800 - 841-3000

-- EXAMPLE 1 VPD
-- This example shows row owner security method.
-- A DB user can only view the records he inserted/owned.
-- Ctl_UPD_USer column records who inserted a record
--

-- Assume DBA643 has DBA role and develops a db application to allow AChen and JDoe to use the application.
-- One of the application table is Cust for customers. DBA643 grants privileges to Jsmith and Bjohnson for them
-- to select, add, delete, and update records in the Cust table. However, they can only do these things for their own records.
-- For example, they can only SELECT or UPDATE or DELETE their own records.
-- The following code implement a policy function to enforce the row level security.

-- Make sure DBA643 has DBA role as its session role

-- Step 1 Logon as DBA643 DBA, then create Cust table and a trigger to insert record owner

```
Drop table cust;
CREATE TABLE Cust(
Cus_no number(4) constraint emp_pk Primary Key,
Cus_name varchar2(30),
Cus_Address varchar2(40),
Cus_phone number(10),
Sales_Rep_ID Number(4),
```

Final Test

* DEC 13
11/27/23 - Read!

- vpd and injection Payroll question

```
CTL_UPD_USER Varchar2(30));


CREATE OR REPLACE TRIGGER Trg_Insert_user
Before INSERT
ON CUST For each Row
begin
  :new.ctl_upd_user := user;
end;
/

--Dba643 inserts a few testing records
Insert into Cust (cus_no,
Cus_name,cus_address,cus_phone,Sales_Rep_ID)
values(101,'Joe Pluber','897 Apple Lane',8076568956,111);
Insert into Cust (cus_no,
Cus_name,cus_address,cus_phone,Sales_Rep_ID)
values(102,'Bob Bednark','657 High
Field',6156568956,222);
Insert into Cust (cus_no,
Cus_name,cus_address,cus_phone,Sales_Rep_ID)
values(103,'John Kerns','3456 Field Lane',3206568956,111);
Insert into Cust (cus_no,
Cus_name,cus_address,cus_phone,Sales_Rep_ID)
values(104,'Giavaly Gato','6789 Goettens
Way',3205678901,222);


-- Step 2
DROP PUBLIC SYNONYM CUSTOMERS;
```

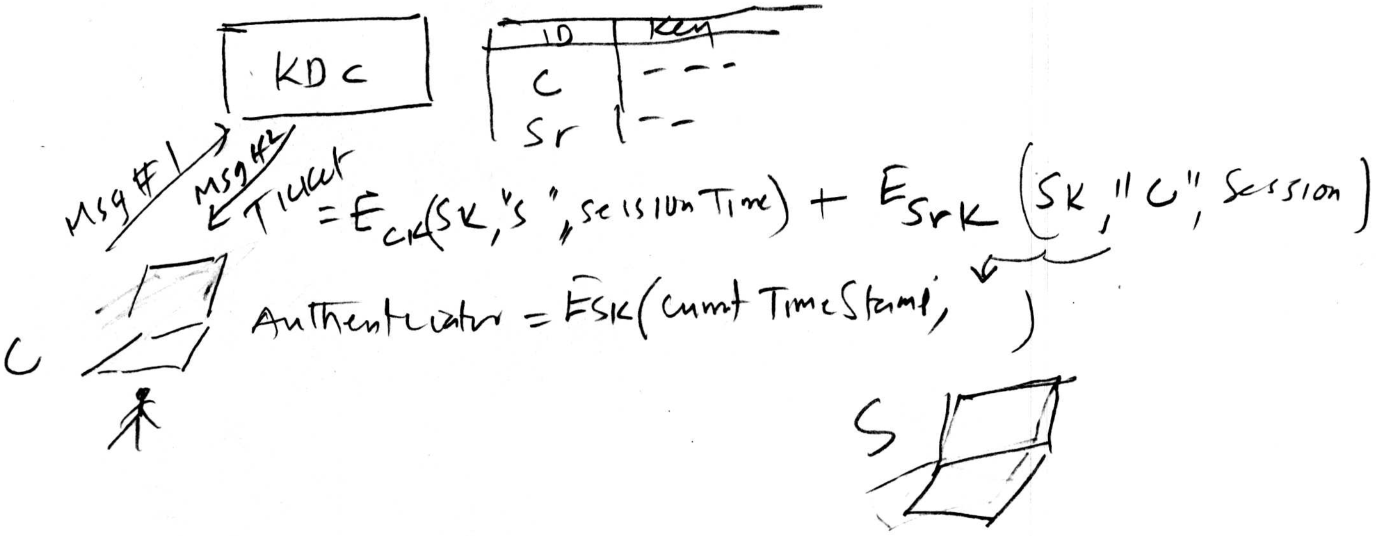
```
CREATE PUBLIC SYNONYM Customers For CUST;
Grant SELECT, UPDATE, INSERT, DELETE on
Customers to ACHEN, JDOE;

--Step 3 create policy function; When DBA643 SELECT, he
should see all records

Create or Replace FUNCTION Sec_Fun_Cust
(P_schema_name IN varchar2, P_object_name IN
varchar2) Return varchar2 IS
 V_where Varchar2(300);
BEGIN
 If User = 'DBA643' then
  V_where := '';
 Else
 V_where := 'CTL_UPD_USER = USER';
 END IF;
 RETURN V_Where;
END;
/

-- Step 4
-- DROP policy function if it was created; only dba can drop
and add policy; When table is dropped, it's gone
--exec DBMS_RLS.DROP_Policy('DBA643', 'Cust',
'Row_Owner_Sec');

-- Add policy function
exec DBMS_RLS.ADD_Policy
('DBA643','Cust','Row_Owner_Sec','DBA643','Sec_Fun_Cust
','SELECT,UPDATE, DELETE, INSERT',TRUE);

-- Step 5  Logon as ACHEN to test

conn Achen/&ACHEN_Password@localhost/IA643;
prompt "see no record";
Select * From Customers;
-- Then insert two records
Insert into Customers (cus_no,
Cus_name,cus_address,cus_phone,Sales_Rep_ID)
values(301,'Joe Pluber','897 Apple Lane',8076568956,111);
Insert into Customers (cus_no,
Cus_name,cus_address,cus_phone,Sales_Rep_ID)
values(302,'Bob Bednark','657 High
Field',6156568956,222);
Prompt "should see two records"
Select * From CUSTOMERS;
Prompt "Should not allow deletion of the customer whose
cus_no is 201"
DELETE FROM Customers WHERE cus_no=201;

-- Clean up
-- Drop table cust;
-- DROP Function Sec_Fun_Cust;

/*
DBMS_RLS.ADD_POLICY (
   object_schema   IN VARCHAR2 := NULL,
   object_name     IN VARCHAR2,
   policy_name     IN VARCHAR2,
  function_schema IN VARCHAR2 := NULL,
  policy_function IN VARCHAR2,
  statement_types IN VARCHAR2 := NULL,
  update_check    IN BOOLEAN  := FALSE,
  enable          IN BOOLEAN  := TRUE);
*/
```

Password

Sno33

Kerberos

- Two way
- Use Symmetric Key
- Need Key Distribution Center



ID	Key
C	---
Sr	--

Ticket $= E_{CK}(SK, "S", \text{Session Time}) + E_{SrK}(\underbrace{SK, "C"}, \text{Session})$

Authenticator $= E_{SK}(\text{Current TimeStamp}, ")$