グループ間でのファイル共有を柔軟かつ安全に行うための新方式検討*

辛　星漢†　　古原　和邦†　　今井　秀樹‡†

†産業技術総合研究所情報セキュリティ研究センター
305-8568 茨城県つくば市梅園 1-1-1
seonghan.shin@aist.go.jp

‡中央大学理工学部
112-8551 東京都文京区春日 1-13-27

あらまし 本稿では、クラウドストレージサービス（特に、パブリックモデル）においてグループ間で安全にかつ柔軟にデータを共有できる様々な方式について検討する。

On Flexible and Secure File Sharing for Group Members

SeongHan Shin†　　Kazukuni Kobara†　　Hideki Imai‡†

†Research Center for Information Security, AIST
1-1-1 Umezono, Tsukuba City, Ibaraki 305-8568, JAPAN
seonghan.shin@aist.go.jp

‡Faculty of Science and Engineering, Chuo University
1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, JAPAN

Abstract In this paper, we discuss several possible solutions, allowing group members to share data flexibly and securely in the usage of cloud storage services.

1　はじめに

近年、コンピュータ資源をネットワーク経由でサービスとして提供するクラウドコンピューティングへの関心が高まっている。特に、利用者がどこでも自分のデータにアクセスできるクラウドストレージサービス（例えば、Google GDrive, Apple iCloud, Amazon S3 など）はすでに利用できる状況である。なお、クラウドストレージサービスを使うことで第三者もしくはグループ間で手軽にデータが共有できる利点も挙げられる。その一方、すべてのデータがクラウドサービス提供者の管理下に置かれるため、センシティブなデータのセキュリティが大きな障害になりつつある。

1.1　問題設定及び本稿の目的

本稿では、次のような状況を考える。利用者はあるグループメンバー（$n \geq 1$）と共有したいデータ（例えば、各種ファイルなど）をクループ鍵（GK）で共通鍵暗号で暗号化して、その暗号文をクラウドサービス提供者が運用するストレージに保存する。当然、その利用者は暗号化に使われたクループ鍵（GK）を他のメンバーに安全に渡さなければならない。しかしながら、グループ鍵を共有する既存方式は以下のいずれかの問題を抱えている。

- 悪意のある管理者或いはサーバ侵入者による共有データまたは利用者の（認証用のハッシュ化された）パスワード[1]への不正アク

本成果の一部は経済産業省次世代高信頼・省エネ型 IT 基盤技術開発事業の助成により行われた。また、辛星漢は一部科研費 22760285 の助成を受けた。

[1]計算能力の向上により、ハッシュ化されたパスワードから元のパスワードを求めることは容易になってきている [14]。そして、多くの利用者は同じパスワードを複数の

セス

- 紛失した或いは盗難された端末が解析されることによる共有データまたは利用者のパスワード[2]への不正アクセス
- グループ鍵や端末を紛失することによるデータへのアクセス権の喪失

本稿の目的はクラウドサービスにおいて「グループ間で手軽にデータを共有できる」機能を有効にしつつ、上記の問題を解決する方法について検討を行うことである。

2 安全性及び機能要件

ここでは、クラウドストレージサービスの利用にあたって1つの利点である「グループ間で手軽にデータを共有する機能」に対して、以下の安全性及び機能要件を付け加えることを考える。

要件１（利便性）： 利用者が記憶すべき情報は短いパスワード１つのみでよいこと

要件２（サーバの不正や漏洩に対する安全性）： 悪意のある管理者およびサーバに侵入した攻撃者が共有データまたは利用者のパスワードを入手できないこと

要件３（可用性）： 利用者が端末を紛失したり、端末が盗難されたとしても共有データは入手可能であること

要件４（利用者からの漏洩に対する安全性）： 利用者の端末やそのバックアップを入手した攻撃者が共有データまたは利用者のパスワードを入手できないこと

要件５（時間差の漏洩に対する安全性）： 時間差でサーバと利用者の端末両方に記録されている情報を入手した攻撃者が共有データまたは利用者のパスワードを入手できないこと

3 既存方式

本節では、クラウドストレージサービスの利用においてグループ鍵（GK）を共有する既存方式がそれぞれ前述の安全性及び機能要件をどこまで満たすかについて検討する。

3.1 方式Ｉ：公開鍵暗号を利用する方法

方式Ｉは各利用者に公開鍵・秘密鍵対を持たせ、グループで共通鍵 GK を共有する際には、形成したいグループメンバーの公開鍵を入手し、それらの公開鍵で GK を暗号化して送信する[3]。

グループを作成したい利用者がオンライン上にいなくとも、作成したい人の都合でグループを作れるという利点があるが、この方式では秘密鍵とそのバックアップをどこかに保管するかが課題となる。多くの場合、以下のいずれかの方法で運用されるがそれぞれ問題を抱えている。

- （パスワードで暗号化された）秘密鍵が利用者の端末に置かれる。この場合は、利用者からの漏洩によりパスワードのオフライン全数探索が可能になるため、要件４、５を満たさない。
- プロキシサーバ上に（パスワードで暗号化された）秘密鍵を置き、ネットワーク経由でそのプロキシサーバ―に接続する。この場合は、サーバからの漏洩によりパスワードのオフライン全数探索が可能になるため、要件２、５を満たさない。
- 利用者の端末或いはサーバに耐タンパー装置を設け、そこに秘密鍵を入れる。これに対しては、さまざまなサイドチャネル攻撃（例えば、[10] など）により鍵を抜き出す方法が知られている。

サービスで使いまわしており、1つのパスワードが分かると被害が他のサービスに及ぶことになる。

[2]公開鍵暗号の秘密鍵をパスワードで暗号化したり、共通鍵暗号の暗号鍵としてパスワードをそのまま利用している場合は、パスワードで暗号化された秘密鍵や冗長性のある共通鍵暗号文を入手できればパスワードのオフライン全数探索により元のパスワードを求めることが可能である。

[3]IETF XMPP WG でも S/MIME を使った同様の議論が行われた [13]。

また、PGP, S/MIME など既に利用可能なツールが多数存在するが、鍵生成とバックアップ、相手の鍵の入手、鍵への署名と信用度の設定など、一般利用者には難解な処理が多いこともあり、一般には広く普及していない。なお、この利便性の問題を解決するために、IETF PKIX WG では、サーバ側で公開鍵・秘密鍵対を生成し、サーバで秘密鍵をバックアップする方法 [16] について議論を再開している[4]。しかしながら、この方法も要件４、５およびサーバでの秘密鍵バックアップにより一時的要件２が満たされない。

3.2 方式 II：マスター鍵を用いる方法

方式 II はマスター鍵を KDC (Key Distribution Center) 或いは KGC (Key Generation Center) に持たせる方法で KPS（例えば、[18]）、ID ベース暗号（例えば、[17, 6]）、関数暗号（例えば、[7]）などがある。利用者がグループで共通鍵 GK を共有する際には、グループメンバーだけが復号できるポリシーに基づいて GK を暗号化して送信する。

この方式では、利用者の秘密鍵または暗号文にポリシーを埋め込むことでグループ鍵管理を柔軟に行える利点がある。例えば、Cheung ら [8] はグループ鍵の forward/backword secrecy を確保するために CP-ABE (Ciphertext-Policy Attribute-Based Encryption) を使ってグループメンバーへ鍵を再分配しなくても membership revocation を可能にした。しかし、利用者は自分の秘密鍵とそのバックアップを安全に保管しなければならないため、方式 I と同じように要件４、５を満たさない。また、KDC や KGC からマスター鍵が漏洩されてしまうとすべてのグループ鍵を復元することができる[5]ため、KDC や KGC は要件２、５を満たさない。

[4]同様の提案は 2001 年、2005 年にも行われたが、安全性上公開鍵・秘密鍵対は利用者側で生成することが原則であることもあり、Internet-Draft の有効期限は切れた。ただし、利用者のスキルやバックアップなどのことを考えると、このような運用にせざるを得ないのではないかという機運は高まりつつある。

[5]ID ベース暗号では key escrow 問題と呼ばれる。

3.3 方式 III：グループ鍵共有プロトコルを使う方法

方式 III はグループを形成したメンバーとオンラインでグループ鍵共有（Group Key Exchange (GKE)）プロトコルを実行する方法である。通常、能動的な攻撃者に安全な GKE はグループメンバーの間で行われる認証手段によって大きく PKI ベース GKE（例えば、[4, 12, 11]）、パスワードベース GKE（例えば、[2, 1]）などに分けられる。

この方式では、GKE を共有するデータごとに実行することでグループ鍵（GK）の forward secrecy が容易に達成できる利点がある。しかしながら、PKI ベース GKE においては、方式 I, II と同様に、利用者の秘密鍵とそのバックアップを安全に保管しなければならないため、要件４、５を満たさない。また、パスワードベース GKE はグループごとに異なるパスワードをメンバーの間で設定しないといけないため要件１を満たさないし、常に短いパスワードに対するオンライン全数探索が可能になる問題がある。

4 提案方式

ここでは、２節で示したすべての安全性及び機能要件を満たすために、LR-AKE (Leakage-Resilient Authenticated Key Exchange) のクラスタモード [15] を用いて既存方式を改良する方法について検討する。基本的なアイディアは利用者の端末とクラウドサービス提供者が管理する認証サーバの間で LR-AKE を認証方式として採用し、クラウドの public model における要件３（可用性）を満たすためにサーバを２台使うクラスタモードにすることである。ちなみに、LR-AKE のクラスタモード [15] では２台のサーバが結託したとしてもそこで分散保存されたデータ鍵（dk）が復元できないようになっている。

これから、LR-AKE のクラスタモードを LR-AKE クラスタに、利用者の端末を LR-AKE クライアントに、クラウドサービス提供者により管理される認証サーバを LR-AKE サーバに呼ぶことにする。

4.1 既存方式 I を改良する方法

既存方式 I の主な問題点は、利用者の秘密鍵がそのまま或いはパスワードにより暗号化された状態で利用者の端末又はプロキシサーバ上に保存されることにある。しかしながら、この欠点は利用者の秘密鍵を LR-AKE クラスタ [15] のデータ鍵と同じように LR-AKE クライアントと 2 台のサーバに分散保存することで解決できる。これにより、グループを形成したい利用者がオフラインでグループを作れる利点を保ちながら安全性と機能要件を満たすことができる。

また、LR-AKE クライアントが自動的に公開鍵・秘密鍵対を生成し、LR-AKE により設立された安全な通信路を通してその生成された鍵の公開鍵証明書の取得処理や他人の公開鍵の検索、失効確認などの処理を行えば、利用者は煩わしい処理を意識することなく短にパスワードとグループ鍵（GK）を共有したいメンバーの ID を LR-AKE に打ち込むだけで、グループ鍵を共有できるため利便性の問題も解消できる。また、異なる LR-AKE サーバを利用している利用者間においても、公開鍵の信頼性は公開鍵証明書や公開鍵への電子署名を使うことなどにより確保できるため、グループ鍵を容易に共有することが可能である。ただし、これらの利便性を実現するためには、全体の構成や実装が複雑になるという欠点がある。

4.2 既存方式 II を改良する方法

既存方式 II の問題点は利用者の秘密鍵の保存場所とそのバックアップであるが、これは前述のとおり、秘密鍵を LR-AKE クラスタ [15] で分散保存することにより解決できる。また、もう一つの問題点であるマスター鍵の漏洩に対しては、マスター鍵を複数の KGC に分散させ閾値以上の KGC が協力して利用者の秘密鍵を発行したり [6]、複数の KGC から発行された秘密鍵を組み合わせて利用者の秘密鍵を生成したり [9] することで解決できる。なお、LR-AKE クラスタ [15] を用いた際には以下の二通りが考えられる。これらの利点としては、マスター鍵を短い期間で更新できることがある。

4.2.1 Functional Encryption with LR-AKE (FEL)

これは 4.1 節と同様に、ID ベース暗号や関数暗号で使われる利用者の秘密鍵を LR-AKE クラスタに分散保存して、グループ鍵を取得（つまり、暗号文を復号）する際には LR-AKE サーバに接続し秘密鍵を復元する仕組みである。

既存方式 II では、関数暗号を使って柔軟なグループ鍵管理ができる利点とクループメンバーはサーバに接続することなくグループ鍵（GK）が共有できる利点があった。しかし、FEL においては LR-AKE サーバへの接続が前提となるため、後者の利点の恩恵は小さく、処理の重いペアリング演算を導入してまでこの機能を維持する必要性は小さい。

4.2.2 Pairing-Less Functional Encryption with LR-AKE (PFE)

既存方式 II を LR-AKE クラスタと組み合わせる場合は、通常の ID ベース暗号や関数暗号で必要となるペアリング演算を省略できる。このペアリング演算を省略した方式を Pairing-Less Functional Encryption with LR-AKE (PFE) と呼び、その具体的な構成例を以下に提案する。基本的に関数暗号自体は使わないが、利用者がグループ鍵（GK）へのポリシーを作成しそのポリシーに従って LR-AKE サーバがグループメンバーへ GK を転送する仕組みである。なお、LR-AKE サーバはグループ鍵の生成に必要となるマスター鍵（MK）を持っている。

1. 利用者と LR-AKE サーバは LR-AKE を使って相互に認証された安全な通信路を確立する。
2. 利用者は、グループ鍵を共有したいメンバーの ID の集合 G および乱数 N を 1 で確立した安全な通信路を通して LR-AKE サーバに送る。（なお、乱数 N は同じグループ中で異なるクループ鍵を生成する場合などに用いる。）
3. LR-AKE サーバは、集合 G の中に 1 で認証した利用者の ID が含まれているかを確認

し、含まれていなければ追加する。（また、IDの代わりに属性を指定することも可能である。その場合、サーバは1で認証された利用者が指定された属性を持っていることを確認し、持っていなければ処理を中断する。その際、指定内容がIDなのか属性ATなのかを区別するために、フィールドの識別子も用いる必要がある。）

4. LR-AKEサーバは集合GのIDをソートし、ソートされたIDを乱数Nと連結し、それとマスター鍵（MK）を使ってグループ鍵（GK）を生成する。生成関数として例えばHMACを使う場合、以下のように計算できる。

   $GK = \mathsf{HMAC}_{MK}$ (N:nonce||ID:id1||ID:id2||AT:attribute1||$\cdots$)

   ここで、N:, ID:, AT:はその後の文字列の種類を表す識別子であり、|| は連結を表すが、単に文字列を連結するのではなく、セパレータシンボルを挟んで連結することを意味する。セパレータシンボルはnonce, id, attributeなどで指定できてはならない。

5. LR-AKEサーバは、4で生成されたGKをグループメンバーにそれぞれLR-AKEにより確立された安全な通信路を通して送信する。

4.3 既存方式IIIを改良する方法

既存方式IIIの問題点は、利用者の秘密鍵の管理或いはグループごとに異なるパスワードの設定にあるが、これらは前述のとおり、秘密鍵や異なるパスワードをLR-AKEクラスタ [15] で分散保存することにより解決できる。ここで、GKE認証用のパスワードは利用者が覚える必要がないことに注目されたい。なお、LR-AKEクラスタ [15] を用いた際には以下の二通りが考えられる。

4.3.1 GKE with LR-AKE

単純な改良として、GKE認証用の秘密鍵やパスワードをLR-AKEクラスタに分散保存して、グループ鍵を共有する際には該当するGKEプロトコルをグループメンバー間で実行する仕組みである。特に、パスワードベースGKEを使う場合はグループごとに異なるパスワードをオフライン全数探索ができない範囲（例えば、128ビット）のランダムな乱数にしておく。

4.3.2 LR-AKEベースGKE

そもそもLR-AKEを使う時点で、グループメンバーはLR-AKEサーバに接続しなければならないため、より自然な改良としてLR-AKEの認証に基づいたGKEが考えられる。つまり、データを共有したいメンバーはそれぞれLR-AKEサーバと相互認証を行い、そこで確立された安全な通信路を通して(unauthenticated) GKE（例えば、[5, 3] など）を実行する。もちろん、すべてのメッセージはLR-AKEサーバ経由で転送されるため、グループ鍵（GK）の秘匿性はsemi honestなサーバに対して成り立つ。

4.4 LR-AKEのグループアカウントを利用する方法

これはLR-AKEにおけるアカウント間連携機能をグループで利用する方法である。LR-AKEでは各利用者に対して複数のアカウントを発行し（ポリシーにより利用者が発行することも可能）、それらを複数の端末で利用することが可能である。これらの端末間ではLR-AKEのアカウント間連携機能により、LR-AKEクラスタに分散保存したデータ（鍵）を安全に共有することができる。なお、ここで分散共有するデータは、盗聴者は勿論、端末を盗んだり取得したりした攻撃者やサーバに侵入した攻撃者に対しても秘匿できる。

この方式はLR-AKEを導入するだけで利用できるという利点もあるが、欠点としては作成するグループ毎にLR-AKEアカウントを各利

用者に安全に配布しなければならない点が挙げられる。

5 まとめ

本稿では、クラウドストレージサービス（特に、パブリックモデル）の利用においてグループ間で安全にかつ柔軟にデータ鍵を共有するために、まず満たすべき安全性及び機能要件をまとめて、それに基づいて既存方式を分析した後、されにそれぞれの既存方式を LR-AKE クラスタ [15] を用いて改良する方法について検討した。

参考文献

[1] M. Abdalla, E. Bresson, O. Chevassut, and D. Pointcheval, "Password-based Group Key Exchange in a Constant Number of Rounds", In *Proc. of PKC2006*, LNCS 3958, pp. 427-442, Springer-Verlag, 2006.

[2] E. Bresson, O. Chevassut, and D. Pointcheval, "Group Diffie-Hellman Key Exchange Secure against Dictionary Attacks", In *Proc. of ASIACRYPT2002*, LNCS 2501, pp. 497-514, Springer-Verlag, 2002.

[3] E. Bresson, O. Chevassut, and D. Pointcheval, "The Group Diffie-Hellman Problems", In *Proc. of SAC2002*, LNCS 2595, pp. 325-338, Springer-Verlag, 2002.

[4] E. Bresson, O. Chevassut, D. Pointcheval, and J. J. Quisquater, "Provably Authenticated Group Diffie-Hellman Key Exchange", In *Proc. of 8th ACM-CCS*, pp. 255-264, ACM, 2001.

[5] M. Burmester and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System", In *Proc. of EUROCRYPT'94*, LNCS 950, pp. 275-286, Springer-Verlag, 1994.

[6] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing", In *Proc. of CRYPTO2001*, LNCS 2139, pp. 213-229, Springer-Verlag, 2001.

[7] D. Boneh, A. Sahai, and B. Waters, "Functional Encryption: Definitions and Challenges", In *Proc. of TCC2011*, LNCS 6597, pp. 253-273, Springer-Verlag, 2011.

[8] L. Cheung, J. A. Cooley, R. Khazanand, and C. Newport, "Collusion-Resistant Group Key Management Using Attribute-Based Encryption", In *Proc. of International Workshop on Group-Oriented Cryptographic Protocols*, 2007.

[9] L. Chen, K. Harrison, N. P. Smart, and D. Soldera, "Applications of Multiple Trust Authorities in Pairing based Cryptosystems", In *Proc. InfraSec2002*, LNCS 2437, pp. 260-275, Springer-Verlag, 2002.

[10] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems", In *Proc. of CRYPTO'96*, LNCS 1109, pp. 104-113, 1996.

[11] H. J. Kim, S. M. Lee and D. H. Lee, "Constant-Round Authenticated Group Key Exchange for Dynamic Groups", In *Proc. of ASIACRYPT2004*, LNCS 3329, pp. 245-259, Springer-Verlag, 2004.

[12] J. Katz and M. Yung, "Scalable Protocols for Authenticated Group Key Exchange", In *Proc. of CRYPTO2003*, LNCS 2729, pp. 110-125, Springer-Verlag, 2003.

[13] M. Miller, "End-to-End Object Encryption & Signing", IETF XMPP WG, July 2001. Available at `http://www.ietf.org/proceedings/81/slides/xmpp-3.pdf`.

[14] P. Oechsl, "Making a Faster Cryptanalytic Time-Memory Trade-Off", In *Proc. of CRYPTO2003*, LNCS 2729, pp. 617-630, Springer-Verlag, 2003.

[15] 恩田 泰則、辛 星漢、古原 和邦、今井 秀樹、「クラウド環境に適したオンラインデータ分散管理方式」、*2011* **年暗号と情報セキュリティシンポジウム**、3F2-3、2011.

[16] J. Schaad, S. Turner and P. Timmel "CMC Extensions: Server Key Generation", IETF Internet-Draft, July 2011. Available at `https://datatracker.ietf.org/doc/draft-turner-pkix-cmc-serverkeygeneration/`.

[17] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes", In *Proc. of CRYPTO'84*, LNCS 196, pp. 47-53, Springer-Verlag, 1985.

[18] T. Matsumoto and H. Imai, "On the Key Predistribution System: A Practical Solution to the Key Distribution Problem", In *Proc. of CRYPTO'87*, LNCS 293, pp. 185-193, Springer-Verlag, 1988.