IaaS における管理 VM への画面情報漏洩の防止

西村直樹[†1]　江川友寿[†1]　光来健一[†1,†2]

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 を管理する権限をもつ VM（管理 VM）経由でアクセスすることで，ユーザ VM における障害発生時でもリモート管理が可能となる．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，画面情報が他者へ漏洩するリスクが高まる．この問題を解決するために，本稿では管理 VM 経由でリモート管理を行う際にユーザ VM の画面情報の漏洩を防ぐシステム *FBCrypt-V* を提案する．FBCrypt-V はユーザ VM の仮想フレームバッファを VMM 内で二重化して暗号化し，クライアント側で復号する．FBCrypt-V を Xen および TightVNC に実装し，画面情報が管理 VM に漏洩しないことを確認した．

Preventing Information Leakage from Screens via Management VMs in IaaS

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

In IaaS clouds, the users manage their virtual machines (user VMs) remotely. Even at failures on user VMs, they can perform remote management if they access their VMs via privileged VMs called management VMs. However, since management VMs are not always trustworthy in IaaS, screen information of user VMs may leak. To solve this problem, this paper proposes *FBCrypt-V*, which prevents information leakage from screens of the user VMs via the management VMs. FBCrypt-V replicates and encrypts the virtual frame buffers of the user VMs in the VMM and decrypts them in the client sides. We have implemented FBCrypt-V in Xen and TightVNC and confirmed that screen information does not leak.

†1 九州工業大学
†2 独立行政法人科学技術振興機構，CREST

1. はじめに

近年，ネットワークを介してサービスを提供するクラウドコンピューティングの利用が広がっている．Infrastructure as a Service（IaaS）はユーザに仮想マシン（VM）を提供し，ユーザはクラウド上の 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 や SSH などを用いて，ネットワーク経由で VM に直接アクセスすることによって VM 内部のソフトウェアの管理を行う．Desktop as a Service（DaaS）でも同様に，ユーザはネットワーク経由で VM 上のデスクトップ環境にアクセスする．しかし，ネットワークや OS の障害時には VM へのアクセスが不可能になってしまい，以降の VM の管理が困難になってしまう．

提供された VM（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）を経由してアクセスすることが考えられる．しかし，この方法には情報漏洩のリスクが高まるという問題がある．これは IaaS においては管理 VM を十分に信頼できるとは限らないためである．例えば，管理 VM のセキュリティ対策が十分でない場合，攻撃者に侵入される可能性がある．また，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．その結果，管理 VM を経由してユーザ VM の画面情報が盗聴されることもあり得る．

この問題を解決するために，管理 VM 経由でリモート管理を行う際にユーザ VM の画面情報の漏洩を防ぐシステム *FBCrypt-V* を提案する．FBCrypt-V では，ユーザ VM の画面情報が格納されている仮想フレームバッファ（VFB）を VMM が二重化し，管理 VM には暗号化した VFB を参照させる．これら 2 つの VFB の間の同期を VMM が取り，暗号化された VFB の情報はクライアント側で復号する．これにより，管理 VM から取得できる画面情報は常に暗号化されることになり，画面上に表示される機密を守ることができる．FBCrypt-V ではリモートアテステーションを用いることにより VMM を信頼する．

我々は，準仮想化 Linux を対象として FBCrypt-V を Xen 4.1.1[1] および TightVNC Java Viewer 2.0.95[2] に実装した．VMM は起動時に行われるユーザ VM から管理 VM への通信を監視することで，VFB の情報を取得して複製を作成する．ユーザ VM が VFB を更新した時に管理 VM の VNC サーバに送られるリクエストを監視して，2 つの VFB 間の同期をとる．VFB の暗号化には改変を加えた RC5[3] を用いた．FBCrypt-V を用いて実験を行い，管理 VM 経由の VNC 接続時において，ユーザ VM の画面情報が漏洩していないことを確認した．

以下，2 章で IaaS 環境における画面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-V 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-V 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 への画面情報漏洩

IaaS のユーザは，VNC クライアントを使用する場合にはユーザ VM の VNC サーバに接続することでユーザ VM 内部のソフトウェアの管理を行う．この管理方法の問題点は，VM を動作させているマシンが組織外部のクラウドに置かれているため，VNC サーバへの接続ができなくなると VM のリモート管理を行えなくなることである．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アクセス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，接続することができない．

このような状況でもユーザ VM の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 で VNC サーバを動作させることが望ましい．管理 VM とはユーザ VM の起動や停止，マイグレーションなどの制御を行える特権を持った VM のことである．管理 VM 経由でユーザ VM の仮想デバイスを通してユーザ VM にアクセスすると，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操作することができ，より柔軟な VM の管理が可能になる．例えば，ユーザ VM へのネットワーク接続ができなくなってもキーボード入力などを行うことができ，OS の起動時にも起動メッセージを見ることができる．既存の IaaS プロバイダはこのような管理方法を提供していないことが多いが，Xen や VMware ESX サーバでは管理 VM 経由でユーザ VM を管理できるようになっている．

しかし，管理 VM 経由でユーザ VM にアクセスすると情報漏洩のリスクが高まる．クラウド環境においては管理 VM が十分に信頼できるとは限らないためである．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，どのデータセンタで自分の VM が動いているか分からないこともある．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，第三者の攻撃により管理 VM の制御が奪われる恐れがある．またシステム管理者自身に悪意があった場合，管理 VM の中で容易に不正を働くことができる．

攻撃者や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，ユーザ VM の画面情報は容易に盗聴されてしまう．ユーザ VM の画面情報は**図 1** のように，管理 VM の VNC

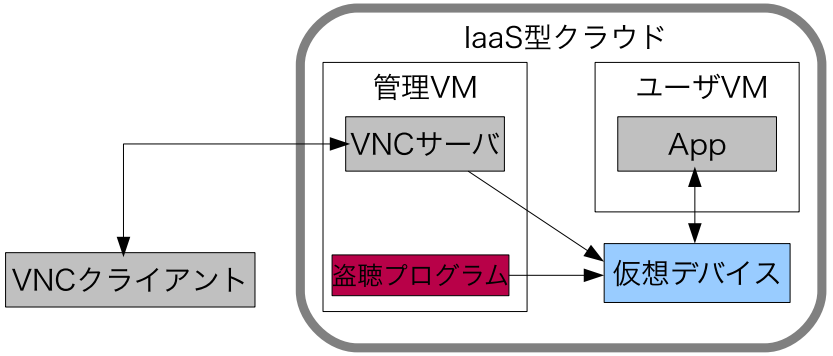

図 1 画面情報の盗聴

サーバが仮想ビデオカードにアクセスするのと同様の方法で取得することができる．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暗号化が可能であるが，仮想デバイスはネットワークとは関係なく直接管理 VM から参照可能であるため，ネットワークの暗号化では漏洩を防ぐことができない．ユーザ VM 上で VNC サーバを動かす場合はユーザ VM と VNC クライアントの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なかった．

ユーザ VM の画面情報が漏洩す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，ユーザが VM 上の設定ファイルにパスワードを直接記述しなければならない場合，画面に表示されたパスワードを盗み見られてしまう恐れがある．また，ユーザが DaaS によって提供される仮想デスクトップ上でオンラインショッピングを行う場合，クレジットカード番号を入力したり，ソフトウェアキーボードを用いて暗証番号を入力したりすることがある．これらの番号は画面を監視していれば容易に盗むことができる．また，ウェブサイトの閲覧履歴やメールの内容等も盗むことが可能である．

3. FBCrypt-V

本稿では，ユーザ VM の画面情報が管理 VM に漏洩するのを防ぐシステム FBCrypt-V を提案する．

3.1 脅威モデル

FBCrypt-V は管理 VM 内で画面情報が盗まれる攻撃を想定している．よって管理 VM および管理 VM を用いてユーザ VM を管理している一般のシステム管理者は信頼しない．

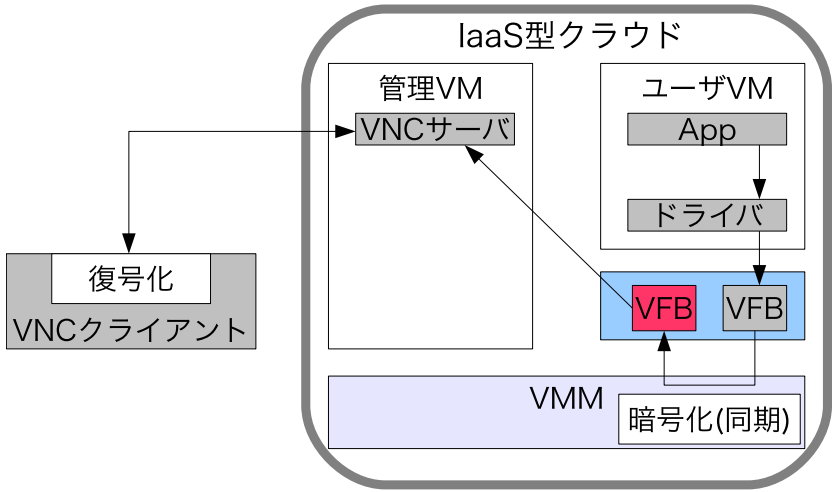

図 2 FBCrypt-V のシステム構成

ただし，攻撃者がユーザ VM に直接侵入して情報を盗む攻撃は想定しない．ユーザ VM はユーザに正しく管理されているとし，パスワードやソフトウェアに脆弱性はないものとする．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．

3.2 FBCrypt-V

FBCrypt-V では**図 2** のように，VMM がユーザ VM の仮想ビデオカードのフレームバッファ（VFB）を二重化し，一方の VFB を暗号化する．ユーザ VM からアクセスされた場合は暗号化されていない VFB を使用し，管理 VM からの場合は暗号化された VFB を使用する．VMM はユーザ VM が VFB に行った更新を検出し，これら 2 つの VFB の間で同期をとる．管理 VM の VNC サーバは暗号化された VFB から取得した画面情報を VNC クライアントに送ることになるため，VNC クライアントでその情報を復号する．このようにして，ユーザは通常通りにユーザ VM の画面情報を取得することができる．

FBCrypt-V を用いることで，管理 VM やユーザ VM に変更を加えることなく管理 VM への画面情報の漏洩を防ぐことができる．管理 VM からは暗号化された VFB にしかアクセスすることができず，暗号化されていない VFB へのアクセスは VMM によって禁止される．VFB を暗号化しても管理 VM の VNC サーバはそのまま動作させることができる．

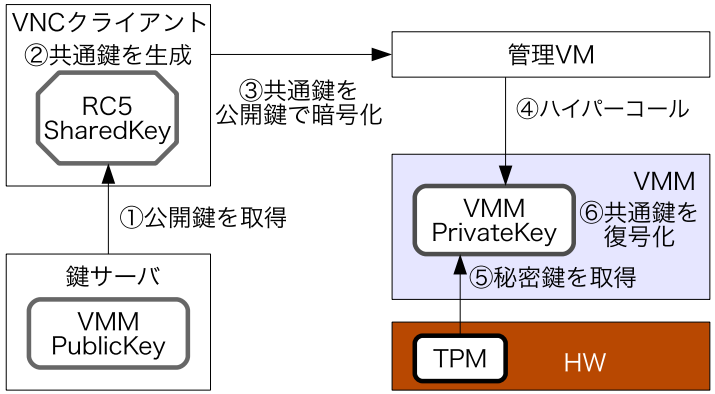

図 3 リモートアテステーションを利用した鍵の共有

VNC サーバは画面の内容を認識せず，暗号化された画面を実際の画面と考えて処理を行うためである．一方，ユーザ VM からは暗号化されていない VFB にアクセスできるため，既存のデバイスドライバをそのまま使うことができる．

FBCrypt-V はリモートアテステーションを用いてクラウド内の VMM を信頼する．リモートアテステーションはマシンの起動時に VMM のハッシュ値を計算し，それを第三者が検証することでコード改ざんの有無を安全に確認する技術である．これはシステム管理者であっても改ざんが不可能な TPM などのハードウェアの機能を使うことで実現される．システムの導入時点での不正な改ざんを防ぐために，VMM のインストールおよびリモートアテステーションの設定はクラウド内でも信頼できる少数の管理者が行う．

画面情報の暗号化と復号化に用いる共通鍵は VNC クライアントと VMM の間で安全に共有させる．リモートアテステーションを利用した鍵の共有の手順を**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 が改ざんされていないこと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 の公開鍵を登録しておく．VNC クライアントは取得した VMM の公開鍵を使っ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 VNC 接続を開始するたびに新しい共通鍵の共有が可能となる．VMM の秘密鍵は TPM によって封印され，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 の実行ファイルを見られても，秘密鍵が漏洩することはない．

4. 実　　装

我々は FBCrypt-V を Xen 4.1.1[1] および TightVNC Java Viewer 2.0.95[2] 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 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 を対象とした．

4.1 画面情報の処理の流れ

FBCrypt-V はドメイン U を起動する時に VFB の二重化を行う．従来の準仮想化 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 を起動すると Linux に組み込まれた準仮想化ビデオドライバ（fbfront）がメモリ上に VFB 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，VFB として確保されたメモリ領域の情報をドメイン 0 に通知する．FBCrypt-V ではこの通知を VMM が横取りし，ドメイン U のメモリを新たに確保して VFB の複製を作成する．そして VMM はドメイン U からの通知を書き換えて，複製した VFB のメモリ領域の情報をドメイン 0 に渡す．このようにすることで，ドメイン 0 がドメイン U と同じ VFB を共有しているかのように見せる．

FBCrypt-V はドメイン 0 用に複製した VFB を暗号化し，これら 2 つの VFB の間の同期を取る．ドメイン U で画面が書き換えられて VFB が更新されると，画面の更新領域の情報がドメイン 0 に逐次通知される．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 VNC サーバが更新領域の通知を受け取ると暗号化された VFB を読み込み，VNC クライアントに暗号化された画面情報を送る．VNC クライアントでは受信したデータを復号してから画面への描画を行う．このようにして，ドメイン U の VFB と VNC クライアントの間で画面情報が暗号される．

4.2 VFB の二重化

FBCrypt では VMM が確実に VFB の二重化を行えるようにするために，VMM が独力で VFB 情報ページを特定する．VFB 情報ページはドメイン U とドメイン 0 の間で共有され，ドメイン U が作成した VFB の情報をドメイン 0 に渡すためのメモリページである．VFB 情報ページの情報はドメイン U の起動時にドメイン 0 の XenStore と呼ばれる簡易データベースに登録される．VMM はドメイン U から XenStore へのアクセスを監視することで VFB 情報ページのページ番号を取得することができる．この特定手法はドメイン 0 に依存しないため VFB 情報ページの詐称を防ぐことができる．また，VFB 情報ページの情報を VMM に登録するようにドメイン U のカーネルを修正する必要もないため，既存のゲスト OS をそのまま利用することができる．

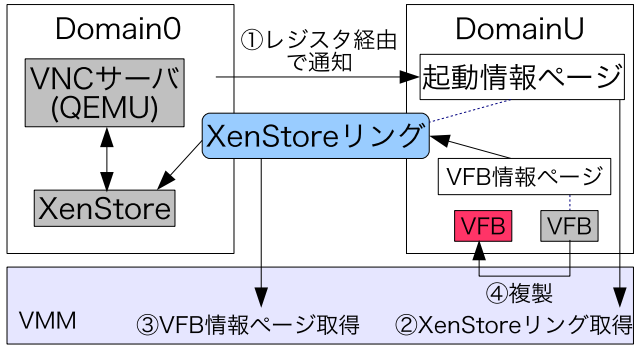

図 4　ドメイン U 起動時の VFB の二重化

ドメイン U から XenStore へのアクセスを監視するために，VMM はドメイン U とドメイン 0 の間の通信路である XenBus を監視する．XenBus は準仮想化環境におけるドメイン間通信のために使われ，XenStore にアクセスする際には XenStore リングと呼ばれる送受信用の 2 つのリングバッファとイベントチャネルが用いられる．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．VFB 情報ページを登録する時には，*device/vfb/0/page-ref* というパス情報と VFB 情報ページのページ番号が書き込まれる．XenBus を使った送信を行う際にはドメイン 0 にイベントが送られるため，VMM はイベント送信のタイミングで XenStore リングの監視を行う．イベントが送信されるたびに VMM は XenStore リングから 1 つの要求を盗み見て，パス名が対象の VFB に関するものであればデータ部から VFB 情報ページのページ番号を取得する．

VMM はドメイン U の起動時にドメイン 0 からドメイン U に渡される起動情報ページから XenStore リングを特定する．起動情報ページには XenStore リングが含まれるメモリページのページ番号が格納されている．起動情報ページはドメイン 0 とドメイン U の間で共有されているが，VMM は認識していなかった．そこで，VMM はドメイン U の起動時

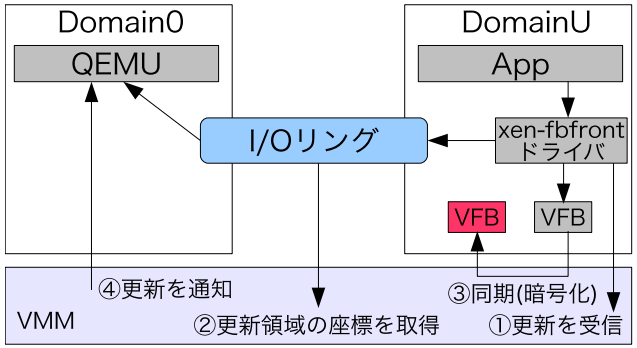

図 5 画面更新時の VFB の同期

に仮想 CPU の RSI レジスタに設定される起動情報ページの仮想アドレスを取得する．そして，この仮想アドレスを起動情報ページのページ番号に変換する．

VMM は VFB 情報ページを取得した後で VFB の複製を作成する．VFB 情報ページ内に VFB として使われているメモリ領域と VFB のサイズが格納されている．VFB 情報ページを使って渡される VFB はページテーブルのような木構造で格納されている．これはドメイン U のカーネル内で連続するメモリ領域に VFB を確保しても，VMM のメモリ上で連続しているとは限らないためである．VFB 情報ページにはページディレクトリの配列が格納されており，ページディレクトリのページの中には VFB を構成している各ページのページ番号が格納されている．VMM はこの格納形式に従って VFB を複製し，複製した VFB を指すように VFB 情報ページのページディレクトリの配列を上書きする．

情報漏洩を防ぐために，FBCrypt-V はドメイン 0 がドメイン U のメモリ上の暗号化されていない VFB を直接参照することを禁止する．ドメイン 0 がドメイン U のメモリを参照するには VMM の機能を利用してメモリマップを行う必要がある．ドメイン 0 が暗号化されていない VFB のメモリページをマップしようとした時には，VMM が対応する暗号化された VFB のページをマップする．この機能は未実装であるが，容易に実装が可能である．

4.3 VFB の同期

FBCrypt-V ではドメイン U において VFB が更新されると，VMM が更新された領域を検出する．VFB が更新された時，ドメイン U の fbfront ドライバは XenBus を使ってドメイン 0 の VNC サーバに更新領域を通知する．XenBus に送信を行うとドメイン U からドメイン 0 にイベントが送られるため，VMM はこのイベントを捕捉することで VFB の更新を検出することができる．この通信には VFB 情報ページの中に確保されている I/O リングと呼ばれるリングバッファが使われる．I/O リングを使って通知されるのは更新領域の範囲だけであり更新内容は含まれないため，ドメイン 0 への情報漏洩の危険はない．

検出した画面領域に対して，VMM は 2 つの VFB 間での同期を行う．更新領域は矩形の左上の座標と矩形のサイズで指定されるため，この矩形に含まれるピクセルデータだけをコピーする．任意の矩形に対応する VFB 上のメモリ領域は一般に不連続となる．しかし，ドメイン U で X サーバを動作させると矩形の左上の x 座標は必ず 0 になり，幅は画面の横幅と等しくなっている．そのため，更新領域の矩形は VFB 上で連続しており，効率よくコピーを行うことができる．

VFB の同期を行う際に，VMM は RC5[3] をベースとした暗号アルゴリズムを用いて更新された領域を暗号化する．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われている 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分かってしまう．これはピクセルが同じ色情報を表すならば暗号化した結果も同じになるためである．そこで，FBCrypt-V で用いる RC5 には暗号化とキー生成のアルゴリズムのそれぞれに，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．こうすることで同じ色情報をもつピクセルでも暗号化した結果が異なるようになる．

FBCrypt-V では実際には 2 ピクセル単位で暗号化を行っている．Xen の VNC サーバは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うため VNC クライアントにおいて復号できなくなる．ブロックサイズを 24 ビットにすると実装が複雑になり，安全性の低下も懸念されるため，2 ピクセル分の 48 ビットをブロックサイズとした．

更新領域が暗号ブロックを部分的にしか含まない場合には，暗号ブロックを完全に含むように I/O リングの更新領域を書き換える．例えば，更新領域の x 座標や幅が奇数の場合に問題となる．更新領域が暗号ブロックの境界にまたがると，暗号ブロックの一部だけが VNC クライアントに送られてしまい，復号することができなくなる．このため，VNC サーバがクライアントに送る更新領域が大きくなってしまう可能性があるが，Xen の現在の実装では更新領域が暗号化を行う 2 ピクセルにまたがることはなかった．

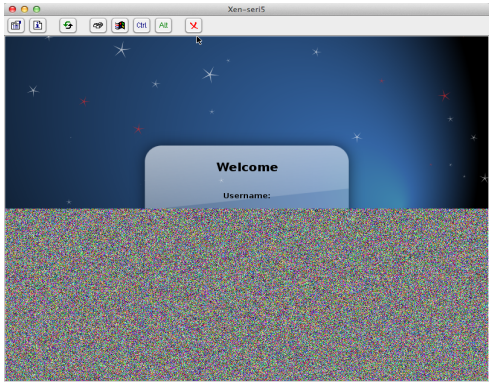

図 6 上半分のみ復号化した画面

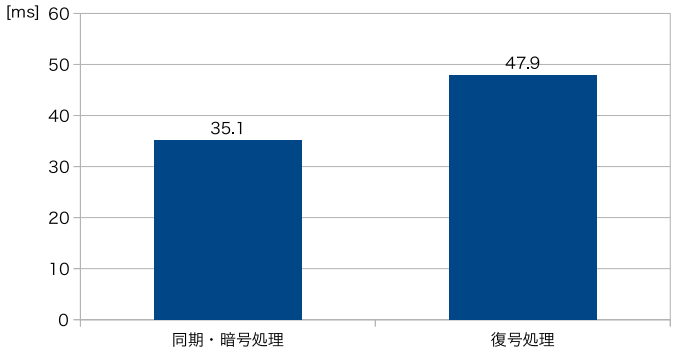

図 7 同期・暗号処理と復号処理にかかる時間

5. 実　　験

実験にはユーザ VM を動かすマシンとして Intel Core i7 870 の CPU，4GB のメモリ，ギガビットイーサネットを搭載したマシンを用い，VNC クライアントを動かすクライアントマシンとして Intel Xeon W3550 の CPU，6GB のメモリ，ギガビットイーサネットを搭載したマシンを用いた．これらのマシンはギガビットイーサネットスイッチを用いて接続した．サーバーマシンでは VMM として FBCrypt-V を実装した Xen 4.1.1，ドメイン 0 とドメイン U の OS には Linux 2.6.39.3 を用いた．また，ドメイン 0 には 2GB，ドメイン U には 1GB のメモリを割り当てた．クライアントマシンでは VNC クライアントとして FBCrypt-V を実装した TightVNC Java Viewer 2.0.95 を Windows 7 上で動作させた．Java ランタイムはバージョン 1.6.0_24 を用いた．

5.1 画面情報の漏洩防止の確認

まず盗聴プログラムを作成して，ドメイン 0 からドメイン U の画面情報が実際に盗めることを確認した．作成した盗聴プログラムは，VNC サーバ内で VFB を扱うコードの途中に VFB のデータを一定間隔でファイルにダンプする．この盗聴プログラムを用いてドメイン U の画面情報を取得できることが確認できた．また，ダンプした画面情報を動画のように連続して表示するアプリケーションを作成し，ユーザの操作内容を追跡できることも確認できた．

FBCrypt-V を実装した VNC クライアントにおいて，デモのために上半分のみを復号化した時の画面を図 6 に示す．FBCrypt-V では VMM で暗号化された VFB を管理 VM が参照する．そのため管理 VM 上の盗聴プログラムからは図 6 の下半分のように何が表示されているのか分からない．ネットワーク上で通信を盗聴されても，同様に画面情報は漏洩しない．以上より，FBCrypt-V によって管理 VM に画面情報を漏洩させることなく，安全にユーザ VM をリモート管理できることが確認できた．

5.2 オーバヘッド

FBCrypt-V の同期・暗号化と復号化によるオーバヘッドを測定する実験を行った．この実験では，スクリーンセーバを用いて解像度 800 × 600 のドメイン U の画面全体を更新する際の同期・暗号処理と復号処理にかかる時間を測定した．暗号化・復号化されるデータ量はおよそ 1.4MB であった．また，RC5 のラウンド回数は 16，キー長は 192 ビットとした．測定結果を図 7 に示す．復号化にかかる時間の方が同期・暗号化に比べて約 13ms 長い．この理由としては，暗号化は VMM 内で C 言語を用いて実装されているのに対して，復号化は TighetVNC 内で Java を用いて実装されていることによる違いが考えられる．

5.3 レスポンスタイム

FBCrypt-V を用いた場合と従来システムを用いた場合について，VNC クライアントにおけるレスポンスタイムの比較を行った．この実験では，ドメイン U 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ってから，全画面が再描画されるまでの時間を測定した．レスポンスタイムの内訳を調べるために，VNC サーバが

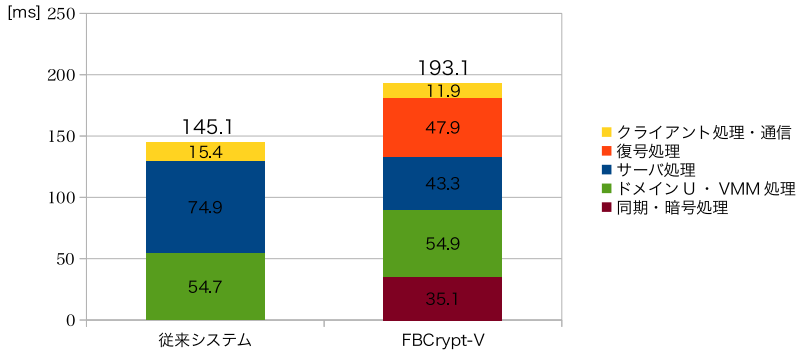

図 8 キーボード入力から画面描画までの処理時間の内訳

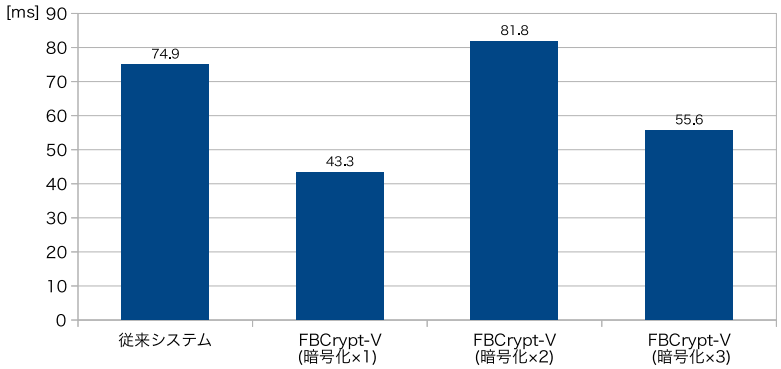

図 9 暗号化の負荷とサーバ処理時間の関係

キーボード入力を受け取ってから画面情報を VNC クライアントに送信し終わるまでの時間，および，キーボード入力をドメイン U に渡してから画面の更新通知を受け取るまでの時間を測定した．測定結果を図 8 に示す．

FBCrypt-V を用いた場合，従来システムに比べておよそ 48ms の遅延が発生した．遅延の主な理由は，同期・暗号化と復号化の処理がキーボード入力から画面描画までの間で行われるためであると考えられる．しかし，この遅延時間は同期・暗号化と復号化の処理時間の合計よりも小さい．これは，VNC サーバでの処理にかかる時間が従来システムより短くなっているためである．

FBCrypt-V を用いた場合にサーバマシンで増加している処理は VMM における同期・暗号処理のみであるため，暗号処理の負荷を変えてサーバ処理にかかる時間の傾向を調べた．暗号処理の負荷を変えるために，FBCrypt-V の暗号処理を 1〜3 回施した場合について測定を行った．実験の結果，暗号処理にかかる時間は暗号化を施した回数に比例して増えたが，サーバ処理にかかる時間は図 9 のように増減した．今回の実験では VNC サーバ内の処理内容は全く同じはずであるため，VM のスケジューリングが 1 つの原因として考えられる．この原因の特定は今後の課題である．

6. 関 連 研 究

FBCrypt[4] は，管理 VM 経由でリモート管理を行う際にユーザ VM へのキーボード入力情報の漏洩を防ぐシステムである．キーボード入力をユーザの VNC クライアントで暗号化し，クラウド側の VMM で復号化を行う．管理 VM で取得できるキーボード入力情報は暗号化されており，復号化したキーボード入力情報には管理 VM からアクセスできないようになっている．ただし，画面情報の暗号化は行われていないため，キーボードで入力した内容が画面に表示された場合やソフトウェアキーボードを使った場合，画面情報からキーボード入力情報が漏洩してしまう．

VMCrypt[5] は，ユーザ VM のメモリから管理 VM への情報漏洩を防ぐシステムである．管理 VM がユーザ VM のメモリページをマップする際に，必要に応じて VMM がユーザ VM のメモリを暗号化することで，メモリの内容を盗み出せないようにする．ユーザ VM と管理 VM が共有するメモリは暗号化しないため，VFB に使われるメモリ領域は暗号化されない．FBCrypt-V と同様に VFB を暗号化することも可能だが，VMCrypt ではページをマップした後は同期をとらないため，VFB への更新が反映されない．

VMware vSphere Hypervisor (ESXi)[6] 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 の管理を行うことができる．管理 VM を経由しないため，管理 VM への画面情報の漏洩は起こらない．しかし，VNC サーバに脆弱性があった場合，画面情報が漏洩する可能性がある．FBCrypt-V では VNC サーバが扱う画面情報は VMM によって暗号化されているため，このような場合でも情報漏洩の危険性はない．また，セキュリティ上，VMM には最小限の機能のみを持たせるべきである．

Xoar[7] 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，VNC サーバが攻撃を受けると画面情報が漏洩する．

BitVisor[8] 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 からの I/O は 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．ただし，画面情報の暗号化は行われていない．

CloudVisor[9] 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，画面情報に関しては考慮されていない．

Xen VNC Proxy（xvp）[10] 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 はウェブ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 を起動・停止したりすることも可能である．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. まとめと今後の課題

本稿では管理 VM 経由でリモート管理を行ってもユーザ VM の画面情報の漏洩を防ぐシステム FBCrypt-V を提案した．FBCrypt-V はユーザ VM の仮想フレームバッファ（VFB）を VMM 内で二重化して暗号化し，VNC クライアントで復号する．そのため管理 VM で画面情報を盗聴されても，データは暗号化されているため情報が漏洩することはない．FBCrypt-V を Xen および TightVNC に実装し，画面情報が管理 VM に漏洩しないことを確認した．

今後の課題としては，より適した暗号方式として，AES の導入を検討している．AES のブロック長は 128 ビットであるため，16 ピクセル単位で暗号化するなどの変更が必要になる．また，VNC クライアントからのキーボードやマウスの入力情報を暗号化する FBCrypt と統合することで，入出力ともに管理 VM 経由のリモート管理における安全性を向上させる．さらに完全仮想化に対応することにより，Windows ゲスト OS でも利用できるようにする．

参 考 文 献

1) 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.
2) TightVNC Group. TightVNC. http://www.tightvnc.com/.
3) RonaldL. Rivest. The RC5 Encryption Algorithm. 2001.
4) 江川友寿, 光来健一. 管理 VM へのキーボード入力情報情報漏洩の防止. 第 118 回 OS 研究会, 2011.
5) 田所秀和, 光来健一, 千葉滋. IaaS 環境における VM のメモリ暗号化による情報漏洩の防止. 第 117 回 OS 研究会, 2011.
6) VMware Inc. VMware vSphere Hypervisor. http://www.vmware.com/.
7) Patrick Colp, Mihir Nanavati, Jun Zhu, William Aiello, George Coker, Tim Deegan, Pete Loscocco, and Andrew Warfield. Breaking Up is Hard to Do: Security and Functionality in a Commodity Hypervisor. *23rd ACM Symposium on Operating Systems Principles (SOSP)*, 2011.
8) Takahiro Shinagawa, Hideki Eiraku, Kouichi Tanimoto, Kazumasa Omote, Shoichi Hasegawa, Takashi Horie, Manabu Hirano, Kenichi Kourai, Yoshihiro Oyama, Eiji Kawai, Kenji Kono, Shigeru Chiba, Yasushi Shinjo, and Kazuhiko Kato. BitVisor. *Proc. Intl. Conf. Virtual Execution Environments and VEE'09*, pp. 121–130, 2009.
9) Fengzhe Zhang, Jin Chen, Haibo Chen, and Binyu Zang. CloudVisor: Retrofitting Protection of Virtual Machines in Multi-tenant Cloud with Nested Virtualization. 2011.
10) xvp Project. Xen VNC Proxy：Cross-platform VNC-based and Web-based Management for Citrix XenServer and Xen Cloud Platform. http://www.xvpsource.org/.