Projet CRY.ME

Cible de sécurité

Les documents du projet CRY.ME ont été rédigés par CryptoExperts. Les dernières versions incluent quelques modifications apportées par l'ANSSI.

Table des matières

1 Mise en garde

Cette cible de sécurité a été rédigée dans le cadre du projet CRY.ME (https://github.com/ANSSI-FR/cry-me) qui met en œuvre une messagerie sécurisée contenant plusieurs vulnérabilités cryptographiques à des fins éducatives.

Elle contient toutes les informations nécessaires (biens, hypothèses, menaces, fonctions de sécurité, etc.) pour une certification de sécurité de premier niveau (CSPN). Néanmoins, elle n'est pas conforme à toutes les exigences de l'ANSSI en termes d'éligibilité à une certification CSPN.

2 Introduction

2.1 Objectif du document

Ce document constitue la cible de sécurité du produit `CRY.ME` dans le cadre d'une évaluation CSPN.

2.2 Identification du produit

Nom du produit	`CRY.ME`
Version évaluée	1.0
Organisation éditrice	CryptoExperts
Lien vers l'organisation	`https://www.cryptoexperts.com`
Nom commercial du produit	`CRY.ME`
Catégorie de produit	Messagerie sécurisée

3 Argumentaire (description) du produit

3.1 Description générale du produit

CRY.ME est un logiciel de messagerie instantanée. CRY.ME permet l'échange sécurisé de messages entre deux ou plusieurs contacts. La sécurité des messages est garantie par du chiffrement de bout en bout.

La figure 1 donne un aperçu des composants qui forment le produit :
- le serveur **matrix-crx** ;
- les téléphones des utilisateurs sur lesquels est installée l'application ;
- les jetons ou *token* d'authentification des utilisateurs

et des connexions qui les relient.

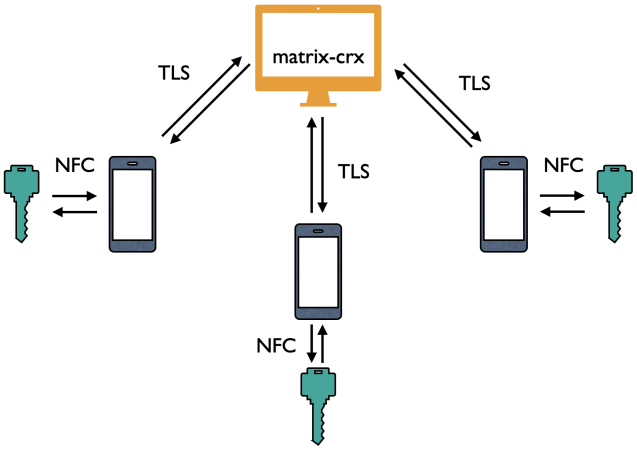

FIGURE 1 – Représentation schématique du produit

3.1.1 Application CRY.ME

L'application CRY.ME permettant l'échange de messages est disponible sur la plateforme Google Play Store.

3.1.2 Serveur CRY.ME

Les applications CRY.ME communiquent leurs messages au travers d'un serveur. Ce serveur permet la mise en relation de messages entre les contacts et le stockage de données sensibles chiffrées.

3.1.3 Jetons d'authentification CRY.ME

Les applications CRY.ME font appel à un jeton ou *token* Yubikey par utilisateur pour son authentification auprès du serveur.

3.2 Description de l'utilisation du produit

A l'issue de l'installation de l'application CRY.ME sur son téléphone, l'utilisateur se crée une identité auprès du serveur **matrix-crx**. Il choisit un nom d'utilisateur et un mot de passe. Il transmet également au serveur le certificat de son *token*. Une fois cette première étape franchie, l'utilisateur a accès à de nombreuses fonctionnalités, notamment la connexion à l'application, l'envoi de messages, l'envoi de pièce jointe et le stockage sécurisé qui sont décrites ci-après.

3.2.1 Connexion

Lorsqu'un utilisateur ayant déjà un compte souhaite s'authentifier auprès du serveur, il se sert de deux facteurs d'authentification : son mot de passe et sa clé **yubikey**. Le serveur s'assure de la correction du mot de passe et envoie un *challenge* à faire signer par la **yubikey**. La signature est vérifiée en utilisant le certificat envoyé au moment de la création de compte.

3.2.2 Envoi d'un message

Les communications entre deux ou plusieurs utilisateurs sont organisées dans des salons ou *rooms* qui constituent des pages de conversations. L'envoi d'un message dans un nouveau salon nécessite un échange de clé préalable entre tous les couples d'utilisateurs de ce salon. Cet échange de clé est réalisé dans des sessions `Olm` et les échanges de messages une fois la communication établie dans des sessions `Megolm`.

3.2.3 Envoi d'une pièce jointe

Pour envoyer une pièce jointe `pj` de manière sécurisée dans une conversation, l'application chiffre la pièce jointe puis la transmet au serveur qui la met à disposition par un lien *url* public. Toutes les données permettant son déchiffrement et sa vérification sont envoyées dans la conversation `Megolm`.

3.2.4 Stockage sécurisé

Chaque utilisateur peut stocker ses clés privées et ses clés de sessions `Megolm` sur le serveur. Pour cela, il génère une clé de stockage à partir d'un mot de passe personnel. La connaissance de ce mot de passe lui permet notamment de retrouver l'historique de ses conversations lors d'une nouvelle connexion.

3.3 Description de l'environnement prévu pour son utilisation

L'application `CRY.ME` s'installe sur des équipements mobiles Android. Elle utilise un token d'authentification Yubikey. Le serveur `CRY.ME` est opéré par la société CryptoExperts et est hébergé par la société OVH.

3.4 Description des hypothèses sur l'environnement

L'application `CRY.ME` doit être installée sur un système sain, correctement mis à jour.

Le serveur ne fait pas partie de la cible d'évaluation (hors TOE).

3.4.1 Installation et initialisation

— Le serveur `CRY.ME` est déployé à partir du code source sur un serveur OVH (instance b2-7) possédant un système avec une mémoire RAM de 7 Go, une capacité de stockage de 50 Go SSD et un CPU à 2 noyaux cadencés à plus de 2 GHz. Les caractéristiques techniques sont détaillées à l'adresse suivante :

https://www.ovhcloud.com/fr/public-cloud/prices/.

— Le token Yubikey d'un utilisateur est initialisé avec un code PIN par défaut qui doit être modifié à l'installation.

3.4.2 Utilisateur

— L'utilisateur se doit d'activer la fonctionnalité de chiffrement de bout en bout de l'application.
— L'utilisateur se doit de s'appuyer sur un canal authentique (typiquement, un échange de vive voix, en face à face ou par téléphone) pour vérifier l'appareil d'un autre utilisateur.

3.4.3 Téléphone

— Le téléphone est considéré sain et non perdable.

3.4.4 Serveur

— Le serveur n'est pas considéré sûr.
— La remise de tous les messages est garantie.

3.5 Description des dépendances

L'application est installée sur les téléphones Android, version 11. L'application utilise une YubiKey 5 NFC (https://www.yubico.com/fr/product/yubikey-5-nfc/).

3.6 Description des utilisateurs typiques concernés

Les utilisateurs disposent de privilèges identiques et échangent des messages entre eux. Ils disposent chacun de l'application installée sur un système Android et d'un token d'authentification.

3.7 Définition du périmètre de l'évaluation

Le périmètre de l'évaluation comprend les fonctionnalités de l'application `CRY.ME` :

- l'enregistrement et la connexion d'un utilisateur ;
- l'ajout d'un contact ;
- l'envoi de messages et de pièces jointes à un contact ;
- la création de groupe et l'envoi de messages de groupes ;
- les communications entre l'application et le serveur ;
- la création et la restauration d'une sauvegarde de clé chiffrée.

Aucune procédure de réinitialisation de mot de passe n'est mise en place. En cas de perte de mot de passe, l'utilisateur est invité à créer un nouveau compte.

La sécurité du serveur `CRY.ME` ne fait pas partie de la cible. En effet, l'application doit pouvoir fonctionner correctement avec un serveur compromis tant que la disponibilité des messages est garantie. La couche physique de transport des communications application-token (NFC, USB) ne fait pas non plus partie de la cible.

Sont également exclues de la cible :

— les attaques logiques sur le téléphone (rooter le téléphone, applications malveillantes, etc.),
— les attaques physiques sur le téléphone (attaques invasives ou semi-invasives, attaques par canaux auxiliaires, attaques par injection de fautes, etc.),
— les attaques physiques sur le token.

4 Description de l'environnement technique du produit

4.1 Matériel compatible ou dédié

L'application `CRY.ME` est installée sur un matériel Samsung Galaxy M12 sur lequel est installé le système d'exploitation Android (version 11). L'application nécessite un token YubiKey 5 NFC avec lequel elle communique

- en USB pour la génération de la clé RSA,
- en NFC pour authentifier l'utilisateur auprès du serveur.

Le serveur est déployé sur un serveur OVH (instance b2-7).

4.2 Système d'exploitation retenu

Le système d'exploitation retenu pour le téléphone est Android 11. Le système d'exploitation retenu pour le serveur est Debian 10.

5 Description des biens sensibles que le produit doit protéger

Les biens sensibles que le produit `CRY.ME` doit protéger selon les critères de confidentialité (C) et d'intégrité (I) sont les suivants :

Bien	**C**	**I**
Identité de l'utilisateur		X
Messages échangés	X	X
Pièces jointes échangées	X	X
Clés sauvegardées	X	X

La protection des biens identifiés suppose la bonne utilisation de l'application par l'utilisateur, notamment :
- l'utilisateur active la fonctionnalité de chiffrement de bout en bout de l'application,
- l'utilisateur s'appuie sur un canal authentique (typiquement, un échange de vive voix, en face à face ou par téléphone) pour vérifier l'appareil d'un autre utilisateur.

6 Description des menaces

6.1 Agents menaçants

Les agents menaçants sont les suivants :
- les attaquants extérieurs étant capables d'envoyer des messages à destination des utilisateurs de l'application ;
- les attaquants extérieurs, capables d'intercepter et de modifier les flux de communications entre l'application et le serveur ;
- les attaquants disposant d'un accès ponctuel ou permanent au *token* d'authentification d'un (autre) utilisateur ;
- les attaquants disposant d'un accès privilégié au serveur `CRY.ME` ;
- les utilisateurs de l'application, essayant d'accéder à ou de compromettre des données qui ne leur sont pas destinées.

6.2 Menaces

Les menaces identifiées sont les suivantes :

M1. Modification d'informations. Un attaquant injecte des données dans les communications entre l'application et le serveur et modifie les données échangées (messages applicatifs et protocolaires).

M2. Interception d'informations. Un attaquant intercepte les communications entre l'application et le serveur (messages applicatifs et protocolaires) et récupère des informations sensibles.

M3. Usurpation d'identité auprès du serveur. Un attaquant se fait passer pour un correspondant légitime auprès du serveur.

M4. Usurpation d'identité auprès d'un autre utilisateur. Un attaquant se fait passer pour un correspondant légitime auprès d'un autre utilisateur.

M5. Récupération des clés sauvegardées. Un attaquant récupère les clés sauvegardées sur le serveur par un utilisateur.

Les attaques sur la disponibilité du serveur (*e.g.* attaques par deni de service) ne font pas partie des menaces identifiées.

7 Description des fonctions de sécurité du produit

F1. Authentification d'un utilisateur auprès du serveur

Les protocoles *sign-up* et *sign-in* servent à la création d'un compte utilisateur et la connexion d'un utilisateur à ce dernier. Il s'agit d'une authentification à deux facteurs :

- l'utilisateur s'authentifie auprès du serveur avec le mot de passe qu'il a choisi lors du *sign-up* ;
- l'utilisateur s'authentifie avec son *token* d'authentification. Concrètement, le serveur envoie un *challenge* à l'utilisateur qui le complète par un *timestamp*. Sur demande de l'utilisateur (qui se connecte au *token* grâce à un code PIN), le *token* signe la concaténation du *challenge* et du *timestamp* en utilisant une clé générée soit par le *token* directement, soit par l'application, selon la préférence de l'utilisateur. La signature (et la validité du timestamp correspondant) est vérifiée par le serveur en utilisant le certificat transmis lors du *sign-up*.

F2. Authentification d'un utilisateur auprès d'un autre utilisateur

Chaque utilisateur peut générer des clés *cross-signing* dès sa première connexion et a la possibilité de les conserver chiffrées sur le serveur. Ces clés de signature permettent de signer les clés d'autres utilisateurs et les clés de ses différents appareils.

Avant de signer leurs clés maître respectives, un protocole spécifique permet à deux utilisateurs A et B de vérifier leurs appareils respectifs afin d'éviter des attaques *man-in-the-middle* en se transmettant des données par d'autres voies de communication (*e.g.*, oralement).

F3. Chiffrement des messages et des pièces jointes

Les conversations entre deux ou plusieurs utilisateurs sont traitées par l'application comme des conversations de groupe. Chaque utilisateur dans le groupe commence par partager avec tous les autres utilisateurs sa clé *ratchet symétrique* servant à dériver les clés utilisées pour le chiffrement (et l'authentification) de ses messages. Le partage de la clé *ratchet symétrique* se fait au moyen d'une session `Olm` entre l'utilisateur et chacun des autres utilisateurs du groupe. Une session `Olm` entre deux appareils se base sur l'algorithme *double ratchet* afin d'établir un canal sécurisé dont l'authenticité est assurée par les clés *cross-signing*. Cette étape est réalisée par chaque utilisateur à chaque nouveau changement de sa clé *ratchet symétrique*. Des protocoles spécifiques sont ensuite définis pour l'envoi et la réception de messages chiffrés ou l'envoi et la réception de pièces jointes chiffrées.

F4. Authentification des messages et des pièces jointes

Un motif d'authentification est associé à tout message ou pièce jointe émis lors d'une conversation de groupe (entre deux ou plusieurs utilisateurs) pour être ensuite vérifié par le(s) destinataire(s).

F5. Chiffrement, stockage et récupération des clés des utilisateurs

La solution fournit des fonctionnalités de stockage sécurisé (chiffré, authentifié) et récupération des clés privées sur le serveur. Un utilisateur qui souhaite créer une sauvegarde de ses clés privées sur le serveur commence par générer une clé symétrique nommée *secret storage* qui servira au chiffrement des clés de sauvegarde. Cette clé est dérivée d'un mot de passe choisi par l'utilisateur. Ensuite, en utilisant la clé *secret storage*, l'utilisateur peut chiffrer ses clés privées et les stocker sur le serveur, et peut également les récupérer et les déchiffrer. Pour la sauvegarde des messages de conversation `Megolm` chiffrés, l'utilisateur crée une sauvegarde chiffrée des clés de session `Megolm` et les stocke sur le serveur. L'utilisateur peut ensuite récupérer ces clés `Megolm` en utilisant le protocole dédié.

8 Couverture des menaces

Le tableau suivant illustre la couverture des menaces par les différentes fontions de sécurité.

	M1. Modification d'informations	M2. Interception d'informations	M3. Usurpation d'identité auprès du serveur	M4. Usurpation d'identité auprès d'un autre utilisateur	M5. Récupération des clés sauvegardées
F1. Authentification d'un utilisateur auprès du serveur			✗		
F2. Authentification d'un utilisateur auprès d'un autre utilisateur				✗	
F3. Chiffrement des messages et des pièces jointes		✗			
F4. Authentification des messages et des pièces jointes	✗				
F5. Chiffrement, stockage et récupération des clés des utilisateurs					✗