

CompTIA Security+

Lecture 02

SY0-701

Cryptographic Solutions

By
Dr. Khaled ELMenshawy

Lecture

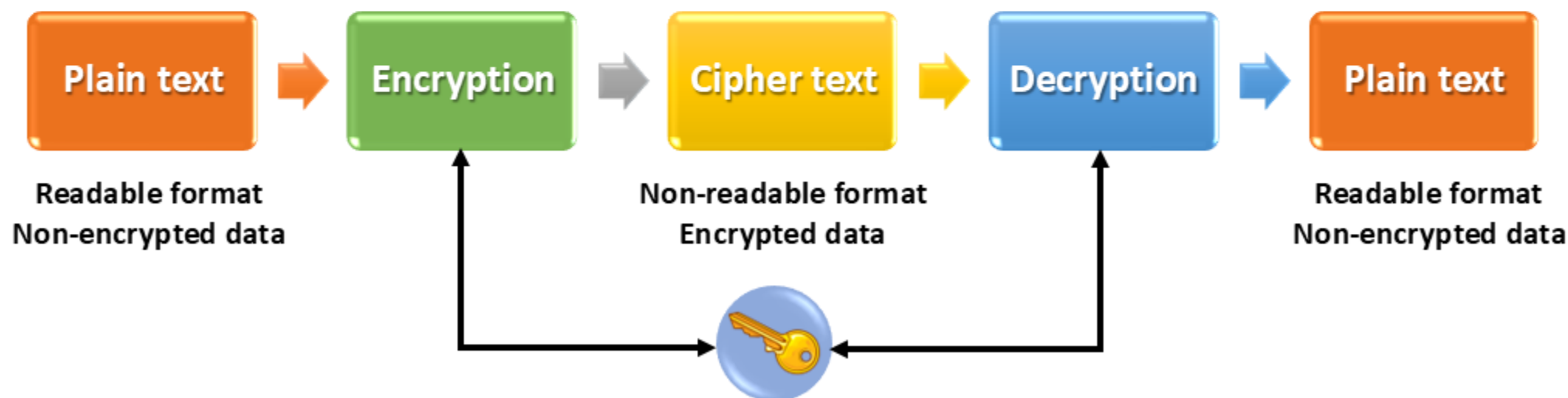
02

LEARNING OUTCOMES

- Understanding Cryptography.
- Recognize types of Encryption.
- Recognize different algorithms of symmetric encryption.
- Recognize different algorithms of asymmetric encryption.
- Understanding the Hash function and Digital Signature.
- Understanding Digital Certificates and PKI.
- Recognize Authentication Protocols
- Recognize Encryption Tools
- Understanding Obfuscation.
- Understanding Blockchain.

What is Cryptography?

Cryptography is the practice of writing and solving codes in order to hide the true meaning of the information.



What is Encryption?

Encryption is the process of converting ordinary information, known as *plaintext data*, into an unintelligible format known as *ciphertext data*. Encryption is used to protect data at rest, data in transit, and data in use.

What is a Cipher?

A Cipher is an *algorithm* that perform encryption and decryption.

What is a Key?

A Key is an essential piece of information that *determines the output of the cipher*. The longer the key, the more secure it's going to be.

What is Encryption?

Encryption is the process of converting ordinary information, known as *plaintext data*, into an unintelligible format known as *ciphertext data*. Encryption is used to protect data at rest, data in transit, and data in use.

What is a Cipher?

A Cipher is an *algorithm* that perform encryption and decryption.

What is a Key?

A Key is an essential piece of information that *determines the output of the cipher*. The longer the key, the more secure it's going to be.

Types of encryption

Symmetric



Confidentiality



Non-repudiation



Plain text

Encryption
Algorithm

Ciphertext

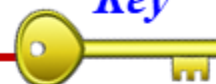
Decryption
Algorithm

Plain text



A single key that's used to encrypt and decrypt that data.

Key



Asymmetric



Confidentiality



Non-repudiation



Plain text

Encryption
Algorithm

Ciphertext

Decryption
Algorithm

Plain text



Two keys one for encryption and the other for decryption

Ahmed Public Key



Ahmed Private Key



Symmetric	Asymmetric
It only requires a single key for both encryption and decryption.	It requires two keys, a public key and a private key, one to encrypt and the other one to decrypt.
The size of cipher text is the same or smaller than the original plain text.	The size of cipher text is the same or larger than the original plain text.
The encryption process is 100 to 1000 times faster.	The encryption process is slow.
It is used when a large amount of data is required to transfer.	It is used to transfer small amounts of data.
It only provides confidentiality.	It provides confidentiality, authenticity, and non-repudiation.
The length of key used is 128 or 256 bits.	The length of key used is 2048 or higher.
Key distribution by either symmetric or asymmetric encryption.	Key distribution by public announcement of public key, public available directory, public key authority, or public key certificate.

Types of Cipher

Block

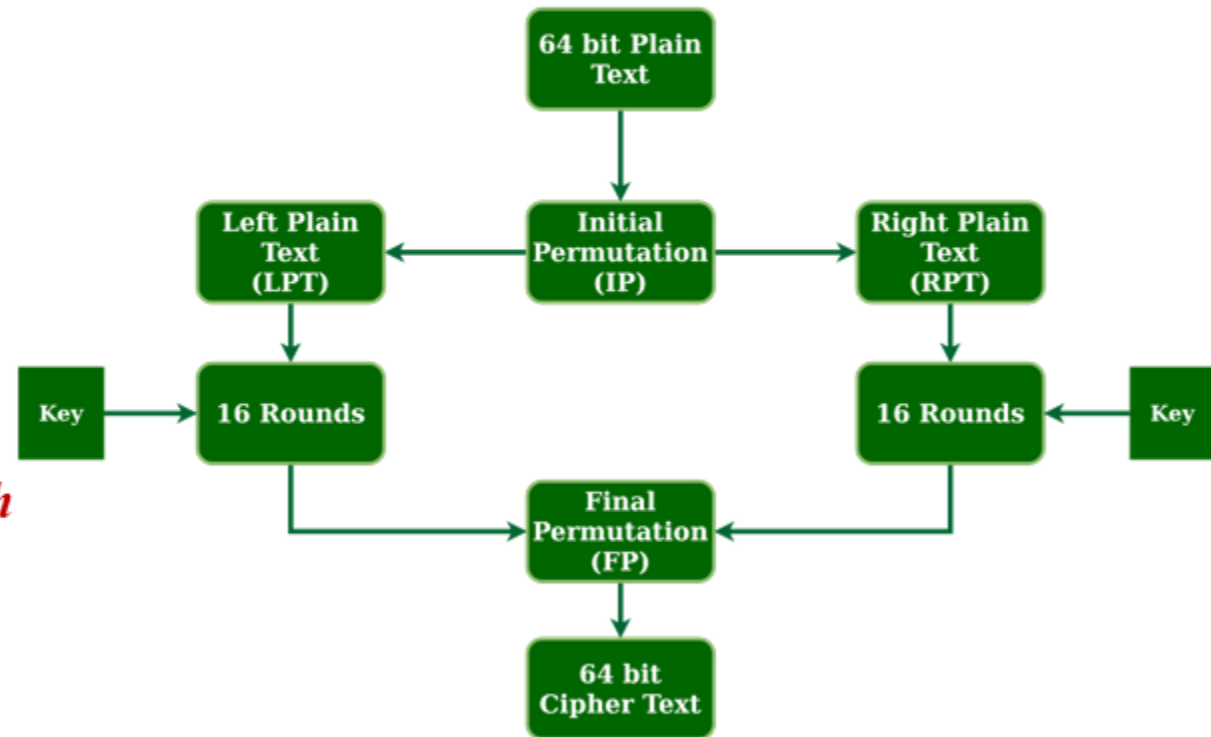
Breaking the input into fixed length blocks of data before performing their encryption functions. For example, a message that was one kilobyte in size, we could break that up into 16 blocks of 64 bytes each. Each of these 16 blocks could then be processed by the cipher, and then output out the other side of this algorithm

Stream

*Performs their computations and **encryptions a single byte at a time**. This makes it **a bit-by-bit process**, and they utilize a key stream generator by mixing with the input plain text **using an exclusive or (XOR) function**. They're very well-suited for **securing real-time communication data streams** like IoT sensors, streaming audio or streaming video.*

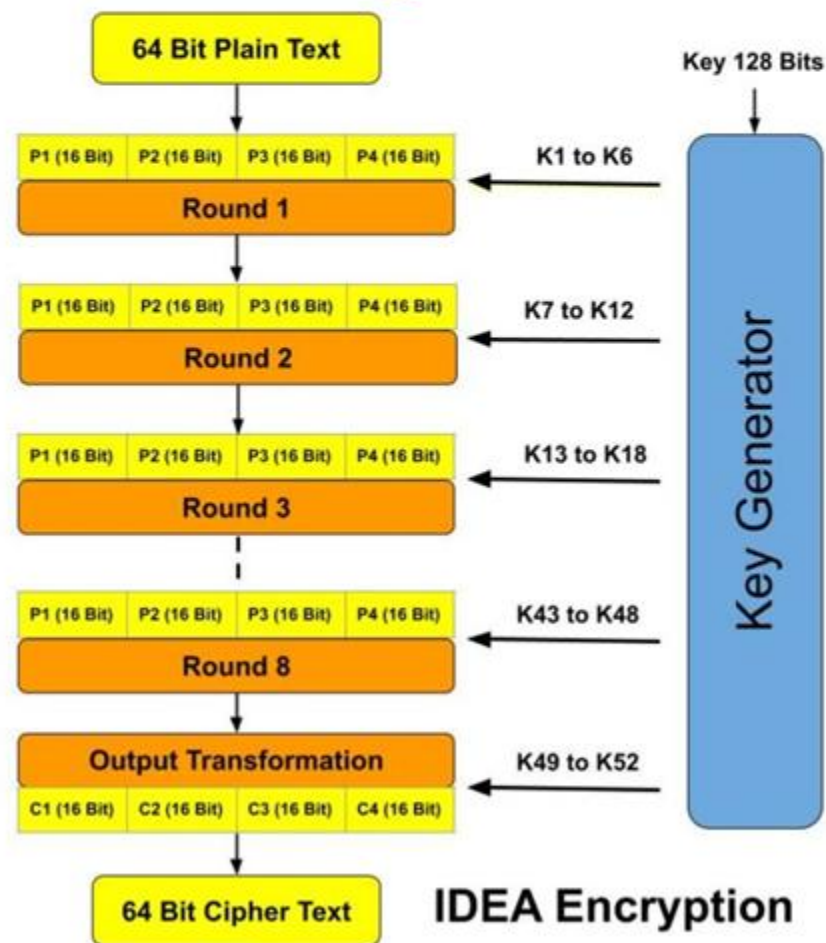
Data Encryption Standard (DES)

- **DES** is a *64-bit block symmetric cipher* (because the key works only on 64 bits of data at a time).
- The *key is 56 bits*, with 8 bits used for error checking and *16 rounds*.
- Making DES more secure:
 - **3DES**: *each block of data is encrypted 3 times with 3 different keys.*
- DES encryption is obsolete today but 3DES will continue to be a reliable encryption algorithm at least through 2030.



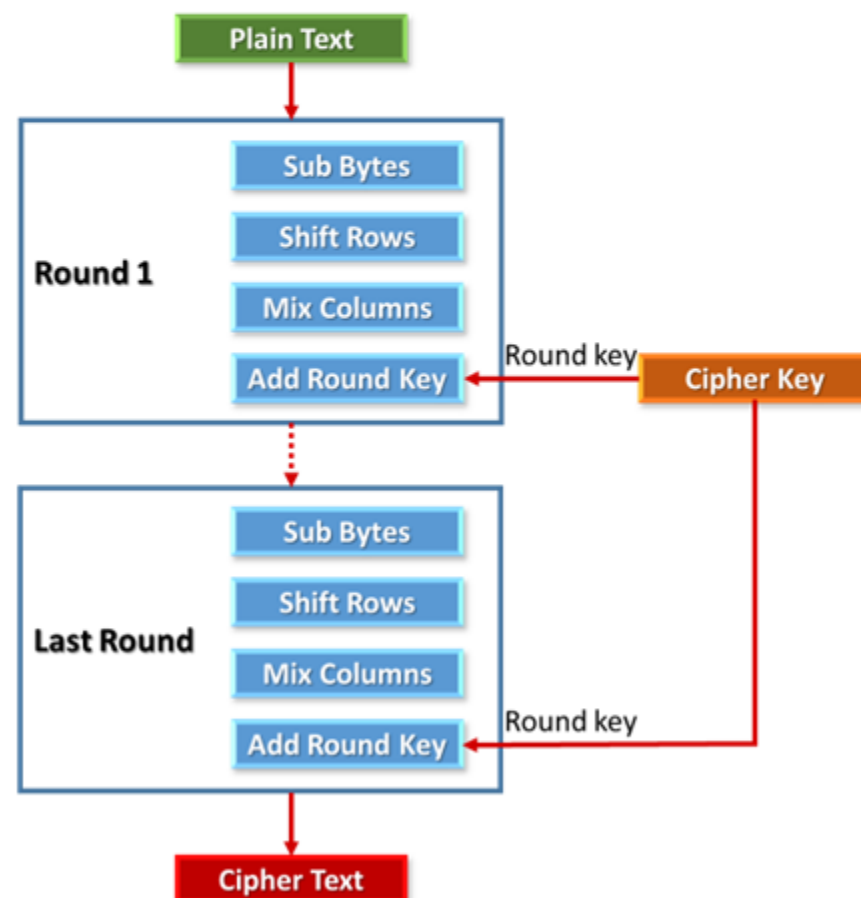
International Data Encryption Algorithm (IDEA)

- **IDEA** is a *64-bit block symmetric cipher*.
- The *key is 128 bits*.
- Total *8.5 rounds (52 sub-key)*.
- Although **IDEA** encryption is considered secure, *it is still vulnerable to brute-force attacks*.



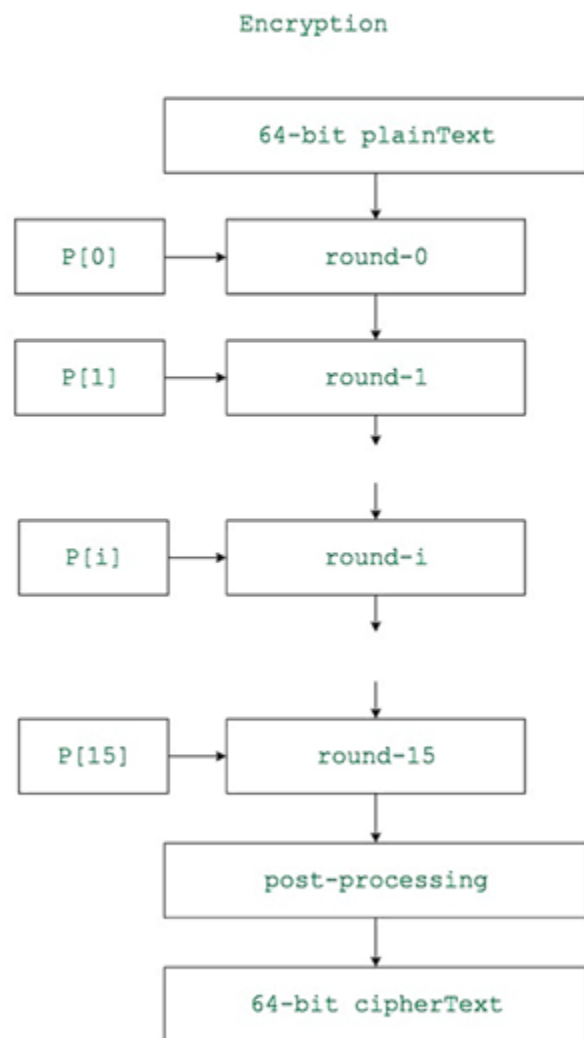
Advanced encryption standard (AES)

- **AES** is a **128-bit block symmetric cipher** (because the key works only on 128 bits of data at a time).
- The **key size can be 128/192/256 bits**.
- The number of rounds depends on the **key length** as follows :
 - **128 bit key – 10 rounds**
 - **192 bit key – 12 rounds**
 - **256 bit key – 14 rounds**
- **AES** is the most widely used encryption algorithm.
- Both AES and 3DES are unbreakable with current technology.



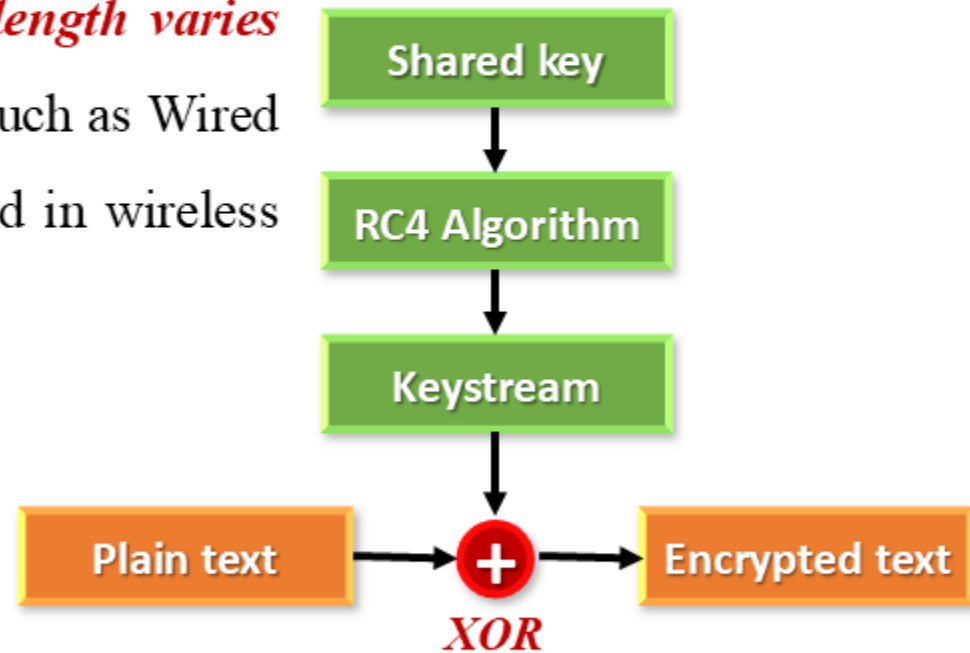
Blowfish and Two fish

- **Blowfish** is a *64-bit block symmetric cipher* (because the key works only on 128 bits of data at a time).
- The *key size can be 32-bits to 448-bits variable size*.
- The number of rounds *16 rounds*.
- **Two fish** is a *128-bit block symmetric cipher*, and The *key size can be 128-, 192-, or 256-bits*.
- Both **Blowfish** and **Two fish** were released as open-source products that people could use.



RC4, RC5, and RC6 Cipher

- **RC4** is a *stream symmetric cipher*, which used a shared *key length varies from 40 to 2048 bits*. It is the basis for a number of protocols such as Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA) used in wireless communications, and Secure Socket Layer (SSL).
- **RC5 & RC6** are a *32-, 64- and 128-bit variable block symmetric cipher*, which used a *variable key length up to 2048 bits*. Number of *rounds vary between 0 to 255 rounds*.



Diffie-Hellman

- **Diffie-Hellman** is a *key-exchange and secure key distribution protocol* that enables two parties communicating over public channel to establish a mutual secret without it being transmitted over the Internet.
- **Diffie-Hellman** is widely used whenever you're setting up a VPN tunnel or other encryption tunnels that require a symmetric algorithm's shared secret key, which is a private key, to be exchanged first before setting up that symmetric tunnel. And then using asymmetric algorithms like Diffie-Hellman, we can do that in a secure and scalable manner.

Rivest, Shamir, and Adleman (RSA)

- **RSA** is *asymmetric variable-length block sizes cipher* can support *key sizes between 1024 bits and 4,096 bits*.
- Message: just a bit pattern which can be represented by an integer number e.g., $m = 10010001$. This message is represented by the decimal number **145**.
- Thus, encrypting a message is equivalent to encrypting a number. which gives a new number (the ciphertext).
 - Choose two large prime numbers p, q . (e.g., 1024 bits each).
 - Compute $n = p * q, z = (p-1) * (q-1)$.
 - Choose e (with $e < n$) that has no common factors with z (e, z are “relatively prime”).
 - Choose d such that $ed \bmod z = 1$.
 - *Public key* is (n, e) . *private key* is (n, d) . [$C = M^e \bmod n, M = C^d \bmod n$]

$\underbrace{(n, e)}_{K_B^+}$

$\underbrace{(n, d)}_{K_B^-}$

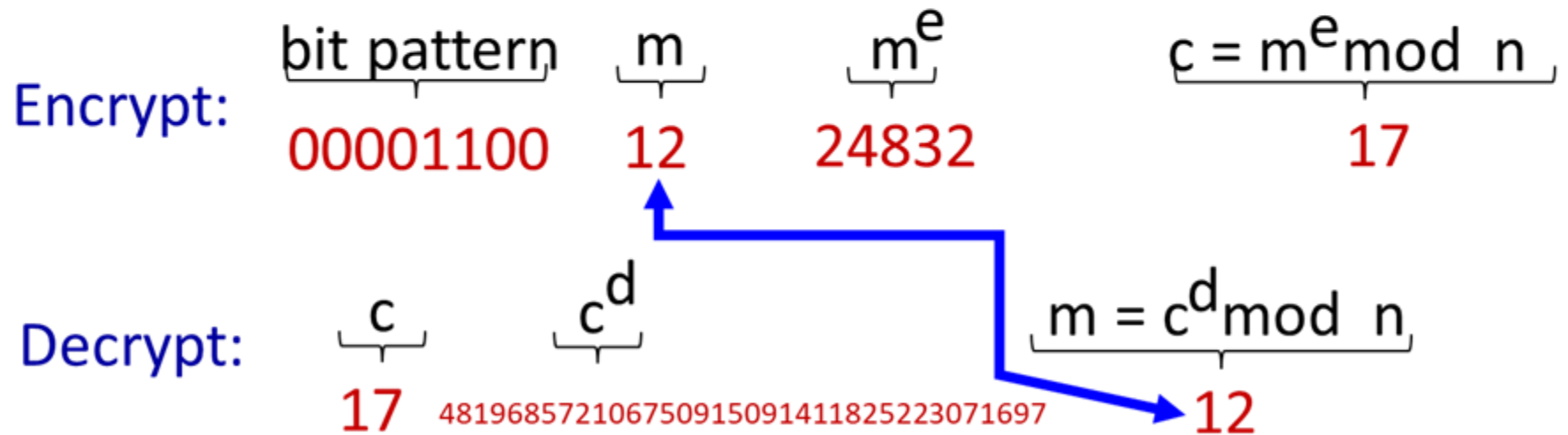
Chooses $p=5$, $q=7$. Then $n=35$, $z=24$.

$e=5$ (so e , z relatively prime).

$d=29$ (so $ed \bmod z = 1$).

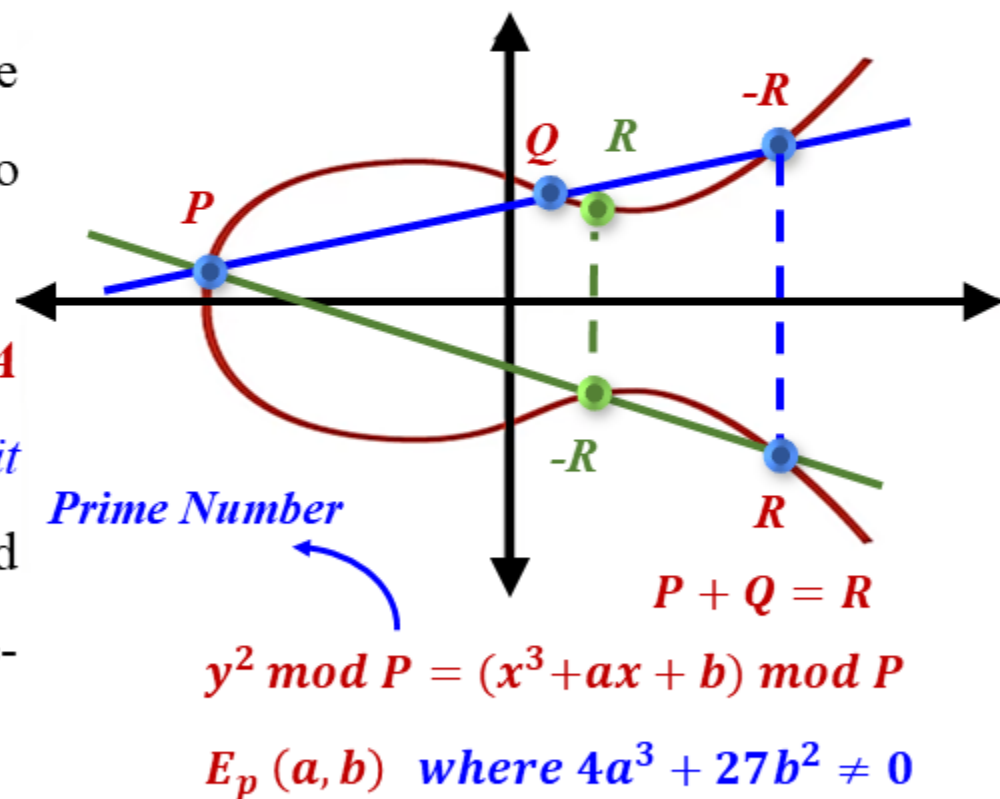
Encrypting 8-bit messages.

RSA Example



Elliptic Curve Cryptography (ECC)

- **ECC** is heavily used in mobile devices, and it's based on the algebraic structure of elliptical curves ($y^2 = x^3 + ax + b$) to define its keys.
- **ECC's** algorithm is about *six times more efficient than an RSA* algorithm (*a 256-bit key with ECC, actually have the same security as a 2,048-bit key with RSA*). For this reason, **ECC** is used in a lot of things like tablets and smartphones and other mobile-based implementations (low-power computing devices).



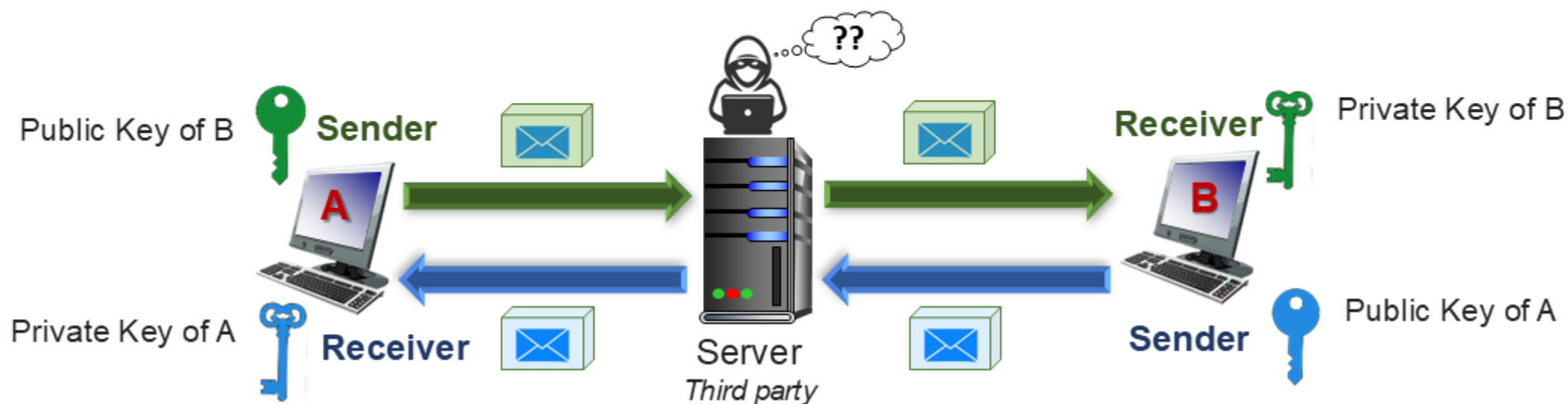
Elliptic Curve Cryptography (ECC)

➤ There are a few variations of ECC including:

- **ECDH**, or the Elliptic Curve Diffie-Hellman.
- **ECDHE**, or the Elliptic Curve Diffie-Hellman Ephemeral, which uses a different key for each portion of the key establishment process inside of the Diffie-Hellman key exchange.
- **ECDSA**, or the Elliptic Curve Digital Signature Algorithm for all of digital signature needs.

Pretty Good Privacy (PGP)

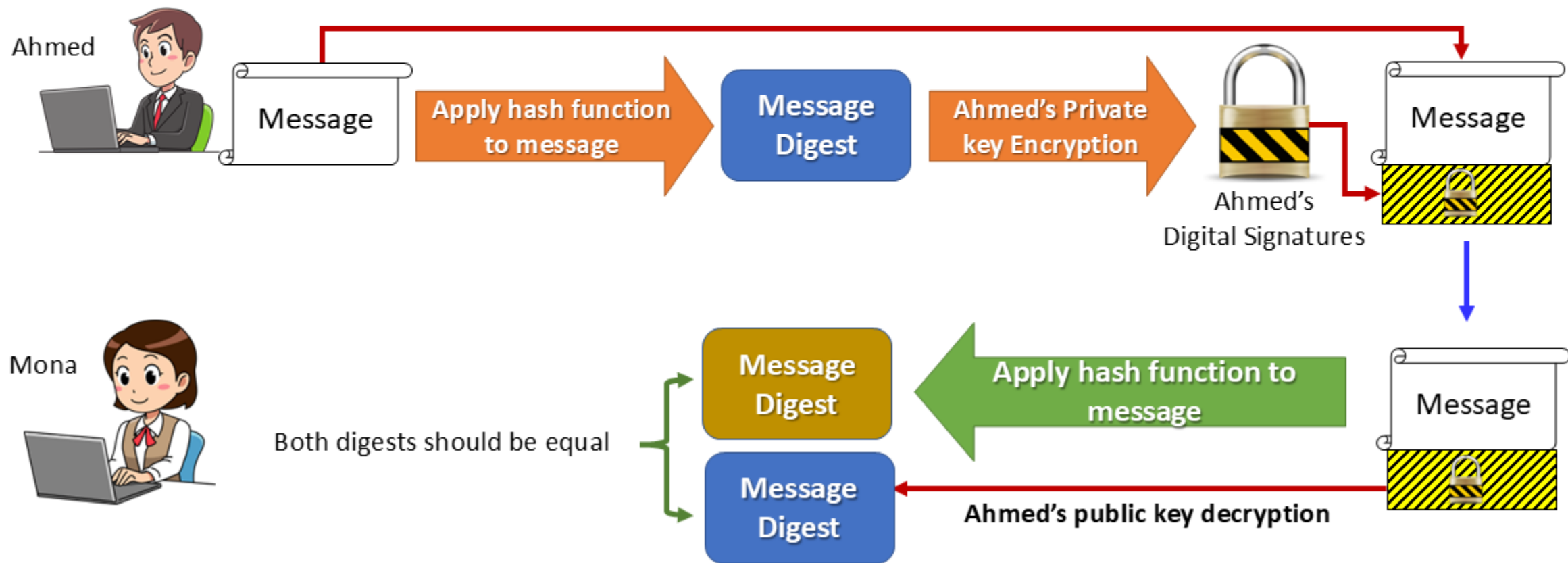
- **End-to-end encryption (E2EE)** is a type of asymmetric encryption and used to secure communication that prevents third parties from accessing data while it's transferred from one end system to another.



- Hash function takes a *string of variable length* and encodes as a *fixed-length hash value* called message digest. It does that via complex mathematical operations and it is *one-way function property*.
- *Collision Resistant property*; impossible to find two different inputs that produce the same output.
- A hash value is a like a fingerprint used to ensure the message's integrity.
- There are several different classes of hash functions. The most used are:
 - **Message Digest Algorithm (MD5)**; it produces a *128-bit hash value*. it can only create a limited number of unique values, and this can lead to two files having the exact same resulting hash digest, which we call a *collision*.

- **Secure Hash Algorithm (SHA)**; comprise of four SHA algorithms; SHA-1, SHA-2, and SHA-3.
 - ❖ **SHA-1** creates *160-bit hash digest*.
 - ❖ **SHA-2** is a family of hash functions that create *224 bits up to 512 bits hash digest*
 - ❖ **SHA-1 and SHA-2** are going to perform a different number of rounds of mathematical computations to create their hash digest, anywhere from *64 to 80 rounds*.
 - ❖ **SHA-3** just like SHA-2 can go between *224 bits and 512 bits hash digest*. is that it has a major increase in security because it uses *120 rounds* of computations to create its message digest for each unique file.
- **RACE Integrity Primitives Evaluation Message Digest (RIPEMD)**; provides *160-bit, 256-bit and 320-bit hash value* .
- **Hash-based Message Authentication Code (HMAC)**; will be paired with other algorithms to check the integrity of a message like HMAC-MD5 or HMAC-SHA1, or HMAC-SHA256.

Digital signatures provide more security than other electronic signatures. It is analogous to hand-written signatures.

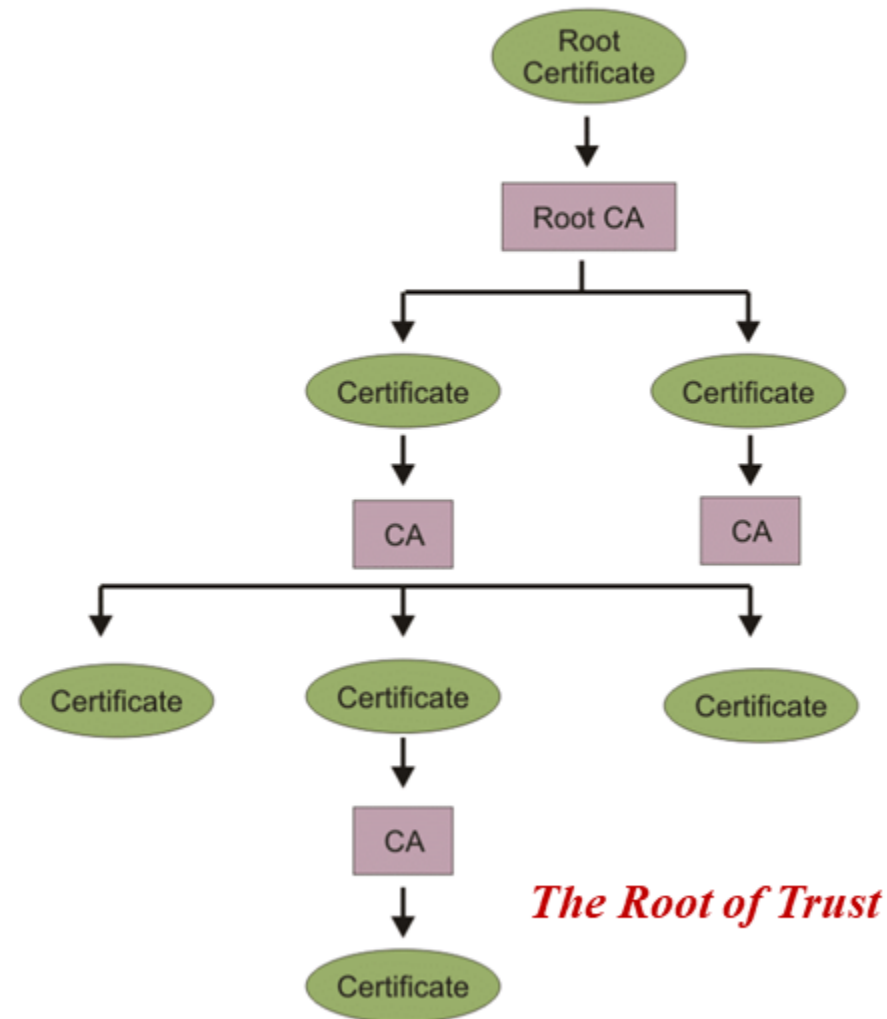


- **A digital certificate** is a digitally signed electronic document that *binds a public key with a user's identity*, these certificates are commonly going to use the *X.509 protocol standard*. Each certificate contains the public key of a user and is *signed with the private key of a trusted certification authority*.
- **A certificate authority (CA)** is the one responsible for issuing digital certificate, the CA may be a third-party organization such as VeriSign or internal to organization.
- **Revocation** of digital certificates happens when:
 - Certification is not used any more.
 - Details of certifications are changed.
 - The certificate owner's private key is compromised



- **Wildcard Certificates** are going to *allow all the sub-domains to use the same public key* certificate and have it displayed as valid. When using multiple domain names with the same certificate, the **Subject Alternative Name (SAN)** field should be modified instead of using the wild card.
- **Single-Sided certificates** are creating a secure session that can be established between the server who's going to identify itself to the client's web browser *using the server's digital certificate (single sided)*.
- **Dual-Sided certificates** are creating a secure session that can be established between the server and the client's web browser *using the digital certificate of both server and client to authenticate connection (dual sided)*.
- **Self-Signed certificates** are digital certificates that are *signed by the same entity* whose identity it's certifying, *rather than by a trusted certificate authority* or third party.

- **Third-party Certificates** are digital certificates *issued and signed by a trusted certificate Authority*, which we call a CA. Now, these certificate authorities will have *their root certificates embedded into major web browsers and operating systems*, which means that any certificate they issue is inherently going to be trusted by that system or browser.
- **The Root of Trust** is a root certificate that can be used to issue other certificates in the TLS/SSL system. These certificates are issued by a verified certificate authority (CA).



- **Certificate Authority** is the *trusted third party who's going to issue these digital certificates* and therefore the certificate is also going to contain their name, their digital signature, the serial number for that certificate, the issue and expiration date, and the version of that certificate.
- **Registration Authority** is responsible for receiving certificate signing requests from applicants and forwards these requests to a Certificate Authority (CA).
- **Certificate Signing Request (CSR)** is a *block of encoded text that contains information about the entity that's requesting the certificate*, including the organization's name, domain name, locality, and country.
- **Certificate Revocation List (CRL)** is an online list of digital certificates that the certificate authority has already revoked.

- **Online Certificate Status Protocol (OCSP)** is *determining the revocation status of any single digital certificate using its serial number* and it operates as an alternative to the certificate revocation list.
- **OCSP stapling** is an *alternative to OCSP* that will allow the certificate holder to *get the OCSP record from the server at regular intervals* and include it as part of the SSL/TLS handshake.
- **Public key pinning** allows an HTTPS website to *resist impersonation attacks* from those who are trying to present fraudulent certificates by presenting a set of trusted public keys to the user's web browser as part of its HTTP header. Now, if the web browser doesn't get the matching public key from the certificate authority, then it knows that that website was compromised.
- **Key recovery agents** allows the *restoration of a lost or corrupted key to be performed*.

Public Key Infrastructure (PKI)

PKI is an entire system of hardware, software, policies, procedures and people that is based on asymmetric encryption used to *protect communications between the webserver and the clients*.

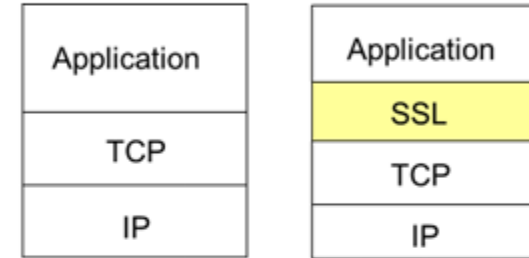
➤ **PKI has a common elements such as;**

- **Digital Certificates**; it's a form of electronic identification for websites and organizations.
- **Certificate Authority (CA)**; are going to issue *digital certificates* and these certificate authorities are also going to *keep the level of trust between all the certificate authorities around the world intact*. e.g., *VeriSign /Symantec, Kaspersky, Comodo, and DigiCert*.
- **Key escrow**; refers to the process where *cryptographic keys are going to be stored in a secure, third-party location*, which is effectively called an escrow. In case of an organization loses access to their encryption keys, these keys can be retrieved from that escrow account.

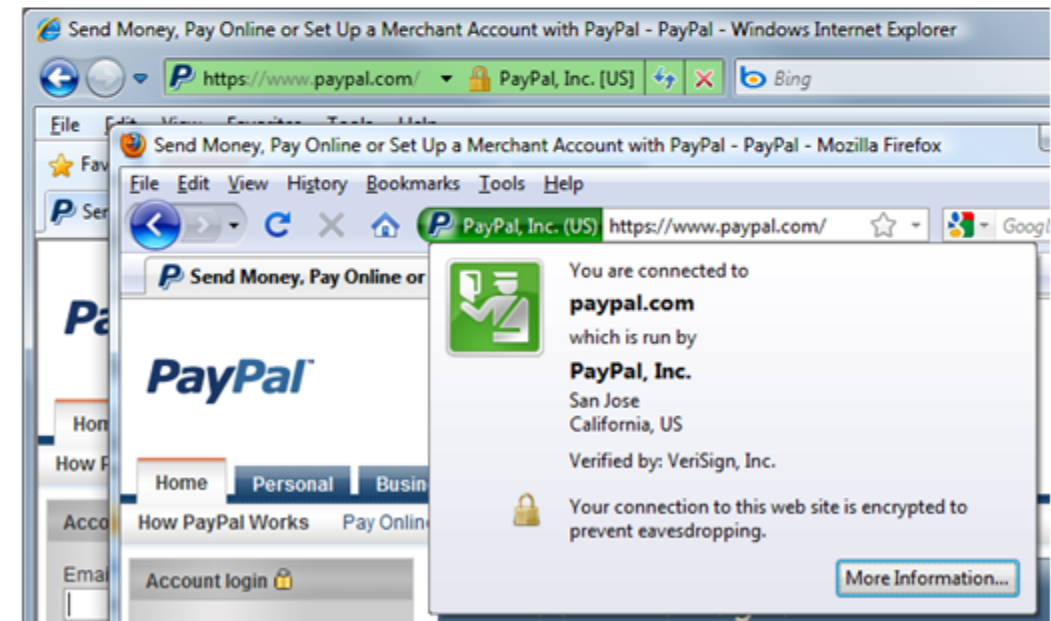
Public Key Infrastructure (PKI)

Transport Layer Security & Secure Socket Layer (TLS/SSL)

- **Secure Sockets Layer (SSL)** is a *communication protocol* that creates a *secure connection between two devices* or applications on a network.
- **Transport Layer Security (TLS)** is the upgraded version of **SSL** that fixes existing SSL vulnerabilities.
- Used with **HTTPS** protocol (Secured version of HTTP).
- The TLS/SSL encryption is created with a *SSL certificate*.
Certificates are issued by a certification company and contains information and instructions to transmit information between a web server and a user using a public key and a private key.

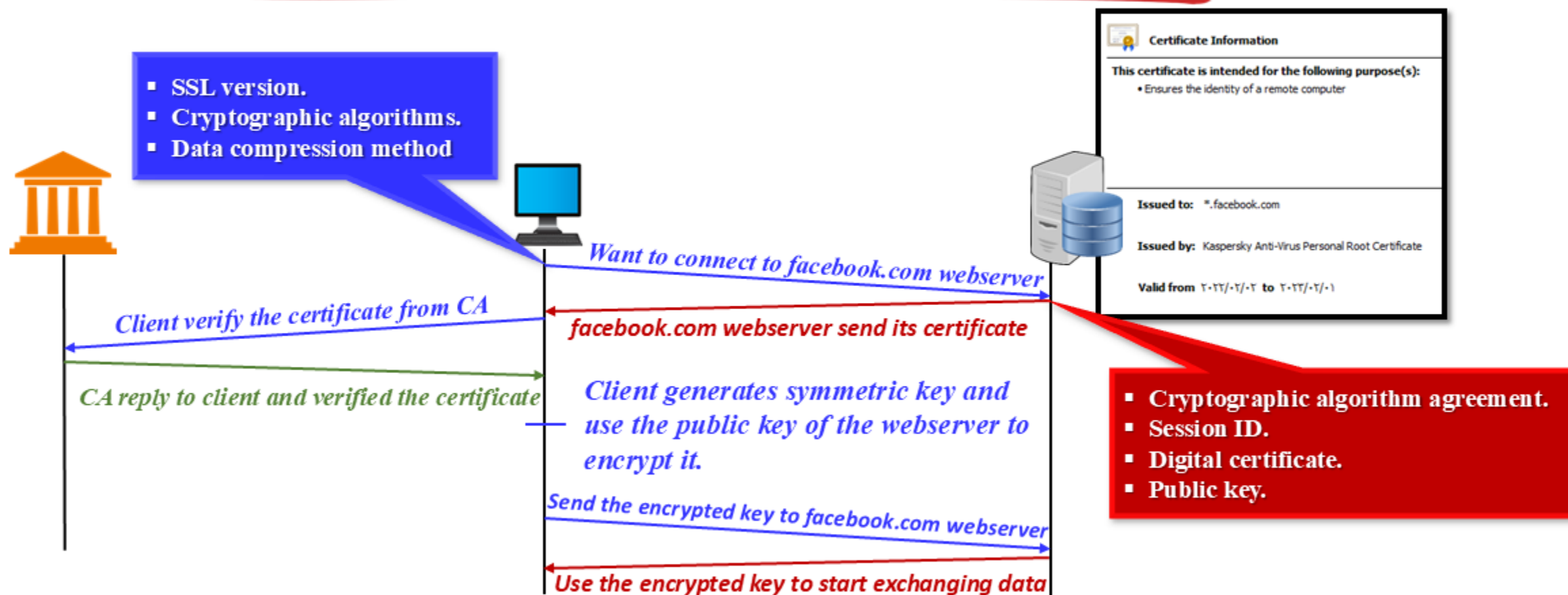


Normal application Application with SSL

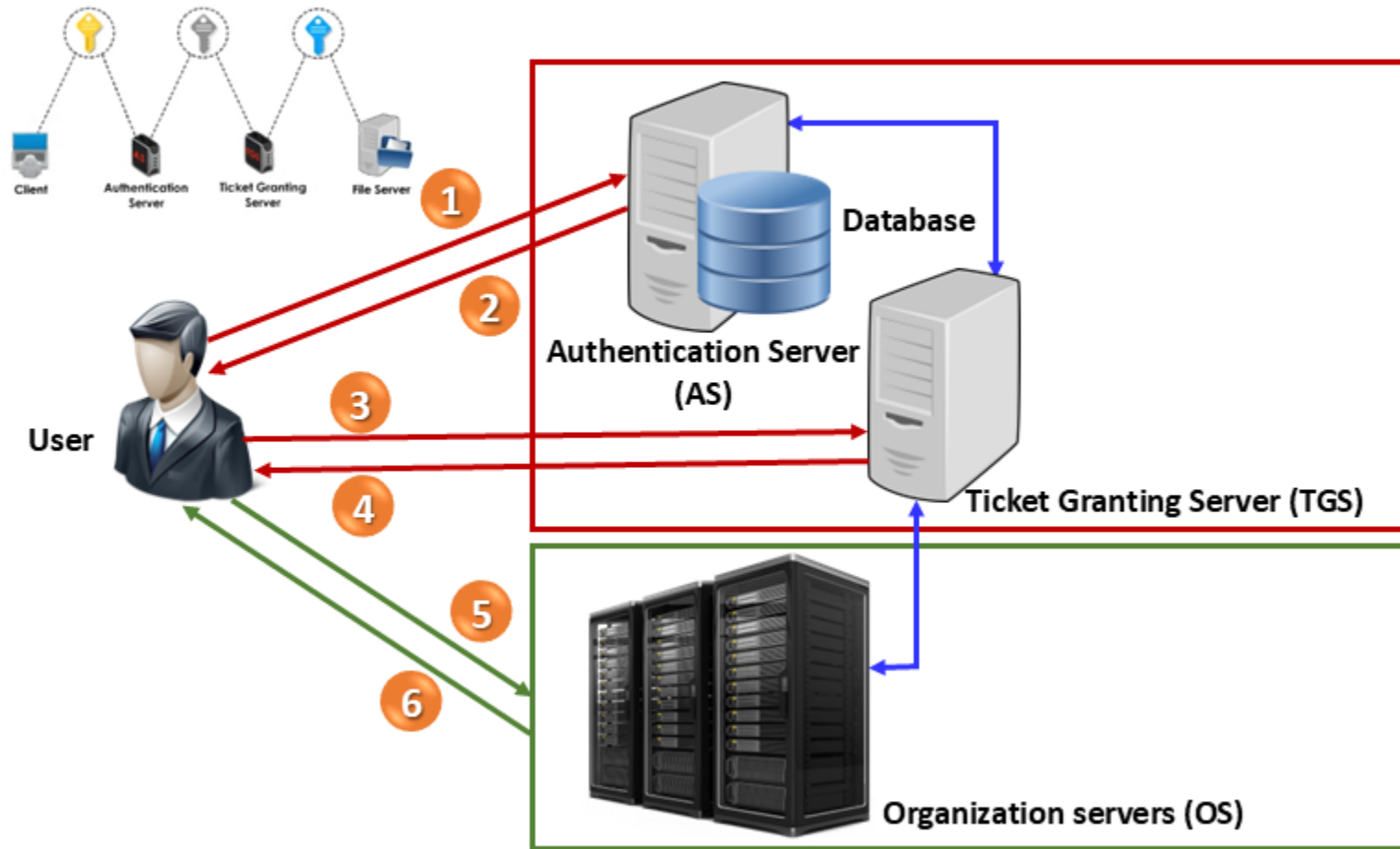


Public Key Infrastructure (PKI)

Transport Layer Security & Secure Socket Layer (TLS/SSL)



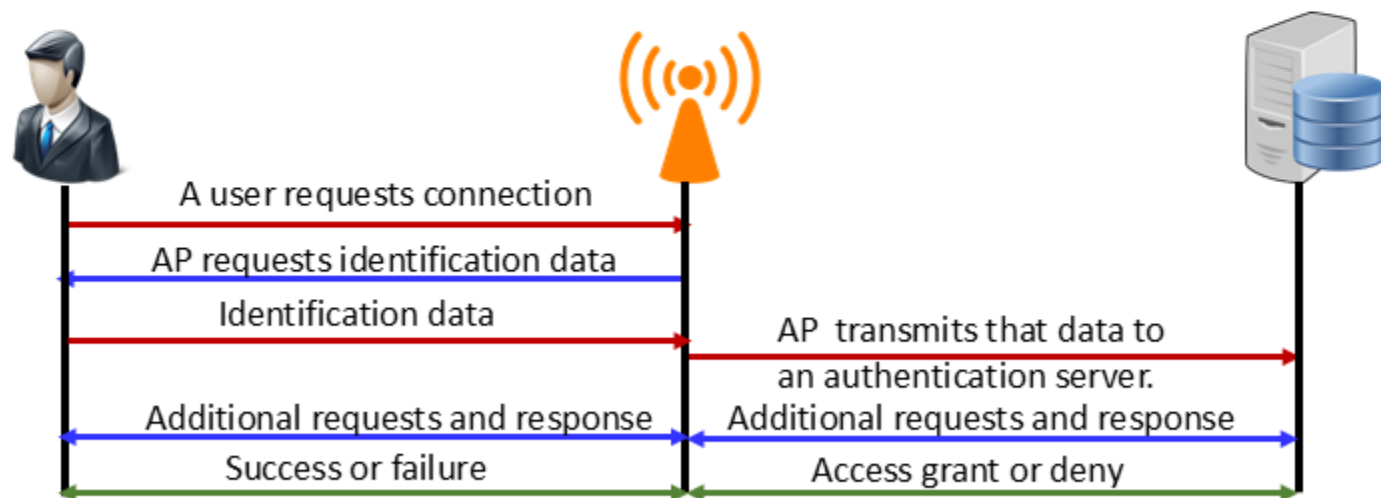
- An authentication protocol is defined as a computer system communication protocol which designed specifically to securely transfer authenticated data between two parties:
 - ***Password Authentication Protocol (PAP)*** is an authentication protocol that transmits data (passwords) in plain readable text as a single readable file.
 - ***Shiva Password Authentication Protocol (SPAP)*** is a proprietary form of **PAP** that offers more security than PAP. With SPAP, passwords are processed through a reversible encryption system.
 - ***Challenge Handshake Authentication Protocol (CHAP)*** is an industry standard communication protocol that uses the **MD5** Hashing scheme for authentication.
 - ***Time-Based One-Time Password (TOTP)*** is an authentication protocol that uses an algorithm to generate what is called a temporary passcode used for authenticating access to any system. This passcode is changed every 40 to 60 seconds.



1	User login and requests for ticket-granting service, the request is encrypted using the user password.
2	Authentication Server decrypts the user request using the saved user password and verifies user's access right using database and then gives the user ticket-granting-ticket (TGT) encrypted with a secret key.
3	User send (TGT) to Ticket Granting Server which decrypts TGT using the secret key from AS.
4	TGS sends the user a token encrypted with another secret key shared with (OS).
5	The user sends the token to the (OS) which decrypt it with the secret key that shard with TGS.
6	OS grants access to the user.

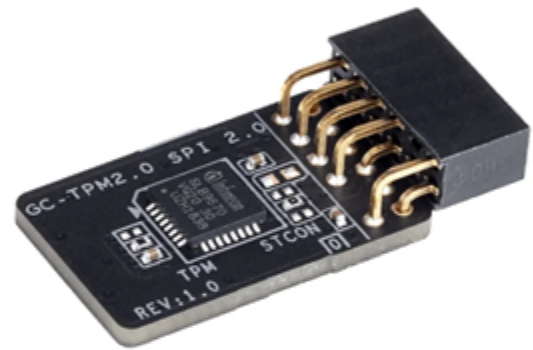
- **Internet Protocol Security (IPSec)** encryption is one part of the protocol suite that allows the information to be securely transmitted and received across a network often used in setting up *virtual private networks (VPNs)*. IPSec uses authentication (verification of identity and/or origin) and encryption (scrambling a message so that it has to be decoded to be understood) to send data across a network via:
 - **Authentication headers (AH)**; which ensure the integrity and verify the origin (source) of the data (verifying user identity).
 - **Encapsulating Security Payload (ESP)**; which works to ensure data confidentiality (protecting the data).

- **Extensible Authentication Protocol (EAP)** is a protocol for wireless networks that expands the authentication methods between the supplicant (the Wi-Fi workstation) and the authentication server (Microsoft IAS or other). There are several authentication types of EAP include EAP-MD5, EAP-TLS, EAP-PEAP, EAP-TTLS, and EAP-FAST.



Some various encryption tools that are employed to protect our enterprise networks and systems. These encryption tools include:

- **Trusted Platform Module (TPM)** is a dedicated microcontroller that's designed to *secure hardware through integrated cryptographic keys*. One of the common uses of the trusted platform module is with the BitLocker drive encryption feature found inside of your Windows operating system.
- **Hardware Security Module (HSM)** is a physical device that safeguards and *generating, protecting, and managing keys used for encrypting and decrypting data and creating digital signatures and certificates*, and is primarily used for mission critical situations like financial transactions.



- **Key Management System (KMS)** is an integrated approach for *generating, distributing, and managing cryptographic keys for devices and applications*. The key management system is a centralized mechanism that controls the entire lifecycle of these cryptographic keys, from creation to destruction.



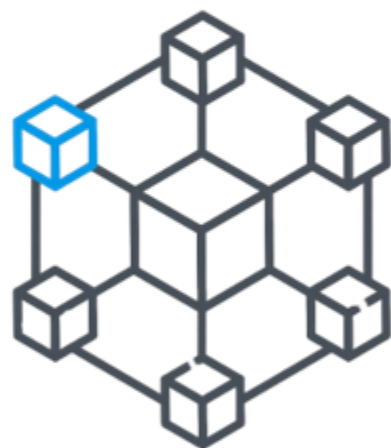
- **Secure Enclaves** is a *co-processor that's integrated into the main processor* of some devices, and it's designed with the sole purpose of *ensuring data protection*.

- **Steganography** is the technique of *hiding data within an ordinary, non secret file or message to avoid detection*; the hidden data is then extracted at its destination. It is frequently used alongside encryption for an extra layer of security (try this site “<https://stylesuxx.github.io/steganography/>”).
- **Tokenization** is the process of *replacing sensitive data with nonsensitive equivalents called tokens (unique identification symbols)* that retain all the essential information about the data without compromising its security. Tokenization is often used to protect credit card data, bank account information and other sensitive data handled by payment processors.
- **Data Masking** is used to *protect our data by ensuring that it remains recognizable* but does not actually include sensitive information.

[illegible]

Ledger Book

Any deposit, withdrawal, or transfer of money in the bank is recorded in a digital Ledger.



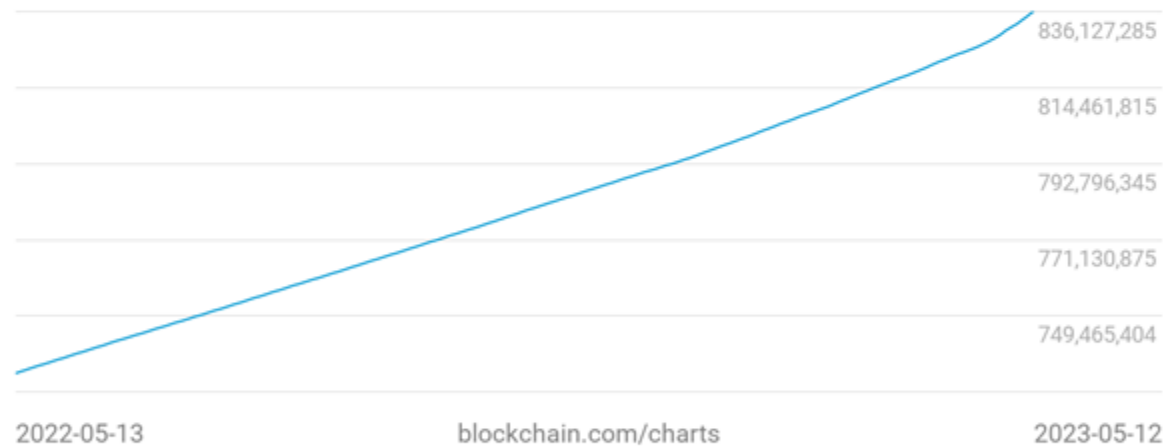
A **blockchain** is a constantly growing ledger that keeps a permanent record of all the transactions that have taken place in a secure, chronological and immutable way.

4

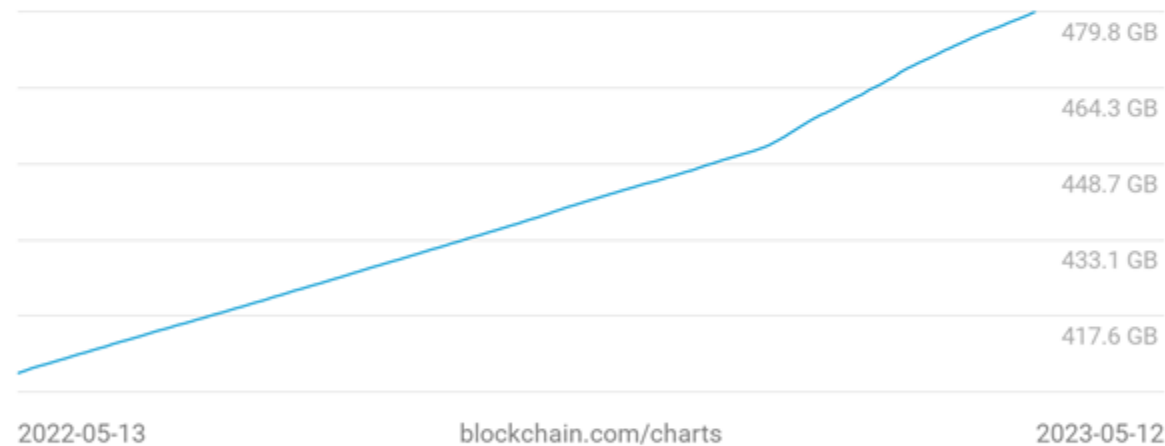
5

Constantly growing ledger

Total Number of Transactions
836,265,458



Blockchain Size
479.9 GB



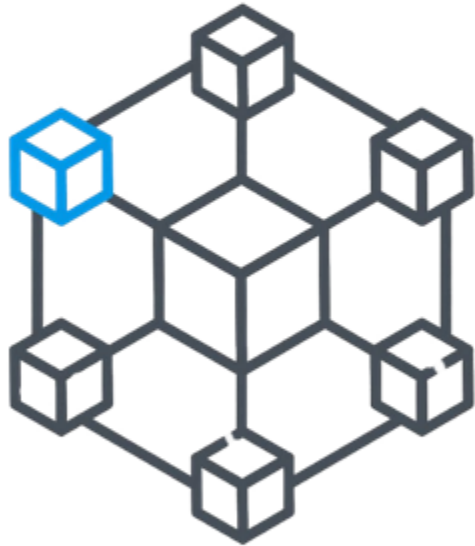
Permanent record of all the transactions

Advanced Details

Hash	4a5e-a33b 6	Time	03 Jan 2009 08:15:05
Age	14y 4m 9d 0h 31m 18s	Inputs	1
Input Value	—	Outputs	1
	\$0.00	Output Value	50.00000000 BTC
Fee	0 BTC		\$1,342,912
	\$0.00	Fee/B	-
Fee/VB	-	Size	204 Bytes
Weight	816	Weight Unit	-
Coinbase	Yes	Witness	No
RBF	No	Locktime	0
Version	1	BTC Price	\$26,858.24



Secure



REACHABLE BITCOIN NODES

Updated: Thu May 25 19:47:55 2023 EEST

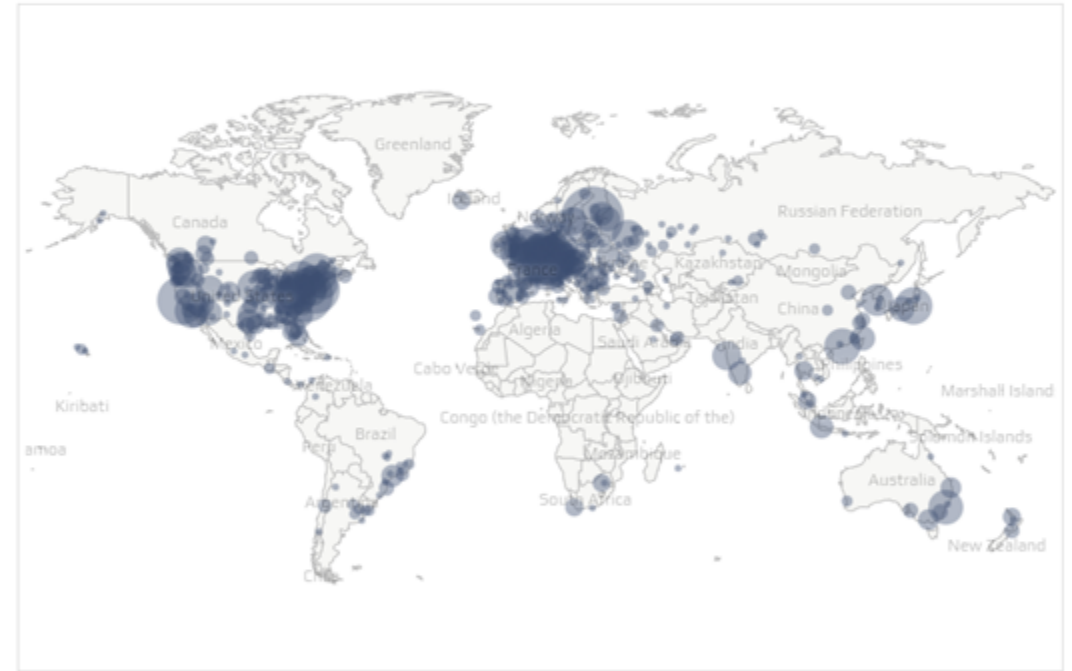
17258 NODES

CHARTS

IPv4: -2.5% / IPv6: -0.6% / .onion: +4.8%

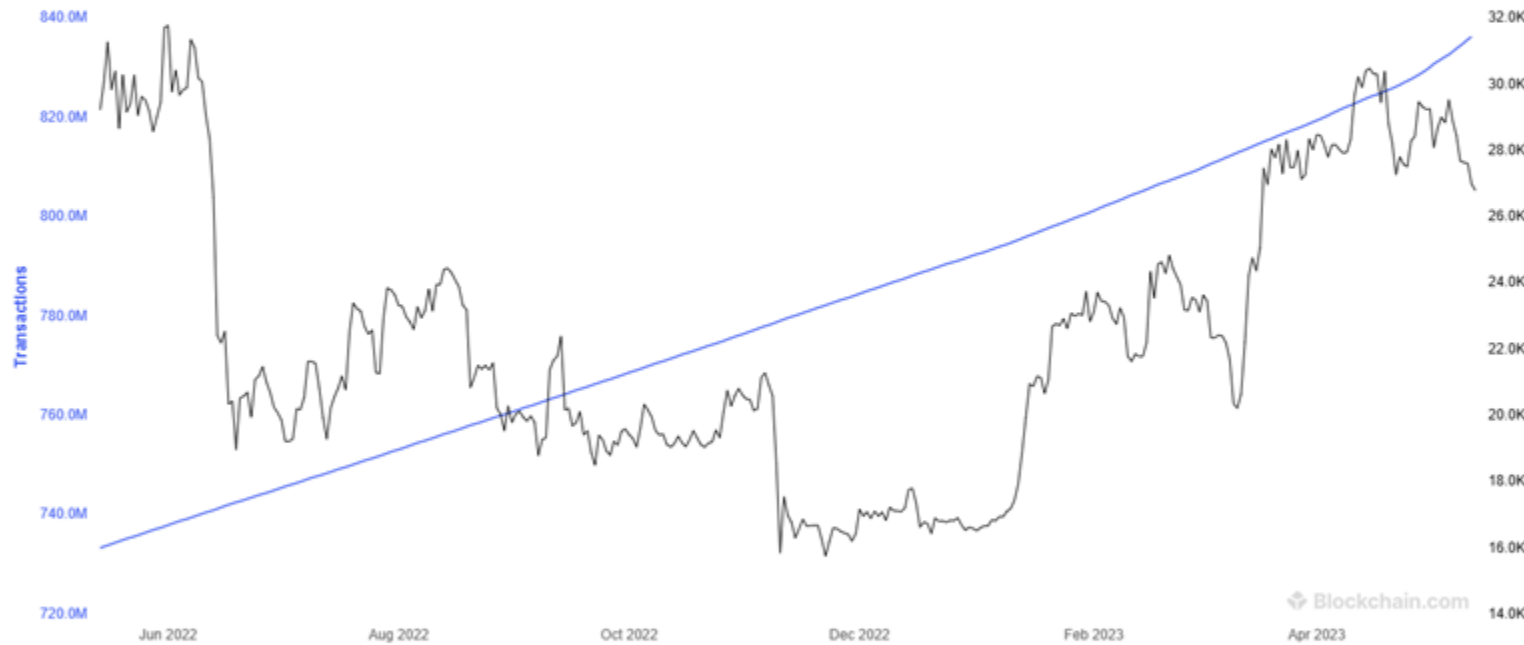
Top 10 countries with their respective number of reachable nodes are as follows.

RANK	COUNTRY	NODES
1	n/a	10617 (61.52%)
2	United States	1653 (9.58%)
3	Germany	1369 (7.93%)
4	France	476 (2.76%)
5	Netherlands	341 (1.98%)
6	Canada	301 (1.74%)
7	Finland	258 (1.49%)
8	United Kingdom	203 (1.18%)
9	Russian Federation	177 (1.03%)
10	Switzerland	160 (0.93%)



Map shows concentration of reachable Bitcoin nodes found in countries around the world.

LIVE MAP

Chronological

Hash 99c3-653a 5/13/2023, 21:06:33	0.01171698 BTC \$314.19
Hash 7172-d726 5/13/2023, 21:06:33	1.13369311 BTC \$30,399.64
Hash 56ea-bfac 5/13/2023, 21:06:33	0.00335752 BTC \$90.03
Hash b218-b1a4 5/13/2023, 21:06:33	0.99359761 BTC \$26,643.02
Hash 1409-ca15 5/13/2023, 21:06:33	1.21030241 BTC \$32,453.90
Hash d6aa-dde5 5/13/2023, 21:06:33	0.00315612 BTC \$84.63
Hash cbae-dd68 5/13/2023, 21:06:33	0.00266936 BTC \$71.58
Hash 8ed1-04d7 5/13/2023, 21:06:33	0.00002897 BTC \$0.78
Hash e2e5-d18e 5/13/2023, 21:06:33	0.01898831 BTC \$509.17
Hash a54d-5074 5/13/2023, 21:06:33	0.00096292 BTC \$25.82
Hash e984-495b 5/13/2023, 21:06:33	0.00002869 BTC \$0.77

Immutable