

# Pedagogy behind crypto CTF challenges and a way to improve them

Emanuel Nicolás Herrador  
emanuel.nicolas.herrador@unc.edu.ar



Facultad de Matemática,  
Astronomía, Física y  
Computación



UNC

Universidad  
Nacional  
de Córdoba

## What's CTF?

- Capture The Flag is a competition for cybersecurity field.
- Problems are solved obtaining *flags* (words with specific format) and they are given when we broke a weak protocol, corrupted file or system for it.
- Problem categories:

|                        |              |       |
|------------------------|--------------|-------|
| Web<br>Exploitation    | Cryptography | OSINT |
| Binary<br>Exploitation | Forensics    | Misc  |
| Reverse<br>Engineering | Hardware     | PPC   |

## Importance

- Practical experience and exposure to real-world vulnerabilities or attacks.
- Teamwork and community across social websites, jobs and universities. A way to combine all the worlds of learning (self learning, academy, industry).
- Worldwide competitions: *DEF CON*, *ICC*, *ECSC*, *Pwn20wn*, *Google CTF*.
- Gamification way to learn and teach:
  - To improve playing in cybersecurity.
  - Discovering new attacks or ways to think problems.
  - Mixing theory, practice and tricks.
- Used in a lot of universities for competitions, teach and generate more interest in cybersecurity field.

## Complexity for a problemsetter

- Find interesting topics for problems.
- Being creative and make it useful to learn.
- Make it neither too easy nor too difficult.
- Feasible to solve in the way we thought and, also, ensure it as unique to test specific objective skills for evaluation.
- Avoid being solved only with pre-existent automated tools or LLMs (as with *ico* problem of DEFCON finals 2025).

## How would you create a chall in your study field?

- CTF challs are a useful way to learn and teach each area of study.
- Do you want to try creating one for teaching or share with partners?
- Tell me more and some ideas! I'll be doing feedback until October 20.



## Crypto CTF challenge example: Samsara

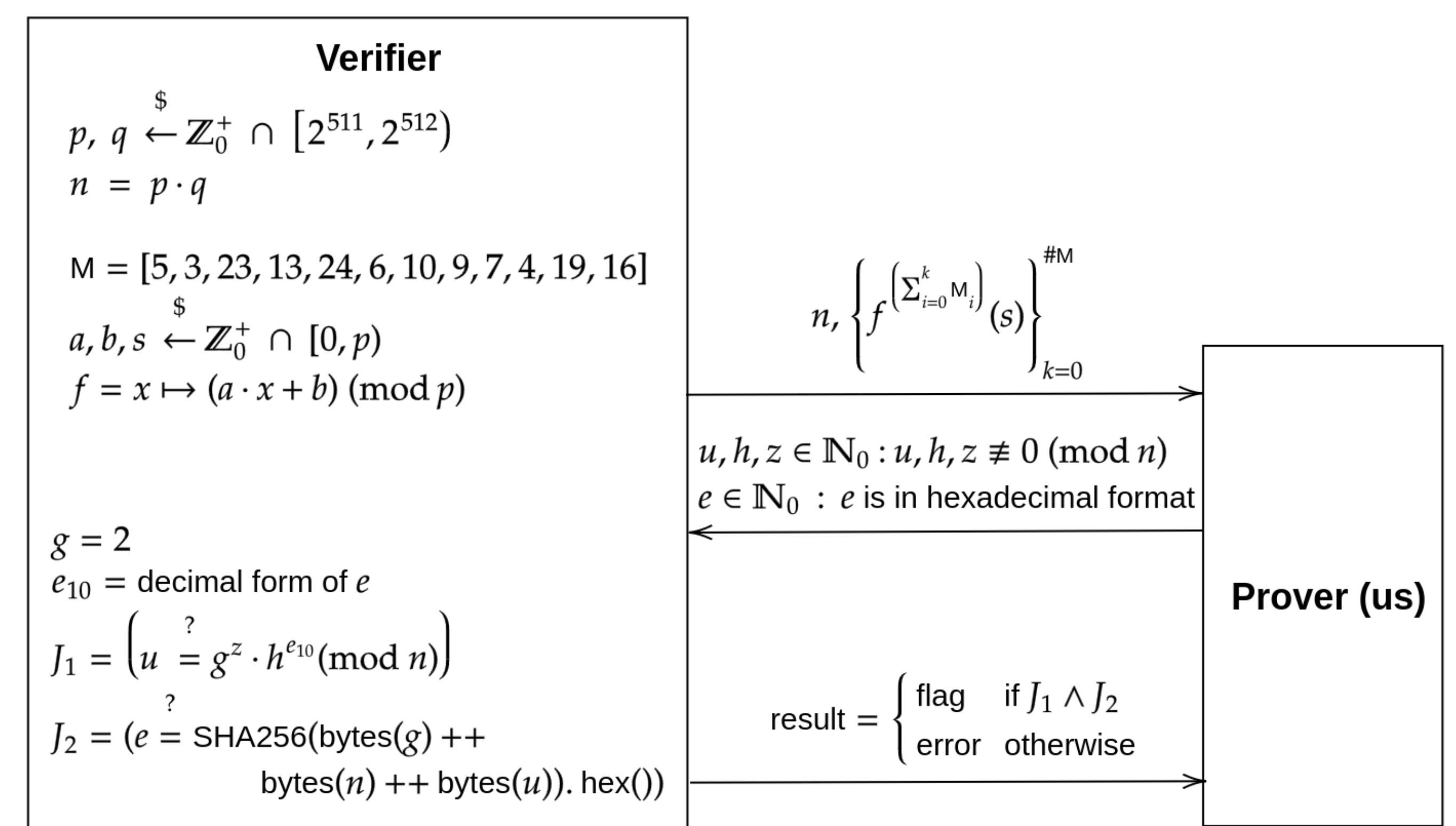
### International Cybersecurity Challenge 2024

(Santiago, Chile)

Topics: Linear Congruential Generator, Non-interactive Girault's proof of knowledge (with Fiat-Shamir transformation) and Frozen Heart Vulnerability.

### Statement

Break the following protocol to capture the flag:



**Solution idea:** We've two main parts. Let see each one separately.

### First part (Linear Congruential Generator)

- LCG with unknown parameters and are given non consecutive terms of this sequence (with its positions).
- Let be  $s_k = f^{(k)}(s)$  the terms we have. An useful property to use is  $s_A - s_B \equiv (a^A - a^B) \left( s + \frac{b}{a-1} \right) \pmod{p}$ .
- Let  $T = pX, S = pY$ , then  $\gcd(S, T) = p \iff \gcd(X, Y) = 1$ . Since the probability that two random numbers are coprime is 61%, we've a good chance to get  $p$  with a little quantity of  $(T, S)$  tries.
- Aiming  $R : R \not\equiv 0 \wedge R \equiv 0 \pmod{p}$ , we'll search for positions values  $A, B, C, D, E, F, G, H \in \left\{ \sum_{i=0}^k M_i \right\}_{k=0}^{\#M}$  such that:
$$0 \equiv (s_A - s_B)(s_C - s_D) - (s_E - s_F)(s_G - s_H) \equiv B^2 K \pmod{p}$$
where  $B = s + \frac{b}{a-1}, K = (a^A - a^B)(a^C - a^D) - (a^E - a^F)(a^G - a^H)$ .
- Since  $B \equiv 0 \pmod{p}$  is unlikely to happen, we aim for  $p|K$ . For that:
$$\{A + C, B + D\} = \{E + G, F + H\} \wedge \{A + D, B + C\} = \{E + H, F + G\}$$
- Let  $m = \#M + 1$ . The search can be done with brute force in  $O(m^8)$ , but searching intelligently in  $O(m^6)$ . The total number of useful combinations is 288 and we're able to get  $p$  and also  $q = \frac{n}{p}$ .

### Second part (Girault's identification non-interactive protocol)

- $h$  isn't used in hash and it's a weakness (*Frozen Heart Vulnerability*).
- Choosing random values  $u, z$ , we've  $e = \text{Hash}(g, N, u)$  and we aim for  $h \equiv (u \cdot g^{-z})^{e_{\text{inv}}} \pmod{n}$  where  $e_{\text{inv}} \equiv e^{-1} \pmod{\phi(n)}$  because:
$$g^z \cdot h^e \equiv g^z \cdot ((u \cdot g^{-z})^{e_{\text{inv}}})^e \equiv g^z \cdot u \cdot g^{-z} \equiv u \pmod{n}$$
- We can compute  $e_{\text{inv}}$  easily with *Extended Euclidean algorithm* because we know  $\phi(n)$ . If it doesn't exist, try another  $u, z$  random values.
- Sending these values to verifier, it should give us the flag message.

### Complete solution and implementation



github.com/helcsnewsxd/latincrypt-2025