9.2.4 Cryptography Algorithms Facts

Cryptographic algorithms come in three types, as follows:

Algorithm	Explanation
Symmetric	Generates a single key that is used for both encryption and decryption. If the key were to fall in the wrong hands, messages encrypted with the key, both past and future, could be decrypted.
Asymmetric	Generates two different yet mathematically related keys. The encryption key can be shared publicly. This is because the public key is used only to encrypt information. It cannot decrypt information at all. The only key that can decrypt the information is the private key.
Hashing	Instead of being used to encrypt information, hashing keys are used for signature verification and data integrity checking. They take a string of characters of an undetermined length and convert it into a string of characters that has a specific length. This output is known as a *digest*. Hashes should not be able to be reconstructed from the output of the hash function.

Use the right algorithm.

- Remember that modern cryptographic algorithms are extremely hard to crack; however, they are not 100% secure. As technology increases, it gets easier to crack the algorithms.
- Avoid already exploited algorithms if possible, including weak algorithms and deprecated algorithms.

General properties of cryptographic algorithms include:

Property	Description
Confusion	The relationship between the key and ciphertext should be as complex as possible.
Diffusion	The amount of change to the ciphertext when there is a change in the input text. The more the amount of change, the better the algorithm.
Collision	A collision is when two or more inputs create the same ciphertext.

Use case challenges.

Use Case	Description
Low Power Devices	Common public-key cryptography protocols perform poorly in low-energy environments making low power devices unsuitable for real-world activities.
Low Latency	One goal of cryptographic algorithms is encrypt and decrypt in a short amount of time. Strong cryptographic algorithms may take hundreds or thousands of clock cycles making them ill suited for low-latency applications.
High Resiliency	Cryptographic algorithms are more susceptible to cracking if information about the keys used in the algorithm is known. The leakage of information about an algorithm is a real-world problem. High resiliency algorithms or leakage-resilient algorithms are harder to crack, even if some information about the algorithm is know.