IaaS クラウドの帯域外リモート管理における信頼性の向上

2012SE260 富山翔太　2012SE263 柘植光将
指導教員：宮澤元

1　はじめに

IaaS(Infrastructure as a Service) クラウドとは，コンピュータハードウェアなどの計算基盤をクライアントに提供するクラウドコンピューティングの形態である．そのクラウドのセキュリティに関しては，外部からの侵入だけでなく内部犯行による情報漏洩を防ぐということが求められる．IaaS クラウドにおいてセキュリティを確保する従来のアプローチとして，システム管理の際の通信方法に関する帯域外リモート管理の改良についての研究 [3] が存在する．帯域外リモート管理とは，管理 VM（Virtual Machine）を介してユーザ VM にアクセスする通信方法である．管理 VM を経由して通信するので，仮想ネットワークの設定ミスなどに強いという利点があるが，管理 VM に悪意がある場合，管理 VM での盗聴を防ぐことができない．文献 [3] では，管理 VM からの盗聴を防ぐことはできているが，VMM（Virtual Machine Monitor）からの盗聴のリスクは排除できていない．

本研究では帯域外リモート管理において，VMM 管理者が信頼できない場合にも安全な通信を行うことができる手法を提案する．具体的には，帯域外リモート管理の通信において，鍵管理を改良し，VMM からの盗聴を防いで安全な通信を実現する．これにより，ユーザ VM 上のメモリを保護しつつ管理 VM だけでなく VMM 自体も完全に信頼することなく，情報漏洩を防ぐことができる．

2　研究の背景

通常，IaaS クラウドでは，ユーザと仮想マシンがシステム管理に際して通信を行う場合は帯域内リモート管理を利用する．これはネットワーク帯域内で SSH(Secure SHell) クライアントとユーザ VM が直接的に通信を行うものである．しかし帯域内リモート管理はユーザ VM の仮想ネットワークの設定ミスなどが発生してしまうと通信が出来なくなるという欠点がある．それに対し帯域外リモート管理は，SSH クライアントがユーザ VM と直接通信を行わない．管理 VM を通してユーザ VM の仮想シリアルコンソールを用いてユーザ VM にアクセスする通信方式を取るので仮想ネットワークの設定ミスの場合などでも通信できる．

2.1　SCCrypt

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システムである [3]．SSH クライアントとユーザ VM が管理 VM を通じて帯域外リモート管理を行う際，仮想シリアルコンソールを介した通信を VMM 内で暗号化することにより管理 VM からの盗聴を防ぐという手法である．この際，セッション鍵を暗号化・復号化する公開鍵と秘密鍵は VMM 上に配置し，SSH ユーザと VMM 上で暗号化・復号化を行うことで帯域外リモート管理の通信におけるセキュリティを向上させている（図 1）．

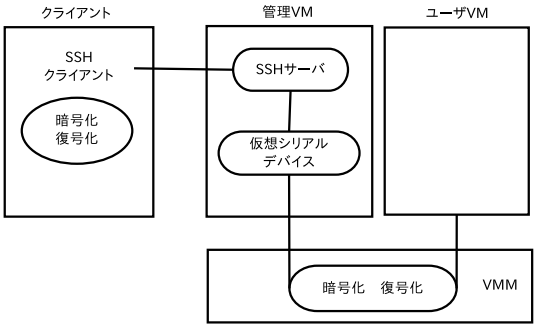

図 1　SCCrypt における帯域外リモート管理の概念図

2.2　SCCrypt の問題点

SCCrypt で解決できるセキュリティ問題は，「帯域外リモート管理における管理 VM からの盗聴」である．一方，SCCrypt では，VMM の管理者からの盗聴，鍵の窃盗などを防ぐことができない．SCCrypt では VMM は完全に信頼して良いという前提をおいているが，リモートアテステーション [1][2] などで VMM 本体が改変されていないことは保障できても，VMM 管理者の悪意による操作を防ぐことはできない．

なお，我々の研究では SCCrypt の前提を引き継ぎ，さらに以下のように前提環境を定める．管理 VM に加え，VMM の管理者も信頼しないようにする．また，ユーザ VM 上に実装を行う際には，ネットワークの設定ミスで通信できなくなった場合，つまり OS の起動後の通信を想定し，起動前の初期設定の通信の保護は考慮しないものとする．

3　鍵管理の改良

本研究では 2.2 節の問題点を解決し，より安全な帯域外リモート管理を実現することを目的としている．本節では，新しい鍵管理方法を提案する．

3.1　鍵管理をユーザ VM で行う仮想シリアル通信

帯域外リモート管理における仮想シリアル通信を暗号化するにあたり，鍵生成をユーザ VM で行なった上で鍵管理をユーザ VM のメモリ上で行い，鍵はディスク上には書き込まない．これにより，VMM 管理者から帯域外リモート管理の通信を盗聴される危険性を排除できる．また，管理 VM のみならず，VMM の管理者からもデータ，鍵を保

護することができるようになるので，クラウドの利用者がVMM の管理者であるクラウドプロバイダをさらに信用しなくて済むことになる．本手法により，帯域外リモート管理を行う場合，管理 VM の管理者と VMM の管理者が同一の場合でも，管理 VM 上及び，VMM 上にデータが渡された時も暗号化が行われており，そのデータの盗聴は行われず，ユーザデータを保護することが可能になる．

仮想シリアル通信の暗号化鍵をユーザ VM で管理する場合のクライアントからユーザ VM への帯域外リモート管理におけるデータの流れを示す（図 2)．まず，クライアントから SSH 通信で管理 VM へ送信する．その後 VMM を介しユーザ VM のシリアルポートに書き込むことでデータを送りユーザ VM にある鍵で復号化する．

なお，鍵管理をユーザ VM で行うだけでは，VMM がユーザ VM のメモリに不正にアクセスして鍵を盗むことを防ぐことはできないが，シャドウページテーブル技術 [4] などを併用し，この問題を解決できる．更に VMM 自体の改変を防ぐためにリモートアテステーション [1][2] という技術も利用できる．これらの詳細は 6 節で示す．

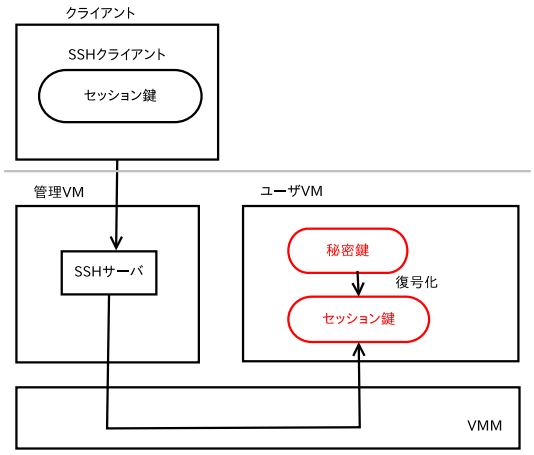

図 2　改良後の帯域外リモート管理の概念図

4　実装

クライアントがユーザ VM とハイブリッド暗号化方式 [5] による暗号化通信を帯域内で行うプログラムを実装した．ユーザ VM で鍵管理を行う帯域外リモート管理の通信を想定し，3 台の PC を用いて二重暗号化通信を行う．

4.1　暗号化方式

我々が用いる暗号化の方法として，排他的論理和と RSA 暗号 [6] の 2 つを用いることでプログラム上でハイブリッド暗号化方式を実装し通信を行った．

4.2　実装内容

平文の各文字とセッション鍵で排他的論理和を取り暗号化を行った後，そのセッション鍵を RSA 暗号化方式にて暗号化する．プログラム上では 3 台の CPU を用いてそれぞれをクライアント，管理 VM，ユーザ VM と仮定し通信を行った．以下にクライアントからユーザ VM にデータを送信する時のプログラムの詳しい流れを表した図 3 を示す．

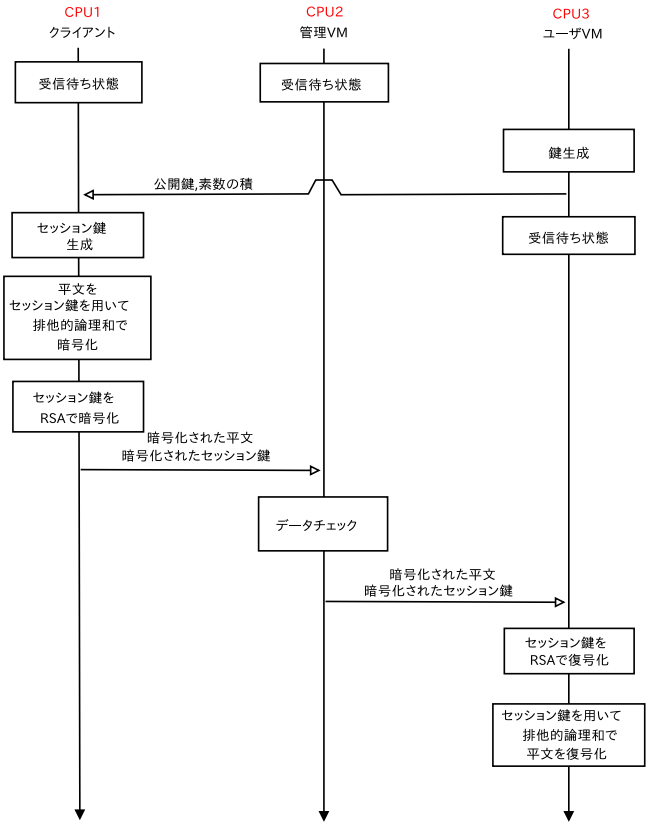

図 3　プログラムの流れ

CPU1 をクライアント，CPU2 を管理 VM，CPU3 をユーザ VM と仮定する．帯域外リモート管理による通信において，クライアントから管理 VM を通してユーザ VM にデータを送信する場合を考える．はじめにユーザ VM で公開鍵，秘密鍵，2 つの素数，素数の積が生成される．その中で公開鍵，素数の積のみを外部に公開するので，CPU2 において RSA 暗号が解読されることはない．通信回数に関しても，暗号化されたデータに対してただ 1 つの暗号化されたセッション鍵を追加して送信するだけなので，暗号化によってデータの通信回数が増えることはない．また，鍵生成においては乱数の生成に多少の時間が掛かるので桁制限を行うことで負荷を軽減することにした．この方法により，クライアントは管理 VM と VMM からデータが盗聴されていたとしても，安全に通信を行うことが可能になる．

5　実験

実装したプログラムを用いて排他的論理和と RSA 暗号による暗号化が行われているかを確かめる実験を行った．具体的な内容は，4 節のプログラムの流れで触れたように，まずユーザ VM からクライアントへ公開鍵と素数の積を送信し，次にクライアントにおいて，Hello!World!!という文字列を管理 VM を介してユーザ VM へ送信するという

ものである．

5.1 実験環境

予備実験では 3 つ端末を使用し，その中でそれぞれをクライアント，管理 VM，ユーザ VM と仮定し通信を行った．本研究に使用した 3 つの CPU の仕様を以下の表 1，2，3 に示す．

表 1 クライアントの仕様

CPU	Core(TM) i5-2520M
OS	Ubuntu 12.04
HDD	200GB
メモリ	4GB
コア数	4
クロック周波数	2.5GHz

表 2 管理 VM の仕様

CPU	Core(TM) i7-4790k
OS	debian8.2
HDD	500GB
メモリ	16GB
コア数	4
クロック周波数	4.0GHz

表 3 ユーザ VM の仕様

CPU	Core(TM) i7-4790k
OS	debian7.9
HDD	4GB
メモリ	512MB
コア数	2
クロック周波数	4.0GHz

5.2 実験の経過

先述の流れに従いプログラムを実行した．まず，ユーザ VM からクライアントへ公開鍵，素数の積の送信が行われる．以下に鍵を送信したユーザ VM の図 4，鍵を受け取ったクライアントの図 5 を示す．

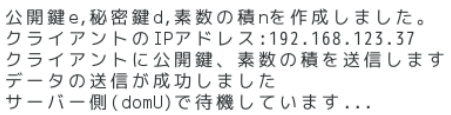

図 4 鍵を送信したユーザ VM

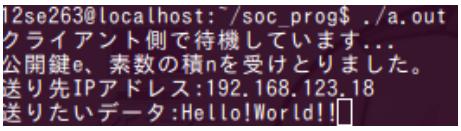

図 5 鍵を受け取ったクライアント

以上の結果より無事に鍵の送受信が出来たことが確認できる．鍵の送信後，送信側であるクライアントは暗号化を行い，管理 VM にデータを送信する．そして管理 VM で送られてきたデータのチェックを行い，元の平文が暗号化されているかどうかを確認する．以下に送られてきたデータのチェックを行う管理 VM の図 6 を示す．

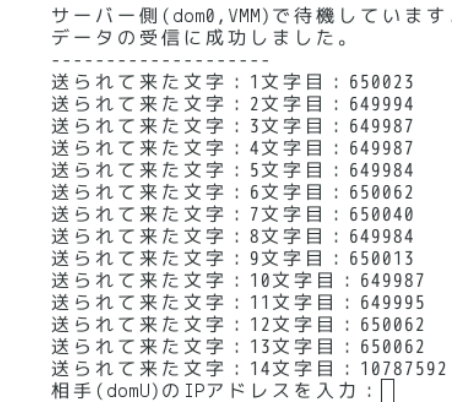

図 6 データチェックを行う管理 VM

13 文字目までが Hello!World!!であり，14 文字目は RSA によって暗号化されたセッション鍵である．どちらも暗号化が行われている．また，管理 VM が知り得る公開鍵と素数の積では容易に復号化できず，例え管理 VM や VMM から帯域外リモート管理の盗聴が行われていたとしても平文，及びセッション鍵の安全性が保障されていることを確認できる．

次にこの暗号化されたデータをユーザ VM に送信する．以下にデータを受信したユーザ VM の図 7 を示す．

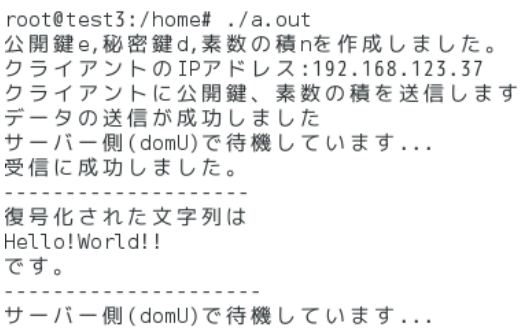

図 7 データを受信したユーザ VM

ここで，ユーザ VM でデータの復号化が行われ，Hello!World!!という文字列に復号できたことを確認することが出来る．したがって，管理 VM に盗聴されていても

安全にクライアントからユーザ VM へデータ送信が出来ていると言える．

5.3 考察

この実験の結果，管理 VM では文字，セッション鍵共に暗号化が行われており，Hello!World!!という文字列，及びセッション鍵が容易に推測できないことを確認した．また，クライアントからユーザ VM へ無事にデータが送られており，元の平文に復号できたことを確認した．これにより通信は暗号化により安全に行なわれているということを確認できた．なお，この暗号化では RSA と排他的論理和を用いた簡単な暗号化しか実装できていない．このままの暗号化方式では同じ文字は同じ数値になってしまったり，暗号化の数値を単純に桁数で制限しているなどの問題が残る．したがって暗号化が行われているからといってデータが完全に保護されるとは言い切れないので，さらに実用性のある暗号化を行う必要がある．

6 関連研究

本研究と組み合わせて使用するセキュリティ技術について紹介する．

6.1 シャドウページテーブルによるアクセス制限

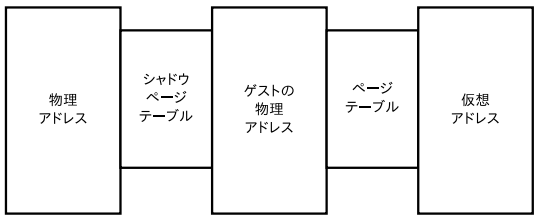

図 8 シャドウページテーブルの概念図

シャドウページテーブルとは，通常のページテーブルとは異なり，真の物理アドレスをゲスト OS にとっての物理アドレスに変換するものである．物理アドレスはシャドウページテーブルによってゲストの物理アドレスに変換される．そしてこのアドレスがページテーブルによって仮想アドレスに変換される (図 8)．

シャドウページテーブルによるアクセス制限を用いると仮想マシンの追加，削除を行える管理インターフェースからユーザ VM への不正アクセスを制限することが可能である．管理インターフェースとは，Xen における管理 VM のことである．ハイパーバイザが管理するシャドウページテーブルに特殊なビットを付与し，ハイパーバイザがアクセス権限チェックを行った際にそのビットをチェックすることで，不正アクセスを禁止することが出来る．この手法によって管理インターフェースからユーザ VM 上のメモリを保護することが可能となっている．本研究では，ユーザ VM 上で保持する秘密鍵を保護するために用いる．

6.2 リモートアテステーション

リモートアテステーション [1][2] とは，プラットフォームの完全性を外部のサーバで検証し，完全性を保障するための仕組みである．検証の方法としては，まず保障したい対象のハッシュ値を TPM（Trusted Platform Module）チップと呼ばれる LSI（Large Scale Integration）チップの内部で計算し，その値を外部の検証サーバへと転送する．そして，転送された値と検証サーバに事前に登録されている値を比較する事によって保障対象が正しいものであるかを証明する．

7 まとめ

我々は IaaS クラウドの帯域外リモート管理の暗号化における鍵管理の管理方法を改良し，管理 VM や VMM の管理者が悪意を持つ場合でも盗聴を防いで安全に通信する手法を提案した．

提案した暗号化方式と鍵管理の方法で行った帯域内暗号化通信の実験により，クライアントとユーザ VM の間で正しく暗号化が行われていることを確認した．

今後の課題は，帯域外通信でも正常に動作することを確認することである．また，ユーザ VM の OS が立ち上がる前，スタートアップに提案手法を組み込むことである．

8 参考文献

[1] J.Christopher Bare: “Attestation and Trusted Computing”, https://courses.cs.washington.edu/courses/csep590/06wi/finalprojects/bare.pdf (参照 2016/1/5)

[2] Lavina Jain , Jayesh Vyas : “Security Analysis of Remote Attestatio”, http://seclab.stanford.edu/pcl/cs259/projects/cs259_final_lavina_jayesh/CS259_report_lavina_jayesh.pdf (参照 2016/1/5)

[3] 梶原 達也，光来 健一 : “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”, 情報処理学会研究報告 Vol.2014-OS-130 No.13 2014/7/28

[4] 村上 航規 他 : “不正な管理者によるゲスト情報の窃盗, 改変を防止するクラウドアーキテクチャ”, Computer Security Symposium 2013 21-23 October 2013

[5] 出口 雄一 :“ITpro by 日経コンピュータ ハイブリッド暗号方式—共通鍵暗号と公開鍵暗号を組み合わせる”, http://itpro.nikkeibp.co.jp/article/COLUMN/20060620/241303/?rt=nocnt (参照 2016/1/5)

[6] R.L. Rivest, A. Shamir, and L. Adleman:“A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, https://people.csail.mit.edu/rivest/Rsapaper.pdf (参照 2016/1/5)