Rapport du projet

Ámchemmere Mohamed , M1 informatique

8 avril 2020

Résumé

Dans le cadre de l'UE obligatoire Sécurité Informatique, il est demandé de concevoir une application qui met l'accent sur la notion de Cryptographie sur la plateforme Android. Dans ce cadre j'ai donné mon maximum et avec beaucoup de volonté et de motivation pour réussir à coder au mieux le projet.

J'ai dû surmonter et pallier avec mes erreurs et faiblesses pour concevoir et répondre aux cahier des charges ainsi que pour avoir une bonne application. A terme de quelques semaines de travail, je peux me permettre de dire que j'ai réussi à mettre en place ce projet, qui répond au mieux aux demandes du cahier des charges.

1 Introduction

La cryptographie , utilise des concepts issus de nombreux domaines (Informatique, Mathématiques, Electronique). Toutefois, les techniques évoluent et trouvent aujourd'hui régulièrement racine dans d'autres branches (Biologie, Physique, etc.)

1.1 Vocabulaire de base

— **Cryptologie** : Il s'agit d'une science mathématique comportant deux branches : la cryptographie et la cryptanalyse Cryptographie : La cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des don- nées de manière confidentielle sur un support donné.

— **Chiffrement** : Le chiffrement consiste à transformer une donnée (texte, message, ...) afin de la rendre incompréhensible par une personne autre que celui qui a créé le message et celui qui en est le destinataire. La fonction permettant de retrouver le texte clair à partir du texte chiffré porte le nom de déchiffrement.

— **Texte chiffré** : Appelé également cryptogramme, le texte chiffré est le résultat de l'application d'un chiffrement à un texte clair.

— **Clef** : Il s'agit du paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement. Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, elle diffère pour les deux opérations.

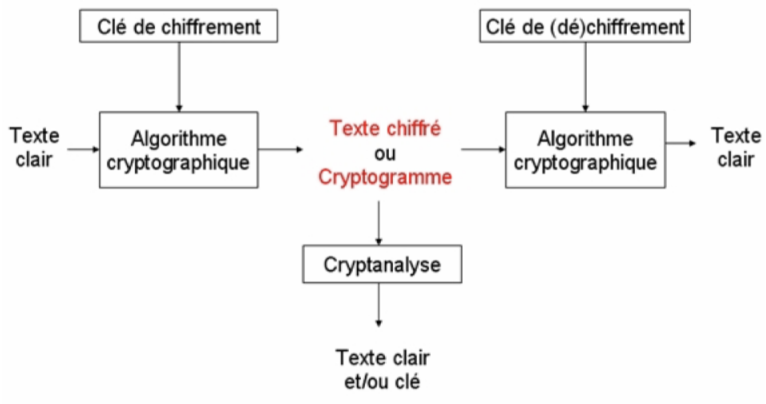


FIG. 2.1 – Protocole de chiffrement

Dans ce cadre , je me permets à vous présenter les différentes ficelles de la conceptions de mon application Android , qui s'appelle : **FastCipher**

2 Description de l'application

Cipher One est une application qui met l'accent sur le concept du chiffrement , elle vous permet choisir parmi une liste de plusieurs chiffrements , ce qui va vous permettre une facilité à crypter , décrypter les textes que vous voulez , sans oublier qu'elle vous donne aussi la possibilité de mettre la clé que vous voulez . Du coup c'est une application qui vous résume toute la recherche que vous faites sur Intyernet pour chiffrer vos textes en utilisant un chiffrelment , c'est une application qui va vous encourager à débuter dans le domaine de cryptographie.

FastCipher une application très utile :

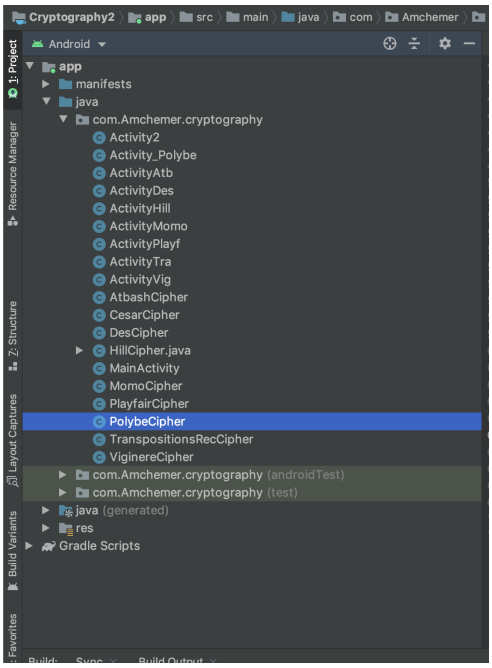
— • Si vous voulez Tester les chiffrements.
— •Si vous voulez comprendre les bases de chiffrement.
— •Si vous ne voulez pas perdre du temps en cherchant les chiffrement sur Internet .
— • Si vous voulez s'amuser en manipulant les textes et les clefs.

3 Architecture générale :

Cette partie du rapport sera consacrée a présenter l'architecture générale de l'application réalisé avec JAVA , ainsi que je vais essayer de vous présenter quelques classes classes pour que vous compreniez le travail en total.

3.0.1 Architecture du projet

Dans l'image ci-dessous vous pouvez voir l'architecture de mon projet :



En gros pour chaque chiffrement , j'ai effectué une classe oû on trouve deux fonctions , la première pour Crypter et la deuxièume pour decrypter . Et c'est ce que vous pouvez Voir dans les images qui viennent :

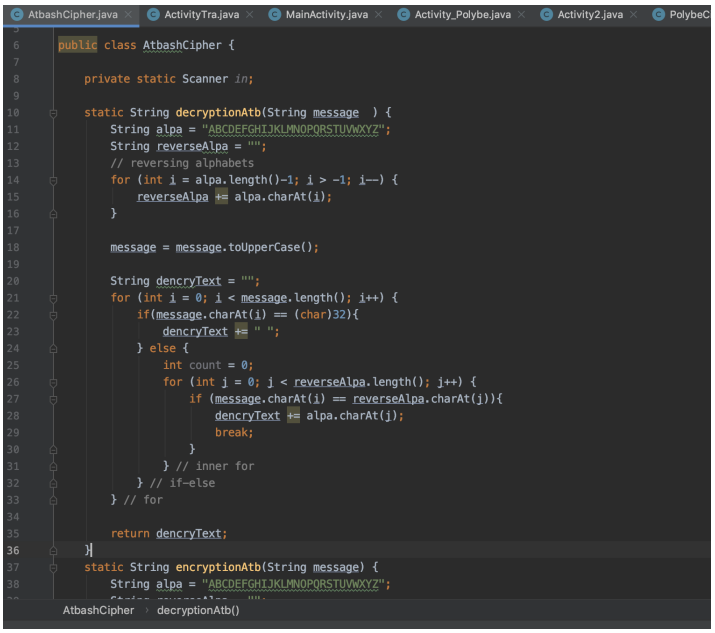
3.0.2 Classe CesarCipher

Pour la classe CesarCipher ,
— la fonction pour Crypter est : encryptC
— La fonction pour décrypter est : decryptC

```java
package com.Amchemer.cryptography;

public class CesarCipher {

        public static String encryptC(String texto , int chave) {
            StringBuilder textoCifrado = new StringBuilder();
            int tamanhoTexto = texto.length();

            for (int c = 0; c < tamanhoTexto; c++) {
                int letraCifradaASCII = (texto.charAt(c)) + (chave);

                while (letraCifradaASCII > 126) {
                    letraCifradaASCII -= 94;
                }

                textoCifrado.append((char) letraCifradaASCII);
            }

            return textoCifrado.toString();
        }

        public static String decryptC(String textChiffre , int chave) {
            StringBuilder texto = new StringBuilder();
            int tamanhoTexto = textChiffre.length();

            for (int c = 0; c < tamanhoTexto; c++) {
                int letraDecifradaASCII = (textChiffre.charAt(c)) - (chave);

                while (letraDecifradaASCII < 32) {
                    letraDecifradaASCII += 94;
                }

                texto.append((char) letraDecifradaASCII);
            }
```

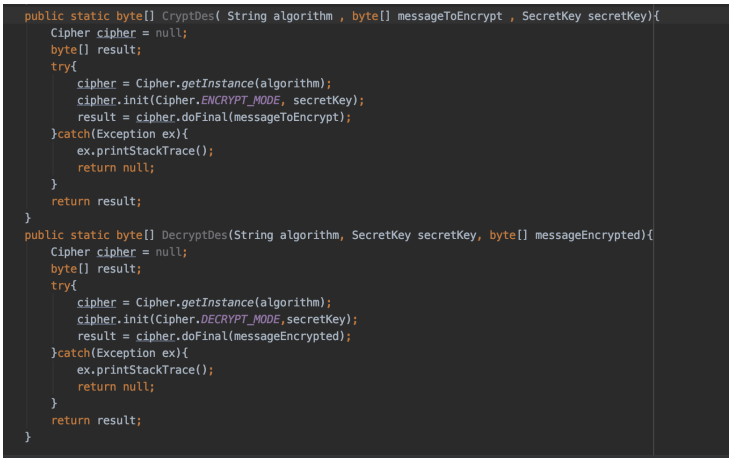
3.0.3 Classe AtbashCipher

Pour cette classe :
— la fonction pour Crypter est : encryptionAtb
— La fonction pour décrypter est : decryptionAtb
— L'alphabet est : (ABCDEFGHIJKLMNOPQRSTUVWXYZ)
Et c'est ce que vous pouvez constater dans la partie de code ci-dessous :




```java
public class AtbashCipher {

    private static Scanner in;

    static String decryptionAtb(String message  ) {
        String alpa = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        String reverseAlpa = "";
        // reversing alphabets
        for (int i = alpa.length()-1; i > -1; i--) {
            reverseAlpa += alpa.charAt(i);
        }

        message = message.toUpperCase();

        String dencryText = "";
        for (int i = 0; i < message.length(); i++) {
            if(message.charAt(i) == (char)32){
                dencryText += " ";
            } else {
                int count = 0;
                for (int j = 0; j < reverseAlpa.length(); j++) {
                    if (message.charAt(i) == reverseAlpa.charAt(j)){
                        dencryText += alpa.charAt(j);
                        break;
                    }
                } // inner for
            } // if-else
        } // for

        return dencryText;
    }
    static String encryptionAtb(String message) {
        String alpa = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
```

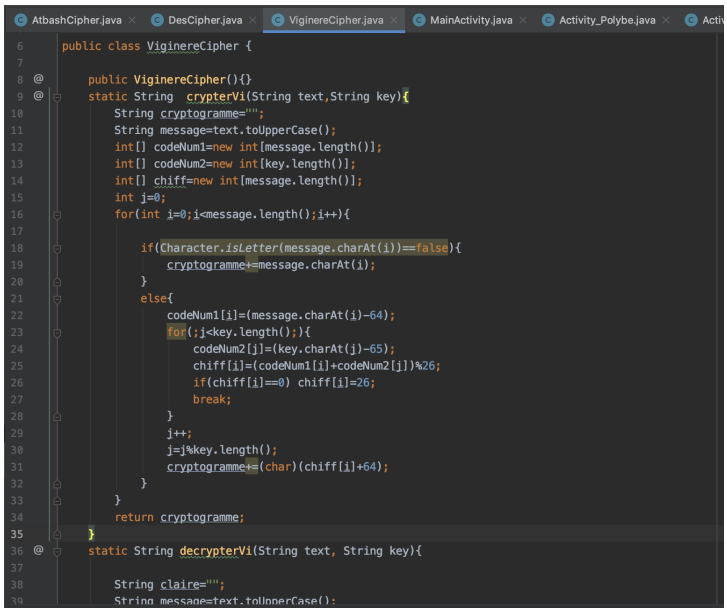
3.0.4 Classe DES



```java
public static byte[] CryptDes( String algorithm , byte[] messageToEncrypt , SecretKey secretKey){
    Cipher cipher = null;
    byte[] result;
    try{
        cipher = Cipher.getInstance(algorithm);
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        result = cipher.doFinal(messageToEncrypt);
    }catch(Exception ex){
        ex.printStackTrace();
        return null;
    }
    return result;
}
public static byte[] DecryptDes(String algorithm, SecretKey secretKey, byte[] messageEncrypted){
    Cipher cipher = null;
    byte[] result;
    try{
        cipher = Cipher.getInstance(algorithm);
        cipher.init(Cipher.DECRYPT_MODE,secretKey);
        result = cipher.doFinal(messageEncrypted);
    }catch(Exception ex){
        ex.printStackTrace();
        return null;
    }
    return result;
}
```

3.0.5 Classe Viginere

Pour la classe ViginereCipher :
— la fonction pour Crypter est : crypterVi
— La fonction pour décrypter est : decrypterV



3.0.6 Classe RsaCipher

Pour la classe RsaCipher :
— la fonction pour Crypter est : EncryptRsa
— La fonction pour décrypter est : DecryptRsa

```
public static byte[] EncryptRsa(byte[] data, RSAPublicKey publicKey) throws Exception{
    Cipher cipher = Cipher.getInstance("RSA");
    cipher.init(Cipher.ENCRYPT_MODE,publicKey);
    byte[] cipherBytes = cipher.doFinal(data);
    return cipherBytes;
}

public static byte[] DecryptRsa(byte[] data, RSAPrivateKey privateKey) throws Exception{
    Cipher cipher = Cipher.getInstance("RSA");
    cipher.init(Cipher.DECRYPT_MODE, privateKey);
    byte[] cipherBytes = cipher.doFinal(data);
    return cipherBytes;
}
```

3.0.7 Classe MomoCipher

Por le chiffrement personnalisé , ça sera XOR , comme vous voyez dans l'image qui vient .

Le chiffrement XOR : est un système de cryptage basique mais pas trop limité. Ainsi, il a beaucoup été utilisé dans les débuts de l'informatique et continue à l'être encore aujourd'hui car il est facile à implémenter, dans toutes sortes de programmes.

Mécanisme : Le XOR est un opérateur logique qui correspond à un "OU exclusif" : c'est le (A OU B) qu'on utilise en logique mais qui exclue le cas où A et B sont simultanément vrais. Voici sa table de vérité :

A	B	A XOR B
FAUX	FAUX	FAUX
FAUX	VRAI	VRAI
VRAI	FAUX	VRAI
VRAI	VRAI	FAUX

— la fonction pour Crypter est : cipherEncryptionM
— La fonction pour décrypter est : cipherDecryptionM

```java
public class MomoCipher {

    private static Scanner in;

    public static String cipherDecryptionM(String msg , String key) {
        String hexToDeci = "";
        for (int i = 0; i < msg.length()-1; i+=2) {
            // splitting hex into a pair of two
            String output = msg.substring(i, (i+2));
            int decimal = Integer.parseInt(output,  radix: 16);
            hexToDeci += (char)decimal;
        }
        // decryption
        String decrypText = "";
        int keyItr = 0;
        for (int i = 0; i < hexToDeci.length(); i++) {
            // XOR Operation
            int temp = hexToDeci.charAt(i) ^ key.charAt(keyItr);

            decrypText += (char)temp;
            keyItr++;
            if(keyItr >= key.length()){
                // once all of key's letters are used, repeat the key
                keyItr = 0;
            }
        }
        return decrypText;
    }
    public static String cipherEncryptionM(String msg , String key) {
        String encrypHexa = "";
        int keyItr = 0;
        for (int i = 0; i < msg.length(); i++) {
            // XOR Operation
```

3.0.8 Classe DesCipher

Le chiffrement DES : Le D.E.S. (ou Data Encryption Standard) naît en 1975 suite à une requête d'I.B.M. en 1960 pour son programme de recherche sur le chiffrement informatique. Au début, les spécialistes de la N.S.A. (National Security Agency, le service de sécurité intérieure américain) se cassent les dents dessus donc I.B.M. est contraint de l'utiliser sous une forme plus simple que prévu. L'utilisation du D.E.S. se généralise alors peu à peu dans les administrations américaines. Depuis, le D.E.S. est remis à niveau tous les 5 ans environ pour faire face à la puissance croissante des ordinateurs qui le mettent en péril.

Inconvenients : Aujourd'hui, le D.E.S. est fortement menacé par les puissances de calcul des ordinateurs. Il n'est en effet pas impossible de balayer la plupart des clés pour casser le code. Un nouveau système, le A.E.S. (Advanced Encryption Standard) est prévu pour le remplacer.
Pour la classe RsaCipher , vous pouvez voir les deux fonctions que j'ai utilisées :

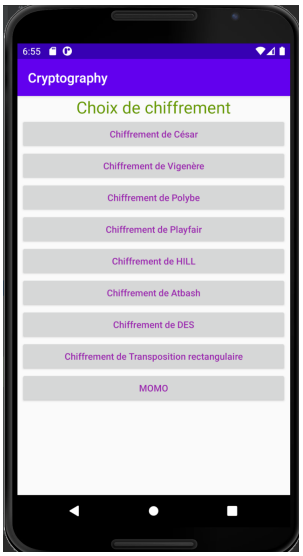
— la fonctio que jn pour'ai utilisé Crypter est : CryptDes
— La fonction pour décrypter est : DecryptDes

```java
public static byte[] CryptDes( String algorithm , byte[] messageToEncrypt , SecretKey secretKey){
    Cipher cipher = null;
    byte[] result;
    try{
        cipher = Cipher.getInstance(algorithm);
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        result = cipher.doFinal(messageToEncrypt);
    }catch(Exception ex){
        ex.printStackTrace();
        return null;
    }
    return result;
}
public static byte[] DecryptDes(String algorithm, SecretKey secretKey, byte[] messageEncrypted){
    Cipher cipher = null;
    byte[] result;
    try{
        cipher = Cipher.getInstance(algorithm);
        cipher.init(Cipher.DECRYPT_MODE,secretKey);
        result = cipher.doFinal(messageEncrypted);
    }catch(Exception ex){
        ex.printStackTrace();
        return null;
    }
    return result;
}
```

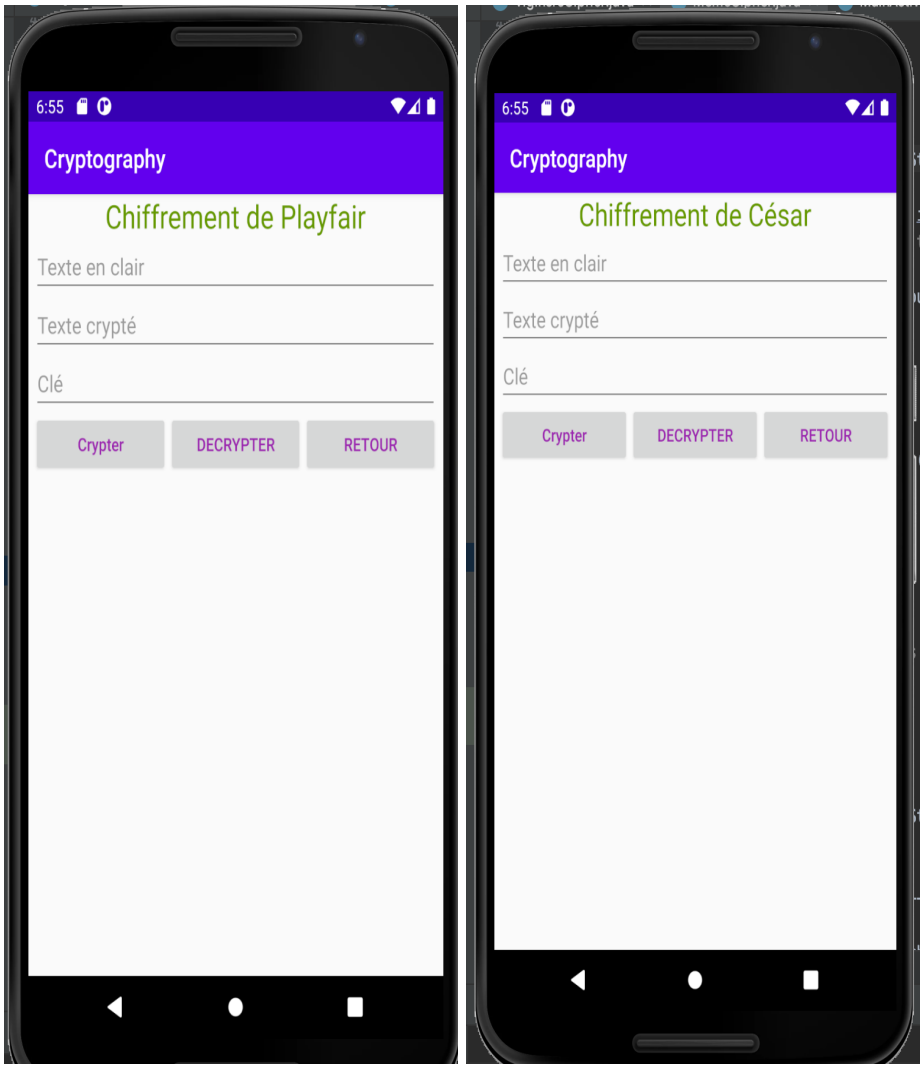
4 Les interfaces

En gros l'application possède 2 vues :

— **La vue des choix des chiffrement :**
Comme vous voyez , Cette vue est à la forme d'une liste dans laquelle l'utilisateur peut choisir un chiffrement .



— **La vue de chiffrement choisi :**
Après le choix d'un chiffrement , et comme vous pouvez voir dans les deux image ci-dessous, l'utilisateur peut saisir le mot qu'il veut avec la clé aussi , ensuite il pourra choisir s'il veut Crypter ou Decrypter en utilisant les Boutons , Et en cliquant sur Retour il reviendra à la vue principale des choix.



5 Conclusion

En conclusion, ce projet fut une belle épreuve qui m'a permis d'acquérir d'avantage d'expérience dans le développement d'applications pour mobiles. L'année de L3 m'a permis d'emmagasiner de nombreuses connaissances.Cette année de Master, par contre, m'a permis d'améliorer mes compétences à travers la pratique. Mon savoir a été mis en application en situation réelle. Sur le plan personnel, il a été très fructueux de travailler , chercher et de pratiquer. Ce projet m'a permis de d'améliorer mes capacités rédactionnelles et de mettre de côté nos différents et idées antagonistes afin de pouvoir créer très bonne application.

- Youtube
- Openclassroom
- Androidstudio
- SupInfo