情報処理学会論文誌 Vol.61 No.4 978–989 (Apr. 2020)

コマンド真正性検証を用いたセキュアなATM設計法

緒方 日佐男[1,2,a)] 石川 智祥[2] 宮本 範親[2] 松本 勉[1]

受付日 2019年7月29日, 採録日 2020年1月16日

概要：海外では，ATM（Automated Teller Machine）システムは，様々な論理攻撃を受けて不正出金が行われており，深刻な社会問題となっている．既存対策は ATM 制御部である PC の保護に重点が置かれているが，ATM の運用では数日おきの紙幣補充回収による内部アクセスや，定期的な PC のソフトウェアやデータ更新が発生するので，PC へのマルウェア侵入の懸念があった．そこで，筆者らは現金処理モジュールといった，ATM 内の周辺デバイスに送られるコマンドの真正性を，周辺デバイス自身が検証する「コマンド真正性検証法」を提案した．しかし，磁気カードを用いた出金取引に本検証法を適用すると，既存のセキュリティ制約が少ないために適用箇所や適用の仕方は様々であるうえ，既存運用への影響の最小化といった考慮すべき観点も多数存在するため，適用設計が難しいという問題があった．そこで，本論文では出金取引中の多様な論理攻撃対策，既存運用や周辺デバイス改造への影響最小化を考慮しながら，コマンド真正性検証法を ATM に適用する最適なシステム設計法を提案する．

キーワード：ATM，セキュリティ，マルウェア，ネットワーク，暗号，デバイス

Secure ATM Device Design by Control Command Verification

Hisao Ogata[1,2,a)] Tomoyoshi Ishikawa[2] Norichika Miyamoto[2] Tsutomu Matsumoto[1]

Received: July 29, 2019, Accepted: January 16, 2020

Abstract: Recently, criminals frequently utilize logical attacks to install malware in the PC of Automated Teller Machines (ATMs) for the sake of unauthorized cash withdrawal from ATMs. Existing security measures primarily try to protect executable files in the PC so as not to be compromised by malware. Such security measures are not so effective or efficient because frequent physical/logical accesses inside each ATM are required in existing ATM operations, for example, once a few days to a week periodical cash replenishment and collection for cash services, and once a quarter periodical software/data updating. To cope with the issues, we proposed an ATM security measure called "Command Verification" that a peripheral device itself verifies a control command received from the PC in other paper. When the measure is applied to magnetic stripe card transactions, many applied systems are derived because of less security constraints resulted from the existing security standards for magnetic stripe card transactions. Proper applied systems should be selected among these many candidate systems from three points of view: preventing a wide range of logical attacks, being harmonized with existing ATM operations, and minimizing to modify existing peripheral devices. This paper proposes a design method to apply the Command Verification to ATM devices/systems by considering the three points.

Keywords: ATM, security, malware, network, cryptography, device

1. はじめに

海外では，ATM（Automated Teller Machine）システムは様々な論理攻撃を受けて 30 カ国以上で不正出金が行われている．経済成長にともない世界的に ATM の台数は増加しているので，ATM システムへの論理攻撃は

[1] 横浜国立大学
Yokohama National University, Yokohama, Kanagawa 240–8501, Japan
[2] 日立オムロンターミナルソリューションズ株式会社
Hitachi-Omron Terminal Solutions, Corporation, Owariasahi, Aichi 488–8501, Japan
[a)] hisao_ogata@hitachi-omron-ts.com

深刻な社会問題となっている．ATM は金融機関内の勘定系ホストコンピュータ（勘定系ホスト）と広域網で接続されており，ATM 内部は制御部である PC と，カードリーダや現金処理ユニットといった周辺デバイスとで構成され，制御部と周辺デバイスは USB や RS-232C ケーブルで接続されている．論理攻撃における主な攻撃対象は，暗号保護されてない広域網 [1], [2], [3]，USB/RS-232C ケーブル [1], [2], [3], [4], [5], [6]，マルウェア対策が不十分な PC [1], [2], [3], [7], [8], [9] である．論理攻撃によって，最終的には取引の裏付けのない現金出金コマンドを，ATM 内の現金処理ユニットに送信して不正出金を行う．このような論理攻撃に対し，対策ガイドラインや対策方法が，複数国の公的機関や ATM ベンダ，学会から出ている [1], [10], [11], [12], [13], [14]．主な対策内容は，ATM と勘定系ホスト間の暗号通信，PC と周辺デバイス間通信の暗号化や，PC のマルウェア対策強化である．

上記既存対策は，PC 上の実行ファイルの真正性を維持しなければならない．しかし，ATM の運用では数日おきの紙幣補充・回収による内部アクセスや，定期的コンテンツ・ソフトウェアの更新が発生するので，上記の実行ファイル真正性維持に必要なマルウェア対策が，無効化される危険がある．具体的には，ATM の扉を開けると PC に直接アクセス可能な ATM が多いため，PC のハードディスクの内容を書き換えてマルウェアを仕込む [1], [2], [3]，あるいは，コンテンツ・ソフトウェア配信サーバを乗っ取り，マルウェアを ATM に配信する，等である [3]．海外 ATM では，PC 上のアプリケーション開発に必要な API（Application Programming Interface）[15] や，通信仕様 [16] は標準化が進んでいて公開されており，ATM 運用にかかわる作業員の流動性も高いため，マルウェア開発と ATM への感染に内部犯罪が疑われる事例 [9] も存在する．前述のように，運用では ATM 内 PC に物理的・論理的アクセスが必要なので，守るべき実行ファイルの真正性を維持する仕組みが必要であるが，運用管理によって真正性維持を図る場合には，運用・管理コストの増大が課題となる．特に，金融機関が 1 万台以上といった大規模に ATM を運用している場合，限られた人員による運用管理で，真正性維持を図るのは困難である．

上記の課題に対し，筆者らは PC の実行ファイルを厳密に保護する代わりに，PC から送られてきたコマンドの真正性を，周辺デバイス自身が検証する「コマンド真正性検証法」とその基本的枠組みを提案し，IC カードを用いた出金取引への適用例を示した [17]．ATM はコマンドに応じて周辺デバイスを受動的に動作させているので，不正なコマンドを周辺デバイスという最後の砦で止めるのは，当たり前な考えであるが有効である．周辺デバイスには耐タンパ性ハードウェアが実装され，その耐タンパ性ハードウェアを用いて，周辺デバイスのファームウェアと暗号処理の真正性が担保される．一方，コマンドの真正性検証に必要な情報は，その情報源となり，かつ，耐タンパ性ハードウェアを実装している他の周辺デバイスから暗号化されて送られてくる．基本的枠組みは，ATM 以外にも適用可能な，抽象的なモデルで表現されている．したがって，IC カードに比べてセキュリティ対策が少ない磁気カードを用いた出金取引に適用する場合，本枠組みの適用カ所や適用の仕方は様々である．出金取引における多様な論理攻撃への対処，既存運用への影響最小化，周辺デバイス改造影響の最小化，の 3 つの観点から適用システムの選択が必要である．

本論文では，上記 3 つの観点を考慮しながら，コマンド真正性検証法を適用する ATM システム設計法を提案する．さらに，磁気カードを用いた出金取引に提案設計法を適用した結果，提案設計法適用前のシステム数が 135 だったのに対し，適用後は 3 に減らせることを示す．

本論文では，2 章で既存の ATM システムと想定される論理攻撃を説明したうえで，コマンド真正性検証法の概要と課題を示す．3 章では，コマンド真正性検証法を用いたシステム設計法について説明する．4 章では，磁気カードを用いた出金取引への適用と評価結果を示す．5 章で結論を述べる．

2. コマンド真正性検証法

2.1 ATM システムと磁気カード出金取引に対する脅威

図 1 に ATM の構成概要を示す．ATM は PC と周辺デバイスで構成され，PC と周辺デバイスは USB/RS-232C ケーブルで接続されている．PC は 3 つの論理階層からなり，金融機関やサードパーティが提供するマルチベンダアプリケーション，周辺デバイス制御の共通インタフェースを提供する ATM プラットフォーム，オペレーティングシステム（OS）からなる．ATM プラットフォームには，仕様がデファクト標準になっている “CEN/XFS” と呼ばれる API 群 [15] が存在する．これらの API 群は，90 年代に基本仕様が策定され暗号保護がないことから，しばしば不正出金マルウェア攻撃に悪用されてきた．

磁気カードを用いた出金取引の流れとデータの流れをそ

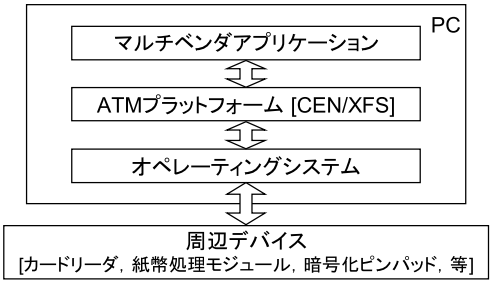

図 1 ATM 構成概要
Fig. 1 Overview of an ATM system.

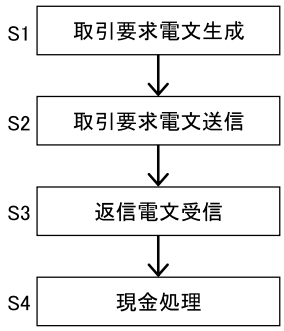

図 2 既存の出金取引の流れ
Fig. 2 Existing cash withdrawal transaction.

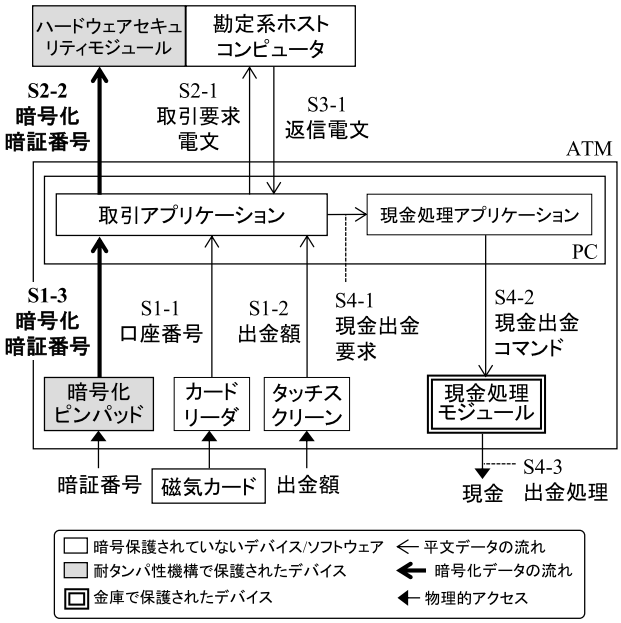

図 3 出金取引のデータの流れ
Fig. 3 Data flow of cash withdrawal transaction.

れぞれ図 **2**，図 **3** に示す．IC カード取引では EMV® [*1]仕様 [18], [19] に基づき，耐タンパ性のある IC カードの暗号機能で，送受信電文と IC カードから出力される口座番号の真正性が確保されるのに対し，磁気カードにはそのような機能がないので，暗証番号を除いて図 2 の 4 ステップすべてが脆弱となりうる．図 3 において，PC 上のマルチベンダアプリケーションは，取引アプリケーションと現金処理アプリケーションの 2 つからなると想定する．なお，ATM プラットフォームと OS は省略してある．クレジットカード取引のセキュリティ規格である PCI（Payment Card Industry）規格 [20], [21] や国際標準 [22] に基づき，暗証番号は暗号化ピンパッドと呼ばれる耐タンパ性のあるデバイス内で暗号化して出力される．勘定系ホスト側には耐タンパ性のあるハードウェアセキュリティモジュール（HSM）[23] が設置され，そこで復号と照合が行われる．現金処理ユニットは金庫内に格納され，2 人以上でないとアクセスできない厳重な管理がされている．既存出金取引のデータの流れを以下に示す．

S1 取引要求電文生成：取引アプリケーションが口座番号 S1-1 と出金額 S1-2 を周辺デバイスから取得し，取引要求電文を生成するとともに，暗号化暗証番号 S1-3 を取得する．

S2 取引要求電文送信：取引要求電文 S2-1 と暗号化暗証番号 S2-2 が，ATM から勘定系ホストと HSM にそれぞれ送信される．

S3 返信電文受信：勘定系ホストが口座残高を確認して取引承認/拒否を判断し，その判断結果である取引承認フラグを含む返信電文 S3-1 を ATM に返信する．

S4 現金処理：返信電文 S3-1 に基づき，取引アプリケーションが現金出金要求 S4-1 を現金処理アプリケーションに出す．現金処理アプリケーションは，現金出金コマンド S4-2 を生成して現金処理モジュールに送信し，現金処理 S4-3 を行う．

図 3 の各データの流れに対する脅威と保護資産，攻撃箇所と攻撃内容を**表 1** に示す．最終的に現金処理モジュールから不正出金させるため，攻撃を取引サブプロセスごとに，攻撃箇所を広域網，PC，USB/RS-232C といった階層的に分類，モデル化している．ここで，勘定系ホスト，HSM，暗号化ピンパッドは，安全に保護されていると想定する．また，コマンド真正性検証法で用いる周辺デバイスも，同様に安全に保護されていると想定する．

表 1 現金盗難の論理攻撃
Table 1 Logical attacks to steal cash from ATMs.

#	取引サブプロセス	脅威	保護資産	攻撃箇所	攻撃内容
A1	取引要求電文作成	改ざん	口座番号，出金額	USB/RS-232C	-悪意あるデバイスを用いて，1 つの暗証番号に対し口座番号を次々に変えるリバースブルートフォース攻撃を行い，他人の口座から不正出金する． -出金額を改ざんして意図しない多額の出金を行い，戸惑う ATM 利用者から現金を強奪する．
A2			口座番号，出金額	PC	-悪意あるデバイスの代わりにマルウェアを用いて A1 の攻撃を行う．
B1	取引要求電文送信，返信電文受信	中間者攻撃	取引要求電文，返信電文	広域網	-取引要求電文中の口座番号，出金額を改ざんし，A1 の攻撃を行う． -口座残高や暗証番号に関わらず，取引承認返信電文を送信して不正出金する．
C1	現金処理	不正出金	現金出金コマンド	PC	-マルウェアを用いて取引の裏付けのない不正出金を行う．
C2				USB/RS-232C	-外部 PC を現金処理モジュールに直接接続し，現金出金コマンドを送信して不正出金を行う．
D1		一時的な出金妨害	現金出金コマンドの伝送時間	PC	-マルウェアを用いて現金出金コマンドの送信を一時的に妨害し，ATM 利用者が立ち去った後にコマンドを再送して現金を奪取する．
D2				USB/RS-232C	-マルウェアの代わりに悪意のあるデバイスを用いて D1 の攻撃を行う．

[*1] EMV は EMVCo の米国および諸国における登録商標です．

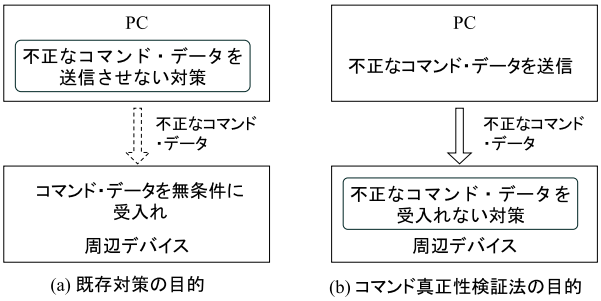

図 4 既存対策とコマンド真正性検証法の目的比較
Fig. 4 Objectives of existing measures and command verification.

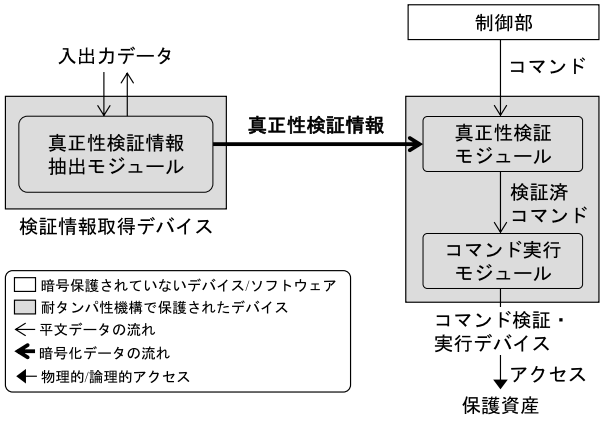

図 5 コマンド真正性検証法の基本的枠組み
Fig. 5 Fundamental framework of "Command Verification".

2.2 コマンド真正性検証法適用上の問題点

既存対策とコマンド真正性検証法のそれぞれの目的を**図 4** に示す．ATM の PC と周辺デバイスは，既存制御システムと同様にコントローラ・アクチュエータモデルで制御されている．周辺デバイスは PC からのコマンド・データを無条件に受け入れるので，既存対策は不正なコマンドやデータを PC に送信させない対策，あるいは，USB/RS-232C ケーブルを暗号保護することを目的とする（図 4 (a)）．それに対し，本検証法では，周辺デバイス自身が不正なコマンド・データを受け入れないことを目的とする（図 4 (b)）．

コマンド真正性検証法の基本的枠組みを**図 5** に示す．既存の周辺デバイスは，コマンドの真正性検証に必要な情報を保持していないので，2 種類のデバイスを定義する．1 つは "検証情報取得デバイス" であり，もう 1 つは "コマンド検証・実行デバイス" である．2 つのデバイスは次のように動作する．検証情報取得デバイスは，入出力データからコマンドの真正性検証情報を抽出し，暗号保護してコマンド検証・実行デバイスに送信する．コマンド検証・実行デバイスは，保護資産にアクセスするコマンドを制御部から受信すると，真正性検証モジュールでその真正性を検証し，検証されたコマンドはコマンド実行モジュールで実行され，保護資産にアクセスする．コマンドの真正性検証情報抽出処理や，真正性検証処理とコマンド実行は，耐タンパ機構で保護されたデバイス内で行われるので，正当なコマンドだけが保護資産にアクセスできる．

文献 [17] で示された，IC カードを用いた出金取引への基本的枠組みの適用例では，送受信電文と IC カードから出力される口座番号の真正性は，EMV 仕様に基づき IC カードと勘定系ホストによって担保される．そのため，表 1 の A1，A2 口座番号改ざん，B1 取引要求電文と返信電文の中間者攻撃への対策は不要であり，C1，C2 の不正出金対策のみを基本的枠組みの適用対象とした．しかし，基本的枠組みを磁気カード取引に適用する場合は，EMV 仕様でカバーされた脅威も含めて表 1 の全脅威をカバーする必要がある．なお，文献 [17] では A1，A2 出金額改ざんや D1，D2 一時的な出金妨害は，モデルの簡単化のために対象としなかった．

基本的枠組みを磁気カード取引に適用するには，次の要件を満たす必要がある．

(1) 多様な論理攻撃への対処
表 1 に示すように，各取引サブステップに対し，攻撃対象資産と攻撃箇所が異なる多様な論理攻撃が存在するので，これらを包括的に防御する必要がある．

(2) 既存運用への影響最小化
既存運用への影響をできるだけ小さくする必要がある．特に，周辺デバイス間や周辺デバイスと勘定系ホスト間の暗号通信に必要な暗号鍵設定作業は，攻撃対象 [24] となるため，厳格な管理が必要であるとともに最小化すべきである．また，ATM に障害が発生したとき，保守員が周辺デバイスを保守部品と入れ替えて修理する場合があるため，周辺デバイス交換にともなう厳密な暗号鍵設定作業は，作業効率の面からも最小化すべきである．

(3) 周辺デバイス改造影響の最小化
図 5 の入出力データが電文の場合，検証情報取得デバイスには，電文構文解析というアプリケーション機能の一部を実装する必要がある．また，真正性検証モジュールが他の周辺デバイスのコマンド検証を行う場合は，コマンド構文解析という他のデバイス機能の実装が必要である．これらの実装には，既存の周辺デバイスの改造が必要であり，複数ベンダが周辺デバイスを提供する際には，ベンダ間で仕様の開示・調整が必要となりシステム構築の足かせとなる．そこで，アプリケーション機能や他のデバイスの仕様をサポートする周辺デバイス数を最小化する必要がある．

磁気カード出金取引では，コマンド真正性検証法の運用箇所や適用の仕方は様々であるので，上記 3 要件を満たすシステムを設計するには，適切な設計法が必要である．

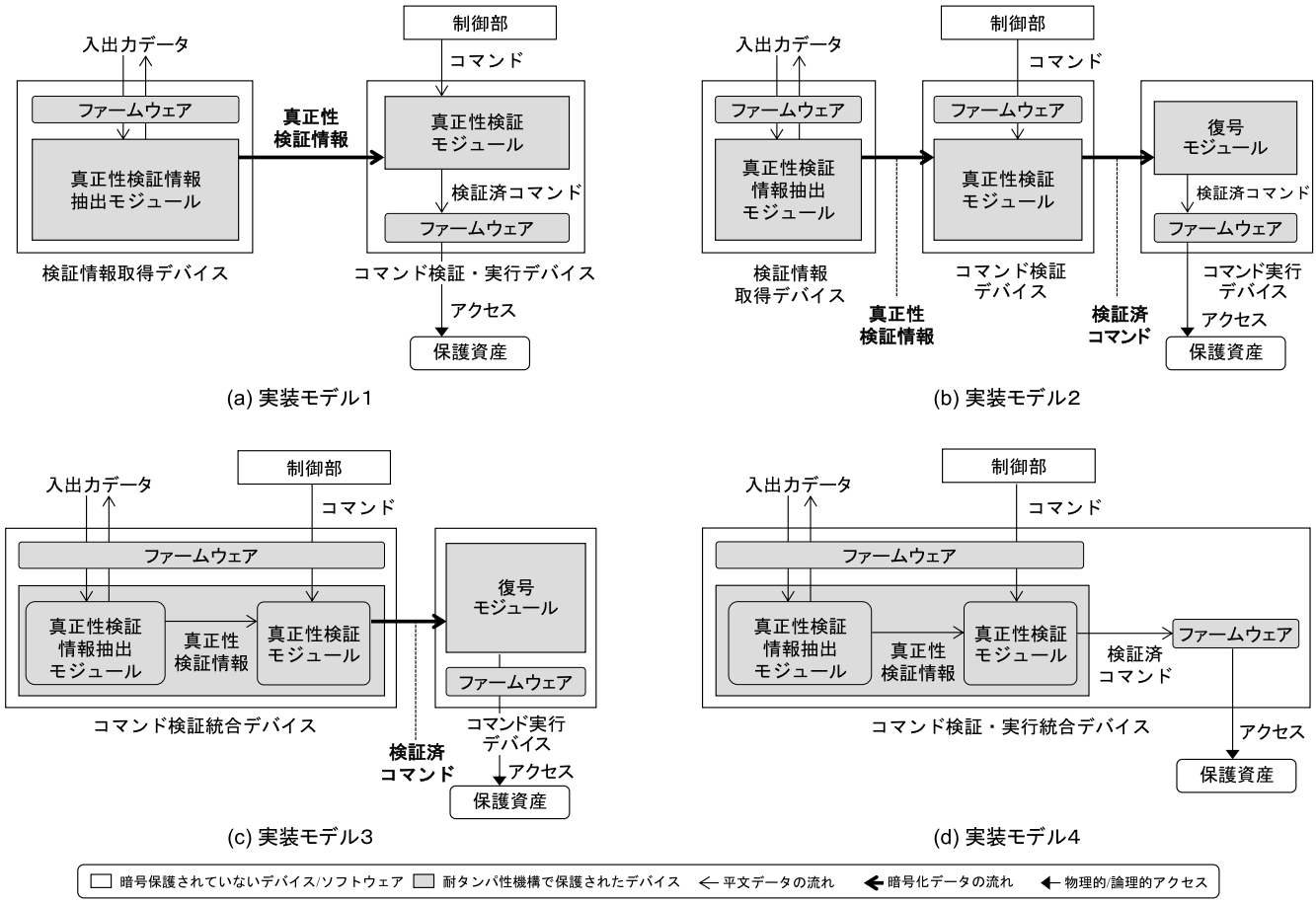

図 6 基本的枠組みの実装モデル
Fig. 6 Implementation models of the fundamental framework.

3. 基本的枠組みの適用設計法

本章では，2.2 節で説明した 3 要件を満たしながら，磁気カード出金取引に適用できる，基本的枠組みの適用設計法について説明する．本設計法は 2 段階からなる．まず基本的枠組みの実装モデルの分析を行い，推奨モデルを選択する．次に，3 つの適用ステップに従い，出金取引の各取引サブステップに推奨モデルを適用ながら，全体システムを設計する．

3.1 実装モデル

コマンド真正性検証法の基本的枠組みに対する実装モデルを**図 6** に示す．各周辺デバイスは，耐タンパ性のハードウェアと，ファームウェアを含む既存の制御メカニズムからなる．図には明示していないが，ファームウェアの真正性は，耐タンパ性ハードウェアに格納された電子署名で担保されている．図 6 (a) は基本的枠組みをそのまま実装したモデルである．図 6 (b) は，図 6 (a) の右側のコマンド検証・実行モジュールを分離独立させたモデルであり，それぞれのデバイス間は，暗号通信で結ばれている．図 6 (c) は，図 6 (b) の左側 2 つのデバイスを 1 つのデバイスに統合した実装モデルである．図 6 (d) は，図 6 (b) の 3 つのデバイスを 1 つに統合したモデルである．

表 2 磁気カード取引に対する実装モデルの比較
Table 2 Comparison of implementation models for magnetic stripe card transactions.

#	特　徴	実装モデル 1	実装モデル 2	実装モデル 3	実装モデル 4
1	周辺デバイス間暗号通信数	1	2	1	0
2	コマンド伝送時間の真正性検証	可	不可	不可	可
3	コマンド伝送時間以外の真正性検証	可	可	可	可
4	複数ベンダの周辺デバイス採用時の修正量	小	大	大	小

磁気カード出金取引における各実装モデルの特徴を**表 2** にまとめる．これらの特徴は 2.2 節の要件より導出されている．No.1 は要件 (2) の暗号鍵設定作業最小化より，No.2, No.3 は要件 (1) の多様な論理攻撃への対処，特に表 1 の D1，D2 のコマンド伝送時間の検証より，No.4 は要件 (3) よりそれぞれ導出される．結論として，望ましい特徴を備える実装モデル 1，4 が推奨される．特徴 1 に関して，要件 (2) のデバイス間暗号通信数を最小化する観点で，実装モデル 2 は推奨されない．実装モデル 4 はデバイス間暗号通信がなく理想的だが，入出力データ取得とコマンドの検証

が 1 つのデバイスで実行できる場合にのみ選択可能であるため，モデル 1，3 も許容すべきである．特徴 2 に関して，実装モデル 2，3 は推奨できない．これらモデルのコマンド実行デバイスには検証機能がないので，検証済みコマンドの伝送時間を検証できず，表 1 の D1，D2 の攻撃に対処できない．特徴 3 は全モデルが満たすことができる．特徴 4 に関して，実装モデル 2，3 のコマンド検証デバイスは，コマンド実行デバイスのコマンド構文解析機能の実装が必要なので，要件 (3) を満たせず推奨できない．

3.2 提案設計法の概要

提案する適用設計法は，2.2 節で示した 3 つの要件を満たしながら，システマティックに基本的枠組みの適用システムを設計するための方法であり，3 つの適用ステップと各適用ステップに対するガイダンスからなる．

ステップ 1：全取引サブプロセスに対し，保護資産と保護資産を狙う論理攻撃をリストアップする．

ガイダンス 1 現金を奪取するために，各論理攻撃は 1 つの取引中の異なる取引サブプロセスの保護資産を狙う．そのため，各取引サブプロセス内処理だけでなく，取引サブプロセス間もあわせて整合性の検証が必要である．それらを考慮して，全取引サブプロセスに対する保護資産と保護資を狙う論理攻撃をリストアップする．

ステップ 2：保護資産にアクセスするコマンドの真正性検証に必要な情報を特定し，その情報ソース，ならびに，その情報を安全に取得できるデバイスを決定する．

ガイダンス 2 コマンド真正性検証情報の真正性を確保するため，その情報ソースにできるだけ近いデバイスで，かつ，安全な形で真正性検証情報を取得する．

ステップ 3：基本的枠組みの推奨実装モデルを用いて，保護資産アクセスコマンドの真正性を検証するデバイスと，検証済みコマンドを実行するデバイスを決定する．

ガイダンス 3-1 各論理攻撃を防御するため，コマンド真正性検証に適当なデバイスを選択する．コマンドに含まれるデータとパラメータも真正性検証対象である．コマンド送信時間の検証が必要な場合は，推奨される実装モデルが 2 つしかないことに注意する．

ガイダンス 3-2 既存運用と整合できる実装モデルを選択する．1 つのポイントは，厳密な手順が必要な，暗号通信の暗号鍵設定作業を最小化できるモデルを選択することである．

ガイダンス 3-3 システム構築のために，多くのベンダが周辺デバイスを容易に提供できるよう，PC 上のアプリケーション機能や，他の周辺デバイスの機能を実装する必要のある周辺デバイス数を最小化する．

ガイダンス 3-4 各取引サブプロセスにおいて，保護資産にアクセスするコマンドの真正性を，“シームレス” に検証できる適切なデバイスを選択する．ここで “シームレス” とは，1 つの取引サブプロセスでコマンド真正性を検証するデバイスが，次の取引サブプロセスにおいて，コマンドの真正性検証情報を提供するデバイスになることである．結果として，選択されたデバイス群は，取引全体の資産を保護するため，取引サブプロセス間の一連の整合性のチェーンを提供する役割を担う．

なお，磁気カード取引では，任意の周辺デバイスが勘定系ホストコンピュータとの通信デバイスになれるので，“シームレス” なデバイスを作ることができる．一方，IC カード取引の場合，勘定系ホストコンピュータとの通信デバイスは IC カードだけなので，“シームレス” なデバイスを作ることができない．その場合は，IC カードに接するカードリーダが IC カードの入出力データを直接参照することができるので，ガイダンス 2 に基づき，カードリーダを代替デバイスにすることができる．

4. 提案設計法を用いた適用設計

4.1 磁気カード出金取引への適用

3 章で説明した適用設計法を用いて，コマンド真正性検証法の基本的枠組みを，磁気カード出金取引に適用する設計手順について記述する．

ステップ 1：全取引サブプロセスに対し，保護資産と保護資産を狙う論理攻撃をリストアップする．

取引サブプロセスごとに，保護資産とそれを狙う論理攻撃について表 1 を基に**表 3** にまとめる．

ステップ 2：保護資産にアクセスするコマンドの真正性検証に必要な情報を特定し，その情報ソース，ならびに，その情報を安全に取得できるデバイスを決定する．

コマンドの真正性検証情報と，その情報取得デバイスを**表 4** にまとめる．周辺デバイスからの入力と，勘定系ホストとの通信に基づいて ATM は動作するので，入出力を担う周辺デバイス，ならびに，勘定系ホストとの通信を担う周辺デバイスから検証情報を取得する．S1 において，通常タッチスクリーンには暗号機能がないので，ガイダンス 2

表 3 保護資産と論理攻撃
Table 3 Protective property and logical attacks.

#	取引サブプロセス	保護資産	攻撃箇所	攻撃
S1	取引要求電文生成	口座番号，出金額	USB/RS-232C ケーブル	A1 改ざん
			PC	A2 改ざん
S2	取引要求電文送信	取引要求電文	広域網	B1 中間者攻撃
S3	返信電文受信	返信電文		
S4	現金処理	現金出金コマンドの出金額	PC	C1 不正出金
			USB/RS-232C ケーブル	C2 不正出金
		現金出金コマンドの送信時間	PC	D1 一時的な出金妨害
			USB/RS-232C ケーブル	D2 一時的な出金妨害

表 4 コマンド真正性検証情報と情報取得デバイス
Table 4 Command verifying information and information acquiring device.

#	取引サブプロセス	保護資産	真正性検証情報	検証情報取得デバイス
S1	取引要求電文生成	口座番号，出金額	磁気カード上の口座番号	カードリーダ
			入力された出金額	暗号化ピンパッド
S2	取引要求電文送信	取引要求電文	取引要求電文に対するメッセージ認証コード(MAC1)	カードリーダ，暗号化ピンパッド，現金処理モジュール，のいずれか
S3	返信電文受信	返信電文	返信電文に対するメッセージ認証コード(MAC2)	勘定系ホスト
S4	現金処理	現金出金コマンドの出金額	ホスト承認金額	S2 と同じ手順で決定
		現金出金コマンドの伝送時間	返信電文受信時間	S2 と同じ手順で決定

に基づいて出金額は暗号化ピンパッドから取得する．S2 において，既存の磁気カード取引では勘定系ホストとの通信を担う安全なデバイスは存在しないので，検証情報取得デバイスを任意の周辺デバイスに担わせる．ここでは，カードリーダ，暗号化ピンパッド，現金処理モジュールがあげられるが，どのデバイスを選択するかは，次のステップ 3 との整合性の中で決定する．S4 において，現金出金コマンドの真正性検証情報は，取引要求電文中の出金額，ならびに，返信電文中の取引承認フラグをあわせた，ホスト承認金額となる．検証情報取得デバイスは，勘定系ホストとの通信を担うデバイスなので S2 と同じ手順で決定する．現金出金コマンドの伝送時間の真正性検証情報は，勘定系ホストからの返信電文受信時間である．現金処理モジュールで現金出金コマンドを受信した時間と，返信電文受信時間の差分が基準より長ければ出金妨害があったと判断する．検証情報取得デバイスは S2 と同じ手順で決定する．

ステップ 3：基本的枠組みの推奨実装モデルを用いて，保護資産アクセスコマンドの真正性を検証するデバイスと，検証済コマンドを実行するデバイスを決定する．

ガイダンス 3-1，3-2，3-3 に従い，3.1 節で推奨された実装モデル 1，4 が適用候補として選択される．ステップ 2 で述べたホスト承認金額を電文から抽出するには，電文の構文解析というアプリケーションの機能が必要である．そのため，ガイダンス 3-3 に従い，カードリーダ，暗号化ピンパッド，現金処理モジュールの中から，いずれか 1 つを勘定系ホストとの通信を担うデバイスとして選択する．カードリーダを選択した場合に，各取引サブプロセスの整合性を確保するデータの流れと適用実装モデルを**図 7** に示す．図ではデバイスの提供機能ごとにデータの流れを描いている．ガイダンス 3-4 に従い，取引サブプロセスごとに次の検証機能が実装される．

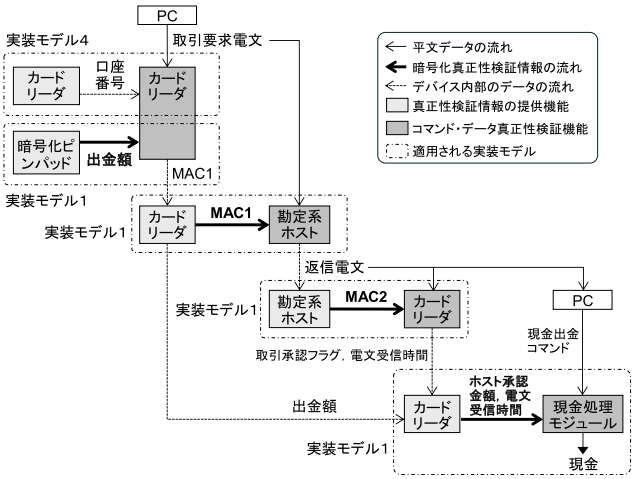

図 7 取引サブプロセス間の整合性を確保するデータの流れ
Fig. 7 Data flow ensuring consistency among transaction sub-processes.

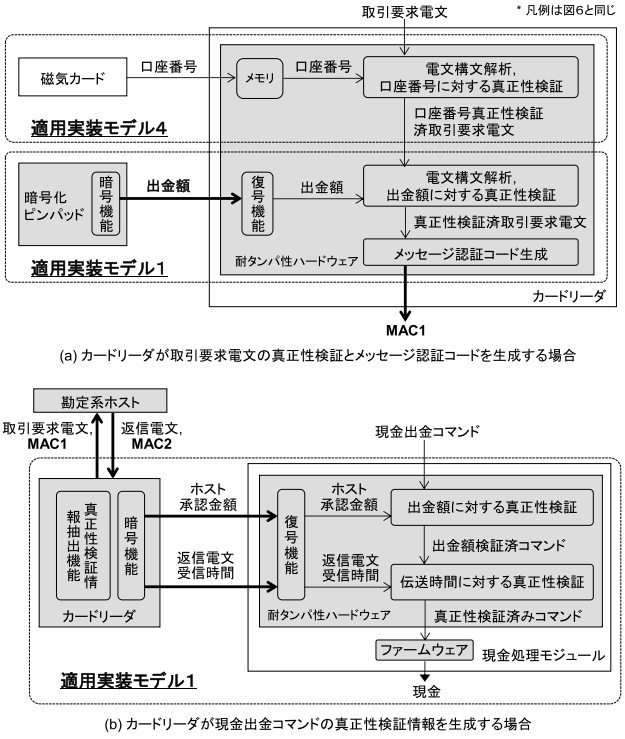

図 8 カードリーダが勘定系ホストと通信する場合の実装
Fig. 8 Implementation examples of card reader communicating with the host computer.

(S1) 取引要求電文生成

カードリーダは，内部で転送された口座番号と，暗号化ピンパッドから受信した暗号化出金額を用いて，取引要求電文の真正性を検証したうえで，メッセージ認証コード（MAC1）を生成する．これは 1 つのデバイスで真正性検証情報取得と検証を行う実装モデル 4 と，他のデバイスから真正性検証情報を受信する実装モデル 1 の組合せである（**図 8** (a)）．

(S2) 取引要求電文送信

PC は取引要求電文を，カードリーダは MAC1 をそれぞ

表 5 適用される実装モデルのまとめ
Table 5 Summary of applied implementation models.

(S1) 取引要求電文生成

保護資産		口座番号，出金額（USB/RS-232C ケーブル，PC 上）	
真正性検証情報		磁気カード上の口座番号	入力された出金額
検証情報取得デバイス		カードリーダ	暗号化ピンパッド
検証実行デバイス	カードリーダ	実装モデル 4	実装モデル 1
	暗号化ピンパッド	実装モデル 1	実装モデル 4
	現金処理モジュール	実装モデル 1	実装モデル 1

(S2) 取引要求電文送信

保護資産		取引要求電文		
真正性検証情報		MAC1		
検証情報取得デバイス		カードリーダ	暗号化ピンパッド	現金処理モジュール
検証実行デバイス	勘定系ホスト	実装モデル 1	実装モデル 1	実装モデル 1

(S3) 返信電文受信

保護資産		返信電文		
真正性検証情報		MAC2		
検証情報取得デバイス		勘定系ホスト		
検証実行デバイス	カードリーダ	実装モデル 1		
	暗号化ピンパッド	実装モデル 1		
	現金処理モジュール	実装モデル 1		

(S4) 現金処理

保護資産		現金出金コマンドの出金額		
真正性検証情報		ホスト承認金額		
検証情報取得デバイス		カードリーダ	暗号化ピンパッド	現金処理モジュール
検証実行デバイス	現金処理デバイス	実装モデル 1	実装モデル 1	実装モデル 1
保護資産		現金出金コマンドの伝送時間		
真正性検証情報		返信電文受信時間		
検証情報取得デバイス		カードリーダ	暗号化ピンパッド	現金処理モジュール
検証実行デバイス	現金処理デバイス	実装モデル 1	実装モデル 1	実装モデル 1

れ送信し，勘定系ホストはそれらを用いて受信電文を検証する．この処理は，実装モデル 1 に分類され，カードリーダが検証情報取得デバイス，勘定系ホストがコマンド検証・実行デバイスとなる．

(S3) 返信電文受信

勘定系ホストは返信電文とメッセージ認証コード（MAC2）を送信し，カードリーダは受信電文の真正性を検証する．この処理は，実装モデル 1 に分類され，勘定系ホストが検証情報取得デバイス，カードリーダがコマンド検証・実行デバイスとなる．

(S4) 現金処理

カードリーダは，取引要求電文内の出金額と返信電文内の取引承認フラグから，ホスト承認金額を生成する．また，カードリーダが返信電文を受信した時間を，返信電文受信時間とする．カードリーダはそれらを暗号化して，現金処理モジュールに送信する．この処理は，実装モデル 1 に分類され，これはカードリーダが検証情報取得デバイス，現金処理モジュールがコマンド検証・実行デバイスとなる（図 8 (b)）．勘定系ホストとの通信を担うデバイスとして，暗号化ピンパッド，現金処理モジュールを選択した場合も含めて，取引サブプロセスごとに実装モデルを**表 5** にまとめた．表では勘定系ホストとの通信を担う周辺デバイスごとに，適用される実装モデルを色分けしている．

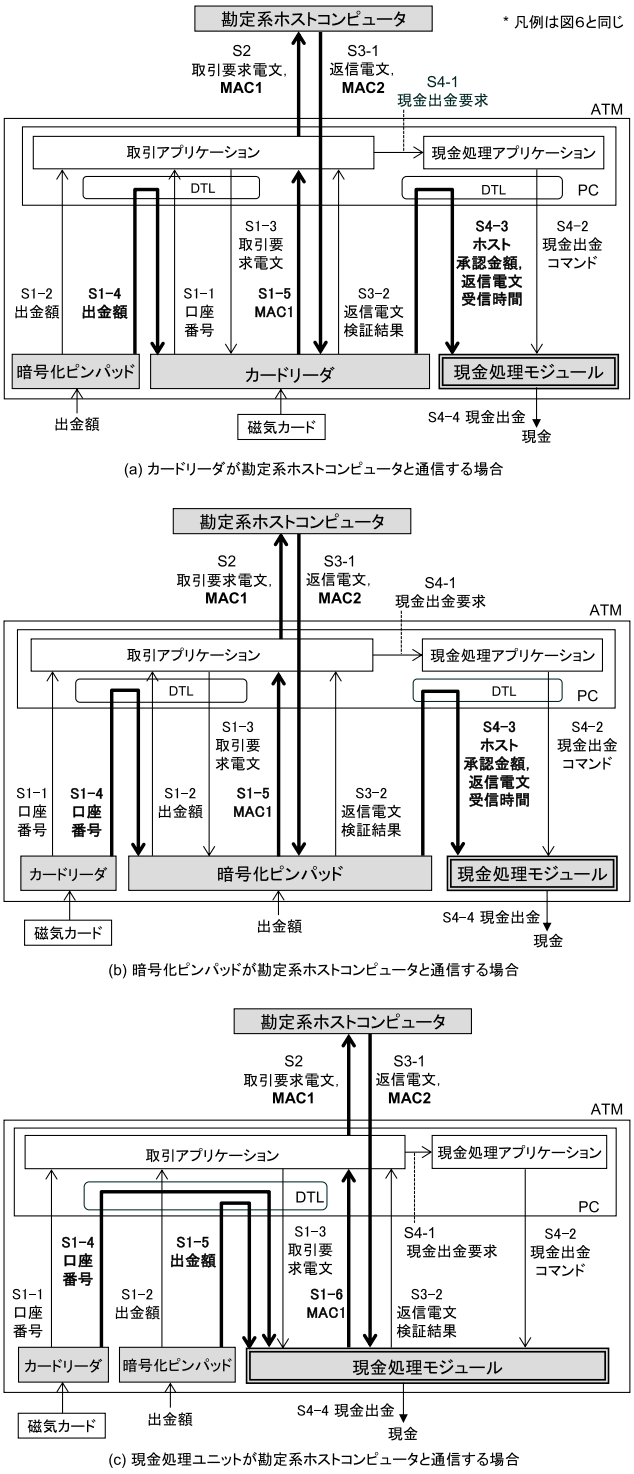

図 9 コマンド真正性検証法の実装例
Fig. 9 Implementation examples of the fundamental framework of “Command Verification”.

4.2 適用システムの詳細データフロー

カードリーダ，暗号化ピンパッド，現金処理モジュールのそれぞれが，勘定系ホストとの通信を担う場合の基本的枠組み適用例を**図 9** に示す．周辺デバイスどうしを直接接続する物理的通信線は存在しないので，周辺デバイス間の暗号通信は，周辺デバイスと PC 間の USB/RS-232C ケーブルを利用する．周辺デバイス間の通信路を提供する

ために，データ転送ライブラリ（以下，‘DTL’）が PC 内に新たに導入され，DTL は CEN/XFS API の下に存在すると想定する．図 9 (a) は，カードリーダが勘定系ホストとの暗号通信を担う場合の適用システムのデータフローを示す．図では暗証番号にかかわるシステムは省略されている．耐タンパ性を持つセキュアエレメントが，カードリーダ，現金処理モジュールにそれぞれ実装され，一方，勘定系ホストにはハードウェアセキュリティモジュールが実装されている．ここでセキュアエレメントとは，たとえば Java Card™ *2でプログラム可能なセキュリティチップである．適用システムにおける暗号通信の暗号鍵管理は機密性，完全性，真正性を満たすため，PCI 規格要件 [20], [21], [23], [25] か，あるいは，EMV 仕様 [18] に準拠し，暗号通信のセッションはあらかじめ生成されていると想定する．図 9 (a) の詳細処理フローを以下に示す．

(S1) 取引要求電文生成

取引アプリケーションがカードリーダから口座番号 S1-1 を，暗号化ピンパッドから出金額 S1-2 をそれぞれ取得し，取引要求電文 S1-3 を生成した後に DTL 経由でカードリーダに送信する．DTL では，暗号化された出金額 S1-4 を暗号化ピンパッドから取得し，取引要求電文 S1-3 とともにカードリーダに送信する．カードリーダは，取得した出金額 S1-4 と内部に保持している口座番号で，取引要求電文の真正性を検証後 MAC1 S1-5 を生成し，取引アプリケーションに送信する．

(S2) 取引要求電文送信

取引アプリケーションは取引要求電文と MAC1 S2 を勘定系ホストに送信する．

(S3) 返信電文受信

勘定系ホストは取引要求電文の真正性を MAC1 で検証し，ATM 利用者の口座残高を確認後に取引承認フラグ含む返信電文と MAC2 S3-1 を ATM に返信する．ATM では受信データをカードリーダに転送し，返信電文の受信時間を記録するとともに，その電文検証後，返信電文検証結果 S3-2 を取引アプリケーションに返信する．

(S4) 現金処理

現金処理アプリケーションは現金出金要求 S4-1 を受信後，現金出金コマンド S4-2 を DTL 経由で現金処理モジュールに送信する．DTL では，カードリーダからホスト承認金額と返信電文受信時間 S4-3 を暗号化された形で取得し，現金出金コマンド S4-2 とともに，現金処理モジュールに送信する．現金処理モジュールでは S4-3 のデータを用いて現金出金コマンドの真正性を検証後に現金出金 S4-4 を行う．

図 9 (b) は暗号化ピンパッドが勘定系ホストとの暗号通信を担う場合の適用例であり，暗号化ピンパッドとカードリーダの役割が図 9 (a) と逆になっている．図 9 (c) は現金処理モジュールが勘定系ホストとの暗号通信を担う場合の適用例である．これら適用例の詳細フローは省略する．図 9 の中で最も推奨される適用例は，既存システムや運用の詳細仕様に依存するので，本論文の議論の対象外とする．

*2 Java は，Oracle Corporation およびその子会社，関連会社の米国およびその他の国における登録商標です．文中の商品名等は各社の商標または登録商標である場合があります．

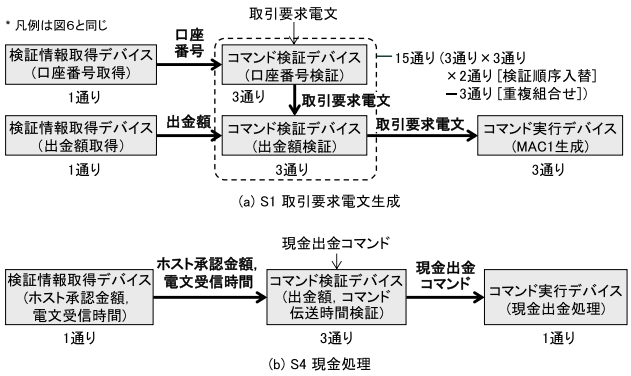

図 10 周辺デバイス組合せ数見積りの考え方
Fig. 10 Combination numbers of peripheral devices.

4.3 提案設計法の評価

提案設計法の効果を評価するために，基本的枠組みを単純に磁気カード出金取引へ適用した場合の適用システムの全組合せ数を見積もる．図 6 (b) の実装モデル 2 は，基本的機能を持つ 3 デバイスから構成されるので，本見積りのために使用する．見積りは 2 つのステップからなる．第 1 ステップでは，カードリーダ，暗号化ピンパッド，現金処理モジュールの 3 つのデバイスを用いて，各取引サブプロセスにおける，デバイスの組合せ数を見積もる．第 2 ステップでは，各取引サブステップで得られたデバイスの組合せ数を掛け算して，全体のデバイス組合せ数を計算する．

第 1 ステップの見積りは以下のとおりである．**図 10** (a) に示すように，S1 取引要求電文生成では，検証情報取得デバイスは口座番号を出力するカードリーダと，出金額を出力する暗号化ピンパッドである．これらは磁気カード出金取引では固定されているので，デバイスの組合せ数は 1 通りである．コマンド検証デバイスに関して，1 つの取引要求電文に対し口座番号検証デバイスと，出金額検証デバイスが独立に必要なので，3 つのデバイスからそれぞれ選択し，順番に検証を行うので，9 通り（3 通り × 3 通り）の組合せがある．さらに，口座番号と出金額の検証順序を逆にできるので，18 通り（9 通り × 2 通り）の組合せがあるが，口座番号と出金額の検証デバイスが一致する場合は，検証順序を逆にすることに意味がないので，3 通りを除いて 15 通り（18 通り − 3 通り）である．コマンド実行デバイス，すなわち，MAC1 生成デバイスは 3 つのデバイスから独立に選択することが可能である．よって，S1 取引要求電文生成におけるデバイスの全組合せ数は 45 通り（1 通り × 15 通り × 3 通り）である．

S2 取引要求電文送信において，MAC1 の暗号通信セッションは MAC1 生成デバイスと勘定系ホストの間で確立されるので，勘定系ホストとの通信を担うデバイスと MAC1 生成デバイスは一致する必要がある．よって，検証情報取得デバイス，コマンド検証デバイス，コマンド実行デバイスの組合せ数は 1 通りである．S3 返信電文受信も同様に組合せ数は 1 通りである．S4 現金処理において，図 10 (b) に示すように，検証情報取得デバイスは，勘定系ホストとの通信を担うデバイスと一致するので 1 通りである．コマンド検証デバイスは 3 つのデバイスから選択可能なので，組合せ数は 3 通りである．コマンド実行デバイスは，現金処理モジュールなので，組合せ数は 1 通りである．よって，S4 現金処理の組合せ数は 3 通り（1 通り ×3 通り ×1 通り）である．

見積りの第 2 ステップでは，S1〜S4 のデバイスの組合せ数を掛け算して，135 通り（45 通り ×1 通り ×1 通り ×3 通り）である．4.1 節で述べたとおり，提案設計法を用いることで，この組合せ数を 3 通りに減らすことができる．

4.4 提案周辺デバイスのアーキテクチャ

提案する周辺デバイスのアーキテクチャ例を**図 11** に示す．カードと端末間の相互認証用セキュアエレメントの装填スロットが，カードリーダには一般に備え付けられている．提案するカードリーダでは，コマンド真正性検証法に用いるセキュアエレメントを本スロットに装填可能である．また，暗号化磁気ヘッドにもセキュアエレメントが実装されており，磁気ヘッドで読み取った口座番号が，カードリーダ内で不正アクセスされないように，スロットに実装されたセキュアエレメントと暗号通信でつなげられている．そのような構成は PCI 規格 [20], [25] に同様な要件があり現実的である．加えて，制御部のファームウェアはセキュアエレメント内に格納された電子署名により不正改造から保護されていると想定する．制御部 RAM 上のファームウェアも上記 PCI 規格に基づき，その電子署名を用いた自己テストで真正性が保たれているとする．

既存の現金処理モジュールに関して，機能拡張のためのシリアル・インタフェースが備わっていることが多く，セキュアエレメントを実装した回路基板をそのシリアル・インタフェースに接続することが可能である．制御部のファームウェアは，カードリーダと同様に RAM 上を含めてセキュアエレメント内の電子署名で保護されていると想定する．さらに，現金処理モジュールは厳重にアクセス管理されている金庫で保護されており，ファームウェアは論理的にも物理的にも保護されている．暗号化ピンパッドに関して，既存のピンパッドは PCI 要件 [20] に基づきタンパ検知応答回路につながる筐体で保護されている．既存の暗号機能はファームウェアで実装されているので，追加の暗号機能はそのファームウェアを修正することで実装可能である．

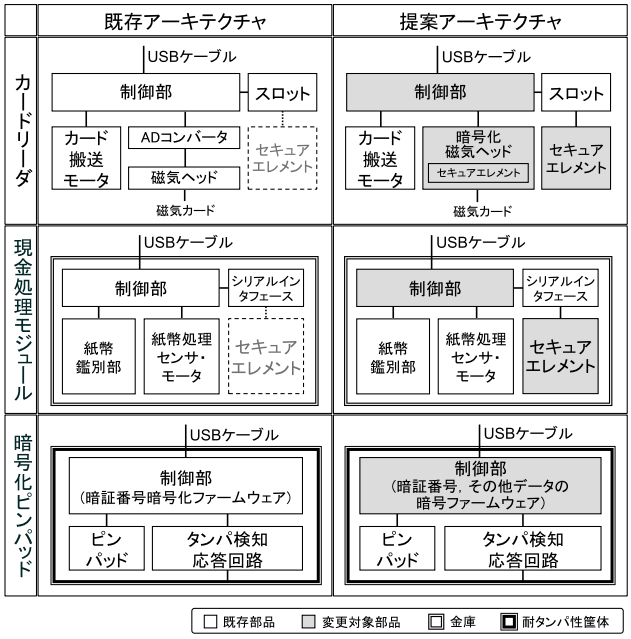

図 11 既存周辺デバイスと提案周辺デバイスの比較
Fig. 11 Comparison of existing devices and proposing devices.

提案アーキテクチャは，デバイス改修量，処理性能，耐障害性についても以下の理由に基づき実現性がある．上記で説明したように，提案アーキテクチャと同様なカードリーダは市販されているうえ，現金処理モジュールや暗号化ピンパッドに関しては，ある国の規格により提案アーキテクチャと同様な実装を求められている．文献 [17] では，図 9 (a) の現金出金コマンドの出金額一致検証についてプロトタイプを開発し，処理時間や実現性を検証済みである．

5. まとめ

本論文では，制御されるデバイス自身が制御コマンドの真正性を検証する，コマンド真正性検証法の基本的枠組みの適用設計法を提案した．本設計法は 3 つの適用ステップからなる．ステップ 1 では，すべての取引サブプロセスにおいて，保護すべき資産とそれを狙う論理攻撃をリストアップする．ステップ 2 では，資産にアクセスするコマンドの検証情報を取得する検証情報取得デバイスを決定する．ステップ 3 では，既存運用への影響や周辺デバイス改造の最小化の面で推奨される実装モデルを用いて，取引サブプロセスにおけるコマンド検証デバイスが，次の取引サブプロセスでは検証情報取得デバイスになるように，コマンド検証・実行デバイスを決定する．これにより，取引サブプロセスを対象とする様々な論理攻撃をシームレスに防御するだけでなく，既存運用や周辺デバイス改造の影響を最小化することが可能になる．分類・モデル化された不正出金攻撃を想定した磁気カード出金取引に本設計法を適用したところ，詳細検討すべき適用候補システムの組合せ数

は 135 から 3 に減らすことができた．本設計法は，磁気カードを用いた ATM の入金・送金取引だけでなく，決済取引に基づいて動作する券売機や自動販売機にも適用できると期待される．これらへの適法は将来の課題である．

謝辞 本研究の成果の一部は，国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務の結果得られたものである．

参考文献

[1] European law enforcement agency: Guidance and recommendations regarding logical attacks on ATMs (2015), available from ⟨https://www.ncr.com/content/dam/ncrcom/content-type/brochures/EuroPol_Guidance-Recommendations-ATM-logical-attacks.pdf⟩ (accessed 2019-07-07).

[2] NCR: ATM SECURITY EXPL A INING ATTACK VECTORS, DEFENSE STRATEGIES AND SOLUTIONS (2018), available from ⟨https://www.ncr.com/content/dam/ncrcom/content-type/white_papers/12518fin-b-atm_security_attack_vectors_and_solutions_update-fin-web.pdf⟩ (accessed 2019-07-07).

[3] Trend Micro Forward-Looking Threat Research (FTR) Team and Europol's European Cybercrime Centre (EC3)：最新脅威解説「ATM マルウェア」(2017)，入手先 ⟨https://appweb.trendmicro.com/doc_dl/select.asp?type=1&cid=239⟩（参照 2019-11-02）.

[4] EUROPOL: 27 arrested in successful hit against ATM Black Box attacks, Press Release (2017), available from ⟨https://www.europol.europa.eu/newsroom/news/27-arrested-in-successful-hit-against-atm-black-box-attacks⟩ (accessed 2019-07-07).

[5] Symantec: Backdoor.Padpin, Press Release, Symantec Security Response (2014), available from ⟨https://www.symantec.com/security_response/writeup.jsp?docid=2014-051213-0525-99&tabid=2⟩ (accessed 2019-07-07).

[6] The European Association for Secure Transactions (EAST): EAST reports 2016 crime stats for Europe's ATMs; black box attacks up 287 percent (2017), available from ⟨https://www.atmmarketplace.com/news/east-reports-2016-crime-stats-for-europes-atms-black-box-attacks-up-287-percent/⟩ (accessed 2019-07-07).

[7] Kaspersky Lab.: Tyupkin Virus (Malware) | ATM Security, available from ⟨https://www.kaspersky.com/resource-center/threats/tyupkin-malware-atm-security-malware⟩ (accessed 2019-07-07).

[8] Symantec Official Blog: Backdoor.Ploutus Reloaded – Ploutus Leaves Mexico (2013), available from ⟨http://www.symantec.com/connect/blogs/backdoorploutus-reloaded-ploutus-leaves-mexico⟩ (accessed 2019-07-07).

[9] The Times of India: ATM JACKPOT WITH MALWARE, TIMES NATION |Politics & Policy (2015), available from ⟨http://www.pressreader.com/india/the-times-of-india-mumbai-edition/20150509/282003260992233⟩ (accessed 2019-07-07).

[10] China Zhijian Publishing House: GA 1280-2015, Security requirements for automatic teller machines (in Simplified Chinese) (2015), available from ⟨http://www.bzcbs.cn/produce/showonebook.asp?strid=77844⟩ (accessed 2019-07-07).

[11] ATM marketplace: ATMs left behind as Windows XP support ends (2014), available from ⟨https://www.atmmarketplace.com/articles/atms-left-behind-as-windows-xp-support-ends/⟩ (accessed 2019-07-07).

[12] An NCR white paper: NCR LOGICAL SECURITY (2017), available from ⟨https://www.ncr.com/content/dam/ncrcom/content-type/brochures/17fin5025_a_sec_rqts_protect_logical_attacks_wp.pdf⟩ (accessed 2019-07-14).

[13] Bräuer, J., Gmeiner, B. and Sametinger, J.: A Risk Assessment of Logical Attacks on a CEN/XFS-based ATM Platform, *International Journal on Advances in Security*, Vol.9, No.3&4, pp.122–132, ISSN 1942-2636 (2016).

[14] Kai, S., Ishiakawa, T., Ogata, H., Miyamoto, N. and Sanada, T.: Accelerating Global Business through ATM Security Practices, *IPSJ Digital Practice*, Vol.9, No.3, pp.700–715 (2018), available from ⟨https://ipsj.ixsq.nii.ac.jp/ej/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=190389&item_no=1&page_id=13&block_id=8⟩.

[15] CEN: Extensions for Financial Services (XFS) interface specification Release 3.30–Part 1: Application Programming Interface (API)–Service Provider Interface (SPI)–Programmer's Reference, European Committee for Standardization (2015), available from ⟨ftp://ftp.cen.eu/CWA/CEN/WS-XFS/CWA16926/CWA%2016926-1.pdf⟩ (accessed 2019-07-14).

[16] ISO 8583-1:2003, ISO 8583-2:1998, ISO 8583-3:2003, Financial transaction card originated messages – Interchange message specifications, available from ⟨https://www.iso.org/standard/31628.html, https://www.iso.org/standard/23632.html, https://www.iso.org/standard/35363.html⟩ (accessed 2019-07-14).

[17] Ogata, H., Ishikawa, T., Miyamoto, N. and Matsumoto, T.: An ATM security measure for smart card transactions to prevent unauthorized cash withdrawal, *IEICE Trans. Information and Systems*, Vol.E102-D, No.3, pp.559–567 (2019).

[18] EMVCo, LLC: EMV Integrated Circuit Card Specifications for Payment Systems Book 2 Security and Key Management Version 4.3 (2011), available from ⟨https://www.emvco.com/terms-of-use/?u=wp-content/uploads/documents/EMV_v4.3_Book_2_Security_and_Key_Management_20120607061923900.pdf⟩ (accessed 2019-07-14).

[19] EMVCo: EMV Integrated Circuit Card Specifications for Payment Systems Book 3 Application Specification, Version 4.3 (2011), available from ⟨https://www.emvco.com/wp-content/uploads/documents/EMV_v4.3_Book_3_Application_Specification_20120607062110791.pdf⟩ (accessed 2019-07-14).

[20] PCI SSC: Payment Card Industry (PCI) PIN Transaction Security (PTS) Point of Interaction (POI) Modular Security Requirements Version 5.1 (2018), available from ⟨https://www.pcisecuritystandards.org/documents/PCI_PTS_POI_SRs_v5-1.pdf⟩ (accessed 2019-07-14).

[21] PCI SSC: Payment Card Industry (PCI) PIN Security Requirements Version 2.0 (2014), available from ⟨https://www.pcisecuritystandards.org/documents/PCI_PIN_Security_Requirements_v2__Dec2014_b.pdf⟩ (accessed 2019-07-14).

[22] ISO 9564-1:2017, ISO 9564-2:2014, Financial services–Personal Identification Number (PIN) management and security, (accessed https://www.iso.org/standard/68669.html, https://www.iso.org/standard/61448.html) (accessed 2019-07-14).

[23] PCI SSC: Payment Card Industry (PCI) PIN Transaction Security (PTS) Hardware Security Module (HSM) Modular Security Requirements Version 3.0 (2016),

available from ⟨https://www.pcisecuritystandards.org/documents/PCI_HSM_Security_Requirements_v3_2016_final.pdf⟩ (accessed 2019-07-14).
[24] IOActive, Inc.: IOActive Security Advisory (2017), available from ⟨https://ioactive.com/pdfs/ATM_security-advisory_FINAL_v4-davis_cm.pdf⟩ (accessed 2019-07-07).
[25] PCI SSC: Payment Card Industry (PCI) Point-to-Point Encryption: Solution Requirements and Testing Procedures Version 3.0 (2019), available from ⟨https://www.pcisecuritystandards.org/documents/P2PE_v3.0_Standard.pdf⟩ (accessed 2020-01-23).

緒方 日佐男 （正会員）

1989 年九州大学大学院理学研究科物理学専攻修士課程修了．同年 (株) 日立製作所入社，中央研究所配属．ニューラルネットワークとパターン認識の研究に従事．2004 年日立オムロンターミナルソリューションズ (株) に入社．生体認証と ATM セキュリティの開発に従事．現在，横浜国立大学の博士課程後期課程に所属．電子情報通信学会会員．ISO/IEC JTC1 SC37/WG3 国内幹事．

石川 智祥

2001 年京都大学工学部卒業．同年オムロン (株) に入社．ATM ソフトウェア開発に従事．2004 年日立オムロンターミナルソリューションズ（株）に入社．ATM ソフトウェアとセキュリティ設計に従事．CEN/XFS ワークショップ会員．

宮本 範親

1986 年京都大学院大学工学研究科数理工学専攻修了．同年日本鋼管（株）入社．鋼鉄のプロセス制御システムの開発に従事．1987 年オムロン（株）に入社．UNIX ワークステーションソフトウェアとマルチプロセッサシステムの OS の開発に従事．2004 年日立オムロンターミナルソリューションズ（株）に入社．ATM デバイスと海外事業戦略立案に従事．

松本 勉

1986 年東京大学大学院工学系研究科電子工学専攻博士課程修了．工学博士．同年より横浜国立大学勤務．現在，同大学・環境情報研究院教授および先端科学高等研究院情報・物理セキュリティ研究ユニット主任研究者および産業技術総合研究所サイバーフィジカルセキュリティ研究センター長．CRYPTREC 暗号技術検討会座長，日本学術会議連携会員を兼任．情報・物理セキュリティの研究教育に 1981 年より従事．この間，日本銀行金融研究所客員研究員，独カールスルーエ大学客員教授，日本学術振興会学術システム研究センター専門研究員，国際暗号学会 IACR 理事等を歴任．暗号学国際会議 ASIACRYPT，電子情報通信学会暗号と情報セキュリティシンポジウム SCIS，バイオメトリクス研究専門委員会，ハードウェアセキュリティ研究専門委員会等の創設に貢献．電子情報通信学会業績賞，第 5 回ドコモ・モバイル・サイエンス賞，第 4 回情報セキュリティ文化賞，2010 年文部科学大臣表彰・科学技術賞等受賞．