Excellent réflexe 🔥 — c'est **exactement la mentalité CTF** :
👉 comprendre les principes ✅
👉 mais **aller vite avec les bons outils** 🧰

En CTF crypto, **le skill le plus utile n'est pas de tout faire à la main**, mais de savoir **identifier le type d'attaque** → puis **utiliser l'outil ou script adapté en 2 minutes**.

Voici donc une **grande "cheatlist"** que les équipes pro utilisent pour les crypto challenges en CTF.

🧭 1️⃣ Références & Sites indispensables (bases, algos, doc)

Type	Ressource	Utilité
🧠 **Crypto base**	CryptoHack.org	*Site d'exos crypto CTF progressif* (RSA, AES, DH, padding oracle…)
📚 **Références math & crypto**	CryptoPals	Exos sur RSA, CBC, CTR, etc.
📖 **Cheat sheets / writeups**	CTFTime Writeups	Lire les solutions d'anciens CTFs crypto.
🔍 **Explains attacks**	CryptoAttacks.io	Base de données d'attaques (RSA, ECC, etc.)
🧮 **Algorithmes mathématiques**	Alpertron ECM/Factor DB	Factoriser ou résoudre mod equations online
🧰 **Scripts / Libs Python**	PyCryptodome Docs	API de chiffrement Python moderne

---

👷 2️⃣ Outils Linux / Terminal indispensables

🔸 RSA & Math

Outil	Commande / Usage	Description
🧩 RsaCtfTool	python3 RsaCtfTool.py --publickey key.pub --attack all	Trouve automatiquement *small e, wiener, common modulus, shared prime…*

Outil	Commande / Usage	Description
🧮 msieve, yafu, factordb.com	msieve n	Facto rapide de n = p*q
🧰 rsatool	Génère clés RSA, calcule n, d, e, φ	
🧠 sage (SageMath)	Python pour les attaques plus avancées (Wiener, Lattice, etc.)	

🔸 AES / Modes

Outil	Usage	Détails
🧪 openssl	openssl enc -aes-128-cbc -in m.txt -out c.bin -K key -iv iv	Tester rapidement AES (CBC, ECB, CTR, GCM…)
🧬 CyberChef	Interface web (https://gchq.github.io/CyberChef)	“Swiss army knife” — conversion, AES, base64, XOR, etc.
🧱 padbuster	perl padBuster.pl URL ciphertext blocksize	Automatiser attaque *padding oracle (CBC)*
🧰 hashcat	hashcat -m 12000 hash.txt wordlist.txt	Brute-force des hashs ou PBKDF (passwords)

🔸 Diffie-Hellman / ECDH

Outil	Usage
sage	résout les discrete logs (Pollard rho, Pohlig-Hellman)
pari/gp	petit outil math pour tester mod p
dhparam (OpenSSL)	génère / vérifie paramètres DH
RsaCtfTool	détecte parfois *shared primes* entre DH/RSA

🔸 Hash & Passwords

Outil	Commande	Description
🧩 hashid	hashid hash.txt	Devine le type de hash
⚡ hashcat	hashcat -m <id> hash.txt wordlist.txt	Brute-force GPU
🧰 john	john --format=<type> hash.txt	Brute-force CPU
📜 Wordlists	/usr/share/wordlists/rockyou.txt	Dictionnaire standard CTF

🔸 Conversions & Manipulations

Outil	Commande / Description
xxd	convertit hexa <-> binaire
base64	encode / decode base64
python3 -m base64	base64 rapide
gmpy2, sympy (Python)	calculs mod n, inverse, racine, etc.
sagemath	manipulations modulo, algos crypto

🌐 3️⃣ Sites / Plateformes Online pratiques

Outil	Lien	Usage
🔹 **CyberChef**	https://gchq.github.io/CyberChef	conversions (hexa, base64, XOR, AES, etc.)
🔹 **Factordb**	https://factordb.com	factorisation RSA
🔹 **Dcode.fr**	https://www.dcode.fr	décodeurs classiques (césar, vigenère, RSA, etc.)
🔹 **Alpertron ECM**	https://www.alpertron.com.ar/ECM.HTM	factorisation/mod solver
🔹 **CrypTool**	https://www.cryptool.org	visualisation crypto (PC app)

Outil	Lien	Usage
🔹 **Boxentriq Cipher Identifier**	https://www.boxentriq.com/code-breaking/cipher-identifier	détecte automatiquement le type de chiffre

---

🐍 4️⃣ Librairies Python CTF-friendly

Lib	Installation	Description
pycryptodome	pip install pycryptodome	AES, RSA, SHA, padding…
pwntools	pip install pwntools	Automatisation réseau (padding oracle, bitflip)
gmpy2	pip install gmpy2	Maths rapides (mod, inverse, racines entières)
sagecell / sagemath	—	Maths avancées
sympy	pip install sympy	Résolution d'équations modulaire
requests	pip install requests	Interagir avec des serveurs (padding oracle)

⚙️ 5️⃣ Raccourcis utiles en CTF

Cas	Commande rapide
Convertir hex → int	int("deadbeef",16)
Convertir bytes → long	from Crypto.Util.number import bytes_to_long
Trouver inverse mod	pow(a, -1, n)
Trouver racine cubique entière	int(mpmath.root(x,3))
XOR deux fichiers	xor(file1, file2) dans CyberChef
Tester AES rapidement	openssl enc -aes-128-cbc -K <key> -iv <iv>
Générer clé AES random	openssl rand 16

🚀 6️⃣ Stratégie en CTF Crypto

Étape	Action
1. Identifier le type de crypto	RSA ? AES ? Hash ? XOR ? Base64 ?
2. Reconnaître la structure	clé publique ? ciphertext ? IV ? nonce ?
3. Deviner le mode	CBC ? ECB ? CTR ?
4. Utiliser l'outil adéquat	RsaCtfTool, CyberChef, Padbuster, etc.
5. Automatiser avec un script	Python rapide avec PyCryptodome ou pwntools
6. Vérifier	avec openssl ou CyberChef

🧩 7️⃣ Kit minimum à installer localement avant un CTF crypto

sudo apt install python3 python3-pip openssl hashcat john git

pip install pycryptodome pwntools gmpy2 sympy requests

git clone https://github.com/Ganapati/RsaCtfTool.git

git clone https://github.com/AonCyberLabs/PadBuster.git

Optionnel : installer **SageMath** si tu veux attaquer des RSA plus complexes :
👉 https://www.sagemath.org/

🧠 8️⃣ Bonus : Reconnaissance rapide d'un challenge crypto

Type de fichier / donnée	C'est probablement
Clé publique .pem avec BEGIN PUBLIC KEY	RSA
iv=..., ciphertext=...	AES CBC
nonce / counter	AES CTR / GCM
mod p, g, A, B	Diffie-Hellman
Hashs longs de 64 caractères	SHA256 / SHA512
Texte qui change légèrement	XOR / Substitution

Souhaites-tu que je te fasse une **fiche PDF “Crypto CTF CheatSheet” complète** (avec tous les outils, commandes, sites et exemples par type d’attaque : RSA, AES, DH, Hash, XOR, etc.) — que tu puisses imprimer ou garder pendant un hackathon ?