

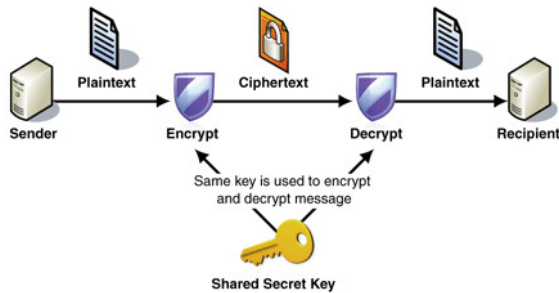
Lab 1. Criptarea si decriptarea datelor intr-o baza de date

Cuvinte cheie:	
<ul style="list-style-type: none">• criptare• algoritm simetric/asimetric de criptare• tehnica padding• tehnica chaining (inlantuiri)	<ul style="list-style-type: none">• DBMS_CRYPTO• Transparent Data Encryption• confidențialitate , integritate• hashing

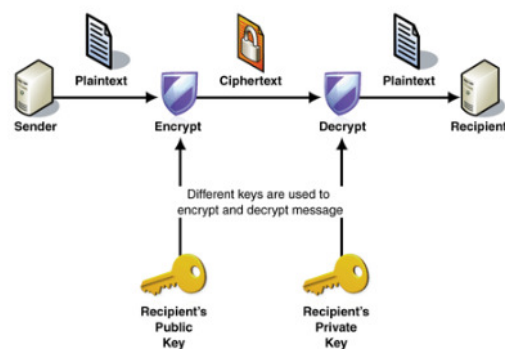
Criptarea datelor reprezintă un mod de a proteja datele curente si cele arhivate, de a le conferi confidențialitate. Exemple de date care necesita criptare? Ex: parole, coduri PIN, numărul de securitate pe cartile de credit, etc.

Elementele de baza intr-un sistem de criptare sunt:

- algoritmul de criptare, metoda prin care este alterata valoarea;
- cheia de criptare, de a carei siguranța depinde vulnerabilitatea datelor criptate.



CRIPTARE SIMETRICA



CRIPTARE ASIMETRICA

Sursa: <http://msdn.microsoft.com/en-us/library/ff650720.aspx>

Oracle suporta **algoritmi de criptare**:

- **simetrici** (care folosesc aceeași cheie pentru criptarea datelor si pentru decriptarea datelor) pentru criptarea datelor stocate;
- **asimetrici** (in care receptorul generează 2 chei: o cheie privata ce va fi folosita de el pentru decriptare si o cheie publica pe care o trimite emitentului pentru a cripta mesajul) pentru autentificarea utilizatorilor bazei de date si pentru comunicația intre client si baza de date. Algoritmii de criptare asimetrici sunt parte a opțiunii contra-cost Oracle Advanced Security.

Reținem ca pentru criptarea datelor stocate Oracle utilizează algoritmi de criptare simetrici.

Recapitulare scurta a câtorva algoritmi simetrici de criptare, disponibili si in Oracle:

- **DES (Data Encryption Standard)**

Sistemul DES cripteaza un bloc de text clar de 64 biti intr-un text criptat tot de 64 biti, utilizând 56 biți dintr-o cheie de 64 biti.

Doua cursuri dedicate acestui subiect si modalitatilor de atac se regăsesc la adresele¹.

- **3-DES (Triple Data Encryption Standard)**

Are la baza formula $c = \text{DES}_{k3}(\text{DES}_{k2}^{-1}(\text{DES}_{k1}(m)))$ in varianta 3DES-edes

sau formula $c = \text{DES}_{k3}(\text{DES}_{k2}(\text{DES}_{k1}(m)))$ in varianta 3DES-eee

unde $k1, k2, k3$ chei de 56 biți (folosind astfel împreuna 168 biți dintr-o cheie de 192 biți solicitata), DES_k este criptarea DES cu cheie k , DES_k^{-1} este criptarea DES cu cheie k , m este blocul de 64 de biti original.

Daca $k1=k2$ sau $k2=k3$ sau $k1=k2=k3$, varianta 3DES devine DES.

Un curs dedicat acestui subiect se regaseste la adresa².

- **AES (Advanced Encryption Standard)**

Sistemul AES cripteaza un bloc de text clar de 128 biti intr-un text criptat tot de 128 biti, utilizând o cheie de 128,192 sau 256 biți.



Un curs dedicat acestui subiect se regaseste la adresa³.

Se observa ca algoritmi de criptare enumerați anterior lucrează cu blocuri de dimensiune fixa, stabilita (64 biti=8 bytes la DES si 3-DES, respectiv 128 biti=16 bytes la AES). Un fragment de date in clar va fi segmentat in blocuri de dimensiunea ceruta de algoritm si algoritmul va fi aplicat pe fiecare bloc astfel obținut.

Cum tratam cazul in care dimensiunea datelor in clar NU este multiplu de dimensiunea ceruta a blocului? → Se utilizează **tehnica padding** de completare a ultimului segment din fragmentul de date in clar pana la dimensiunea unui bloc.

Ne amintim ca la funcțiile pe șiruri aveam LPAD,RPAD.

```
SELECT LPAD('A',3,'#'), RPAD('B',3,'@') FROM DUAL;
```

 LPAD('A',3,'#')	 RPAD('B',3,'@')
###A	B@@

In vederea criptării, se poate opta pentru padding cu zero-uri sau pentru schema de padding PKCS#5.

Fie dim_bloc dimensiunea in bytes a blocului ceruta de algoritm

dim_date dimensiunea totala in bytes a fragmentului de date in clar

Schema de padding PKCS#5 calculează pentru ultimul segment din fragment diferența

$$d = \text{dim_bloc} - (\text{dim_date} \text{ MOD } \text{dim_bloc})$$

si completează fiecare octet lipsa cu valoarea hexa $0x0d$.

Exemplu: $\text{dim_bloc}=8$, $\text{dim_date}=100 \rightarrow d=8 - (100 \text{ MOD } 8) = 8 - 4 = 4$

La ultimul segment, cel de 4 bytes, se face padding cu 0x04040404 (adica 00000100 00000100 00000100 00000100)

¹ http://www.galaxyng.com/adrian_atanasiu/cursuri/crypt/c5.pdf si

http://www.galaxyng.com/adrian_atanasiu/cursuri/crypt/c6.pdf

² http://www.galaxyng.com/adrian_atanasiu/cursuri/crypt/c5.pdf

³ http://www.galaxyng.com/adrian_atanasiu/cursuri/crypt/c7.pdf

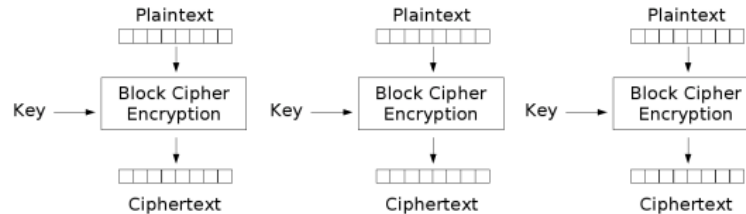
Retinem ca padding-ul se aplica inaintea criptarii si este inlaturat dupa decriptare.

Cum tratam cazul când fragmentul de date in clar consta din mai multe blocuri de criptat?

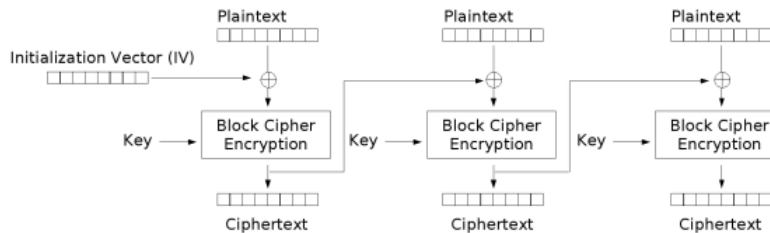
→ Se utilizează **tehnica chaining (inlantuirii)**, care stabilește dacă pentru un bloc criptarea este independenta sau dependenta de criptarea blocurilor anterioare din fragmentul in clar.

In Oracle sunt disponibile variantele următoare de chaining:

- Electronic Code Book (CHAIN_ECB) – fiecare bloc este criptat independent de celelalte blocuri din fragment. Dezavantajul este ca se pot identifica șabloane repetitive in fragment;

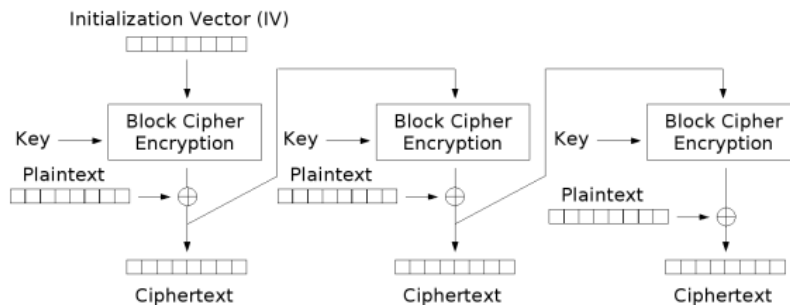


- Cipher Block Chaining (CHAIN_CBC) – pentru fiecare bloc, înaintea criptării, este aplicat XOR cu un vector. Pentru primul bloc din secvență se folosește un vector de inițializare, pentru un bloc din restul secvenței se folosește ca vector de biți rezultatul criptării blocului precedent.



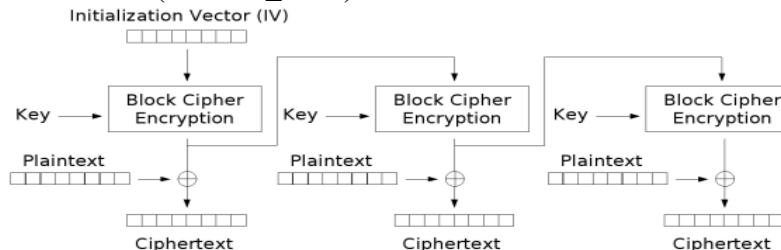
Sursa: http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

- Cipher Feedback (CHAIN_CFB) :



Sursa: http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

- Output Feedback (CHAIN_OFB) :



Sursa: http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

1.1. Criptarea datelor prin cod PL/SQL

Avem la dispozitie pachetele:

DBMS_CRYPTO (introdus odata cu Oracle 10g)

DBMS_OBFUSCATION_TOOLKIT (este deprecated incepand cu Oracle 10g)⁴

Observație importanta: executați următoarea comanda fiind logati ca SYS AS SYSDBA pentru a da utilizatorului nume_user drept de execuție pe pachetul DBMS_CRYPTO (altfel veți primi eroarea

 Error(12,19): PLS-00201: identifier 'DBMS_CRYPTO' must be declared):

GRANT EXECUTE ON dbms_crypto TO nume_user;

Sintaxa (pentru Oracle 10g):

CRIPTARE:

```
dbms_crypto.encrypt(  
  fragment_clar IN RAW,  
  mod_operare IN PLS_INTEGER,  
  cheie IN RAW,  
  vector_inializare IN RAW DEFAULT NULL)  
RETURN RAW;
```

DECRIPTARE:

```
dbms_crypto.decrypt(  
  fragment_clar IN RAW,  
  mod_operare IN PLS_INTEGER,  
  cheie IN RAW,  
  vector_inializare IN RAW DEFAULT NULL)  
RETURN RAW;
```

unde

Mod_operare= Cod_aloritm	+ Cod_padding	+ Cod_chaining
DBMS_CRYPTO.ENCRYPT_DES	DBMS_CRYPTO.PAD_PKCS5	DBMS_CRYPTO.CHAIN_CBC
DBMS_CRYPTO.ENCRYPT_3DES_2KEY	DBMS_CRYPTO.PAD_ZERO	DBMS_CRYPTO.CHAIN_CFB
DBMS_CRYPTO.ENCRYPT_3DES	DBMS_CRYPTO.PAD_NONE	DBMS_CRYPTO.CHAIN_ECB
DBMS_CRYPTO.ENCRYPT_AES128		DBMS_CRYPTO.CHAIN_OFB
DBMS_CRYPTO.ENCRYPT_AES192		
DBMS_CRYPTO.ENCRYPT_AES256		
DBMS_CRYPTO.ENCRYPT_RC4		

⁴ http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/d_obtool.htm

ALTE FUNCTII UTILE:

* Conversie VARCHAR2 → RAW

```
utl_i18n.string_to_raw(  
    data      IN VARCHAR2 CHARACTER SET ANY_CS,  
    dst_charset IN VARCHAR2 DEFAULT NULL)  
RETURN RAW;
```

unde dst_charset = 'AL32UTF8'

Alternativ se poate utiliza, daca in baza de date este setat character set-ul la AL32UTF8:

```
utl_raw.cast_to_raw(sirul IN VARCHAR2) RETURN RAW;
```

* Conversie RAW → VARCHAR2 cu caractere

```
utl_i18n.raw_to_char(  
    data      IN RAW,  
    src_charset IN VARCHAR2 DEFAULT NULL)  
RETURN VARCHAR2;
```

unde dst_charset = 'AL32UTF8'

* Conversie RAW ↔ VARCHAR2 cu hexa

```
RAWTOHEX ( data IN RAW) RETURN VARCHAR2  
HEXTORAW (data IN VARCHAR2) RETURN RAW
```

Aspecte privind managementul cheilor de criptare a datelor

Este dificil pentru utilizatorii bazei de date **sa genereze manual chei** eficiente de criptare, de lungimea solicitata de algoritmi de criptare.

In ce privește furnizarea manuala a cheii de criptare sub forma unui sir de caractere (convertit apoi in RAW), lungimea șirului se calculează astfel:

$$L_{\text{sir}} = \text{Lungime_cheie_in_biti} / 8$$

Exemplu: pentru ENCRYPT_AES128, cheia este de 128 biti=> șirul va avea lungimea

$$L_{\text{sir}} = 128/8 = 16$$

Furnizarea cheii '1234567890123456' va fi acceptata, intrucat are 16 caractere

In timp ce cheia '1234' va ridica excepția 'key length too short'

Analog, pentru restul algoritmilor, pe baza tabelului⁵:

Constant	Effective key length
ENCRYPT_DES	56
ENCRYPT_3DES	156
ENCRYPT_AES128	128
ENCRYPT_AES192	192
ENCRYPT_AES256	256

⁵ Feuerstein Steven (2009) Oracle PL/SQL Programming (editia 5). Editura O'Reilly. ISBN 978-0-596-51446-4. Capitolul 23 „Application security and PL/SQL”

Alternativa o reprezintă **generarea automata a cheilor** de dimensiunea dorita:

cheie RAW (nr_bytes);

cheie:= DBMS_CRYPTO.randombytes (nr_bytes);

Funcția *randombytes* implementează algoritmul Pseudo-Random Number Generator.

Odată obținute, cheile secrete trebuie păstrate în siguranță, întrucât divulgarea lor poate compromite securitatea datelor criptate.

Opțiuni:

| ---- o cheie la nivelul bazei de date |--- stocata in baza de date (intr-o tabela speciala)

| |--- stocata intr-un fisier extern bazei de date

|

| ---- o cheie la nivel de înregistrare --- stocata in baza de date (intr-o tabela speciala)

|

| ---- o combinație între cele anterioare – exista o cheie master la nivelul bazei de date si cate o cheie la nivel de înregistrare. Atât la criptare cat si la decriptare se folosește o cheie hibrida = cheia master XOR cheie înregistrare

(funcția PL/SQL UTL_RAW.bit_xor)

Reținem ca opțiunea unei chei hibride este cea mai eficienta dintre opțiunile enumerate;

- *daca se fura baza de date cu totul, datele nu vor putea fi decriptate , in cazul stocării cheii master in sistemul de fișiere;*
- *daca se divulga cheia master si o cheie de înregistrare, celelalte înregistrări raman protejate.*

Transparent Data Encryption(TDE) este o facilitate oferita începând cu Oracle 10g, care permite declararea unor coloane criptate la nivelul unei tabele a bazei de date. La inserarea datelor în coloanele declarate criptate, Oracle automat cripteaza datele si le stocheaza criptate în baza de date. Orice operație SELECT va decripta automat datele din baza. *Reținem ca **Transparent Data Encryption** nu face diferențiere între utilizatori, oferindu-le tuturor valoarea decriptata a datelor.*

Nu orice coloana poate fi declarata 'criptata'; coloanele din cheia externa (foreign key) NU pot fi criptate TDE.

Fie tabela CONT (id_cont#, serie_card, posesor, sold) pentru care dorim sa declaram criptate coloanele serie_card si sold:

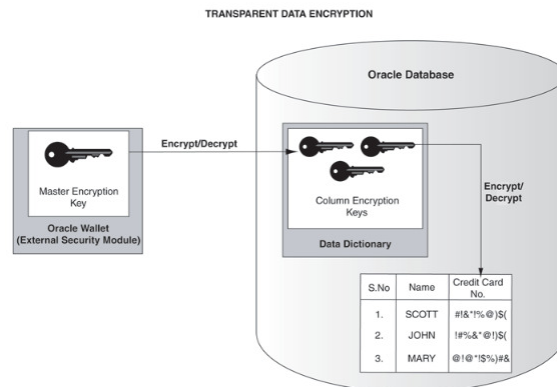
ALTER TABLE cont MODIFY (serie_card ENCRYPT USING 'AES128');

ALTER TABLE cont MODIFY (sold ENCRYPT USING 'AES128');

Pentru toate coloanele criptate dintr-o tabela T se folosește o aceeași cheie privata Key_T. Daca avem mai multe tabele T1,T2,..Tn care conțin fiecare diverse coloane criptate, rezulta *n* chei private Key_T1, Key_T2,..Key_Tn .

Fiecare cheie privata Key_Tj , j=1,n , este criptata la rândul ei cu o cheie master Key_Master si rezultatul criptării ei este stocat în dicționarul datelor.

Cheia master este stocata extern bazei de date într-un wallet. *Astfel, Transparent Data Encryption previne decriptarea datelor în cazul furtului bazei de date.*



Sursa: http://docs.oracle.com/cd/B28359_01/network.111/b28530/asotrans.htm

Pași ⁶:

<i>La criptare automata</i>	<i>La decriptare automata</i>
Obținerea cheii master Key_Master din wallet-ul extern; Decriptarea cheii private Key_Tk folosind cheia master; Criptarea datelor de inserat folosind cheia privata Key_Tk; Stocarea datelor criptate in coloanele tablei.	Obținerea cheii master Key_Master din wallet-ul extern; Decriptarea cheii private Key_Tk folosind cheia master; Decriptarea datelor folosind cheia privata Key_Tk; Returnarea rezultatului.

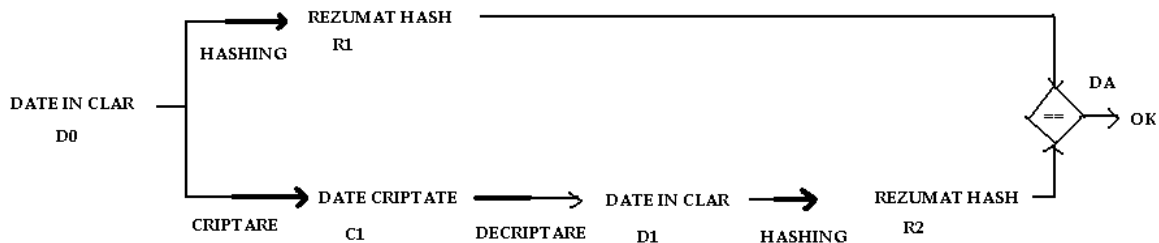
Mai multe detalii in documentatia Oracle ⁷.

Asigurarea integritatii datelor criptate

Criptarea datelor le asigura confidențialitatea, dar nu le garantează si integritatea. Astfel, datele criptate pot fi modificate.

Pentru a preveni acest pericol, in afara de criptarea datelor originale, se utilizează tehnica de hashing, de rezumare, a datelor originale. Hashing-ul are doua proprietati importante:

- nu permite descifrarea valorii originale;
- este determinista, adică aplicat repetitiv pe aceleași date generează același rezultat.



Oracle permite algoritmi de hashing: MD5 si SHA-1 ⁸.

Sintaxa:

```
DBMS_CRYPTO.Hash (
    sir_original IN RAW,
    mod_operare IN PLS_INTEGER) RETURN RAW;
```

unde mod_operare ∈ {DBMS_CRYPTO.HASH_MD5, DBMS_CRYPTO.HASH_SH1}

⁶ http://docs.oracle.com/cd/E11882_01/server.112/e10575/tdpsg_securing_data.htm#CHDCGGBH

⁷ http://docs.oracle.com/cd/E11882_01/server.112/e10575/tdpsg_securing_data.htm#CHDCGGBH,
http://docs.oracle.com/cd/B28359_01/network.111/b28530/asotrans.htm

⁸ http://www.galaxyng.com/adrian_atanasiu/cursuri/cript/cr2_2.pdf

Exerciții:

1. Scrieți procedura CRIPTARE1 care criptează un șir primit ca parametru folosind algoritmul DES, cheia '12345678', padding cu zero-uri și metoda de chaining ECB. Apelați procedura pentru șirul de caractere 'Text în clar' dintr-un bloc PL/SQL fără nume.

2. Scrieți procedura DECRYPTARE1, care decriptează un șir primit ca parametru folosind algoritmul DES, cheia '12345678', padding cu zero-uri și metoda de chaining ECB. Apelați procedura în același bloc PL/SQL fără nume de la exercitiul 1.

3.

Datele de salarizare (id_employee și salary) ale tabelului EMPLOYEES vor fi criptate (AES-128, PAD_PKCS5, CHAIN_CBC) și stocate în tabelul EMPLOYEES_CRIPT astfel:

- înregistrările impare(1,3,...) vor fi criptate cu cheia CHEIE_IMPAR și
- înregistrările pare(2,4,...) vor fi criptate cu cheia CHEIE_PAR.

Cele două chei vor fi generate automat și stocate în baza de date.

Să se creeze secvența SECV_IDCHEIE și tabelul TABELA_CHEI (idcheie#, cheie, tabela).

Să se creeze procedura CRIPTARE_PAR_IMPAR fără parametrii.

În cadrul procedurii să se genereze în mod automat 2 chei private CHEIE_IMPAR și CHEIE_PAR pe câte 16 bytes fiecare. Cheile se vor stoca în tabelul TABELA_CHEI, cu cheia primară din secvența SECV_IDCHEIE.

4. Încercați să alterați prin UPDATE valoarea criptată a salariului primului angajat (ca număr de ordine) din tabelul EMPLOYEES_CRIPT. Setati-i salariul la valoarea 0x1F4 (adică 500 în decimal).

Actualizarea a reușit?

5. Să se creeze procedura DECRYPTARE_PAR_IMPAR fără parametrii, perechea celei de la exercitiul 4. În decriptare se folosesc cheile salvate în TABELA_CHEI, în aceeași ordine (impar,par).

Datele decriptate se stochează în tabelul EMPLOYEES_DECRYPT.

Comparați salariile primului angajat din tabelele EMPLOYEES și EMPLOYEES_DECRYPT.

6. Creați o funcție REZUM_MD5 ce returnează rezumatul hash (MD5) pentru înregistrarea din tabelul EMPLOYEES corespunzătoare angajatului cu employee_id 104. Stocați rezultatul într-o variabilă bind rezumat1. Actualizați salariul acestui angajat acordându-i un spor de 20%.

Creați un nou rezumat hash al acestei înregistrări și stocați rezultatul în variabilă bind rezumat2. Ce observați?

Bibliografie :

[1] http://www.galaxyng.com/adrian_atanasiu/cript.htm

[2] Feuerstein Steven (2009) Oracle PL/SQL Programming (editia 5). Editura O'Reilly. ISBN 978-0-596-51446-4. Capitolul 23 „Application security and PL/SQL”

[3] http://docs.oracle.com/cd/E11882_01/server.112/e10575/tdpsg_securing_data.htm#CHDCGGBH