ClawCTF Official Writeup

Challenge: Symmetry Is a Leak
Category: Cryptography
Difficulty: Hard / Final Tier

Challenge Overview

In this challenge, participants were provided with multiple ECDSA signatures along with an additional leaked value per signature. While ECDSA is cryptographically secure when nonces are generated randomly, this challenge intentionally introduced a structured weakness in nonce generation. The objective was to identify this weakness, recover the private signing key, and extract the embedded flag.

Given Files

Participants received only two files: README.md describing the challenge context, and output.txt containing public parameters, signatures, and leaked values. No source code or solver hints were provided.

Step 1: Identifying the Vulnerability

Each signature included a large leaked integer. The consistent size and structure of these leaks strongly suggested deterministic behavior tied to the nonce. This pointed toward a biased or partially leaked nonce attack on ECDSA.

Step 2: Understanding the Leak Structure

By analyzing the growth rate and symmetry of the leaked values, it was deduced that the nonce was split into five high-order decimal blocks. These blocks were interpreted in a non-standard base and multiplied by their reversed representation, creating a symmetric polynomial leakage.

Step 3: Linearization of the Leakage

Expanding the polynomial allowed the leak to be rewritten as a linear combination of derived variables. This transformed the problem into solving a bounded linear equation, enabling recovery of intermediate values associated with the nonce digits.

Step 4: Recovering Nonce Digits

The intermediate variables formed a small nonlinear system that was efficiently solved using Groebner bases. This revealed the most significant decimal digits of each ECDSA nonce.

Step 5: Lattice Attack on ECDSA

With partial nonce knowledge available, the problem reduced to a Hidden Number Problem. A carefully constructed lattice was built using the ECDSA equations and reduced using the LLL algorithm. From the reduced basis, the private key was directly recovered.

Step 6: Flag Extraction

The private key contained the flag embedded in its high-order bits. Converting the recovered private key to bytes revealed the final flag.

Final Flag

ClawCTF{th3_cl4ws_w3r3_n3v3r_r4nd0m_9f1a72cdd9e3a11}

Conclusion

This challenge demonstrated how subtle structure in cryptographic randomness can entirely compromise security. It required a combination of cryptographic knowledge, algebraic reasoning, and lattice-based techniques. The challenge was designed to reward deep understanding rather than brute force approaches.