CSC12001
Data Security in Information System

Database Encryption

PhD. Phạm Thị Bạch Huệ - ptbhue@fit.hcmus.edu.vn
M.S. Lương Vĩ Minh – lvminh@fit.hcmus.edu.vn

fit@hcmus
KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Agenda

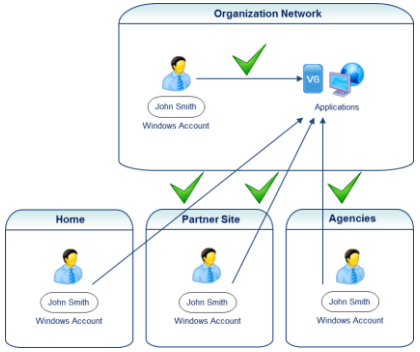
1. Introduction
2. Data Encryption Level
3. Review Database Encryption Solution
4. Database Encryption Issues
5. Stored-Encrypted Data Model
6. Implementation in an DBMS

fit@hcmus

Introduction

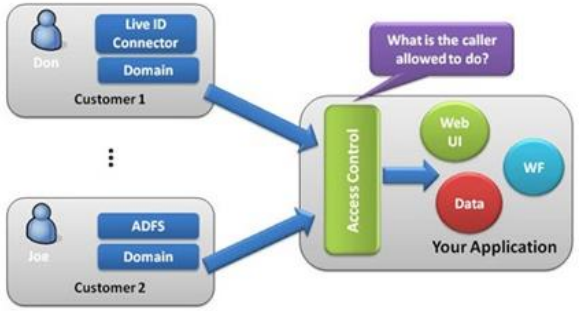
Database Encryption

Introduction



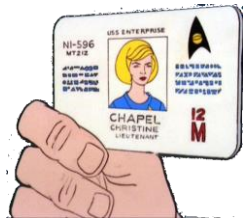
Authentication

Xác thực người dùng



Access Control

Điều khiển truy cập



Identification

Định danh người dùng

Audting

Giám sát hoạt động

Data Encryption

Mã hóa dữ liệu

Data Encryption

- Data Encryption: Hiding data by converting it to the no-sensible to the attackers.
- Final barriers to the data attacker when they are bypass other data protection mechanisms.

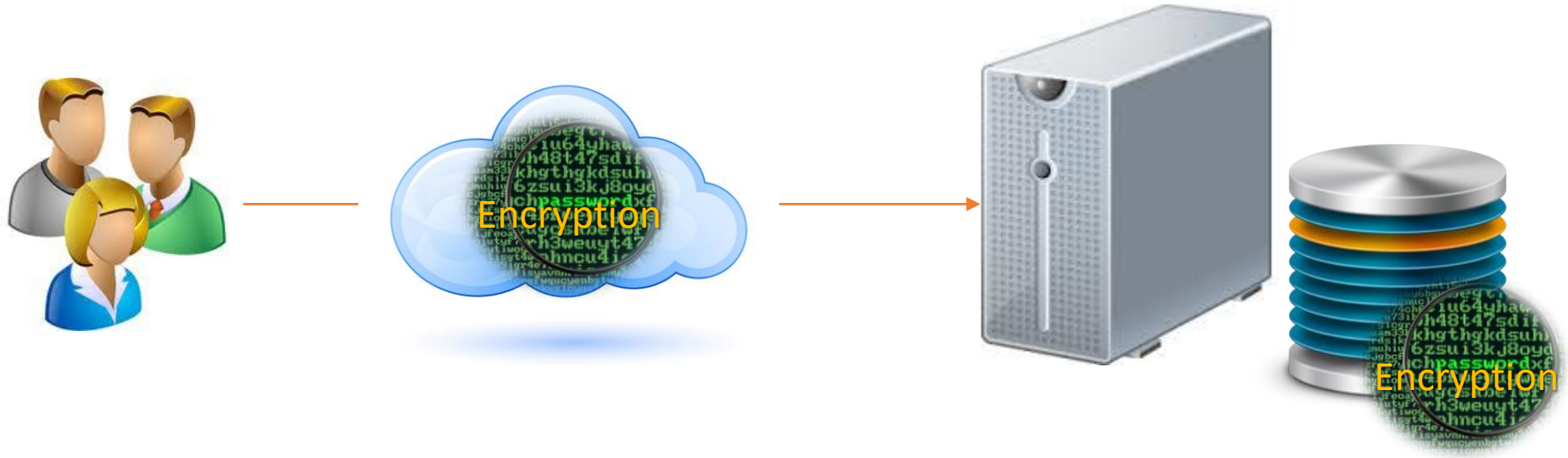
Meaning Data →

Encryption

Mã hóa dữ liệu

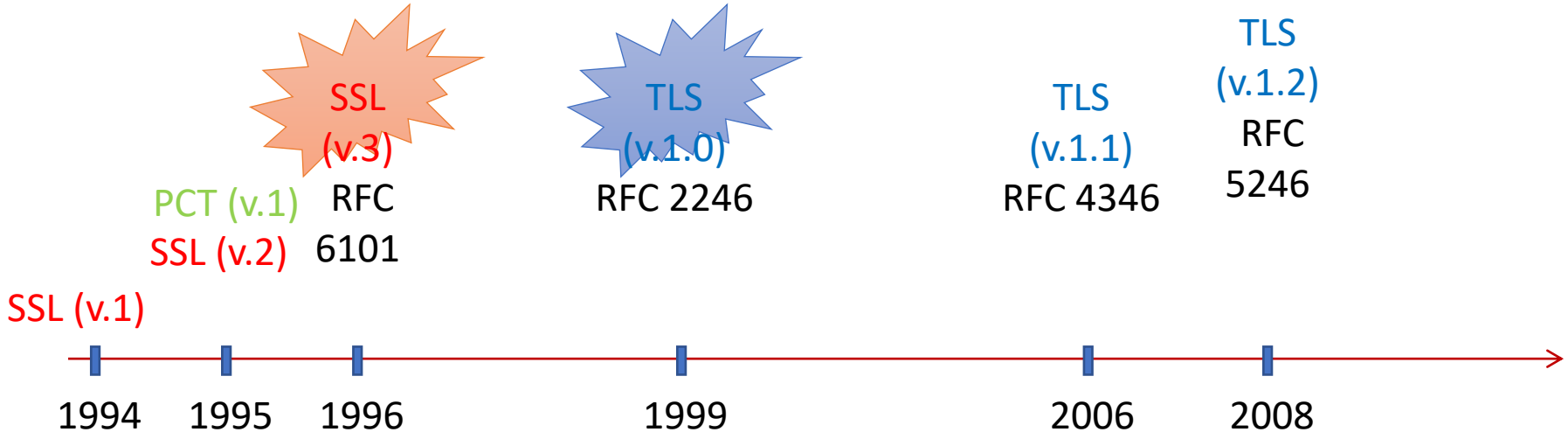
Data Encryption

- Data Encryption can be implemented in:



Data Encryption on Transmission

- SSL (Secure Socket Layer) – Netscape
- PCT (Private Commnunication Technology) – Microsoft
- TLS (Transport Layer Security) - IETF (Internet Engineering Task Force)



Protection data during transition is very important.
However, most of data attacking occur at the end-pont of the data storage.

Data Encryption Methods

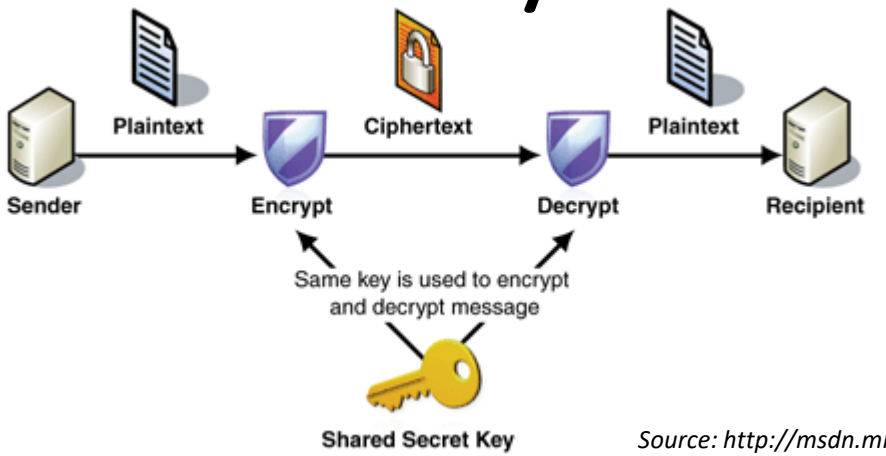
- Cryptograpy:
  - Symmetric Cryptography
  - Asymmetric Cryptography
  - Hybrid Cryptography
  - Cryptography Hash function

Meaning Data

Encryption

Mã hóa dữ liệu

Symmetric Cryptograhy



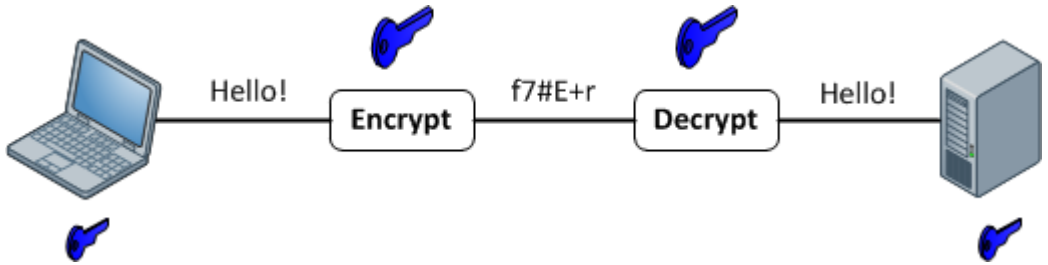
Source: http://msdn.microsoft.com/en-us/library/ff650720.aspx

- Symmetric Cryptography
- Using a Shared Secret Key to encrypt and decrypt data.
- Algorithm simple, short key length → Fast processing speed, suitable for large amount of security data.

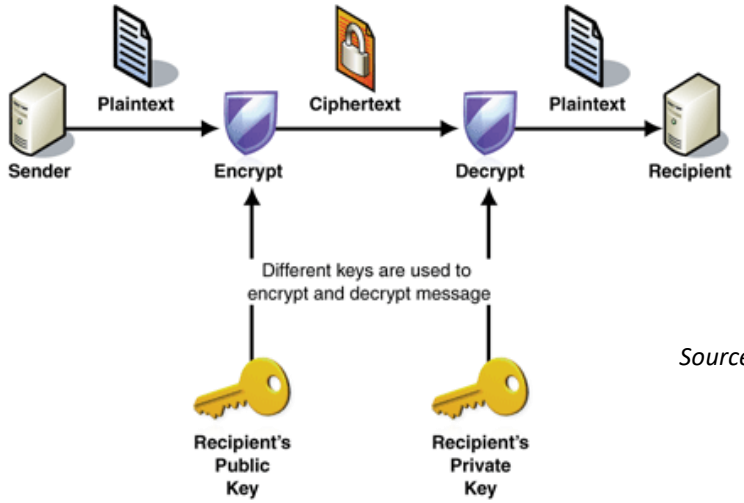
- Difficulty in Secret key distribution → Need a secured key management system (key generation, key distribution, key storage, key renew, and key lifetime management)
- NOT provide resistance to repudiation of responsibility; Can't prove who really sent the data.

Symmetric Cryptograhy

- **Common algorithms:**
  - Block Cipher:
    - Data Encryption Standard (DES).
    - Triple Data Encryption Standard (3DES).
    - Advanced Encryption Standard (AES - Rijndael).
    - BlowFish, TwoFish, Serpent
  - Stream Cipher:
    - RC4



Asymmetric Cryptograhy



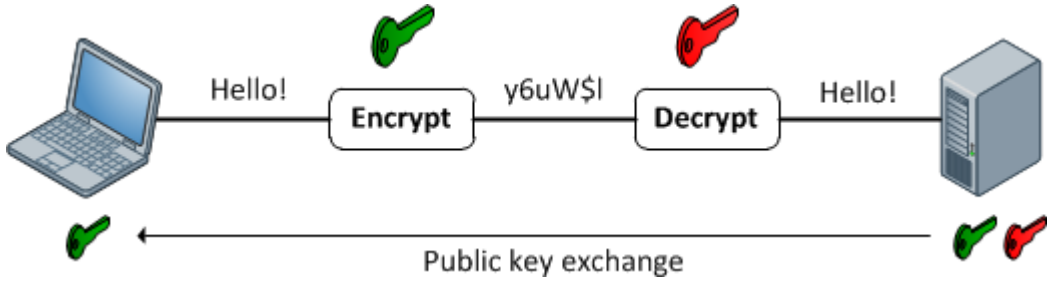
Source: http://msdn.microsoft.com/en-us/library/ff650720.aspx

- Asymmetric Cryptography (Public-key Cryptography)
- Using a key-pair: Public key (encrypt data) and Private key (decrypt data).
- Solve the key distribution problem.

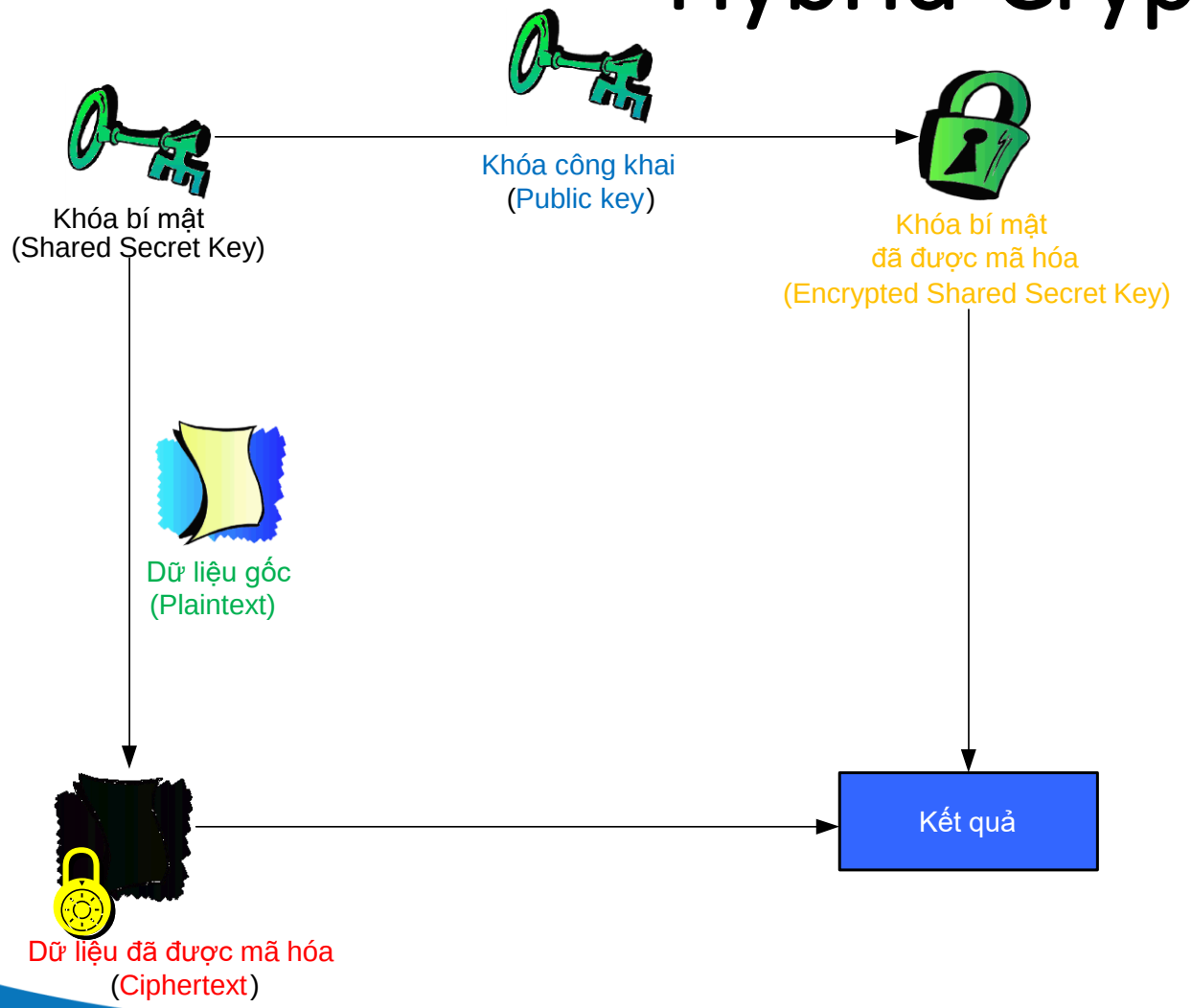
- Complex algorithms → Slow processing speed.
- Suitable to protect small-size data.

Asymmetric Cryptograhy

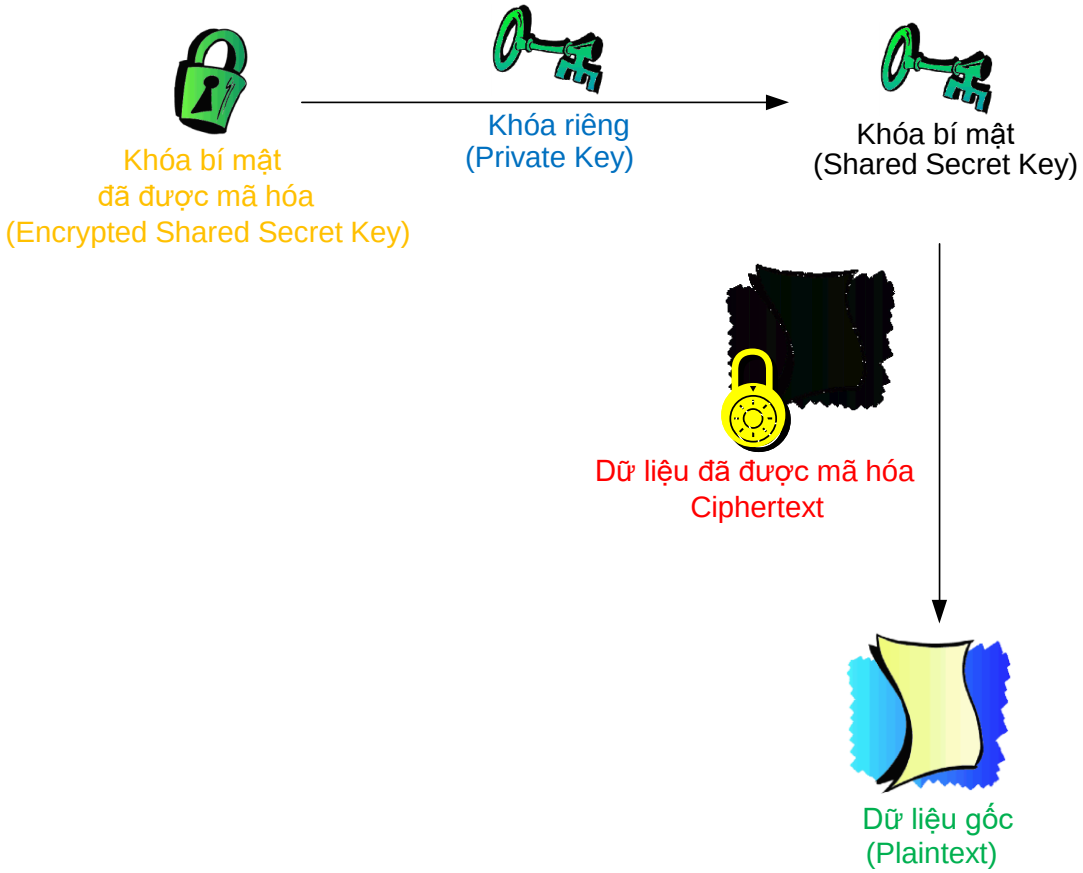
- **Common algorithms:**
  - Diffie-Hellman key exchange
  - Rivest-Shamir-Adleman (RSA)
  - Digital Signature Algorithms (DSA)
  - ElGamal
  - Elliptic Curve Cryptography (ECC)
  - Paillier cryptosystem



Hybrid-Cryptography



Hybrid-Cryptography



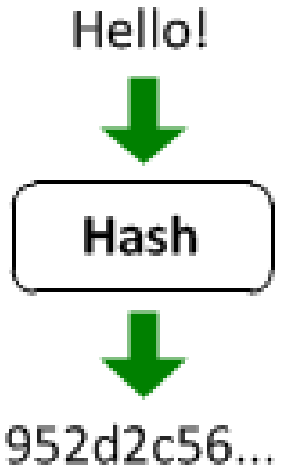
Hybrid-Cryptography

- By combining symmetric and asymmetric cryptography
- Get the Advantages:
  - Fast speed process of symmetric cryptography.
  - Secure key distribution of asymmetric cryptography

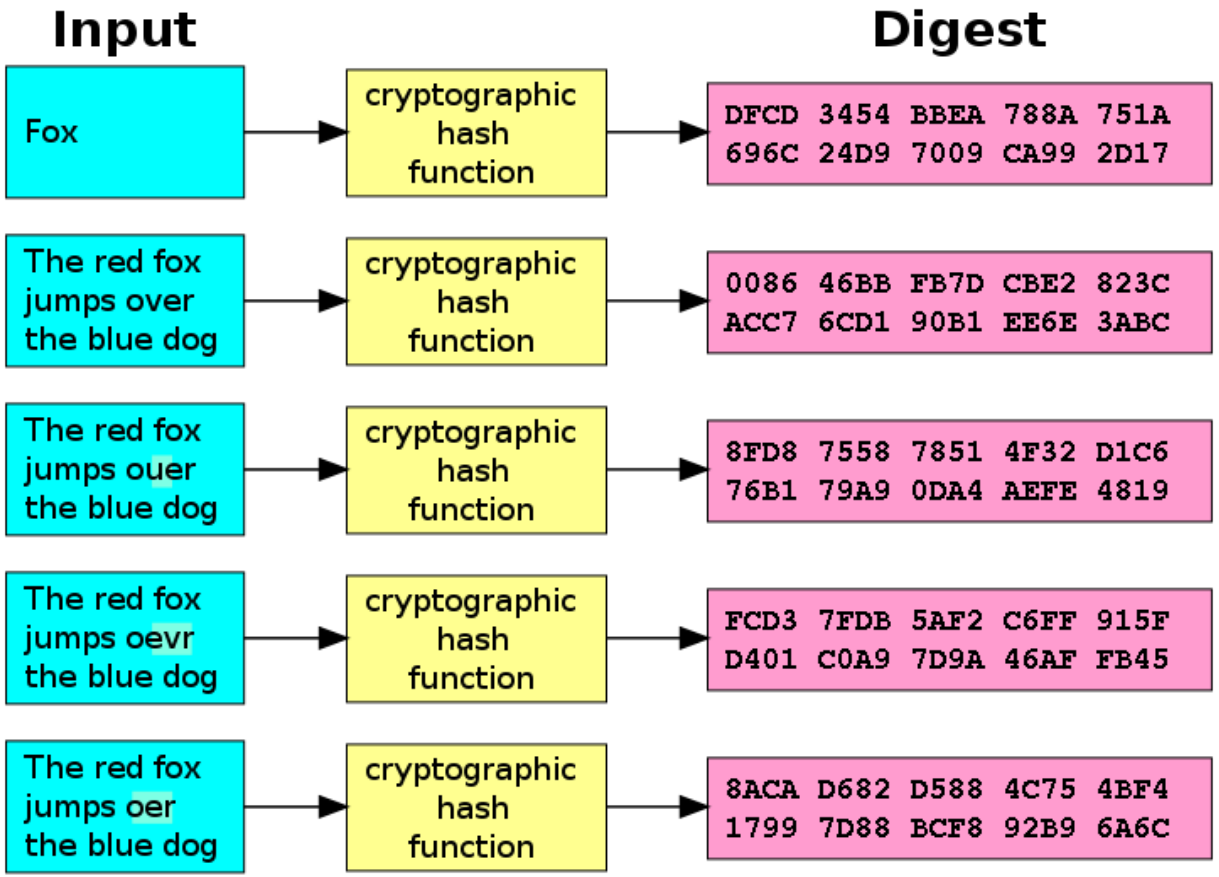
Cryptographic Hash Function

- Cryptographic Hash Function
- One-way Cryptography function without decryption after hashing (encrypt) data.
- Hashed data has the same size (depend on each hash algorithms)
- Hashing Collision of hash functions are very rare.

- Used to authorization the data modification.
- Common algorithms:
  - MD5
  - SHA-0, SHA-1, SHA-224, SHA-256, SHA-512



Cryptographic Hash Function



fit@hcmus

Data Encryption Level

Database Encryption

Data Encryption Level

1. Application Level
2. Storage Level
3. Database Level

Data Encryption Level - Application Level

- Data Encryption and Descryption are executed inside the application.
- Suitable to the grant permission application.
- Using the cryptography library of programming language: JCE (Java-based application) hoặc MS-CAPI (Microsoft-based application)

Data Encryption Level - Application Level

- Protect data to rishes:
  - Storage device stolen
  - Prevent data storage level attack
  - Prevent database administrator to access the sensity data.

- Limitation:
  - Data in database cannot be shared to other application.
  - To sharing data, need to change the encrypt data key.

Data Encryption Level - Storage Level

- Encrypt and Decrypt database files with an unique secret key.
- Implemeted at Operation System level.

- Suitable to protect storage data file and offline data files.
- Protect data in the stolen storage device.

- Many commercial applications implement the data encryption at storage level in their application.

Data Encryption Level - Storage Level

- Limitations:
  - Cannot select data in database to protect.
  - Cannot grant access control in the smaller unit data (table, rows, column)
  - Cannot protect data to attacks in the application level or database level.
  - Cannot prevent System Administrator to access to the encrypted files.
  - Cannot prevent to access the encrypted files when the operation system is lost control.
  - Cause the performance problem when reading / writing data to the encrypted files database.

Data Encryption Level - Database Level

- Encryption and Decryption data is processed in the DBMS level.
- Using stored-procedure / trigger to implement.
- Can select data to protect at the different data unit (table, row, column, value, all database, …).
- Easy to share encrypted data between different application.
- Prevent the attacks: stolen storage devices, SQL injection, DBA Access…

Data Encryption Level - Database Level

Database Encryption Levels:

- Attribute value (cấp độ giá trị thuộc tính)
- Record/Row level (cấp độ bộ/dòng)
- Column/Atribute level (cấp độ cột/thuộc tính)
- Page/Block level (cấp độ trang/khối)

Data Encryption Level - Database Level

Limitations:

- When change the data type / data size of the encrypted attributes → Need to change the stored procedure / trigger at once.
- Slow the database system significantly due to time consuming for encrypting / decryption data.
- NOT prevent to attack at the application level.

fit@hcmus

Review Database Encryption Solution

Database Encryption

Review Database Encryption Solution

Pros:

- Encrypt data in Database can prevent / hide data from the intruders, even the DBA (if they are not authorized to access the data).
- Database Encryption is the efficiently method to protect data to the attack at storage level.

Review Database Encryption Solution

Cons:

- Increase the amount of processing when accessing to the data. Moreover, increasing the data storage capacity.
- Limit the DBMS to implement the basic data access methods.
- Requires proper key management policy
- Secret Key is the most import component:
  - Lost key → data will be revealed / exposed.
  - Lost key → data cannot be descrypted.

Review Database Encryption Solution

Thuật toán	100 bytes x 100.000 lần mã hóa	120 bytes x 83.333 lần mã hóa	16 KB x 625 thao tác mã hóa
AES (16B)	365 ms	334 ms	194 ms
DES (8B)	327 ms	354 ms	229 ms
Blowfish (8B)	5280 ms	4409 ms	170 ms

Linux, 2.8 Ghz PIV, 1Gbyte RAM + thư viên OpenSSL

Source: Bala Iyer, Sharad Mehrotra, Einar Mykletun, Gene Tsudik2, and Yonghua Wu, A Framework for Efficient Storage Security in RDBMS, EDBT 2004, LNCS 2992, pp. 147–164, 2004

Conclusion

Database Encryption is not the best solution to protect data in database, due to:

1. Database Encryption cannot implement data access control.
2. Database Encryption must not effect the outcome of data access control.
   Example: A có quyền SELECT trên bảng NHANVIEN thì khi mã hóa xong A không bị ngăn cản dữ liệu mà A được phép xem
3. Encrypt entire database is not a solution to protect data in Database level.

Database Encryption Issues

Database Encryption

Encrypt to Primary key, Foreign Key and Constraints data Issue

- Data in Primary key is sentitive → Encrypt Primary key data.

- **Solution:** Encrypt data at the primary column by using:
  - Same secret key + same IV (initial vector).
  - Different secret key for each data rows.
  - Same secret key + Different IV for each data rows.

- What is the issue when encrypt data at Primary key column?

Encrypt to Primary key, Foreign Key and Constraints data Issue

- Violation the PK constraint (when using different key at each row or same secret key with the different IV)
  → Remove the PK Constraint + install self-check procedure.

- Violation the FK constraint
  → Encrypt the value at FK with the same secret key & IV

- Violation to the other constraints on FK and PK
  → Remove all constraint. Then, encrypt data and re-create constraint.

- Existing integrity constraints cannot be fulfilled (due to the nature of data)
  → Install the self-check procedure / function / trigger.

Indexing in Encrypted Data Issue

- Database Indexing → Speed up the search function in database.
- If data in the indexing field need to protect, two cases need to be addressed:
  - Indexing for the encrypted data.
  - Indexing the data before encrypting.
- Some DBMS discourage indexing the encrypted data because somehow, the DBMS require to descrypt all encrypted data for searching.

Query in Encrypted data Issue

- Exact searching issue (=, in, not in, ..) → use the same secret key and IV for the searched keyword and the encrypted data in database.
- Approximated searching issue (like, >, <, …) → Usually, require to lookup all data if not using the indexing mechanisms.
- **Solution:** apply the cryptography hash function to the partial sentitive data and store it to the same row in the difference column.

Implementation in a DBMS

Database Encryption

Read the Oracle Reference Document

DBMS_CRYPTO

- Consist cryptography functions/procedures.
- Work with basically oracle data types, including RAW and LOB (used to store images/sound data).
- Support BLOB and CLOB with many encoding.
- Algorithms:
  - Data Encryption Standard (DES), Triple DES (3DES, 2-key and 3-key)
  - Advanced Encryption Standard (AES)
  - MD5, MD4, and SHA-1 cryptographic hashes
  - MD5 and SHA-1 Message Authentication Code (MAC)
- Block Cipher modifier:
  - Padding options: có PKCS (Public Key Cryptographic Standard) #5
- Four block cipher chaining modes: có Cipher Block Chaining (CBC).

DBMS_CRYPTO & DBMS_OBFUSCATION_TOOLKIT

Package Feature	DBMS_CRYPTO	DBMS_OBFUSCATION_TOOLKIT
Cryptographic algorithms	DES, 3DES, AES, RC4, 3DES_2KEY	DES, 3DES
Padding forms	PKCS5, zeroes	none supported
Block cipher chaining modes	CBC, CFB, ECB, OFB	CBC
Cryptographic hash algorithms	MD5, SHA-1, MD4	MD5
Keyed hash (MAC) algorithms	HMAC_MD5, HMAC_SH1	none supported
Cryptographic pseudo-random number generator	RAW, NUMBER, BINARY_INTEGER	RAW, VARCHAR2
Database types	RAW, CLOB, BLOB	RAW, VARCHAR2

DBMS_CRYPTO: Data Types

Type	Description
BLOB	A source or destination binary LOB
CLOB	A source or destination character LOB (excluding NCLOB)
PLS_INTEGER	Specifies a cryptographic algorithm type (used with BLOB, CLOB, and RAW datatypes)
RAW	A source or destination RAW buffer

DBMS_CRYPTO: Cryptoraphy Hash Functions

- Input data is RAW or LOB data.
- Encrypted data length is 128 bit.
- Used to hash data to verify the data modification.

Name	Description
HASH_MD4	Produces a 128-bit hash, or message digest of the input message
HASH_MD5	Also produces a 128-bit hash, but is more complex than MD4
HASH_SH1	Secure Hash Algorithm (SHA). Produces a 160-bit hash.

DBMS_CRYPTO: MAC Functions

- MAC is one-way cryptography hash function, but it needs a secret key.
- Used to authorize when transferring files.

Name	Description
HMAC_MD5	Same as MD5 hash function, except it requires a secret key to verify the hash value.
HMAC_SH1	Same as SHA hash function, except it requires a secret key to verify the hash value.

DBMS_CRYPTO: Cryptography functions

Name	Description
ENCRYPT_DES	Data Encryption Standard. Block cipher. Uses key length of 56 bits.
ENCRYPT_3DES_2KEY	Data Encryption Standard. Block cipher. Operates on a block 3 times with 2 keys. Effective key length of 112 bits.
ENCRYPT_3DES	Data Encryption Standard. Block cipher. Operates on a block 3 times.
ENCRYPT_AES128	Advanced Encryption Standard. Block cipher. Uses 128-bit key size.
ENCRYPT_AES192	Advanced Encryption Standard. Block cipher. Uses 192-bit key size.
ENCRYPT_AES256	Advanced Encryption Standard. Block cipher. Uses 256-bit key size.
ENCRYPT_RC4	Stream cipher. Uses a secret, randomly generated key unique to each session.

DBMS_CRYPTO: Cryptography functions

Name	Description
DES_CBC_PKCS5	ENCRYPT_DES + CHAIN_CBC + PAD_PKCS5
DES3_CBC_PKCS5	ENCRYPT_3DES + CHAIN_CBC + PAD_PKCS5

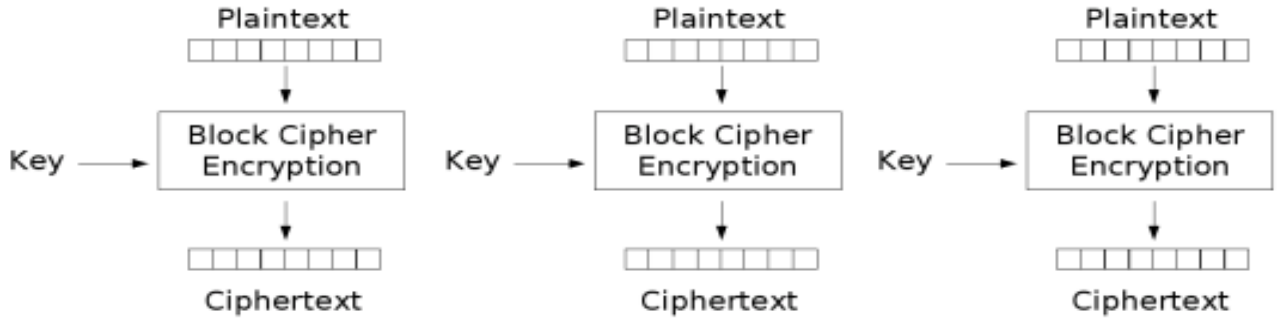
DBMS_CRYPTO: Cryptography functions

Name	Description
CHAIN_ECB	Electronic Codebook. Encrypts each plaintext block independently.
CHAIN_CBC	Cipher Block Chaining. Plaintext is XORed with the previous ciphertext block before it is encrypted.
CHAIN_CFB	Cipher-Feedback. Enables encrypting units of data smaller than the block size.
CHAIN_OFB	Output-Feedback. Enables running a block cipher as a synchronous stream cipher. Similar to CFB, except that *n* bits of the previous output block are moved into the right-most positions of the data queue waiting to be encrypted.

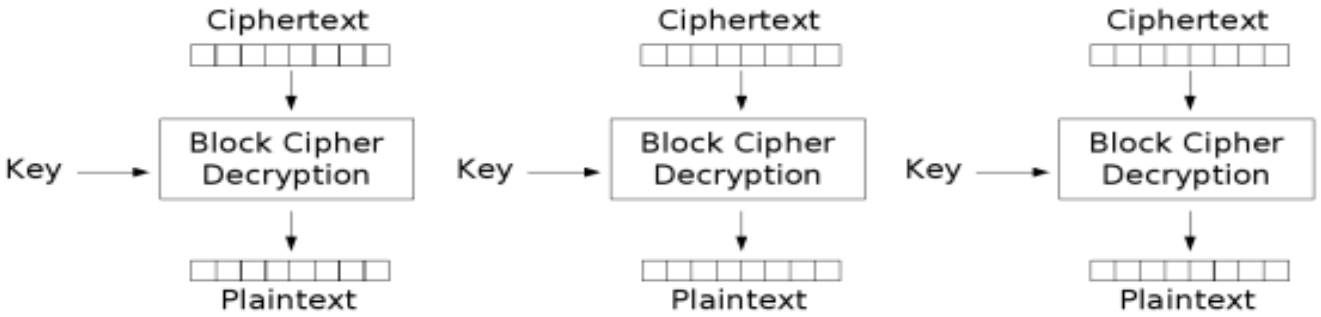
DBMS_CRYPTO: Cryptography functions

Name	Description
PAD_PKCS5	Provides padding which complies with the PKCS #5: Password-Based Cryptography Standard
PAD_NONE	Provides option to specify no padding. Caller must ensure that blocksize is correct, else the package returns an error.
PAD_ZERO	Provides padding consisting of zeroes.

ECB

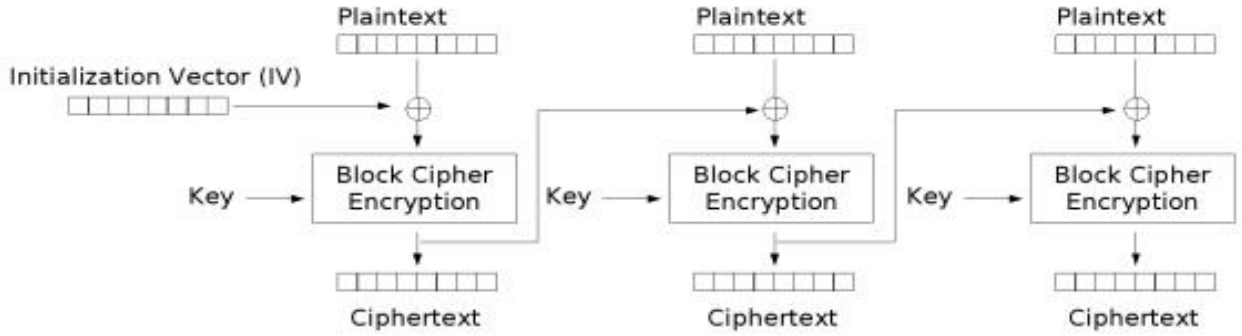


Electronic Codebook (ECB) mode encryption

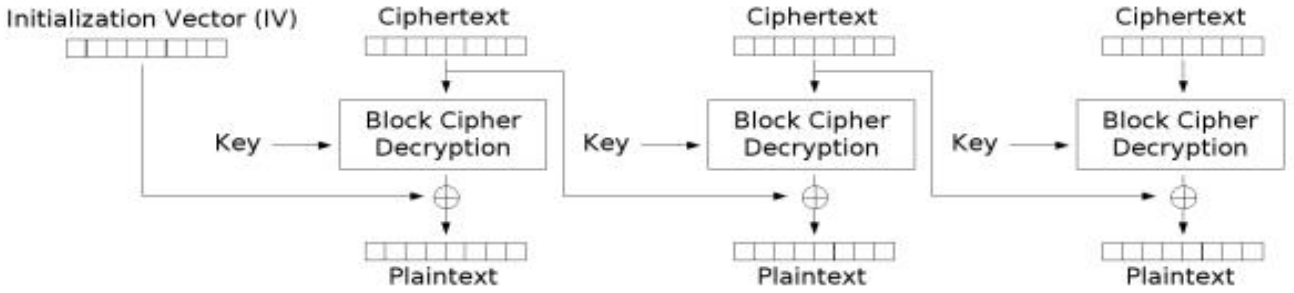


Electronic Codebook (ECB) mode decryption

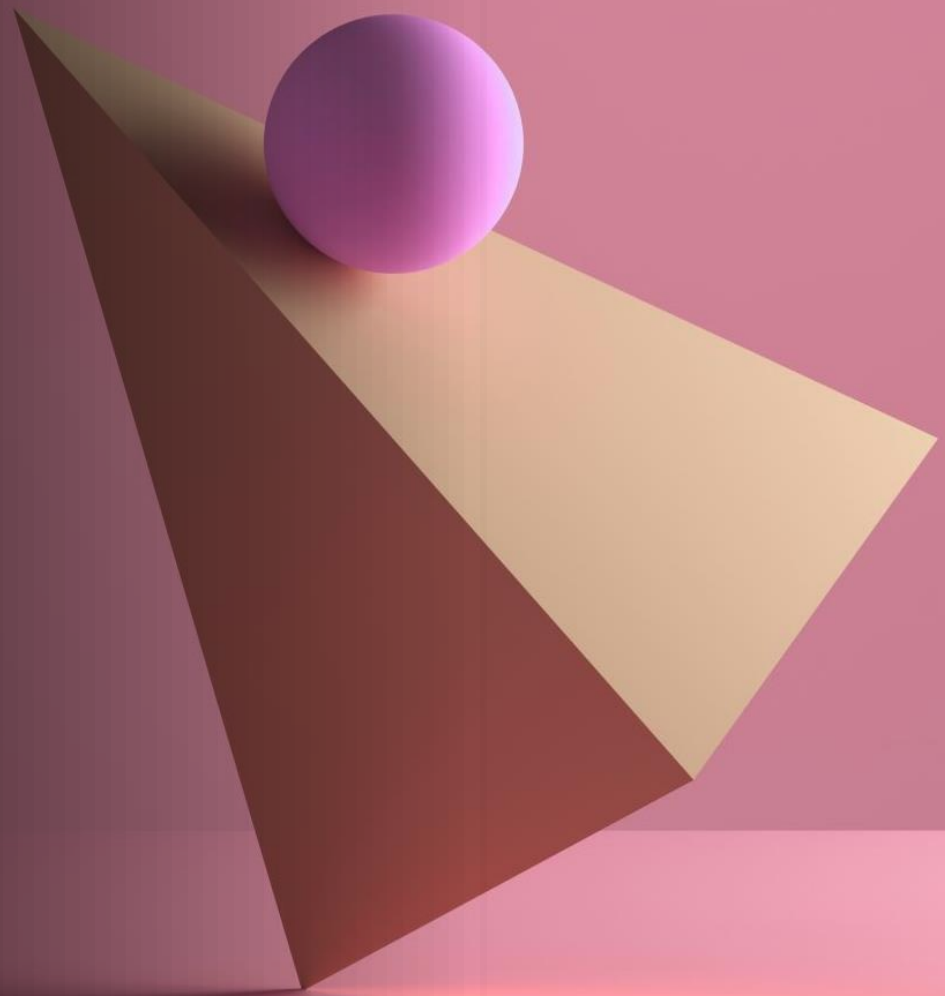
CBC



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption



Q&A

PhD. Phạm Thị Bạch Huệ

M.S Lương Vĩ Minh