情報処理学会論文誌 Vol.53 No.2 1–9 (Feb. 2012)

IaaS環境における安全な帯域外リモート管理機構

江川 友寿[1] 西村 直樹[1] 光来 健一[1,2]

受付日 2011年11月4日, 採録日 2011年12月1日

概要：IaaS 型クラウドにおいて，ユーザは提供された VM（ユーザ VM）をリモートから管理する．ユーザ VM を管理する権限を持つ VM（管理 VM）経由で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 においては管理 VM が信頼できるとは限らないため，帯域外リモート管理に伴うキーボード入力やビデオ出力は外部もしくは内部の攻撃者によって容易に盗聴されてしまい，重大な情報漏洩につながる危険がある．この問題を解決するために，本稿では IaaS 環境においても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い，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

Secure Out-of-band Remote Management in IaaS Clouds

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

Received: November 4, 2011, Accepted: December 1, 2011

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, the management VM is not always trustworthy in IaaS clouds. Outside or inside attackers in the management VM can easily eavesdrop on the keyboard inputs and video outputs in out-of-band remote management. 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

Infrastructure as a Service（IaaS）はユーザに仮想マシン（VM）を提供するクラウドサービスである．ユーザはサービス提供のためのハードウェアを用意することなく，クラウド上の VM を必要な時に必要なだけ利用することができる．ユーザは VNC などのリモート管理ソフトウェアを用いて，提供された VM（ユーザ VM）にネットワーク経由で直接アクセスして内部のシステムの管理を行う．しかし，ネットワークや OS の障害時にはユーザ VM へのアクセスができなくなり，以降の管理が困難になってしまう．

ユーザ VM の障害時でも管理を行う方法としては，ユーザ VM を管理する権限をもつ VM（管理 VM）を経由して帯域外リモート管理を行うことが考えられる．しかし，IaaS 環境においてこのような帯域外リモート管理を行うと情報漏洩のリスクが高まるという問題がある．なぜなら，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]．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がある．また，クラウドプロバイダの悪意を持ったシステム管理者が攻撃を行う可能性も考えられる [3]．例として管理 VM 内

[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

の VNC サーバが改ざんされた場合，ユーザ VM へのキーボード入力や画面操作中のビデオ出力が盗聴され，パスワードやクレジットカード番号などの機密情報が攻撃者に漏洩してしまう．

この問題を解決するために本稿では，IaaS 環境においても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により，管理 VM からの情報漏洩を防ぐ．ユーザ VM へのキーボード入力は，VNC クライアントにより暗号化され，VMM により復号化される．ユーザ VM からのビデオ出力は，VMM によって暗号化され，VNC クライアントにより復号化される．また，FBCrypt はユーザ VM に対するキーボード入力の完全性のチェックも行う．これにより，攻撃者によるユーザ VM への不正な入力は即座に検出される．

我々は，FBCrypt を Xen 4.1.1 [4] および TightVNC [5] に実装した．キーボード入力については準仮想化と完全仮想化に対応しており，準仮想化では VMM がユーザ VM の I/O リングに復号化して書き込み，完全仮想化では IN 命令のエミュレーション時に VMM が復号化する．ビデオ出力については準仮想化のみ対応しており，VMM が仮想フレームバッファ（VFB）を二重化し，管理 VM には暗号化した VFB を参照させる．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，および暗号化と復号化における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 IaaS 環境における情報漏洩の問題について述べる．3 章で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 FBCrypt を用いて行った実験について述べる．6 章で関連研究に触れ，7 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．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．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．

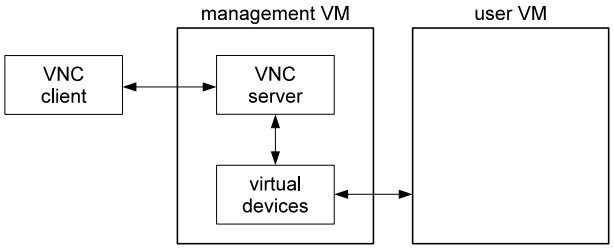


図 1 管理 VM を用いた帯域外リモート管理

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 を用いてユーザ 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 は特権を持った VM であり，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 上で VNC サーバを動作させ，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により，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，仮想ビデオカードへのアクセスを通して OS の起動時にも起動メッセージを見ることができる．Xen や KVM，VMware vSphere Hypervisor などの代表的な仮想化ソフトウェアは帯域外リモート管理機能を備えており，以上のようなメリットから，帯域外リモート管理の機能を提供する IaaS プロバイダが現れている [6]．

しかし，IaaS 環境においては，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]．IaaS 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，システム管理者の怠慢により管理 VM に脆弱性が残っていたりすると，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．

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 か

ら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 内の仮想ビデオカード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

この問題を解決するために，本稿では IaaS 環境においても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する [7]．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．また，VMM には脆弱性がないものとし，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．一般に，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．

ユーザ VM はユーザに正しく管理されているものとし，パスワードやソフトウェアに脆弱性はないものとする．入出力の暗号化と復号化を行うクライアント環境も十分に信頼できるものとし，クライアント PC もしくは VNC クライアントからの情報漏洩はないものとする．

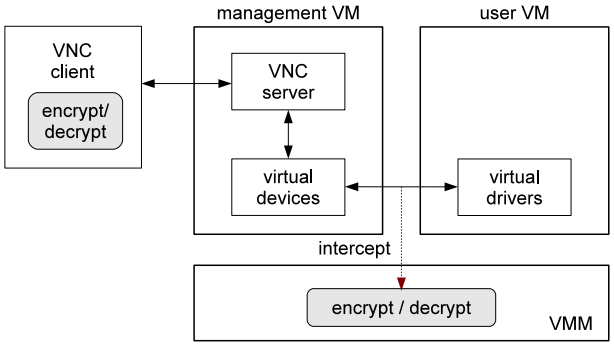


図 2 FBCrypt の構成

3.2 FBCrypt

安全な帯域外リモート管理を実現するために，FBCrypt は VNC クライアントと VMM でユーザ VM への入出力を暗号化する．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 を経由する入出力情報が漏洩することはない．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およびマウス入力については，入力時に VNC クライアントが暗号化し，ユーザ VM に渡される時に 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，仮想キーボードがユーザ VM にキーボード入力を渡す際に VMM が介入して復号化する．マウス入力についても同様である．暗号化にストリーム暗号を用いることで，リプレイ攻撃も防ぐことができる．およびユーザ VM の仮想入力デバイスへ入力情報の書き込みを行う．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や仮想マウスにアクセスして情報を取り出すことができるため，デバイスドライバへの修正は不要である．

VMM でキーボード入力やマウス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の意図した入力をユーザ VM に送るのは難しいが，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 でチェックすることにより，ユーザ VM には正しい入力のみを渡す．これにより，正しくない入力を即座に検出することができる．

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クトを描画する時，ビデオ出力が管理 VM 内の仮想ビデオカードに渡される．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フレームバッファに書き込む．フレームバッファ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フレームバッファを読み込み，更新された領域のビデオ出力を VNC クライアントへ送る．VNC クライアントは受信したビデオ出力を復号化し，自身のウィンドウに反映させる．管理 VM のフレームバッファを暗号化しても 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せずに，暗号化されたフレームバッファを暗号化されていないフレームバッファとして処理するからである．仮想入力デバイスと同様に，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．

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）[8] によってプラットフォームの完全性を外部から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．

FBCrypt は Xen のようなタイプ 1 VMM の利用を仮定しており，VMM は管理 VM から守られる．Xen において管理 VM（ドメイン 0）は特権を与えられているが，アドレス空間は分離されている．そのため，管理 VM から VMM 内の秘密鍵を盗んだり，VMM 内の処理を書き換えたりすることはできない．一方，KVM のようなタイプ 2VMM 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 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．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．VMM の秘密鍵は TPM 内によって封印（暗号化）され，正しい VMM が起動されてアテステーションに成功した時だけ取り出すことができる．そのため，管理 VM から VMM の実行ファイルが見られたとしても秘密鍵が漏洩することはない．

4. 実装

我々は FBCrypt を Xen 4.1.1 [4] および TightVNC Java Viewer 2.0.95 [5] に実装した．Xen においては管理 VM はドメイン 0，ユーザ VM はドメイン U となる．管理 VM 内で動作する IO エミュレータの QEMU に VNC サーバが含まれている．ユーザ VM 内で動作させるゲスト OS として準仮想化 Linux 2.6.39.3 と完全仮想化 Linux2.6.39.3 を対象とした．

4.1 キーボード入力の暗号化・復号化

キーボード入力は VNC クライアントによって暗号化され，ドメイン 0 上の VNC サーバに送信される．VNC サーバからドメイン U にキーボード入力が渡されるまでの処理は準仮想化と完全仮想化で異なる．

4.1.1 復号処理

準仮想化におけるキーボード入力の復号処理を**図 3** に示す．VNC サーバが受信したキーボード入力を仮想キーボードに渡すと，仮想キーボードは新規に追加されたハイパーコールを用いてその入力情報を VMM に渡す．VMM は，受け取ったキーボード入力を復号化しドメイン U の I/O リングに書き込みを行う．この I/O リングは仮想キーボードの入力キューとして使われるリングバッファである．従来は，VNC サーバがキーボード入力を受け取ると仮想キーボードが直接，ドメイン U の I/O リングに書き込みを行っていた．FBCrypt では，VMM が従来と同様に I/O リングにキーボード入力情報を書き込むことにより，ドメイン U の準仮想化キーボードドライバ（kbdfront）は従来通りに I/O リングからキーボード入力情報を読み出すことができる．

完全仮想化におけるキーボード入力の復号処理を**図 4** に示す．VNC サーバが受信したキーボード入力を仮想キーボードに渡すと，仮想キーボードは自身にその情報を格納する．ドメイン U が仮想キーボードに対して 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．この時にドメイン 0 のキーボードと通信してキーボード入力を受け取り，このキーボード入力はレジスタを経由して，ドメイン U に渡される．仮想キーボードが受け取るキーボード入力は暗号化されているため，VMM は

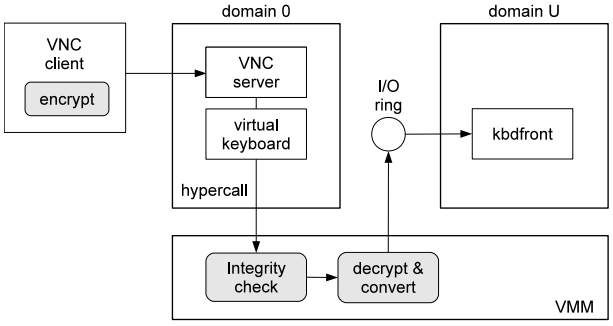


図 3 準仮想化におけるキーボード入力の暗号化・復号化

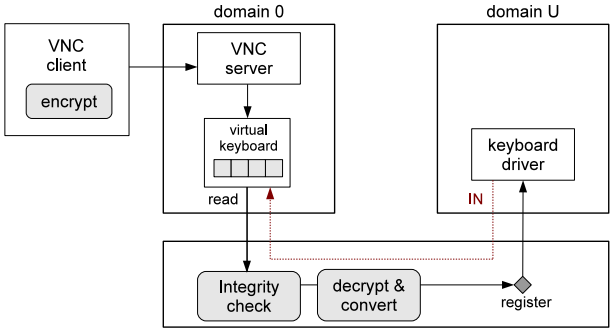


図 4 完全仮想化におけるキーボード入力の暗号化・復号化

キーボード入力をレジスタにセットする際に復号化を行う．

4.1.2 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，そのシーケンス番号，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．VNC サーバが受信した MAC を VMM に渡した際に，VMM は一緒に渡されたキーボード入力からハッシュ値を計算して照合を行う．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．

4.1.3 キーコードへの変換

ドメイン U の OS は仮想化されていないハードウェアキーボードからと同様に，キーボード入力をキーコード（スキャンコード）で受け取るように実装されている．VMM によって復号化されたキーボード入力は ASCII コードであるため，ドメイン U に渡す際にキーコードに変換する必要がある．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 ASCII コードが暗号化されているため VMM で復号化した後に変換を行う．

4.1.4 I/O リングのアクセス

準仮想化において，FBCrypt の VMM は，ドメイン U 起動時に準仮想化キーボードの入力キューとして使われる IO リングを認識する．ドメイン U は，起動時に仮想キーボードドライバで I/O リングを含むメモリページである xenkbd ページを初期化した後，このページのフレーム番号をドメイン 0 の XenStore に登録する．XenStore はドメイン間で情報を共有するため簡易データベースである．VMM は，ドメイン U とドメイン 0 の間の通信を監視し，xenkbd ページの情報を取得する．XenStore との通信の監視の詳細は 4.3 節で述べる．

従来，ドメイン 0 の仮想キーボードがドメイン U の I/O リングにキーボード入力を書きこんでいたが，FBCrypt ではドメイン 0 からドメイン U の I/O リングにアクセスすることを禁止する．これは復号化後のキーボード入力をドメイン 0 から盗まれるのを防ぐためである．ドメイン 0 は，XenStore から I/O リングを含む xenkbd ページのフレーム番号を取得して，そのメモリページを自身のメモリ空間にマップすることで，復号後のキーボード入力を盗み見ることができてしまう．そこで，FBCrypt の VMM はドメイン 0 に対して，xenkbd ページのマップを禁止する．ドメイン 0 がドメイン U のメモリページをマップするためには，ページテーブルへのエントリの追加が必要となる．物理メモリを管理しているのは VMM なので，ドメインのページテーブル変更の際には，ハイパーコールを実行する必要がある．このハイパーコールがドメイン 0 によって発行されており，かつ事前に XenStore リングから取得した xenkbd ページのフレーム番号と，要求されたフレーム番号が一致した場合，ページテーブルへのエントリの追加を失敗させる．

4.1.5 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 [9] を用いた．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．FBCrypt では CyaSSL [10] 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 [5] では，Java 標準 API を用いた．VNC クライアントは，キーボード入力を 2 バイト単位で暗号化して VNC サーバに送る．2 バイト単位で暗号化するのは，Enter キーや Delete キーなど 2 バイト値で表現される入力に対応するためである．

4.2 ビデオ出力の暗号化・復号化

準仮想化においては，ドメイン U の準仮想化ビデオドライバ（fbfront）が仮想ビデオカードのフレームバッファ（VFB）を直接更新し，ドメイン 0 の VNC サーバがそれを読み込むように設計されている．そこで，VFB を二重化して，ドメイン U には暗号化されていない VFB を見せ，ドメイン 0 には暗号化した VFB を見せる．完全仮想化についても同様であるが，現在のところ，未実装である．

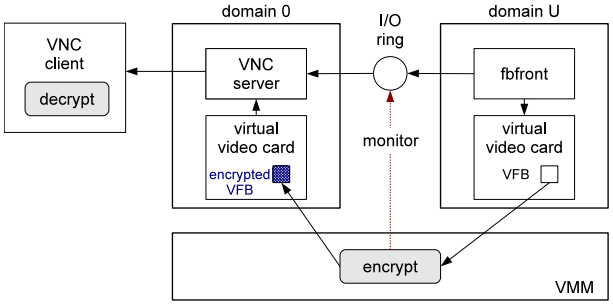


図 5 暗号化されたビデオ出力の送信

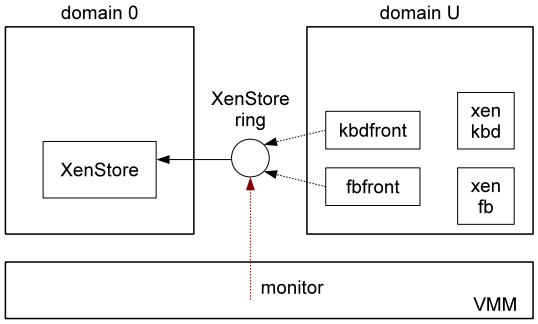


図 6 XenStore リングの監視

4.2.1 VFB の二重化

FBCrypt の VMM は，ドメイン U 起動時に仮想ビデオカードの VFB の二重化を行う．準仮想化 Linux では，ドメイン U のメモリ上に確保された VFB をドメイン 0 と共有している．ドメイン U は起動時に準仮想化ビデオドライバでメモリ上に VFB を割り当てて，VFB のページフレーム番号が格納される xenfb ページを作成する．ドメイン U は VFB をドメイン 0 と共有できるようにするために，xenfb ページをドメイン 0 上の XenStore に登録する．VMM は 4.3 節のようにしてこの通知を横取りし，ドメイン 0 のメモリを新たに確保して VFB の複製を作成する．その後，VMM はドメイン U からの通知を書き換えて，複製した VFB のページフレーム番号をドメイン 0 に渡す．このようにすることで，ドメイン 0 にはドメイン U と異なる VFB を見せることができる．

4.2.2 VFB の同期・暗号化

FBCrypt はドメイン 0 用に複製した VFB を暗号化し，適宜これら 2 つの VFB 間の同期をとる．ドメイン U で画面が書き換えられて VFB が更新されると，準仮想化ビデオドライバは I/O リングを用いて画面の更新領域の情報をドメイン 0 に通知する．この I/O リングの情報も上記の xenfb ページから取得することができる．VMM はこの通知を横取りし，更新領域に該当する VFB の内容を暗号化しながらドメイン 0 用の VFB にコピーする．ドメイン 0 の仮想ビデオドライバが更新領域の通知を受けとると VNC サーバが呼び出される．VNC サーバは暗号化された画面情報を VNC クライアントに送る．VNC クライアントでは受信したデータを復号化してから画面への描画を行う．ビデオ出力が暗号化される処理の流れを**図 5** に示す．

4.2.3 VFB へのアクセス禁止

FBCrypt はキーボード入力を格納するためのの I/O リングと同様に，ドメイン 0 がドメイン U の VFB を直接参照することを禁止する．ドメイン U の VFB には暗号化されていない画面情報が格納されているためである．ドメイン 0 がドメイン U の VFB のメモリページをマップしようと試みた場合，VMM はエラーを返す．キーボード入力処理の場合と異なる点は，画面の更新領域の通知に使用される I/O リングへのアクセスは禁止していないことである．I/O リングで通知される情報は更新領域の範囲のみであり，更新内容は含まれない．よって，I/O リングへのアクセスを禁止しなくともドメイン 0 への情報漏洩の危険はない．

4.2.4 暗号アルゴリズム

VFB の暗号化には RC5 [11] をベースとした暗号アルゴリズムを用いる．一般に，更新領域は任意の矩形となるため，ピクセル単位で暗号化を行えるようにするのが望ましい．RC5 はブロック暗号であるが，Xen の VFB で 1 ピクセルを表現するのに使用されている 32 ビットをブロックサイズとすることができる．しかし，ピクセルごとに同じ鍵を使って暗号化を施した場合，画面に表示されているおおよその内容が推測できてしまう．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．そこで FBCrypt で用いる RC5 には暗号化と鍵生成のアルゴリズムそれぞれに，暗号化を施すピクセルの x 座標と y 座標の要素を追加した．したがって，同じ色情報を持つピクセルでも異なる暗号結果が得られるようになる．

実際には実装上の問題のため，VFB の暗号化は 2 ピクセル単位で行っている．Xen の VNC サーバは VNC クライアントにピクセルデータを送信する際に，ピクセルの 32 ビットのうち色情報が格納されていない上位 8 ビットをクリアしている．このため，32 ビット単位で暗号化を施すと送信時にデータが欠損してしまい，VNC クライアントにおいて正常に復号化できなくなる．しかし，ブロックサイズを 24 ビットにすると安全性の低下が懸念されるため，2 ピクセル分の 48 ビットをブロックサイズとした．

4.3 XenStore リングの監視

準仮想化において，ドメイン U から XenStore に登録される情報を VMM から取得するため，FBCrypt の VMM はドメイン 0 とドメイン U 間の通信に使われる XenStore リングを監視する．XenStore リングは I/O リングと同様のバッファリングである．ドメイン U が XenStore に構成情報を送信する際には，16 バイトのヘッダ情報に続けて XenStore に格納される形式のパス情報とデータを XenStore リングに書き込む．例えば，仮想キーボードの情報を登録する時には，`device/vkbd/0/page-ref` というパス情報と `xenkbd`

ページのマシンフレーム番号が書き込まれる．仮想ビデオカードの情報を登録する時には，device/vfb/0/page-ref というパス情報と xenfb ページのマシンフレーム番号が書き込まれる．FBCrypt の VMM は，ドメイン U が XenStore リングに書き込みを行った後にドメイン 0 に通知するイベントをトリガーにして XenStore リングを逐次監視し，必要となる情報を取得する．

FBCrypt の VMM は XenStore リングを監視するために，ドメイン U 起動時に XenStore リングのアドレスを取得する．XenStore リングのマシンフレーム番号は，ドメイン U 起動時にドメイン 0 からドメイン U に渡される起動情報ページに格納されているが，従来の VMM は起動情報ページを管理していない．そこで，ドメイン U 起動時に仮想 CPU のレジスタにセットされるアドレスから起動情報ページを特定し，XenStore リングのアドレスを取得する．

まず，FBCrypt の VMM は，ドメイン U 起動時に仮想 CPU の RSI レジスタに設定される起動情報ページのアドレスを取得する．このレジスタにセットされているアドレスはドメイン U の仮想アドレスであるため，VMM はこのアドレスをマシンフレーム番号に変換する．通常，ドメイン U の仮想アドレスは疑似物理フレーム番号に変換してから，ドメイン U の持つ P2M テーブルを用いてマシンフレーム番号に変換する．P2M テーブルはドメイン U が扱う疑似物理フレーム番号から VMM が扱うマシンフレーム番号への変換テーブルである．しかし，ドメイン U 起動時には P2M テーブルが作成されていないため，P2M テーブルからマシンフレーム番号を直接求めることはできない．そこで VMM が管理している M2P テーブルを用いて，ドメイン U の全てのページについてマシンフレーム番号から疑似物理フレーム番号に変換していく．そして，疑似物理フレーム番号が一致した時のマシンフレーム番号が起動情報ページのマシンフレーム番号となる．

起動情報ページのマシンフレーム番号を取得したら，そのメモリページを VMM のメモリ空間にマップし，そのページに格納されている XenStore リングのマシンフレーム番号を取得することができる．この情報を用いて XenStore リングの置かれているメモリページをマップすることで，VMM による XenStore リングの監視が可能となる．この手法はドメイン 0 に依存しないためマシンフレーム番号の詐称を防ぐことができる．

5. 実験

FBCrypt により情報漏洩が防止ができていることを確認し，FBCrypt を用いた場合の VNC のレスポンスタイムを測定するための実験を行った．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．VM 用のマシンでは VMM として FBCrypt を実装した Xen 4.1.1 を動作させ，管理 VM とユーザ VM では Linux 2.6.32.21 を動作させた．このマシンは 3.5GB のメモリを搭載しており，ドメイン 0 には 3GB，ドメイン U には 512MB をそれぞれ割り当てた．クライアントマシンでは VNC クライアントとして，FBCrypt を実装した TightVNC Java Viewer 2.0.95 を Linux 2.6.38.8 上で動作させた．

```
[root@Domain0 ~]# tail VNC_KEYLOG2
r o o t  p a s s w a r d
```

図 7 管理 VM からのキーボード入力の盗聴

```
[root@Domain0 ~]# tail VNC_KEYLOGG
_ ? ? ? % w ? $ ? V f  [00 1f]: %
 ? t [00 03] " ? ? | " ? ? ? " [00 16]? ? ?
```

図 8 暗号化されたキーボード入力

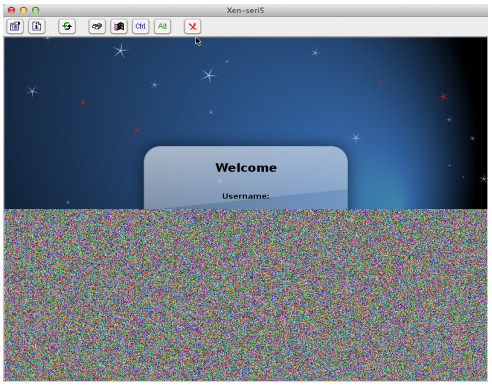


図 9 暗号化されていない画面 (上半分) と暗号化された画面 (下半分)

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なった．このキーロガーは VNC クライアントから送られてきたキーボード入力情報をファイルに保存する．FBCrypt を用いずに帯域外リモート管理を行なった場合，VNC クライアントで入力したキーボード入力は**図 7** のように盗聴され，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 クライアントで行なったキーボード入力は**図 8** のように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なった．このスクリーンキャプチャは，仮想ビデオカードの VFB を一定間隔でファイルに保存する．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 から画面の内容は認

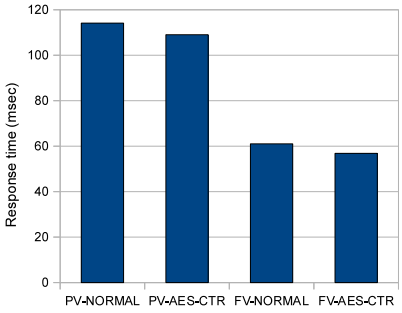


図 10 キーボード入力の平均レスポンスタイム

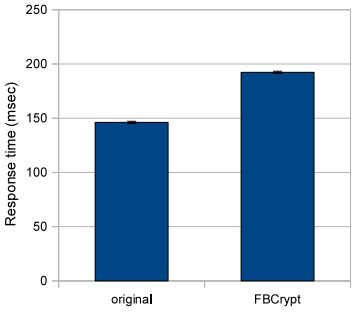
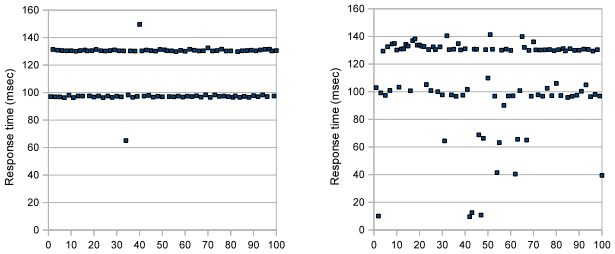
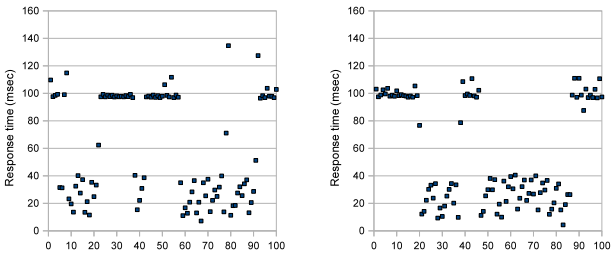


図 13 全画面更新時の平均レスポンスタイム



(a) オリジナル (b) FBCrypt

図 11 キーボード入力のレスポンスタイム（準仮想化）



(a) オリジナル (b) FBCrypt

図 12 キーボード入力のレスポンスタイム（完全仮想化）

識できず，情報が漏洩しないことを確認した．画面下半分のみを暗号化した時の様子を**図 9** に示す．

5.2 VNC 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フレームバッファ変更要求をクライアントが受け取るまでの時間を測定した．キーボード入力を 100 回行った際の実験結果を**図 10** に示す．準仮想化（PV）において， オリジナルと比較すると FBCrypt の方が 5.1ms 速くなった．完全仮想化（FV）においても同様に， FBCrypt の方が 4.2ms 速くなった．キーボード入力を 100 回行った時の個々のレスポンスタイムをプロットしたものを**図 11** と**図 12** に示す．準仮想化ではレスポンスタイムに二極化が見られ，オリジナルはレスポンスタイムのばらつきが少なくほぼ一定であるのに対し，FBCrypt ではばらつきが見られる．完全仮想化においては，オリジナルと FBCrypt 双方にレスポンスタイムの極端な二極化が見られる．これらの二極化やばらつきには，VM のスケジューリングが影響していると考えられる．

次に，全画面更新時（800x600）におけるビデオ出力の暗号化について，FBCrypt とオリジナルでレスポンスタイムの比較を行った．この実験では，ユーザ VM のスクリーンセーバを解除するために VNC クライアントに対してキーボード入力を行なってから，全画面が再描画されるまでの時間を測定した．実験結果を**図 13** に示す．FBCrypt を用いた場合，オリジナルと比較して 46ms の遅延が見られた．全画面更新は画面描画としては最も負荷のかかる処理となるため，46ms の遅延は最悪の場合の値となる．

6. 関連研究

Xoar [12] では VNC サーバ（QEMU）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）[13] 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．

セキュアな実行時環境の研究 [2] や VMCrypt [14] はユーザ VM のメモリやレジスタから管理 VM へ情報が漏洩することを防ぐ．管理 VM がユーザ VM のメモリをマップする際に，VMM がそのメモリ内容を暗号化する．これらのシステムを用いれば，ユーザ VM の I/O リングや VFB

も暗号化されるため，管理 VM に対してアクセス制限を行う必要がなくなる．これらは FBCrypt と同じく，管理 VM を信頼しないという前提で設計されており，FBCrypt と併用することにより管理 VM に対するユーザ VM のセキュリティをさらに向上させることができる．

CloudVisor [7] 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，準仮想化における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．

BitVisor [15] ではゲスト OS のストレージやネットワークの暗号化を VMM が行う．ゲスト OS の I/O を VMM が監視しており，ウイルス感染や USB メモリの盗難，紛失といった事態が発生してもクライアント PC からの情報漏洩を防ぐことができる．BitVisor には管理 VM にあたるものがないため，暗号化・復号化処理は VMM で行われる．そのため，VMM が信頼できれば情報漏洩の危険性はない．しかし BitVisor は帯域外リモート管理を提供していない．

Xen VNC Proxy [16] は管理 VM 経由でユーザ VM を操作するオープンソースツール群である．xvp は Web ブラウザでサーバホストやその上で動作する VM を管理でき，新規に VM を作成したり，VM の起動や停止を行える．xvp の通信には VNC プロトコルを拡張したものが使用されている．

7. まとめ

本稿では，IaaS 環境においても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．FBCrypt は，VNC クライアントと VMM でユーザ VM に対する入出力の暗号化を行い，管理 VM への情報漏洩を防ぐ．FBCrypt を Xen と TightVNC Java Viewer 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，完全仮想化ゲスト OS への対応である．完全仮想化に対応させることにより，Windows など準仮想化 Linux 以外の OS でも安全な帯域外リモート管理が可能となる．現在のところ，キーボード入力の暗号化については対応できているが，入力の完全性のチェック，および，ビデオ出力の暗号化については実装を行っているところである．また，VFB の暗号化にも AES を用いることを検討している．

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] TechSpot News: Google Fired Employees for Breaching User Privacy, `http://www.techspot.com/news/40280-google-fired-employees-for-breaching-user-privacy.html` (2010).

[4] P.Barham, B.Dragovic, K.Fraser, S.Hand, T.Harris, A.Ho, R.Neugebauer, I.Pratt and A.Warfield: Xen and the Art of Virtualization, *In Proc. of the 19th Symposium on Operating Systems Principles*, pp. 164–177 (2003).

[5] TightVNC Group: TightVNC, `http://www.tightvnc.com/`.

[6] rackspace: rackspace, `http://www.rackspace.com`.

[7] 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).

[8] Trusted Computing Group: TPM Main Specification, `http://www.trustedcomputinggroup.org/` (2011).

[9] NIST: Advanced Encryption Standard (AES)., *FIPS Publication 197* (2001).

[10] yassl: yassl - Embedded SSL Library for Applications,Devices,and the Cloud, `http://www.yassl.com/yaSSL/Home.html`.

[11] Rivest, R. L.: The RC5 Encryption Algorithm, *Proc. Workshop Fast Software Encryption* (1994).

[12] 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).

[13] VMware Inc.: VMware vSphere Hypervisor, `http://www.vmware.com/`.

[14] 田所秀和，光来健一， 千葉滋：Preventing Information Leakage from Virtual Machine's Memory in IaaS Clouds, Advanced Computing Systems (ACS) Vol 5 No.4) (2012).

[15] Shinagawa, T., Eiraku, H., Tanimoto, K., Omote, K., Hasegawa, S., Horie, T., Hirano, M., Kourai, K., Oyama, Y., Kawai, E., Kono, K., Chiba, S., Shinjo, Y. and Kato, K.: BitVisor, *Proc. Intl. Conf. Virtual Execution Environments and VEE'09*, pp. 121–130 (2009).

[16] xvp Project: Xen VNC Proxy：Cross-platform VNC-based and Web-based Management for Citrix XenServer and Xen Cloud Platform （参照 2012/6/28）.