Vol.2013-DPS-154 No.22
Vol.2013-CSEC-60 No.22
2013/3/14

IaaSクラウドの帯域外リモート管理における情報漏洩の防止

江川 友寿[1]　西村 直樹[1]　光来 健一[1,2]

概要：IaaS クラウドにおいて，ユーザ VM を管理する権限を持つ VM（管理 VM）を経由した帯域外リモート管理を行うことで，ユーザ VM の障害時でも管理が可能となる．しかし，IaaS クラウドにおいては管理 VM が必ずしも信頼できるとは限らないため，管理 VM から情報が漏洩する危険がある．この問題を解決するために，本稿では IaaS クラウドにおいて安全な帯域外リモート管理を可能にするシステム *FBCrypt* を提案する．FBCrypt は，VNC クライアントと IaaS クラウド内の仮想マシンモニタ（VMM）でユーザ VM に対する入出力の暗号化を行うことで，管理 VM への情報漏洩を防ぐ．FBCrypt を Xen と TightVNC に実装し，キーボード入力とビデオ出力が漏洩しないことを確認した．

キーワード：仮想マシン，リモート管理，情報漏洩

Preventing Information Leakage in Out-of-band Remote Management of IaaS Clouds

Tomohisa Egawa[1]　Naoki Nishimura[1]　Kenichi Kourai[1,2]

Abstract: In Infrastructure-as-a-Service (IaaS) clouds, the users remotely manage the systems in the provided virtual machines (VMs) called user VMs. Out-of-band remote management via the management VM allows the users to manage their systems even on failures inside the user VMs. However, information leakage can occur via the management VM because it is not always trustworthy in IaaS clouds. To solve this security issue, this paper proposes *FBCrypt* for enabling secure out-of-band remote management. FBCrypt encrypts the inputs and outputs in a VNC client and the virtual machine monitor to prevent information leakage via the management VM. We have implemented FBCrypt in Xen and TightVNC and confirmed that the keyboard inputs and video outputs did not leak.

Keywords: Virtual machine, remote management, information leakage

1. はじめに

IaaS クラウドはデータセンタ内でホストされた仮想マシン（VM）をユーザに提供する．ユーザは必要に応じて提供された VM（ユーザ VM）を VNC などのリモート管理ソフトウェアを用いて管理を行う．ユーザ VM 内のネットワークや OS の障害時においても管理を行うために，IaaS 提供側は管理 VM と呼ばれる特別な特権を持った VM を経由する帯域外リモート管理を提供している．ユーザ VM にネットワーク経由で直接アクセスする従来のリモート管理とは異なり，帯域外リモート管理では，ユーザ VM 内ではなく管理 VM 内の VNC サーバに接続する．この VNC サーバは仮想キーボードや仮想ビデオデバイスなどの仮想デバイスを使ってユーザ VM に直接アクセスする．よって，ユーザ VM のネットワークが VM 内部の設定ミスにより切断されたり，システムクラッシュが起きたとしても，ユーザは VM の管理を継続することができる．

しかしながら，帯域外リモート管理には情報漏洩のセキュリティリスクがある．なぜなら，IaaS クラウド中の管理 VM は必ずしも信頼できるとは限らないからである [1], [2], [3], [4], [5]．例えば，管理 VM のセキュリティ対策が十分でない場合，外部の攻撃者に侵入される可能性

[1] 九州工業大学
Kyushu Institute of Technology
[2] 独立行政法人科学技術振興機構，CREST

Vol.2013-DPS-154 No.22
Vol.2013-CSEC-60 No.22
2013/3/14

がある．また，IaaS の管理者が内部の攻撃者として振る舞う可能性も考えられる [6]．攻撃者により，管理 VM 内のサーバを不正に改ざんされるとリモート管理に伴う入出力が容易に盗聴されてしまう．例として，攻撃者はクライアントから送信されるキーボード入力からパスワードを入手したり，重要な情報が表示されたユーザ VM の画面のスクリーンショットを盗聴することができてしまう．

この問題を解決するために本稿では，IaaS クラウドの帯域外リモート管理において管理 VM への情報漏洩を防ぐシステム *FBCrypt* を提案する．FBCrypt は仮想マシンモニタ（VMM）を用いて，VNC クライアントとユーザ VM 間の帯域外リモート管理における入出力の暗号化・復号化を行う．これにより，ユーザ VM には透過的に管理 VM への情報漏洩を防ぐことができる．IaaS 内の VMM の正当性を保証するために，FBCrypt は信用できる第三者機関を利用したリモートアテステーションを行う．

FBCrypt は仮想化ソフトウェア Xen 4.1.1 [7] および VNC クライアント TightVNC [8] を用いて実装されている．キーボード入力は VNC クライアントで暗号化され，管理 VM からユーザ VM に渡される際に VMM が介入して復号化が行われる．一方で，画面のピクセル情報を暗号化するために，VMM はユーザ VM が保持する VRAM を複製し管理 VM に見せる．FBCrypt を用いた実験を行い，帯域外リモート管理においてキーボード入力とビデオ出力が管理 VM に漏洩しないこと，および FBCrypt のオーバヘッドが許容範囲内であることを確認した．

以下，2 章では，帯域外リモート管理における情報漏洩の問題点について述べる．3 章では，この問題を解決する FBCrypt について述べ，4 章でその実装の詳細について述べる．5 章では FBCrypt を用いて行った実験について述べる．6 章では FBCrypt のセキュリティ課題について述べる．7 章で関連研究について述べ，8 章で本稿をまとめる．

2. 管理 VM への情報漏洩

IaaS により提供された VM をリモート管理するために，一般に，ユーザは VNC クライアントを用いてユーザ VM 上で動作する VNC サーバに接続する．この管理手法は管理対象システムに直接アクセスするため，帯域内リモート管理と呼ばれる．キーボード入力は VNC クライアントから VNC サーバへ送られ，ユーザ VM 内で生成されるビデオ出力は VNC サーバから VNC クライアントへ送信される．VNC クライアントとサーバ間の通信は仮想プライベートネットワーク（VPN）や SSH を使用することで，暗号化することができ，入出力情報は保護される．しかし，帯域内リモート管理にはユーザ VM の障害に弱いという欠点がある．なぜなら，ユーザ VM 内でネットワークやファイアウォールの設定を間違えると，VNC サーバにネットワーク接続することができなくなるためである．また，ユーザ VM 内の OS を起動している間や OS がクラッシュした時には VNC サーバ自体が動作していないためリモート管理が行えない．このことはリモートでシステムの詳細な挙動を把握したり障害の究明を行う際に問題となりえる．

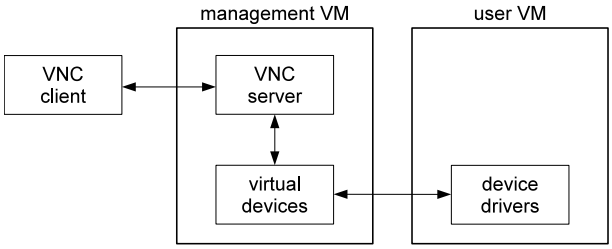

図 1 管理 VM を用いた帯域外リモート管理

このような状況でもユーザ VM のリモート管理を継続できるようにするためには，**図 1** のように管理 VM を経由してユーザ VM に間接的にアクセスする**帯域外リモート管理**を行う必要がある．管理 VM とは，全てのユーザ VM にアクセスする特権を持った VM のことであり，ハードウェア上で直接動作するハイパーバイザ型 VMM（Xen や Hyper-V）において提供される．管理 VM は，ユーザ VM に提供される仮想デバイスのエミュレーションを行う．管理 VM 中の VNC サーバは，ユーザ VM の仮想デバイスに直接アクセスすることでユーザ VM に依存しないリモート管理が可能となる．帯域外リモート管理は，ユーザ VM の VNC サーバやネットワークに依存しない．このためユーザは，ユーザ VM の障害時でもローカルコンソールからログインしているかのように VM を操作することができ，より柔軟なリモート管理が可能となる．例えば，ユーザ VM へのネットワーク接続が行えなくても管理 VM の仮想キーボードを経由して入力を行うことができ，設定ファイルの修正を行うことができる．また，仮想ビデオデバイスへのアクセスを通して OS の起動メッセージを見ることができる．

しかし，IaaS クラウドにおいて，管理 VM を用いた帯域外リモート管理には情報漏洩のリスクを増加させる懸念がある．なぜなら，IaaS 内部の管理 VM は十分に信頼できるとは限らないためである [1], [2], [3], [4], [5]．IaaS クラウド上の VM はデータセンタ間をマイグレーションで移動することがあり，その結果，セキュリティ意識の低いシステム管理者のいるデータセンタで VM が動作する可能性も考えられる．このような環境では，管理 VM のセキュリティ対策が不十分である可能性があり，外部の攻撃者により管理 VM の制御が奪われる恐れがある．また，IaaS 内部のシステム管理者自身に悪意があった場合，管理 VM の中で不正を行うことは容易である [6]．

外部の攻撃者や内部のシステム管理者によって管理 VM の権限が悪用された場合，帯域外リモート管理における入出力は容易に盗聴されてしまう．VNC クライアントと

VNC サーバ間のネットワーク上では，VPN や SSH トンネリングなどで入出力の暗号化が可能である．しかし，暗号化された入出力は管理 VM 内で復号化されるため，管理 VM への入出力情報の漏洩を防ぐことはできない．ユーザ VM で VNC サーバを動かす帯域内リモート管理では，ユーザ VM と VNC クライアント間で暗号化されるため，このような管理 VM への情報漏洩のリスクは存在しなかった．

帯域外リモート管理におけるユーザ VM へのキーボード入力は，管理 VM の VNC サーバを改ざんすることで容易に盗聴可能である．VNC サーバは VNC クライアントからキーボード入力を受け取るため，例えば，攻撃者はクレジットカード番号やログインパスワードなどを盗聴することができる．一方，攻撃者はユーザ VM のビデオ出力を管理 VM 内の仮想ビデオデバイスから得ることができる．ユーザ VM のスクリーンショットを取られるとシステムのセキュリティが低下したり，ユーザのプライバシが侵害されたりする．例えば，ソフトウェアキーボードを使って盗聴されないようにパスワードを入力したとしても，攻撃者は画面の情報からパスワードを知ることができる．また，画面を監視することでメールの内容やウェブブラウジングの履歴など，ユーザ VM の管理者が行った全ての操作を記録することができてしまう．

3. FBCrypt

管理 VM への情報漏洩を防ぐために，本稿では IaaS クラウドにおいて安全な帯域外リモート管理を可能にするシステム *FBCrypt* を提案する．

3.1 脅威モデル

FBCrypt は，外部の攻撃者や悪意をもった IaaS 管理者によって管理 VM が攻撃を受ける状況を想定している．攻撃者は管理 VM の管理者権限を奪って，OS まで変更できるものとする．本稿では，VNC クライアントと管理 VM 上で動作する VNC サーバ間でやり取りされる情報が管理 VM 上で盗聴されることに焦点を当てる．

FBCrypt は，IaaS プロバイダ自体は信頼できると仮定する [1], [2], [3], [4], [5]．VMM やハードウェアを管理する責任を持つ少数の管理者は信頼するが，管理 VM でユーザ VM を日常的に管理し，悪意をもってシステムを改ざんする可能性がある一般のシステム管理者は信頼しない．もし，一般のシステム管理者が VMM やハードウェアのメンテナンスを行う場合には，信頼できる管理者がチェックを行うものとする．したがって，VMM は正しくメンテナンスされており，脆弱性がないものとする．また，ユーザ VM が動作するハードウェアに物理的にアクセスして情報を盗む攻撃は想定しない．なぜなら，VM が稼働しているデータセンタのサーバルームは厳重に守られているからである．

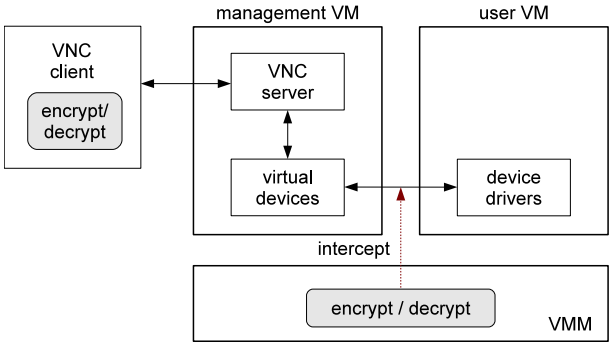

図 2 FBCrypt のアーキテクチャ

3.2 FBCrypt

FBCrypt は，VNC クライアントとユーザ VM 間の入出力を VMM を用いてユーザ VM には透過的に暗号化・復号化する．これにより，攻撃者が管理 VM を不正に改ざんして盗聴を試みたとしても管理 VM を経由する入出力を盗聴することはできない．FBCrypt のシステム構成を**図 2** に示す．

3.2.1 入出力の暗号化

ユーザ VM へのキーボード入力は，入力時に VNC クライアントが暗号化し，ユーザ VM が読み取る時に VMM が復号化する．VNC クライアントにより暗号化されたキーボード入力はまず，管理 VM の VNC サーバに送信される．VNC サーバは受信したキーボード入力を管理 VM 内に作られたユーザ VM 用の仮想キーボードに渡す．最終的に，ユーザ VM が仮想キーボードから入力を受け取る際に VMM が介入して復号化する．

暗号化にストリーム暗号を用いることで，リプレイ攻撃を防ぐことができる．ユーザ VM のゲスト OS は従来と同じインタフェースで仮想キーボードにアクセスして入力を受け取ることができるため，デバイスドライバへの修正は不要である．

VMM でキーボード入力の復号化を行う際には，攻撃者による入力の改ざんを検出するために完全性のチェックも行う．VNC クライアントから送られる入力は信頼できない管理 VM を経由するため，攻撃者により不正な入力が挿入・削除されたり，VNC クライアントからの入力が改ざんされたりする可能性がある．入力は暗号化されるため，攻撃者が意図した入力をユーザ VM に送ることは困難だが，暗号化だけでは正しくない入力が送られるのを防ぐことはできない．そこで，VNC クライアントから送られる入力にはメッセージ認証コード（MAC）を付加して VMM でチェックし，正しくない入力はユーザ VM に渡さずに破棄する．

ユーザ VM からのビデオ出力については，ユーザ VM による画面の更新時に VMM がビデオ出力を暗号化し，VNC クライアントが画面の更新情報を受け取った時に復号化する．ユーザ VM のアプリケーションが画面上にオブジェ

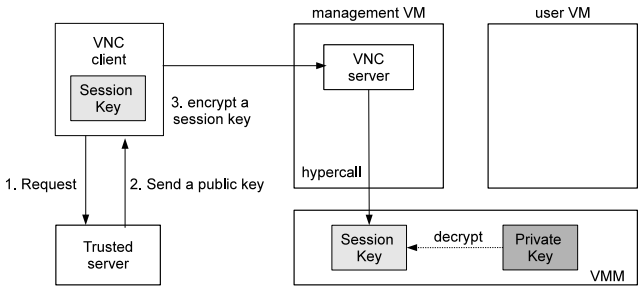

図 3 FBCrypt における鍵管理

クトを描画すると，ビデオ出力が管理 VM 内の仮想ビデオカードに送られる．この際に VMM が介入してビデオ出力を暗号化して仮想ビデオカードの VRAM に書き込む．VRAM は画面情報を保持するためのメモリ領域である．管理 VM の VNC サーバは暗号化された VRAM を読み込み，更新された領域のピクセル情報を VNC クライアントへ送る．VNC クライアントは受信したピクセル情報を復号化し，自身のウィンドウに反映させる．仮想ビデオカードの VRAM を暗号化しても VNC サーバは問題なく動作する．なぜなら，VNC サーバは画面の内容には干渉しないからである．仮想キーボードと同様に，仮想ビデオカードもまたユーザ VM のゲスト OS に従来と同じインタフェースを提供するため，ゲスト OS に修正を加える必要はない．

管理 VM やネットワーク内の攻撃者は，VNC クライアントへ送られる画面情報を盗聴することはできない．送信される画面情報は暗号化されており，これらは VMM もしくは VNC クライアントによってのみ復号化される．管理 VM 中の攻撃者は VRAM 全体に直接アクセスすることができるが，それらを復号化することはできない．加えて，暗号化された画面更新情報を改ざんすることはできない．たとえもし，攻撃者が暗号化されたある領域を別の領域にコピーしたとしても，複製された領域は正しく復号化されない．なぜなら FBCrypt は座標データも考慮して，ピクセル情報を暗号化するからである．したがって，攻撃者によって悪意ある画面更新情報が生成されたとしても，容易に検出することが可能である．そのような画面更新が正しく復号化されない際には，ユーザはそのような攻撃が行われたことに容易に気づくことができる．

3.2.2 VMM の安全性

FBCrypt は IaaS の外部に検証サーバを設置してリモートアテステーションを用いることで，IaaS 内の VMM を信頼する．リモートアテステーションは，耐タンパ性ハードウェア（TPM）[9] によってプラットフォームの完全性を第三者機関で検証する仕組みである．FBCrypt は TPM を用いて VMM のハッシュ値を計算し，検証サーバに署名付きデータを送信する．検証サーバは署名の妥当性を確認した後，ハッシュ値を照合して VMM の完全性を検証する．VMM のインストール及びリモートアテステーションの設定は，IaaS 内でも信頼できる少数の管理者が行うものとし，この設定は信頼できない IaaS 管理者によって変更されないものとする．

VMM は自身が持つ保護機構により，管理 VM からの攻撃を受けない．通常，高い特権を持って動作する管理 VM は制限なくほとんどのハードウェアにアクセスすることができる．しかし，Xen のようなハイパーバイザ型 VMM においては管理 VM と VMM はメモリ空間が分離されているため，管理 VM から VMM の CPU 状態や使用しているメモリ領域にアクセスすることはできない．したがって，管理 VM 上の攻撃者は，FBCrypt のセキュリティ機構を無効化するために VMM のコードを改ざんするといったことはできず，暗号化に用いる秘密鍵などの機密データを VMM から盗み出すこともできない．KVM のようなホスト型 VMM では，管理 VM に相当するホスト OS 内で VMM が動作するため，VMM の安全性を保つことはできない．

3.2.3 ユーザ VM からの情報漏洩の防止

ユーザ VM のメモリや CPU 状態は，セキュアな実行環境（SRE）[2], [5] や VMCrypt [4] を用いることで管理 VM から保護することができる．管理 VM はマイグレーション時などにユーザ VM を管理するために，ユーザ VM の全てのリソースにアクセスすることができる．セキュアな実行環境や VMCrypt は，管理 VM に対してユーザ VM のメモリや CPU 状態を暗号化する．FBCrypt とこれらを併用することで，管理 VM がユーザ VM の CPU レジスタやメモリからキーボード入力やビデオ出力を盗聴することを防ぐことができる．

3.2.4 鍵管理

FBCrypt は，VNC のセッション毎に入出力の暗号化に用いる共通鍵を VNC クライアントと VMM の間で安全に共有する．FBCrypt における鍵の管理を**図 3** に示す．ユーザが VM にアクセスする際には，鍵サーバから接続先の VMM の公開鍵を取得する．この際に，接続先の VMM の完全性をリモートアテステーションにより確認する．鍵サーバには信頼できる IaaS の管理者によりあらかじめ正当な VMM の公開鍵が登録されているとする．VNC クライアントは取得した VMM の公開鍵を用いて共通鍵を暗号化して管理 VM に送信する．管理 VM は暗号化された共通鍵をそのまま VMM に送り，VMM は自身の秘密鍵で共通鍵を復号化する．これにより，各ユーザの VNC クライアントと VMM 間でセッション毎に新しい共通鍵の共有が可能となる．

4. 実装

我々は FBCrypt を Xen 4.1.1 [7] および TightVNC Java Viewer 2.0.95 [8] に実装した．VMM に追加したコード行数は 5497 行である．Xen においては管理 VM はドメイン

Vol.2013-DPS-154 No.22
Vol.2013-CSEC-60 No.22
2013/3/14

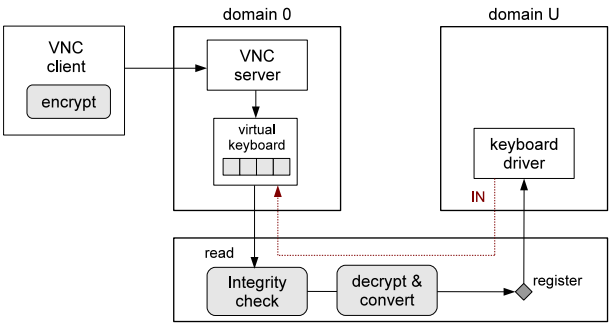

図 4 キーボード入力の暗号化・復号化

0，ユーザ VM はドメイン U となる．VNC サーバや仮想デバイス群はドメイン 0 内の QEMU の一部として動作する．FBCrypt はゲスト OS として，準仮想化と完全仮想化の両方に対応している．我々は Linux 2.6.32 と Linux 3.2.0，Windows 7，FreeBSD 9.1 で FBCrypt が動作することを確認した．紙面の都合上，ここでは完全仮想化における実装についてのみ述べる．

4.1 キーボード入力の暗号化・復号化

FBCrypt は VNC クライアントからドメイン U 内のキーボードドライバにキーボード入力を安全に送信する．

4.1.1 キーボード入力処理

キーボード入力処理の流れを**図 4** に示す．VNC サーバが受信したキーボード入力は，暗号化されたまま仮想キーボードのキューに格納される．よって，ドメイン 0 がキューに格納された情報を盗聴することはできない．ドメイン U が仮想キーボードのキューから情報を取り出すために IN 命令を発行すると，VMM がそれをトラップしてエミュレーションを行う．この時に VMM は，キューからキーボード入力を取り出して復号化し，仮想 CPU のレジスタ経由でドメイン U に渡す．レジスタは SRE を用いてドメイン 0 に対して暗号化することにより，ドメイン 0 からレジスタの内容が盗聴されることを防ぐことができる．

VMM は復号化したキーボード入力をレジスタに格納する前に，コード変換を行う．仮想キーボードから入力を読み取る IN 命令は，キーボード入力をキーコードとして受け取る仕様になっている．キーコード（スキャンコード）はキーを押したり離したりした場合に生成される．しかしながら，VMM によって復号化されたキーボード入力は，VNC が扱う keysym のままである．keysym とは X ウィンドウシステムで定義されており，通常のキーについては ASCII コードと一致する．VMM は対応表を用いて，keysym をキーコードに変換する．従来は VNC サーバが ASCII コードをキーコードに変換していたが，FBCrypt では VNC サーバで変換ができない．なぜなら，キーボード入力は暗号化されているからである．

4.1.2 キーボード入力の暗号アルゴリズム

キーボード入力の暗号化にはストリーム暗号として AES の CTR モード [10] を用いた．AES-CTR モードは，カウンターブロックの値を AES で暗号化することによりキーストリームを生成し，その値との排他的論理和をとることでデータを暗号化する．また，キーストリーム中のキーを全て使いきった時に，カウンターの値をインクリメントして新たにキーストリームを生成する．AES-CTR モードなどのストリーム暗号を用いることにより，同じ入力に対しても毎回異なる暗号結果を得ることができる．AES-CTR モードの実装のために，FBCrypt では CyaSSL [11] ライブラリを VMM に組み込み，VNC クライアントである TightVNC Java Viewer [8] では，Java 標準 API を用いた．

VNC クライアントが終了した時は，内部状態であるキーストリームは破棄されてしまう．しかし，VMM は単独では VNC 接続が終了したかどうかを認識することができないため，内部状態を保持し続けるかどうかの判断ができない．VNC クライアントと VMM で内部状態を同期させるために，VNC クライアントが VNC サーバに再接続した際に，VNC サーバは VMM に内部状態を初期化するためのハイパーコールを VMM に発行する．

4.1.3 不正なキーボード入力の検知

FBCrypt の VMM は，キーボード入力の復号化を行う前に，入力の完全性のチェックを行う．VNC クライアントがキーボード入力を暗号化して VNC サーバに送信する際に，キーボード入力，シーケンス番号，VMM と共有している共通鍵からハッシュ値を計算し，メッセージ認証コード（MAC）として一緒に送信する．ユーザ VM がキーボード入力を読み取る際に，VMM はキーボード入力からハッシュ値を計算して MAC との照合を行う．入力の完全性を確認できれば，復号処理を実行する．照合結果が異なる場合は，攻撃者による改ざんや挿入・削除などの不正があったと見なして復号化は行わず入力を破棄する．VNC セッション開始時に共有した秘密鍵とシーケンス番号により，攻撃者が正しいハッシュ値を計算することはできず，リプレイ攻撃も防ぐことができる．

4.2 ビデオ出力の暗号化・復号化

FBCrypt は，ドメイン U のビデオドライバが更新したピクセル情報を VNC クライアントに安全に送信する．

4.2.1 ビデオ出力処理

更新されたピクセル情報はドメイン U のビデオドライバから VNC クライアントへ送られる．これを**図 5** に示す．従来は，ドメイン 0 の仮想ビデオカードがドメイン U 内にビデオメモリ（VRAM）を確保し，ビデオドライバと VRAM を共有していた．これはドメイン 0 がドメイン U 用にメモリマップド VRAM をエミュレートしていることと等しい．FBCrypt において，VMM は仮想ビデオカード

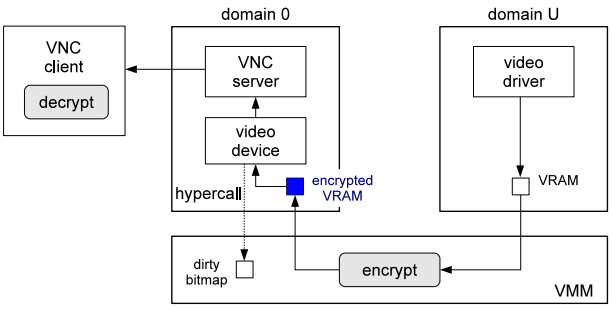

図 5 画面情報の安全な送信

用に VRAM を複製して暗号化を行う．ドメイン U 内のアプリケーションがオブジェクトを描画する時，ビデオドライバが自身の VRAM を更新すると，VMM がオリジナルの VRAM と複製した VRAM の同期を行う．

仮想ビデオカードがドメイン U の VRAM をマップしようとした時に，VMM は通常の VRAM として使用されるメモリとは別に新たに VRAM を複製し，それをマップする．通常の VRAM のマッピングとドメイン U 内に新たに複製した VRAM であることを区別するために，ドメイン 0 の仮想ビデオカードはハイパーコールを使用して，VMM に VRAM の物理アドレスとそのサイズを通知する．もし悪意ある仮想ビデオカードが元の VRAM とは異なるアドレスを VMM に通知したとすると，複製されていない VRAM を直接マップすることができる．しかし，その VRAM のメモリ領域は VRAM として登録されていないので，VMCrypt や SRE を用いることによって暗号化される．よって VRAM の情報は漏洩しない．

4.2.2 VRAM 同期の最適化

VMM は VNC クライアントから画面更新要求が送られてきた時にだけ VRAM の同期を行う．ユーザ VM が VRAM を更新してもすぐには同期を行わないことで，同期の頻度を減らす．VNC サーバが画面更新要求を受け取ると，ドメイン 0 の仮想ビデオカードが更新領域のチェックを行うために，VMM から dirty bitmap を取得する．そこで，VMM にハイパーコールが発行されたタイミングでオリジナルの VRAM と複製した VRAM の同期をとる．

VMM はこの dirty bitmap を利用して更新された領域のみの同期を行う．dirty bitmap は VMM 中で VRAM の更新を追跡するために使用される．dirty bitmap の各ビットは VRAM の各ページに対応しており，一致するページが更新された時にビットがセットされる．この仕組みはダーティページの追跡や，変更されたページのみマイグレーション先に送信するライブマイグレーションで用いられている．VMM はオリジナルの VRAM 中のダーティページの内容のみを暗号化して，複製した VRAM に書き込みを行う．

4.2.3 ビデオ出力の暗号アルゴリズム

ピクセル情報の暗号化と復号化のために，FBCrypt は 24 ビットカラーの 2 ピクセルを扱えるよう 48 ビットをブロックサイズとして改変した RC5 [12] を用いる．一般に，更新領域は任意の矩形となるため，VRAM は 1 ピクセル単位で暗号化することが望ましい．しかし，24 ビットのブロックサイズでは標準ブロックサイズである 32 ビットよりも暗号強度の低下が懸念されたため，48 ビットのブロックサイズで暗号化するようにした．加えて，FBCrypt は各ピクセルの位置座標も考慮して暗号化を行う．もし位置座標を考慮せずに同じ鍵を使ってピクセル情報を暗号化した場合，ドメイン 0 内の攻撃者は画面に表示されているおおよその内容を盗聴することができてしまうからである．これはピクセルが同じ色情報を表すなら暗号化した結果も同じになるためである．

5. 実験

FBCrypt により情報漏洩が防止できていることを確認し，そのオーバヘッドを測定する実験を行った．実験には Intel Core 2 Quad Q9550 2.83GHz の CPU を搭載したマシンを VNC クライアント用と VM 用にそれぞれ用意し，ギガビットイーサネットを用いて接続した．サーバマシンでは改変した Xen 4.1.1 を動作させて，オリジナルの Xen と比較した．各ドメインには CPU コアを一つずつ割り当てて，メモリはドメイン U には 1GB，ドメイン 0 には 3GB 割り当てた．ドメイン U において Linux 2.6.32.21 を動作させ，画面の解像度は 800 × 600 に設定した．ドメイン 0 においては Linux 3.1.1 を動作させた．クライアントマシンでは改変した TightVNC Java Viewer 2.0.95 を Linux 2.6.38.8 上で動作させた．クライアントマシンのメモリは 8GB であった．

我々は入力の暗号化のために AES-CTR，MAC の計算のために SHA-1 を用いた．画面の暗号化には RC5 を用いた．ブロックサイズは 48 ビット，鍵長は 192 ビット，ラウンド数は 16 とした．

5.1 入出力情報の漏洩防止の確認

まず，管理 VM 上の VNC サーバにキーロガーを組み込み，ユーザ VM へのキーボード入力の盗聴を行った．FBCrypt を用いずに帯域外リモート管理を行った場合，VNC クライアントで入力したキーボード入力はログインユーザ名とパスワードが平文のまま記録された．一方で FBCrypt を使用した場合，VNC クライアントで行ったキーボード入力は暗号化されて記録されており，情報が漏洩していないことを確認した．

次に，管理 VM の VNC サーバにスクリーンキャプチャを実装し，帯域外リモート管理においてユーザ VM の画面情報の盗聴を行った．FBCrypt を使用しない場合はユーザ VM の画面が記録され，攻撃者は表示された内容を取得できた．しかし，FBCrypt を使用した場合，管理 VM から画

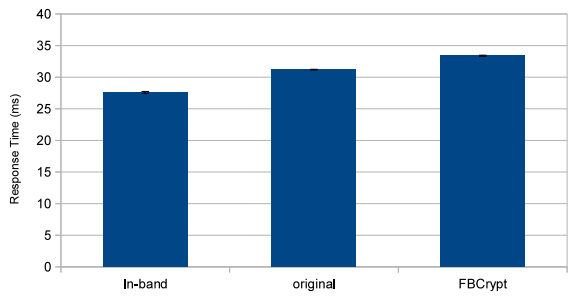

図 6 キーボード入力のレスポンスタイム

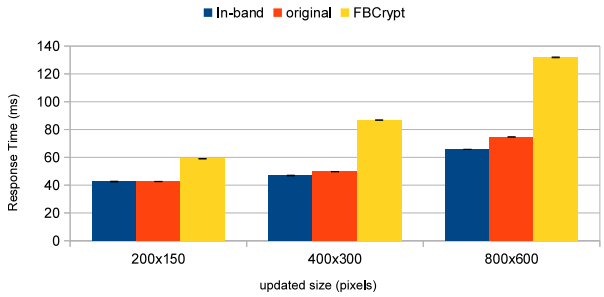

図 7 画面更新のレスポンスタイム

面の内容は認識できず情報が漏洩しないことを確認した．

5.2 キーボード入力のレスポンスタイム

キーボード入力一回あたりのレスポンスタイムを FBCrypt とオリジナルで比較した．VNC クライアントでキーボード入力を行い，文字が表示されて画面が書き換わることで送られてくる画面更新をクライアントが受け取るまでの時間を測定した．この実験において，ドメイン U はキーボード入力を読み取り，対応する文字をターミナル上に表示した．我々はレスポンスタイムを 10 回測定し，その平均値と標準誤差を得た．比較のために，VNC サーバをドメイン 0 ではなくドメイン U で動作させて接続する帯域内リモート管理のレスポンスタイムも測定した．実験結果を**図 6** に示す．FBCrypt により，レスポンスタイムはオリジナルより 2.2ms 増加した．したがって，FBCrypt によるレスポンスタイムの増加は入力に支障が出るレベルではないことがわかった．

5.3 画面出力のレスポンスタイム

様々なサイズの画面更新におけるレスポンスタイムを測定した．レスポンスタイムは，VNC クライアントが画面更新要求を送信してから対応する画面更新を受け取り，自身のウィンドウに描画するまでの時間である．測定には，ドメイン U 内で任意のサイズの矩形を更新し続けるベンチマークプログラムを使用した．レスポンスタイムを 10 回測定し平均と標準誤差を得た．また，比較のために帯域内リモート管理の場合も測定した．レスポンスタイムを**図 7** に示す．更新される画面サイズが大きくなるにつれて，レスポンスタイムも長くなっていることが分かる．

6. 議論

6.1 マウス入力の暗号化

FBCrypt はマウス入力の暗号化には未対応であるが，マウスの動き（位置）とクリックの暗号化も必要である．マウスカーソルの位置とクリックの情報から，ユーザ VM 内での操作が漏洩する危険があるためである．例えば，キー配列が固定のソフトウェアキーボードを使った場合，キー入力を盗まれる可能性がある．しかし，マウスに関するイベントはマウスカーソルの位置が変わる度に生成されるため，キーボード入力やマウスクリックのイベントよりも，はるかに高い頻度で VNC サーバにイベントが送信される．したがってマウスの動きを暗号化・復号化すると，キーボード入力時より大きな遅延が発生すると考えられる．

6.2 画面の更新領域の漏洩対策

VNC サーバは更新があった画面領域のみをクライアントに送信するため，FBCrypt によって画面を暗号化したとしても，更新領域の情報から情報漏洩につながる可能性がある．例として画面を連続的にモニタされた場合，6.1 で述べたように，マウスカーソルの動きからユーザの入力した情報が漏洩する危険がある．対策としては，暗号ブロックのサイズを大きくして変更された画面領域の特定を難しくする方法が考えられる．暗号ブロックのサイズを大きくするほど安全になるが，それだけ VNC クライアントに送られるデータ量が増えるという問題がある．

6.3 暗号化の負荷軽減

FBCrypt による VRAM の暗号化はメモリページ単位で行われるため，実際の更新領域以外の部分も暗号化される場合がある．24 ビットカラーの場合，1 ページには 1365.5 ピクセルが X 方向に順に配置されている．文字やマウスカーソルのように横幅が小さい矩形が更新された場合でも，暗号化される画面領域は横幅一杯に拡がるため，VMM に大きな負荷がかかる．したがって，VRAM の暗号化の際には VNC サーバと同様にしてピクセル単位で更新領域を検出する処理を VMM に追加する必要があると考えている．

一方，ハードウェアを活用して VRAM の暗号化を高速化することも考えられる．多くの CPU コアを搭載したサーバにおいて，使用率が低い複数のコアに並列処理させる方法や，インテルの AES-NI [13] などの暗号化処理を高速化する CPU 命令を用いるといった方法がある．

6.4 正しいユーザ VM への接続

VNC クライアントがユーザの意図する VM に正しく

接続されることを保証する必要がある．現在の実装では，ユーザは管理 VM のホスト名または IP アドレスと，VM の番号に対するポート番号を指定して VNC 接続を行う．意図した管理 VM かどうかは，サーバ証明書によって確認することができる．しかし，意図した VM かどうかを確認するのは難しい．悪意をもった管理者が用意した VM に接続させられた場合，ユーザはそのことに気がつかない可能性がある．その結果，VM からの情報漏洩につながる危険がある．そのため，ユーザが自分の VM かどうかを判別する手段が必要である．

7. 関連研究

Xoar [14] では QEMU に組み込まれた VNC サーバを QemuVM と呼ばれる専用の VM で動作させる．Xen では従来より，QEMU をスタブドメインと呼ばれる VM で動かすことが可能である．この専用 VM の中で小さな OS である mini-os を動かすことにより，VM が攻撃を受ける可能性を低くすることができる．しかし，もし VNC サーバが攻撃を受けるとリモート管理に伴う入出力情報が漏洩する．加えて，このアーキテクチャは悪意を持った IaaS 管理者による内部からの攻撃については考慮していない．

VMware vSphere Hypervisor（ESXi) [15] では VMM 内で VNC サーバを動作させており，クライアントは VMM 経由でユーザ VM の帯域外リモート管理を行うことができる．管理 VM を経由しないため，リモート管理に伴う入出力情報の漏洩は発生しない．しかし，VNC サーバに脆弱性があった場合，VMM 自体に攻撃の影響が及ぶことになり，入出力情報が漏洩する可能性がある．FBCrypt は VNC サーバが改ざんされたとしても情報が漏洩することはない．

CloudVisor [3] では VMM の下でセキュリティモニタを動作させ，ユーザ VM のメモリやストレージの暗号化を行う．CloudVisor は管理 VM だけでなく VMM も信頼しておらず，ユーザ VM のメモリやストレージから管理 VM および VMM への情報漏洩を防ぐことができる．ただし，帯域外リモート管理の入出力情報の扱いについては考慮されていない．

8. まとめ

本稿では，IaaS クラウドにおいて安全な帯域外リモート管理を可能にするシステム FBCrypt を提案した．帯域外リモート管理において，管理 VM 経由の情報漏洩を防ぐために，FBCrypt は，VMM を用いて VNC クライアントとユーザ VM 間の入出力の暗号化・復号化を行う．キーボード入力は，VNC クライアントによって暗号化され，ユーザ VM が読み出す時に VMM によって復号化される．画面出力は，VNC クライアントからの画面更新要求を受け取った際に VMM によって暗号化され，クライアントによって復号化される．FBCrypt を Xen と TightVNC に実装し，キーボード入力とビデオ出力が漏洩しないことを確認した．今後の課題は，SSH など VNC 以外のリモート管理ソフトウェアに FBCrypt を適用させることである．

参考文献

[1] Santos, N., Gummadi, K. P. and Rodrigues, R.: Towards Trusted Cloud Computing, *Proc. Workshop Hot Topics in Cloud Computing* (2009).
[2] Li, C., Raghunathan, A. and Jha, N. K.: Secure Virtual Machine Execution under an Untrusted Management OS, *Proc. Intl. Conf. Cloud Computing*, pp. 172–179 (2010).
[3] Zhang, F., Chen, J., Chen, H. and Zang, B.: CloudVisor: Retrofitting Protection of Virtual Machines in Multi-tenant Cloud with Nested Virtualization, *Proc. Symp. Operating Systems Principles*, pp. 203–216 (2011).
[4] Tadokoro, H., Kourai, K. and Chiba, S.: Preventing Information Leakage from Virtual Machines' Memory in IaaS Clouds, *IPSJ Online Transactions*, Vol. 5, pp. 156–166 (2012).
[5] Li, C., Raghunathan, A. and Jha, N. K.: A Trusted Virtual Machine in an Untrusted Management Environment, *IEEE Transactions on Services Computing*, Vol. 5, No. 4, pp. 472–483 (2012).
[6] TechSpot News: Google Fired Employees for Breaching User Privacy, http://www.techspot.com/news/40280-google-fired-employees-for-breaching-user-privacy.html (2010).
[7] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A.: Xen and the Art of Virtualization, *Proc. Symp. Operating Systems Principles*, pp. 164–177 (2003).
[8] TightVNC Group: TightVNC, `http://www.tightvnc.com/`.
[9] Trusted Computing Group: TPM Main Specification, `http://www.trustedcomputinggroup.org/` (2011).
[10] NIST: Advanced Encryption Standard (AES)., *FIPS Publication 197* (2001).
[11] yaSSL: CyaSSL Embedded SSL Library, `http://www.yassl.com/` (2013).
[12] Rivest, R. L.: The RC5 Encryption Algorithm, *Proc. Workshop Fast Software Encryption* (1994).
[13] Intel, Inc.: Intel Advanced Encryption Standard Instructions, http://software.intel.com/en-us/articles/intel-advanced-encryption-standard-instructions-aes-ni (2012).
[14] Colp, P., Nanavati, M., Zhu, J., Aiello, W., Coker, G., Deegan, T., Loscocco, P. and Warfield, A.: Breaking Up is Hard to Do: Security and Functionality in a Commodity Hypervisor, *Proc. Symp. Operating Systems Principles*, pp. 189–202 (2011).
[15] VMware Inc.: VMware vSphere Hypervisor, `http://www.vmware.com/`.