Data encryption

So far, you've learned that data encryption is the process of converting data from a readable format to an encoded format. In this reading, you'll dive deeper into data encryption, and learn about Advanced Encryption Standard (AES) encryption and Rivest-Shamir-Adleman (RSA) encryption. Then, you'll explore encryption at rest and encryption in transit. You'll also explore examples of different types of encryption.

Data encryption basics

Data encryption is the process of converting data from plaintext (readable data) to ciphertext (encoded data). This process is completed using mathematical cryptographic algorithms. A **readable** format is data that's human-readable and written in plaintext. An **encoded** format is data that's transformed into a sequence of text and symbols that won't make logical sense to a reader.

Remember, encryption protects data from being changed, compromised, or stolen. Encryption scrambles data into a secret code that can only be unlocked using a unique digital key. This type of key is known as an encryption key. An encryption key is a sequence of bits or characters used by an encryption algorithm to scramble and unscramble data. Encrypted data is protected while at-rest or in-transit, or while being processed.

When using cloud services, the party responsible for encrypting data varies. If you perform encryption yourself before sending data to the cloud, that's called client-side encryption. With client-side encryption, you're responsible for creating and managing your encryption keys before you send your data to the cloud. In server-side encryption, the cloud provider is responsible for managing encryption keys.

If encryption methods are secure, they use such a large number of potential keys, so that the number of possibilities is too large for attackers to guess the correct sequence. This also makes it difficult for attackers to try every possible combination and successfully calculate the correct string of characters to perform a brute force attack. An encryption key is like a password. When the length of a password is long, and the characters used are complex, it's more difficult for the attacker to guess every character correctly and crack the password.

Encryption has four functions:

1. **Confidentiality:** Keep the data's contents secret
2. **Authenticity:** Verify the data's origin
3. **Integrity:** Validate the content of the data, and ensure it hasn't been altered since it was sent

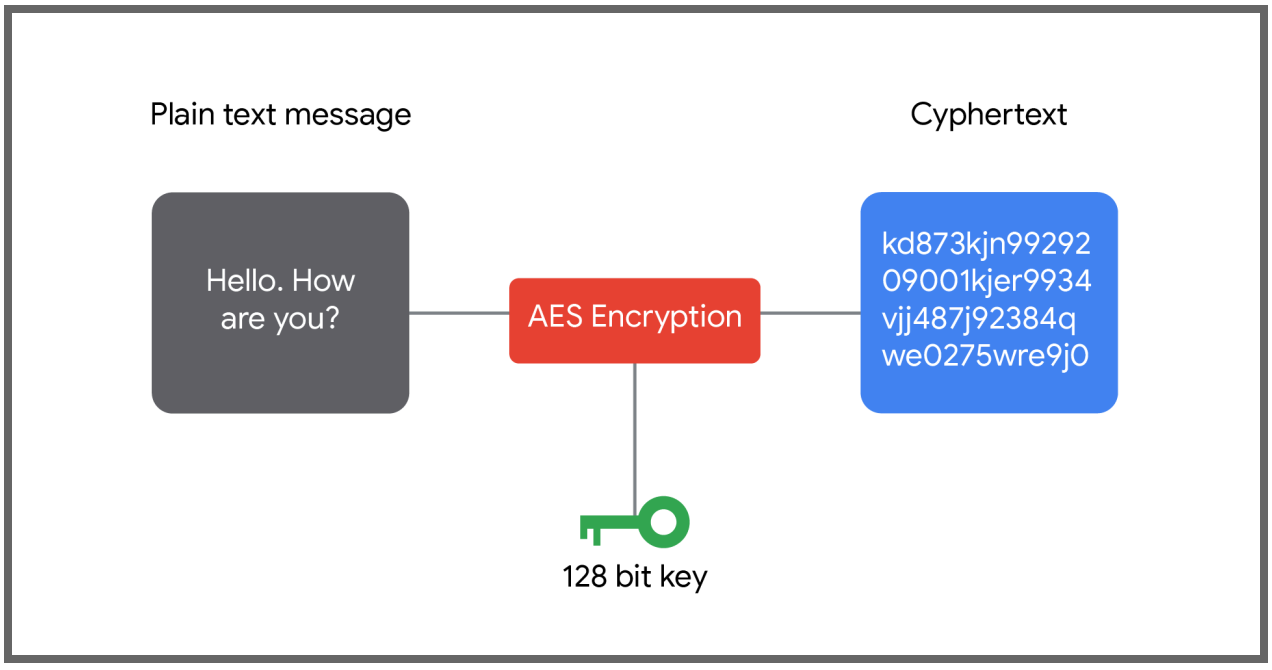
4. **Nonrepudiation:** Stop the data's sender from denying they were the origin of the data (This does not apply to symmetric encryption because anyone who has the key can encrypt.)

Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) was adopted by the United States government in 2001, and is a commonly used encryption standard still in use. It was designed on a block cipher of 128 bits, and can have key bit lengths of 128, 192, or 256 bits. The block cipher principle it was designed on is called a substitution-permutation network (SPN). SPN takes blocks of plaintext and keys, and applies alternating rounds of permutation and substitution layers. This is how it produces the final ciphertext. AES requires less memory than other encryption types, allows for fast encryption and decryption, and can be implemented easily.

AES is also symmetric. It's used for passwords, online banking credentials, and password-protected files that need to be encrypted. It works with many of the technologies people use every day, including Wi-Fi, mobile apps, tools used for archives and compression, password managers, operating system components, and programming language libraries like the libraries used for Python.

Here's an example. You use a mobile messaging application from a social network to privately send a photo and a message to your friend. This is something you don't want to share with the public. When you send the photo and message, they're encrypted using AES encryption.



Example of data encryption in use

Here's an example of data encryption in use:

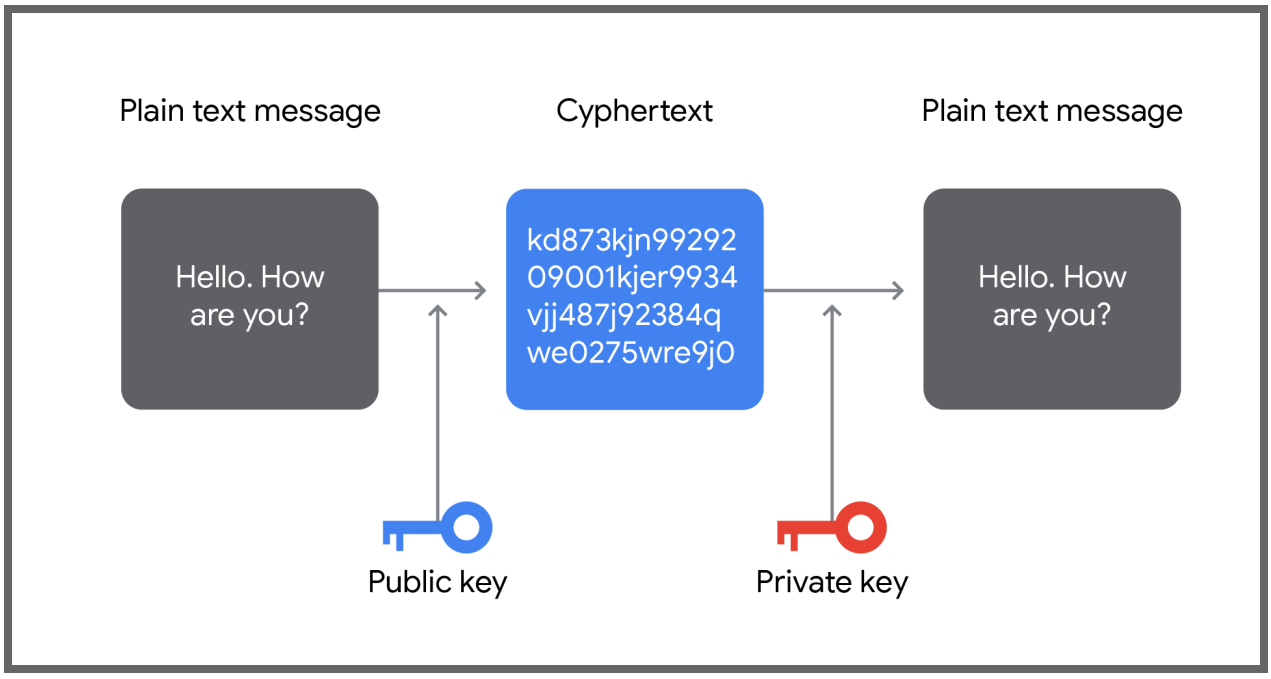
1. A user starts a web browser and requests a page with a hypertext transfer protocol secure (HTTPS) prefix.
2. The user's web browser contacts the server on the server's port.
3. The server responds with its Transport Layer Security (TLS) or SSL certificate.
4. The user's browser takes the certificate and uses it to verify the remote server's identity, and to take the remote server's public key.
5. The browser creates a session key and encrypts it with the public key it got from the server in its TLS/SSL certificate.
6. The server decrypts the session key with its private key.
7. The user's web browser, the client, uses the session key and encrypts all communications.

Rivest-Shamir-Adleman (RSA) encryption standard

The Rivest-Shamir-Adleman (RSA) encryption standard was created by Ron Rivest, Adi Shamir, and Leonard Adleman, three MIT researchers who introduced the method in 1977.

Here's how RSA operates:

- RSA creates a key by factoring two prime numbers and an auxiliary value.
- Keys made with RSA can be large. Typical sizes are 2,048 or 4,096 bits.
- RSA is a form of asymmetric encryption. One of the key differences between symmetric and asymmetric encryption is that asymmetric encryption uses a pair of keys, a private key and a public key. The private key stays with one party while the public key can be distributed.
- RSA is slower than AES, and it uses more resources.
- It's one of the original forms of asymmetric encryption, but it's still widely in use.



Some advantages of data encryption

Data encryption has a few advantages, including:

- It protects data across devices when the data moves between devices or servers. For example, when you use a banking app, it keeps data secure from attackers.
- It ensures data integrity, so malicious actors can't use the data to commit extortion or fraud, or alter important documents.
- It protects digital transformations. Data encryption protects data while it's at-rest on the server, in-transit to the cloud, or while it's being processed by workloads.
- It helps you meet the requirements for compliance.

Some challenges of data encryption

Encryption has many benefits, but it also has some challenges, including ransomware attacks and key management. In a ransomware attack, data is held hostage. Encryption usually protects data, but malicious actors can hold data hostage with it. These attackers can hold the hostage data for ransom by encrypting it and forcing the organization to pay for its release.

If the keys that encrypt and decrypt data are not secure, malicious actors can concentrate their attacks on getting the keys. Also if the organization loses encryption keys in a disaster or from an attack, the data will be locked away from that organization. That's why it's important to have a secure key management system.

Key takeaways

Encryption is important for keeping data safe. It protects data from being changed, compromised or stolen. Encryption can be used to provide confidentiality, authenticity, integrity, and nonrepudiation. There are several common encryption standards, including (AES) and (RSA). AES is a symmetric encryption standard that's used for tasks like online banking and encrypting personal messages RSA is one of the earliest forms of asymmetric encryption in computer science, and it's still in use. Data encryption is a vital aspect of data protection. It keeps transactions and devices secure. It has many benefits that increase security posture. But, it also has some drawbacks, which are important to be aware of.