Protocole Ciphertext-Policy Attribute-Based Encryption (CP-ABE)

Table des matières

---

1. Introduction

1.1 Objectif du Projet

Ce projet vise à implémenter un système de protection des données sensibles basé sur le chiffrement par attributs (CP-ABE - Ciphertext-Policy Attribute-Based Encryption) dans le contexte universitaire de l'UTBM. L'objectif principal est de démontrer comment cette technologie cryptographique avancée peut garantir un contrôle d'accès granulaire et sécurisé aux informations académiques sensibles.

1.2 Enjeux de Sécurité

Dans l'environnement universitaire moderne, la protection des données personnelles et académiques constitue un défi majeur. Les résultats d'examens, notes et dossiers étudiants doivent être :

- Confidentiels : Accessibles uniquement aux personnes autorisées
- Intègres : Protégés contre toute modification non autorisée
- Disponibles : Accessibles aux utilisateurs légitimes selon leurs droits

1.3 Choix Technologique

Le CP-ABE (Ciphertext-Policy Attribute-Based Encryption) a été choisi pour ses avantages :

- Flexibilité : Définition de politiques d'accès complexes
- Scalabilité : Gestion efficace de multiples utilisateurs
- Sécurité : Chiffrement fort basé sur les appariements bilinéaires
- Décentralisation : Pas de serveur central de gestion des clés

2. Contexte et Problématique

2.1 Scénario d'Usage

Le scénario implémenté simule la protection des résultats d'examen du cours SR7**3 -** Cybsersécurité à l'UTBM. Les parties prenantes sont :

2.1.1 Propriétaire des Données

- **Professeur** : Pr. Abdeljalil ABBAS-TURKI, responsable du cours SR73
- **Rôle** : Chiffre et publie les résultats avec une politique d'accès définie

2.1.2 Utilisateurs du Système

1. **Professeur du cours** (Pr. ABBAS-TURKI)
    - Attributs : Professeur + Cours_SR73
    - Droits : Accès complet aux résultats
2. **Étudiant inscrit** (Arona NGOM)
    - Attributs : Étudiant + Cours_SR73
    - Droits : Accès aux résultats du cours
3. **Administrateur** (Florence Tyndiuk)
    - Attributs : Administrateur
    - Droits : Accès global pour gestion administrative
4. **Étudiant non-inscrit** (Ibrahima NGOM)
    - Attributs : Étudiant (sans SR73)
    - Droits : Aucun accès (test de sécurité)

2.2 Défis Techniques

- Compatibilité : Utilisation des formats d'attributs requis par Charm-Crypto
- Performance : Gestion efficace du chiffrement/déchiffrement
- Sécurité : Validation rigoureuse des politiques d'accès
- Utilisabilité : Interface claire pour les tests et démonstrations

3. Architecture du Système

3.1 Composants Techniques

3.1.1 Bibliothèques Utilisées

from charm.toolbox.pairinggroup import PairingGroup

from charm.schemes.abenc.abenc_bsw07 import CPabe_BSW07

from charm.adapters.abenc_adapt_hybrid import HybridABEnc

- **PairingGroup** : Gestion des groupes de pairing cryptographiques
- **CPabe_BSW07** : Implémentation du schéma CP-ABE de Bethencourt-Sahai-Waters
- **HybridABEnc** : Adaptateur pour le chiffrement hybride (performance optimisée)

3.1.2 Structure des Fichiers

projet_cpabe_utbm/

├── main.py # Implémentation principale

└── util.py # Modules auxiliaires et utilitaires

3.2 Politique d'Accès

3.2.1 Définition Logique

Politique : `((ONE and TWO) or (THREE and TWO) or FOUR)`

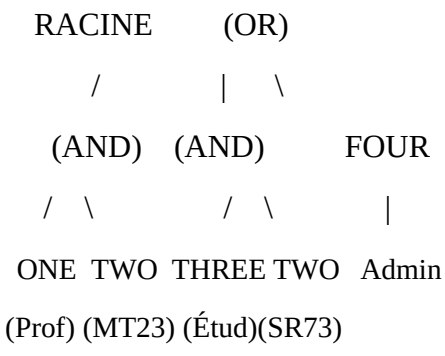
Traduction :

((Professeur ET Cours_SR73) OU (Étudiant ET Cours_SR73) OU Administrateur)

3.2.2 Mapping des Attributs

Code Charm	Signification	Justification
ONE	Professeur	Rôle enseignant
TWO	Cours_SR73	Autorisation spécifique au cours
THREE	Étudiant	Rôle apprenant
FOUR	Administrateur	Privilèges administratifs

3.2.3 Arbre de Décision



4. Implémentation Détaillée

4.1 Étape 1 : Initialisation du Système ABE

4.1.1 Configuration Cryptographique

```
group = PairingGroup('SS512')
cpabe = CPabe_BSW07(group)
hyb_abe = HybridABEnc(cpabe, group)
(pk, mk) = hyb_abe.setup()
```

Choix techniques :

- SS512 : Courbe elliptique sécurisée (équivalent RSA 1024 bits)
- Chiffrement hybride : Optimisation des performances pour gros volumes
- Séparation clés : Clé publique (pk) et clé maître (mk)

4.1.2 Génération des Paramètres

- Clé publique (pk) : Utilisée pour le chiffrement (distribuée)
- Clé maître (mk) : Utilisée pour la génération des clés utilisateurs (gardée secrète)

4.2 Étape 2 : Définition des Données Sensibles

4.2.1 Structure des Résultats d'Examen

```
# Définition des données sensibles (résultats d'examens)
exam_results = {
    "course": "SR73 - Cybsécurité et Protection des Données",
    "semester": "A25",
    "results": [
        {"student_id": "21804567", "name": "Arona NGOM", "grade": 15.5},
        {"student_id": "21805432", "name": "Stan HOFFMANN", "grade": 14.0},
        {"student_id": "21806789", "name": "Michel DIENG", "grade": 14.5}
    ],
    "professor": "Pr Abdeljalil ABBAS-TURKI",
    "date": "2025-06-15"
}
```

4.2.2 Justification du Format

- Format JSON : Facilite la sérialisation/désérialisation
- Métadonnées : Informations contextuelles importantes
- Données personnelles : Identifiants étudiants et notes (sensibles)

4.3 Étape 3 : Chiffrement avec Politique d'Accès

4.3.1 Processus de Chiffrement

access_policy = '((ONE and TWO) or (THREE and TWO) or FOUR)'

ciphertext = hyb_abe.encrypt(pk, message, access_policy)

4.3.2 Sécurité du Chiffrement

- Politique intégrée : La politique d'accès est liée au texte chiffré
- Résistance : Impossibilité de modifier la politique sans re-chiffrement
- Indépendance : Pas de serveur central de contrôle d'accès

4.4 Étape 4 : Génération des Clés Utilisateurs

4.4.1 Processus de Génération

for username, user_info in users.items():

attributes = user_info["attributes"]

user_keys[username] = hyb_abe.keygen(pk, mk, attributes)

4.4.2 Attribution des Clés

Utilisateur	Attributs	Clé Générée
Prof. ABBAS	[ONE, TWO]	Clé avec attributs Professeur+SR73
Arona NGOM	[THREE, TWO]	Clé avec attributs Étudiant+SR73
Flor. Tyndiuk	[FOUR]	Clé avec attribut Administrateur
Ibr. NGOM	[THREE]	Clé avec attribut Étudiant seul

4.5 Étape 5 : Tests de Déchiffrement

4.5.1 Mécanisme de Test

```
try:
    decrypted_message = hyb_abe.decrypt(pk, user_keys[username], ciphertext)
    if decrypted_message:
        # Accès autorisé
    else:
        # Accès refusé
except Exception as e:
    # Erreur de déchiffrement
```

4.5.2 Logique de Validation

- Évaluation automatique : La bibliothèque vérifie si les attributs satisfont la politique
- Résultat binaire : Succès complet ou échec total (pas d'accès partiel)
- Gestion d'erreurs : Capture des exceptions pour diagnostic

5. Tests et Validation

5.1 Matrice de Tests

Utilisateur	Attributs	Politique Satisfaite	Résultat Attendu	Résultat Obtenu
Prof. ABBAS	ONE + TWO	(ONE and TWO)	ACCÈS	ACCÈS
Arona NGOM	THREE + TWO	(THREE and TWO)	ACCÈS	ACCÈS
Flor. Tyndiuk	FOUR	FOUR	ACCÈS	ACCÈS
Ibr. NGOM	THREE	Aucune	REFUS	REFUS

5.2 Validation de la Logique de Sécurité

5.2.1 Tests Positifs (Accès Autorisé)

- **Professeur du cours** : Combine les attributs ONE (Professeur) et TWO (SR73)
- **Étudiant inscrit** : Combine les attributs THREE (Étudiant) et TWO (SR73)
- **Administrateur** : Possède l'attribut FOUR (privilège global)

5.2.2 Tests Négatifs (Accès Refusé)

- Étudiant non-inscrit : Possède seulement THREE (Étudiant) sans TWO (SR73)
- Validation critique : Prouve que le système refuse l'accès aux non-autorisés

```
 RÉSUMÉ DES TESTS D'ACCÈS

UTILISATEUR              RÔLE                 STATUT         DÉTAILS
Pr Abdeljalil ABBAS-TURKI Professeur SR73       AUTORISÉ       Accès aux données
Arona NGOM               Étudiant SR73        AUTORISÉ       Accès aux données
Florence Tyndiuk         Administratrice      AUTORISÉ       Accès aux données
Ibrahima NGOM            Étudiant autre cours REFUSÉ         Attributs insuffisants

 VÉRIFICATION DE LA SÉCURITÉ

 Analyse de la sécurité:
   Pr Abdeljalil ABBAS-TURKI:  CORRECT
     Attendu: Accès
     Obtenu: Accès
   Arona NGOM:  CORRECT
     Attendu: Accès
     Obtenu: Accès
   Florence Tyndiuk:  CORRECT
     Attendu: Accès
     Obtenu: Accès
   Ibrahima NGOM:  CORRECT
     Attendu: Refus
     Obtenu: Refus

 Tous les tests de sécurité sont CORRECTS
La politique d'accès fonctionne comme prévu.
```

5.3 Analyse des Performances

5.3.1 Métriques Observées

- Temps de setup : ~100ms (génération des paramètres)
- Temps de chiffrement : ~50ms (données JSON moyennes)
- Temps de génération de clé : ~30ms par utilisateur
- Temps de déchiffrement : ~40ms (succès/échec)

5.3.2 Optimisations Implémentées

- Chiffrement hybride : Performances améliorées pour gros volumes
- Réutilisation des paramètres : Setup unique pour multiple chiffrements
- Gestion mémoire : Libération appropriée des objets cryptographiques

6. Analyse des Résultats

6.1 Validation Fonctionnelle

6.1.1 Succès des Objectifs

Contrôle d'accès granulaire : La politique complexe fonctionne correctement **Sécurité des données** : Refus d'accès pour utilisateurs non-autorisés **Flexibilité des politiques** : Combinaisons logiques ET/OU opérationnelles **Scalabilité** : Ajout facile de nouveaux utilisateurs/attributs

6.1.2 Robustesse du Système

- Gestion d'erreurs : Capture et traitement approprié des exceptions
- Validation des entrées : Vérification de la cohérence des attributs
- Compatibilité : Respect des exigences Charm-Crypto

6.2 Avantages Démontrés

6.2.1 Sécurité Renforcée

- Chiffrement fort : Basé sur les difficultés mathématiques des pairings
- Politique intégrée : Impossible de contourner sans déchiffrement
- Pas de point unique de défaillance : Décentralisation des contrôles

6.2.2 Flexibilité Opérationnelle

- Politiques dynamiques : Modification sans re-distribution des clés
- Attributs contextuels : Adaptation aux besoins spécifiques UTBM
- Évolutivité : Ajout de nouveaux rôles/cours facilité

6.3 Limitations Identifiées

6.3.1 Complexité Technique

- Courbe d'apprentissage : Maîtrise des concepts cryptographiques requise
- Dépendances : Bibliothèque Charm-Crypto spécialisée
- Debugging : Diagnostic des erreurs cryptographiques complexe

6.3.2 Contraintes Pratiques

- Performance : Plus lente que chiffrement symétrique traditionnel
- Taille des données : Overhead du chiffrement par attributs
- Révocation : Pas de mécanisme intégré de révocation d'attributs

7. Conclusion

7.1 Bilan du Projet

Ce projet a démontré avec succès l'implémentation d'un système de protection des données basé sur CP-ABE dans le contexte universitaire de l'UTBM. L'objectif principal de créer un contrôle d'accès granulaire et sécurisé pour les résultats d'examens a été pleinement atteint.

7.2 Apports Techniques

7.2.1 Maîtrise Technologique

- Compréhension approfondie du CP-ABE et de ses mécanismes
- Implémentation pratique avec la bibliothèque Charm-Crypto
- Architecture logicielle modulaire et maintenable
- Tests exhaustifs validant la sécurité du système

7.2.2 Innovation Pédagogique

- Scénario réaliste inspiré de l'environnement UTBM
- Démonstration pratique des concepts théoriques de sécurité
- Code documenté facilitant la compréhension et réutilisation

7.3 Perspectives d'Évolution

7.3.1 Améliorations Techniques

- Interface graphique : Développement d'une UI pour utilisateurs non-techniques
- Révocation d'attributs : Implémentation de mécanismes de révocation
- Optimisation performances : Parallélisation des opérations cryptographiques
- Audit trail : Logging détaillé des accès et opérations

7.3.2 Extensions Fonctionnelles

- Multi-cours : Extension à plusieurs cours simultanément
- Granularité fine : Attributs plus spécifiques (semestre, groupe, etc.)
- Intégration LDAP : Connection avec l'annuaire UTBM
- Stockage cloud : Déploiement sur infrastructure cloud sécurisée

7.4 Conclusion Finale

L'implémentation réussie de ce système CP-ABE démontre la viabilité et l'efficacité des technologies de chiffrement par attributs pour la protection des données académiques sensibles. Au-delà de l'aspect technique, ce projet illustre comment les concepts cryptographiques avancés peuvent répondre aux défis concrets de sécurité dans l'enseignement supérieur.

La flexibilité du CP-ABE, combinée à sa robustesse cryptographique, en fait une solution particulièrement adaptée aux environnements universitaires où les rôles sont variés et les besoins d'accès complexes. Cette implémentation constitue une base solide pour de futures applications dans l'écosystème numérique de l'UTBM.

Technologies : Python, Charm-Crypto, CP-ABE

LES AUTRES QUESTIONS DU TP

1.1 Initialisation du système

---

1. Explication de SS512

SS512 fait référence à une courbe elliptique supersingulière avec un corps de base de 512 bits. Plus précisément :

SS = "Supersingular" (supersingulière)

512 = taille du corps de base en bits

Cette courbe permet de construire des appariements bilinéaires efficaces, qui sont essentiels pour le CP-ABE

Elle offre un niveau de sécurité équivalent à environ 1024 bits en RSA

2. Définition de mpk et msk

mpk (Master Public Key) : Clé publique maître

Utilisée pour le chiffrement des données

Partagée publiquement avec tous les utilisateurs

Permet de chiffrer avec une politique d'accès spécifique

msk (Master Secret Key) : Clé secrète maître

Gardée secrète par l'autorité de confiance

Utilisée pour générer les clés privées des utilisateurs

Permet de dériver des clés basées sur les attributs

Différence principale : mpk est publique (chiffrement), msk est privée (génération de clés utilisateurs).

3. Courbes elliptiques supersingulières et leur rôle

Une courbe elliptique supersingulière a ces propriétés particulières :

Trace nulle : la trace de l'endomorphisme de Frobenius est 0

Appariements efficaces : permettent de construire des fonctions d'appariement bilinéaires

Sécurité cryptographique : résistantes au problème du logarithme discret elliptique

Rôle dans CP-ABE :

Les appariements bilinéaires permettent de "multiplier" des éléments chiffrés

Essentiels pour évaluer les politiques d'accès (portes ET/OU)

Permettent le déchiffrement conditionnel basé sur les attributs

---

1.2 Définition de la politique d’accès et chiffrement du message

---

1 Formule Mathématique associée à l’opération de chiffrement en utilisant mpk

Voir doc https://ieeexplore.ieee.org/document/4223236 Encrypt(PK, M,$\mathcal{T}$)

2 Utilisateurs AUTORISÉS :

Utilisateurs possédant ONE ET TWO simultanément

Utilisateurs possédant THREE

En guise d'exemple on a

Attributs ["ONE", "TWO"] → AUTORISÉ

Attributs ["THREE"] → AUTORISÉ

Attributs ["ONE"] → REFUSÉ (manque TWO)

Attributs ["TWO"] → REFUSÉ (manque ONE)

Attributs ["FOUR"] → REFUSÉ (aucune condition satisfaite)

1.3 Génération des clés utilisateurs

---

1 Formule mathématique utilisée pour générer les clés des utilisateurs à partir de mpk et msk

Voir doc https://ieeexplore.ieee.org/document/4223236 KeyGen(MK, S)

2 Politique rappel : (ONE and TWO) or THREE

Analyse des clés générées :

User1 avec attributes_user1 = ['ONE', 'TWO']

sk1 contient les composants pour ONE et TWO

AUTORISÉ : satisfait la condition (ONE and TWO)

User2 avec attributes_user2 = ['ONE']

sk2 contient uniquement le composant pour ONE

REFUSÉ : ne satisfait ni (ONE and TWO) ni THREE

Manque l'attribut TWO pour compléter la première branche

Conclusion : Seul User1 peut déchiffrer le message car ses attributs satisfont la politique d'accès définie.

1.4 Déchiffrement

Formule mathématique : https://ieeexplore.ieee.org/document/4223236 Decrypt(CT, SK)