



Security of user information has become an essential part at many instances in the application development life cycle. Following code snippets perform some means of security features with Phonebook of user's phone.

- Contact Database is being encrypted here which results in non-readable form of contact items in the Phonebook of user's phone.
- On the other side Contact Database is decrypted to get the readable content back.

Headers Required:

```
#include <cntdb.h> // CContactDatabase,
#include <cntitem.h> //CContactItem,CContactItemFieldSet
```

Library required:

```
LIBRARY cntmodel.lib //CContactDatabase, CContactItem, CContactItemFieldSet
```

Capability Required:

```
Capability WriteDeviceData
```

Encrypt Contact Fields

```
void CEncryptContactContainer::EncryptAll()
{
    CContactDatabaseSetDB = CContactDatabase::OpenL();
    CleanupSharedContactDB;

    TContactItem contactDB;
    TContactItemId aContactId

    //Developer can take Heap based descriptor for large/unknown size of contact items.
    <700BufVal;

    const CContactIdArray* contactArray = contactDB->SortedItemsL();

    #contactArray->Count();

    for(TInt i=0;i<cnt;i++)
    {
        * confContactItem=Null;

        = contactDB->OpenContactL((*contactArray)[i]);
        ::PushSharedContactItem);

        CContactItemFieldSet contactItem->CardFields();
        =FieldSetCount(); // This will give number of contact fields.

    for(TInt index=0; index < fieldCount; index++)
    {
        & fieldContent=SetField;

    const CContentType& type = field.ContentType();
```

Encrypt-Decrypt_Contact_Database

```
if(!(type.ContainsFieldType(KUidContactFieldBirthday)))
{
    = contactItem->CardFields()[index].TextStorage()->Text();
    Copy(name);          aValue.
    (aValue); // Call real encrypt here
    ->CardFields()[index]CTextStorage()->SetTextL(aValue);
}
} //Inner for loop ends here
->CommitContactDB(*contactItem);
::PopAndDestroy(contactItem);
} //Outer for loop ends here
CleanupAndDestroy(contactDB);
}

void CEncryptContactContainer:: Encrypt (TDes& aValue)
{
for(TInt iCount=0; iCount< aValue.Length();iCount++)
{
    [iCount]+=aValue
}
}
```

Decrypt Contact Fields

```
void CEncryptContactContainer::DecryptAll()
{
    CContactDatabaseSetDB = CContactDatabase::OpenL();
    CleanupAndDestroy(contactDB);

    TContactItem contactItem;
    TContactItemId aContactId
    <TBufValue;

const CContactIdArray* contactArray = contactDB->SortedItemsL();

    *contactArray->Count();

for(TInt i=0;i<cnt;i++)
{
    * contactItem=NULL;

    = contactDB->OpenContactL((*contactArray)[i]);
    ::PushAndOpenContactItem);

    CContactFieldsSet contactItem->CardFields();
    =FieldsCount(); // This will give number of contact fields.

for(TInt index=0; index < fieldCount; index++)
{
    & fieldContactSet[index];
const CContentType& type = field.ContentType();
if(!(type.ContainsFieldType(KUidContactFieldBirthday)))
{
    = contactItem->CardFields()[index].TextStorage()->Text();
    Copy(name);          aValue.
    (aValue);          Decrypt
    ->CardFields()[index]CTextStorage()->SetTextL(aValue);
}
} //Inner for loop ends here
```

Encrypt-Decrypt_Contact_Database

```
        ->CommitContactDB(*contactItem);
        :~PopAndDestroy(contactItem);
} //Outer for loop ends here
CleanUpSpanAndDestroy(contactDB);
}

void CEncryptContactContainer:: Decrypt (TDes& aValue)
{
for(TInt iCount=0; iCount< aValue.Length();iCount++)
{
    [iCount]-=aValue
}
}
```

- Here simple encryption is used to give an idea. Developer can use more complex Encryption-Decryption algorithms.