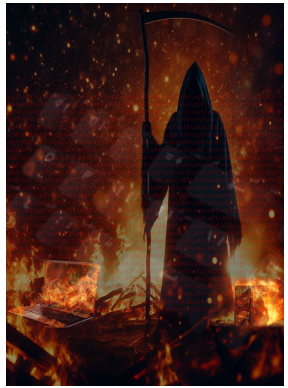
Who Dis?

Mikhail Sudakov

- Hacker, developer, researcher, educator
- Father, gamer, cat owner, meme enthusiast
- Favorite phrases: “I don’t know”, “Try harder!”
- Sr. Staff Security Researcher @ Trellix
- Grim Reaper (OSEE), $OSCE^3$



Socials:

- https://www.linkedin.com/in/mikhail-sudakov
- https://x.com/msudakov0

- Cryptology (from Greek):
  - Kryptos – “hidden” or “secret”
  - Logos/logia – “study” or “knowledge”

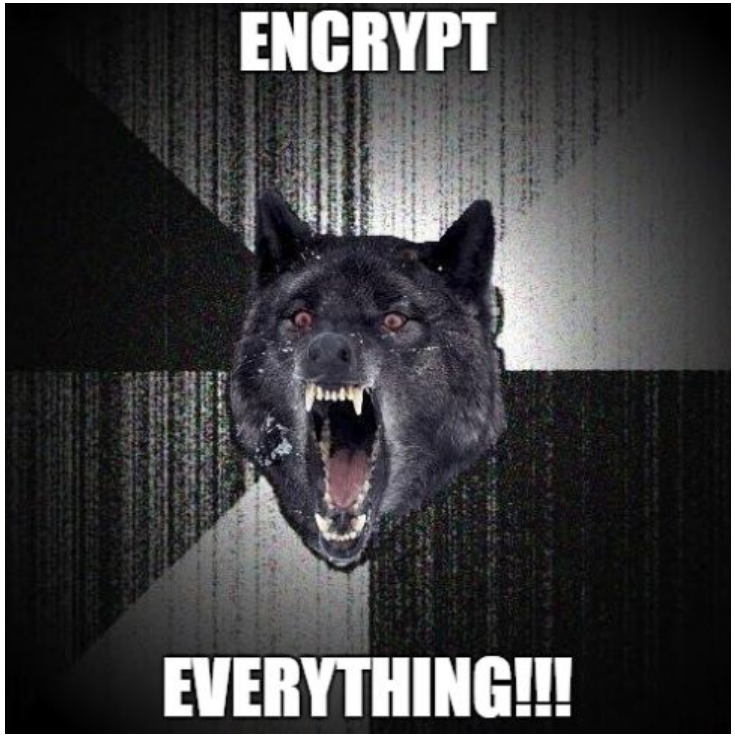
```
00110101 00110010 00110011 00110010
00110011 00111001 00110111 00110110
00110101 01000001 00110100 00110011
00110100 00110010 00110111 00110001
00110110 00110010 00110011 00110010
00110100 00111001 00110110 00111000
00110100 00111001 00110101 00110001
00110011 01000100 00110011 01000100
```

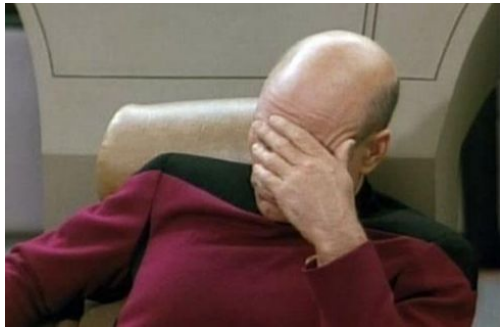
- Includes:
  - **Cryptography**:
    - Science of hiding the meaning of a message (“writing codes”)
  - **Cryptanalysis**:
    - Science of forcefully revealing the meaning of a message (“breaking codes”)
  - **Steganography**:
    - Science of hiding the mere presence of a message (e.g. writing with finger on glass)
  - **Steganalysis**:
    - Science of revealing the presence of a hidden message
  - Other fields…

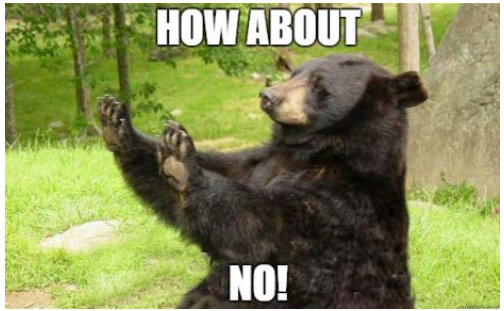
Crypto…

- Is…
  - Brutally complex
  - An incredibly powerful tool
  - A fundamental part of many security mechanisms and protocols
- Is not…
  - A hammer to strike every nail with
  - A solution to all of the world's (security) problems
  - At all reliable unless implemented very precisely and very correctly
  - Something you should **EVER** be inventing and using yourself!

What Cryptology Is Absolutely **NOT**



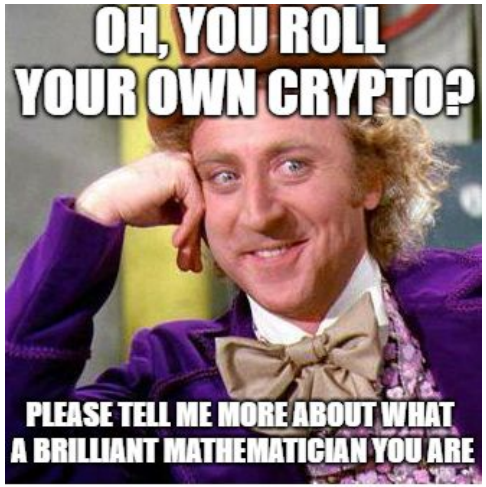




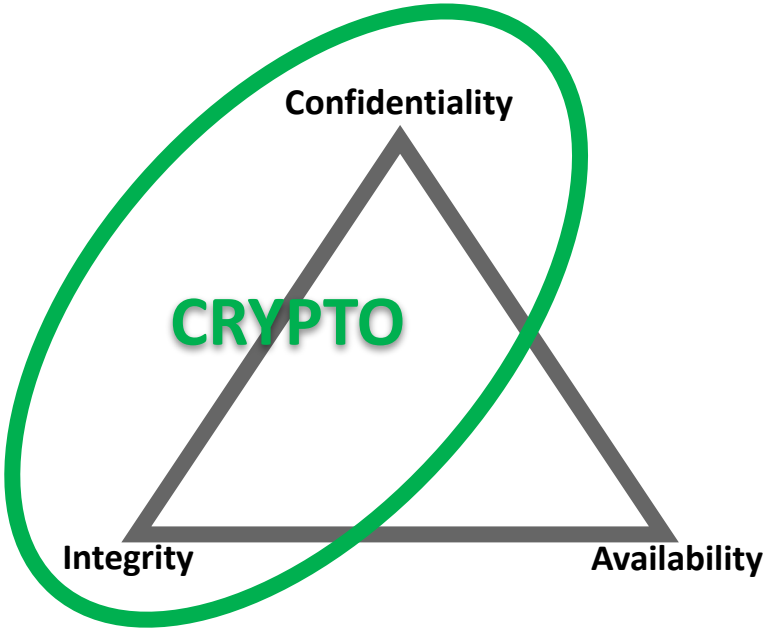
If you only get one thing out of this…

Let it be this:

→ Never, **EVER**, under any circumstances roll your own crypto



Crypto in the Triangle of Security



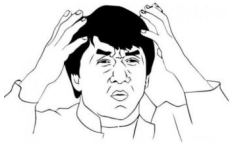
Threat Model (Adversarial Capability)

Adversarial Strength

- **Ciphertext-Only Attack (Known-Ciphertext)**: only a set of ciphertexts
  - *E.g. "a3feac02f2ddbf1c558dc26ec1e68a8a952e0924d" – GO!*
- **Known-Plaintext Attack**: a set of both plaintexts and ciphertexts
  - *E.g. Bletchley Park team Vs Enigma in WW2*
- **Chosen-Plaintext Attack**: can encrypt arbitrarily
  - *Active attacker, defeats non-randomized encryption*
- **Chosen-Ciphertext Attack**: CPA + can decrypt arbitrarily
  - *E.g. padding oracle attack, defeats non-authenticated encryption*

“Secure” Against ***What***??

Cipher	Security Against CO	Security Against KP	Security Against CP	Security Against CC
Caesar Shift				
Vigenère Cipher				
Enigma Machine				
AES				



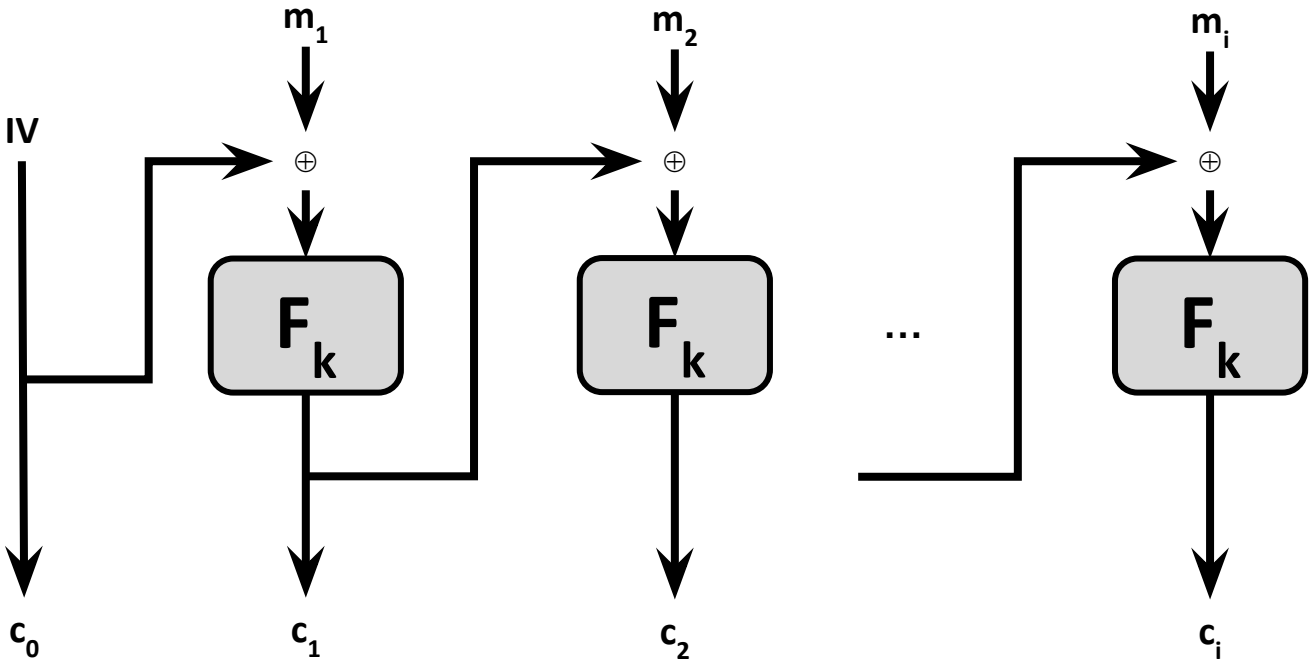
AES - Advanced Encryption Standard

- Block cipher:
  - 128-bit block size
- Subset of Rijndael ciphers:
  - Designers: Vincent Rijmen and Joan Daemen of Belgium
- Varying key lengths:
  - 128, 192, and 256 bits
- Federal standard since 2002
- Superseded DES and 3DES
- Based on substitution-permutation network design
  - Does not use a Feistel network (like DES)

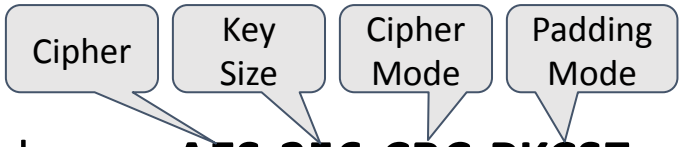
Important AES Elements

- IV (Initialization Vector)
  - Allows for randomized encryption to thwart Chosen-Plaintext Attacks
  - Gives cipher a direction/entropy/chaos/unpredictability
  - Is a block of 16 bytes (e.g. 0x4740e55b7fdf643945e33c400144d129)
- Cipher Mode of Operation
  - How the underlying cipher algorithm is implemented on actual blocks
  - Examples: CTR, ECB, **CBC**, PCBC, CFB, OFB, GCM (special), etc.
- Padding Mode
  - For many block ciphers, it is necessary to pad the plaintext in a block
  - Encoding vs encryption
  - Examples: `NULL` bytes, `1|00…00`, ISO10126, **PKCS7**, etc.

AES-CBC (Cipher Block Chaining) Mode



Attack Specifications



- Crypto scheme: **AES-256-CBC-PKCS7**
- Target ciphertext: `a81d5cff84c7badbf53bad80df3c542ba78b0abc8beedea1f7545232c38baed2`
- Ciphertext structure:

Initialization Vector (IV)	Actual AES-CBC Ciphertext
`a81d5cff84c7badbf53bad80df3c542b`	`a78b0abc8beedea1f7545232c38baed2`

!! DEMO TIME !!

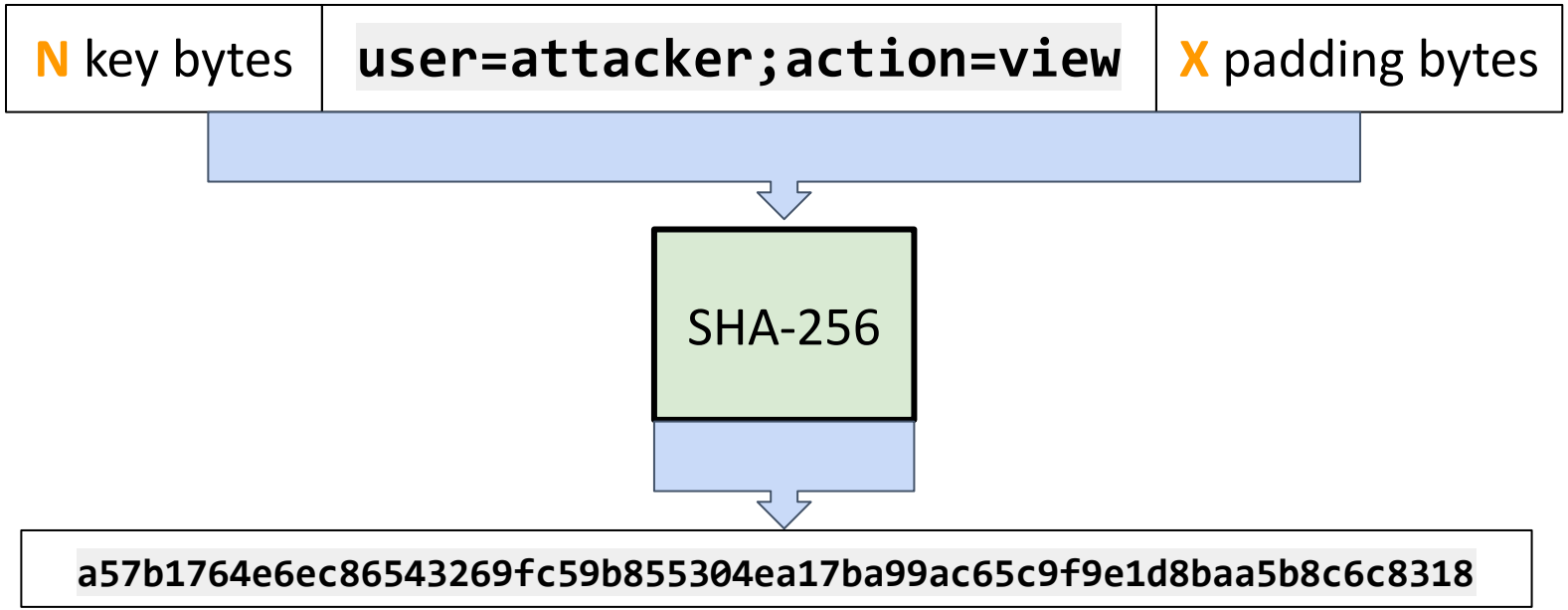
Some Hash Functions

Length Extension Attack

- MD5 (Message Digest Algorithm 5) by Ronald Rivest
  - 128-bit output
  - Collisions found in 2004 – **completely broken** and mustn't be used for crypto purposes ever again
- SHA-1 (Secure Hash Algorithm 1) by the NSA
  - 160-bit output
  - Collisions found in 2017 – **broken** and mustn't be used anymore for crypto purposes
- **SHA-2 (Secure Hash Algorithm 2) by the NSA**
  - **256 or 512-bit output – no known weaknesses**

- SHA-3 (Secure Hash Algorithm 3) by Bertoni, Daemen, Peeters, and Van Assche
  - Is a subset of the broader hashing family called "Keccak"
  - Very different in design than the NSA's SHA family (not vulnerable to length extension attack)
  - 224, 256, 384, and 512-bit outputs – no known weaknesses
  - Was selected as the winner of the NIST's hash function competition
  - Is the current hashing standard as of August 5, 2015

SHA-256 Cookie Integrity Scheme



SHA-256 Length Extension Attack

- [Theoretical] Attack on the *Merkle-Damgård construction*, not on SHA
  - It exploits the length extension property of such algs. when used incorrectly

- We want to compute a valid hash for the cookie string:

  `user=attacker;action=view;admin=true`

- Cannot guess/forge a hash for an arbitrary message, without knowing the key…
  - That would mean finding a collision against the algorithm

SHA-256 Length Extension Attack

- …but, we can append an arbitrary string to the end of the original cookie message, like so:

  `user=attacker;action=view<PADDING_BYTES>;admin=true`

  - Unfortunately, padding bytes are required
  - Padding bytes must be precisely computed

- To do that, we must know the full length of the original message
  - Already know the cookie message itself, so we must find the key length
    - Key length can be easily guessed or brute-forced

!! DEMO TIME !!