**🧠 Completed Learning Path: Smooth Cryptics**

**Status:** ✅ Completed (Weeks 1–8)

**Focus:** Cryptography, Secure Communication, Encryption Simulation

**✨ What I Learned**

* **Classical Ciphers**  
  Caesar, Vigenère, Monoalphabetic Substitution — the old-school building blocks of secrecy.
* **Modern Encryption Techniques**  
  AES, RSA, PKCS standards, hybrid encryption — and why real-world cryptography is never just one algorithm.
* **Data Integrity & Authentication**  
  HMACs, digital signatures, hashing with SHA256, and verifying message origins.
* **Base64 & Encoding Basics**  
  How to safely transmit binary across systems that only speak in text.
* **Key Exchange Protocols**  
  Simulated Diffie-Hellman exchange using modular math and secrets-based key generation.
* **Security Concepts**  
  Confidentiality, authenticity, integrity — and how they’re enforced in practice.

**🧪 Favorite Projects**

* 🔐 **RSA PGP-style Secure Messenger**  
  Simulated asymmetric encryption and digital signature handling, complete with tamper-detection and .pem key saving.
* 🌊 **Condiment Chat (Hybrid Encryption)**  
  AES + RSA hybrid scheme with EAX mode, base64 encoding, and signed messages between Milk and Eel — secure comms, saga-style.
* 🤝 **Diffie-Hellman Key Exchange Simulator**  
  Randomized private/public key generation with tamper detection and shared secret validation.

**🎯 Takeaways**

“I now understand how secure messaging works not just in theory, but line by line in Python.”  
— *Me, with a cigarette in one hand and a secure hash in the other*

* Confidence level: 🔒 6.5/10 (with cheat sheet); working on full autonomy.
* Most mind-bending concept: Primitive roots + the real magic of pow()
* Cryptography is **fun, practical, and weirdly poetic.**

**💡 What’s Next?**

➡️ Onward to **Cyber Mayhem 101**, where theory meets hacking simulation.  
Time to take this encryption knowledge… and weaponize it (ethically, of course).