鍵失効機能を持つ属性ベース暗号の実装評価

苦木 大輔 †　　内田 恵 †　　近藤 伸明 †　　五十嵐　寛 ‡

† 株式会社　神戸デジタル・ラボ　　‡ 金沢工業大学

あらまし 近年，クラウドコンピューティングが脚光を浴びている一方，企業が利用する場合，クラウドサービス内に預けたデータの改ざん・漏洩といった危険性が不安視され，普及の足枷となっている．これらの不安感を払拭するため，利用者が暗号化してデータを保管するという対策が考えられるが，多数の利用者がデータを共同で利用する事が想定されるクラウドサービスにおいては既存の公開鍵方式は適していない．この問題を解決するため，利用者が保有する属性を指定して暗号化を行う属性ベース暗号方式が提案されている．本稿では，鍵失効機能を持つ属性ベース暗号方式の事業化を目指し，ウインドウズ及びアンドロイド端末上で実装及び性能評価を行う．

Implementation and Evaluation of Attribute Based Encryption with Revocation Function.

Daisuke Nigaki†　　Megumi Uchida†　　Nobuaki Kondo†　　Yutaka Igarashi‡

†Kobe Digital Lab.
Eiko Bldg. 5F, Edocho 93, Kobe city 650-0033, JAPAN
kondo@kdl.co.jp

‡College of Engineering, Kanazawa Institute of Technology
7-1, Ougigaoka, Ishikawa city 921-8501, JAPAN

Abstract Although Cloud Computing has attracted a great deal of attention for years, it also raises a lot of security concerns. Especially the integrity and the confidentiality of the data stored in the cloud. In order to ease these concerns, applying encryption seems to be the applicable solution. But since the data will be shared among users, existing public key cryptography becomes an inadequate approach. Thus we propose the use of Attribute-Base Encryption to solve the aforesaid problem. In this paper, we will implement our Attribute-Based Encryption system with key revocation, and evaluate the performance on both Windows and Android mobile.

1　はじめに

近年，脚光を浴びているクラウドコンピューティングは利用時に，プライバシー情報や機密性の高いデータをクラウドサービス提供者に渡して処理を行うため，データの機密性保護に関するセキュリティ上の問題が普及の妨げとなっている．本研究は，クラウドコンピューティングの上記の問題を解消した安全・安心なクラウドコンピューティングサービスを提供する基盤を構築する事を目的とした属性ベース暗号方式の実装し性能評価を行った．クラウド内のデータを暗号化して保管を行い，クラウド運用者や他のユーザが機密データを閲覧する事を防止し，万が一の機密データの漏洩時にも適切なユーザの秘密鍵が無いとデータを復号できないようにする．暗号化方式について，現在広く利用されている共通鍵暗号方式と公開鍵暗号方式は，ク

ラウドコンピューティングのような，多数のユーザが大規模なデータを共有して使用する新しいネットワーク利用形態においては，鍵管理効率などの問題があり，新たに様々な暗号理論が提案されている．その一つが属性ベース暗号方式であり，ユーザが持つ属性を指定し暗号化を行い，指定した条件を満たす属性を持つユーザのみが復号できる方式である．今回は，更に鍵管理効率を向上させるために失効リスト機能を属性ベース暗号方式に追加し，クラウドコンピューティング利用に適したデータ共有システムの構築を行う．

2　関連研究

Sahai, Waters がアクセス制限機能を持つ暗号方式として属性ベース暗号の提案を始めに行った． [2] そのスキームでは，従来の公開鍵暗号方式とは異なり，一人のユーザに対してのデータを暗号化ではなく，一つの暗号文に対しての各ユーザの秘密鍵が適合すれば，復号できる方式である．秘密鍵の内部に，復号できる属性の条件が含まれており，暗号文が持つ条件を満たす場合のみ復号する事が可能になる．Goyal らは，より一般的な属性暗号方式を提案した．ユーザの秘密鍵の中に持つアクセス権限をアクセスツリーと呼ばれる木構造を用いて表した．[3] Bethencourt らは，属性ベース暗号方式において，暗号文に復号条件を持たせる方式の提案を行った．Pirretti らは，鍵失効機能を追加した属性ベース暗号方式の提案を行った．[5] [4] において満永らは、鍵失効機能を追加した属性ベース暗号方式の実装および性能評価を行った。

3　属性ベース暗号方式

3.1　利用フロー

本研究では，属性ベース暗号方式に失効リストを追加した新方式の研究・実装をおこなった．利用フローは下図の通りである．

1. （鍵生成局）マスター鍵の作成
2. （各ユーザ）属性に応じた秘密鍵の作成依頼
3. （鍵生成局）身元を確認して個人の秘密鍵を送付
4. （鍵生成局）各ユーザの ID と H（ID）の組み合わせを失効リスト管理者に通知
5. （鍵生成局）公開鍵を公開
6. （送信者）復号できる人の属性を指定しデータを暗号化してデータベースに格納
7. （受信者）暗号化されたデータを復号・利用（ただし，自分の属性に合うデータのみ利用可能）．また復号時に H（ID）を突合し，秘密鍵が失効していないか確認

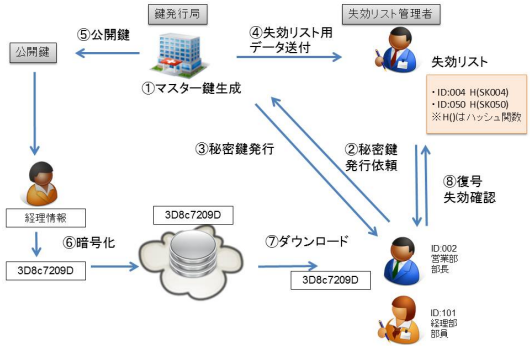

図 1: 属性ベース暗号の利用フロー

3.2　実装方式

今回，Android 端末において動作する属性ベース暗号方式を，アクセスツリー構造を持つ [1] の方式に基づき 160bit の楕円曲線上で実装評価を行った．各パラメータおよび機能は下記の通りである．

- マスター鍵：mk
- 公開鍵：pk
- (ユーザ x の)ID：ID_x
- (ユーザ x の) 属性：att_x
- (ユーザ x の) 秘密鍵：sk_x
- ファイル：M
- 暗号化されたファイル：C

図 2: Android 端末の操作画面

- 復号条件関数：*policy*()
　ユーザ x の属性が復号条件を満たすとき $policy(att_x) = 1$　とする

1. Setup() → *pk*, *mk*
   公開鍵およびマスター鍵を生成する
2. Keygen(*mk*, att_x) → sk_x
   マスター鍵とユーザ x の属性から，ユーザ x の秘密鍵を生成する
3. Encrypt(*pk*,*M*,*policy*()) → *C*
   復号条件を指定しファイルの暗号化を行う
4. Decrypt(*C*,sk_x)　→ *M*, if policy(att_x) =1
   ユーザの属性が復号条件を満たすときのみ復号を行う
5. Revoke(ID_x)
   ユーザ x の鍵を失効させる

4　評価と考察

4.1　Android アプリケーション

[4] では、Linux 環境の PC 上で、失効機能を持つ属性ベース暗号方式の実装および性能評価を行った．今回，Android 端末上で動作する暗号化と復号機能の実装を行った．商用化を目指し利用しやすいアプリケーション実装を目的としているため，少ない操作でデータ保管用ファイルサーバと暗号化ファイルおよび復号ファイルとの通信を行えるよう「Encrypt and Upload」と「Download and Decrypt」という複合的な機能実装を行った．

4.2　実験概要

表 1 に示す性能の計算機を用いて，属性ベース暗号方式の暗号化・復号にかかる処理時間の測定を行う．理論的に，暗号化および復号の計算処理コストは、ファイルの容量と復号条件の属性数に依存する．本研究では，(1) ファイル容量に対する暗号化処理時間の変化，(2) 復号条件の属性数に対する暗号化処理時間の変化，(3) ファイル容量に対する復号処理時間の変化，(4) 復号条件の属性数に対する復号処理時間の変化の測定を行った．なお「Ecnrypt and Upload」と「Download and Decrypt」とも、暗号化処理・復号処理に加え、ファイルサーバとクライアント間の転送時間も含まれている．

表 1: 実験環境

Android 端末	
OS	Android 2.2
CPU	NVIDIA Dual Cortex-A9 1GHz
Memory	512M
通信速度 (ベストエフォート)	
端末→サーバ	5.4Mbps
サーバ→端末	11Mbps

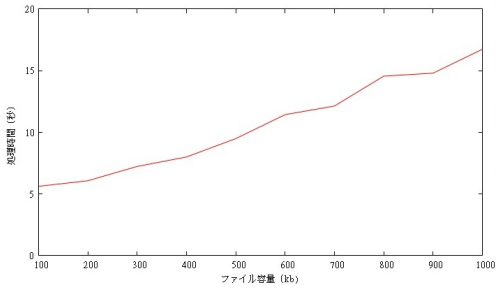

図 3: ファイル容量に対する暗号化処理時間

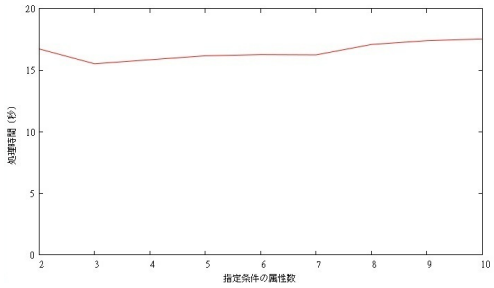

図 4: 復号条件の属性数に対する暗号化処理時間

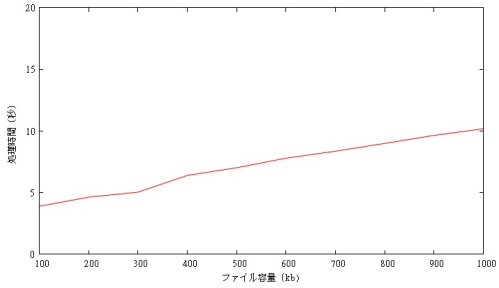

図 5: ファイル容量に対する復号処理時間

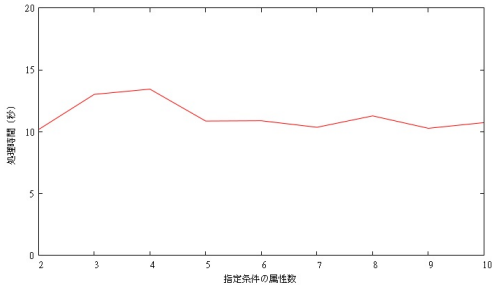

図 6: 復号条件の属性数に対する復号処理時間

5 まとめ

本稿では，属性ベース暗号の実装を行い，性能評価を行った．暗号化および復号に要するの処理時間は図に示す通りであり，実際にサービス提供可能なものであると考えられる．大容量のファイルを取り扱ったり、より大規模なクラウドサービス環境での利用を想定すると，処理速度の向上が必要となる．今後，処理速度を早くするために，より効率的なデータ構造やアルゴリズムの研究を行う．

謝辞

本研究の一部は，経済産業省「平成 23 年度企業・個人の情報セキュリティ対策促進事業（新世代情報セキュリティ研究開発事業）」の委託研究に基づいて行った．

参考文献

[1] J.Bethencourt, A.Sahai, and B.Waters. Ciphertext-Policy Attribute-Based Encryption , 2007. Proceedings of IEEE Symposium on Security and Privacy, pp. 321-334, 2007.

[2] W.Sahai and B.Waters. Fuzzy Identity Based Encryption, 2005. Proceedings of Eurocrypt 2005, volume 3494 of LNCS, pp. 457-473.

[3] A.Sahai V.Goyal, O.Pandey and B.Waters. Attribute-based encryption for fine-grained access control of encrypted data, 2006. Proceedings of the 13th ACM conference on Computer and communications security 2006, pp. 89 - 98.

[4] 満永拓邦，オマール イスマイル，久野祐介，田所成久，近藤伸明，藤木裕之，五十嵐寛 鍵失効機能を持つ属性ベース暗号の実装評価 情報処理学会第 73 回全国大会， 2011.

[5] P.McDaniel M.Pirretti, P.Traynor and B.Waters. Secure Atrribute-Based Systems, 2006. Proceedings of the 13th ACM conference on Computer and communications security 2006, pp. 99-112.