6Y-07

IoTデバイスにおける共通鍵暗号と完全準同型暗号を組み合わせたセンサデータ暗号化の高速化

松本　茉倫†　　小口　正人†

†お茶の水女子大学

1. はじめに

スマートフォンを始めとするIoTデバイスの普及によって，取得したデータをクラウドサービス上で統計分析し，分析結果を活用することが期待されている．センサデータの中には，秘匿性が高いデータが存在しており，必ずしも安全とは言えないクラウドサービス上では個人情報を保護する必要がある．そこで，暗号文同士の加算・乗算が可能な完全準同型暗号(以下 FHE:Fully Homomorphic Encryption)が注目されている．しかし，公開鍵暗号であるFHEによる暗号化は共通鍵暗号に比べて処理に時間がかかり，暗号文サイズが大きいことが計算能力の低いIoTデバイス上での実装の課題となっている．先行研究[1]では，現在主流な共通鍵暗号であるAESとFHEを組み合わせることで，暗号化の高速化と通信量削減を提案している．先行研究[2]では，実際にLaptopを用いてAESとFHEを組み合わせた暗号の実装・評価を行った[3]．本研究では，スマートフォンのようなIoTデバイスにおいて，AESとFHEを組み合わせることでFHEによる暗号化の高速化と通信量削減を提案する．

2. 先行研究

Gentryら(2015)は提案手法-概要の(4)の処理にあたる，FHEによるAESの復号処理の実装と評価を行った[2]．AESを評価に使用した理由としては，広く使われている暗号であり，並列処理や最適化しやすい構造であるためとしている．FHEのライブラリであるHElibを用いて，2880Bの平文をAESで暗号化し，FHEで暗号化された(10ラウンド+1)×128bitのAESの鍵で復号する設計になっている．実験に使用したLaptopの性能を表1に示す．

表1　先行研究マシン性能

Device	OS	CPU	Number of Cores	Processor Speed	RAM
Lenovo X230	Ubuntu 14.04	Intel Core i5-3320M	2	2.6 GHz	4GB

実装はシングルスレッドであるため，1つのコアのみ使用し，2880BのAES暗号文の復号に18分，16Bあたり6秒を要した．

Speeding up the sensor data encryption combined Common Key Cryptosystem with Fully Homomorphic Encryption on the IoT Device

† Marin MATSUMOTO, † Masato OGUCHI

Ochanomizu University (†)

3. 従来手法

提案手法の比較として，FHEのみを暗号化に使用する従来システムの概要を図1に示す．

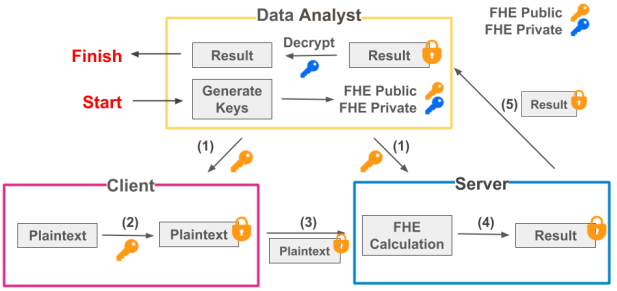

図1　従来システム(FHE Only)

(1) Data AnalystがFHEの公開鍵と秘密鍵を生成し，Client(IoTデバイスユーザ)とServer(クラウドサービス)に公開鍵を送信．
(2) ClientはFHEの公開鍵で暗号化された暗号文(Ctxt_PK)を生成．
(3) Ctxt_PKをServerに送信．
(4) ServerはCtxt_PKを使って分析．
(5) Serverは分析結果をData Analystに送信．

従来システムの問題点として，Clientにおける暗号化に時間がかかること，暗号文サイズが大きくなることが挙げられる．

4. 提案手法

4.1 概要

本研究では，図2の共有鍵暗号AESと公開鍵暗号のFHEを組み合わせた暗号化システムを提案する．

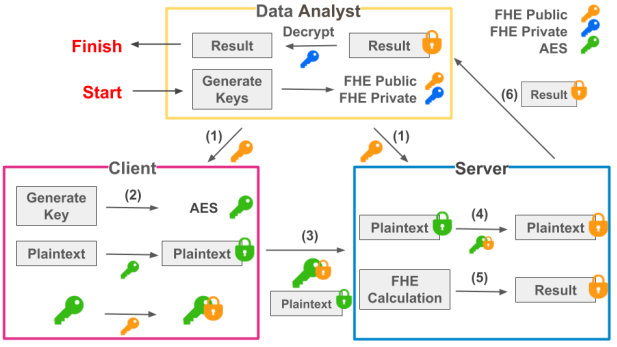

図2　提案システム(AES+FHE)

(1) Data AnalystがFHEの公開鍵と秘密鍵を生成し，Client(IoTデバイスユーザ)とServer(クラウドサービス)に公開鍵を送信．
(2) ClientはAES共通鍵(AESKey)を生成し，共通鍵で暗号化された暗号文(Ctxt_AES)とFHEの公開鍵で暗号化された共通鍵(AESKey_PK)を生成．
(3) ClientはCtxt_AESとAESKey_PKをServerに送信．
(4) ServerはCtxt_AESをAESKey_PKでAES暗号文の復号を行い，FHEの暗号文(Ctxt_PK)を取得．
(5) ServerはCtxt_PKを使って分析．
(6) Serverは分析結果をData Analystに送信．

平文が増加しても共通鍵で暗号化すれば良いため，従来手法よりも高速に暗号化できることが予想される．

4.2 FHEによるAES暗号文の復号処理

AESでは，16Bずつ区切った平文を16Bの鍵を使って暗号化し，16Bずつの暗号文を生成する．復号には，鍵と暗号文のXOR演算や定数と暗号文のAND演算を行う．FHEでは暗号化したままでXOR演算もAND演算も行うことができるため，暗号化されたAESの鍵と暗号文があれば，AESのみの復号を行い，FHEの暗号文にすることが可能である．

5. 実験

5.1 実験環境

ClientにはGoogle Pixel 3，ServerにはMacBook Proを使用した．各マシンの性能を表2に示す．

表2 マシン性能

Device	OS	CPU	Number of Cores	Processor Speed	RAM
Google Pixel 3	Android 9.0	ARM Cortex-A75 ARM Cortex-A55	8	2.5 GHz 1.6 GHz	4GB
MacBook Pro	OS X 10.15	Intel Core i5	4	1.4GHz	8GB

5.2 実験結果

平文のサイズを16B，64B，128B，192B，256B，320Bと変化させた．提案手法と従来手法のClientにおける平文の暗号化時間の比較を図3に示す．

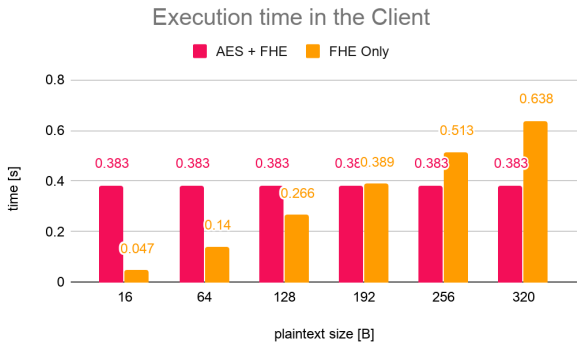

図3 Clientにおける平文の暗号化時間

平文サイズが大きい場合は従来手法FHE Onlyより提案手法のAES+FHEの方が高速で，平文サイズが増加していくにつれて，AES+FHEとFHE Onlyの差が顕著にあらわれている．一方で，平文が短い場合はAES+FHEよりも，FHE Onlyのほうが高速である．また，図4のClientが送信するファイルサイズにおいても，同様の傾向がみられる．

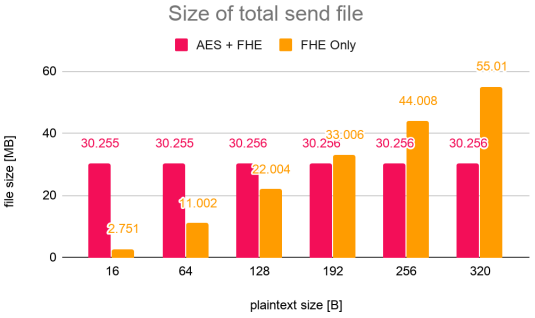

図4 Clientが送信するファイルサイズ

提案手法でのみServerが行うAES暗号文復号の実行時間を図5に示す．

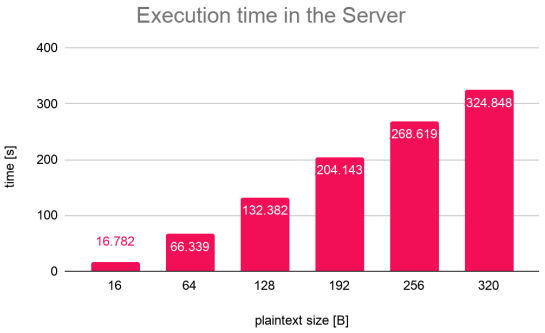

図5 提案手法のServerにおけるAES暗号文の復号時間

6. まとめと今後の課題

スマートフォンをClientとして共通鍵暗号とFHEを組み合わせた暗号を実装し，FHEの暗号化の高速化を行った．その結果，平文が長い場合は従来手法よりも提案手法の方がClientにおける暗号化が高速で，通信量を削減することが可能になった．今後はAES以外の共通鍵暗号とFHEを組み合わせた暗号の実装を検討している．

謝辞

本研究は一部，JST CREST JPMJCR1503 の支援を受けたものである．

参考文献

[1] Kristin Lauter, Michael Naehrig, Vinod Vaikuntanathan: Can homomorphic encryption be practical?, Proc. of the 3rd ACM workshop on CCSW '11, pp. 113–124, 2011
[2] Craig Gentry, Shai Halevi, Nigel P. Smart: Homomorphic Evaluation of the AES Circuit, January 3, 2015
[3] 佐藤 宏樹，馬屋原 昂，石巻 優，今林 広樹，山名 早人: 完全準同型暗号のデータマイニングへの利用に関する研究動向，第 15 回情報科学技術フォーラム，2016