5W-05

PDEの考え方を用いた暗号化ファイルシステムの検討

柴崎凌我† 稲村浩† 中村嘉隆†
公立はこだて未来大学 システム情報科学部†

1 はじめに

プライバシーに関わる機密データの保護は重要である．機密情報の存在の否認を可能にするPDE(Plausibly Deniable Encryption)が提案されており，鍵の開示の強要攻撃を阻止する．暗号化ファイルシステムにおいて主記憶装置への攻撃は一般に想定されていないが，クラウドでの利用や仮想化技術の広まりによって今や検討すべき課題となった．本研究では信頼できる実行環境（TEE：Trusted Execution Environment）を用いPDEの考え方に基づいた暗号化ファイルシステムについて実現可能なシステム構成を検討し，本提案の性能面について基礎的な評価を行った．

2 関連研究

2.1 Plausibly Deniable Encryption

PDEは暗号化手法の1つとしてCanettiら[1]によって提案された．従来の暗号化とは異なり機密情報に囮鍵と秘密鍵の両方で復号化が可能な特殊な暗号化を施す．囮鍵を用いて復号化をすると囮の平文が得られ，秘密鍵で復号化を行うと本来の平文が得られる．PDEはこの特性を利用し，攻撃者に本来の情報に気づかせないことで，データを鍵開示の強要から保護する．しかし，暗号文のサイズが大きくなるため，攻撃者に特殊な暗号を施していることに疑念を持たせる可能性がある．機密情報の痕跡がファイルシステムおよび物理記憶媒体から取得される可能性があり実用的な方法と言えない．このことより，単純な暗号化ではなく，隠しボリューム等の技術を利用してPDEの考え方を実現する方法[2,3]が提案された．しかし，いずれの手法においても実行時における機密情報の存在を司る主記憶装置が攻撃されることは想定されていない．

2.2 Intel Software Guard Extensions

Intel Software Guard Extensions(以下Intel SGXと表す)はCPUの拡張アーキテクチャである．Intel SGXは以下のTEEの機能を提供する．エンクレーブと呼ばれる暗号化領域を作成し，復号結果を外部から取得させずに処理を可能にし，検査改竄のできないCPUモードで演算を実行する．Intel SGXには次の制約事項が存在する．まず，信頼された領域であるエンクレーブはプログラムとデータのサイズに128MB程度の上限が存在すること．もう1つは，暗号化され保護されたコードの実行時には呼び出し毎にレイテンシが増大することである．

3 研究の目的

本研究はPDEの考え方を用いた暗号化ファイルシステムをTEEを活用して実現するPTEE FS (PDE with Trusted Execution Environment File System)の提案を行う．TEEとしてIntel SGXを用いたシステムを提案する．

4 PTEE FSの設計

4.1 脅威モデル

攻撃者は永続記憶装置及び主記憶装置のスナップショットが取得可能であると仮定する．攻撃者は所有者に復号鍵開示の強要が可能であり，復号鍵を取得した時点で鍵開示の強要を停止することとする．攻撃者は本システムの構成を知っているが，復号化鍵及び復号化パスワードを保持しておらず，攻撃者はマウントしているクライアントPCに対するアクセス権も保持していないこととする．

4.2 PTEE FSのシステム構成

任意の時点でのメモリ検査攻撃に対処するためPTEE FSサーバで平文ファイルを扱わない．メモリ検査に脆弱なユーザ空間で動作するアプリケーションは攻撃者から安全なクライアント側でのみ実行されるものとする．本ファイルサーバで用いる遠隔アクセスプロトコルはNFS[4]に基づく．

4.3 PTEE FSの実装における保護対象

本システムではPTEE FSは暗号化したままのファイルを操作するため，ファイルの暗号化や復号化は重要ではない．PTEE FSサーバは囮領域と隠し領域を永続記憶装置上に作成する．クライアントから提示された鍵によってアクセスする領域の切り替えを行う認可制御部にはTEEを用いてメモリ検査攻撃に対する保護を設ける．

A Study on Crypt File System based on PDE concept
Ryouga Shibazaki†, Hiroshi Inamura†, Yoshitaka Nakamura†

5　実験と評価

本設計においてFSはNFSに準じた動作を行い、概ね一回のNFS RPCにてTEE保護下の認可制御部の処理を一度実行するものと考えると，そのオーバヘッドドが性能に与える影響を検討する必要がある．

5.1　実験環境と実験方法

提案手法においてIntel SGXを利用した際の性能の一部を評価するため以下の2つの実験を行った．実験はCorei7-6700を搭載した計算機を使用した．

[実験1] I/Oサイズに対するIntel SGXの呼び出しオーバヘッドを評価するモデルを与える．保護されたコードの復号化，エンクレーブの呼び出しと脱出，CPUモードの切り替えが呼び出し毎に行われる．エンクレーブ外からエンクレーブへのバッファを引き渡す関数とエンクレーブからエンクレーブ外へバッファを引き渡す関数を呼び出すプログラムを作成し，I/Oサイズによるオーバヘッドの変化の測定を行った．

[実験2] 実用的な利用例についてNFSの実行トレースを取得し、実験1で得たオーバヘッドのモデルを用いることでTEE利用の有無が与える影響の一例を示す．openssh-8.0p1のビルドと削除を対象の応用例とし、NFSサーバに配置したソースコードをクライアント側で操作する際のトラフィックを実行トレースとし，RPCのREAD，WRITEコールで引き渡される一連のバッファのサイズを取得する．その一連のバッファサイズを実験1で利用したプログラム上でエンクレーブの呼び出しと脱出のバッファサイズとして，実行時間を計測し本システム利用時のIntel SGXのオーバヘッドとして評価を行った．

5.2　実験結果・考察

実験1の結果を図1に表す．

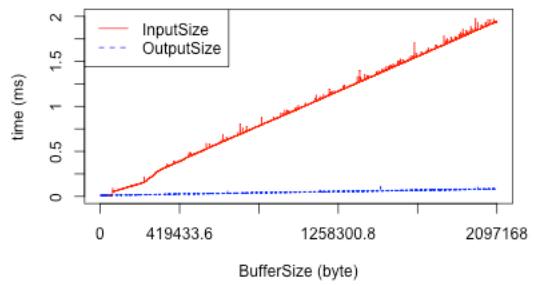

図 1 I/0サイズによるオーバヘッドの変化

オーバヘッドはInput Sizeが64kBより大きい時は線形となるため，単回帰分析により，以下のようなオーバヘッドの算出式を導き出せる．

$$OH = \begin{cases} 0.0097, & i < 0.064 \\ 0.9560 * i + 0.0354 * o - 0.0002, & i \geq 0.064 \end{cases}$$

OH：オーバヘッド(ms)
i：入力バッファ(MB)
o：出力バッファ(MB)

実験2の計測結果について考察を行う．NFSの実行トレースから取得した一連のReadサイズに対するTEE利用によるオーバヘッドの和は37.52msとなり，Writeサイズに基づくオーバヘッドは151.58msとなる．よってRPCでの一連のバッファサイズによるオーバヘッドの総和は189.1msとなる，Openssh-8.0p1のビルドにはローカルのファイルシステム上で40.132sec掛かるため．オーバヘッドの総和とビルド時間を足した値のうちオーバヘッドの割合は0.47%であり，TEEを用いる際にかかるオーバヘッドは許容される可能性があると考える．

6　おわりに

本研究はPDEの考え方を用いて存在の否認が可能な暗号化ファイルシステムの，信頼できる実行環境での実現の提案を行った．ハードウェアによる保護された実行環境としてIntel SGXを利用した．本研究では従来のPDEでは想定されていなかった主記憶装置への攻撃に対して耐性のあるシステムの提案並びに，本システムの性能面に関して部分的な評価を行った．提案システム上でIntel SGXを用いる際にかかるオーバヘッドが性能に与える影響は許容される可能性がある事が判明した．

参考文献

[1] Rein Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. “Deniable Encryption”. In Burton S. Kaliski, editor, Advances in Cryptology — CRYPTO ’97, Lecture Notes in Computer Science, pp. 90– 104. Springer Berlin Heidelberg, 1997.

[2] Ross Anderson, Roger Needham, and Adi Shamir. “The Steganographic File Sys- tem”. In Information Hiding, pp. 73–82. Springer, Berlin, Heidelberg, April 1998.

[3] Shijie Jia, Luning Xia, Bo Chen, and Peng Liu. “DEFTL: Implementing Plau- sibly Deniable Encryption in Flash Trans- lation Layer”. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS ’17, pp. 2217–2229, New York, NY, USA, 2017. ACM.

[4] B. Callaghan, B. Pawlowski, and P. Staubach. “NFS Version 3 Protocol Specification”. https://www.ietf.org/rfc/rfc1813.txt, June 1995. (accessed 2019-12- 24T00:47:14Z).