高機能暗号ライブラリにおける誤用リスクの分析

小寺 健太[1,*] 藤田 真浩[1] 川合 豊[1] 早坂 健一郎[1]
髙橋 碧斗[2] 金岡 晃[2]

概要：近年，属性ベース暗号や ID ベース暗号などの高機能暗号の研究が進展し，これらを実装したライブラリも公開されている．高機能暗号は RSA や AES などの従来暗号と比較して，暗号化されたデータに対する柔軟なアクセス制御や計算処理など，理論的に優れた性質を有する．しかし，これらの暗号方式の実用化において，理論上の安全性と実装上の安全性の間にはギャップが存在する．特に，高機能暗号の複雑なアルゴリズム構造や多様なパラメータ設定は，実装時に従来暗号では見られない新たなリスクを生む可能性がある．先行研究では，属性ベース暗号のライブラリ開発者を対象としたインタビュー調査により，ライブラリ開発時の考慮事項が明らかにされた．しかし，当該研究では実装上の問題点とその対策が混在して議論されており，暗号方式に固有の特性に起因する誤りの分析は十分に行われていない．本論文では，IPA によるソフトウェア開発上の誤り分類体系を活用し，既存研究で報告されている従来暗号ライブラリの誤用事例 57 件と，先行研究のインタビュー結果を再分析することで抽出した属性ベース暗号ライブラリの誤用事例 7 件を体系的に比較分析した．その結果，属性ベース暗号ライブラリにおいて「タイミング誤り」が新たに顕在化し，「データの誤り」の範囲が拡大することが明らかとなった．さらに，属性ベース暗号の誤用事例はすべて既存の IPA の誤り分類に対応可能であり，既存のソフトウェア品質管理手法を属性ベース暗号の持つ複雑さに適応させることで対応可能であることが示唆された．

キーワード：高機能暗号，暗号ライブラリ，誤用リスク

Analysis of Misuse Risks in Advanced Cryptographic Libraries

Kenta Kodera[1,*] Masahiro Fujita[1] Yutaka Kawai[1] Kenichiro Hayasaka[1]
Aoto Takahashi[2] Akira Kanaoka[2]

Abstract: In recent years, theoretical research on advanced cryptography such as attribute-based encryption and identity-based encryption has progressed, and libraries implementing these schemes have been made publicly available. Advanced cryptography possesses theoretically superior properties compared to conventional cryptography such as RSA and AES, including flexible access control and computational processing on encrypted data. However, in the practical implementation of these cryptography, there exists a gap between theoretical and implementational security. In particular, the complex algorithmic structures and diverse parameter configurations of advanced cryptographic libraries may introduce novel risks during implementation that are not observed in conventional cryptographic libraries. Prior research has revealed considerations during library development through interview surveys targeting developers of attribute-based encryption libraries. However, in that study, implementation issues and their countermeasures were discussed in a mixed manner, and insufficient analysis was conducted on errors arising from characteristics inherent to the cryptographic schemes. In this paper, we systematically compared and analyzed 57 cases of misuse in conventional cryptographic libraries reported in existing research and 7 cases of misuse in an attribute-based encryption library extracted through re-analysis of interview results from prior research, utilizing the IPA's error classification system for software development. The results revealed that "timing errors" newly emerged in the attribute-based encryption library, and the scope of "data errors" expanded. Furthermore, all misuse cases of the attribute-based encryption library could be addressed by existing IPA error classifications, suggesting that existing software quality management methods can be adapted to handle the complexity inherent in attribute-based encryption.

Keywords: Advanced cryptography, Cryptographic libraries, Misuse risks

1. はじめに

暗号技術は現代のデジタル社会において情報セキュリティの根幹を成す重要な技術である．インターネット通信や電子商取引，IoT デバイスなど，あらゆる分野で暗号技術が活用されており，その重要性は年々高まっている．しかし，暗号アルゴリズム自体が数学的に安全であっても，その実装や利用方法において誤用が発生すると，暗号アルゴリズムを活用したシステム全体のセキュリティが脅かされる可能性がある．

実際，暗号ライブラリの誤用が多くの箇所で発生していることが知られている．例えば Egele らの研究では，Google Play で公開された Android アプリケーションを調査した結果，暗号 API を利用するアプリケーションのうち 88%が少なくとも 1 つ以上の誤用や設定の間違いを含んでいたことが報告されている[1]．また，Meng らによる研究では，Stack

1 三菱電機株式会社 情報技術総合研究所
Mitsubishi Electric Corporation, Information Technology R&D Center
2 東邦大学
Toho University

* Kodera.Kenta@df.MitsubishiElectric.co.jp

Overflow のフォーラムにおける回答のうち，承認された回答が示すコード中にさえ複数の脆弱性が含まれていたことが報告されている[2].

こうした事実に対応し，RSA や AES などの従来暗号ライブラリにおけるユーザビリティ評価や誤用防止策の研究が行われている[3][4][5][6][7][8][9]. 暗号ライブラリの開発者または利用者を対象としたインタビュー調査や実装を伴う実験の結果の分析や，暗号ライブラリの誤用を検知するツールの開発など，より安全な暗号ライブラリの利活用に向けた取り組みがなされている.

従来暗号ライブラリにおいても誤用の問題は深刻であるが，近年注目を集めている属性ベース暗号，準同型暗号などの高機能暗号のライブラリでは，その複雑さから多様で深刻な誤用リスクが発生する可能性が懸念される．高機能暗号は，データを暗号化したまま処理できるなど，従来の暗号と比較して様々な付加価値を有する[10]. 例えば，属性ベース暗号では暗号文の受信者が持つ肩書きなどの属性に応じて，暗号文の復号可否を制御できる[11].

暗号ライブラリの誤用リスクを効果的に減らすためには，まず誤用の全体像を正確に把握し，各誤用の根本原因を特定した上で，適切な対策を講じることが重要である．従来暗号ライブラリにおけるユーザビリティの向上が図られている一方で，現状では従来暗号ライブラリと高機能暗号ライブラリにおける誤用リスクの特徴や範囲の違いについて，体系的な分析が行われていない．著者らの先行研究では，属性ベース暗号のライブラリ開発者に対してインタビュー調査を実施し，当該暗号ライブラリを用いて開発を行った際に，ライブラリ利用者のためにどのような配慮をしてきたかを明らかにした[12]. しかし，当該研究では実装上の問題点とその対策が混在して議論されており，暗号方式に固有の特性に起因する誤用リスクの分析は十分に行われていない．また，属性ベース暗号のライブラリを用いた実装実験結果の分析によって，ライブラリ利用上の困難さは API そのものの複雑さだけでなく，設計上の透明性不足やドキュメント構成，エラー対応支援の欠如に起因していることを考察した[13]．こうした要素は高機能暗号に特有のものではなく，従来暗号ライブラリを含めた暗号技術に共通する要素である可能性も考えられるが，当該研究では十分な議論がなされていない.

このような背景から，本研究では，IPA の「組込みソフトウェア開発における品質向上の勧め［バグ管理手法編］」の分類体系[14]を活用し，既存研究で報告されている従来暗号ライブラリの誤用事例と，文献[12]で実施した属性ベース暗号ライブラリの開発者へのインタビュー調査結果に基づく誤用事例を体系的に分析することで，両ライブラリが有する誤用リスクの特徴を明らかにする．すなわち本研究の目的は，以下のリサーチクエスチョン（RQ）を解明することである.

RQ： 従来暗号ライブラリと属性ベース暗号ライブラリが有する誤用リスクの特徴に差はあるか？

2. 関連研究

暗号ライブラリの誤用は，暗号技術の実用化が進むにつれて注目されるようになった研究分野である. これまでに，ライブラリの実態調査や静的解析ツール，ユーザー実験など，様々なアプローチによって暗号ライブラリの誤用事例が調査されている.

Egele らは，Google Play で公開された Android アプリケーションを調査し，暗号 API を利用するアプリケーションのうち 88%が少なくとも 1 つ以上の誤用や設定の誤りを含むことを示した[1]．特に，ECB モードの利用，鍵の固定，初期化ベクトル（IV）の固定，PBE のイテレーション数不足，乱数シードの固定，ソルトの固定などの誤用パターンを特定した.

Acar らは，Python の共通鍵暗号や公開鍵暗号に関する主要な暗号ライブラリを対象として，ライブラリ利用者に対するユーザー実験を行った[3]. RSA での不適切な鍵長の使用，鍵生成時の不適切なランダム性を持つソースの使用，暗号化キーの非暗号化状態での保存などの誤用パターンを特定し，安全な暗号ライブラリのためにはライブラリのシンプルさだけでなく，整備されたドキュメントやサンプルコード，鍵管理のための補助機能が重要であることを示した.

Rahaman らは，Java における暗号ライブラリの誤用を検出するツールである CryptoGuard を提案した[8]．開発に伴い，予測可能な暗号鍵の使用，全てのホストを受け入れるカスタムホスト名検証器，暗号学的に安全でない擬似乱数生成器の使用など，多様な誤用パターンを体系化した.

Chatzikonstantinou らは，弱い暗号，弱い実装，弱い鍵，弱い暗号学的パラメータの 4 つのカテゴリに分類した誤用パターンを定義し，Android アプリケーションにおける暗号技術の使用状況を評価した[9]．例えば，弱い暗号化アルゴリズムの使用，ハードコードされた暗号化キーの使用，予測可能な PRNG シードの使用などの問題を詳細に分析し，AES や RSA の利用の推奨を含む，アプリ開発者向けのベストプラクティスを提案した.

高機能暗号は，データを暗号化したまま処理できるなど，従来の暗号と比較して様々な付加価値を有する[10]．例えば属性ベース暗号や ID ベース暗号，グループ署名，検索可能暗号などの暗号方式が高機能暗号として注目されている[11][15][16][17]．属性ベース暗号は高機能暗号の一つであり，暗号文の受信者が持つ肩書きなどの属性に応じて，暗号文の復号可否を制御できる公開鍵暗号方式である．属性ベース暗号のライブラリとしては，復号のための制御構造を暗号文に埋め込む方式や，秘密鍵に埋め込む方式などのライブラリが公開されている[18][19][20].

表 1　暗号ライブラリ誤用の分類基準

IPA 分類名	分類の基準
データの誤り	本来使用すべき値を使用できていないことによる誤用
ロジックの誤り	通常であれば実装すべき処理が不足していることによる誤用
インターフェースの誤り	本来使用すべき関数を使用できていないことによる誤用
タイミングの誤り	関数の実行順序に関する誤用
リソースの誤り	値の初期化や領域の解放処理に関する誤用
エラーチェックの誤り	通常であれば実装すべきエラーチェック処理が存在しないことによる誤用
機能の欠如	本来実行すべき機能全体が抜けていることによる誤用
機能の実装誤り	不要な処理を実行することによる誤用

著者らの先行研究[12]では，属性ベース暗号のライブラリ開発者に対してインタビュー調査を実施し，当該暗号ライブラリを用いて開発を行った際に，ライブラリ利用者のためにどのような配慮をしてきたかを明らかにした．シンプルな設計や安全なデフォルト設定，充実したドキュメント，分かりやすいエラーメッセージの実現が従来暗号ライブラリと同様に有効であることに加え，属性値などの特有の概念や，複雑なアルゴリズムへの対処が重要であることを示した．

IPA の「組込みソフトウェア開発における品質向上の勧め［バグ管理手法編］」は，ソフトウェア開発におけるバグの体系的な分類と管理手法を示すものである[14]．同ガイドでは，ソフトウェア開発時の誤りを「データの誤り」「ロジックの誤り」「インターフェースの誤り」などの詳細なカテゴリに分類し，それぞれの誤りの根本原因を特定するための基盤を提供している．

このように，従来暗号ライブラリの誤用リスクについては多数の実態調査や対策手法が提案されている一方で，高機能暗号ライブラリについては十分な研究がなされていない．著者らの先行研究において属性ベース暗号ライブラリ開発時の考慮事項や利用時の困難さの要因が特定されているものの，暗号方式に固有の特性に起因する誤用リスクの分析は行われていない．したがって，本研究では IPA のソフトウェア開発における誤りの分類体系を利用し，従来暗号ライブラリと属性ベース暗号ライブラリの誤用リスクを体系的に比較分析することで，属性ベース暗号ライブラリが有する誤用リスクの特徴を明らかにする．

3. 研究手法

本研究では，IPA のバグ分類体系を基準として，以下の 2 つのデータソースから誤用事例を抽出し，体系的な分析を行う．

(1) 従来暗号ライブラリの誤用リスクに関する既存研究

(2) 文献[12]における属性ベース暗号ライブラリ開発者へのインタビュー調査結果

3.1 暗号ライブラリ誤用の分類基準

本研究では，暗号ライブラリの誤用事例を IPA の分類体系に対応付けるため，表 1 の分類基準を設定した．

3.2 従来暗号ライブラリの誤用事例抽出

Egele ら，Acar ら，Rahaman ら，Chatzikonstantinou らの既存研究から，従来暗号ライブラリの誤用事例を抽出した．各誤用事例について，表 1 の分類基準に基づいて IPA の分類体系に対応付けを行った．

3.3 属性ベース暗号ライブラリの誤用事例抽出

文献[12]で実施した属性ベース暗号ライブラリ開発者 2 名へのインタビューデータから，誤用事例に言及している箇所を特定し，開発者が懸念していた誤用パターンや実際に発生した問題を抽出した．次に既存研究と同様に，表 1 の分類基準に基づいて IPA の分類体系に対応付けた．

3.4 誤用事例の分類

抽出した誤用事例から IPA 分類への対応付けの妥当性を評価するため，暗号学とソフトウェア工学の知識を持つ 2 名の著者が独立して分類作業を実施し，その後，両者で議論を行って分類結果の妥当性を確認した．分類が困難な事例については，IPA の誤り分類の定義と本研究で設定した分類基準に立ち返って検討を行い，最終的な分類を決定した．

4. 結果

4.1 従来暗号ライブラリの誤用事例

従来暗号ライブラリの誤用事例の調査対象とした Egele ら，Acar ら，Rahaman ら，Chatzikonstantinou らの既存研究から，それぞれ 6 件，11 件，16 件，24 件の事例を抽出した．合計 57 件の誤用事例を IPA 分類に対応付けた結果を表 2 に示す．

4.2 属性ベース暗号ライブラリの誤用事例

属性ベース暗号ライブラリ開発者のインタビュー結果から抽出した誤用事例 7 件を IPA 分類に対応付けた結果を表 3 に示す．

表 2　従来暗号ライブラリの誤用と IPA 分類の対応

IPA 分類名	主な誤用事例
データの誤り	ECB モードの利用，鍵の固定，IV の固定，予測可能な暗号鍵，静的ソルト，短い鍵の使用
ロジックの誤り	PBKDF2 でのイテレーション回数不足
インターフェースの誤り	HTTP の使用，暗号学的に安全でない PRNG，64 ビットブロック暗号の使用
タイミングの誤り	該当なし
リソースの誤り	暗号化・復号後の内部バッファの未処理
エラーチェックの誤り	全てのホストを受け入れるカスタムホスト名検証器
機能の欠如	暗号化の使用が観察されない，秘密鍵の非暗号化状態での保存
機能の実装誤り	該当なし

表 3 属性ベース暗号ライブラリの誤用と IPA 分類の対応

誤用事例	IPA 分類名
誤った楕円曲線パラメータを入力	データの誤り
乱数のシードに常にゼロを入力	データの誤り
属性値の上限値を超過	データの誤り
同一属性カテゴリの値を複数種類入力	データの誤り
AES のカウンター値に常に 1 を入力	データの誤り
役割と異なる関数を実行	タイミングの誤り
属性更新後の古い秘密鍵を入力	タイミングの誤り

5. 考察

5.1 RQ への回答：従来暗号ライブラリと属性ベース暗号ライブラリが有する誤用リスクの特徴の差異

本研究の分析結果から，従来暗号ライブラリと属性ベース暗号ライブラリにおける誤用リスクの特徴について，以下の知見が得られた．

5.1.1 タイミング誤りの顕在化

属性ベース暗号ライブラリにおいてタイミング誤りが新たに出現した．タイミング誤りとは，関数の実行順序に関する誤用である．従来暗号ライブラリでは，対象とした誤用事例において当該分類に分類されるものは存在しなかった一方で，属性ベース暗号ライブラリでは複数の誤用事例が分類された．

具体的には，「役割と異なる関数を実行」と「属性更新後の古い秘密鍵を入力」という 2 つの誤用事例である．前者は，属性ベース暗号における複数の役割（暗号化者，復号者，鍵生成局）において，自身の役割に対応しない関数を実行してしまう誤用である．後者は，利用者の属性が変更された際に，新しい属性に対応する秘密鍵ではなく，古い属性に対応する秘密鍵を使用してしまう誤用である．これらのタイミング誤りの事例は，属性ベース暗号が持つ複雑な処理シーケンスと状態管理に起因すると考えられる．従来の暗号化・復号という単純な 2 段階処理から，鍵生成局と呼ばれる第三者によってマスター秘密鍵とマスター公開鍵が生成され，ユーザーごとに秘密鍵を生成・配布したのちに，暗号化・復号を行うという多段階の処理が必要となる．さらにユーザーの持つ属性の更新に伴って秘密鍵を更新する場合は，適切な鍵の利用が必須となる．このように，処理の順序や状態の管理が複雑化し，新たな誤用の原因となっている．

5.1.2 データ誤りの範囲拡大

データの誤りについて，従来暗号ライブラリでも最も多くの事例が確認されており，属性ベース暗号ライブラリにおいても同様に多数の事例が確認された．これは，暗号ライブラリにおいてパラメータ設定の適切さが重要であり，誤用リスクの主要因であることを示している．

属性ベース暗号におけるデータの誤りには，乱数のシードや AES のパラメータ設定など従来暗号ライブラリにおいても確認されていた事例が確認された．さらに，属性値の上限や同一属性カテゴリに関する制限など，属性ベース暗号特有のデータ構造に関連した新たな誤用事例が確認された．属性ベース暗号では，設定すべきパラメータの種類と複雑性が増加しているため，データの誤りに関わる誤用リスクがより顕著に現れており，従来暗号と比較して誤用範囲が拡大している．

5.1.3 既存の誤用分類の適用可能性

属性ベース暗号における誤用事例は，すべて IPA の既存のバグ分類を使って分類可能であった．属性ベース暗号特有の複雑性により新たな誤用事例が出現したものの，既存のソフトウェア開発における誤りの枠組みで捉えることができる．これは，属性ベース暗号の誤用リスクが，本質的には従来のソフトウェア開発における誤りの延長線上にあることを示している．

5.2 属性ベース暗号の誤用リスクの本質的特徴

以上の結果から，属性ベース暗号ライブラリの誤用リスクが，根本的に新しい種類の問題ではなく，従来暗号ライブラリで見られる事例の複雑化と，特定の分類の顕在化に

よって生じることが示唆される．従来暗号ライブラリにおいて確認されなかったタイミング誤りの出現は，属性ベース暗号特有の処理と役割という新しい要素によるものであるが，ソフトウェア開発における誤りの分類体系で捉えることができる．

したがって，属性ベース暗号ライブラリの誤用リスク対策において，全く新しいアプローチを開発するのではなく，既存のソフトウェア品質管理手法を属性ベース暗号の複雑さに適応させることで対応可能であることが示唆される．

5.3 機能の実装誤りの不在に関する考察

本研究の分析において，興味深い知見として，従来暗号ライブラリと属性ベース暗号ライブラリの両方において，「機能の実装誤り」に分類される誤用事例が確認されなかった点が挙げられる．IPA の分類における「機能の実装誤り」とは，不要な処理を実行してしまう誤用を指しており，例えばデバッグ用の print 文の消し忘れによる鍵の漏洩が該当すると考えられる．

この結果には複数の解釈が可能である．第一に，既存研究が主にデバッグコードの残存のような一般の実装レベルの問題を対象としていない可能性が考えられる．第二に，暗号ライブラリの開発においては，セキュリティ上の理由から開発プロセスでのコードレビューが厳格に行われており，不要な処理の混入が他の分野と比較して少ない可能性がある．

ただし，この知見については慎重な解釈が必要である．本研究の調査範囲は限定的であり，より大規模で包括的な調査を実施することで，機能の実装誤りに該当する事例が発見される可能性は十分にある．特に，実際の開発現場における詳細な事例収集や，より多様な暗号ライブラリを対象とした調査により，この分類に該当する誤用リスクが明らかになる可能性がある．

5.4 研究の限界

5.4.1 データソース数の限界

従来暗号ライブラリと属性ベース暗号ライブラリの事例数に大きな差があるため，統計的な比較の信頼性に限界がある．特に，属性ベース暗号ライブラリで確認されなかった誤用分類について，実際に誤用リスクが発生しないのか，単に事例数が少ないために観察されなかったのかを判断するために，事例数を増やす必要がある．

本研究では属性ベース暗号ライブラリ開発者 2 名のインタビューデータを使用したが，これは属性ベース暗号ライブラリ全体の誤用リスクの傾向を代表するには不十分である．より多くの開発者からのデータ収集によって，結果の一般性やライブラリ全体の傾向について新たな知見が得られる可能性がある．

また本研究で調査した既存研究やインタビューの範囲では「機能の実装誤り」に該当する事例が確認されなかったが，実際に当該分類の誤用が存在しないことを意味するかは不明である．より包括的な調査により，見落とされている誤用の分類が発見される可能性がある．

5.4.2 組織におけるバイアス

インタビュー対象者 2 名が同一組織に属していることにより，特定の開発文化や方針に偏った誤用リスク認識が得られている可能性がある．異なる組織，異なる地域，異なる開発アプローチを持つ開発者からのデータ収集により，より一般化可能な誤用リスクの知見獲得に繋がる可能性がある．

5.4.3 暗号方式の限定性

本研究では属性ベース暗号のみを対象としたため，他の高機能暗号における誤用リスクが十分に捉えられていない．複数の高機能暗号を対象とした比較研究によって，各暗号方式特有の複雑さが誤用リスクに与える影響を調査する必要がある．

6. おわりに

本研究では，IPA のソフトウェア開発におけるバグ分類体系を活用し，従来暗号ライブラリと属性ベース暗号ライブラリの誤用事例を体系的に比較分析した．その結果，属性ベース暗号ライブラリにおいて「タイミング誤り」が新たに顕在化することが明らかとなった．これは，属性ベース暗号が持つ多段階処理と複数の役割による複雑性に起因することが示唆された．また「データの誤り」については，従来暗号ライブラリで確認されていた基本的なパラメータ設定の誤りが共通して現れる一方で，属性値が持つ制限の誤りなど，属性ベース暗号に特有のデータ構造における誤りが確認され，従来暗号と比較して誤用範囲が拡大することが示された．特筆すべき知見として，属性ベース暗号の誤用事例はすべて IPA の誤り分類を用いて分類可能であり，根本的に新しい種類の問題は確認されなかった．このことから，属性ベース暗号ライブラリの誤用リスク対策において，既存のソフトウェア品質管理手法を属性ベース暗号が有する複雑さに適応させることで対応可能であることが示唆された．

今後の研究として，より多くの属性ベース暗号ライブラリの開発者を対象とした大規模な実証研究を行うことで，得られた知見の一般性について検証するとともに，属性ベース暗号以外の高機能暗号を対象とした誤用リスクの分析を行うことで，高機能暗号全体における誤用リスクの体系化が必要である．

参考文献

[1] Egele, M., Brumley, D., Fratantonio, Y., Kruegel, C.: An empirical study of cryptographic misuse in android applications, Proc. 2013 ACM SIGSAC Conference on Computer & Communications Security, No.12, pp.73-84 (2013).

[2] Meng, N., Nagy, S., Yao, D., Zhuang, W., Argoty, G.A.: Secure coding practices in Java: challenges and vulnerabilities, Proc. 40th International Conference on Software Engineering, No.12, pp.372-

383 (2018).
[3] Acar, Y., Backes, M., Fahl, S., Garfinkel, S., Kim, D., Mazurek, M.L., Stransky, C.: Comparing the Usability of Cryptographic APIs, Proc. IEEE Symposium on Security and Privacy (SP), pp.154-171 (2017).
[4] Ukrop, M. and Matyas, V.: Why Johnny the Developer Can't Work with Public Key Certificates: An Experimental Study of OpenSSL Usability, Proc. Topics in Cryptology -- CT-RSA 2018, pp.45-64 (2018).
[5] Mindermann, K., Keck, P., Wagner, S.: How Usable are Rust Cryptography APIs?, Proc. IEEE International Conference on Software Quality, Reliability and Security (QRS'18), pp.143-154 (2018).
[6] Votipka, D., Fulton, K.R., Parker, J., Hou, M., Mazurek, M.L., Hicks, M.: Understanding security mistakes developers make: qualitative analysis from build it, break it, fix it, Proc. 29th USENIX Security Symposium, pp.109–126 (2020).
[7] Fischer, K., Trummová, I., Gajland, P., Acar, Y., Fahl, S., Sasse, A.: The Challenges of Bringing Cryptography from Research Papers to Products: Results from an Interview Study with Experts, Proc. 33rd USENIX Security Symposium, pp.7213-7230 (2024).
[8] Rahaman, S., Xiao, Y., Afrose, S., Shaon, F., Tian, K., Frantz, M., Kantarcioglu, M., Dang, T., Pratap, V., Sunshine, J.: CryptoGuard: High Precision Detection of Cryptographic Vulnerabilities in Massive-sized Java Projects, Proc. 2019 ACM SIGSAC Conference on Computer and Communications Security, pp.2455-2472 (2019).
[9] Chatzikonstantinou, A., Ntantogian, C., Karopoulos, G., Xenakis, C.: Evaluation of cryptography usage in android applications, Proc. 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), pp.83-90 (2016).
[10] CRYPTREC 暗号技術調査ワーキンググループ（高機能暗号）: CRYPTREC 暗号技術ガイドライン（高機能暗号）(online), available from <https://www.cryptrec.go.jp/report/cryptrec-gl-2005-2022.pdf> (accessed 2025-08-15).
[11] Sahai, A. and Waters, B.: Fuzzy Identity-Based Encryption, Proc. EUROCRYPT 2005, pp.457-473 (2005).
[12] 小寺健太, 藤田真浩, 堀込光, 高橋碧斗, 金岡晃: 多分その方がみんな幸せになるっていう感じですね：開発者インタビューに基づく高機能暗号ライブラリ開発時の考慮事項の抽出, コンピュータセキュリティシンポジウム 2024 論文集, pp.523-530 (2024).
[13] 髙橋碧斗, 金岡晃, 早坂健一郎, 川合豊, 小寺健太, 藤田真浩: OpenABE を用いた開発者向け実験による高機能暗号のユーザビリティと誤用リスクの評価, セキュリティ心理学とトラスト（SPT）, 2025-SPT-60 巻, 64 号, pp.1-8 (2025)
[14] IPA 独立行政法人情報処理推進機構: 組込みソフトウェア開発における品質向上の勧め［バグ管理手法編］(online), available from <https://www.ipa.go.jp/archive/publish/qv6pgp000000010b6-att/000027629.pdf> (accessed 2025-08-15).
[15] Shamir, A.: Identity-Based Cryptosystems and Signature Schemes, Proc. CRYPTO 1984, pp.47-53 (1984).
[16] Chaum, D. and Heyst, E.V.: Group Signatures, Proc. EUROCRYPT 1991, pp.257-265 (1991).
[17] Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public Key Encryption with Keyword Search, Proc. EUROCRYPT 2004, pp.506-522 (2004).
[18] OpenABE(online), available from <https://github.com/zeutro/openabe/> (accessed 2025-08-15).
[19] Ciphertext-Policy Attribute-Based Encryption(online), available from <https://acsc.cs.utexas.edu/cpabe/> (accessed 2025-08-15).
[20] Charm: A Framework for Rapidly Prototyping Cryptosystems (online), available from <https://github.com/JHUISI/charm> (accessed 2025-08-15).