論　文　概　要

九州工業大学大学院情報工学府　情報創成工学専攻

学生番号	17675023	氏　名	中島 健児
論文題目	Intel SGX を利用する巨大なアプリケーションのマイグレーション機構		

1　はじめに

近年，サーバで扱う機密情報が爆発的に増えてきており，情報漏洩対策の重要性が高まっている．従来，OS やハイパーバイザにおいて情報漏洩対策が行われてきたが，これらに対する攻撃も数多く報告されている．そこで，最近の Intel 製 CPU は SGX と呼ばれる機能を提供し，アプリケーションのメモリ上に CPU 固有の鍵で暗号化された領域である Enclave を作成することを可能にする．Enclave 内で機密情報を扱うことにより，OS やハイパーバイザが乗っ取られたとしても Enclave 内の情報を盗聴したり改ざんしたりすることはできない．しかし，Enclave のメモリは Enclave を作成した CPU でのみ復号可能であるため，アプリケーションを別のホストにマイグレーションするとアプリケーションの実行を継続できない．

本研究では，巨大な Enclave メモリを利用する SGX アプリケーションの効率のよいマイグレーションを可能にするシステム MigSGX を提案する．

2　MigSGX

MigSGX では Enclave 自身に状態の保存・復元を行わせることにより，マイグレーション後に Enclave の実行を継続可能にする．MigSGX では図??のように，MigSGX マネージャが SGX アプリケーションの状態を保存して移送先ホストに転送し，その状態を復元することでマイグレーションを行う．その際に，SGX アプリケーション内の MigSGX ランタイムと通信し，Enclave 内の MigSGX ライブラリを呼び出して Enclave の状態の保存･復元を行う．MigSGX マネージャはパラサイト機構を用いてコードを送り込むことにより，MigSGX ランタイムとの安全な通信を実現する．

Enclave の状態を保存する際には，MigSGX ライブラリは Enclave のヒープ領域やデータ領域を MigSGX マネージャとの間で確立した共有メモリに書き出す．MigSGX マネージャが共有メモリに書き込まれたデータを読み取ると，MigSGX ライブラリは次のデータを書き出す．このようにして，巨大な Enclave メモリであっても小さなサイズの共有メモリを用いて順に移送先ホストに転送する．その際に，Enclave の外部で動作している MigSGX マネージャは攻撃を受ける可能性があるため，Enclave 内の暗号鍵を用いてデータを暗号化する．

Enclave の状態を復元する際には，MigSGX ランタイムは新しく Enclave を作成する．その際に，マイグレーション前に Enclave 内で使われていたアドレスが再利用できるように，Enclave メモリを SGX アプリケーション内の同じアドレスに配置する．そして，Enclave 内の MigSGX ライブラリを呼び出し，保存されたデータを復号しながら共有メモリ経由で Enclave のメモリを上書きする．

3　実験

Enclave に割り当てるメモリ領域のサイズを変えながら SGX アプリケーションのマイグレーションにかかる時間を測定した．実験の結果，基本的にはメモリサイズにほぼ比例してマイグレーション時間が増加することがわかった．次に，マイグレーション時に Enclave メモリ全体を SGX アプリケーションのメモリに書き出してから転送する手法とメモリ使用量の比較を行った．小さな共有メモリを利用する場合のメモリ使用量はほぼ共有メモリのサイズ分の増加に留まっていたが，Enclave メモリ全体を書き出す手法ではほぼ Enclave メモリのサイズ分増加した．

4　まとめ

本研究では，巨大な Enclave メモリを効率よく転送して，SGX アプリケーションのマイグレーション後に Enclave を継続的に実行可能にするシステム MigSGX を提案した．今後の課題は，Enclave の状態の暗号化・復号化に利用する鍵を安全に共有する仕組みの実装である．

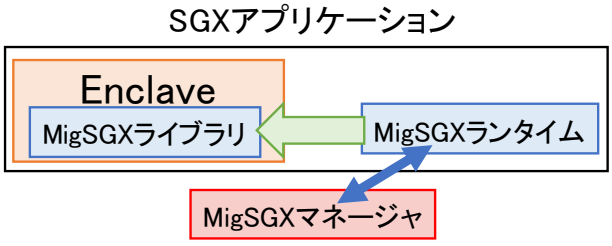

図 1: Enclave の状態の保存・復元