小口研究室　研究紹介　(2025年度)
(お茶の水女子大学理学部情報科学科)

SQLiteを用いた強制アクセスポリシ管理手法の設計と実装
（研究担当:喜多　陽花）

研究背景

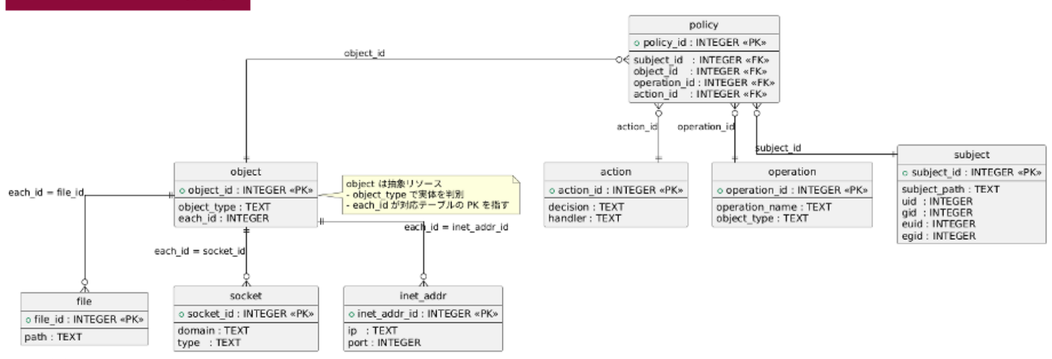
- 不正アクセスは深刻な脅威として注目されている
- 前回までの研究で、Unixドメインソケットを用いて強制アクセス制御を行うシステムuMACを設計・実装(認可機構は未実装)
- アクセスポリシとは
  - 誰がどのリソースにアクセスできるかを定めたルール
  - 機密性が求められる

研究背景

- uMACの認可部分の実装
- SQLiteを用いて、データベースでアクセスポリシを管理
- SQLCipherを用いて、データベース全体を暗号化
  - 認可ポリシの機密性を確保する

✓ユーザ空間で完結
✓アクセスポリシの追加・削除・更新をデータ操作として反映
✓ポリシ更新時に再起動を必要としない動的な管理

データベースの設計



設計方針
- 認可判定に必要な情報を要素に分解し，正規化されたデータとして管理することで冗長性を排除する．
- 将来的な条件追加や拡張を考慮し，既存スキーマへの影響を最小限に抑えた形でポリシ表現を拡張可能とする．
- 認可判定をSQL による問い合わせとして記述可能なデータモデルとする．
- 認可判定は単一のSQL クエリで表現し，実行時に与える条件値のみを変更して判定を行うことを可能とする．

認可処理とSQLクエリ

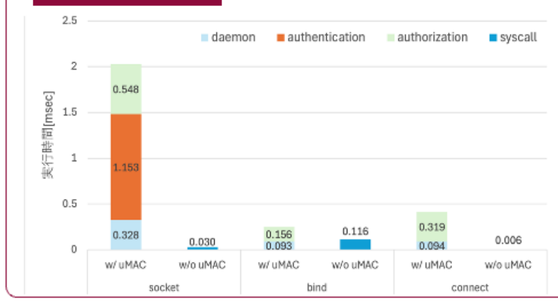
- 認可処理では，subject，object，operation が一致するポリシをデータベースから探索
- 対応するaction に基づいて操作の可否を決定
- 処理を単一のSQL クエリとして記述
  - 認可判定のためのテンプレートを1 つ用意
  - アクセス要求に含まれるsubject/object/operation の値をパラメータとして埋め込んで実行
- ポリシごとにクエリ自体を変更する必要はない．

```
WITH objects AS (
SELECT *
  FROM object o
  LEFT JOIN socket    s ON o.each_id = s.socket_id    AND o.object_type="socket"
  LEFT JOIN inet_addr i ON o.each_id = i.inet_addr_id AND o.object_type="inet_addr"
  LEFT JOIN file      f ON o.each_id = f.file_id      AND o.object_type="file"
)
SELECT * FROM action
WHERE action_id = (
  SELECT action_id
  FROM policy
  WHERE
subject_id = (
      SELECT subject_id FROM subject
      WHERE (subject_path, uid, gid, euid, egid)
        = ("/root/MAC/main/server_application", 0, 0, 0, 0))
  AND object_id = (
      SELECT object_id FROM objects
      WHERE (ip, port) = ("192.168.10.100", 3000))
AND operation_id = (
      SELECT operation_id FROM operation
      WHERE (name, object_type) = ("connect", "inet_addr")));
```

管理者側の操作

- 管理者は，アクセスポリシ全体をYAML 形式のポリシ記述ファイルとして記述
- ポリシ記述ファイルには，有効なすべてのアクセスポリシが明示的に記述され、アクセスポリシデータベースの内容と一致するように生成．
- ポリシ記述ファイルは，専用の変換ツールによって読み取られ，アクセスポリシデータベースへと変換
- アクセスポリシデータベースは，システムの実行時に参照されるものであり，管理者は直接編集しない
- 実行時の認可判定処理を効率化しつつ，ポリシ管理の複雑さを管理者から分離

評価



- 認可判定時のデータベース問い合わせが一定のオーバヘッドを発生させている
- socket においては認証処理が最大のボトルネック
- インターネットアクセスを行うアプリケーションにおいては，通信に伴うネットワーク遅延が支配的
- システムコール単体における認証・認可処理のオーバヘッドは，実運用上の性能に与える影響は限定的

- 測定はDocker 上で単一マシン内
- 評価対象とするシステムコール：socket，bind，connect
- uMAC で認証・認可を行わずに実行した場合，uMAC を用いずそのまま実行した場合の2 通りの処理時間を測定
- 処理時間は，各操作を10 回実行し，その平均値を算出

IoTデータのプライバシ保護基盤構築に向けた暗号変換手法の評価
（研究担当:関　萌乃）

研究背景

- IoTデバイスの急速な普及
- 収集データをパブリッククラウドに蓄積・解析
  - 個人に関する情報を含む → **適切なプライバシ保護**
  - データ収集者における漏洩・不正利用のリスク → **暗号化**

▶ **データの収集・蓄積・解析を安全に行える基盤構築**

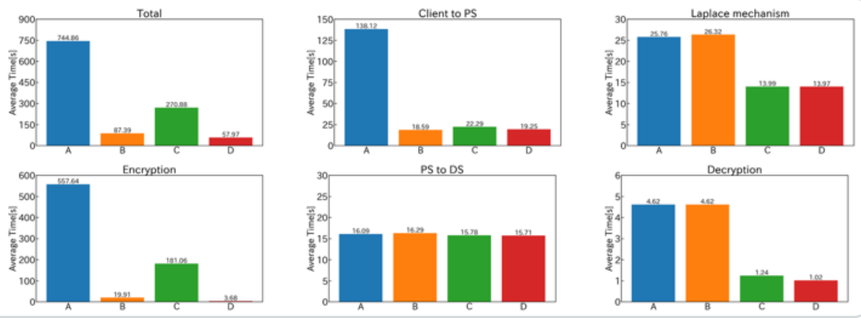
①事前実験

- **準同型暗号のみ**を用いたシステム
  - 暗号変換は含まない
  - LDPのラプラスメカニズムを利用
  - 準同型暗号ライブラリ：Microsoft SEAL
- クライアントのマシン，並列化の有無が異なる4つのパターンを実装・評価
  - 並列化：OpenMP
- サーバと比較してスマートフォンの処理速度は**非常に遅い**
  - 暗号化，クライアント→パブリックサーバのデータ転送で顕著

→ **並列化により短縮**

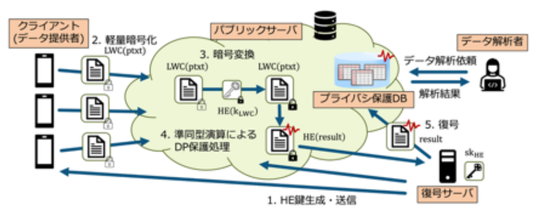
		クライアント	
		スマートフォン	サーバ
並列化	なし	パターンA	パターンB
	あり	パターンC	パターンD

※Laplace mechanismの内訳
- パラメータ送信
- ノイズ生成・暗号化
- ノイズの転送（ノイズサーバ→パブリックサーバ）
- 暗号文への準同型加算



システムデザイン

- IoTデバイスではデータを軽量暗号で暗号化
- パブリックサーバで準同型暗号に変換
  - 準同型暗号上で軽量暗号の復号操作
- 準同型演算を用いて差分プライバシ(DP)保護処理
  - 必要に応じて繰り返し暗号化データにアクセス

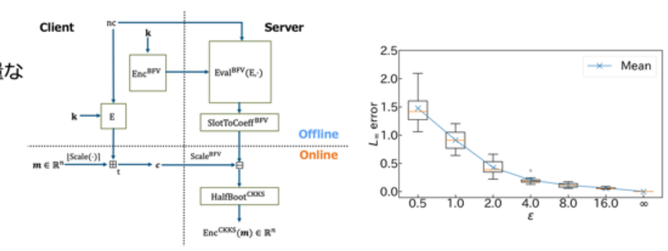


- データ収集者を信頼する必要なし
- 反復的な処理は計算資源の豊富なパブリックサーバで実行
  - **クライアントへの負荷**を削減
  - 対話可能性の高いモデルと同等の**高精度な解析**

②暗号変換の導入

HEFC(HE-Friendly Cipher)
- LHEやFHEに組み合わせる目的で設計された軽量な共通鍵暗号

RtFフレームワーク[1]
- 実数を扱えるCKKS方式の準同型暗号に対応した暗号変換のフレームワーク
- 剰余計算を用いたストリーム暗号を利用
  - HERA[1], Rubato[2]
- 内部処理においてBFV方式を経由



実験
- Rubatoを使用したRtFフレームワークで暗号変換後，準同型演算でラプラスメカニズムを適用し復号
  - 準同型暗号ライブラリ：Lattigo
- 元データとの誤差
  - L_∞誤差：各属性の推定平均値と真の平均値の誤差の最大値

$$||\mu' - \mu||_\infty = \max_i |\mu'_i - \mu_i|$$

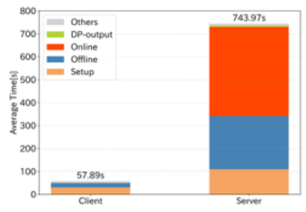
  - L_∞誤差・四分位範囲はϵの値が大きいほど減少傾向

→ 著者らの先行研究[3]とほぼ同様で，暗号変換による誤差の影響は十分小さい
- クライアント
  - 全体の実行時間は1レコードあたり約0.012秒
  - オンラインフェーズは約0.042秒

→ 事前計算でより高速にデータを処理可能
- サーバ
  - オンラインフェーズが全体の半分近く

→ 計算資源を増やすことで短縮可能
  - DPの処理時間はごくわずか

→ **実用化が見込める性能**



	Client	Server
Setup	30.67s	109.48s
Offline	18.39s	231.67s
Online	0.042s	388.61s
DP-output	-	5.14s
others	8.79s	9.06s
total	57.89s	743.97s

HE Key Generation: 149.71s

まとめ

- 提案システムの実現に向けて，①ベースラインおよび②暗号変換を導入したシステムの性能を評価

今後の課題

- クライアントとしてIoTデバイスを用いた実装・評価
- より解析結果の誤差が小さいLDPメカニズムの調査・検討

[1] CHO, Jihoon, et al. Transciphering framework for approximate homomorphic encryption (full version). Cryptology ePrint Archive, 2020.
[2] HA, Jincheol, et al. Rubato: Noisy ciphers for approximate homomorphic encryption (full version). Cryptology ePrint Archive, 2022.
[3] 関萌乃, 山口実靖, 神山剛, 小口正人. IoT データ活用基盤構築に向けた準同型演算による差分プライバシ保護処理の検討. マルチメディア，分散，協調とモバイル(DICOMO2025) シンポジウム

Label Differential Privacyにおける予期せぬ漏洩の監査
（研究担当:関口　ひなた）

研究背景

既存技術

Label Differential Privacy; Label DP とは，特徴量とラベルからなるデータのうち**ラベルのみを加工する**プライバシ保護技術

➢ *ラベルのみを匿名化するため，従来の匿名化手法よりも**精度が高い***

課題

Label DP は高い有用性をもつ一方で，**特徴量とラベルの間の相関**に起因するラベル情報の漏洩を十分に抑制できない

必要な視点
理論保証と実際の漏洩の差を定量化したい

提案手法

理論指標

"予期せぬ漏洩"の評価指標

$$\Delta\varepsilon = \varepsilon_{\mathrm{CF}} - \varepsilon_{\mathrm{LabelDP}}$$

➢ *Label DP-CF を監査基準とみなし，通常の Label DP 適用時にどれだけ追加の漏洩が生じるかを測る*

提案の考え方

ブラックボックスなメカニズムに対しても，尤度比検定（LRT）に基づく監査で**実質的なプライバシ損失**を評価する．

Label DP 後の相関のあるデータ → LRT により TPR, FPR を集計 → DP の定義から ε を計算

評価

手法

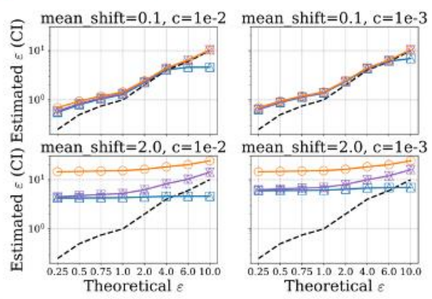
完全 LRT 監査	**理論的に最強** / c-power により監査可能な ε に制限がある
理論分解監査	理論分解定理を利用 / **c-power 制約に強い**

理論分解定理

$$\varepsilon_{CF} = ln\, R_{max} + \varepsilon_{RR}$$

➢ **相関由来の項 と Label DP の設計 ε** に分解できることを証明

実験結果



実験で分かったこと
- ✓ 相関が強いほど $\Delta\varepsilon$ が増大し，設計上の Label DP 保証よりも大きな漏洩を観測
- ✓ 高い ε や強相関の条件では，理論分解監査の方が安定して使いやすい

今後に向けた示唆
予期せぬ漏洩を防ぐメカニズムの構築が必要

https://www.is.ocha.ac.jp/~oguchi_lab/　　oguchi@is.ocha.ac.jp