**CRYPTOGRAPHY**

**Introduction:**

Cryptography is a method of protecting information and communications through the use of codes, so that only those for whom the information is intended can read and process it.

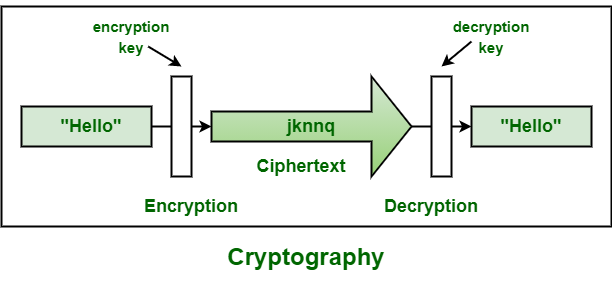
In computer science, cryptography refers to secure information and communication techniques derived from mathematical concepts and a set of rule-based calculations called algorithms, to transform messages in ways that are hard to decipher. These deterministic algorithms are used for cryptographic key generation, digital signing, verification to protect data privacy, web browsing on the internet and confidential communications such as credit card transactions and email.

**Cryptography Techniques:**

Cryptography is closely related to the disciplines of cryptology and cryptanalysis. It includes techniques such as microdots, merging words with images and other ways to hide information in storage or transit. However, in today's computer-centric world, cryptography is most often associated with scrambling plaintext (ordinary text, sometimes referred to as clear-text) into cipher-text (a process called encryption), then back again (known as decryption). Individuals who practice this field are known as cryptographers.

Modern cryptography concerns itself with the following four objectives:

1. Confidentiality: The information cannot be understood by anyone for whom it was unintended.
2. Integrity: The information cannot be altered in storage or transit between sender and intended receiver without the alteration being detected.
3. Non-repudiation: The creator/sender of the information cannot deny at a later stage their intentions in the creation or transmission of the information.
4. Authentication: The sender and receiver can confirm each other's identity and the origin/destination of the information.



**Cryptographic Algorithms:**

Cryptosystems use a set of procedures known as cryptographic algorithms, or ciphers, to encrypt and decrypt messages to secure communications among computer systems, devices and applications.

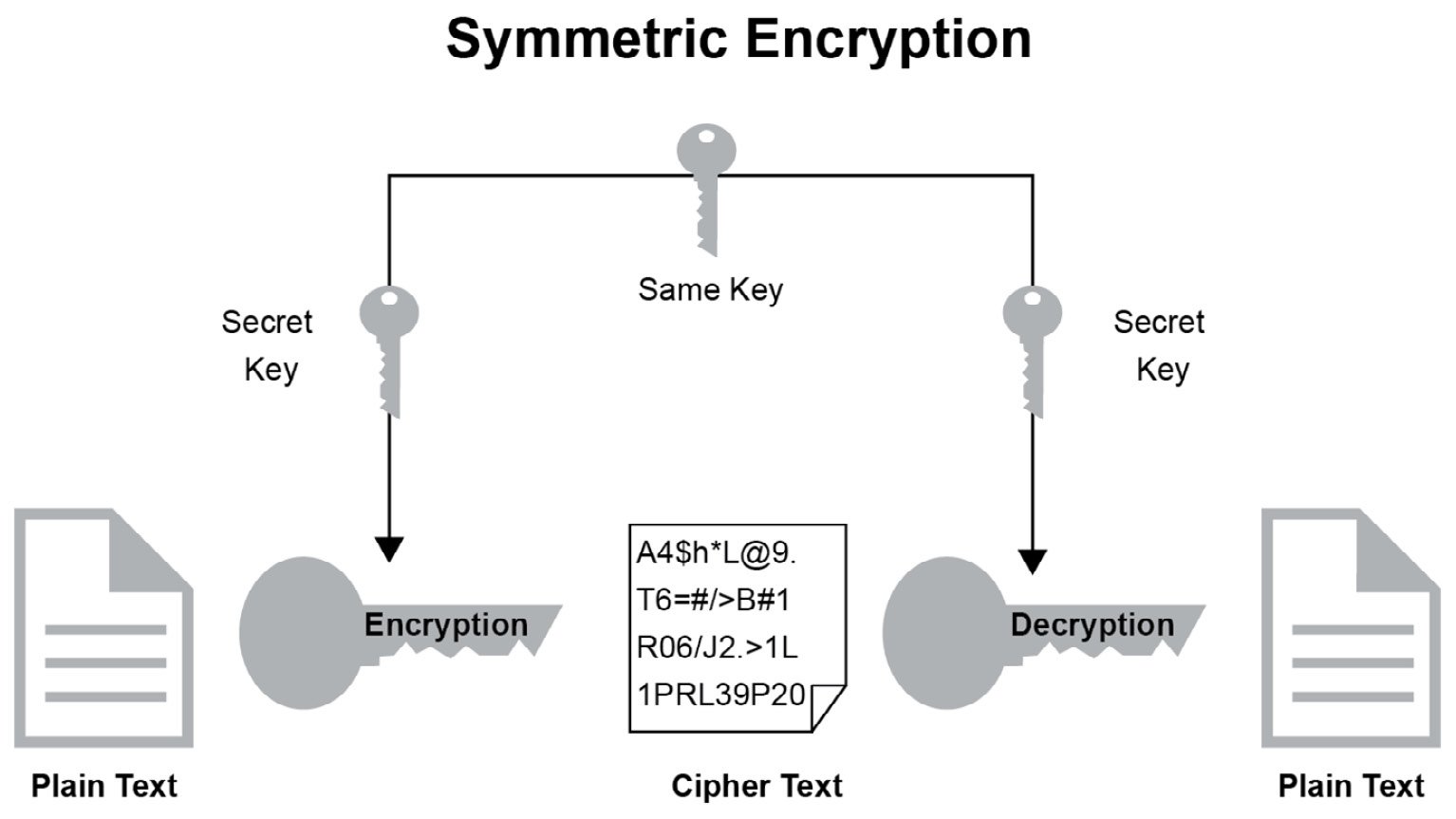
A cipher suite uses one algorithm for encryption, another algorithm for message authentication and another for key exchange. This process, embedded in protocols and written in software that runs on operating systems (OSes) and networked computer systems, involves:

* public and private key generation for data encryption/decryption
* digital signing and verification for message authentication
* key exchange

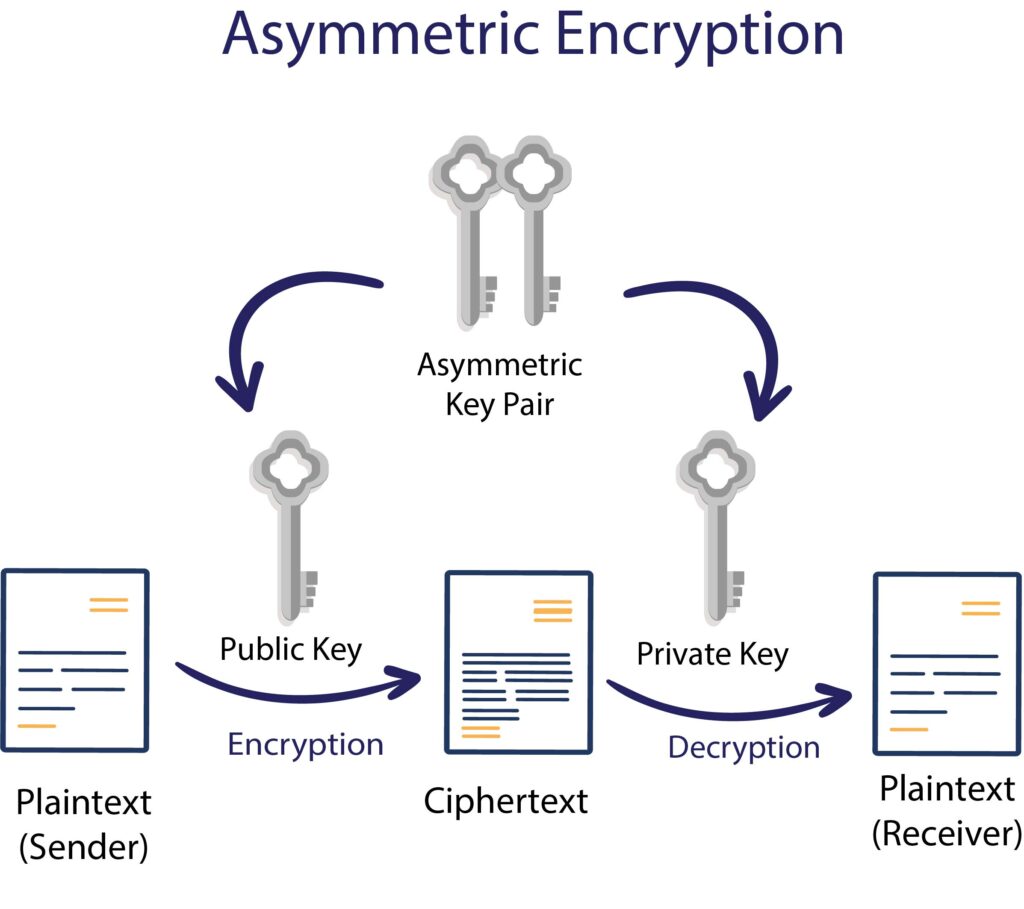
**Types of Cryptography:**

Single-key or symmetric-key encryption algorithms create a fixed length of bits known as a block cipher with a secret key that the creator/sender uses to encipher data (encryption) and the receiver uses to decipher it. One example of symmetric-key cryptography is the Advanced Encryption Standard (AES). AES is a specification established in November 2001 by the National Institute of Standards and Technology (NIST) as a Federal Information Processing Standard (FIPS 197) to protect sensitive information. The standard is mandated by the U.S. government and widely used in the private sector.

In June 2003, AES was approved by the U.S. government for classified information. It is a royalty-free specification implemented in software and hardware worldwide. AES is the successor to the Data Encryption Standard (DES) and DES3. It uses longer key lengths 128bit, 192bit, 256bit to prevent brute force and other attacks.



Public-key or asymmetric-key encryption algorithms use a pair of keys, a public key associated with the creator/sender for encrypting messages and a private key that only the originator knows (unless it is exposed or they decide to share it) for decrypting that information.



Examples of public-key cryptography include:

* RSA, used widely on the internet
* Elliptic Curve Digital Signature Algorithm (ECDSA) used by Bitcoin
* Digital Signature Algorithm (DSA) adopted as a Federal Information Processing Standard for digital signatures by NIST in FIPS 186-4
* Diffie-Hellman key exchange

To maintain data integrity in cryptography, hash functions, which return a deterministic output from an input value, are used to map data to a fixed data size. Types of cryptographic hash functions include SHA-1 (Secure Hash Algorithm 1), SHA-2 and SHA-3.

**Cryptography Tools:**

Technology is getting advanced day by day, and this fact could not be belied. A few years back, the popularity of the internet was as much as it’s not. The advancement in technology that could be witnesses these days is the outcome of the internet’s availability that influenced the researchers to bring new things into the technology’s bucket. With the growth in technology, the number of breaches happening online has also increased, which demanded secure approaches to run operations online. This article will learn about a critical module of cybersecurity that helps to establish information security by encrypting the data.

**Some of the Cryptographic Tools are:**

1. **Security Token/Authentication Token:**

The security token or the authentication token is the one that is considered as the cryptography tool. Using the security token, one can authenticate the user. It is also used to provide state-fullness to the HTTP protocol. The security token has to be encrypted to allow the secure exchange of data. The browser uses the security token that is drafted at the server-side to maintain the state. In normal words, it could be considered as a mechanism that allows remote authentication.

1. **CertMgr.exe:**

This is the tool that Microsoft defines as cryptography tools. The file can be installed as it has the extension. The file allows you to create and manage the certificate. Cryptography is used extensively in certificate managing. It lets you handle the certification, and in addition to that, it also allows you to manage the certificate revocation lists (CRLs). The purpose of cryptography in certificate generation is to ensure that the parties’ data is secure. This tool helps us add the extra bits in security.

1. **JCA:**

The JCA is the other kind of cryptography tool that helps us to endorse encryption. JCA can be defined as the cryptographic libraries in Java. The libraries consist of predefined functions that should be imported before their features could be leveraged. Being a Java library, it works perfectly with the other Java frameworks and helps in application development. The application with the urge of cryptography uses this library to build the application if the development has been carried out using Java.

1. **Docker:**

The other cryptography tool is used to practice encryption. The Docker is nothing but a container that allows the user to develop a large application within it. The data that is kept inside the docket as kept in encrypted form. The mechanism of cryptography has been used very deeply to bring the feature of data encryption. The interesting part about Docker is, in addition to the data, the files are also kept encrypted so that no one without proper key access could access it. In simple terms, the Docker is a container that lets the user hold their critical information and kept it safe so that the entire focus can be on application development. It can also be considered as the cloud storage that lets us handle the data on a shared or dedicated server.

1. **SignTool.exe:**

The signing tool is one of the more important cryptography tools that Microsoft uses to sign the file. This application works as the authority that can add a signature with any of the files. In addition to that, it also adds the time stamp with the file. The timestamp associated with the file can be used for verifying the file. Overall, all of the features provided by this tool is to ensure the integrity of the file. In a nutshell, it is a Microsoft based product that lets the user sign the file and also let them add the timestamp with the file.

1. **Authentication Using Key:**

The data that is encrypted needs the key to decrypt it. The normal data is something that could be understood by the normal user, while the encrypted data is different from the original data. In Authentication using Key, There are two types of approaches in Encryption.

* Symmetric Encryption
* Asymmetric Encryption

In symmetric encryption, the key used to encrypt the data is the same key that will be used to decrypt the data. In asymmetric encryption, the data will be encrypted with one key while it uses another key to get it decrypted. The keys are considered as private and public key based on which key is used for what purpose.

**Algorithms:**

The cryptography algorithms include the following.

In this IoT domain, security matters the most. Though there are many security mechanisms in practice, they do not hold the ability to come up with current day smart applications mainly for the software operating with resource-constraint equipment. In a consequence of this, cryptography algorithms came into practice ensuring enhanced security.

So, few of the cryptographic algorithms are as follows:

* **Triple DES:**

Taking over from the conventional DES mechanism, triple DES was currently implemented in the security approaches. These algorithms permit hackers to ultimately gain the knowledge to overcome in an easy approach. This was the extensively implemented approach by many of the enterprises. Triple DES operates with 3 keys having 56 bits per each key. The entire key length is a maximum of bits, whereas experts would contend that 112-bits in key intensity is more probable. This algorithm handles to make a reliable hardware encryption answer for banking facilities and also for other industries.

* **Blowfish:**

To replace the approaches of Triple DES, Blowfish was mainly developed. This encryption algorithm split up messages into clocks having 64 bits and encrypts these clocks separately. The captivating feature that lies in Blowfish is its speed and efficacy. As this is an open algorithm for everyone, many gained the benefits of implementing this. Every scope of the IT domain ranging from software to e-commerce is making use of this algorithm as it shows extensive features for password protection. All these allow this algorithm to be most prominent in the market.

* **RSA:**

One of the public-key encryption algorithms used to encrypt information transmitted through the internet. It was a widely used algorithm in GPG and PGP methodologies. RSA is classified under symmetric type of algorithms as it performs its operation using a couple of keys. One of the keys is used for encryption and the other for decryption purposes.

* **Twofish:**

This algorithm implements keys to provide security and as it comes under the symmetric method, only one key is necessary. The keys of this algorithm are with the maximum length of 256 bits. Of the most available algorithms, Twofish is mainly known by its speed and perfect to be implemented both in the hardware and software applications. Also, it is an openly accessible algorithm and has been in execution by many.

* **Advanced Encryption System (AES):**

This is the most trusted algorithm technique by U.S administration and many other enterprises. Even though this works efficiently in 128-bit encryption form, 192 and 256 bits are mainly used for huge encryption activities. Being so invulnerable to all hacking systems, the AES technique receives extensive applause for encrypting information in the private domain.

**Applications of Cryptography:**

Applications for cryptography as below.

Conventionally, cryptography was in implementation only for securing purposes. Wax seals, hand signatures and few other kinds of security methods were generally utilized to make sure of reliability and accuracy of the transmitter. And with the arrival of digital transmissions, security becomes more essential and then cryptography mechanisms began to outstrip its utilization for maintaining utmost secrecy.

A few of the applications of cryptography are discussed below.

* **To Maintain Secrecy in Storage:**

Cryptography allows storing the encrypted data permitting users to stay back from the major hole of circumvention by hackers.

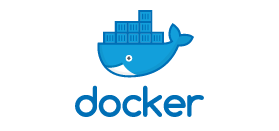
#### Reliability in Transmission:

A conventional approach that allows reliability is to carry out a checksum of the communicated information and then communicate the corresponding checksum in an encrypted format. When both the checksum and encrypted data is received, the data is again check-summed and compared to the communicated checksum after the process of decryption. Thus, effective cryptographic mechanisms are more crucial to assure reliability in message transmission.

* **Authentication of Identity:**

Cryptography is strongly linked to the approach of using passwords, and innovative systems probably make use of strong cryptographic methods together with the physical methods of individuals and collective secrets offering highly reliable verification of identity.

**DOCKER**



Docker is an open source platform for building, deploying, and managing containerized applications.

It enables developers to package applications into containers standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment. Containers simplify delivery of distributed applications, and have become increasingly popular as organizations shift to cloud-native development and hybrid multi-cloud environments.

Developers can create containers without Docker, but the platform makes it easier, simpler, and safer to build, deploy and manage containers. Docker is essentially a toolkit that enables developers to build, deploy, run, update, and stop containers using simple commands and work-saving automation through a single API.

**Container’s working:**

Containers are made possible by process isolation and virtualization capabilities built into the Linux kernel. These capabilities - such as control groups (C-groups) for allocating resources among processes, and namespaces for restricting a processes access or visibility into other resources or areas of the system - enable multiple application components to share the resources of a single instance of the host operating system in much the same way that a hypervisor enables multiple virtual machines (VMs) to share the CPU, memory and other resources of a single hardware server.

As a result, container technology offers all the functionality and benefits of VMs - including application isolation, cost-effective scalability, and disposability - plus important additional advantages:

* **Lighter weight:**

Unlike VMs, containers don’t carry the payload of an entire OS instance and hypervisor; they include only the OS processes and dependencies necessary to execute the code. Container sizes are measured in megabytes (vs. gigabytes for some VMs), make better use of hardware capacity, and have faster start-up times.

* **Greater resource efficiency:**

With containers, you can run several times as many copies of an application on the same hardware as you can using VMs. This can reduce your cloud spending.

* **Improved developer productivity:**

Compared to VMs, containers are faster and easier to deploy, provision and restart. This makes them ideal for use in continuous integration and continuous delivery (CI/CD) pipelines and a better fit for development teams adopting Agile and DevOps practices.

**Why Docker?**

Docker is so popular today that “Docker” and “containers” are used interchangeably. But the first container-related technologies were available for years — even decades (link resides outside IBM) — before Docker was released to the public in 2013.

Most notably, in 2008, Linux-Containers (LXC) was implemented in the Linux kernel, fully enabling virtualization for a single instance of Linux. While LXC is still used today, newer technologies using the Linux kernel are available. Ubuntu, a modern, open-source Linux operating system, also provides this capability.

Docker enhanced the native Linux containerization capabilities with technologies that enable:

* **Improved and seamless portability:**

While LXC containers often reference machine-specific configurations, Docker containers run without modification across any desktop, data centre and cloud environment.

* **Even lighter weight and more granular updates:**

With LXC, multiple processes can be combined within a single container. With Docker containers, only one process can run in each container. This makes it possible to build an application that can continue running while one of its parts is taken down for an update or repair.

* **Automated container creation:**

Docker can automatically build a container based on application source code.

* **Container versioning:**

Docker can track versions of a container image, roll back to previous versions, and trace who built a version and how. It can even upload only the deltas between an existing version and a new one.

* **Container reuse:**

Existing containers can be used as base images—essentially like templates for building new containers.

* **Shared container libraries:**

Developers can access an open-source registry containing thousands of user-contributed containers.

Today Docker containerization also works with Microsoft Windows server. And most cloud providers offer specific services to help developers build, ship and run applications containerized with Docker.

**Docker Tool and Terms**

Some of the tools and terminology you’ll encounter when using Docker include:

**DockerFile:**

Every Docker container starts with a simple text file containing instructions for how to build the Docker container image. DockerFile automates the process of Docker image creation. It’s essentially a list of command-line interface (CLI) instructions that Docker Engine will run in order to assemble the image.

**Docker Images:**

Docker images contain executable application source code as well as all the tools, libraries, and dependencies that the application code needs to run as a container. When you run the Docker image, it becomes one instance (or multiple instances) of the container.

It’s possible to build a Docker image from scratch, but most developers pull them down from common repositories. Multiple Docker images can be created from a single base image, and they’ll share the commonalities of their stack.

Docker images are made up of layers, and each layer corresponds to a version of the image. Whenever a developer makes changes to the image, a new top layer is created, and this top layer replaces the previous top layer as the current version of the image. Previous layers are saved for rollbacks or to be re-used in other projects.

Each time a container is created from a Docker image, yet another new layer called the container layer is created. Changes made to the container such as the addition or deletion of files are saved to the container layer only and exist only while the container is running. This iterative image-creation process enables increased overall efficiency since multiple live container instances can run from just a single base image, and when they do so, they leverage a common stack.

**Docker Containers:**

Docker containers are the live, running instances of Docker images. While Docker images are read-only files, containers are live, ephemeral, executable content. Users can interact with them, and administrators can adjust their settings and conditions using Docker commands.

**Docker Hub:**

Docker Hub is the public repository of Docker images that calls itself the “world’s largest library and community for container images.” It holds over 100,000 container images sourced from commercial software vendors, open-source projects, and individual developers. It includes images that have been produced by Docker, Inc., certified images belonging to the Docker Trusted Registry, and many thousands of other images.

All Docker Hub users can share their images at will. They can also download predefined base images from the Docker file system to use as a starting point for any containerization project.

**Docker Daemon:**

Docker daemon is a service running on your operating system, such as Microsoft Windows or Apple Mac-OS or iOS. This service creates and manages your Docker images for you using the commands from the client, acting as the control centre of your Docker implementation.

**Docker Registry:**

A Docker registry is a scalable open-source storage and distribution system for Docker images. The registry enables you to track image versions in repositories, using tagging for identification. This is accomplished using git, a version control tool.

**Docker deployment and orchestration**

If you’re running only a few containers, it’s fairly simple to manage your application within Docker Engine, the industry de facto runtime. But if your deployment comprises thousands of containers and hundreds of services, it’s nearly impossible to manage that workflow without the help of these purpose-built tools.

**Docker Compose:**

If you’re building an application out of processes in multiple containers that all reside on the same host, you can use Docker Compose to manage the application’s architecture. Docker Compose creates a YAML file that specifies which services are included in the application and can deploy and run containers with a single command. Using Docker Compose, you can also define persistent volumes for storage, specify base nodes, and document and configure service dependencies.

**Kubernetes:**

To monitor and manage container lifecycles in more complex environments, you’ll need to turn to a container orchestration tool. While Docker includes its own orchestration tool (called Docker Swarm), most developers choose Kubernetes instead.

Kubernetes is an open-source container orchestration platform descended from a project developed for internal use at Google. Kubernetes schedules and automates tasks integral to the management of container-based architectures, including container deployment, updates, service discovery, storage provisioning, load balancing, health monitoring, and more. In addition, the open source ecosystem of tools for Kubernetes including Istio and Knative enables organizations to deploy a high-productivity Platform as a Service (PaaS) for containerized applications and a faster on-ramp to server-less computing.

