情報処理学会論文誌 Vol.55 No.7 1–8 (July 2014)

仮想シリアルコンソールを用いた VMの安全な帯域外リモート管理

梶原 達也[1] 光来 健一[1]

受付日 2014年7月1日, 採録日 2014年7月xx日

概要：IaaS クラウドでは VM 内のシステム障害時に管理を行うのが難しくなるが，仮想シリアルコンソールを用いた帯域外リモート管理を行うことで障害時においても管理を行えるようになる．しかし，仮想シリアルコンソールを利用するには，管理 VM と呼ばれる別の VM を経由する必要があり，情報漏洩の危険性が高まる．これは，ユーザ端末から仮想シリアルコンソールへのアクセスを中継する管理 VM がクラウドにおいては必ずしも信頼できるとは限らないためである．この問題を解決するために，本稿では仮想シリアルコンソールを暗号化することで安全に帯域外リモート管理を実行できるようにする *SCCrypt* を提案する．SCCrypt は管理 VM に対して暗号化された入出力を行う仮想シリアルコンソールを提供する．管理 VM からの暗号入力は仮想マシンモニタ（VMM）で復号され，ユーザ VM に送られる．ユーザ VM からの出力は VMM で暗号化され，管理 VM に送られる．入力の暗号化および出力の復号はユーザ端末上の管理クライアントで行う．我々は SCCrypt を Xen と OpenSSH に実装し，コンソール入出力が管理 VM に漏洩しないことを確認した．

キーワード：仮想シリアルコンソール，仮想マシン，クラウド，リモート管理，情報漏洩

Secure Out-of-band Remote Management of Virtual Machines Using Virtual Serial Consoles

Tatsuya Kajiwara[1] Kenichi Kourai[1]

Received: July 1, 2014, Accepted: July xx, 2014

Abstract: In Infrastructure-as-a-Service (IaaS) clouds, it is difficult to manage virtual machines (VMs) on system failures inside them, but out-of-band remote management using virtual serial consoles allows users to manage VMs even on such failures. However, since virtual serial consoles are accessed via another VM called the management VM, information leakage can occur via the VM. This is because the management VM is not always trustworthy in IaaS clouds. To solve this security issue, this paper proposes SCCrypt for enabling secure out-of-band remote management by encrypting virtual serial consoles. SCCrypt provides virtual serial consoles whose inputs and outputs are encrypted against the management VM. The encrypted inputs from the management VM are decrypted in the virtual machine monitor (VMM) and are sent to user VMs. The outputs from user VMs are encrypted in the VMM and are sent to the management VM. Inputs and outputs are encrypted and decrypted by users' management clients, respectively. We have implemented SCCrypt in Xen and OpenSSH and confirmed that the console inputs and outputs did not leak to the management VM.

Keywords: Virtual serial console, virtual machine, clouds, remote management, information leakage

1. はじめに

IaaS 型のクラウドサービスはユーザに仮想マシン（VM）を提供する．ユーザは提供された VM（ユーザ VM）を管理するために，SSH などのリモート管理ソフトウェアを用いてアクセスする．ユーザ VM においてネットワークや OS に障害が起こると管理が難しくなるため，仮想シリアルコンソールが提供されている．これはリモート管理ソフ

[1] 九州工業大学
Kyushu Institute of Technology

トウェアを利用して特権を持った特別な VM（管理 VM）を経由してユーザ VM にアクセスする管理方法で帯域外リモート管理と呼ばれる．帯域外リモート管理ではユーザ VM の仮想シリアルデバイスを利用してユーザ VM にアクセスする．したがってユーザ VM 内のネットワーク設定ミス時や OS 障害時であっても管理が可能である．

しかし，クラウドにおいては，管理 VM を経由して仮想シリアルコンソールを利用すると情報漏洩の危険性が高まる [1], [2], [3], [4], [5]．これは，IaaS クラウド内のユーザ VM のコンソール入出力を処理する管理 VM が必ずしも信頼できるとは限らないためである．例えば，管理 VM に脆弱性があった場合，外部の攻撃者がその脆弱性を利用して管理 VM に侵入することが考えられる．また，悪意を持ったクラウド管理者が管理 VM 上で攻撃を行う可能性も考えられる [6]．悪意を持っていない場合でも好奇心の強いクラウド管理者だった場合，本来知ることのできない情報を知るために覗き見ることもあり得る．このような場合には，管理 VM 内にある仮想シリアルコンソールからユーザ VM のリモート管理のためにやりとりされる入出力が容易に盗聴されてしまう．仮想シリアルコンソールを流れるデータは管理 VM の SSH サーバで復号化された後，または，暗号化される前であるため，平文を盗聴することが可能である．例として，ユーザが入力したパスワードや，ユーザ VM で出力されたセキュリティ情報等の機密情報を簡単に盗まれてしまう．

この問題に対し本稿では，仮想シリアルコンソールを暗号化することにより，帯域外リモート管理における管理 VM への情報漏洩を防ぐ *SCCrypt* を提案する．SCCrypt は管理 VM に対して暗号化した入出力を行う仮想シリアルコンソールを提供する．管理 VM による改ざんを防ぐために，管理 VM の下で動作する仮想マシンモニタ（VMM）で暗号入力を復号してユーザ VM に送り，ユーザ VM からの出力を暗号化して管理 VM に送る．暗号化された仮想シリアルコンソールを用いるには，リモート管理クライアントでコンソール入力を暗号化し，出力を復号する．SCCrypt はクラウド内にある VMM の完全性を保証するために，信頼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用いる．

我々は SCCrypt を Xen 4.1.3 およびリモート管理ソフトウェアの OpenSSH[13] のクライアントに実装した．コンソール入力は SSH クライアントで SSH の既存の暗号化とは別に暗号化され，管理 VM から仮想シリアルコンソールを経由してユーザ VM に送られる際に VMM で復号が行われる．ユーザ VM から送られるコンソール出力についても，ユーザ VM から管理 VM に送られる際に VMM を経由させて暗号化を行い，SSH クライアントで復号を行う．SCCrypt を用いた実験を行い，帯域外リモート管理におけるコンソール入出力が管理 VM 内で暗号化されていることと，SCCrypt のオーバー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，クラウドでの仮想シリアルコンソールを用いた帯域外リモート管理における情報漏洩のリスクについて述べる．3 章では，この問題を解決する SCCrypt について述べ，4 章でその実装について述べる．5 章では，SCCrypt を用いた実験について述べる．6 章では関連研究に触れ，7 章で本稿をまとめる．

2. 仮想シリアルコンソールからの情報漏洩

ユーザは，IaaS クラウドによって提供された VM をリモート管理するために，一般的に SSH などのリモート管理ソフトウェアを用いてユーザ VM に接続する．この管理手法は対象となる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．しかし，帯域内リモート管理には対象システム，今回の場合ではユーザ VM の障害に弱いという欠点がある．ユーザ VM 内でネットワークや設定にミスがあった場合，接続を行うことができなくなるからである．また，ユーザ VM の OS 障害によりリモート管理サーバが正常に動作しなくなると，リモート管理が行えなくなる．その結果，ユーザは原因を調べることが困難となり，障害の究明と解決を行う際に問題となる．

このような状態であってもユーザ VM のリモート管理を行えるようにする手法として，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がある．管理 VM とは，ホスト内の全てのユーザ VM にアクセスする特権を持つ VM であり，ハードウェア上で直接動作しているハイパーバイザ型の VMM（Xen など）で提供されている．管理 VM は，ユーザ VM に提供されている仮想デバイスのエミュレーションを行う．例として，SSH クライアントから帯域外リモート管理を行う場合，管理 VM 内にある SSH サーバは，ユーザ VM の仮想シリアルデバイスに直接アクセスすることにより，ユーザ VM の状態に依存しない管理が行える．帯域外リモート管理において，ユーザは障害時のユーザ VM にもローカルコンソールを用いてログインしているかのようにリモート管理が行える．したがって，ユーザ VM のネットワーク設定にミスがあった場合には仮想シリアルコンソールを経由して設定ファイルの修正が行える．

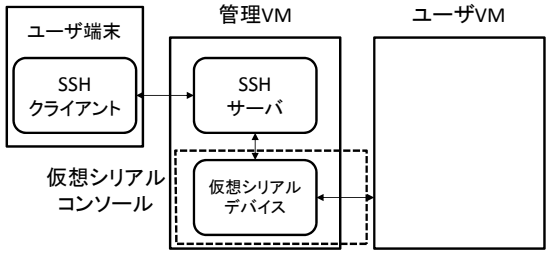

図 1 仮想シリアルコンソールを用いた帯域外リモート管理

しかし，IaaS クラウドにおいて，管理 VM を経由する帯域外リモート管理は情報漏洩のリスクを増加させる可能性がある．それは，クラウド内の管理 VM が十分に信用できるとは限らないからである [1], [2], [3], [4], [5]．IaaS クラウド上の VM はデータセンタの障害などにより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る．その結果，移動先のデータセンタにセキュリティ意識の低いシステム管理者がいることも考えられる．このような環境においては，管理 VM のセキュリティが十分でない懸念があり，外部の攻撃者により管理 VM が攻撃され，制御が奪われる可能性がある．また，外部からの攻撃だけでなくシステム管理者に悪意がある場合，管理 VM を容易に改ざんすることができる [6]．悪意がない場合であっても，帯域外リモート管理による VM 操作の通信内容に興味を持つ，詮索好きなシステム管理者（Honest-but-curious Administrators）の場合には，管理 VM 内で情報収集を行う恐れがある．

外部の攻撃者や，IaaS クラウド内部のシステム管理者によって，管理 VM の権限が悪用されると，帯域外リモート管理による入出力情報が容易に盗聴されてしまう．SSH を用いた帯域外リモート管理では，ユーザ端末内の SSH クライアントと管理 VM 内の SSH サーバの間の通信は暗号化される．しかし，仮想シリアルコンソールへの入力は SSH サーバで復号されるため，管理 VM 内で入力情報が漏洩してしまうのを防げない．同様に，仮想シリアルコンソールからの出力は SSH サーバで暗号化されるため，SSH による暗号化では管理 VM から保護することができない．ユーザ VM 内の SSH サーバと直接通信する帯域内リモート管理では，SSH クライアントとユーザ VM 間が暗号化されているため，このような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いて，ユーザ VM へのコンソール入力は管理 VM 内の SSH サーバや仮想シリアルデバイスを改ざんすることで容易に盗聴ができる．コンソール入力は SSH サーバ経由で仮想シリアルデバイスに送られるため，管理 VM ではログインパスワードのような第三者に知られてはいけない情報を盗聴することができる．また，仮想シリアルコンソールを用いてユーザ VM のセキュリティの設定を行った場合には，コンソール出力が仮想シリアルデバイスと SSH サーバを経由して SSH クライアントに送られるため，攻撃者にセキュリティ設定の詳細を知られてしまう．攻撃者にセキュリティ情報やログインパスワードを知られると，ユーザ VM において外部の攻撃に対し万全なセキュリティ対策をしていても容易にユーザ VM に侵入されてしまう．

3. SCCrypt

本稿では，管理 VM への情報漏洩を防ぐために，仮想シリアルコンソールを暗号化することにより安全な帯域外リモート管理を可能にするシステム *SCCrypt* を提案する．

3.1 脅威モデル

SCCrypt は，外部の攻撃者や悪意をもった IaaS クラウド管理者によって管理 VM が攻撃を受ける状況や悪意はないが詮索好きな管理者が管理 VM で情報収集を行う状況を想定している．本稿では，SSH クライアントから仮想シリアルコンソールを利用する際の入出力情報が管理 VM 上で盗聴されることに焦点を当てる．

SCCrypt では既存研究 [1], [2], [3], [4], [5] と同様に，IaaS クラウドのプロバイダ自体は信頼できるものと仮定している．VMM やハードウェアの管理責任を持つ少数の管理者は信頼するが，日常的に管理 VM でユーザ VM を管理し，悪意や強い好奇心をもっている可能性がある一般のシステム管理者は信頼しない．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．これによって VMM は正常なメンテナンスがされており，脆弱性がないものとする．また，VM が稼働しているデータセンタのサーバルームは物理的に厳重に守られているため，ユーザ VM が動作するハードウェアに物理的にアクセスする攻撃は想定しない．

3.2 SCCrypt

SCCrypt は，管理 VM に対して入出力を暗号化する仮想シリアルコンソールを提供する．この仮想シリアルコンソールは管理 VM から暗号化済みのコンソール入力を受け取り，それを復号してユーザ VM に送る．また，ユーザ VM から暗号化されていないコンソール出力を受け取り，それを暗号化して管理 VM に送る．これにより，攻撃者が管理 VM 内の SSH サーバや仮想シリアルデバイスを不正に改ざんし盗聴を行ったとしても，管理 VM 内では入出力が暗号化され情報漏洩を防ぐことができる．図 2 に SCCrypt の構成を示す．

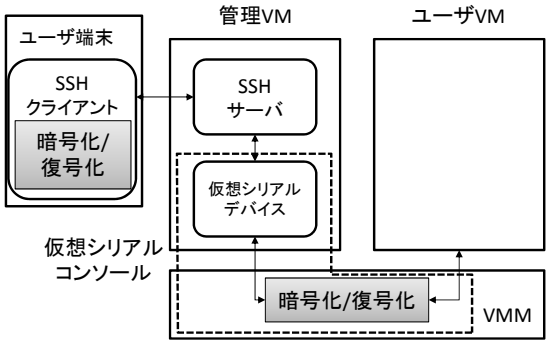

図 2 SCCrypt のシステム構成

管理 VM で復号後もしくは暗号化前のコンソール入出力が盗聴されるのを防ぐために，SCCrypt では管理 VM の下で動作する VMM で入出力の復号化および暗号化を行う．そして，VMM とユーザ VM が入出力を直接やりとり

する．管理 VM 内の仮想シリアルデバイスがコンソール入力を受け取った時は，VMM を経由してユーザ VM に入力が送られる．ユーザ VM からのコンソール出力も VMM を経由して管理 VM 内の仮想シリアルデバイスに送られる．この際に，ユーザ VM のゲスト OS には従来と同じインタフェースを提供するため，ゲスト OS への修正は不要である．このことはクラウド管理者とユーザ VM の管理者が異なるクラウドでは重要である．

SCCrypt では，ユーザはリモート管理クライアントで管理 VM にアクセスして暗号化された仮想シリアルコンソールに接続する．管理 VM にアクセスするためのインタフェースとしては SSH や Web サービスなどが考えられるが，ここでは SSH を用いる場合について説明する．SSH クライアントにおいて，SSH 本来の機能とは別に SCCrypt 用の暗号化を行った上で SSH 本来の暗号化を行う．そのため，SSH クライアントから SSH サーバの間は二重に暗号化された状態で通信が行われる．SSH サーバでは SSH 本来の機能で暗号化されたデータが復号された後，暗号化されたコンソール入力が仮想シリアルコンソールに送られる．ユーザ VM からのコンソール出力は仮想シリアルコンソールで暗号化され，SSH サーバを経由して SSH クライアントに送信される．SSH クライアントは出力を受けとると，まず SSH 本来の復号を行い，次に仮想シリアルコンソールで暗号化されたコンソール出力の復号を行う．

3.3 VMM の完全性

VMM でコンソール入出力の暗号化を行うには，VMM が改ざんされていないことを保証する必要がある．SCCrypt では，IaaS クラウドの外部に信頼できる検証サーバを用意してリモート・アテステーションを用いることで，IaaS 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，耐タンパ性ハードウェア（TPM）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 [7]．SCCrypt は外部から改ざんできない 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，事前に登録されていたハッシュ値と照合して VMM の完全性を検証する．VMM のリモート・アテステーションの設定は，IaaS クラウド内でも信頼できる少数の管理者が行う．

3.4 鍵管理

SCCrypt では，仮想シリアルコンソールへの接続時にコンソール入出力の暗号化・復号化に用いるセッション鍵を SSH クライアントと VMM の間で安全に共有する．まず，SSH クライアントは信頼できる鍵サーバから仮想シリアルコンソールを提供する VMM の公開鍵を取得する．鍵サーバにはあらかじめ信頼できる IaaS クラウド管理者によって VMM の公開鍵が登録されているものとする．SSH クライアントは取得した公開鍵を用いてセッション鍵を暗号化して管理 VM に送信する．管理 VM は暗号化されたセッション鍵を VMM に渡し，VMM では自身の秘密鍵を用いてセッション鍵を復号する．これによって SSH セッション毎に SSH クライアントと VMM の間で新しいセッション鍵を共有することができる．

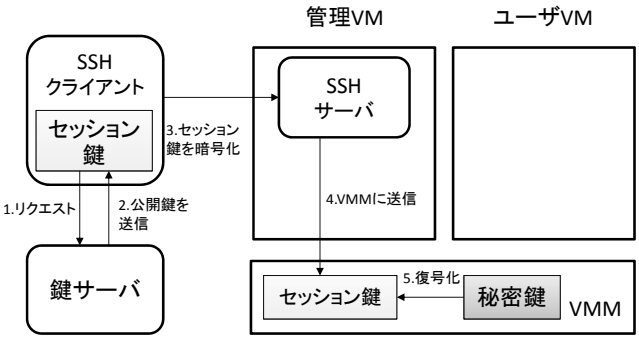

図 3 SCCrypt における鍵共有

4. 実装

我々は SCCrypt を Xen 4.1.3 および OpenSSH 6.0p1[13] に実装した．Xen において管理 VM はドメイン 0 と呼ばれ，ユーザ VM はドメイン U と呼ばれる．仮想シリアルデバイスはドメイン 0 内の QEMU によって提供される．SCCrypt は準仮想化のゲスト OS に対応している．

4.1 コンソール入力の暗号化・復号化

4.1.1 仮想シリアルコンソールの入力処理

従来の仮想シリアルコンソールを用いた帯域外リモート管理における入力処理の流れを示す，まずユーザは SSH などを用いてドメイン 0 に接続する．ドメイン 0 で Xen の管理ツールの一つである xenconsole を実行すると，ドメイン U の仮想シリアルコンソールに接続される．仮想シリアルコンソールへの入力は，QEMU 内で動作している仮想シリアルデバイスに送られる．仮想シリアルデバイスはドメイン U 内のメモリ上に作られたコンソールリングと呼ばれるリングバッファをドメイン U と共有しており，そこにコンソール入力を書き込む．ドメイン U のデバイスドライバはコンソールリングに書き込まれた入力を取得してシリアルコンソールの入力として処理を行う．

4.1.2 VMM による入力の復号

SCCrypt では，暗号化されたコンソール入力がドメイン 0 内の仮想シリアルデバイスに送られる．このときに，SCCrypt の仮想シリアルデバイスはドメイン U のコンソールリングに入力を直接書き込むのではなく，新しく追加したハイパーコールを用いて VMM に入力を渡す．ハイパーコールは VMM を呼び出すための仕組みである．VMM は

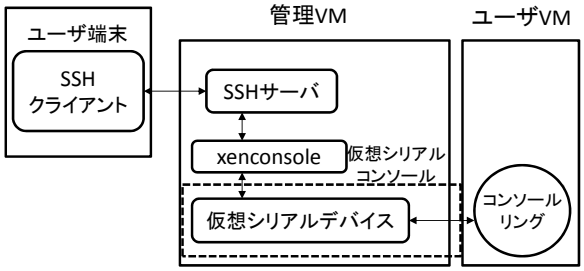

図 4 仮想シリアルコンソールの入出力処理

渡されたコンソール入力を復号し，仮想シリアルデバイスに代わってドメイン U のコンソールリングに書き込む．ドメイン U は従来通り，コンソールリングに書き込まれた入力を取得して処理するだけでよいため，ドメイン U の OS に対する修正は不要である．これによりドメイン U が取得する直前までコンソール入力を暗号化することができる．

ドメイン 0 はドメイン U のメモリにアクセスする権限を持っているため，ドメイン U のメモリをドメイン 0 から保護する手法を組み合わせて用いる．コンソールリングのメモリページのマップを禁止することで，復号されたコンソール入力を盗み見られないようにすることができる．さらには，VMCrypt [5] を用いてドメイン U のメモリ全体を暗号化することで，コンソールリングから取り出された入力も盗み見られないようにすることができる．

4.1.3 SSH クライアントにおける暗号化

暗号化された仮想シリアルコンソールを用いるために，SSH クライアントで入力を暗号化する．入力の暗号化にはストリーム暗号として RC4 を用いる．RC4 では暗号鍵を基に発生させた疑似乱数列と平文の排他的論理和を取ることで暗号化を行う．ストリーム暗号を用いることで，同じ入力に対しても毎回異なる暗号結果となるためリプレイ攻撃にも対処が可能である．RC4 を用いた暗号化は実装中であるため，現在は固定鍵との簡易な排他的論理和を用いている．

SSH クライアントで入力を暗号化することによって二つの問題が生じる．一つは，ドメイン 0 にログインした後で xenconsole コマンドを実行する際に，仮想シリアルコンソールへの入力だけでなく，ドメイン 0 のコマンドラインに入力した文字列も暗号化されてしまうことである．SSH クライアントはドメイン 0 への入力と仮想シリアルコンソールへの入力を区別することはできないため，すべての入力を暗号化するしかない．この問題を解決するために，SSH の機能であるリモートコマンド実行機能を利用する．リモートコマンド実行機能は SSH を用いた接続を行う際に，接続先で実行するコマンドを送信する機能である．この機能を用いることにより，仮想シリアルコンソールに接続するコマンドを仮想シリアルコンソールへの入力とは区別して送信することができる．SSH クライアントはリモートコマンドについては暗号化せずに送信する．

もう一つも上の問題と似ているが，xenconsole コマンドを管理者権限で実行するために必要なパスワード入力が暗号化されてしまうという問題がある．この問題を解決するために，ドメイン 0 において xenconsole コマンドを実行する際にだけパスワードなしで管理者権限を与えるように sudo を設定することで対処した．

```
ssh -t user@192.168.0.70 sudo
/usr/lib64/xen/bin/xenconsole vm1
```

図 5 SSH のリモートコマンド実行を用いた仮想シリアルコンソールへの接続

4.2 コンソール出力の暗号化・復号化

4.2.1 仮想シリアルコンソールの出力処理

従来の仮想シリアルコンソールを用いた帯域外リモート管理における出力処理の流れは，上述した入力処理と逆の順で行われる．ドメイン U から仮想シリアルコンソールに送られるコンソール出力はドメイン U 内の出力用のコンソールリングに書き込まれる．ドメイン 0 内の QEMU の仮想シリアルデバイスはコンソールリングから出力を取得してそのデータを xenconsole に送る．xenconsole はコンソール出力を SSH などを経由してクライアントに送信する．

4.2.2 VMM による出力の暗号化

SCCrypt では，QEMU の仮想シリアルデバイスがコンソールリングから出力を直接受け取る代わりに，ハイパーコールを用いて VMM を呼び出し，VMM からコンソール出力を取得する．呼び出された VMM はコンソールリングから出力を取得して暗号化し，仮想シリアルデバイスに返す．仮想シリアルデバイスは暗号化されたコンソール出力をそのまま xenconsole に送る．

4.2.3 SSH クライアントにおける復号

暗号化された仮想シリアルコンソールを用いるために，SSH クライアントは受信した出力を復号して表示する．仮想シリアルコンソールに接続するための xenconsole コマンドや管理者権限を取得するための sudo コマンドがエラーメッセージを表示した場合に，それらのメッセージは暗号化されていないにも関わらず，復号されてしまうという問題がある．この問題は通常時の動作に支障はないが，正しくエラーメッセージを表示する方法については今後の課題である．

4.3 VMM によるコンソールリングの特定

SCCrypt では，VMM はドメイン U の起動時にコンソールリングのアドレスを特定する．従来の VMM はドメイン U のコンソールリングを認識していないため，復号した入力をコンソールリングに書き込むことも，コンソールリン

グから出力を取得して暗号化することもできない．Xen では，ドメイン U が起動した時に，ドメイン 0 がドメイン U にコンソールリングのアドレスを通知している．そこで VMM はこの通知を監視することにより，ドメイン U のコンソールリングのアドレスを取得する．

5. 実験

SCCrypt によりドメイン 0 での情報漏えいを防止できていることを確認し，SCCrypt によるオーバーヘッドを測定するための実験を行った．VM を動作させるサーバマシンには，Intel Core i7 870 2.93GHz の CPU，4GB のメモリを搭載した PC を用い，クライアントマシンには，Intel Xeon E3-1270 3.40GHz の CPU，8GB のメモリを搭載した PC を用いた．これらのマシンはギガビットイーサネットで接続した．またドメイン U には仮想 CPU を 8 個，メモリを 1024MB 割り当てた．サーバマシンでは VMM として SCCrypt を実装した Xen 4.1.3 を動作させた．ユーザマシンの OS，ドメイン U のゲスト OS には Linux 3.2.0.63 を用い，リモート管理クライアントとして SCCrypt を実装した OpenSSH 6.0p1，リモート管理サーバとして OpenSSH 5.9p1 を用いた．

5.1 情報漏洩の確認

ドメイン 0 上の QEMU の仮想シリアルデバイスにキーロガーを組み込みログファイルを作成させて，仮想シリアルコンソールを使ったドメイン U への入出力の盗聴を行った．SCCrypt が実装されていないオリジナルの Xen では，SSH クライアントで行ったドメイン U へのログイン名とパスワードが平文のままログファイルに記録された．その一方で，SCCrypt が提供する仮想シリアルコンソールではログファイルに SSH クライアントが入力した文字列が暗号化されて文字化けの状態で記録され，入力情報が漏洩していないことを確認した．コンソール出力についても，オリジナルの Xen では全ての出力を平文のまま取得することができたのに対し，SCCrypt では意味をなさない文字列が取得され，暗号化されていることを確認した．

5.2 応答時間

仮想シリアルコンソールへの入力一回当たりの平均応答時間を測定し，オリジナルの Xen と SCCrypt とで比較した．SSH クライアントでコンソール入力を行い，ドメイン U でのエコーバックがコンソール出力として表示されるまでの時間を応答時間とした．100 回測定を行った結果を図 6 に示す．オリジナルの Xen と比較して SCCrypt では応答時間が平均で 0.35 ミリ秒が増加した．現時点の実装では固定鍵との簡易な排他的論理和を用いているため，このオーバヘッドのほとんどは VMM を呼び出すためのハイパーコールによるものと考えられる．RC4 を用いた暗号化は実装中であるが，RC4 による 1 バイトの暗号化には 1 マイクロ秒もかからないため，オーバヘッドはほとんど増加しないことと考えられる．この結果より，SCCrypt によるオーバーヘッドは実際の運用では支障の出るレベルではないことがわかった．

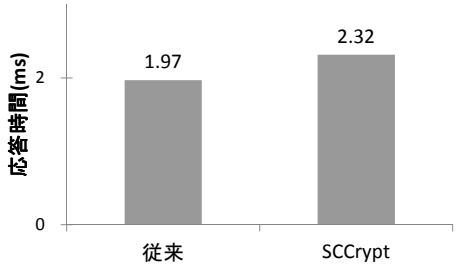

図 6 応答時間の測定結果

5.3 スループット

仮想シリアルコンソールからの出力のスループットを測定し，オリジナルの Xen と SCCrypt とで比較した．cat コマンドで 1000 万文字 (バイト) のテキストファイルを表示させて，1 秒当たりの表示文字数をスループットとした．測定を行った結果を図 7 に示す．僅かな差はあるものの，ほぼ同程度のスループットとなっている．オリジナルよりも SCCrypt の方が僅かによいのは，SCCrypt のオーバヘッドのために SCCrypt の方がコンソールリングにより多くのデータがたまるためであると考えられる．その結果，一度により多くの文字を処理できることになりオーバヘッドが削減される．

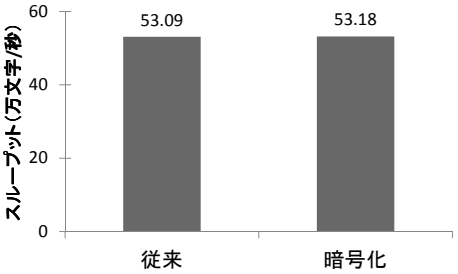

図 7 スループットの測定結果

6. 関連研究

FBCrypt[8] は VNC を用いた帯域外リモート管理において，VNC クライアントと VMM の間で入出力を暗号化することで情報漏えいを防止する．VNC クライアントでキーボードやマウス入力を暗号化して管理 VM 上の VNC サーバに送り，VMM を用いて復号してからユーザ VM に送る．ユーザ VM からのビデオ出力については仮想ビデオカードの VRAM を VMM が暗号化し，VNC サーバ経由で送られた画面情報を VNC クライアントが復号する．SCCrypt

はFBCryptと似ているが，暗号化された入出力を扱うインタフェースを管理VMに提供している点が異なる．そのため，リモート管理ソフトウェアにはSSH以外を用いることもできる．FBCryptではVNCのクライアント・サーバを用いることが前提となっている．また，FBCryptは更新頻度の高いビデオ出力を暗号化するため，オーバヘッドが増加しやすい．その上，ビデオ出力を暗号化していてもマウスカーソルの動きに応じて画面の対応する箇所が変化するため，ユーザVMへの入力が特定される恐れがある．

Xoar[9]は従来，管理VM内で動作していた仮想シリアルデバイスを専用のコンソールVMで動作させる．Xenでは従来より，スタブドメインと呼ばれるVMで仮想シリアルデバイスを動作させることができる．コンソールVMではmini-OSという小さなOSを動作させることで最低限の機能だけを提供することができ，攻撃を受ける可能性を抑えることができる．しかし，仮想シリアルデバイスが攻撃を受けるとリモート管理に伴う入出力情報が漏えいしてしまう．また，IaaSクラウド管理者による内部からの攻撃については考慮されていない．

VMware vSphere Hypervisor (ESXi)[10]ではVNCサーバをVMM内で動作させることで，VNC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．これにより，リモート管理による管理VMへの入出力情報の漏えいを防ぐことができる．しかし，VNCサーバに脆弱性があった場合，VNCサーバが攻撃を受けるとVMM自体に攻撃の影響が及ぶ可能性があり，入出力情報の漏えいが考えられる．

IPMIやIntel AMTが提供するシリアルコンソールは，専用ハードウェアを経由することで帯域外リモート管理を行うことができる．そのため悪意のある管理者であっても入出力の盗聴を行うのは難しい．これらは物理マシン用の管理インタフェースであるが，VM用にはOpenIPMIのlanservシミュレータ[11]やvAMT[14]などが開発されている．現時点では実装されていないが，VMに対して仮想シリアルコンソールを提供することも可能である．しかし，これらはソフトウェアで実現されているため，管理VMへの情報漏洩を防ぐのは難しい．

OpenStackのウェブベース・シリアルコンソール[12]は，ユーザのインスタンスへのCUIを用いた帯域外リモート管理を行うことを可能にしている．Ajaxtermを用いることで，ブラウザ経由で仮想シリアルコンソールにアクセスできる．SCCryptの現在の実装ではリモート管理ツールにSSHを用いているが，Ajaxtermを用いることも可能である．

7. まとめ

本稿では，IaaSクラウドにおいて仮想シリアルコンソールを用いた安全な帯域外リモート管理を可能にするSCCryptを提案した．SCCryptは，帯域外リモート管理において管理VMを経由した情報漏えいを防止するために，管理VMに対して暗号化された仮想シリアルコンソールを提供する．管理VMからの暗号入力はVMMで安全に復号され，ユーザVMに送られる．ユーザVMからの出力はVMMで暗号化され，管理VMに送られる．入力の暗号化および出力の復号はリモート管理クライアントで行う．我々はSCCryptをXenとOpenSSHに実装し，コンソール入出力が漏えいしないことを確認した．

今後の課題は，固定鍵との簡易な排他的論理和による暗号化をRC4を用いた暗号化に置き換えることである．また，現在は準仮想化のユーザVMに対してのみの実装となっているため，完全仮想化のVMに対応できるようにする予定である．完全仮想化ではユーザVMから仮想シリアルデバイスへのアクセス方法が異なるため，準仮想化とは異なる実装を行う必要がある．さらに，SSH以外のリモート管理ソフトウェアにSCCryptを適用することも今後の課題である．

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 Multitenant Cloud with Nested Virtualization, Proc. Symp. Operating Systems Principles, pp. 203-216 (2011).

[4] 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). 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] 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).

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

[8] T. Egawa, N. Nishimura, and K. Kourai, Dependable and Secure Remote Management in IaaS Clouds, Proc. CloudCom 2012, (2012)

[9] 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).

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

[11] vAMT や OpenIPMI の lanserv シミュレータ, http://sourceforge.net/projects/openipmi/
[12] Open Stack Web Based Serial Console, https://wiki.openstack.org/wiki/WebBasedSerialConsole
[13] The OpenBSD Project: OpenSSH, http://www.openssh.org/
[14] 大薗弘記, 光来健一：仮想マシンと物理マシンの一元管理を可能にする仮想 AMT, 第 128 回 OS 研究会, (2014).