Attribute-Based Encryption (ABE) とは?

2025 年 3 月 10 日

1. はじめに

本ドキュメントは、ソフトウェア開発者向けに、属性ベース暗号 (Attribute-Based Encryption, ABE) の基本概念、技術的側面、実際のユースケース、そして ABE における各関係者 (ロール) について説明するものです。なお、従来のユーザ単位の暗号方式 (たとえば、RSA を用いた公開鍵暗号方式など) とは異なり、ABE は「属性」という概念を利用してアクセス制御を実現します。

2. ABE の概要

2.1 基本コンセプト

属性による鍵管理:

従来の暗号方式 (例:RSA 暗号) では、各ユーザに対して個別の鍵 (公開鍵と秘密鍵) が発行され、送信者は受信者の公開鍵を用いて暗号化を行います。

Crypto++を利用した従来の暗号方式 RSA 暗号のコード例 (抜粋)

鍵ペアの生成

```
// 1. 鍵ペアの生成 (公開鍵 & 秘密鍵)
AutoSeededRandomPool rng;
InvertibleRSAFunction privateKey;
privateKey.Initialize(rng, 2048);  // 2048ビット鍵生成

RSA::PublicKey publicKey(privateKey);
RSA::PrivateKey rsaPrivate(privateKey);
```

暗号化

```
// 2. 平文の設定
string plaintext = "Hello, RSA Encryption!";
cout << "Original Text: " << plaintext << endl;

// 3. RSA暗号化
string encrypted;
RSAES_OAEP_SHA_Encryptor encryptor(publicKey);
StringSource(plaintext, true,
    new PK_EncryptorFilter(rng, encryptor,
        new StringSink(encrypted)
    )
);
```

復号

```
// 5. RSA復号
string decrypted;
RSAES_OAEP_SHA_Decryptor decryptor(rsaPrivate);
StringSource(encrypted, true,
    new PK_DecryptorFilter(rng, decryptor,
        new StringSink(decrypted)
    )
);
```

一方、ABE では、暗号化や復号のための鍵生成に「属性」（例：役職、所属部署、資格など）を利用し、属性に基づく柔軟なアクセス制御が可能です。

アクセス制御の柔軟性：

暗号化時に特定の属性や条件を指定することで、該当する属性を持つユーザのみが復号可能となります。ただし、ABE には大きく分けて 2 種類の方式（CP-ABE と KP-ABE）が存在します。

3. ABE における関係者（ロール）

従来の暗号方式では、暗号文の作成者（送信者）と復号者（受信者）の 2 者が中心となり、鍵の保証や管理については暗号技術の外側の仕組みに依存していました。これに対して、ABE では以下のような 3 つの関係者（ロール）が存在します。

暗号文作成者（送信者）

平文から暗号文を作成し、復号者に送信します。従来の方式と同様に、暗号化の役割を担います。

CP-ABE の場合、暗号化時に、特定の属性またはポリシーを設定して、誰が復号可能かを指定します。

復号者（受信者）

暗号文を受信し復号鍵を用いて復号して平文を得ます。従来の方式と同様に、復号の役割を担います。

ユーザは自らの属性情報に基づいて、復号可能かどうかが判断されます。

信頼できる機関（鍵発行機関：Key Generation Center、KGC）

ABE では、システムの根幹として信頼できる鍵発行機関（KGC）が存在します。

KGC はユーザの属性情報をもとに、復号鍵を生成・発行します。

この KGC が、暗号の仕組みそのものに組み込まれることで、鍵を保証する役割を担い、システム全体の安全性を確保します。

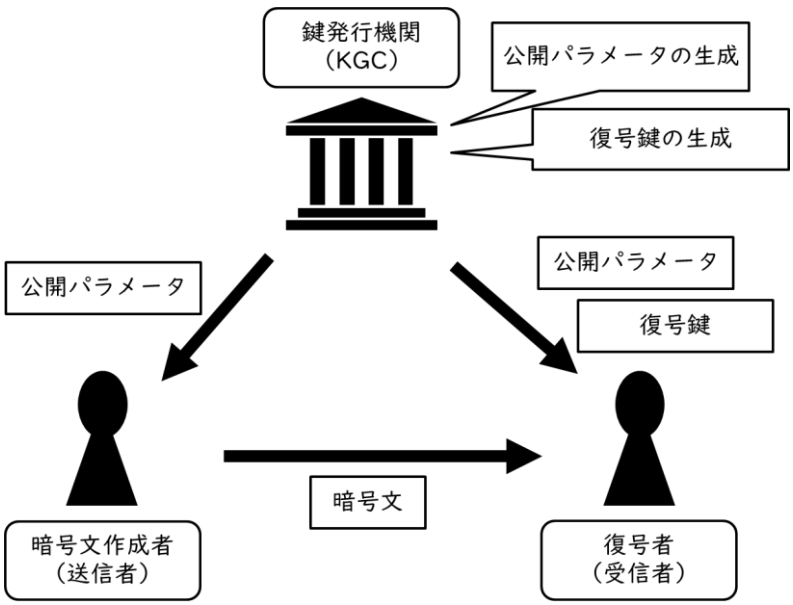

図 1：ABE における関係者

4. CP-ABE と KP-ABE の詳細と比較

4.1 CP-ABE (Ciphertext-Policy ABE)

概要:

暗号化の際に、暗号文に対してアクセス制御ポリシーを直接組み込む方式です。

送信者が「この属性（または条件）を満たすユーザのみが復号可能」というポリシーを暗号文に設定します。

具体例:

暗号化時に「管理者」または「営業部かつ正社員」といった条件を暗号文に組み込み、受信者は自身の属性セットがその条件を満たす場合にのみ復号が可能となります。

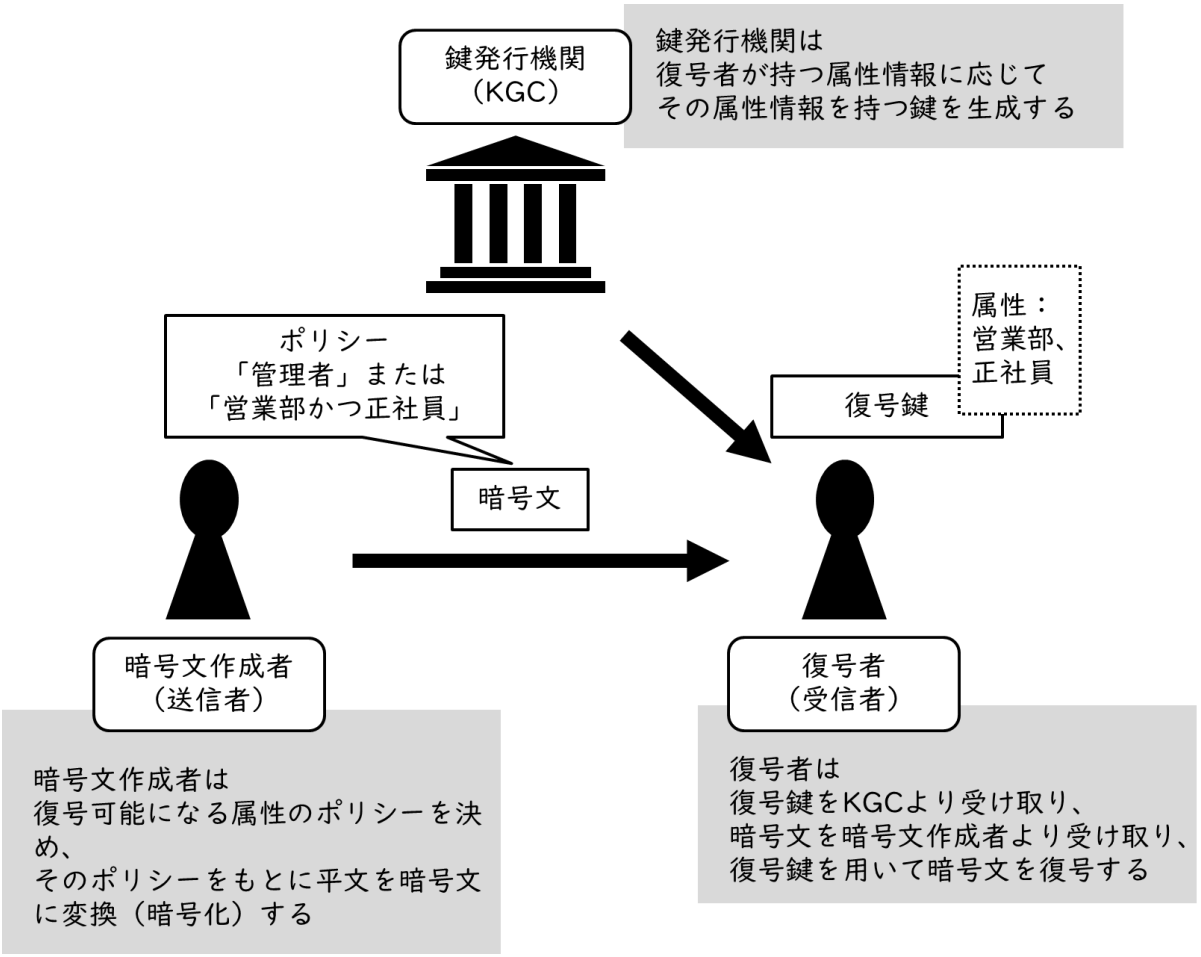

図 2:CP-ABE の例

4.2 KP-ABE (Key-Policy ABE)

概要:

ユーザの秘密鍵にアクセス制御ポリシーが組み込まれる方式です。

暗号文自体は属性情報のみを含み、どの暗号文を復号できるかは、ユーザに発行された秘密鍵内のポリシーによって決定されます。

具体例:

秘密鍵が「管理者」や「営業部かつ正社員」といったポリシーを保持しており、暗号文がその属性情報を含む場合に復号が可能となります。

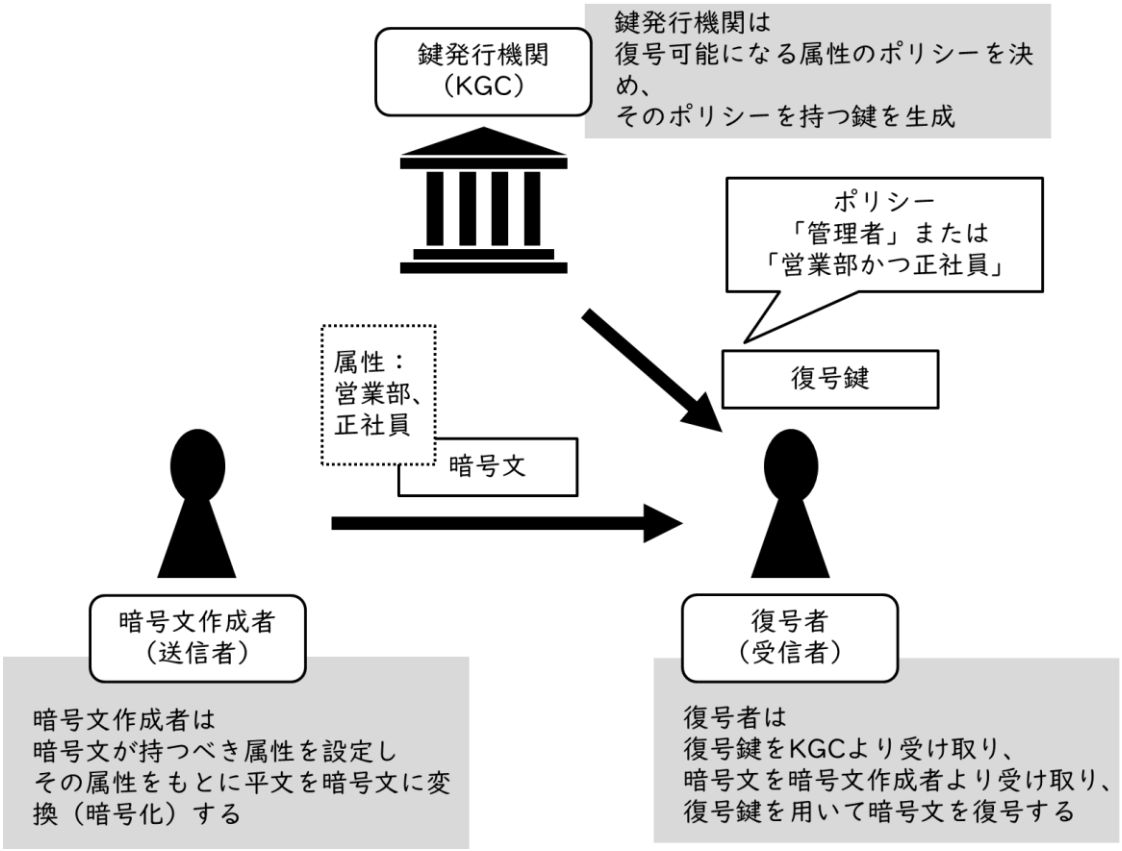

図 3:KP-ABE の例

4.3　比較（機能面・用途面）

機能面:

CP-ABE:

暗号化者がアクセス制御ポリシーを直接設定できるため、データの暗号化時に柔軟なポリシー適用が可能です。

KP-ABE:

鍵発行機関が各ユーザの秘密鍵にポリシーを組み込むため、運用側での一元的なポリシー管理が可能ですが、暗号化者側での柔軟な設定はできません。

用途面:

CP-ABE:

クラウドストレージや公開環境で、暗号化と同時にアクセス制御を明示的に設定したい場合に適しています。

KP-ABE:

中央の鍵管理システムが整備されており、鍵の発行や更新を一元的に管理できる環境において有用です。

ソフトウェア開発者による暗号ライブラリ・API の誤用の実態:

従来暗号であっても誤用が発生している事例があるが、より複雑な高機能暗号では誤用の可能性がさらに高いと予想される。

数学的複雑性および関与する当事者の増加:

高機能暗号は従来暗号と比較して、数学的な背景やプロトコルの複雑性が増すため、開発者が正しく実装することが難しくなる可能性がある。

誤用発生の比較と抑制手法の検討:

従来暗号と比較して、どのような点で誤用が生じやすいのか。

誤用を抑えるためのアプローチとして、従来暗号の既存研究の手法が有効か、もしくは高機能暗号に特有な対策が必要かを評価する。

5. ABE の技術的側面

5.1 アクセス制御の柔軟性

属性の組み合わせ:

複数の属性を組み合わせたポリシーにより、細かいアクセス権の設定が可能です。

動的な組織への適用:

組織の変化やユーザ属性の更新に伴い、鍵やポリシーを動的に管理できるため、変化するアクセスニーズに柔軟に対応できます。

5.2 管理の簡素化

個々の鍵発行の不要性:

従来の暗号方式では、各ユーザに対して個別に鍵を発行し管理する必要がありました。

ABE では、属性に基づいた鍵管理が可能なため、以下の点で大規模なシステムにおいても大きな利点があります:

スケーラビリティ: 多数のユーザに対して個別の鍵を発行・管理する手間が省かれ、システム全体の管理負担が軽減されます。

中央集約的なポリシー管理: システム全体で統一したアクセス制御ポリシーを設定・更新でき、セキュリティポリシーの一元運用が可能です。

鍵の更新と廃止の容易さ: ユーザ属性の変更や鍵の失効があった場合でも、個々の鍵を再発行するのではなく、属性やポリシーの更新で迅速に対応できます。

5.3 セキュリティの向上

細粒度のアクセス制御:

属性を用いることで、必要なユーザにのみアクセス権を厳密に付与でき、情報漏洩リスクの低減が図れます。

鍵管理の一元化:

属性管理システムと連携することで、全体的なセキュリティ管理が容易になり、従来の暗号システムよりも柔軟かつ効率的な鍵保証が実現されます。

実際のユースケース

6.1 企業内データ共有

シナリオ:

社内の機密情報を、部署や役職などの属性に応じて共有する場合。

例として、「経営陣のみ」や「研究開発部かつ正社員」など、属性に基づいたアクセス制御が必要なシーン。

ABE のメリット:

暗号化時に設定されたポリシーにより、指定された属性を持つユーザのみがデータを復号できるため、情報漏洩リスクが低減されます。

6.2 クラウドストレージでのアクセス制御

シナリオ:

クラウド上に保存されたデータに対し、特定のユーザグループのみがアクセスできるよう制御する場合。

ABE のメリット:

データ暗号化と同時に、誰がアクセスできるかを属性やポリシーで制御できるため、第三者による不正アクセス防止に効果的です。

6.3 動的なユーザグループの管理

シナリオ:

メンバーの入れ替わりが頻繁なプロジェクト・チームなど、動的な組織内でのデータ共有。

ABE のメリット:

ユーザ属性が更新されると、システム全体で自動的にアクセス権が反映されるため、従来の個別鍵再発行の手間を省き、運用がスムーズに行えます。

6. 企業で想定される属性

6.1 部署・組織単位の属性

経営層:

- 役員（CEO、COO、CFO など）
- 経営企画室

管理部門:

- 人事部（採用、労務管理）
- 総務部
- 経理部・財務部

営業・マーケティング:

- 営業部（国内・海外）
- マーケティング部

技術部門:

- 開発部（ソフトウェア、システム開発）
- 研究開発部
- IT 運用部

カスタマーサポート:

- サポートセンター

生産・物流:

- 生産管理部
- 物流・倉庫管理部

法務・リスク管理:

- 法務部
- 内部監査部

6.2　役職・職務レベルの属性

役職レベル:

- 経営者・役員
- 部長／マネージャー
- 主任／リーダー
- 一般社員／スタッフ

職務:

- プロジェクトマネージャー
- シニアエンジニア／ジュニアエンジニア
- 営業担当
- カスタマーサポート担当

6.3　その他の属性

雇用形態:

- 正社員
- 契約社員／派遣

勤務地:

- 本社
- 支店／拠点

専門資格・スキル:

- IT 関連資格（例：CCNA、情報セキュリティスペシャリスト）
- 業界特有の資格

プロジェクト・チーム:

- 特定プロジェクト・チーム所属
- 複数部門横断型チーム

7. CP-ABE を用いた社内用途の具体例

7.1 CP-ABE の特徴

暗号化者が暗号文生成時に、どの属性のユーザが復号可能かを明示的に設定します。

暗号文自体にアクセス制御ポリシーが組み込まれるため、データの「受信者」を動的に決定可能です。

7.2 具体例: 経理部の財務データの保護

用途:

- 社内の機密財務データ（四半期決算報告書、内部監査資料など）の暗号化および配布

属性例:

- 部署属性:「経理部」
- 役職属性:「部長」または「シニアスタッフ」
- 雇用形態:「正社員」

ポリシー例 (CP-ABE):

- 暗号文生成時に、ポリシーとして

  ("経理部" AND ("部長" OR "シニアスタッフ") AND "正社員")

  を設定

  → 経理部に所属し、かつ部長またはシニアスタッフであり、正社員であるユーザのみが復号可能

シナリオ:

- 経理部門の担当者が、内部監査用に機密の財務資料を暗号化して社内ポータルにアップロード。
- 復号を希望するユーザは、所属部署・役職・雇用形態に合致している場合に限り、KGC（鍵発行機関）から取得した秘密鍵を用いて資料を復号できる。

8. KP-ABE を用いた社内用途の具体例

8.1 KP-ABE の特徴

ユーザに発行される秘密鍵にアクセス制御ポリシーが組み込まれます。

暗号化者は単にデータに関連する属性（例:「営業部」「正社員」）を暗号文に付加し、復号可能かどうかはユーザ側の秘密鍵ポリシーに依存します。

8.2 具体例: 営業データの配信

用途:

- 営業部が各顧客への販売戦略やキャンペーン情報などのデータを暗号化して社内共有する場合

属性例:

- 部署属性:「営業部」
- 雇用形態:「正社員」
- 勤務地属性（場合に応じて）:「本社」または「支店」

ポリシー例 (KP-ABE):

- 暗号化者は、暗号文に対して、
  {"営業部", "正社員", "勤務地:本社"}
  などの属性情報を付加
- 各ユーザには、KGC が発行する復号鍵に事前にポリシー（例:「属性集合に『営業部』が含まれ、かつ『正社員』である」）を組み込む
  → 秘密鍵ポリシーを満たすユーザのみが、該当属性を含む暗号文を復号可能

シナリオ:

- 営業部門の情報担当者が、営業成績や顧客情報などを暗号化して共有サーバにアップロード。
- ユーザは、所属部署が「営業部」で正社員であれば、KGC によって発行された秘密鍵により、暗号文内の属性が自分のポリシーと一致するため、復号が可能となる。
- 万が一、派遣社員などの属性が付与されたユーザは、秘密鍵のポリシーに合致しないため、復号はできず情報漏洩リスクを低減できる。