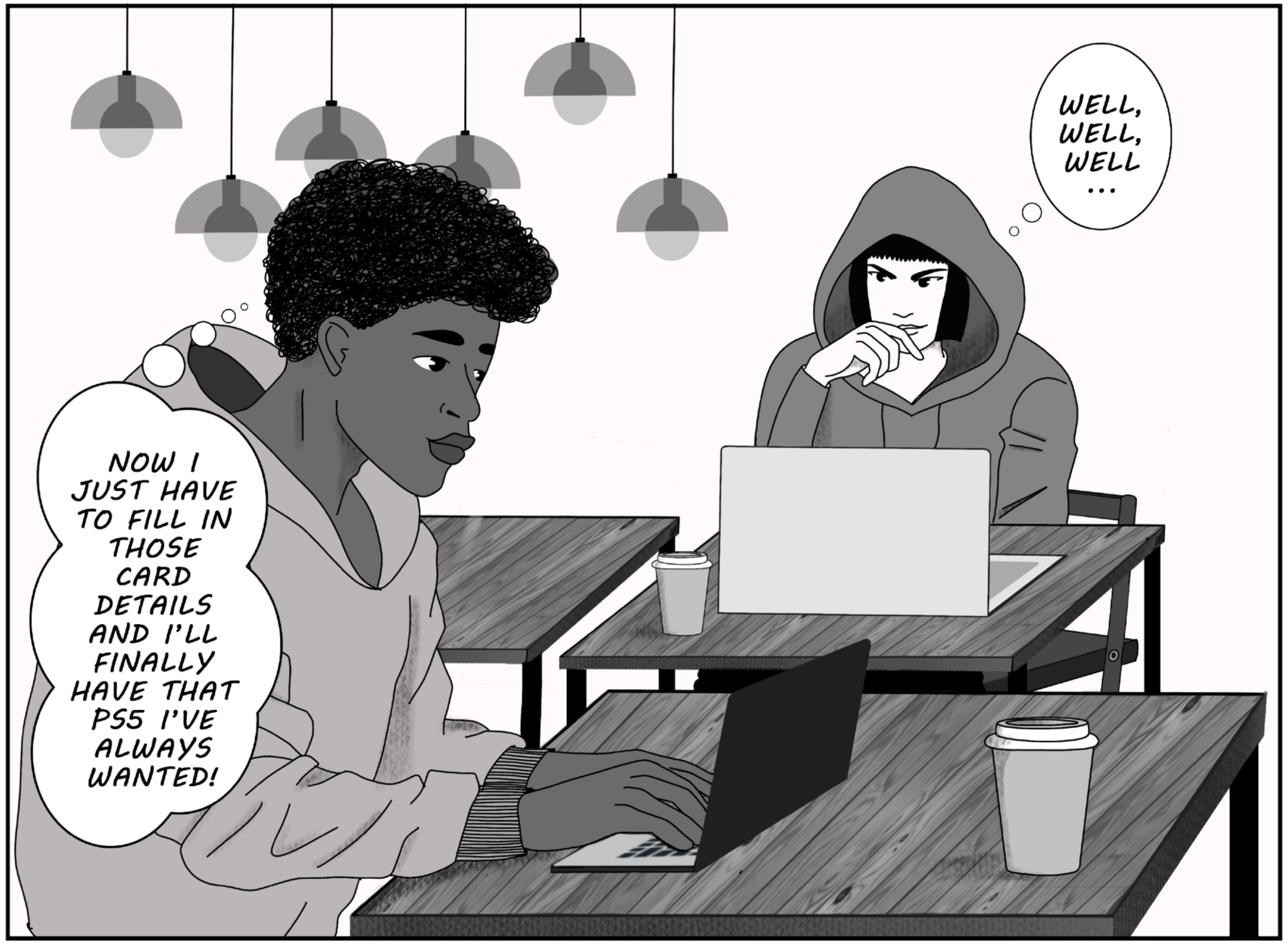
UNLOCKING THE UNKNOWNS

CRYPTOGRAPHY ESSENTIALS FOR SPRING DEVELOPERS

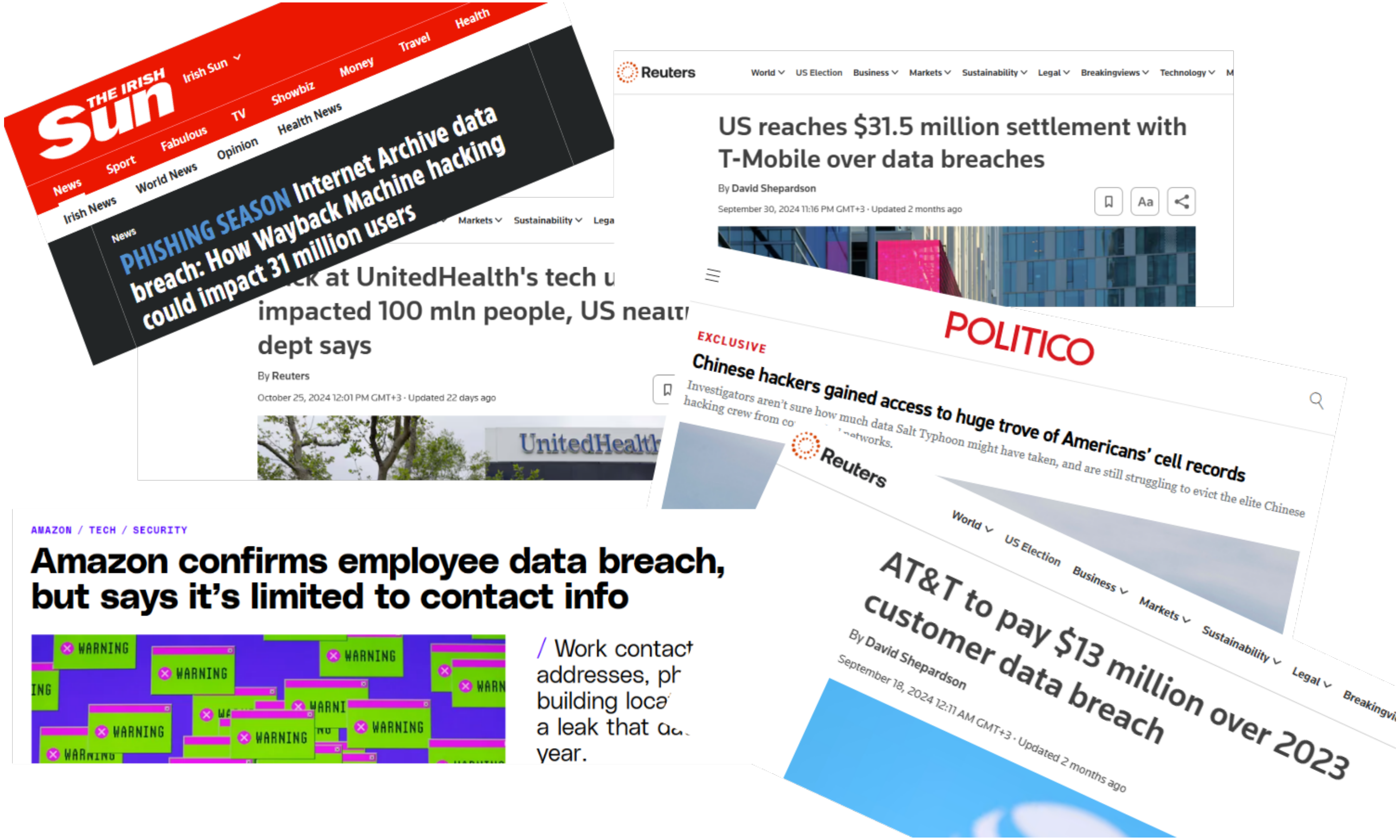
Cryptography
Why should you care?

- Your app is a target
- You store and send secrets
- Network isn't safe
- You might be liable
- Spring makes it easy


WELL, WELL, WELL ...
NOW I JUST HAVE TO FILL IN THOSE CARD DETAILS AND I'LL FINALLY HAVE THAT PS5 I'VE ALWAYS WANTED!

Fact

Developers **can be** legally liable in certain jurisdictions if user data is compromised due to weak encryption or password storage.



THE IRISH Sun Irish Sun

News Sport Fabulous TV Showbiz Money Travel Health

Irish News World News Opinion Health News

News

PHISHING SEASON Internet Archive data breach: How Wayback Machine hacking could impact 31 million users

Reuters World US Election Business Markets Sustainability Legal Breakingviews Technology M

US reaches $31.5 million settlement with T-Mobile over data breaches

By David Shepardson

September 30, 2024 11:16 PM GMT+3 · Updated 2 months ago

Markets Sustainability Lega

…ck at UnitedHealth's tech u… impacted 100 mln people, US healt… dept says

By Reuters

October 25, 2024 12:01 PM GMT+3 · Updated 22 days ago

POLITICO

EXCLUSIVE

Chinese hackers gained access to huge trove of Americans' cell records

Investigators aren't sure how much data Salt Typhoon might have taken, and are still struggling to evict the elite Chinese hacking crew from co… …networks.

Reuters World US Election Business Markets Sustainability Legal Breakingvi…

AT&T to pay $13 million over 2023 customer data breach

By David Shepardson

September 18, 2024 12:11 AM GMT+3 · Updated 2 months ago

AMAZON / TECH / SECURITY

Amazon confirms employee data breach, but says it's limited to contact info

/ Work contact addresses, ph… building loca… a leak that oc… year.

spring io

Spring Security IN ACTION Second Edition
JAVA Ghidul practic și interactiv pentru începători
Laurențiu Spilcă
TROUBLESHOOTING Java
Spring Start Here
Spring Security IN ACTION

youtube.com/@laurspilca

@laurspilca

Laurențiu Spilcă

The basics

- Symmetric encryption
- Asymmetric encryption
- Cryptographic hashing
- Specifications

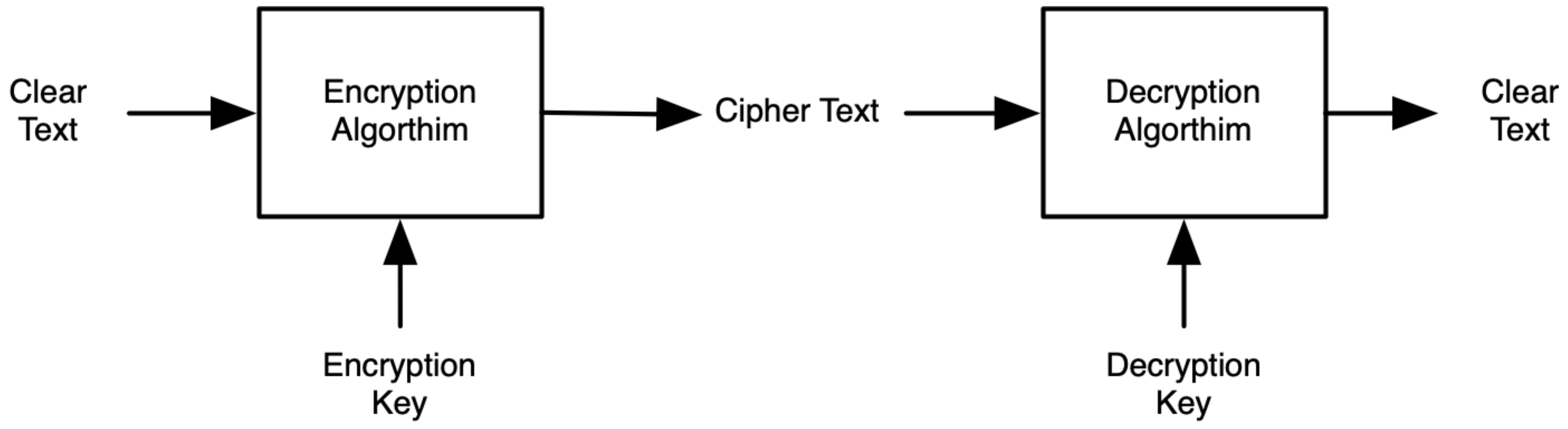
Symmetric encryption

Common algorithm: AES

- One key to rule them all
- Fast!

✓ Good for performance

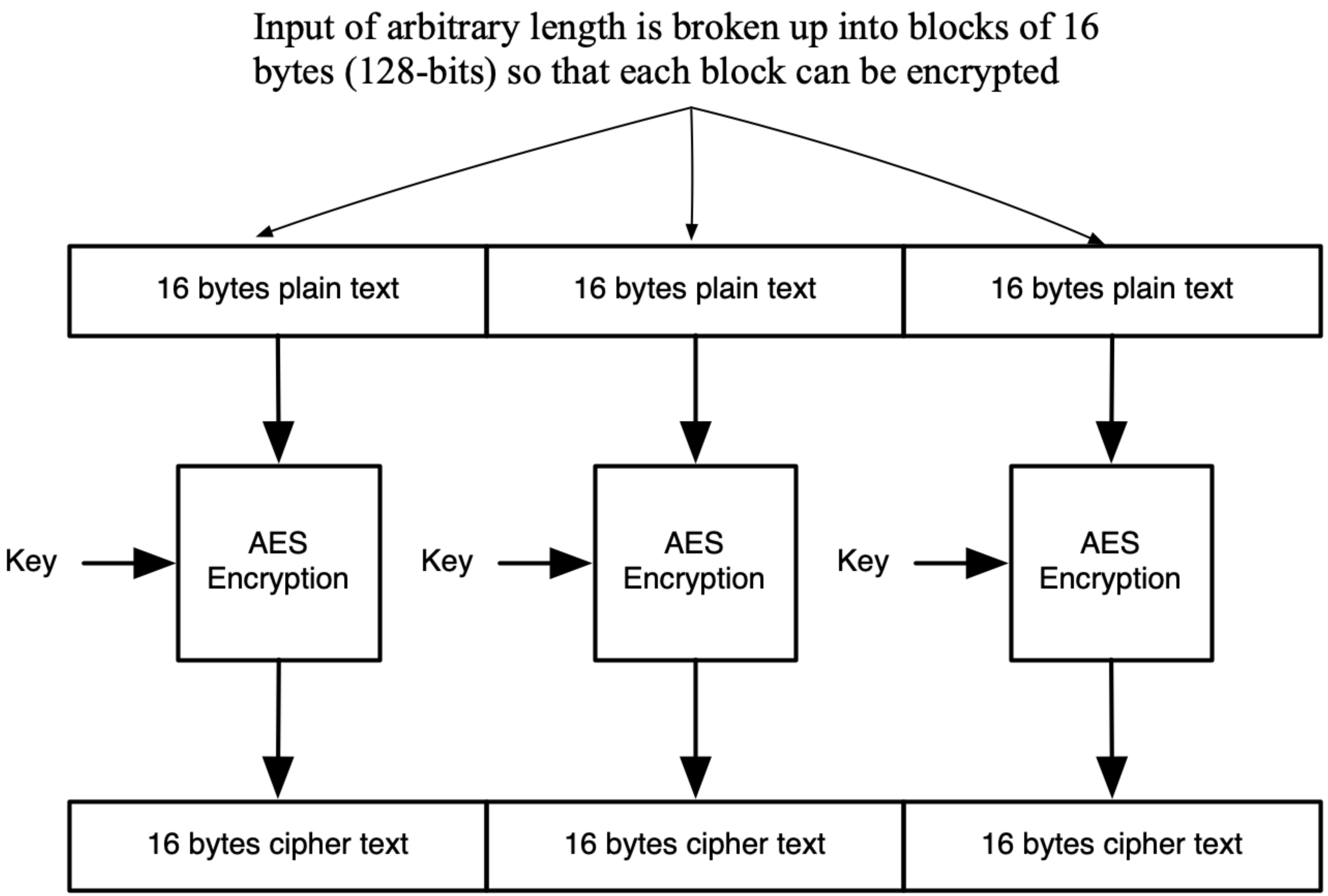
✗ Challenging because of key distribution



Fact

AES has been around since 2001 and replaced DES, which had a backdoor weakness.

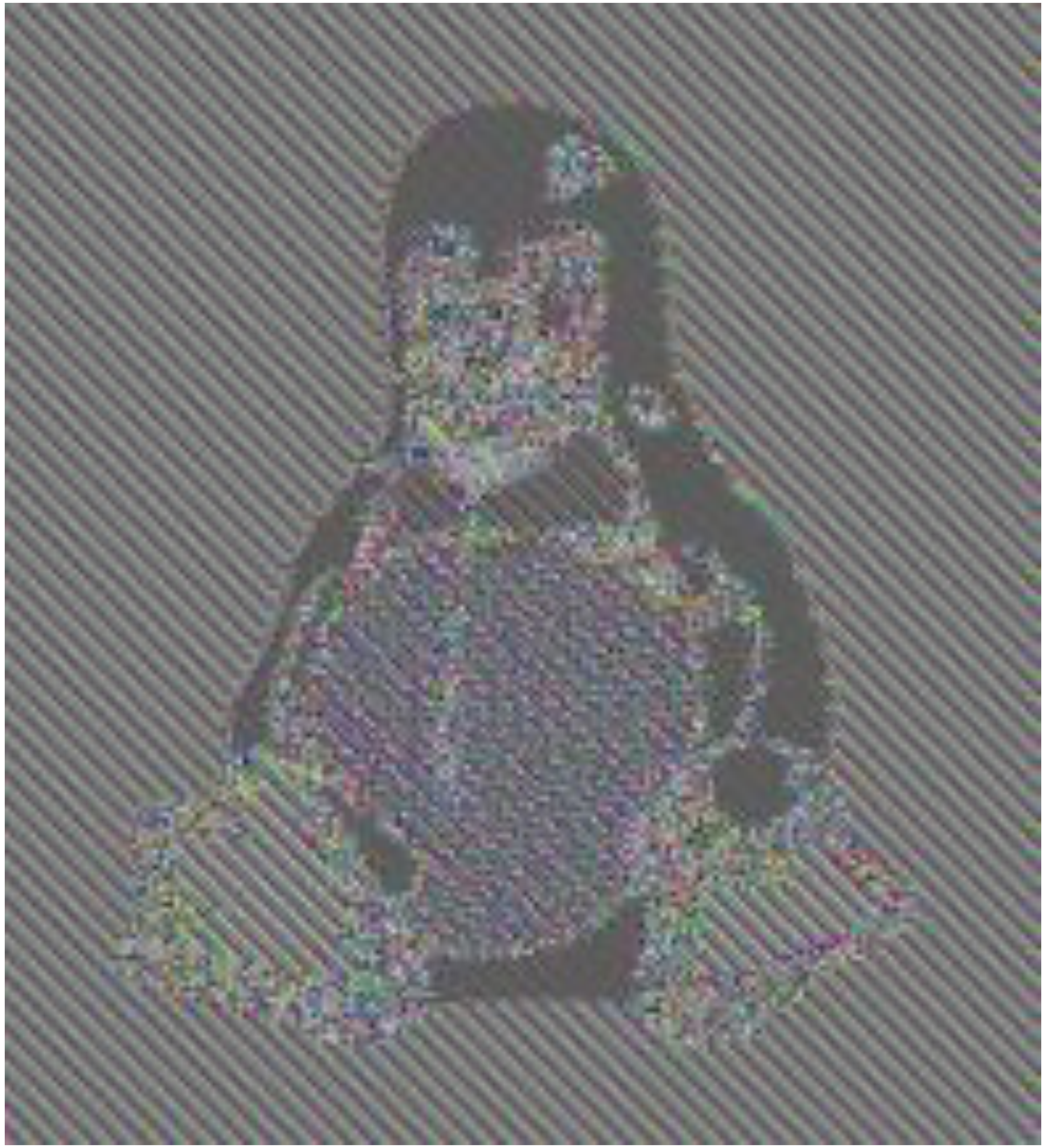
Advanced Encryption Standard



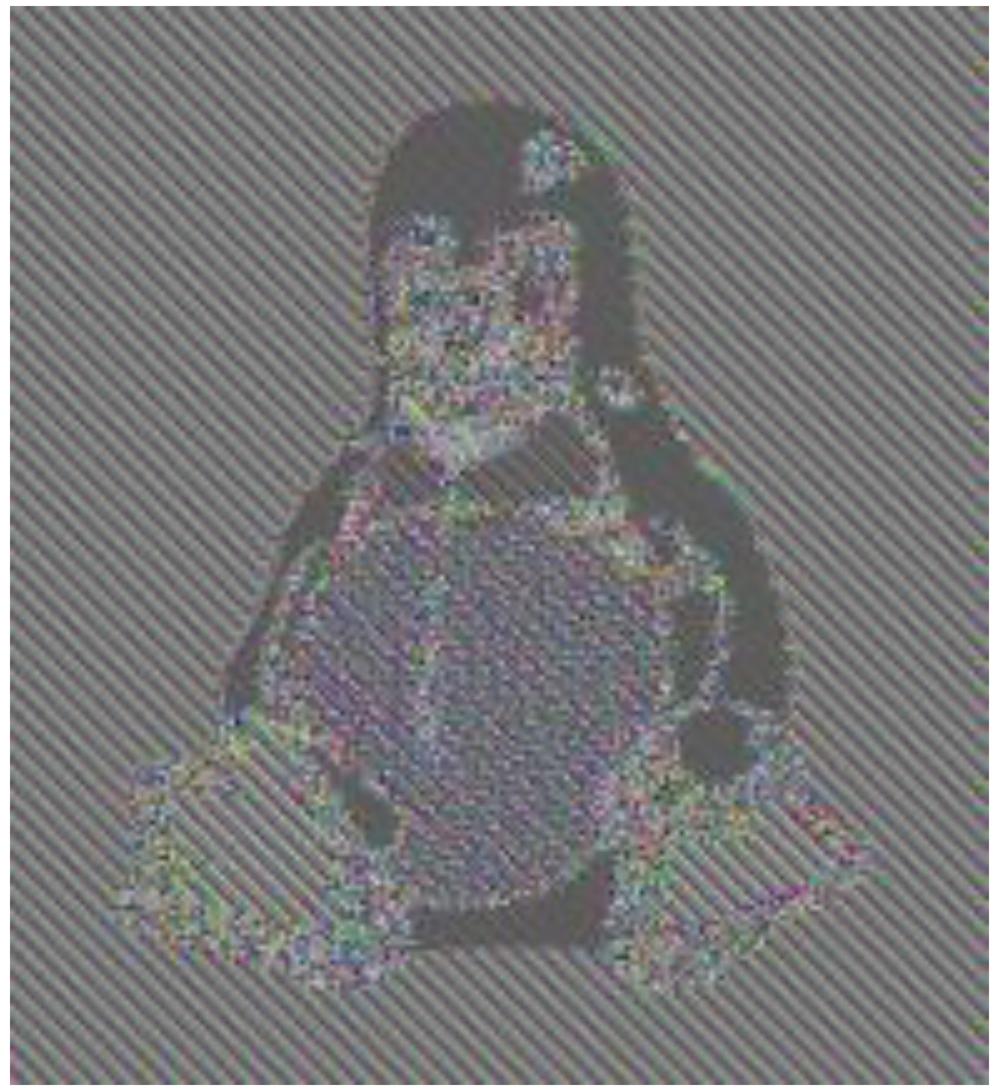
Advanced Encryption Standard

Block cipher mode

- **Electronic Code Book (ECB)**: Encrypts each block of data independently
- **Cipher Block Chaining (CBC)**: Encrypts each block based on the previous block
- **Galois Counter Mode (GCM):** Combines CTR mode for encryption with Galois Field multiplication for authentication



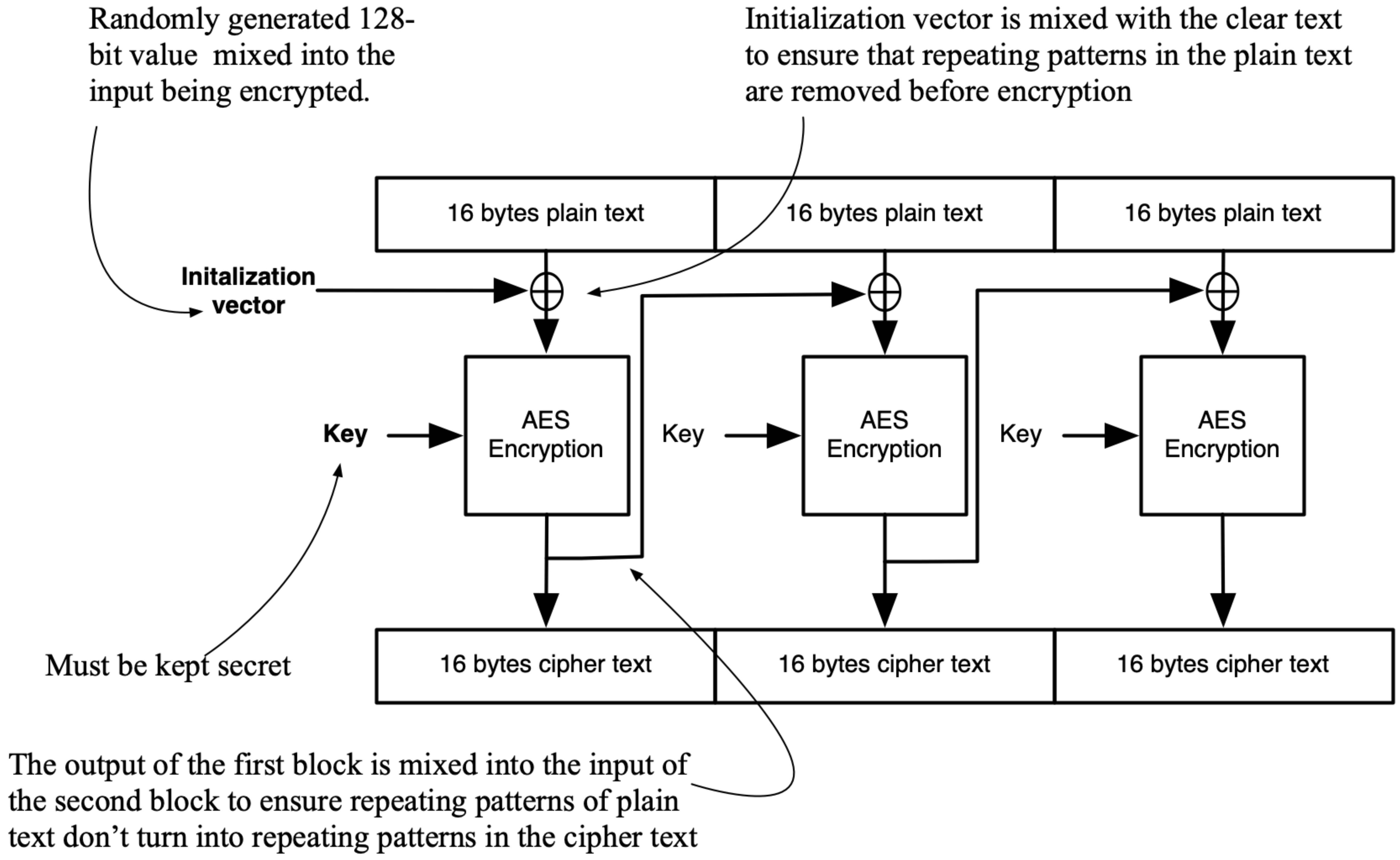
Fact



ECB was famously used in the “Penguin Image Attack” where encrypting a photo of Tux the Linux penguin with ECB revealed... the penguin.

Advanced Encryption Standard

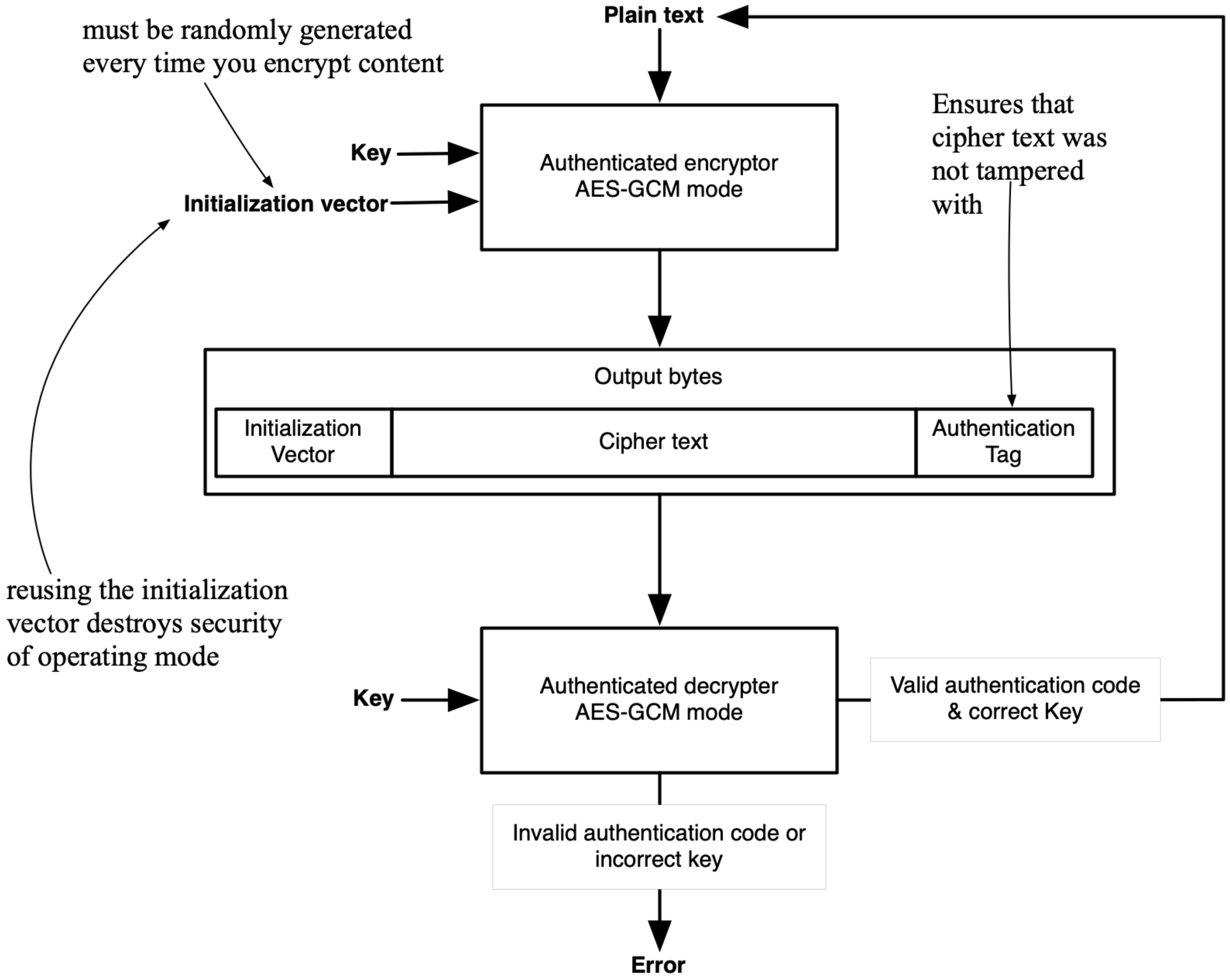
CBC



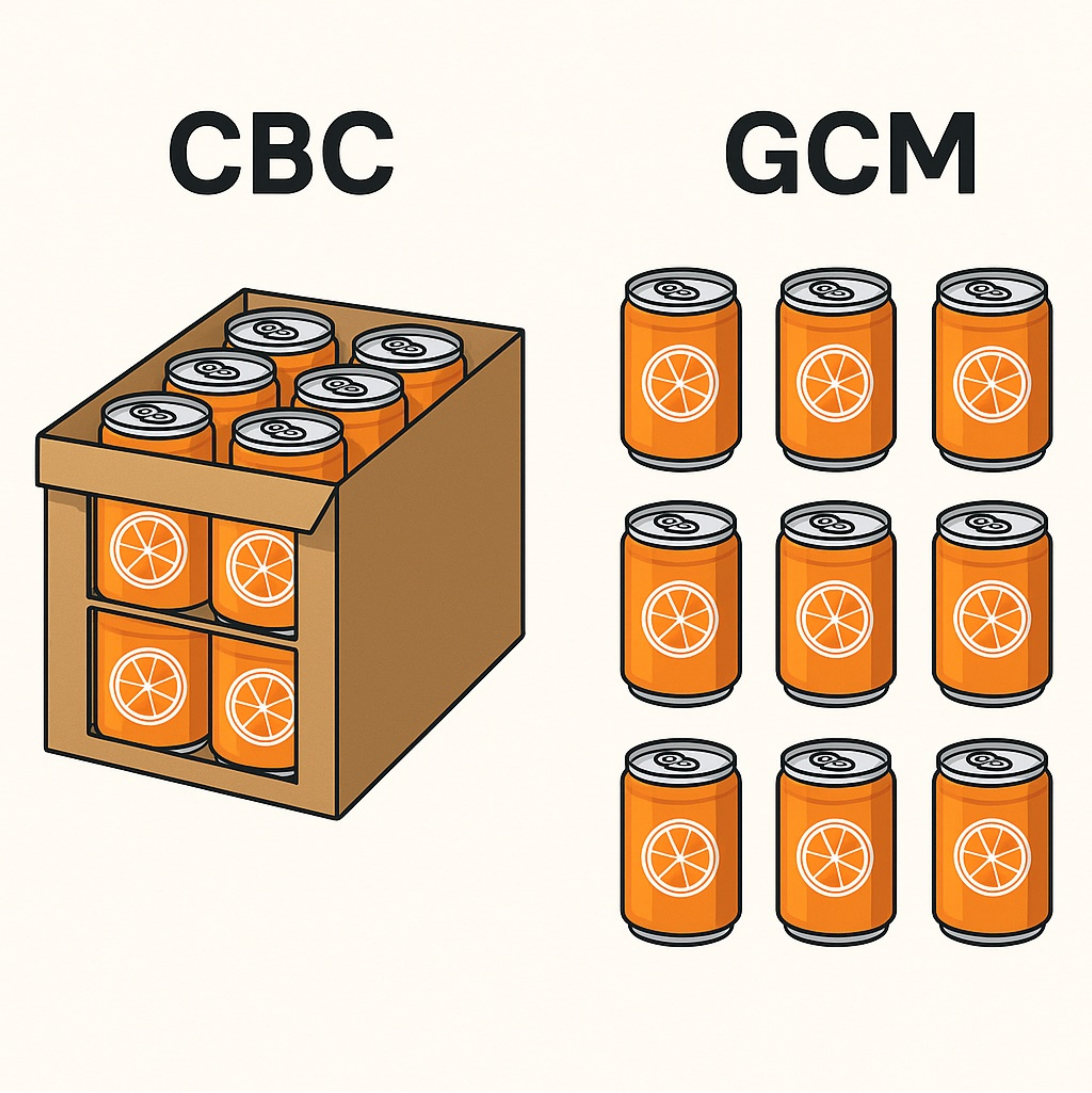
Randomly generated 128-bit value mixed into the input being encrypted.
Initialization vector is mixed with the clear text to ensure that repeating patterns in the plain text are removed before encryption
16 bytes plain text
16 bytes plain text
16 bytes plain text
Initalization vector
Key
AES Encryption
Key
AES Encryption
Key
AES Encryption
Must be kept secret
16 bytes cipher text
16 bytes cipher text
16 bytes cipher text
The output of the first block is mixed into the input of the second block to ensure repeating patterns of plain text don’t turn into repeating patterns in the cipher text

Advanced Encryption Standard

GCM



Plain text
must be randomly generated every time you encrypt content
Key
Initialization vector
Authenticated encryptor AES-GCM mode
Ensures that cipher text was not tampered with
Output bytes
Initialization Vector
Cipher text
Authentication Tag
reusing the initialization vector destroys security of operating mode
Key
Authenticated decrypter AES-GCM mode
Valid authentication code & correct Key
Invalid authentication code or incorrect key
Error


CBC
GCM

Advanced Encryption Standard

Mode	Confidentiality	Integrity	IV required	Parallelizable	Notes
ECB	✅	❌	❌	❌	Insecure
CBC	✅	❌	✅	❌	Good
GCM	✅	✅	✅	✅	Preferred

Spring Security uses GCM under the hood
in certain configurations

Symmetric encryption in practice

- Storing sensitive config data
- Encrypting data at rest
- Securing data in a shared store

Asymmetric encryption

Common algorithm: RSA, ECC

- Key pairs
- Only private key owner can decrypt

✓ Easy key exchange

✗ Slow, high CPU cost

In Spring apps used for:

- JWTs 🔐
- TLS 🌐
- Digital signatures 📜

RSA

3233

RSA

$$61 \times 53 = 3233$$

Trapdoor function

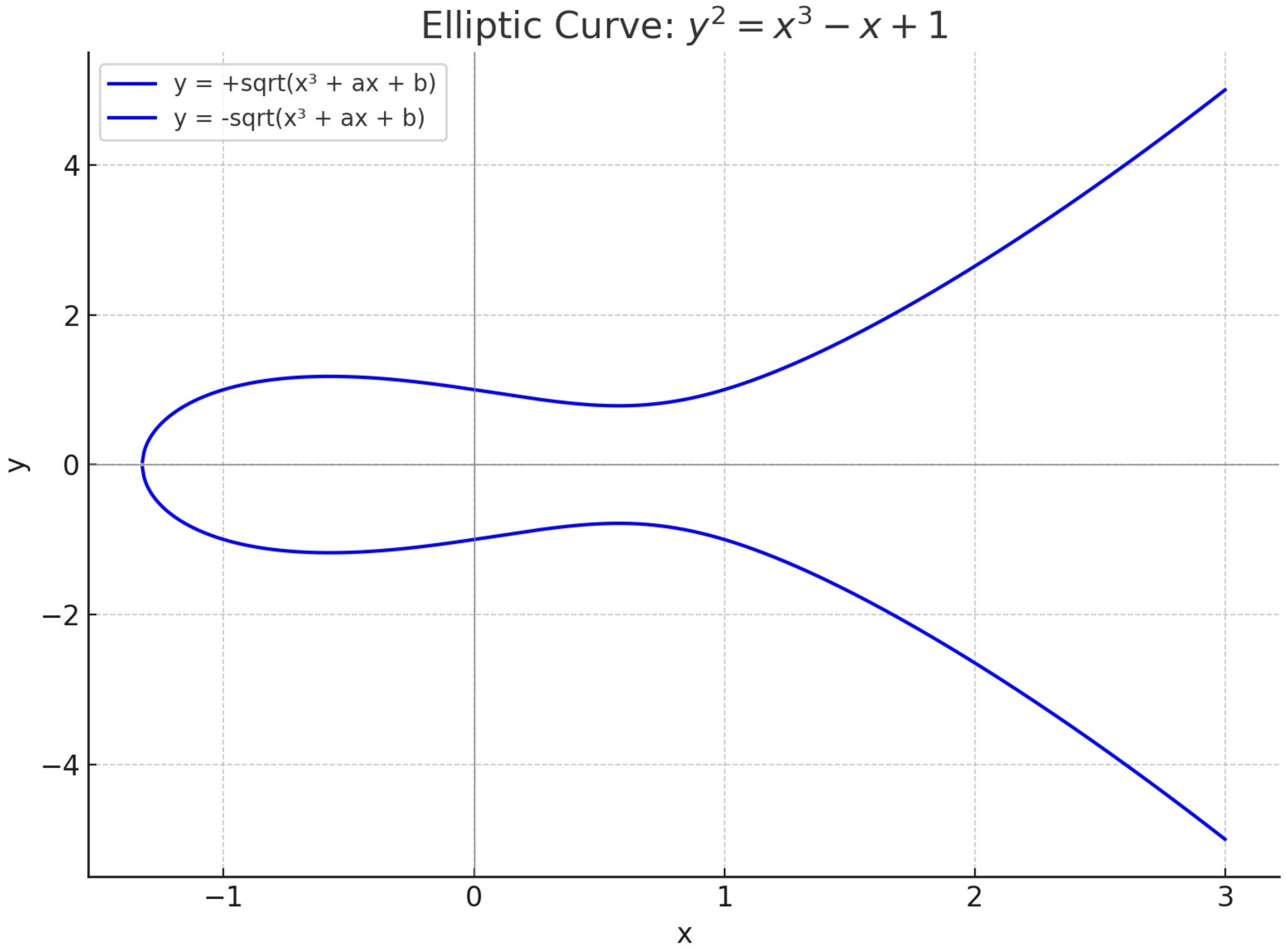
RSA

- Uses large prime numbers and modular arithmetic to generate keys.
- Based on trapdoor one-way functions: easy to multiply, hard to factor.
- Commonly used for digital signatures, TLS, and JWTs
- Keys are large (2048–4096 bits) and operations are slower than ECC.

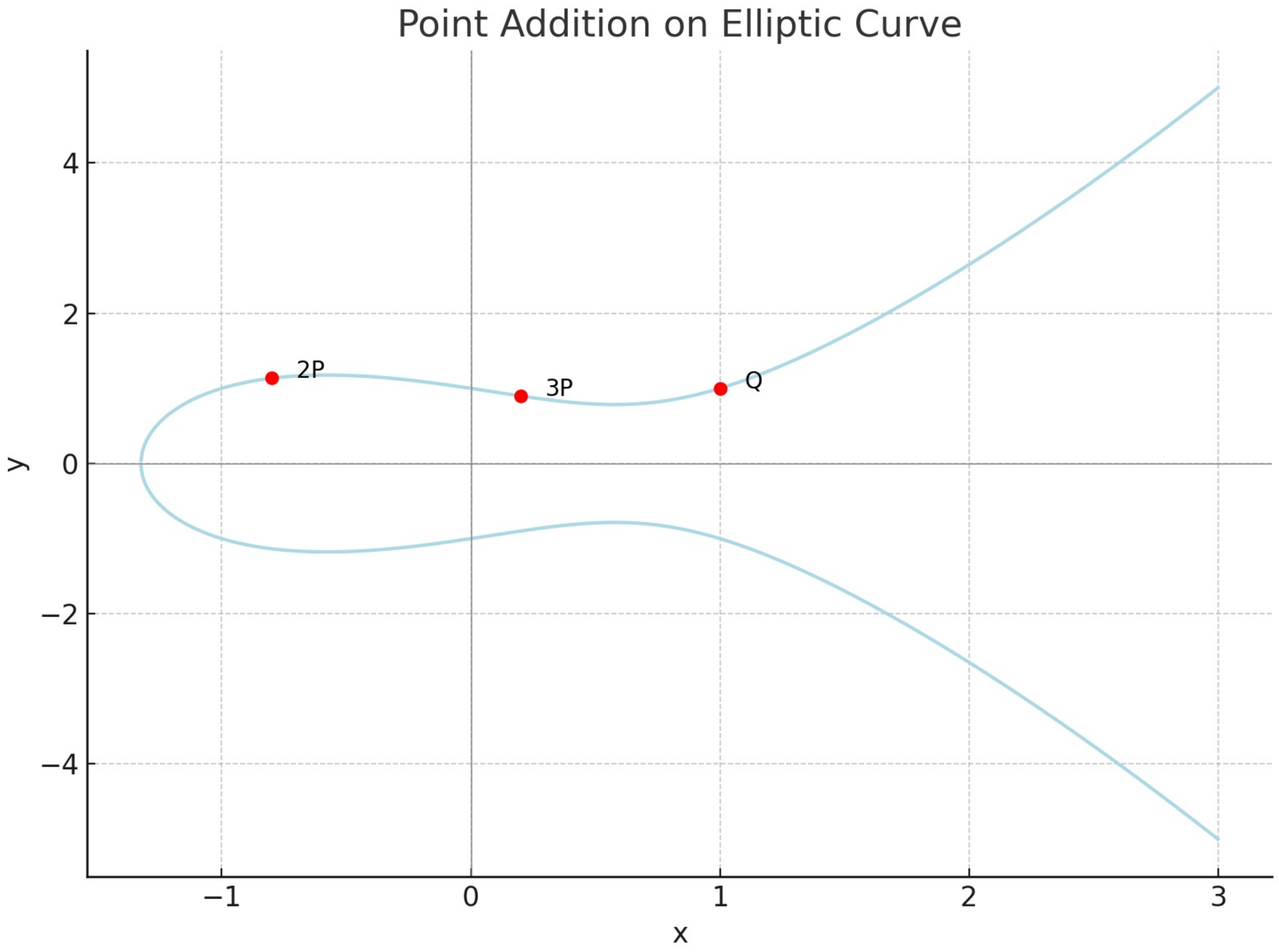
How are the prime numbers chosen?

- Large enough
- Not too close to each other
- Tested

ECC


Elliptic Curve: $y^2 = x^3 - x + 1$
y = +sqrt(x³ + ax + b)
y = -sqrt(x³ + ax + b)
4
2
0
−2
−4
y
−1
0
1
2
3
x

ECC – How many times was P added to get Q?



ECC

- Based on elliptic curve mathematics over finite fields.
- Provides stronger security with smaller keys
- Faster and more efficient ideal for high-performance systems.
- Commonly used in JWTs, TLS, SSH, and blockchain systems.

Asymmetric encryption in practice

- Securing communication channels
- Digital signatures

Securing password handling

- **Never store raw passwords, not even encrypted!**
- **Use one-way hashing algorithms: BCrypt, Argon2, PBKDF2**

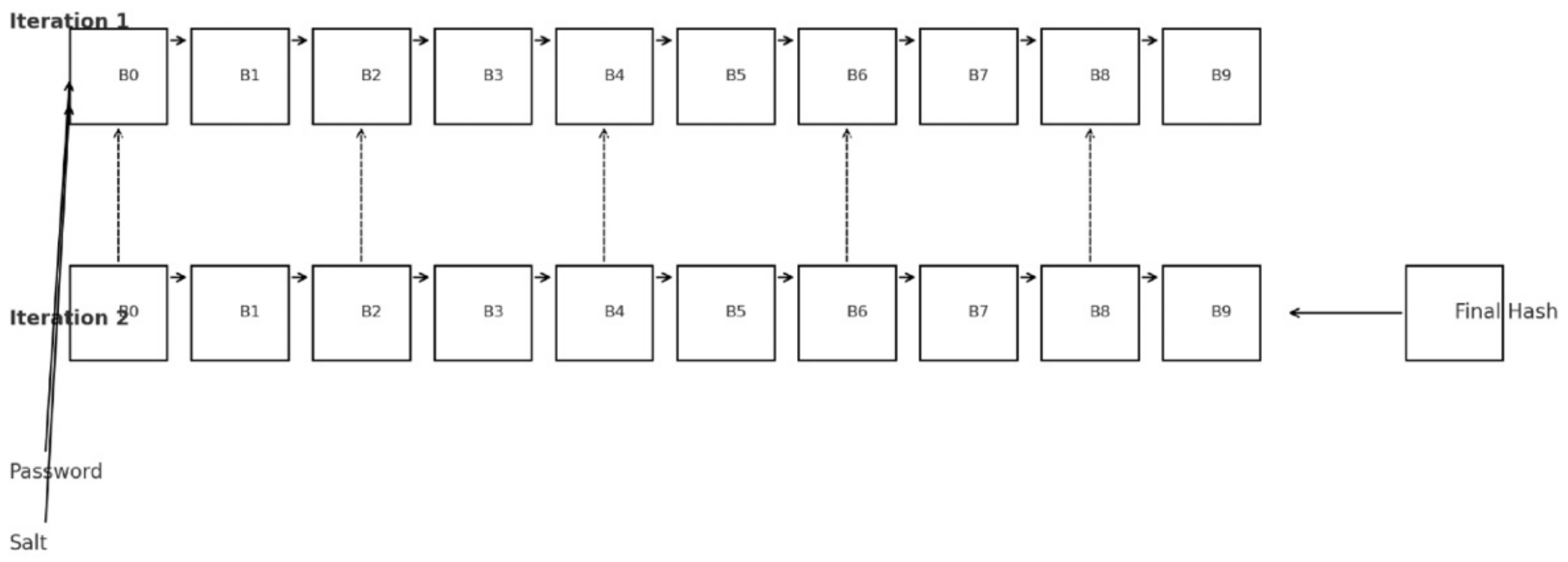
BCrypt is **default** in Spring Security

BCrypt

1. Generate a random salt
2. Combine it with the input before hashing
3. Use Blowfish key extension mechanism
4. Apply multiple (configurable) rounds -> cost // new BCryptPasswordEncoder(12);

$2A$10$<22-CHARACTER-SALT><31-CHARACTER-HASH>

Argon2



1. **Allocate a memory buffer divided into blocks**
2. **Fill each block with pseudo-random data.**
3. **Repeat the process multiple times.**
4. **Combine all memory blocks into a final hash output.**

```
PasswordEncoder encoder = new Argon2PasswordEncoder(
    16,   // salt length
    32,   // hash length
    1,    // parallelism (threads)
    65536, // memory (in KB)
    4     // iterations
);
```

Fact

Argon2 won the Password Hashing Competition (PHC) in 2015.
It can be tuned for time, memory, and parallelism

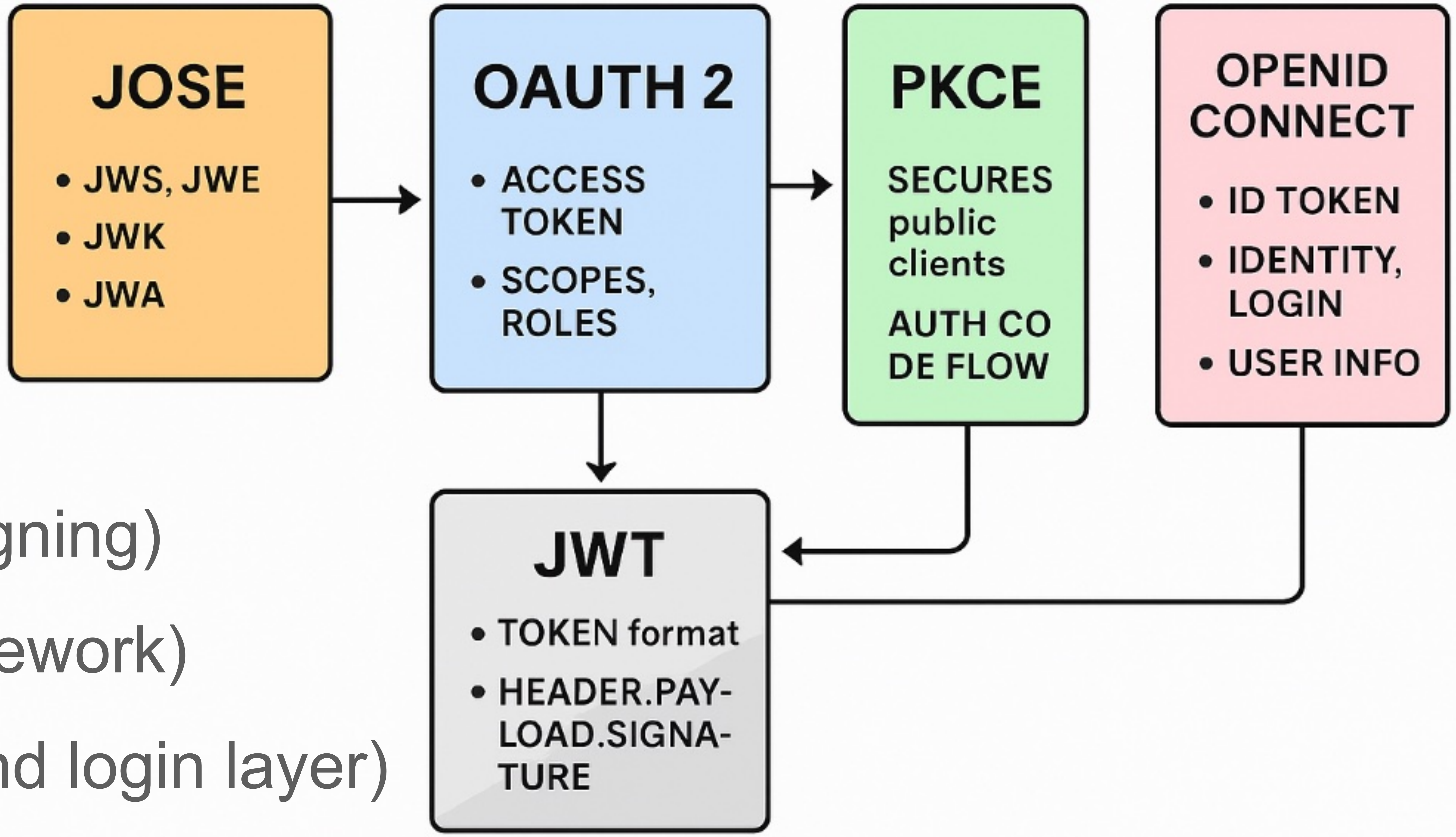
SHA

✓ Good at checking data integrity and digital signatures

✗ Bad for password hashing

Specifications

- JOSE (token encryption / signing)
- OAuth 2 (authorization framework)
- OpenID Connect (identity and login layer)
- JWT (token format)
- PKCE (securing public clients' authentication)
- TLS/mTLS/X.509 (securing communications)



JOSE

- JWS, JWE
- JWK
- JWA

OAUTH 2

- ACCESS TOKEN
- SCOPES, ROLES

PKCE

SECURES public clients

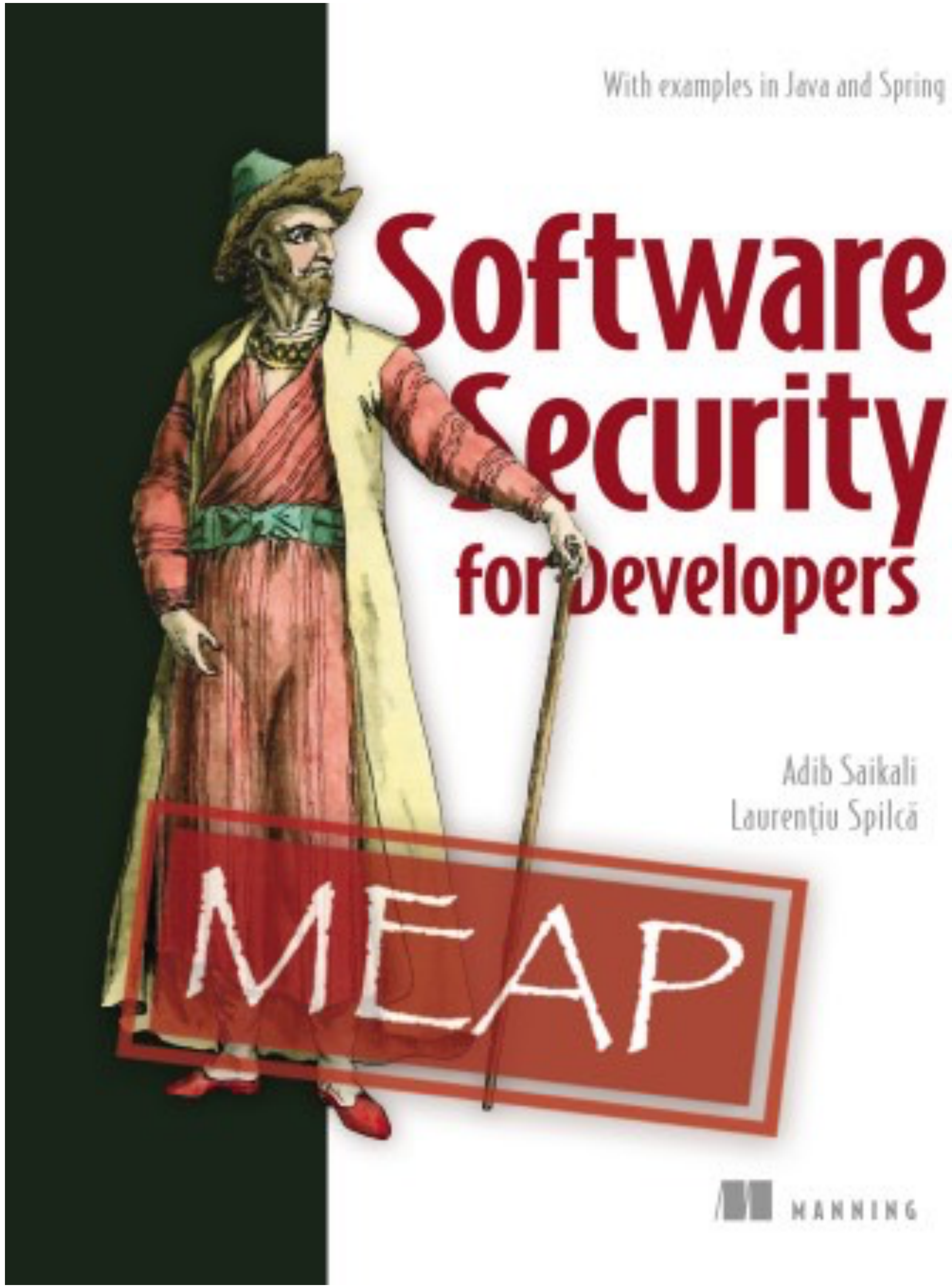
AUTH CO DE FLOW

OPENID CONNECT

- ID TOKEN
- IDENTITY, LOGIN
- USER INFO

JWT

- TOKEN format
- HEADER.PAYLOAD.SIGNATURE



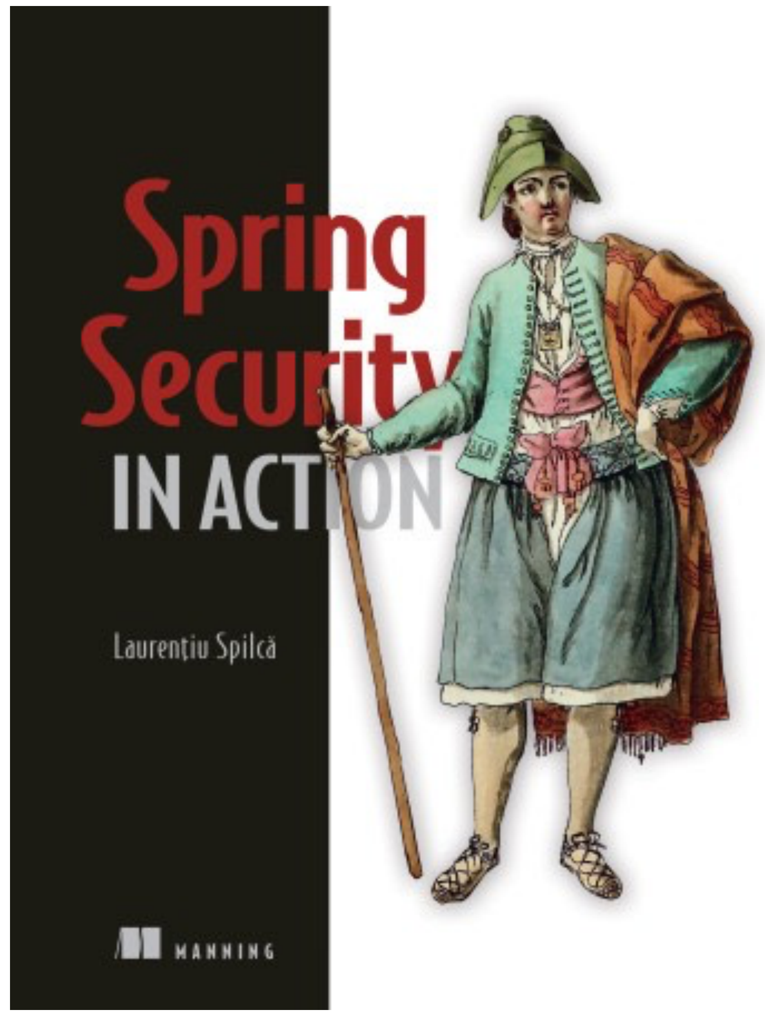
With examples in Java and Spring

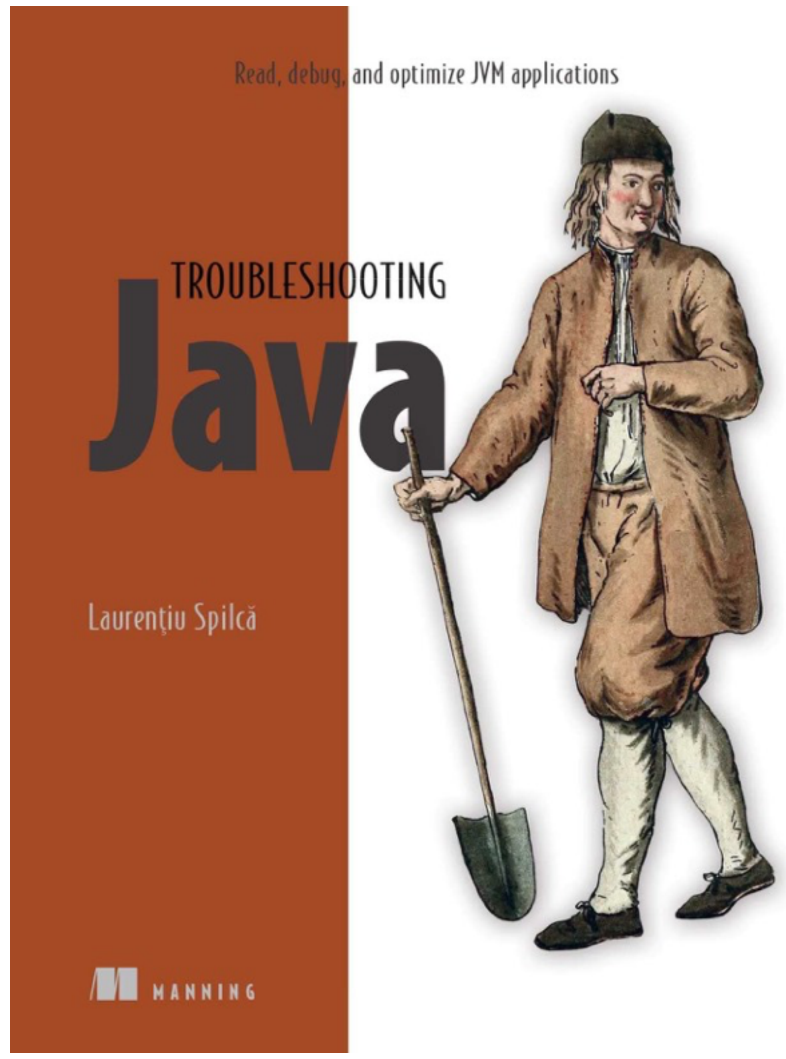
Software Security for Developers

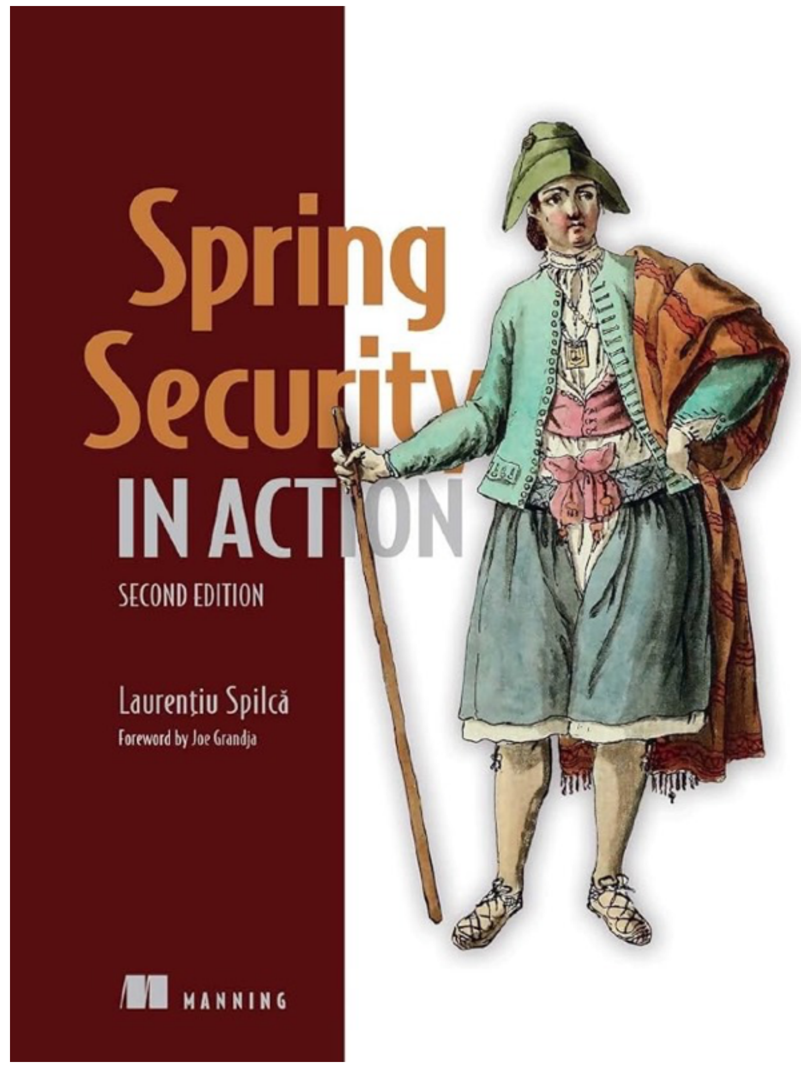
Adib Saikali
Laurenţiu Spilcă

MEAP

MANNING







youtube.com/@laurspilca

@laurspilca

Laurentiu Spilca