九州工業大学 情報工学部 情報・通信工学科 2025 年度 卒業論文 概要

コース	ソフトウェアデザイン	指導教員	光來 健一
学生番号	232C1048	氏　　名	倉光 周作

論文題目	TEE エミュレータのためのメモリ暗号化

1　はじめに

クラウドなどの信頼できない環境において使用中のデータを保護することができる機密コンピューティングが普及してきている．機密コンピューティングは CPU によって提供される隔離実行環境 (TEE) を用いて，TEE 内で実行されるソフトウェアのコードとデータを OS や管理者から保護することを可能にする．TEE によるメモリ保護として，TEE 外のソフトウェアからのアクセスを防ぐメモリアクセス制御や，物理的な攻撃を含めてメモリの盗聴を防ぐメモリ暗号化，メモリの改ざんを検知するメモリ整合性保証などがある．新しい TEE の開発には時間がかかるため，TEE ハードウェアが入手可能になるまでは TEE エミュレータが用いられることが多い．しかし，既存の TEE エミュレータの多くはメモリアクセス制御しか提供しておらず．メモリ暗号化は行われていない．そのため，実際の TEE ハードウェアと TEE エミュレータの動作が異なり，TEE を利用するシステムソフトウェアをテストする際に，メモリ暗号化に関連する不具合を捕捉できないという問題が生じる．

本研究では，TEE エミュレータの基盤として用いられることが多い CPU エミュレータの QEMU においてメモリ暗号化を行う EmuCipher を提案する．

2　EmuCipher

EmuCipher は図 1 のように，エミュレーション対象（ゲスト）の CPU によるメモリアクセス時にデータの暗号化・復号化を行う．メモリにデータを書き込む際に暗号化を行い，メモリからデータを読み込む際に復号を行う．ゲストのメモリ領域には RAM とメモリマップド I/O があるが，EmuCipher は RAM のみを対象とする．メモリアクセスの際に CPU は指定された仮想アドレスを物理アドレスに変換するが，その際に用いられるメモリ上のページテーブルにアクセスする際にも復号を行う．また，CPU がメモリ上に格納された命令をフェッチする際にも復号を行う．

EmuCipher は QEMU の Tiny Code Generator (TCG) を用いたメモリアクセス命令の実行時にデータの暗号化・復号化が行われるようにする．TCG はゲストが用いるターゲット命令を一旦，中間表現に翻訳し，中間表現を JIT コンパイルして実機で用いられるホスト命令を生成する．メモリアクセス命令の実行時には，対象アドレスをソフトウェア TLB から探索し，エントリがなかった場合には QEMU のヘルパー関数を呼び出す．そして，アドレス変換の結果をソフトウェア TLB に格納し，メモリアクセス命令のエミュレーションおよびデータの暗号化・復号化を行う．一方，ソフトウェア TLB にヒットした場合は，取得した物理アドレスが指すメモリに直接アクセスを行い，データの暗号化・復号化を行う．

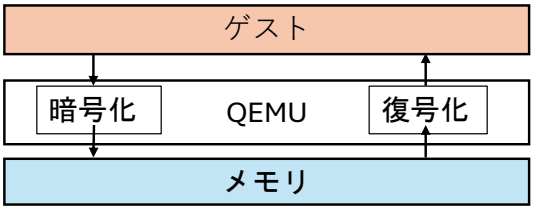

図 1. EmuCipher のシステム構成

EmuCipher では，仮想デバイスが DMA を用いて直接アクセスするメモリは暗号化の対象外とする．そのために，仮想デバイスは DMA 用のメモリアドレスをハッシュ表に登録する．このとき，既に DMA に用いるデータが書き込まれて暗号化されている可能性があるため，メモリの復号化も行う．メモリアクセス命令を実行する際にこのハッシュ表を確認し，対象のメモリアドレスが登録されていれば暗号化・復号化を行わない．

3　実験

EmuCipher を用いて RISC-V プロセッサに対応した教育用 OS である xv6 を起動し，正常に起動して操作が行えることを確認した．また，ゲストのメモリをダンプすることにより，メモリの内容が暗号化されていることを確認した．

次に，EmuCipher とメモリ暗号化を行わない従来の QEMU を用いて xv6 の性能を測定した．起動時間は図 2 に示すようになり，EmuCipher の方が 3 倍長くなった．また，大量のメモリアクセスを行うプログラムを作成し実行したところ，図 3 に示すとおり EmuCipher の方が実行時間が 33 倍長くなった．

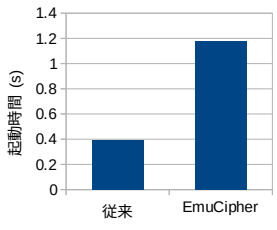

図 2. OS の起動時間

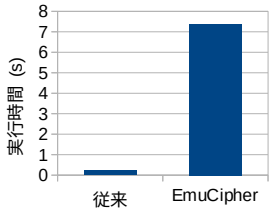

図 3. メモリアクセス時間

4　まとめ

本研究では，TEE エミュレータの基盤として用いられる QEMU においてメモリ暗号化を行う EmuCipher を提案した．今後の課題は，実際の TEE エミュレータにおいてメモリ暗号化を利用できるようにすることである．