POS 端末における VMM を用いたクレジットカード情報の保護

竹腰 開 †　　品川 高廣 ‡　　加藤 和彦 †

†305-8577 茨城県つくば市天王台 1-1-1

satorut@osss.cs.tsukuba.ac.jp, kato@cs.tsukuba.ac.jp

‡113-8656 東京都文京区弥生 2-11-16

shina@ecc.u-tokyo.ac.jp

あらまし POS 端末を対象にしたマルウェアには、端末に接続されたカードリーダーが読み取ったクレジットカードの情報を窃取し外部に送信する事を目的にしているものが多い。カード情報を読み取った直後に暗号化して POS 端末に送るカードリーダーは既に存在するが、その為には専用のカードリーダーが必要となる。本研究では、暗号化に対応していない通常のカードリーダーと VMM を組み合わせて OS にカード情報が入力される前の段階で暗号化を施す事で、特別なカードリーダー無しにカード情報の保護を実現する。暗号化にあたってはカード情報のフォーマットを維持したまま暗号化を施す事で既存の POS アプリケーションに対する改修を最低限に抑える。

VMM-based Protection of Credit Card Data on POS Terminals

Satoru Takekoshi†　　Takahiro Shinagawa‡　　Kazuhiko Kato†

†1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577 Japan

satorut@osss.cs.tsukuba.ac.jp, kato@cs.tsukuba.ac.jp

‡2-11-16 Yayoi, Bunkyo-ku, Tokyo 113-8656 Japan

shina@ecc.u-tokyo.ac.jp

Abstract Criminals use malware to steal credit card data from point-of-sales terminals. Point-to-point encryption between card readers and servers can prevent criminals from stealing the data from POS terminals, but it requires hardware support on card readers. In this paper, we propose a VMM, CardVisor, that allows us to protect credit card data even if a guest OS hosting a POS application is compromised. CardVisor performs encryption inside itself and tells the guest OS and POS application encrypted card data. CardVisor does not require hardware support on card readers and costly reimplementation of POS applications as its encryption is format preserving.

1　はじめに

POS 端末を狙うマルウェアによる脅威は 2005 年頃から徐々に明らかになり始め、2013 年頃からは特に大きい被害が報告されている [1]。2013 年には、米国の小売大手 Target 社で POS マルウェアによる大規模な情報漏洩が発生し、約

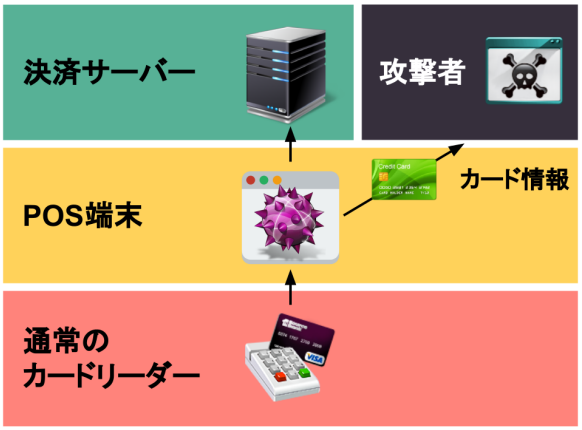

図 1: POS マルウェアによるカード情報窃取の流れ

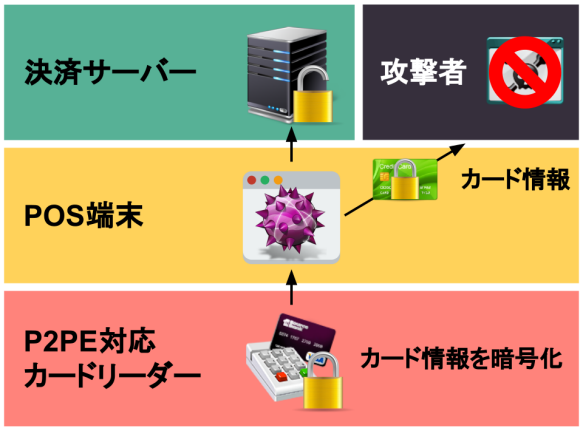

図 2: P2PE 対応カードリーダーによるカード情報の保護

4000 万人分のクレジットカード情報等が漏洩した [3]。また 2014 年には、米国国土安全保障省が、米国内の 1,000 を超える企業で Backoff と呼ばれる POS マルウェアによる感染が発生しているとの推測を発表した [2]。POS マルウェアの主な目的は、POS 端末が持つカード情報の窃取である。窃取されたカード情報は地下マーケットで取引され、偽造クレジットカードの作成などに使われて被害を生んでいる [1]。

クレジットカード決済に対応した一般的な POS 端末の構成では、POS 端末本体がネットワーク経由で決済サーバーと繋がっていて、POS 端末の周辺機器としてクレジットカードリーダーが接続されている。この構成では、カード決済の度にカードリーダーがカード情報を読み取って POS 端末本体に送り、POS 端末本体がカード情報を決済サーバーに送信する。この時、POS 端末本体のメモリー上に一時的にカード情報が記録される。POS マルウェアの多くは、RAM スクレイピング [17] 等の手法でこのカード情報を窃取し、ネットワーク経由で攻撃者に送信する機能を備えている (図 1)。POS 端末では主に OS として Windows Embedded POSReady が採用されており一般の PC 向け Windows と同じ脆弱性が存在し得る上に、攻撃者にとっては従来の Windows 向けマルウェアのノウハウを用いてマルウェアを作成できる為、攻撃対象として狙われやすい [3]。

POS 端末上等でのカード情報の窃取に対する対策としては、ポイント ツー ポイント暗号化 (P2PE) に対応したカードリーダー製品が既に存在する。P2PE 対応カードリーダーは、POS 端末本体へカード情報を送信する前にカード情報を暗号化する。この暗号文は決済サーバーでしか復号できない為、POS 端末上からカード情報を窃取したとしても不正利用できない (図 2)。しかし P2PE 対応カードリーダーはそれ自体が専用のハードウェアであり、POS 端末毎に P2PE 対応カードリーダーを購入し直す必要がある為、導入コストが大きい。

本研究では、POS 端末の上で VMM を動作させ、POS アプリケーションを VMM 上のゲスト OS で動かした上で、VMM がカードリーダーとゲスト OS 間の通信に介入してカード情報を暗号化する手法を提案する。これにより、通常のカードリーダーをそのまま利用しつつ、ゲスト OS が全く信頼できない場合でもカード情報を保護する (図 3)。また、暗号化にあたってはカード情報のフォーマットを維持したまま暗号化を施す事で、POS アプリケーションも既存のものを可能な限りそのまま利用できるようにする。本研究ではこの VMM を CardVisor と呼ぶ事とする。

本研究では、CardVisor 上の Windows Embedded 8.1 でクレジットカード情報を読み取るアプリケーションを動作させ、入力されるカード情報がフォーマットを維持したまま暗号化されている事を確認した。また、暗号化されたカー

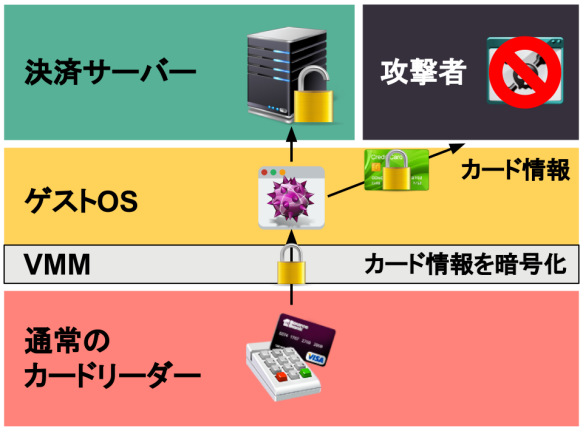

図 3: 提案手法によるカード情報の保護

ド情報を復号し、元のカード情報に戻せる事を確認した。

2 関連研究・製品

カード情報を保護する為の製品としては、カード読み取り直後にカード情報を暗号化する機能を持ったカードリーダーが既に複数販売されている [4][5]。これらのカードリーダーを使用すれば、仮に POS マルウェアによりカード情報が漏洩しても、攻撃者による不正利用を防ぐことが出来る。しかし、これらのカードリーダーはそれ自体が特別なハードウェアであり、既にカードリーダーを持っていても POS 端末毎に新しくカードリーダーを買い直す必要がある為、導入コストが大きい。

Overshadow[15] では、VMM によってゲスト OS 上のプロセスのメモリーページを暗号化し、当該プロセスに対しては透過的に平文のメモリーページを見せつつ、信頼できない OS から機微な情報を保護する手法を提案している。Overshadow でも Virtual DMA といった形でデバイスとの入出力の保護に一部対応しているが、対応は限定的であり、本研究で目的としているようなカードリーダーからの入力の保護等は行えない。

3 設計

3.1 脅威モデル

本研究が対象とする脅威モデルは、ネットワーク経由でのマルウェア感染と特権昇格である。攻撃者はネットワーク経由の何らかの方法で POS 端末をマルウェアにに感染させ、その OS の上で任意のコードを実行できるものとする。その際、攻撃者は特権昇格とカーネル保護機構の回避に成功し、デバイスドライバの差し替えやカーネルへのパッチ適用を行えるとする。攻撃者は POS 端末への物理アクセスを得ず、POS 端末の使用者 (小売店の従業員等) も攻撃者に協力する事は無いものとする。攻撃者の目的は窃取したクレジットカード情報によるカードの不正利用とし、仮に部分的なカード情報 (例えば決済に使用されたカードの所有者名) が得られたとしても不正利用が行えなければ攻撃は失敗とする。

3.2 VMM

本研究の目的は、POS アプリケーションが動作する OS が特権プロセスやデバイスドライバ含めて一切信頼できない場合でもカード情報を保護する事である。このために、設計としてはカードリーダーからの入力を一度全て CardVisor が受け取り暗号化を行った上で、CardVisor がゲスト OS に対して仮想カードリーダーとして動作し、暗号化を施したデータのみをゲスト OS に受け渡す事とする。暗号化に使用する秘密鍵は CardVisor のみがアクセス出来るようにし、暗号化は全て CardVisor の中で完結させる。

3.3 暗号化とフォーマットの維持

クレジットカードにはカード番号, カード所有者名, 有効期限等の様々な情報が記録されており、これらの情報はカードリーダーで読み取れるようになっている。本研究では、POS アプリケーションに対する変更を最小限に抑えつつカードの不正利用を防ぐ為に、カード番号のみを暗号化し他の情報は可能な限りそのままゲス

トOSに入力する事とした。この際、カード番号を含めた全ての情報について、標準規格[8][9]で規定された最大データ長と構成文字種を維持する事とした。

一般にクレジットカード番号では下1桁がチェックディジットとなっており、上15桁をLuhnアルゴリズムで処理した数字が付与されている[13]。CardVisorでは、このチェックディジットの整合性を暗号化した後も維持する。また、カード決済の現場では、ユーザーの利便性の観点からレシートに下4桁を印字するといった事が一般に行われている。この為に、CardVisorではカード番号の下4桁は暗号化した後も平文と同じものを出力する。その上で、暗号化されたカード情報のみから、CardVisorによって暗号化されたカード情報であるかどうかを判別可能にする。

4 実装

4.1 暗号化

カード番号をフォーマットを維持したまま暗号化する場合、一般によく使われているAESなどのブロック暗号をそのまま使うことはできない。CardVisorで暗号化する対象としたカード番号の上12桁は約40bitの情報量を持つが、AESなどのブロック暗号のブロック長(暗号文の最小単位)はそれよりも大きい128bitなどが一般的であり、暗号文をカード番号として埋め込めない。また、ストリーム暗号を採用した場合には、カード番号の下位が一部だけ違うような2つのカード番号で暗号文の上位が一致してしまう特性を利用して解読される恐れがある為、何らかの工夫を施す必要がある。

このため、本研究ではカード番号の暗号化にFNR[10]のリファレンス実装であるLibfnr[11]を用いる事とした。FNRはAESを利用したブロック暗号である。FNRでは128bit以下の任意のbit長をブロック長として設定できる為、ブロック長を40bitに設定して本研究の暗号化に用いる事とした。

また、カード番号は10進であるのに対してブロック長が2進単位である事から、40bitの暗号文でも10進12桁に収まらない事がある。その場合には、Cycle-Walking[14]と呼ばれる手法を用いて暗号文を10進12桁に収める事とした。Cycle-Walkingでは、暗号文が所定の集合(この場合は10進12桁)に属さない場合、暗号文がその集合に属すまで暗号化を繰り返す。復号の際は、同様に平文がその集合に属すまで暗号文を復号する事で、もともとの平文が得られる。ブロック長40bitのFNR暗号において平文から暗号文への撹拌に偏りが無く暗号文を擬似乱数として見なせると仮定した場合、10進12桁に収まる暗号文が得られるまでに必要な暗号化回数の期待値は約1.08回であり、暗号化を繰り返したとしても十分少ない回数で最終的な暗号文が得られる。

4.2 フォーマットの維持

しかし、データ長の維持だけでは3.3で定めた要件を満たせない。カード番号全桁を暗号化するのであれば上15桁を暗号化した上でチェックディジットを計算しなおせばよいが、下4桁を平文と同じにする為には下1桁にあるチェックデジットも平文と同じ数字にしなくてはならず、チェックデジットの再付与は行えない。そこでCardVisorでは、暗号化後のカード番号の上から12桁目(下から5桁目)をチェックディジットの調整桁とし、チェックディジットの整合性が合うように数字を書き換える事とした。また、CardVisorで暗号化されたカード情報とそうでないものを見分ける為に、CardVisorでは暗号化後のカード番号の上から1桁目を8に固定する事とした。クレジットカード等の番号において上6桁は発行者識別番号(Issuer Identification Number, IIN)と呼ばれ、カード発行者毎に使用する範囲が割り当てられ管理されている[12]。本稿執筆時点で上1桁が8のIINはどのクレジットカード会社にも割り当てられていない事から、上1桁が8のクレジットカード番号は存在しない事が保証できる為、CardVisorでは便宜的に上1桁8をCardVisor固有のIINとして扱う事とした。

これらの数字の書き換えにより、暗号化後のカード番号から10進表記で2桁分の情報が失

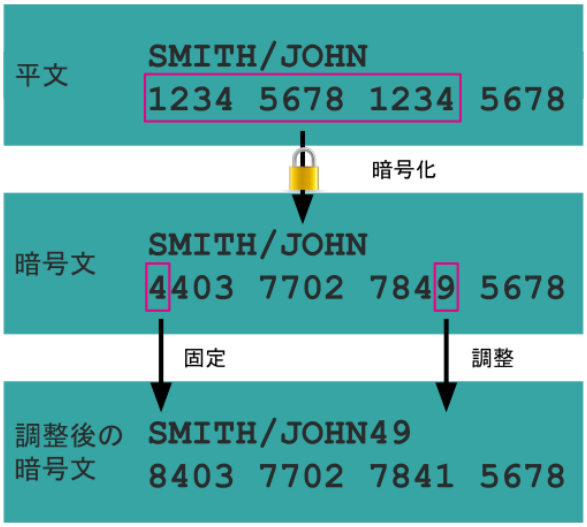

図 4: カード番号と所有者名の変化

われる事になる。暗号化されたカード番号を可逆的に復号する為には失われた情報も決済サーバーに伝える必要がある為、これらの情報をどうにかしてカード情報のいずれかのフィールドに埋め込む必要が発生する。そこで CardVisor では、これらの情報をカード所有者名フィールドの所有者名末尾に付加する事で、決済サーバー側で復号が可能なようにする。カード所有者名フィールドでは数字も使用可能である。暗号化時には所有者名末尾にカード番号の失われた桁をそのまま付加し、復号時には所有者名末尾の 2 文字を切り取ってカード番号の復号に使う事とする。なお所有者名フィールドの最大長は 26 文字である為、所有者名が 24 文字を超えた場合には所有者名の末尾を切り落として失われた桁を付与する。

暗号化とフォーマットの維持に伴うカード番号と所有者名の変化を図 4に示す。

4.3 VMM

CardVisor の実装は BitVisor[6] をベースに行った。BitVisor は Intel/AMD CPU 上で動作するオープンソースの VMM である。BitVisor は動作にホスト OS を必要としない為攻撃面が小さく、また軽量で改変が比較的容易という利点がある。

本研究では、一般に入手可能な磁気カードリーダー (IDTECH 社 IDKA-333312B [7]) を用いて実装と評価を行った。このカードリーダーは PS/2 接続のキーボードとして動作し、クレジットカードを通すとカード情報をキー入力としてコンピューターへ送信する。提案手法では、PS/2 のキーボードコントローラーからゲスト OS への入力を CardVisor でフックして全て受け取り、暗号化を施した上で、CardVisor が仮想キーボードとして動作してゲスト OS にキー入力を渡す事とした (図 5)。

本研究では、POS 端末の起動時に CardVisor を USB 接続のフラッシュメモリーから読み込み、次いで内蔵 HDD に格納されたゲスト OS を起動する事とした。暗号鍵は CardVisor と共にフラッシュメモリーに格納し、CardVisor 読み込み後は USB コントローラーへの入出力をフックして当該フラッシュメモリーへのアクセスを不可能にする事で、ゲスト OS から暗号鍵を読み取ったり CardVisor に対して変更を加える事を不可能とした。

5 評価

評価では、CardVisor 上のゲスト OS でクレジットカード情報を読み取るアプリケーションを動作させ、クレジットカードリーダーからの入力を確認した。その結果、設計で定義した要件の通りカード情報のフォーマットを維持した状態でカード番号が暗号化され、カード情報として不正利用ができない状態になっている事を確認した。その上で、暗号化されたカード情報を別途作成したプログラムで復号し、平文のカード情報に戻せる事を確認した。

6 今後の課題

本研究では、磁気カードリーダーを用いて実装と評価を行ったが、IC カードリーダーへの対応は今後の課題である。

CardVisor では、実装のベースとした BitVisor の制約上、POS 端末が仮想化支援技術の Intel VT か AMD-V に対応している事が必須条件となっている。設計上はこれらの仮想化支援技術

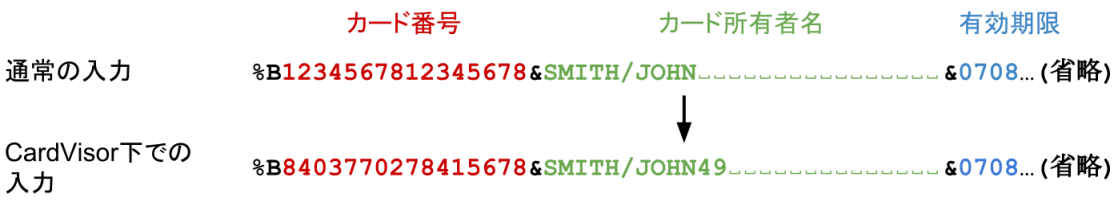

図 5: 暗号化されていないカード情報の入力と CardVisor 下での暗号化されたカード情報の入力

は必須でない為、これらを必要としない VMM をベースとすれば同様のシステムを実装できると考えるが、一部で新たな実装が必要になる。

クレジットカードの決済機器やソフトウェアでは、実際に使用される為には PCI[16] が定めるセキュリティ基準の認証を取得する事が事実上必須になる。本研究は手法の提案に焦点を置いた為それらのセキュリティ基準については参考に留めているが、仮に認証の取得を念頭に置いた場合、新たな課題が出てくることが考えられる。

本研究ではカードの不正利用の防止が目的だった為、暗号化の対象としたのはカード番号のみであったが、カード番号以外の情報 (カードの所有者名等) を暗号化する事も原理的には可能である。カード番号以外の様々な情報も含めた暗号化は今後の課題である。

実装で述べたように、CardVisor の暗号化では、カード番号のフォーマット等を維持する為に失われた情報をカード所有者名のフィールドに埋め込む事で可逆的な復号を可能にしている。しかし、この為にカード番号とカード所有者名が対になって保存・伝送されないと可逆的な復号が出来ない。フォーマット維持の為に失われる情報を格納するだけの余裕がカード番号空間に無い為、本研究で目指した下 4 桁とチェックディジットの維持等の要件を守る限りは、他のフィールドに埋め込むか別途伝送する方法を取らざるを得ないと考える。この点については、必要に応じてフォーマット維持要件の緩和や埋め込む先のフィールドの検討が必要である。また、この情報の埋め込みの為に、カード所有者名が長い場合には所有者名末尾を一部削る必要がある。削る長さが小さく他の情報からもカード所有者の特定は可能な為、この問題については決済サーバー側で対処可能かと考えるが、カード所有者名の完全な一致が必要な場合は他の方法を取る必要がある。

7 結論

POS 端末に感染するマルウェアの脅威は近年大きくなりつつある。POS マルウェアの主な目的は POS 端末が持つクレジットカード情報の窃取であり、窃取されたカード情報は偽造カードの作成等に使われて被害を生んでいる。そのような問題に対処する為に、カード情報をハードウェアで暗号化して POS 端末に送信するクレジットカードリーダーは既に存在する。しかし、そのようなカードリーダーはそれ自体が専用のハードウェアである為、既にカードリーダーを持っていても POS 端末毎に暗号化対応カードリーダーを買い直す必要があり導入コストが大きい。

本研究では、通常の暗号化非対応のカードリーダーと VMM を組合せて、VMM によってカード情報を暗号化する事で、通常のカードリーダーを使用しつつゲスト OS が信頼できない場合でもカード情報を保護する手法を提案した。暗号化はカード情報のフォーマットを維持したまま行い、既存の POS アプリケーションに対する修正を最小限に抑えられるようにした。

実装した VMM を用いて評価を行い、ゲスト OS に入力されるカード情報がフォーマットを維持したまま暗号化されており、暗号化されたカード情報を元のカード情報に戻せる事を確認した。

参考文献

[1] Symantec. A special report on attacks on point-of-sales systems. 2014. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/attacks_on_point_of_sale_systems.pdf

[2] US-CERT. Alert (TA14-212A) Backoff Point-of-Sale Malware. 2014. https://www.us-cert.gov/ncas/alerts/TA14-212A

[3] FFRI,Inc.. Monthly Research Survey of POS Malware. 2014. http://www.ffri.jp/assets/files/monthly_research/MR201409_Survey%20of%20POS%20Malware_ENG.pdf

[4] SecuRED http://www.idtechproducts.com/products/mobile-readers/180.html

[5] MagTek Dynamag - Secure Card Reader Authenticator for magstripe cards - swipe http://www.magtek.com/V2/products/secure-card-reader-authenticators/Dynamag.asp

[6] 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: a thin hypervisor for enforcing i/o device security." Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments. ACM, 2009.

[7] VersaKey Compact http://www.idtechproducts.com/index.php/products/pos-peripherals/76.html

[8] International Organization for Standardization. ISO/IEC 7813:2006. 2006.

[9] International Organization for Standardization. ISO/IEC 4909:2006. 2006.

[10] Sashank Dara and Scott Fluhrer. "FNR: Arbitrary length small domain block cipher proposal." Security, Privacy, and Applied Cryptography Engineering. Springer International Publishing, 2014. 146-154.

[11] Libfnr http://cisco.github.io/libfnr/

[12] Issuer Identification Number (IIN) http://www.ansi.org/other_services/registration_programs/iin_registration.aspx

[13] International Organization for Standardization. ISO/IEC 7812-1:2015, 2015.

[14] John Black and Phillip Rogaway. "Ciphers with arbitrary finite domains." Topics in Cryptology—CT-RSA 2002. Springer Berlin Heidelberg, 2002. 114-130.

[15] Xiaoxin Chen, Tal Garfinkel, E. Christopher Lewis, Pratap Subrahmanyam, Carl A. Waldspurger, Dan Boneh, Jeffrey Dwoskin and Dan R.K. Ports. "Overshadow: a virtualization-based approach to retrofitting protection in commodity operating systems." ACM SIGOPS Operating Systems Review. Vol. 42. No. 2. ACM, 2008.

[16] Official PCI Security Standards Council Site https://www.pcisecuritystandards.org

[17] Securing the point of sale http://www.sciencedirect.com/science/article/pii/S1361372314705573