(Applied) Cryptography

Tutorial #5

Manuel Barbosa (mbb@fc.up.pt) Rogério Reis (rvr@fdcc.c.up.pt)

MSI/MCC/MERSI – 2021/2022

1 - Implement the universal hash function of poly1305 in Sage

- recall $H((K_1, K_2), (M_1, M_2, \ldots)) = K_1 + P(K_2)$ where $P(X) = K_1 + M_1 X + M_2 X^2 + \ldots$
- use F = FiniteField(2**130-5) to define the type of coefficients
- use PR.<X> = PolynomialRing(F) to define the type of polynomials
- define the key to the hash as a pair in F
- define the message as a list in F, which you can cast to a polynomial (careful with 0-th coefficient, which comes from the key)
- computing the hash is evaluating the polynomial at the other key component
- What is the probability that the hash of two fixed messages collide for a randomly sampled key?

2 - Use Python to encrypt a file with AES-GCM

- Make sure you can decrypt it with openSSL (if the command line does not support AEAD in your machine, use this tool `https://github.com/jforissier/aesgcm`).
- Modify the encrypted file
- See if you can still decrypt it with openSSL
- How would this be different if you were using AES-CTR?

3 - Which of the following statements are true and which are false?

- $2n = \mathcal{O}(n$
- $n^2 = \mathcal{O}(n)$
- $n^2 = \mathcal{O}(n \log n)$
- $n \log n = \mathcal{O}(n^2)$
- $3^n = \mathcal{O}(2^n)$
- $\mathcal{O}(2^{n^2}) = 2^{\mathcal{O}(n^2)}$