p > 3) or F2m (where the field size p = 2 m\_). This means that the field is a p x p square matrix, and the points on the curve can only have integer locations within the field.

**Conclusion**

* **Encryption strength:**The main distinction between RSA and ECC certificates is the encryption strength. When compared to other approaches, such as RSA, ECC can provide a level of security that uses fewer processing resources to encrypt and decrypt data.
* **ECC Keys feature:** With a lower key length, Elliptic Curve Cryptography (ECC) delivers the same level of encryption strength as the RSA.ECC and other public key encryption systems use a mathematical technique to combine two separate keys and then use the resulting output to encrypt and decrypt data. One is a public key that anybody can see, and the other is a private key that only the sender and receiver of the data can see.
* **ECC certificates:** As a result, for Public Key Infrastructure, an ECC certificate provides more speed and security than an RSA certificate. Elliptic Curve Cryptography (ECC) provides an equivalent level of encryption strength to the RSA algorithm with a shorter key length.
* **ECC curves:** The elliptic curve over a finite area gives us more security. For contemporary ECC purposes, an elliptic curve is a plane curve over a finite field composed of points fitting the equation: Any point on the curve in this elliptic curve cryptography example can be mirrored over the x-axis and the curve will remain unchanged.
* Use of prime number: Zp (where p is a prime number) elliptic curve When p is a huge prime integer, it indicates that the cipher text is extremely tough to crack. The public and shared keys are both 257 bits long (65 hexadecimal digits, 256 bits due to key compression). The private keys KA and KB are different due to randomness, but the estimated shared secret key across (A) and (B) will always be the same.