IMES DISCUSSION PAPER SERIES

モバイル端末による金融サービスの安全性を高めるために：セキュア・エレメント等の活用

宇根正志（うね まさし）・廣川勝久（ひろかわかつひさ）

Discussion Paper No. 2017-J-15

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES

BANK OF JAPAN

日本銀行金融研究所

〒103-8660 東京都中央区日本橋本石町 2-1-1

IMES Discussion Paper Series 2017-J-15
2017 年 10 月

モバイル端末による金融サービスの安全性を高めるために：セキュア・エレメント等の活用

宇根正志（うね まさし）*・廣川勝久（ひろかわかつひさ）**

要　旨

近年、金融サービスにおいてスマートフォン等のモバイル端末を活用する動きが一段と高まっている。こうしたサービスを安全に提供するためには、モバイル端末におけるサービス利用者や取引内容等の確認、すなわち「認証」が重要である。しかし、最近、モバイル端末を標的とする強力なマルウェアの事例が報告されはじめており、そうしたマルウェアによって、モバイル端末上での認証が正しく行われなくなり、認証用のデータや金融取引の内容にかかるデータが盗取されたり、改変されたりする可能性が懸念されている。本稿では、マルウェアによる攻撃の影響を排除する手段として注目されているセキュア・エレメント（Secure Element）と関連する実行環境としてのトラスティッド・エグゼキューション・エンバイロメント（Trusted Execution Environment）について紹介する。また、そうした手法を金融サービスに活用していく際の留意点や課題についても考察する。

キーワード：認証、スマートフォン、セキュア・エレメント、トラスティッド・エグゼキューション・エンバイロメント、マルウェア、モバイル端末

JEL classification: L86、L96、Z00

*　日本銀行金融研究所企画役（E-mail: masashi.une@boj.or.jp）
**　日本銀行金融研究所テクニカル・アドバイザー
（E-mail: katsuhisa.hirokawa@boj.or.jp）

本稿の作成に当たっては、KDDI 株式会社の磯原隆将課長補佐と情報セキュリティ大学院大学の大塚玲教授から有益なコメントを頂いた。ここに記して感謝したい。ただし、本稿に示されている意見は、筆者たち個人に属し、日本銀行の公式見解を示すものではない。また、ありうべき誤りはすべて筆者たち個人に属する。

目　　次

１．はじめに

近年、スマートフォンやタブレット端末（以下、まとめてモバイル端末という）を通じた金融サービスの提供が一段と活発化している。例えば、「モバイル・バンキング」に加え、複数の金融機関に保有する口座の取引データを加工・提供する「口座情報サービス」や「決済指図伝達サービス」等の新しい金融サービスがモバイル端末を通じて提供されている（中村［2017］）。また、モバイル端末は、電子マネーや消費者信用の媒体としても活用されている（日本銀行決済機構局［2017］）。こうしたなか、モバイル端末を用いたサービス利用者の本人確認や取引内容の確認、すなわち「認証」が、安心安全な金融サービスを実現するうえで一層重要となっている[1]。

その一方、モバイル端末を対象とするマルウェアによる攻撃が、近年、益々高度化している。例えば、マルウェアによって内部のデータの盗取やアプリケーション・ソフトウェアの改変等を試みる攻撃が典型的である（Marczak and Scott-Railton [2016]、Pan [2016]、大塚ほか［2016］、Taylor and Martinovic [2017]）[2]。その他、モバイル端末の動作状況を詳細に観察することで、端末内部の暗号用の鍵を効率的に推測することが可能であるとの研究結果も報告されている（例えば、Lipp *et al.* [2016]、Timmers and Spruyt [2016]、Irazoqui and Guo [2017]）。こうしたマルウェアがさらに高度化し、金融サービス用のアプリケーション・ソフトウェアにおける認証処理を攻撃することが可能になれば、サービス利用者が入力する認証用データ（暗証番号や生体情報等）が盗取されたり、金融取引のデータが改ざんされたりするリスクが生ずる。

上記のような攻撃に対抗する方法として、セキュア・エレメント（Secure Element: SE）を活用するアプローチが注目を集めている[3]。SE は、暗号処理等のセキュリティ機能を有するとともに、外部からの物理的な攻撃に対しても高い安全性を有するモジュールの総称であり、ハードウェアとソフトウェアを組み合わせて実現される[4]。モバイル端末上に SE を装備することによって、通常の

---

1 認証の具体的な手法の選択は、一般に、当該取引のリスクの多寡や実装・運用にかかる費用等に基づき決定される。比較的高額な金融取引では、利用者認証や取引認証に加えて、サービス利用者による取引実行の意思を追加的に確認するケースが考えられる。本稿では、検討をより簡略化し理解しやすくするために、取引実行の意思確認については検討対象外とする。

2 モバイル端末の代表的な OS である Android OS や iOS において、近年各種の脆弱性が報告され、それらを悪用するマルウェアも多数報告されている（トレンドマイクロ［2017］）。

3 例えば、Ahmad *et al.* [2013]、Elenkov [2015]、Ortiz-Yepes [2016]、磯原・竹森・本間［2016］、大塚ほか［2016］、European Union Agency for Network and Information Security [2016b]、International Telecommunication Union [2017]が挙げられる。

4 モバイル端末用の SE は、端末所持者（携帯電話加入者）の ID や端末の認証等に用いられる暗号鍵等を内蔵する UICC（Universal IC Card）、microSD 等の着脱可能なメモリーカード、端末内部に組み込まれる IC チップ（embedded SE と呼ばれる）を活用して実現されることが多い（Elenkov [2015]）。UICC は、SIM（Subscriber Identity Module）とも呼ばれ、通信事業者によっ

実行環境である REE（Rich Execution Environment、例えば Android OS）とは物理的かつ論理的に隔離された、より安全な実行環境を実現することができると期待される。また、SE に関連する実行環境であり、主にソフトウェアによって通常の実行環境から分離された安全な実行環境を実現するトラスティッド・エグゼキューション・エンバイロメント（Trusted Execution Environment: TEE）の開発や実装も進められている。モバイル端末を利用した金融サービスを提供する金融機関等は、マルウェア等による攻撃の高度化に備えて、こうした新しい対策手法の動向をフォローすることが重要であると考えられる。

本稿では、モバイル端末を用いた金融サービスにおける認証にかかる処理をより安全に実現するための技術として、SE に焦点を当てるとともに、関連する実行環境である TEE についても考察する[5]。2 節では、モバイル端末における認証のモデルおよびマルウェアへの対策方針を示す。3 節では、SE と TEE の概要を説明し、4 節では、それらを活用する形態を分類して、各形態における留意点を示す。5 節では、金融機関等が今後 SE や TEE を活用していく際の主な課題を考察する。

２．モバイル端末を利用した金融サービスと認証

本節では、モバイル端末を利用した金融サービスの構成や認証について説明する。そのうえで、モバイル端末上での認証処理を標的とするマルウェアとそれらによる攻撃のリスクや対策方針を説明する。

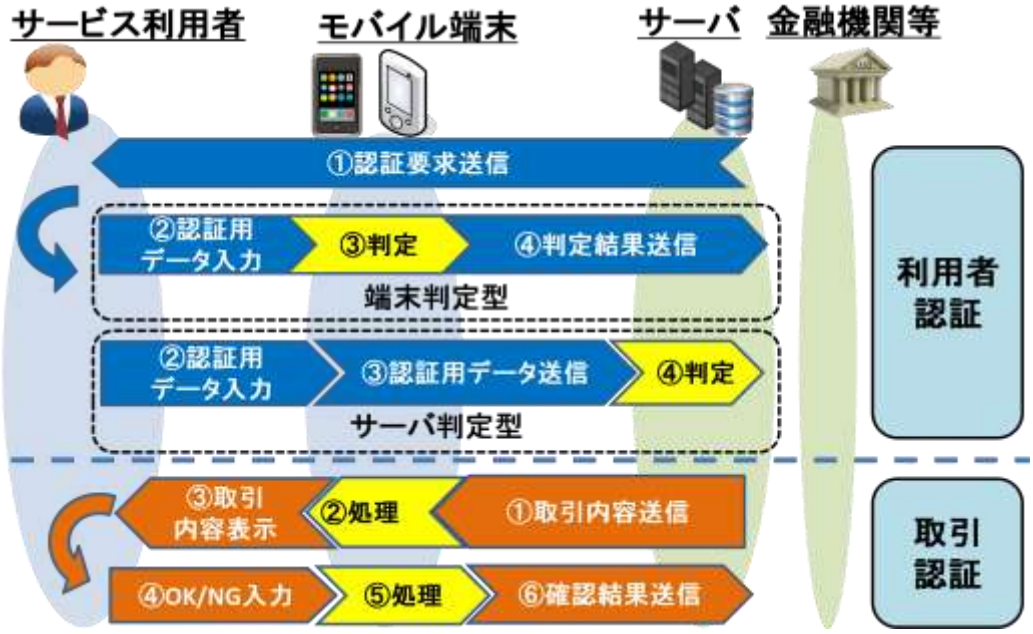
（1）エンティティと認証

モバイル端末で金融サービスを処理するためのシステムは、主に次の 4 つのエンティティから構成される。すなわち、①当該サービスを提供する金融機関や FinTech 企業（以下、金融機関等）、②金融機関等が管理し、サービスの提供にかかる処理を実行する「サーバ」、③金融機関等が提供するサービスを利用する個人である「サービス利用者」、そして、④サービス利用者が有する「モバイ

て発行・管理され、モバイル端末・通信事業者間の通信のほか、複数のアプリケーション・ソフトウェアを動作させる機能を有している。近年、通信事業者が安全な通信路とソフトウェア管理機能を用いた OTA（Over-The-Air）によって遠隔から SIM にアプリケーション・ソフトウェアを導入することが可能となるなどの事情により、新しいソフトウェア導入にかかるコストや利便性が向上した。こうしたことも SIM の活用に注目が集まっている背景の 1 つとみられる。

[5] 安全な金融サービスを実現するためには、モバイル端末の安全性だけでなく、当該サービスの提供にかかるシステム全体の安全性に配慮し、モバイル端末以外にかかるリスクやセキュリティ要件等も検討する必要がある。そうした際のガイドラインやセキュリティ要件集が既に公表されている（例えば、European Union Agency for Network and Information Security [2016a, b]、European Payment Council [2017]、International Telecommunication Union [2017]）。もっとも、モバイル端末における SE の活用等、個々の対策に関しては、最新の研究開発の動向等を踏まえつつ別途検討する必要がある。

図表 1　利用者認証と取引認証のモデル（概念図）



ル端末」から構成される。金融サービスにかかるデータの処理や通信は、サーバとモバイル端末間で行われる[6]。

ここでは、サービス利用者の本人確認（利用者認証）と取引内容の確認（取引認証）という 2 種類の認証処理にフォーカスする。

利用者認証では、サーバからの認証要求がモバイル端末のユーザ・インタフェースを通じてサービス利用者に伝えられ、サービス利用者が認証用データ等をモバイル端末に入力する[7]（図表 1 を参照）。認証用データがモバイル端末で処理された後、モバイル端末で認証の成否が判定されるケース（以下、端末判定型）と、サーバで判定されるケース（以下、サーバ判定型）が考えられる。端末判定型では、認証用データを検証するためのデータが予めモバイル端末に格納され、判定結果がサーバに送信される[8]。

取引認証では、サーバから取引内容のデータがモバイル端末に送信され、モ

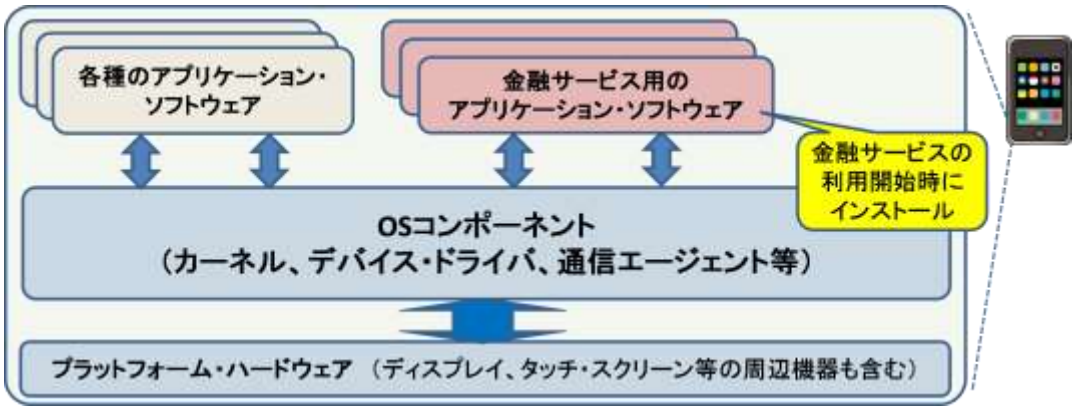
---

6 モバイル端末を通じた金融サービスのユース・ケースやそれらのモデルは European Payment Council [2017]において整理されている。

7 利用者認証の代わりにモバイル端末を認証するケースも考えられる。いずれの認証の有効性も端末内部のデータの取扱いに依存し、利用者認証にかかる要件は、端末認証にかかる要件をカバーすると考えられる。このため、端末認証については本稿では割愛する。

8 生体情報等の機密性が高いデータについては、一般に、取得・生成されたデバイスから他のデバイスにオープンなネットワーク経由で送信することを極力回避することが望ましいとされている（例えば、European Union Agency for Network and Information Security [2016b]）。こうした観点からは、端末判定型が相対的に望ましいといえる。近年注目されている FIDO（Fast IDentity Online）はこのタイプに該当する（FIDO Alliance [2014]、GlobalPlatform [2016a]）。

図表 2　モバイル端末内部の構成（概念図）



バイル端末で処理された後、取引内容がディスプレイに表示される[9]。サービス利用者は、表示内容を確認し、承認するか否かを示すデータ（OK や NG を表すデータ等）を入力する。当該データは、モバイル端末で処理された後、サーバに送信される。

モバイル端末は、主に、①プラットフォーム・ハードウェア（ディスプレイ、タッチ・スクリーン、CPU、メモリ、通信機器等）、②OS コンポーネント（カーネル、デバイス・ドライバ、通信エージェント等）、③アプリケーション・ソフトウェアによって構成される（図表 2 を参照）[10]。通常、金融サービス用のアプリケーション・ソフトウェアは、金融機関等によって作成・準備され、サービス利用者によってインストールされる。また、当該サービスにおける認証処理は、金融サービス用のアプリケーション・ソフトウェアによって実行されている。

（2）保護対象のデータとマルウェアによるリスク

マルウェアがモバイル端末に侵入した場合、認証用データと取引内容の確認にかかるデータを保護することが最重要の課題となる。これらのデータがさらされるリスクはマルウェアの性質に依存する。ここでは、攻撃形態の観点から、井澤・五味［2016］による偽アプリ型と凶悪型の分類を引用し、それぞれのマルウェアによる攻撃とリスクを整理する。

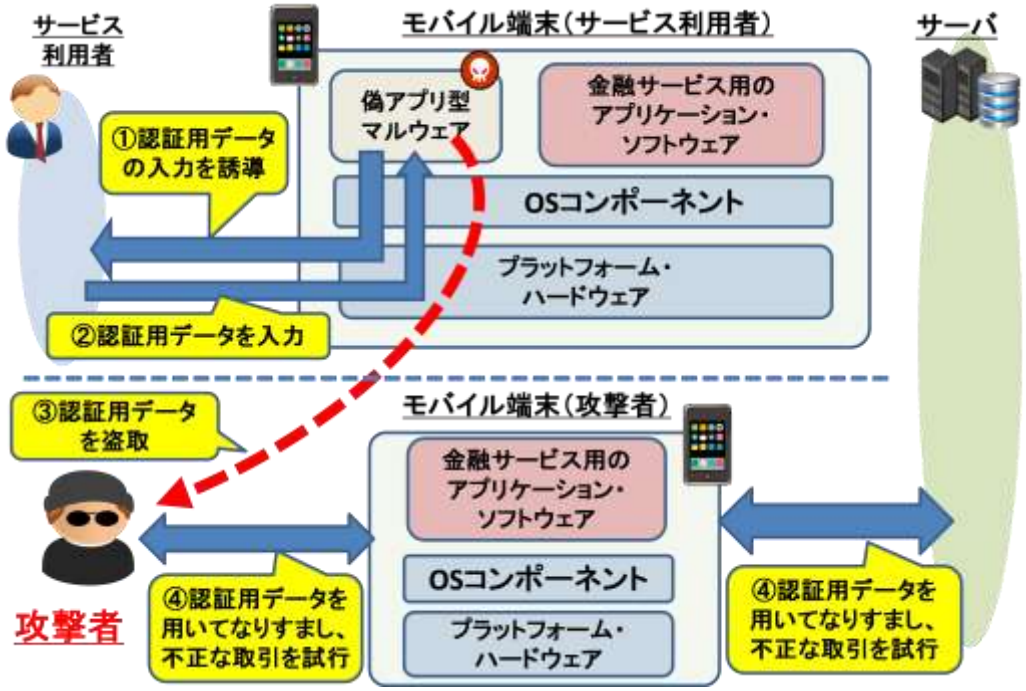
偽アプリ型のマルウェアは、正規の金融サービス用のアプリケーション・ソ

---

9 取引内容にかかるデータをモバイル端末以外のデバイスに送信するケースも考えられるが、サービス利用者の利便性の観点からは、同一のモバイル端末で取引認証を行う手法が望ましい。そうした手法は、FIDO におけるトランザクション確認（Transaction Confirmation）等、既に実装されていることから、ここでも、同一のモバイル端末で取引認証を実行するケースを考える。

10 モバイル端末の実行環境では、通常、アプリケーション・ソフトウェアは他のソフトウェアに直接アクセスできないように制御されている（サンドボックス機能と呼ばれる）。

図表 3　偽アプリ型のマルウェアによる攻撃の流れ（概念図）



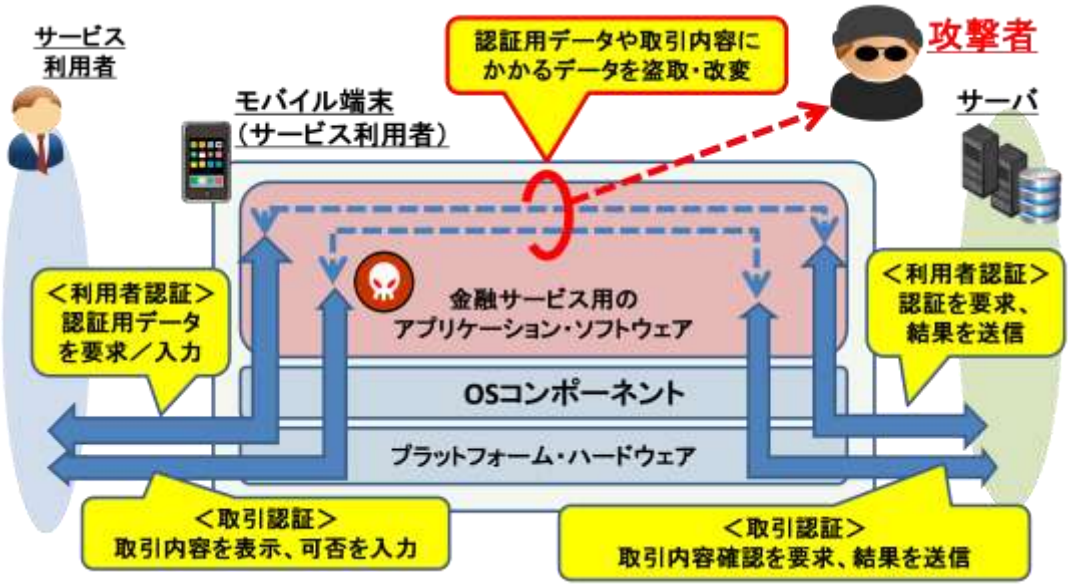
フトウェアとは別のアプリケーション・ソフトウェアとしてインストールされ、正規のソフトウェアのデータにはアクセスできないものと定義される。例えば、正規のソフトウェアに不正なコードを埋め込んで再配布したもの（リパッケージングと呼ばれる）が挙げられる（井澤・五味［2016］、European Union Agency for Network and Information Security [2016a]）。

偽アプリ型のマルウェアは、正規のソフトウェアを装って、認証用データの入力をサービス利用者に促し、入力された認証用データを盗取する可能性がある（図表 3 を参照）。認証用データが盗取されると、利用者認証が適切に機能せず、攻撃者がサービス利用者の端末以外のモバイル端末（金融サービス用のアプリケーション・ソフトウェアがインストールされたもの）に認証用データを入力して、なりすましを成功させるおそれがある。

凶悪型のマルウェアは、正規の金融サービス用のアプリケーション・ソフトウェアのデータにアクセスすることが可能であり、当該ソフトウェアで処理されるデータの盗取やプログラムの改変を実行できるものと定義される。このタイプのマルウェアは、正規の金融サービス用のアプリケーション・ソフトウェアを装ってインストールされることもあれば、全く別のソフトウェアとしてインストールされることもある[11]。

[11] 例えば、マルウェアが OS コンポーネントの管理者権限を奪取し、金融サービス用のアプリケーション・ソフトウェアのデータ等にアクセスする場合が考えられる。また、金融サービス用

図表 4　凶悪型のマルウェアによる攻撃の流れ（概念図）



凶悪型のマルウェアは、偽アプリ型と同様に、利用者認証にかかる認証用データを盗取する可能性がある（図表 4 を参照）。また、サービス利用者が入力した取引内容のデータや、サーバがサービス利用者に向けて送信したデータを改変する可能性もある。取引認証においては、サーバからサービス利用者に送信された取引内容を、モバイル端末のディスプレイへの表示時に改変したり、サービス利用者が入力した取引可否を示すデータを改変したりする可能性が考えられる。

（3）マルウェアによる攻撃への対策方針

偽アプリ型のマルウェアによる、認証用データの盗取やなりすましのリスクに対しては、不正なアプリケーション・ソフトウェアのインストールを防ぐこと（対策方針 1）が第 1 に求められる。例えば、アプリケーション・ソフトウェアの作成者や当該ソフトウェアに対する不正な改変の有無を検証するために、デジタル署名（コード署名と呼ばれる）等を当該ソフトウェアに付与するなどの対応が考えられる[12]。同時に、コード署名が付与されていないソフトウェアを極力インストールしないようにサービス利用者に促すことも、こうした対応に含まれる。

のアプリケーション・ソフトウェアが改変され、それが正規のものを装ってインストールされる場合も想定される。

12 コード署名は、当該ソフトウェアの作成者によって生成される場合のほか、当該ソフトウェアを利用して金融サービスを提供する主体（金融機関等、あるいは、そのサーバ）によって生成される場合も考えられる。

もっとも、こうしたコード署名等による対応がモバイル端末によっては困難なケースもありうるほか、ソフトウェアの不正な改変が検知できずに正規のソフトウェアとして配布されるケースもありうると考えられる。

この場合、コード署名の検証は無効となることから、追加的な対応として、モバイル端末とサービス利用者の対応関係を検証する手段を利用すること（対策方針 2）が考えられる。例えば、モバイル端末内部に、当該端末を所有するサービス利用者と対応付けられた鍵を格納しておき、認証時に、認証用データや当該取引に固有のデータ（例えば、取引日時データ）等のデジタル署名を生成し、当該署名とサービス利用者の対応関係をサーバが検証するという方法がありうる。

凶悪型のマルウェアに対しては、正規の金融サービス用のアプリケーション・ソフトウェアと同じパーミッションを有し、認証用データ等にアクセスできることから、上記の対策方針 1、2 では、攻撃によるリスクを十分に軽減できるとはいえない。そのため、認証にかかる処理を、当該マルウェアの影響が極力及ばない環境で実行すること（対策方針 3）といった対応が求められる。具体的なアプローチとして、ハードウェアを組み合わせた SE の活用が注目を集めている。このアプローチを採用するには専用のハードウェアが必要となるものの、今後、凶悪型のマルウェアによる攻撃がさらに高度化していく可能性に鑑みると、こうした対応が益々重要になってくると考えられる。

３．セキュア・エレメント（SE）等とその活用条件

本節では、SE および関連する実行環境としての TEE の概要について説明する。そのうえで、モバイル端末での活用条件を整理し、凶悪型のマルウェアによる攻撃において活用条件がどう充足されるかを検討する。

（1）SE

イ．認証処理に関する 3 つの条件

SE の定義は文献によって区々であるが、共通点を抽出すると、「（複数の）アプリケーションをその内部で実行する機能を有するモジュールであり、記録された重要情報の保護と暗号や認証等のセキュリティにかかる処理を実行するとともに、ハードウェアに基づく耐タンパー性によって、外部からの物理的な攻撃に対しても一定の安全性を確保できるもの」と表現することができる（Elenkov [2015]）[13]。凶悪型のマルウェアによる影響を排除するために、SE をモバイル端

[13] SE では、上記のような仕組みの実装の適切性を公的機関によって評価・認証することが可能である。例えば、コモン・クライテリア（Common Criteria）に基づく評価・認証や、CMVP（Cryptographic Module Validation Program）／JCMVP（Japan Cryptographic Module Validation

末に搭載し、認証処理を実行することが考えられる。

SE による認証処理が有効となるためには、以下の 3 点を満たすことが求められる。

・条件①：SE 内部で認証にかかる処理を行うアプリケーション・ソフトウェア（以下、SE アプリ）が改変されないこと。

・条件②：SE アプリとサーバの間の通信データが盗取・改変されないこと、または暗号化等によって保護されること。

・条件③：SE アプリとサービス利用者の間の通信データが盗取・改変されないこと。

これらの条件がモバイル端末の SE においてどのように充足されるかを、モバイル端末における代表的な SE である SIM（UICC）を前提に以下で説明する[14]。

ロ．SIM の構成と管理機能

SIM 内部の SE アプリのインストールや削除等は、同じく内部に組み込まれる管理用ソフトウェアであるセキュリティ・ドメイン（Security Domain: SD）によって制御される（図表 5 を参照）[15]。通信事業者が自身の SD を SIM に導入することに加え、SIM を活用するサービス提供主体（以下、サービス提供者）がサービス用の SD を準備し、通信事業者の承認を得てインストールするケースが想定されている（GlobalPlatform [2015a, b]）。金融サービス用の SD の場合、金融機関等が通信事業者の承認のもとで SIM に導入することが考えられる。

SIM に SE アプリをインストールする際には、サービス提供者のサーバとサービス用の SD が相互認証や鍵共有を行った後、サーバが、当該 SE アプリのインストールを承認した証となるデータ（トークンと呼ばれ、サーバの署名等が付与される）を当該 SD に送信する。SD は、トークンを検証し、検証に成功した場合に SE アプリをインストールする。SE アプリの更新等の処理も同様の流れで実施される[16]。

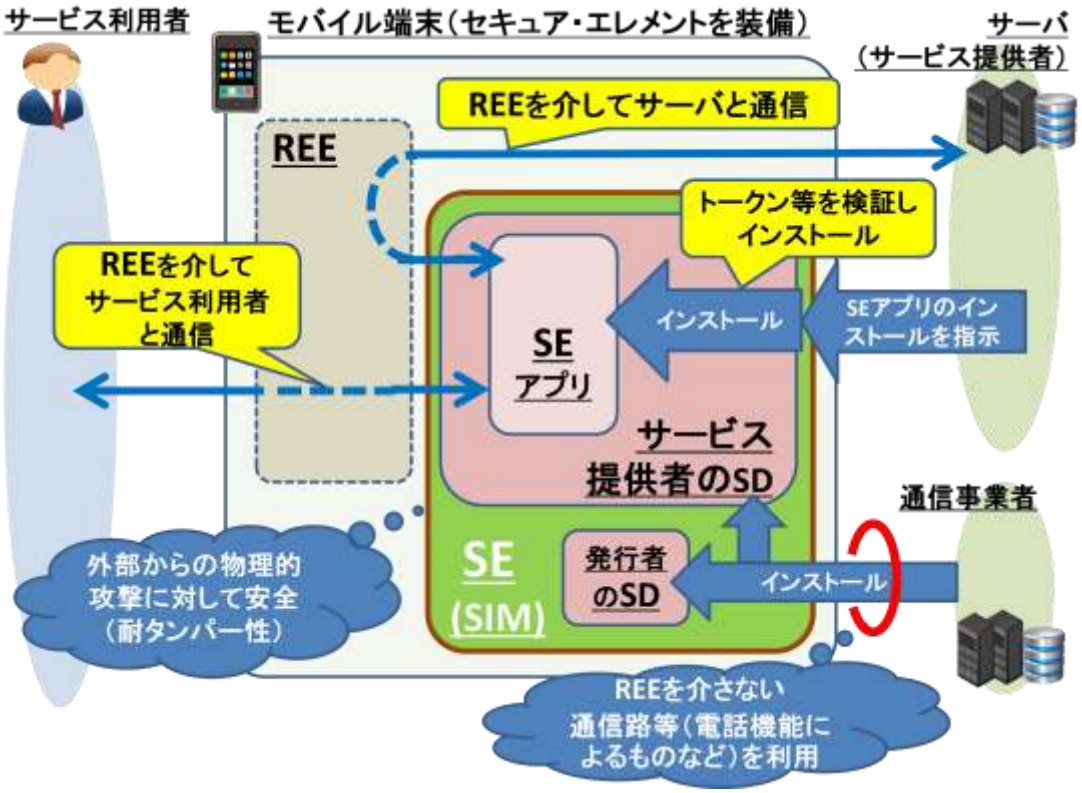
Program）による試験・認証の活用が挙げられる。これらについては 5 節でも触れるほか、詳細については田村・宇根［2008］を参照されたい。

[14] SIM は、通信事業者によって発行され、電話加入者（サービス利用者）の ID、シリアル番号、暗号鍵等を保管するとともに、暗号化や認証等を実行することができる。また、SIM には、内部で複数のアプリケーション・ソフトウェアを動作させる機能も有している。さらに、多くの SIM が GlobalPlatform の技術仕様（Card Specification 等）や SIMalliance の API 仕様（Open Mobile API）に準拠しているといわれている（Elenkov [2015]、SIMalliance [2014]、GlobalPlatform [2015a]）。

[15] SE アプリは、JavaCard ベースの SIM の場合、アプレットと呼ばれる。

[16] SIM とサーバが相互に認識可能であり、REE を経由しない通信路（例えば、電話機能による通信路）を利用するケース（セキュア・チャネル・プロトコル＜Secure Channel Protocol＞を利用するケース）では、トークンの検証を省略することも想定されている。

図表 5 セキュア・エレメントの機能（概念図）



ハ．認証処理に関する 3 つの条件との関係

イ．の認証処理に関する 3 条件に鑑みると、SE としての SIM は次のように評価できる。

条件①の SE アプリの改変に関しては、SE アプリをアンインストールして不正なソフトウェアをインストールする、あるいは、SE アプリに不正な変更を加えるなどの攻撃が考えられる。これに対しては、サーバと SD（あるいは SIM）の間の相互認証や暗号通信のプロトコル、デジタル署名等の暗号方式、署名生成鍵の管理等に問題がなければ、SE アプリへの攻撃は成功しないと考えられる。

また、条件②の SE アプリとサーバの間の通信データの盗取・改変に関しては、SD が相互認証や鍵共有にかかる暗号処理を SE アプリに提供することが想定されており、これを利用したエンド・ツー・エンドでの暗号化によって、通信データの盗取・改変は困難になると考えられる（GlobalPlatform [2015a]）。

他方、条件③の SE アプリとサービス利用者の間の通信データの盗取・改変に関しては、サービス利用者が SE アプリとエンド・ツー・エンドでの暗号通信を実行することは困難であることから、サービス利用者との通信の安全性をどう

確保するかが留意点となる[17]。

（2）SE に関連する実行環境としての TEE

イ．認証処理に関する 3 つの条件

SE の特徴は、耐タンパー性を有するハードウェアを利用することで、物理的な攻撃に対しても安全性を確保することができるように設計されている点にある。ただし、メモリ等の計算リソースの制約が REE に比べて厳しくなるほか、サービス利用者との通信は REE を介して行う必要がある。こうした問題への対応として TEE を活用するアプローチが注目されている（図表 6 を参照）[18]。TEE は、REE から隔離された実行環境であり、その実現方法に関する各種技術仕様が GlobalPlatform によって策定・公開されている（GlobalPlatform [2013, 2014b, 2016b, 2016c, 2016d, 2017a, 2017d]）。

TEE は、SE と異なり、基本的にソフトウェア・ベースの技術によって実現されると同時に、外部からの物理的な攻撃に対する耐タンパー性は想定されていない。また、TEE を利用するためには、TEE を実装するモバイル端末を準備する必要がある[19]。REE で動作する凶悪型のマルウェアに対抗するために、TEE で動作するアプリケーション・ソフトウェア（信頼されたソフトウェア、Trusted Application: TA）に認証処理を実行させ、それ以外の処理を、REE 上の金融サービス用のアプリケーション・ソフトウェアで実行させることが考えられる 。

その際、マルウェアに対する耐性を確保するために、SE の場合と同様に、以下の 3 点を満たすことが求められる。

- 条件①：認証にかかる処理を行う TA を改変させないこと。
- 条件②：TA とサーバの間の通信データの盗取・改変を防止すること、または暗号化等によって保護すること。
- 条件③：TA とサービス利用者の間の通信データの盗取・改変を防止すること。

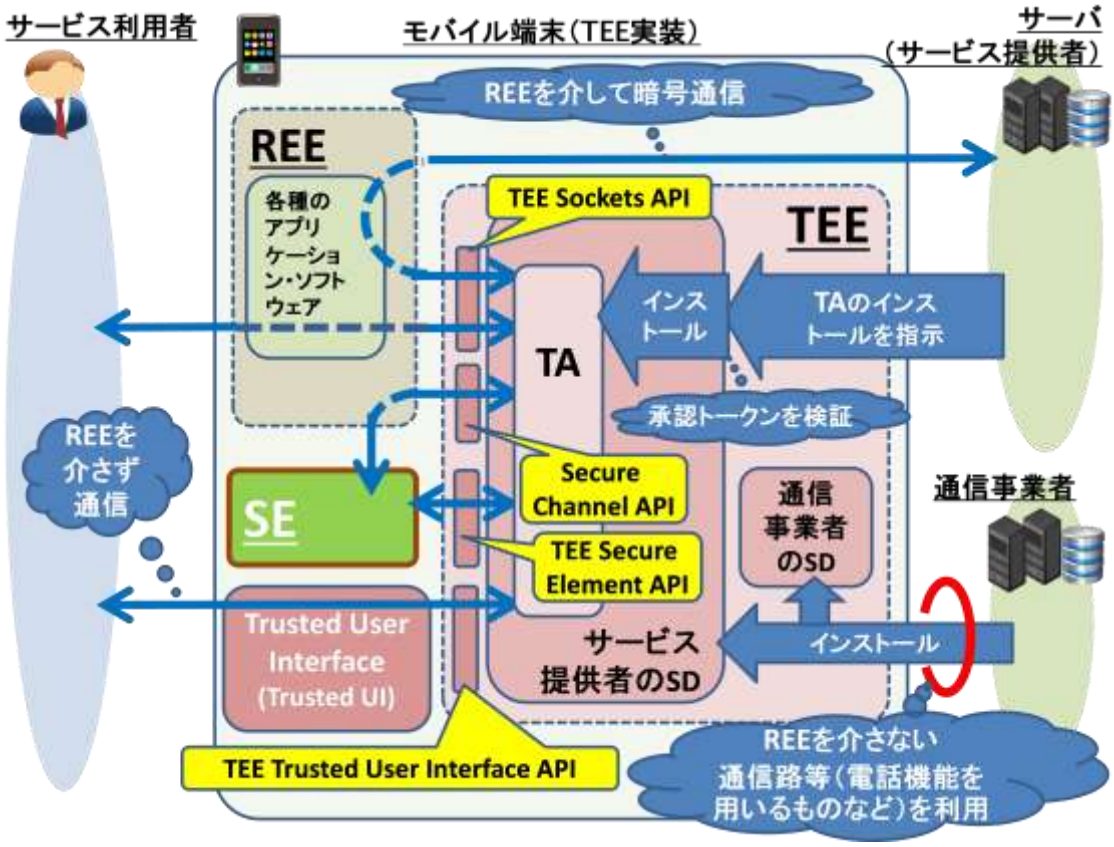
---

[17] 例えば、SIM においては、SE アプリが USAT（Universal Subscriber Identity Module Application Toolkit）を利用してディスプレイへのメッセージ表示や入力データの取得を実行することができる。USAT は、SE アプリによる電話機能を用いた通信接続等を制御するためのプラットフォームであり、REE 上で動作する。SE アプリは USAT を利用してサービス利用者と通信することが可能であるものの、その際は REE を介して通信することになる。もっとも、本節（2）で説明する TEE の Trusted UI を利用できる場合には、REE を介さずにサービス利用者との間で通信することが可能になると考えられる。

[18] TEE という用語は、「REE から隔離された実行環境を実現する技術全般」を指す場合と、「一連の技術仕様によって実現される実行環境」を指す場合がある。本稿では、前者の意味で TEE という用語を使用する。

[19] 一部のスマートフォンでは、TEE 関連の技術仕様に準拠したアーキテクチャや機能が既に採用されているようである（Lu [2015]、Parker [2016]、Trustonic [2015]、吉田ほか［2017］）。

図表 6 TEE の機能（概念図）



ロ．TEE の構成と管理機能

通常のモバイル端末では、REE 上に単一の OS コンポーネントが存在し（Rich OS と呼ばれる。例えば Android OS 等）、サービス利用者がアプリケーション・ソフトウェアをインストールして動作させることができる。一方、TEE を利用するケースでは、モバイル端末内部に、Rich OS に加えて TEE 用の OS コンポーネントが存在する（Trusted OS と呼ばれる）[20]。REE から TEE へのアクセスは Trusted OS によって制御される。

TA のインストールや変更等の管理機能は、SE の場合と同様に、サーバによる

[20] TEE と REE の構成にはさまざまなバリエーションが想定される。GlobalPlatform の TEE System Architecture は、同一のプラットフォーム・ハードウェア上に複数の TEE が存在するケースも例示している（GlobalPlatform [2016d]）。また、TEE 自体については、専用の IC チップ上で実装される例や、メモリ等を REE と共用する例が示されている。また、TEE の一部として動作するハードウェアには複数のモードが設定され、CPU によって、TEE として動作するモードとそうでないモードを切り替えることが可能なケースも知られている（Ahmad *et al.* [2013]、Umar and Mayes [2017]）。

承認の証となるデータ（承認トークン＜authorization token＞）に基づいて、SD によって制御される仕組みとなっている（GlobalPlatform [2016b]）[21]。承認トークンにはサーバによるデジタル署名が付与されており、SD は、その署名を検証することによって承認トークンの正当性を確認する。

ハ．認証処理に関する 3 つの条件との関係

イ．の認証処理に関する 3 条件に鑑みると、TEE を次のように評価できる。

条件①の TA の改変に関しては、マルウェアが TA の改変等を実行するためには、攻撃者がサーバになりすまして当該 SD と相互認証を行った後、承認トークンを偽造して SD に送信しなければならない[22]。したがって、サーバと SD の間の相互認証や暗号通信のプロトコル、承認トークンの生成・検証に用いられるデジタル署名等の暗号方式、署名生成用の鍵の管理等に問題がなければ、攻撃は成功しないと考えられる。

次に、条件②の TA とサーバの間の通信データの盗取・改変に対しては、SE の場合と同様に、データをエンド・ツー・エンドで暗号化して通信する、または、マルウェアによる影響が及ばない通信路を利用する（セキュア・チャネル・プロトコル）といった対策が考えられる。例えば、TA が TEE Sockets API 等を利用して外部のエンティティと暗号通信路（TLS 等を利用）を確立することが考えられる（GlobalPlatform [2016d, 2017a, 2017d]）。このほか、TEE と SE が接続されている場合、TA は SE 経由でサーバと通信することも考えられる。すなわち、TA は、SE アプリとの間で通信するためのインタフェースである TEE Secure Element API を利用して SE アプリと通信し、続いて、SE アプリとサーバの間の通信路を介して通信するというものである（GlobalPlatform [2016c]）。SE が REE に接続されており、TA が REE を介して SE と通信するケースもある。その場合、Secure Channel API を利用して暗号通信路を確立したうえで通信することも想定されている。

条件③の TA とサービス利用者の間の通信データの盗取・改変に関しては、サービス利用者が TA と直接暗号通信を実行することは困難であることから、REE を介さない通信路を別途準備するなどの対応が必要となる可能性がある。こうした場合でも安全な通信路として、TEE にはオプションとして Trusted UI

---

[21] 特に、SD は、TA のインストール／アンインストール、更新（アップデート）、ロック／アンロック等、TA にかかる重要な処理を担う。各 SD はそれぞれ固有の鍵を保持し暗号処理を実行可能であり、（モバイル端末外部の）TA の管理主体と相互認証して暗号通信路を確立する機能も有する。GlobalPlatform [2016d]には、複数の TA が TEE で動作する場合に、種類や管理主体が異なる TA に対応する SD がそれぞれ導入されるケースも規定されている。

[22] 正当な SD をアンインストールし代わりに不正な SD をインストールするという方法も考えられる。もっとも、その場合、当該 SD を管理する上位の SD を不正に操作することが必要である。

（Trusted User Interface）があり、そのためのインタフェースとして TEE Trusted User Interface API が規定されている（GlobalPlatform [2013, 2016d]）[23]。この API は、TA がモバイル端末のタッチ・スクリーン等を介してサービス利用者とデータを安全に交信するための機能を提供する[24]。TEE Trusted User Interface API を利用するための機能やデバイスを備えたモバイル端末を利用できる場合であれば、Trusted UI によるサービス利用者との通信を活用することが考えられる。

４．SE 等によるモバイル端末上での認証の実行形態

モバイル端末上での金融サービスにおいて SE 等を活用して認証処理を実行する場合、認証の実行形態として複数のバリエーションが想定される。今後、それらの実行形態が新しいモバイル端末上で実現され、金融機関等も金融サービスの認証に SE 等を活用できるようになる可能性がある。そのような状況を展望して、以下では、SE 等を活用した認証の実行形態を整理し、各実行形態における対策方針や安全性上の留意点を示す。

（1）認証の実行形態の分類

モバイル端末上での実行環境としては、SE と REE の組合せ、あるいは、SE と REE と TEE の組合せが想定される。また、SE が REE もしくは TEE に装着されるケースも考えられる。これらを踏まえると、実行環境として想定されるのは、①REE と SE の組合せ（以下、RS 型）、②「SE が装着された REE」と TEE の組合せ（以下、RS-T 型）、③REE と「SE が装着された TEE」の組合せ（以下、R-TS 型）の 3 つとなる[25]。なお、SE として SIM を想定する場合、通常のモバイル端末との組合せで実現可能であり、その意味で RS 型の受け皿となる端末は現在広く利用されているといえる。一方、TEE を利用可能なモバイル端末は一部に止まっており、今後の普及が期待される。

上記の 3 種類のうち、RS-T 型と R-TS 型については、認証にかかる処理を担う主体が TA の場合と SE アプリの場合があり、2 つのタイプに分類できる。RS-T

---

[23] TEE Trusted User Interface API の利用が想定されるケースとして、TEE の White Paper では、モバイル端末での金融取引における PIN やパスワードの入力、取引内容やワンタイム・パスワード等の重要な情報の表示と確認等が紹介されている（GlobalPlatform [2015b]）。なお、TEE Trusted User Interface API で想定されるデバイス（タッチ・スクリーン等）は端末に組み込まれる場合、あるいは、有線で接続される場合に限定されるほか、音声デバイスによる入出力やカメラによる入力は対象外とされている。

[24] 例えば、タッチ・スクリーン等の動作時におけるプロセスを TEE が占有し、REE からのアクセスを排除する機能や、TEE Trusted User Interface API による画面がスクリーン上に表示されるタイミングでは、別の画面がその前面に表示されないように制御する機能（オーバー・ディスプレイ攻撃への対策）が想定されている（GlobalPlatform [2013]）。

[25] REE 単独での実行環境もありうるが、凶悪型のマルウェアによる攻撃や対策方針を 2 節で説明していることから、ここでは分類の対象として取り上げない。

図表 7 認証の実行形態の分類

タイプ名	TEE の有無	SE の有無	認証にかかる処理の 実行箇所と実行主体
RS 型	なし	あり（REE に装着）	SE（実行主体：SE アプリ）
RS-T（TA）型	あり	あり（REE に装着）	TEE（実行主体：TA）
RS-T（SE）型			SE（実行主体：SE アプリ）
R-TS（TA）型		あり（TEE に装着）	TEE（実行主体：TA）
R-TS（SE）型			SE（実行主体：SE アプリ）

型において、認証にかかる処理を TA が担うものを RS-T（TA）型と呼び、SE アプリが担うものを RS-T（SE）型と呼ぶ。また、R-TS 型において、認証にかかる処理を TA が担うものを R-TS（TA）型と呼び、SE アプリが担うものを R-TS（SE）型と呼ぶ。この結果、認証の実行形態は論理的には 5 つのタイプに分けられる（図表 7 を参照）。

モバイル端末における認証にかかる処理のうち、利用者認証については、①サービス利用者への認証用データ要求の送信、②サービス利用者からの認証用データの受信、③サーバへの認証用データの転送あるいは判定結果の送信を含む。取引認証については、①サーバからの取引認証要求の受信、②サービス利用者への確認要求の送信、③サービス利用者からの確認結果の受信、④サーバへの確認結果の送信を含む。

これらの実行箇所としては、TEE あるいは SE を想定する。TEE で実行する場合には、認証処理用の TA が安全にインストールされていることとする。他方、SE で実行する場合には、認証処理用の SE アプリが安全にインストールされていることとする。

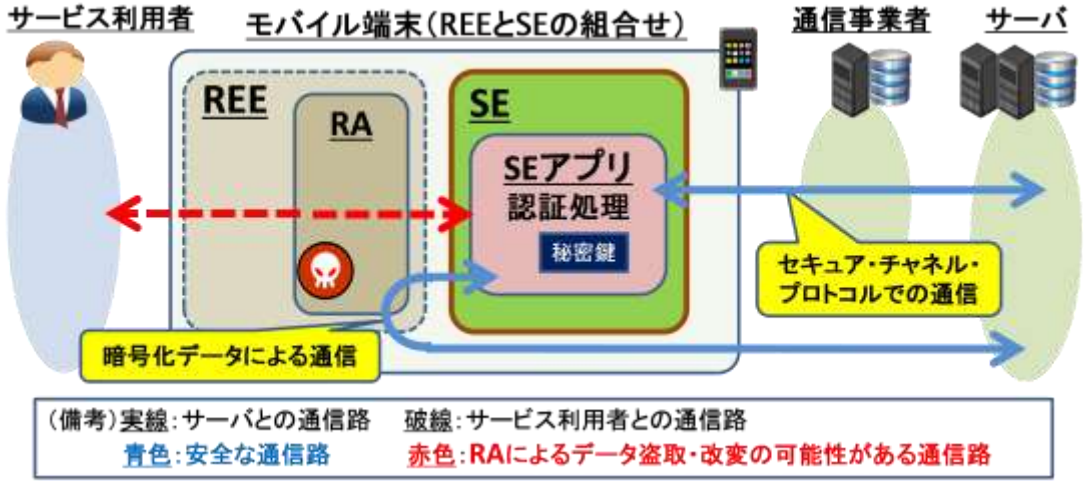
（2）各実行形態におけるマルウェアによる攻撃への対策方針

次に、図表 7 の各実行形態における凶悪型のマルウェアへの対策方針について検討する。特に、凶悪型のマルウェアが、金融サービス用のアプリケーション・ソフトウェア（Rich OS Application: RA）として REE 上で動作し、REE 上で処理されるデータの盗取・改変を試行した場合の影響に絞って検討する[26]。なお、RA は、TA のように、信頼できるソフトウェアとして確認されるわけではない。その際、TEE と SE による内部のデータやソフトウェアの保護・管理機能が有効であり、マルウェアは、TEE と SE の内部のデータ（暗号用の秘密鍵等）の盗取・改変や、TA や SE アプリの改変が困難であるとして検討を行う。

RS 型は、REE に SE が装着され、SE アプリが認証処理を実行するタイプであ

[26] 当該マルウェアが正規のコード署名を有している場合（コード署名検証では当該マルウェアを検知困難な場合）を想定する。

図表 8　RS 型の構成と通信（概念図）



る（図表 8 を参照）[27]。サーバとの通信は、SE アプリが、データを暗号化したうえで REE を介して行うことが考えられる。SE アプリがセキュア・チャネル・プロトコルを利用できる場合には、それによって通信事業者等を経由した通信路を確立することも考えられる。サービス利用者との通信は、REE を介することとなり、マルウェアによる通信データの盗取・改変に留意する必要がある。

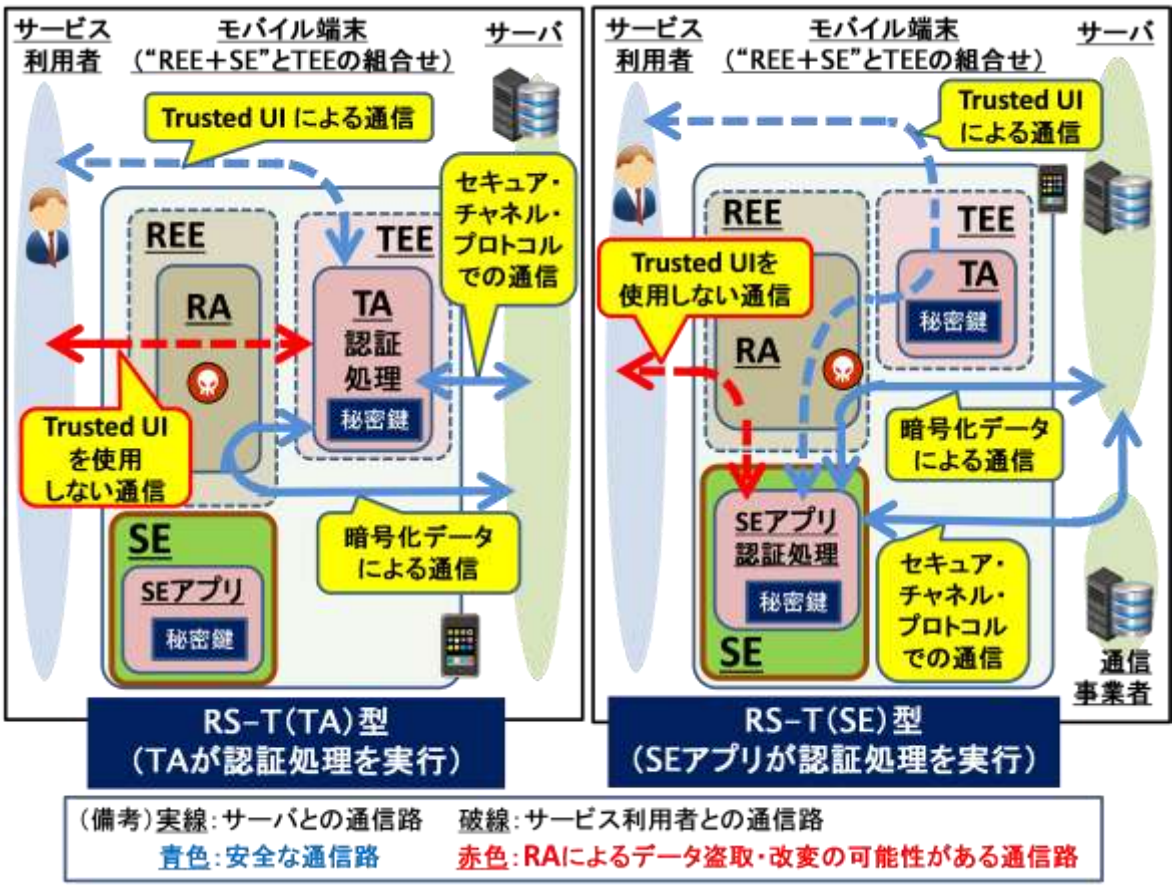
RS-T（TA）型は、REE と TEE が併存し、SE が REE と直接通信するタイプであり、認証処理は TA によって実行される（図表 9 左を参照）。サーバとの通信は、TA がサーバの公開鍵等を用いてデータを暗号化し、REE を介して行うことが可能である。また、TA がセキュア・チャネル・プロトコルを利用可能な場合には、それによってサーバとの間で通信路を確立することも考えられる。サービス利用者との通信は、Trusted UI を利用できれば、安全に行うことができる。そうでない場合、REE を介して行うため、マルウェアによる通信データの盗取・改変に留意する必要がある。

RS-T（SE）型は、REE と TEE が併存し、SE が REE と直接通信するタイプであり、認証処理は SE アプリによって実行される（図表 9 右を参照）[28]。サーバとの通信は、SE アプリがサーバの公開鍵等を用いてデータを暗号化し、REE を介して行うことが可能である。また、通信事業者等との間でセキュア・チャネル・プロトコルを利用可能な場合には、それによって安全な通信路を確立する

[27] セキュア・エレメントとして SIM を活用したモバイル端末による認証方式が大塚ほか［2016］、磯原・竹森・本間［2016］によって提案されているが、これらは RS 型に対応すると考えられる。これらの方式の詳細については補論を参照されたい。

[28] TEE と SIM を組み合わせたモバイル端末上での電子決済方式が Ahmad *et al.* [2013]によって提案されているが、当該方式は RS-T（SE）型に対応すると考えられる。当該方式の詳細については補論を参照されたい。

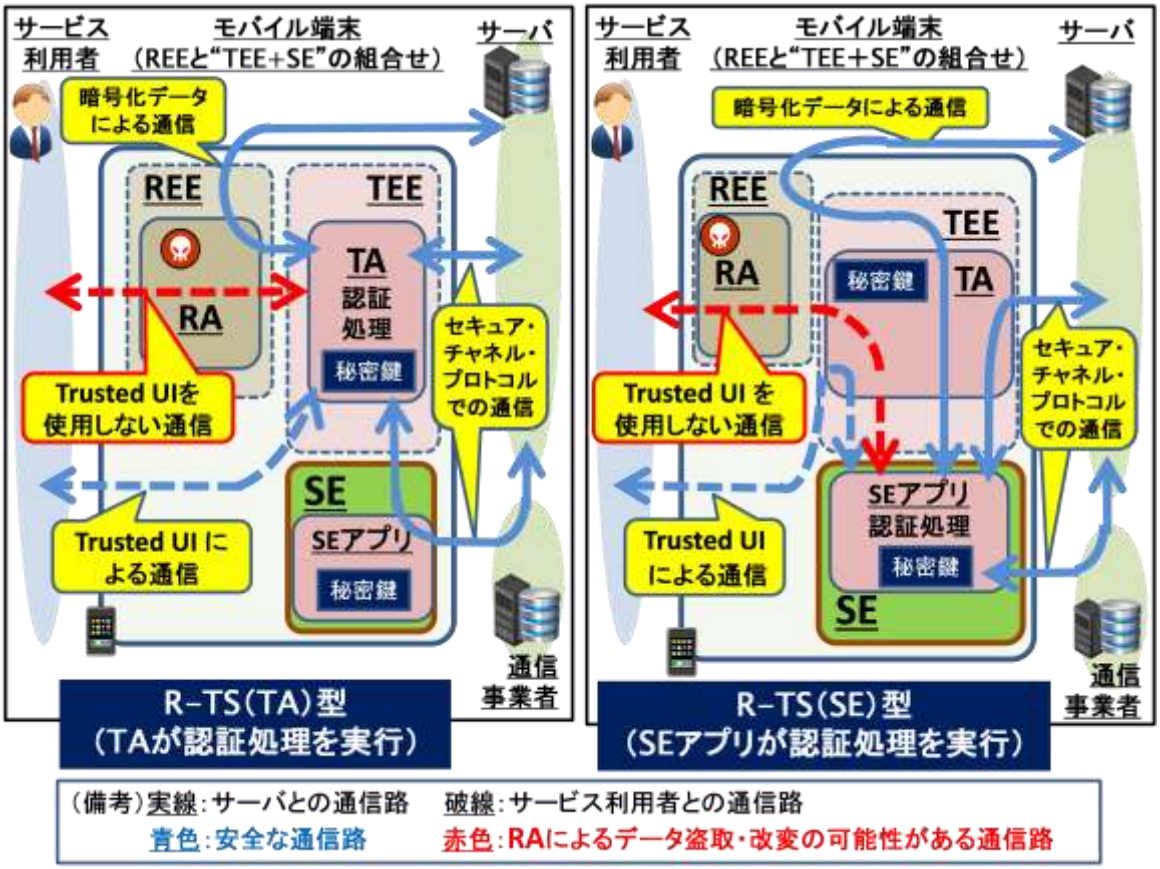
図表 9　RS-T（TA）型と RS-T（SE）型の構成と通信（概念図）



ことも考えられる。サービス利用者との通信は、SE アプリがデータを暗号化して REE 経由で TA に送信した後、Trusted UI を利用できる場合には、それを介して行うことができる。Trusted UI を利用できない場合、REE を介して行うため、マルウェアによる通信データの盗取・改変に留意する必要がある。

R-TS（TA）型は、REE と TEE が併存し、SE が TEE と直接通信するタイプであり、認証処理は TA によって行われる（図表 10 左を参照）。サーバとの通信は、TA がサーバの公開鍵等を用いてデータを暗号化し、REE を介して行うことが可能である。また、TA がセキュア・チャネル・プロトコルを利用できる場合には、サーバとの間で通信路を確立することも考えられる。なお、SE アプリとの間で直接通信するとともに、SE と通信事業者等の間のセキュア・チャネル・プロトコルを利用することが可能な場合には、SE および通信事業者等を経由してサーバと通信することも選択肢となりうる。サービス利用者との通信は、Trusted UI を利用できる場合、安全に行うことができる。Trusted UI を利用できない場合には、REE を介して行うため、マルウェアによる通信データの盗取・改変に留意する必要がある。

図表 10　R-TS（TA）型と R-TS（SE）型の構成と通信（概念図）



R-TS（SE）型は、REE と TEE が併存し、SE が TEE と直接通信するタイプであり、認証処理が SE アプリによって行われる（図表 10 右を参照）。サーバとの通信は、サーバの公開鍵等を用いてデータを暗号化し、TEE と REE を介して行うことができる。通信事業者等とのセキュア・チャネル・プロトコルが利用できる場合には、その通信路を経由してサーバとの間で通信路を確立することも考えられる。なお、TA とサーバの間でセキュア・チャネル・プロトコルが利用できる場合には、暗号化したデータを TA 経由でサーバに送信することも選択肢となりうる。サービス利用者との通信は、TA を介して Trusted UI を利用できる場合、安全に行うことができる。Trusted UI を利用できない場合、REE を介して行うため、通信データの盗取・改変に留意する必要がある。

（3）対策方針のまとめ

SE アプリや TA によるサーバとの通信は、いずれも、データを暗号化する、または、可能な場合にはセキュア・チャネル・プロトコルを利用するという対応が考えられる（図表 11 を参照）。データの暗号化に関しては、SE アプリや TA

図表 11 各実行形態におけるマルウェアの攻撃への対策方針や留意点

<table>
<tr><th rowspan="2">タイプ名</th><th colspan="2">対策方針や留意点</th></tr>
<tr><th>サーバとの通信</th><th>サービス利用者との通信</th></tr>
<tr><td>RS 型</td><td>・通信データを暗号化し REE を介して通信。
・SE のセキュア・チャネル・プロトコルを利用。</td><td>REE を介して通信するため、通信データの盗取・改変に留意。</td></tr>
<tr><td>RS-T（TA）型</td><td>・通信データを暗号化し REE を介して通信。
・TEE のセキュア・チャネル・プロトコルを利用。</td><td rowspan="4">Trusted UI を利用。そうでない場合、REE を介して通信するため、通信データの盗取・改変に留意。</td></tr>
<tr><td>RS-T（SE）型</td><td>・通信データを暗号化し REE を介して通信。
・SE のセキュア・チャネル・プロトコルを利用。</td></tr>
<tr><td>R-TS（TA）型</td><td rowspan="2">・通信データを暗号化し REE を介して通信。
・SE や TEE のセキュア・チャネル・プロトコルを利用。</td></tr>
<tr><td>R-TS（SE）型</td></tr>
</table>

にこうした機能が備わっている場合が多いとみられるが、通信相手（サーバ等）の公開鍵や署名検証用の電子証明書等を事前に入手することなどが別途必要であり、これらの管理について留意が求められる。また、セキュア・チャネル・プロトコルを利用するためには、TEE あるいは SE が REE を介さず通信するための通信用デバイスが必要になる場合がある。

SE アプリや TA によるサービス利用者との通信は、人間・機械間で行うことになる。したがって、エンド・ツー・エンドでの暗号通信は困難であり、REE を経由しない通信路の利用がまず考えられる。TEE を実装するタイプ（RS 型以外のもの）では、Trusted UI を利用するという選択肢がありうる。もっとも、これはオプションとされている機能であり、専用のデバイス等が必要となる場合もある。

このようにみていくと、SE アプリや TA の実行形態として複数の選択肢が存在するなかで、サーバやサービス利用者との通信の安全性を確保することが重要な課題であり、とりわけ、SE アプリや TA によってサービス利用者との通信をどう実現するかが重要であるといえる。

５．SE 等の活用における今後の課題

4 節では SE（SE アプリ）や TEE（TA）による認証処理の実行形態を分類し、対策方針と留意点を示した。これらを踏まえ、金融機関等が SE 等を活用した認証処理を実現するための主な課題について考察する。

まず、金融機関等は、金融サービス用のアプリケーション・ソフトウェアに加えて、認証処理用の SE アプリや TA、これらを管理するための SD 等を用意

し、サービス利用者が自分のモバイル端末にインストールするための環境を整備する必要がある。そのためには、SE や TEE を管理する主体である通信事業者やモバイル端末ベンダー等（以下、通信事業者等）と連携して検討を進めることが求められる。その際、①認証処理の実行形態、②モバイル端末への SE アプリ、TA、SD 等の導入方法、③通信事業者等との役割分担が検討項目になると考えられる。

上記①の認証処理の実行形態に関しては、サービス利用者との通信をどう保護するかが課題となる。サービス利用者との通信は、エンド・ツー・エンドでの暗号化が困難である。しかし、Trusted UI を備えたモバイル端末であれば、それを介して安全に行うことができる（R-S 型を除くタイプが該当）。もっとも、現時点では、TEE や Trusted UI を利用可能なモバイル端末は一部に限られ、別の手段で対応せざるを得ない端末も存在する。このため、足許では、Trusted UI を前提としない SIM のみを利用した手法（例えば、磯原・竹森・本間［2016］、大塚ほか［2016］）が候補になると考えられる。

上記②のモバイル端末への SE アプリ等の導入方法については、SE や TEE が製品レベルで期待どおりの機能を有していることをどう確認するかが重要な課題である。SE に関しては、コモン・クライテリア（Common Criteria）に基づく評価・認証や米国連邦政府や日本の暗号モジュールにかかる認証スキーム（CMVP／JCMVP）による認証の有無とその内容がベンチマークになる（田村・宇根［2008］）。こうした公的機関による認証は、対象となる製品が一定の安全性を確保していることを示す証となり、金融機関等がセキュリティ要件の充足度合いを確認するだけでなく、サービス利用者の安心感や信頼感を醸成するうえでも重要である。TEE に関しても同様であり、最近では、TEE を実現する Trusted OS コンポーネント等がコモン・クライテリアに基づく評価・認証を取得した事例も知られている（Trustonic [2017b]）[29]。今後、こうした事例が拡大し、評価・認証結果を金融機関等が参照できるようになることが期待される。

さらに、SE や TEE の安全性に問題がないとしても、SE アプリや TA が期待した動作を行わない場合、モバイル端末上での安全な認証を実現できなくなる可能性がある[30]。したがって、SE や TEE へのインストールが許容される SE アプリや TA の品質の適切性をいかに確保するかが課題となる。モバイル端末上での

[29] 例えば、Trustonic 社の TEE 用プラットフォームである Kinibi のうち、Trusted OS コンポーネント、および、それに組み込まれた TA や API 等の部分が、コモン・クライテリアに基づく評価・認証を取得している（Trustonic [2017a]）。当該製品のセキュリティ設計仕様書（Security Target）は公開されている。金融機関等は、こうしたセキュリティ設計仕様書を参照することによって、想定されている脅威、セキュリティ対策方針、セキュリティ要件等を確認することができる。

[30] 例えば、吉田ほか［2017］は、TEE を実装する際に、一部のオープンソースのソフトウェアを利用しつつ安全性について適切に配慮しない場合、脆弱性を有する TA が生成され、REE から当該 TA が不正に操作される可能性があることを実証している。

処理を安全に実行する手段として SE や TEE が一段と注目されるようになれば、金融機関等のサービス提供者だけでなく、さまざまなベンダーが SE アプリ等を開発・提供するようになり、不正な SE アプリ等にかかるリスクが高まっていく可能性がある[31]。そうしたリスクへの対応として、SE アプリや TA を作成するための開発ツールの提供、正規の SE アプリや TA の認証、セキュア・チャネル・プロトコル等による安全な通信路を介したインストール等が検討項目として考えられる[32]。これらについて、金融機関等は、通信事業者等と連携しつつ、役割分担を明確にしながら検討を進めていく必要がある。

SE や TEE の最大の特徴は、金融機関等がサービス利用者のモバイル端末内部に、信頼できる実行環境とアプリケーション・ソフトウェアを実現する仕組みを提供する点にあるといえる。インターネットを介したオンライン・バンキング等では、これまでパソコンによる利用が中心となっているが、残念ながら、サービス利用者のパソコン内部に同様の信頼できる実行環境等を実現するに至っておらず、マルウェアによる脅威にさらされる状況が続いている。金融サービスを利用する媒体としてのモバイル端末の普及と歩調を合わせて SE の活用を検討するとともに、今後の普及が期待される TEE や Trusted UI の活用を検討することは、上記のパソコンにおける状況からの改善や脱却という観点からも重要であると考えられる。

金融サービスにおけるモバイル端末のさらなる活用を展望する金融機関等においては、通信事業者、モバイル端末ベンダー、SE や TEE のベンダー等、関係者と密接に連携しつつ検討を進めていくことを期待したい。

以　上

---

31 GlobalPlatform では、通信事業者等がサービス利用者に（複数導入できる）SD の一部を制御可能とする設定にかかる技術仕様を作成している（GlobalPlatform [2017c]）。従来、SE 内部の SD は、SE の発行者である通信事業者等が管理するモデルとされてきたが、上記仕様では、サービス利用者が（例えば、暗証番号等の利用者認証を実行したうえで）SD を制御するとともに、当該 SD のもとで（サービス利用者が選択した）TA を導入するための設定等が規定されている。

32 GlobalPlatform は、SE 上で動作する（GlobalPlatform の技術仕様群に準拠した）アプリケーション・ソフトウェア（SE アプリに相当）を開発するためのベンダー向けキットの提供を開始している（GlobalPlatform [2017b]）。これは、SE アプリや TA を作成するための開発ツールの提供に関連する動きの 1 つと考えることができる。

【参考文献】

井澤秀益・五味秀仁、「次世代認証技術を金融機関が導入する際の留意点：FIDO を中心に」、『金融研究』、第 35 巻第 4 号、日本銀行金融研究所、2016 年、21～54 頁

磯原隆将・竹森敬祐・本間輝彰、「SIM を活用したモバイルバンキングのセキュリティ向上に関する検討」、コンピュータセキュリティシンポジウム 2016 予稿集、情報処理学会、2016 年

大塚玲・佐藤裕之・櫻木正一郎・落合三男・横田勇一・藤澤将吾・本間靖朗・小関松子、「SIM-Sign：実用的な Android 端末向け MitMo 対策技術」、コンピュータセキュリティシンポジウム 2016 予稿集、情報処理学会、2016 年

田村裕子・宇根正志、「情報セキュリティ製品・システムの第三者評価・認証制度について：金融分野において利用していくために」、『金融研究』第 27 巻別冊第 1 号、日本銀行金融研究所、2008 年、53～100 頁

トレンドマイクロ、「2016 年を振り返る：世界のモバイル脅威事情 2・脆弱性の利用と Apple iOS を狙う攻撃」、トレンドマイクロ・セキュリティブログ、2017 年（http://blog.trendmicro.co.jp/archives/14425）

中村啓佑、「金融分野の TPPs と API のオープン化：セキュリティ上の留意点」、『金融研究』第 36 巻第 3 号、日本銀行金融研究所、2017 年、83～110 頁

日本銀行決済機構局、「モバイル決済の現状と課題」、決済システムレポート別冊シリーズ、2017 年

吉田直樹・福島和彦・宮内成典・坂本純一・藤本大介・松本勉、「TEE システムアーキテクチャとそのオープンソース実装のセキュリティ評価」、『信学技報』、ISEC-2017-36、電子情報通信学会、2017 年

Ahmad, Zaheer, Lishoy Francis, Tansir Ahmed, Christopher Lobodzinski, Dev Audsin, and Peng Jiang, "Enhancing the Security of Mobile Applications by Using TEE and (U)SIM," *Proceedings of 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing, and Autonomic and Trusted Computing (UIC/ATC)*, 2013.

Elenkov, Nikolay, *Android Security Internals: An In-Depth Guide to Android's Security Architecture*, No Starch Press, 2015.

European Payments Council, *White Paper Mobile Payments*, Version 5.0, 2017.

European Union Agency for Network and Information Security, *Security of Mobile Payments and Digital Wallets*, 2016a.

————, *Smartphone Secure Development Guidelines*, 2016b.

FIDO Alliance, *FIDO UAF Complete Specifications*, 2014. (https://fidoalliance.org/specifications/download/)

GlobalPlatform, *Annex C: TLS Specification of TEE Sockets API Specification version 1.0.1*, Document Reference: GPD_SPE_103, 2017a.

————, *Card Specification version 2.3*, Document Reference: GPC_SPE_034, 2015a.

————, "GlobalPlatform and FIDO Alliance Sign Memorandum of Understanding," 2016a. (https://www.globalplatform.org/mediapressview.asp?id =1237)

————, “GlobalPlatform Launches Developers’ Kit to Ease and Expedite Development of Secure Mobile Services,” 2017b. (https://www.globalplatform.org/mediapressview.asp?id=1311)
————, “GlobalPlatform Releases Consumer-Centric Model Configuration,” 2017c. (https://www.globalplatform.org/mediapressview.asp?id=1286)
————, *Secure Element Access Control version 1.1*, Document Reference: GPD_SPE_013, 2014a.
————, *TEE Management Framework version 0.0.038*, Document Reference: GPD_SPE_120, 2016b.
————, *TEE Protection Profile version 1.2*, Document Reference: GPD_SPE_ 021, 2014b.
————, *TEE Secure Element API version 1.1.1*, Document Reference: GPD_SPE_024, 2016c.
————, *TEE Sockets API Specification version 1.0.1*, Document Reference: GPD_SPE_100, 2017d.
————, *TEE System Architecture version 1.0.0.27*, Document Reference: GPD_SPE_009, 2016d.
————, *The Trusted Execution Environment: Delivering Enhanced Security at a Lower Cost to the Mobile Market*, White Paper, 2015b.
————, *Trusted User Interface API version 1.0*, Document Reference: GPD_SPE_020, 2013.
GSMA, *NFC Handset APIs & Requirements Version 2.0*, 2011.
International Telecommunication Union, *Security Aspects of Digital Financial Services*, Technical Report of ITU-T Focus Group on Digital Financial Services, 2017.
Irazoqui, Gorka, and Xiaofei Guo, “Cache Side Channel Attack: Exploitability and Countermeasures,” Black Hat Asia 2017, 2017.
Lipp, Moritz, Daniel Gruss, Raphael Spreitzer, Clémentine Maurice, and Stefan Mangard, “ARMageddon: Cache Attacks on Mobile Devices,” *Proceedings of the 25th USENIX Security Symposium*, 2016, pp.549-564.
Lu, Michael, “Mobile Security in the Chinese Market,” Trustonic News, December 4, 2015.
Marczak, Bill, and John Scott-Railton, “The Million Dollar Dissident: NGO Group’s iPhone Zero-Days used against a UAE Human Rights Defender,” Citizen Lab, August 24, 2016.
Ortiz-Yepes, Diego Alejandro, “A Review of Technical Approaches to Realizing Near-Field Communication Mobile Payments,” *IEEE Security & Privacy*, 14 (4), 2016, pp.54-62.
Pan, Jordan, “User Beware: Rooting Malware Found in 3rd Party App Stores,” TrendLabs Security Intelligence blog, Trend Micro, 2016.
Parker, Luke, “Ledger’s TEE trustlet for smartphone bitcoin wallets released,” March 2, 2016. (http://bravenewcoin.com/news/ledgers-tee-trustlet-for-smartphone-bitcoin-wallets-released/)
SIMalliance, *Open Mobile API Specification V3.0*, 2014.
Taylor, Vincent F., and Ivan Martinovic, “Short Paper: A Longitudinal Study of Financial Apps in the Google Play Store,” *Proceedings of Financial Cryptography and Data Security 2017*, 2017.

Timmers, Niek, and Albert Spruyt, "Bypassing Secure Boot using Fault Injection," Black Hat Europe 2016, 2016.

Trustonic, "Trustonic Provides First Trusted Execution Environment for Enterprise Mobility Management," Trustonic Press Releases, April 21, 2015. (https://www.trustonic.com/news-events/pr/first-trusted-execution-environment)

————, *Kinibi v311A Security Target*, 2017a.

————, "Trustonic device security platform achieves world's first TEE security certification from Common Criteria," Trustonic Press Releases, March 16, 2017b.

Umar, Assad, and Keith Mayes, "Trusted Execution Environment and Host Card Emulation," in Mayes, Keith, and Konstantinos Markantonakis, eds. *Smart Cards, Tokens, Security and Applications*, Springer International Publishing, 2017, pp. 497–519.

補論. SE 等を活用した代表的な提案手法

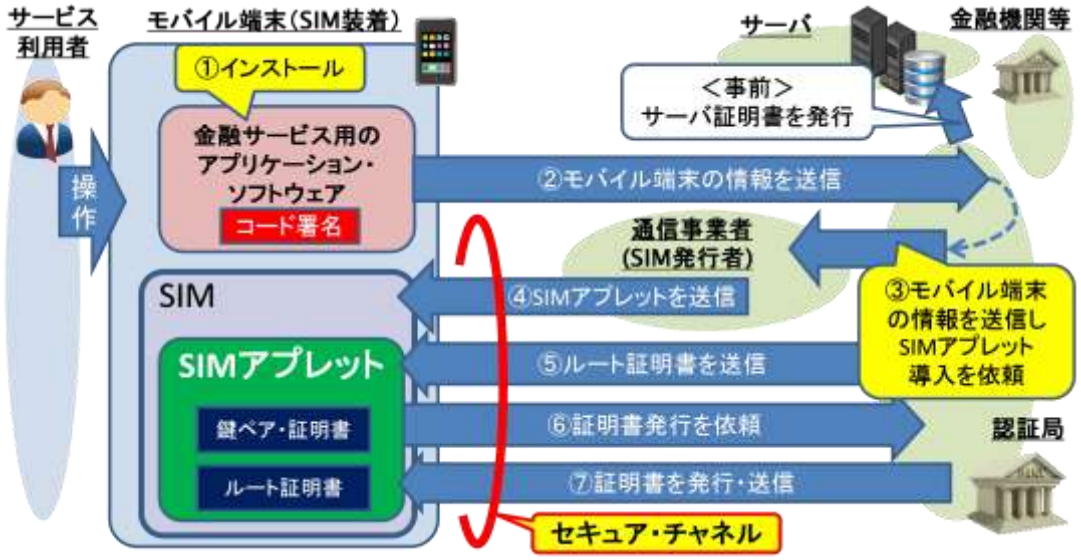
（1）大塚ほか［2016］による SIM を活用した手法（RS 型）

大塚ほか［2016］は、SIM を用いて、金融サービス用のアプリケーション・ソフトウェアが金融機関等によって提供され改変されていないことを確認する手法を提案している（2 節（3）の対策方針 1 に対応）。具体的には、専用のソフトウェア（以下、SIM アプレット）を格納した SIM をモバイル端末に装着し、金融サービス用のアプリケーション・ソフトウェアをインストールする際に、当該ソフトウェアの一貫性をコード署名によって検証する[33, 34]。また、取引を実行する前に、サーバは、SIM アプレットの証明書とサービス利用者（のアカウント）との対応関係を確認する（2 節（3）の対応方針 2 に対応）。

まず、SIM アプレットの導入は、コード署名検証等に用いられる証明書を発行する認証局と、SIM を発行・管理する通信事業者を想定して、主に以下の流れで実施される（図表 A-1 を参照）。

①サービス利用者は、金融サービス用のアプリケーション・ソフトウェア（コー

図表 A-1　大塚ほか［2016］における SIM アプレットの導入手順（概念図）



33 大塚ほか［2016］の手法はモバイル端末の OS として Android OS を前提としている。

34 コード署名の検証は、モバイル端末の OS コンポーネントの一部であり、アプリケーション・ソフトウェアによる SE（ここでは SIM に対応）へのアクセスを制御するセキュア・エレメント・アクセス API で実行される。当該 API は、GSMA の技術仕様である NFC Handset APIs & Requirements（GSMA [2011]）や、GlobalPlatform の技術仕様である Secure Element Access Control（GlobalPlatform [2014a]）で規定されている。大塚ほか［2016］はこれらに準拠した仕組みを提案している。

ド署名付き）をモバイル端末（SIM 装着）にインストールし起動する。インストールの際、コード署名の検証が実施される。
②当該ソフトウェアは、モバイル端末の情報（端末を特定する ID 等）を認証局に送信する。
③認証局は、モバイル端末の情報を通信事業者に送信し、SIM アプレットの導入を依頼する[35]。以下の④から⑦の通信は、セキュア・チャネルを通じて実施される。
④通信事業者は、SIM アプレットを SIM に送信する。
⑤認証局は、SIM アプレットにルート証明書等を送信する。
⑥SIM アプレットは、内部で公開鍵暗号の鍵のペア（複数）を生成し、それらに対する証明書の発行を認証局に依頼する。
⑦認証局は、SIM アプレットに証明書を送信する。

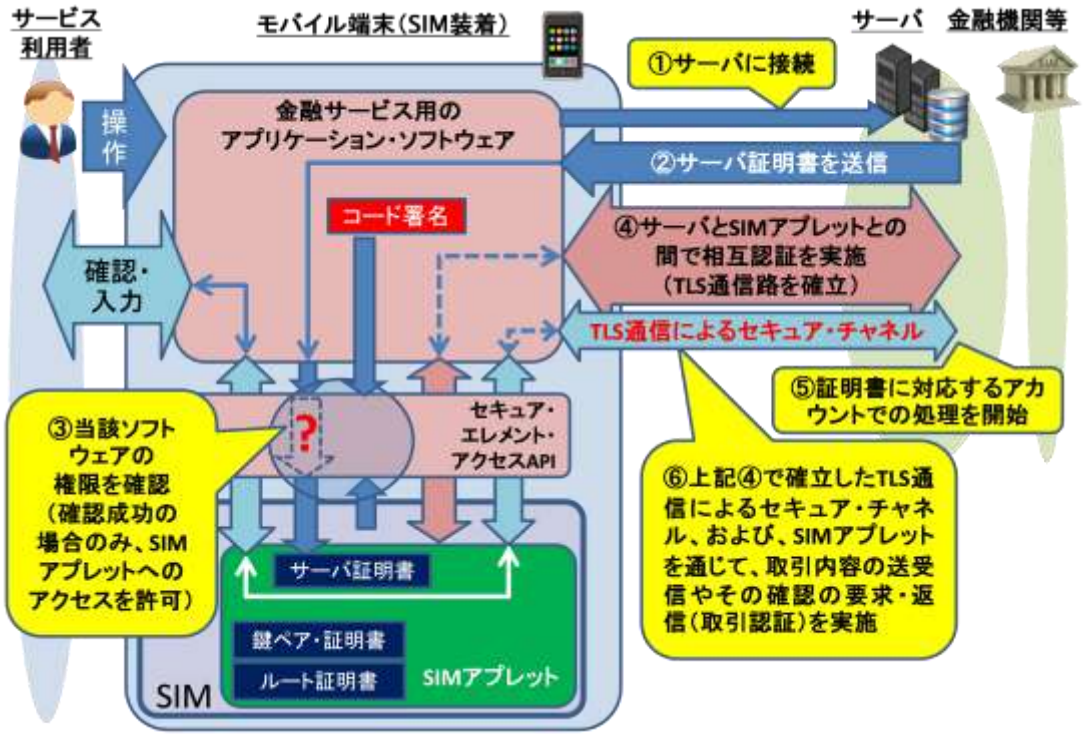
上記によって SIM アプレットを導入した後、サービス利用者がサーバに開設しているアカウントに SIM アプレットを結びつけるための手続が実施される。具体的には、SIM アプレットは、サーバとの間で相互認証を行ったうえで、自分の証明書をサーバに送信する。サーバは、サービス利用者の同意を別途確認したうえで、当該証明書とサービス利用者のアカウントを対応させる。金融サービス用のアプリケーション・ソフトウェアを動作させる際の処理の流れの概要は以下のとおりである（図表 A-2 を参照）。

①金融サービス用のアプリケーション・ソフトウェアはサーバに接続する。
②サーバは、当該ソフトウェアにサーバ証明書を送信する。
③当該ソフトウェアは、セキュア・エレメント・アクセス API に対して、SIM アプレットへのアクセスを要求する。アクセス可能と判断された場合のみ、SIM アプレットへのアクセスが可能となり、SIM アプレットにサーバ証明書を送信する[36]。
④SIM アプレットは、上記③のサーバ証明書等を用いて、当該ソフトウェア経由でサーバとの間で相互認証を実施し、暗号通信路を確立する[37]。

---

35 ここでは、認証局が SIM アプレットの導入を通信事業者に依頼するが、SIM アプレットの提供元は金融機関等になると考えられる。
36 セキュア・エレメント・アクセス API は、SE 内部のアプリケーション（ここでは SIM アプレットに対応）にアクセスする権限の有無を確認する（GlobalPlatform [2014a]、GSMA [2011]）。アクセスする権限の有無を確認する際に用いられる情報は SIM から当該 API に提示される。さらに、SE へのアクセスを要求するアプリケーション・ソフトウェアにかかる証明書が検証されるほか、コード署名が検証される場合も考えられる。
37 大塚ほか［2016］では、RFC 5246 として標準化されている暗号通信プロトコルである TLS（Transport Layer Security）によって相互認証等を実施する旨が示されている。

図表 A-2　SIM アプレットを用いた金融サービスでの処理の流れ（概念図）



⑤その後、SIM アプレットはサーバに対して自身の証明書を送信し、サーバは、当該証明書に対応する（サービス利用者の）アカウントでの処理を開始する（当該アカウントへのログイン）。

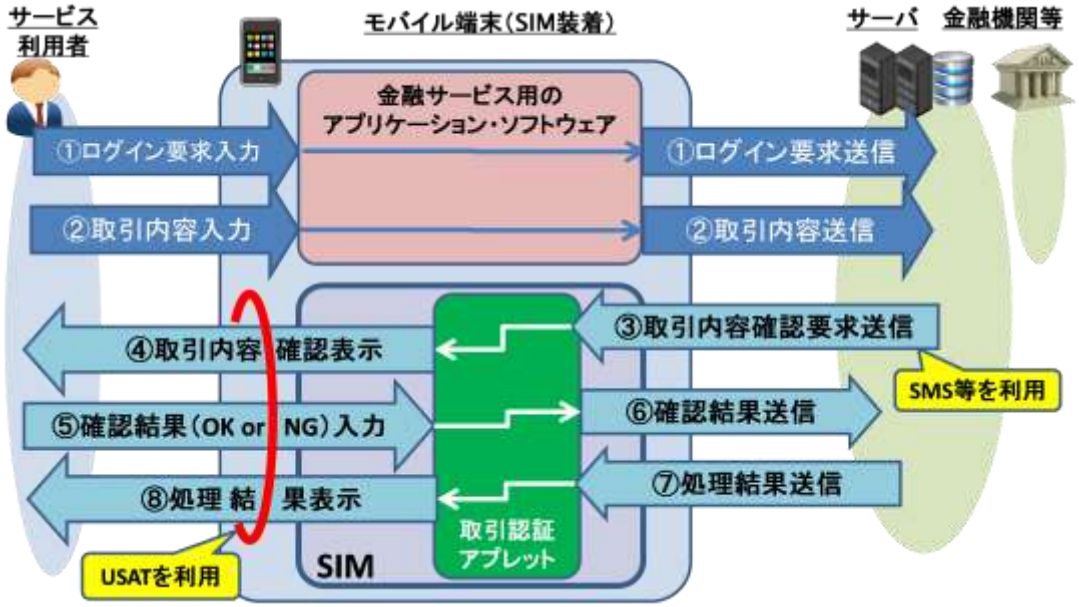
⑥サービス利用者による取引内容のデータの送信、サーバによる取引内容の確認要求、サービス利用者による当該確認要求への返信（取引認証）等、サービス利用者とサーバとの間の通信は、上記④の暗号通信路および SIM アプレットを経由して実施される。

上記③におけるセキュア・エレメント・アクセス API によるアクセス制御が安全性上のポイントになるが、大塚ほか［2016］では、当該 API が正常に動作する状況を前提としている。

（2）磯原・竹森・本間［2016］による SIM を活用した手法（RS 型）

上記の大塚ほか［2016］は、認証にかかる処理を含む、取引にかかるデータの処理や通信全般を金融サービス用のアプリケーション・ソフトウェアが実行する仕組みとしている。一方、磯原・竹森・本間［2016］は、取引認証にかかる処理を、当該アプリケーションを介さず、SIM 内部で実行する手法を提案している（2 節（3）の対策方針 3 に対応）。SIM 内部に格納されている専用のソフ

図表 A-3　磯原・竹森・本間［2016］における取引認証アプレットを用いた処理の流れ（概念図）



トウェア（以下、取引認証アプレット）が、サービス利用者やサーバとの間で取引認証にかかる処理や通信を実施するというものである[38]。

取引認証アプレットが安全に SIM に搭載されているという前提のもとで、モバイル・バンキングでの処理の流れは以下のとおり示されている（磯原・竹森・本間［2016］、図表 A-3 を参照）。

①サービス利用者は、金融サービス用のアプリケーション・ソフトウェアを起動し、ログイン要求としてアカウント情報等を入力するとともに、当該ソフトウェアを介してサーバに送信する。
②ログイン後、サービス利用者は、金融サービス用のアプリケーション・ソフトウェアに取引内容を入力する。取引内容のデータは当該ソフトウェアを介してサーバに送信される。
③サーバは、取引認証を実施するため、SMS（Short Message Service）等を利用して、金融サービス用のアプリケーション・ソフトウェアを介さずに、取引内容にかかるデータを取引認証アプレットに送信する。以下の④～⑧の通信は、金融サービス用のアプリケーション・ソフトウェアを介さない。
④取引認証アプレットは、USAT の機能を用いて、取引内容の確認のための画面を表示する。

[38] 磯原・竹森・本間［2016］では、Android OS での実装が想定されている。

⑤サービス利用者は、表示された取引内容を確認し、確認結果（取引続行の可否）を入力する。
⑥取引認証アプレットは、上記⑤の確認結果をサーバに送信する。
⑦サーバは、取引認証アプレットからの受信内容に基づいて処理続行の可否を決定し、処理が完了した場合、その旨を取引認証アプレットに送信する。
⑧取引認証アプレットは、USAT の機能を用いて処理結果を表示する。

磯原・竹森・本間［2016］では、モバイル端末上の取引認証にかかる処理をほぼ SIM 内部で実現している。取引認証アプレットとサービス利用者の間の通信や関連する処理（上記④、⑤に対応）は、OS コンポーネントの一部である USAT において実行される。したがって、USAT が安全に動作することが前提となる。

（3）Ahmad *et al.* [2013]による SIM と TEE を活用した手法（RS-T（SE）型）

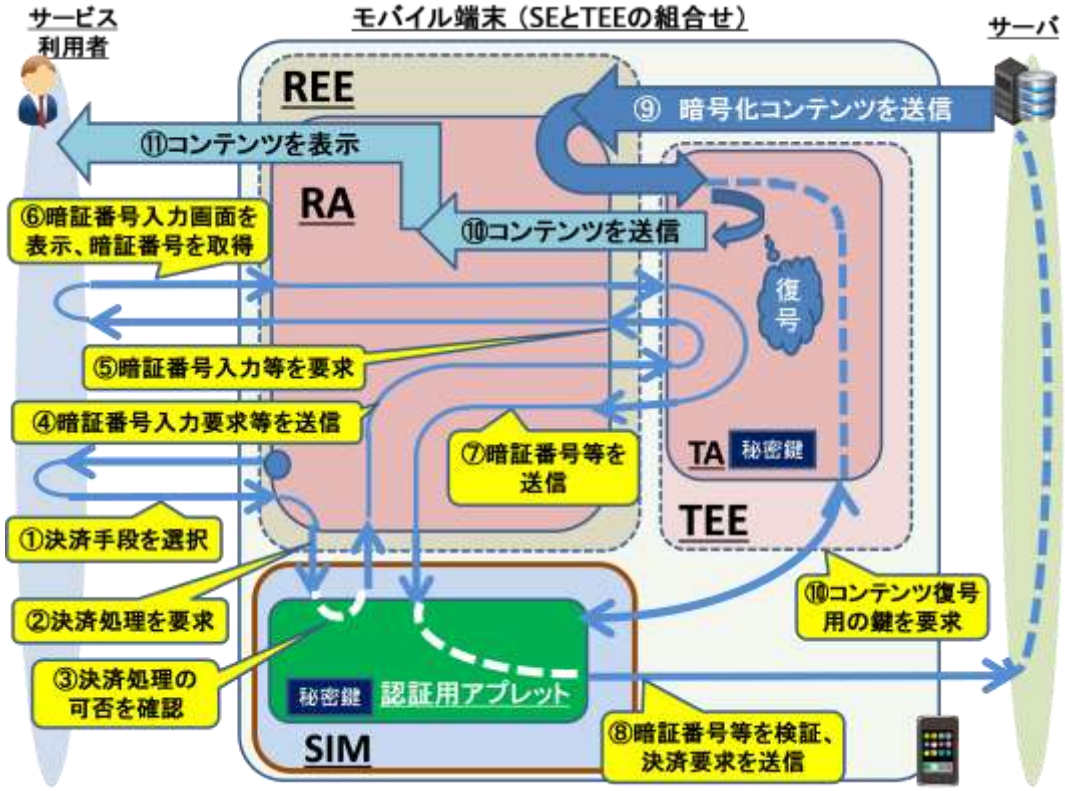
SIM と TEE を組み合わせて活用する手法として、Ahmad *et al*. [2013]によるものが挙げられる。当該手法は、モバイル端末上で有料オンライン・コンテンツを購入・閲覧するためのプロトコルを提案するものであり、サービス利用者の認証や暗号化されたコンテンツの復号等を SIM や TEE で実行する。オンライン・コンテンツの閲覧では、大量のデータを高速で処理する必要が生じる場合がある。そこで、当該手法では、負荷が大きくなる可能性があるオンライン・コンテンツの復号を TEE で実施するとともに、サービス利用者の認証等を SIM において実施するという役割分担を行っている。

SIM は、REE（ここでは Android を想定）に接続され、認証処理を実行するソフトウェア（以下、認証用アプレット）を内蔵するほか、暗号処理用の鍵、電子証明書、利用者認証用の暗証番号（PIN）等を秘密に保管する。TEE 上で動作する TA は、REE を介して SIM と暗号通信を行うとともに、PIN の入力・取得にかかるサービス利用者との通信を実施する。また、TA は、暗号化されたコンテンツの復号用の鍵を SIM から受信し、コンテンツ・サーバから受信する暗号化コンテンツを復号する。

利用者認証やコンテンツの復号にかかる処理の流れは次のとおりである（図表 A-4 を参照）。一連の処理の前に、認証用アプレットと TA は、通信データを暗号化するためのセッション鍵を共有しているとする。

①REE 内のアプリケーション・ソフトウェア（以下、RA）は、有料コンテンツの購入にかかる画面を表示し、サービス利用者は画面上から決済手段を選択する。

図表 A-4　Ahmad *et al.* [2013]の手法による処理の流れ（概念図）



②RA は、選択された決済手段による決済処理を SIM の認証用アプレットに対して要求する。

③認証用アプレットは、サービス利用者によるコンテンツ購入の可否（購入限度額を超過していないかなど）を確認し、決済処理の可否を確認する。

④認証用アプレットは、暗証番号入力要求等のメッセージを暗号化して RA 経由で TA に送信する。暗証番号すべてを入力するのではなく、暗証番号の特定の桁（複数）を示し、それらを入力するように指示する。

⑤TA は、RA に対して暗証番号入力等を要求する画面の表示を指示する。

⑥RA は、サービス利用者に暗証番号入力画面を表示し、入力された暗証番号を取得して TA に送信する。

⑦TA は、上記⑥の暗証番号等を暗号化して認証用アプレットに送信する。

⑧認証用アプレットは、暗証番号等を検証し、成功した場合、コンテンツ購入にかかる決済の実施をサーバに対して要求する。その際、当該購入にかかる取引の ID、サービス利用者の ID、金額等のデータをサーバの公開鍵で暗号化して送信する。

⑨サーバは、上記⑧の暗号化データを復号し、購入にかかる取引の内容を確認したうえで、暗号化したコンテンツを RA に送信する。

⑩TA は、コンテンツを復号するための鍵を SIM に要求し、それを用いてコンテンツを復号して RA に送信する。
⑪RA はコンテンツを表示する。

当該手法では、SIM（認証用アプレット）とサーバの間、および、SIM と TEE（TA）の間での通信データは暗号化されるものの、サービス利用者と TA の間については、Trusted UI の利用を想定していない。この点について、Ahmad *et al.* [2013]では、提案当初、Trusted UI をサポートするモバイル端末が販売されておらず、そうしたユーザ・インタフェースを利用しない手法とした旨が記載されている。