LABs

DI-FCT-UNL
Computer Networks and Systems Security
Segurança de Sistemas e Redes de Computadores
2019/2029

Symmetric Key Managament with keystores

And

Password-Based Encryption (PBEncryption)

Topics and hands-on

- Symmetric Algorithms and Key Generation
  - Key management with keystores
- Password-Based Encryption
- PBEncryption Scheme and Parameters
  - Salts + Counters
  - PBEEncryption with and without parameters

LABs

From the last examples ...

- Key generation/management/storage/use
  How to manage this with keystores ... ?

Key Generation for Symmetric Encryption

- Key Generation Problem / Key Generators
  - Allow the dynamic generation of keys (with pseudo-random properties)

Key Interface (base interface implemented and extended by all objects related to cryptographic keys, including symmetric keys (SecretKeySpec)

  - Key.getAlgorithm() // algorithm for which the key is generated
  - Key.getEncoded()   // key enconding
  - Key.getFormal()   // key format

Symmetric Encryption / Key Generation

LABs

- javax.crypto.KeyGenerator Class (class implementing the key generator factory)
  - KeyGenerator.getInstance()   // expliciting the algorithm
    - Ex: KeyGenerator generator= KeyGenerator.getInstance("AES, "BC");
  - KeyGenerator.Init()   // Init. , Key Size
  - KeyGenerator.generateKey()   // Generate

obj of type: javax.crypto.SecretKey

Keystores (JCEKS)

- See and learn:
  - about keytool
  - About keystores (particularly type jceks)

Keystores of jceks type:

This must be the keystore types to store/manage symmetric (secret) keys.

Today ...

LABs

- Password-Based Encryption

Password-Based Encryption (PBE)

- Key Generation from passwords, secrets or secret seeds …
- "Encryption with "something" the user "Knows" (remember …)
- Practical use:
  - Pros: Key generated for use without the exposition of the final key itself
  - But … How strong is this ?
    - Problem of Shared Secrets / Shared PWDs, Seeds, etc
    - Ex: A Strong Key (ex., AES 256 bits) will not be so strong if my password is weak (ease to be compromize by dictionary-attacks, rainbow-attacks or password-cracking tools
      - Ex., generate a 256 bit AES key from … "sporting" !!!!
    - Same problem of PWD Attacks !

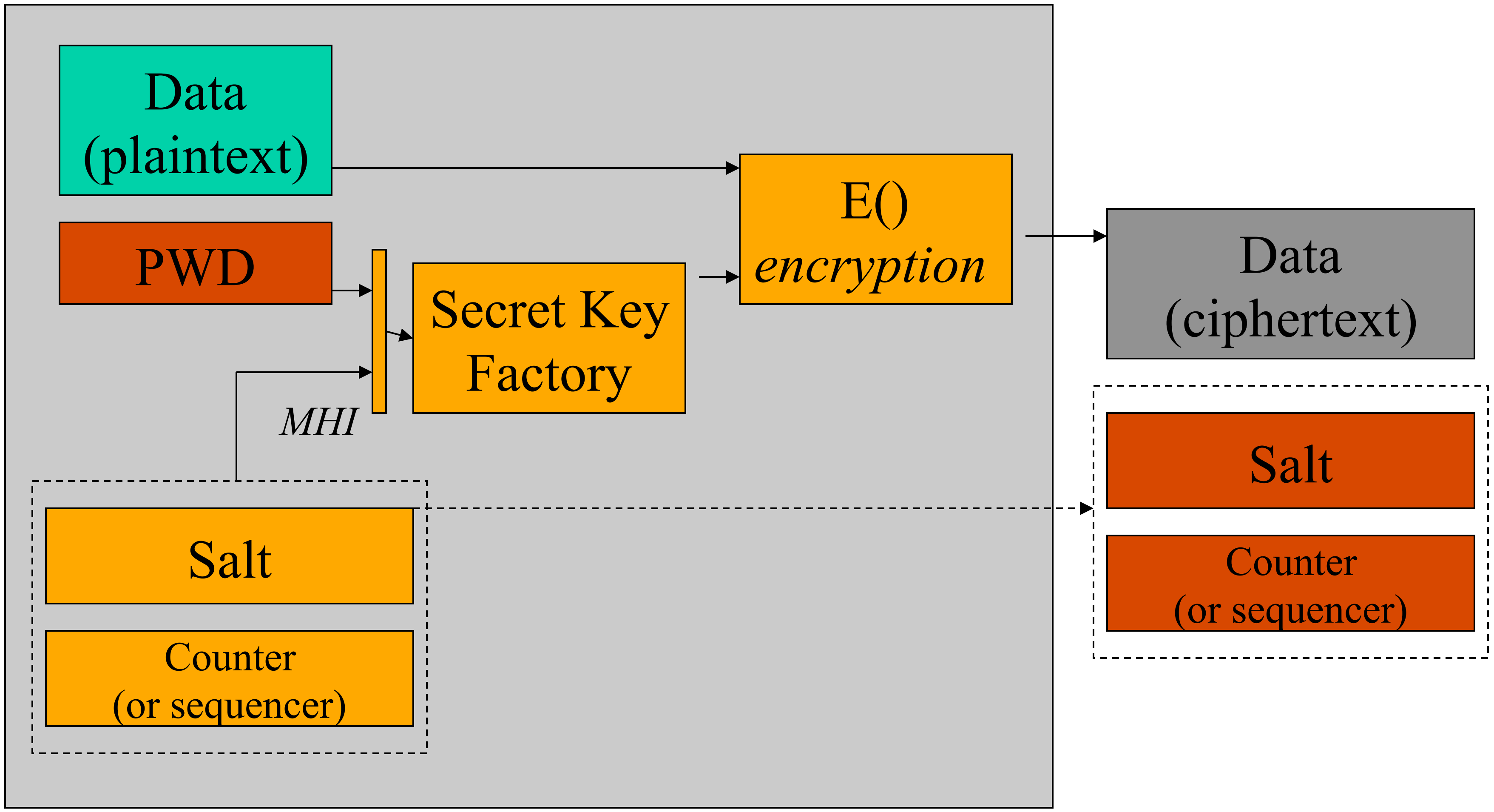
PB Encryption (PBE) in a Nutshell

- Essentially a primitive to encrypt/decrypt using Passwords
  - The PWD is used as the seed to generate a Symmetric Key
  - and the generated key is implicitly used for encryption/decryption

- Standardization for PBE Schemes
  - PKCS #5, PKCS#12
  - S/MIME Scheme (RFC 3211)

  Others
  - PGP Scheme for session keys
    (using ANSI X9.17 + CAST 128 e X.12.17)

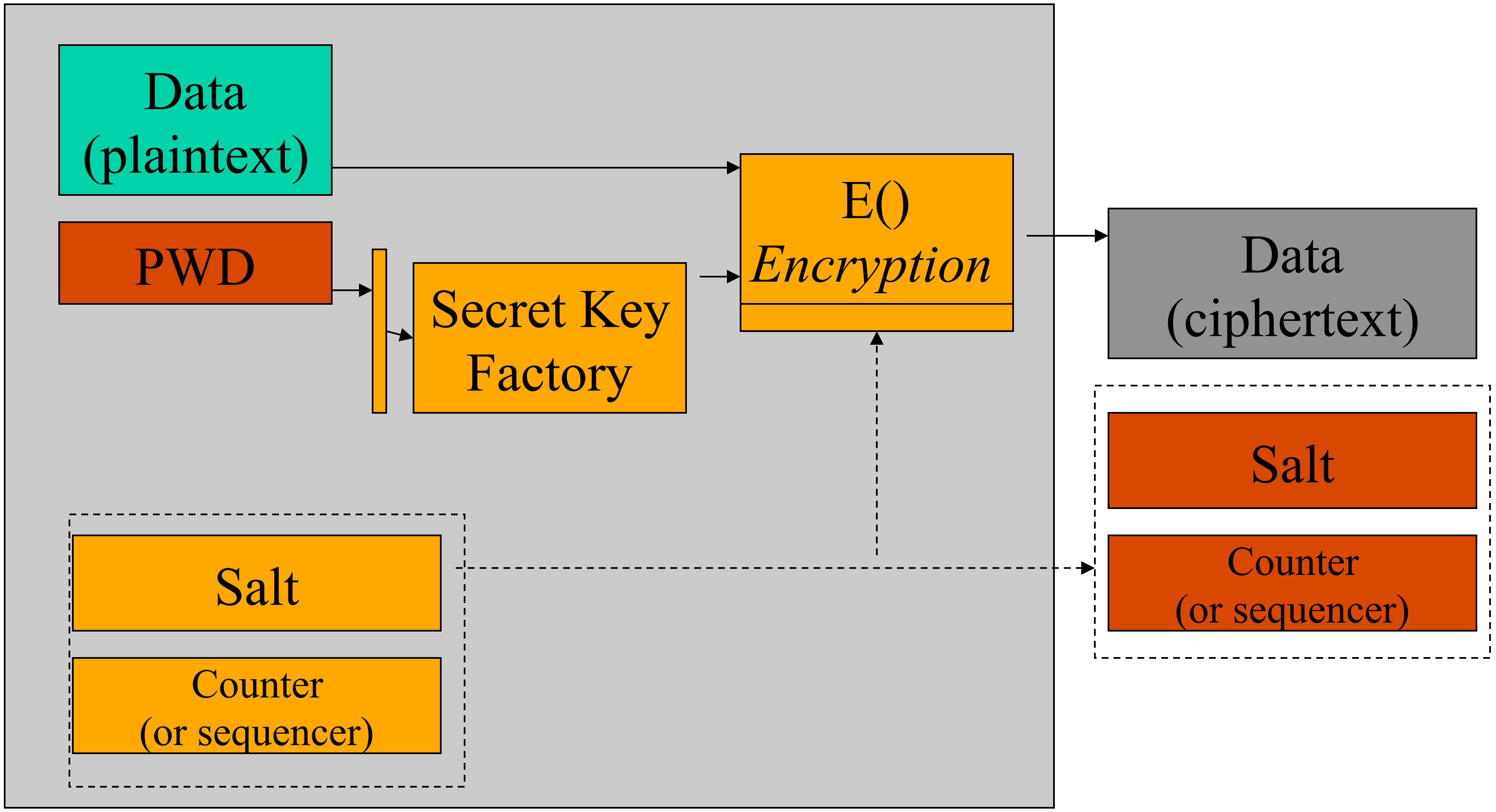
PBE Encryption Scheme



MHI-Mixing hashing pwd input: PBEKeySpec(pwd,salt,cont)
Esquema de cifra sem parametros

PBE Scheme (alternative)

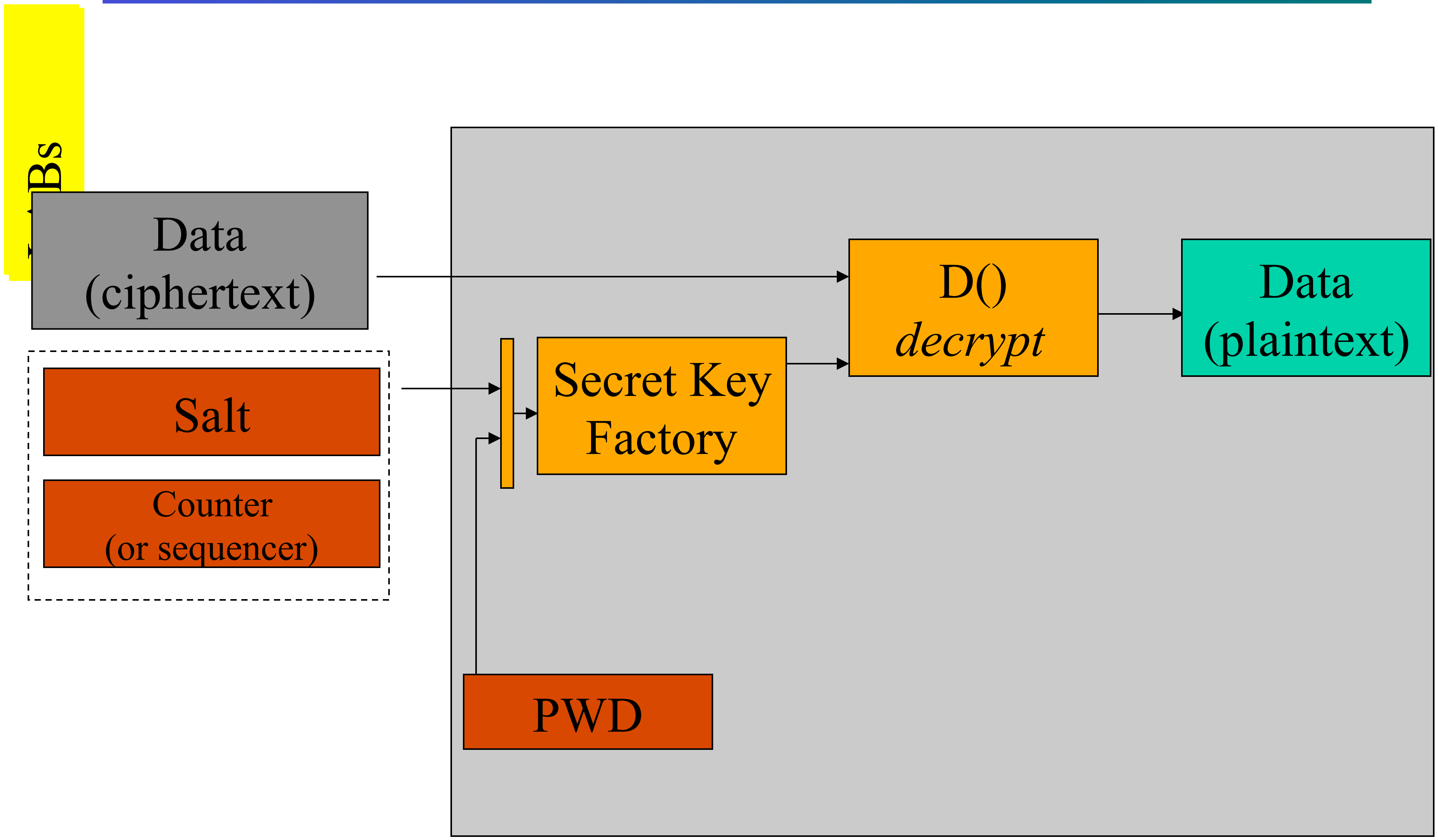
LABs



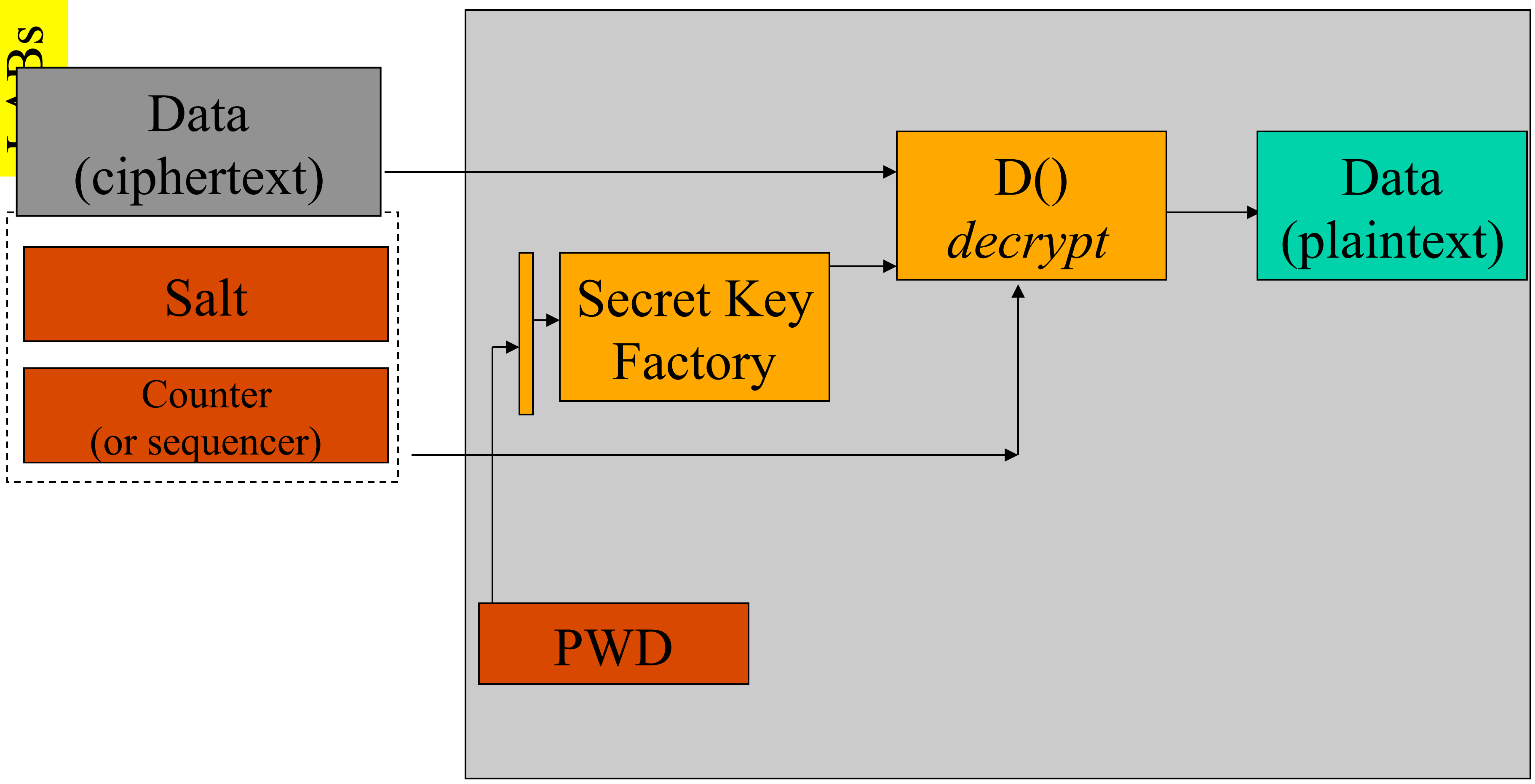
MHI-Mixing hashing pwd input: PBEKeySpec(pwd)

Esquema de cifra com geração da chave final com parametros

Esquema de referência para decifra com PBE



Esquema alternativo para decifra com PBE



PBE na framework Java JCE

- PBEParameterSpec, PBEKeySpec:
  - Classes for Key Generation and Parameters
- SecretKeyFactory: factory to generate symmetric Keys
- Cipher.getInstance: Instantiation of the PBE parameterization (ciphersuite) in the PBE scheme to use

- See examples
  - PBEWithParamsExample()
  - PBEWithoutParamsExample()

Hands-On w/ PBE Schemes

- See the Exercices (Lab)
- See ListAlgorithms (Lab 1) and see the supported PBE Schemes in your Java Framework