🔐 Skill 5 Cryptography

> 💬 "Go and get it" – لا تنتظر كل شيء جاهزًا!

📌 قواعد البحث عن الحلول

1. ابحث باستخدام Google أولًا.
2. استخدم ChatGPT للتوضيح فقط، وليس للحل المباشر.
3. تجنب النسخ المباشر للأسئلة في ChatGPT ❌.

🔥 Day 1: Encoding & Hashing Basics

Topics: Base64, Hex, Hash Functions (MD5, SHA-1, SHA-256)

🔹 **Exercises:**

1. Encode a string into Base64 and decode it back.
2. Convert a string into its Hexadecimal representation.
3. Write a script that calculates the MD5 hash of a given file.
4. Generate the SHA-1 hash of a string input.
5. Create a tool that takes a password and outputs its SHA-256 hash.
6. Verify if two files have the same SHA-256 hash (file integrity check).
7. Bruteforce simple MD5 hashes of 4-digit PINs (0000-9999).
8. Base64 encode a file (binary-safe).
9. Write a Python function that detects if a string is Base64 or Hex.
10. Build a script that hashes text using multiple algorithms (MD5, SHA1, SHA256).

🔥 Day 2: Symmetric Encryption (AES, DES)

Topics: AES, DES, Block Modes (ECB, CBC)

🔹 **Exercises:**

1. Encrypt and decrypt a message using AES-ECB mode.
2. Encrypt and decrypt a message using AES-CBC mode.
3. Generate a random AES key (128/256 bit) using secrets or os.urandom.
4. Write a script to pad and unpad plaintext (PKCS7 padding).
5. Encrypt a file using AES and write the output to disk.
6. Decrypt an AES-encrypted file (provide correct key and IV).
7. Encrypt a string with DES encryption.
8. Compare AES-ECB vs AES-CBC modes (encrypt same plaintext, observe difference).
9. Build a mini tool that encrypts text with user-provided key and saves it to file.
10. Implement AES encryption in CTR mode manually (counter based).

🔥 Day 3: Asymmetric Cryptography (RSA)

Topics: RSA Key Generation, Encryption, Signing

🔹 **Exercises:**

1. Generate an RSA private and public key pair in Python.
2. Encrypt a short message with RSA public key and decrypt with private key.
3. Write a script to sign a message with an RSA private key.
4. Verify an RSA signature with the public key.
5. Save and load RSA keys from .pem files.
6. Build a mini tool that encrypts text using RSA OAEP scheme.
7. Implement a hybrid encryption (RSA + AES): encrypt AES key with RSA.
8. Write a script that encrypts large files using hybrid encryption (RSA for AES key, AES for file).
9. Crack a very small RSA keypair (for educational purposes).
10. Write a script that tests RSA encryption/decryption speed for different key sizes (512, 1024, 2048 bits).
