本資料は、(株) 日立製作所が総務省より受託した、行政業務システム連携推進事業(アクセス手段としての携帯電話の利便性向上方法の検証)の研究成果の一部である。

本資料は、ガイドライン化を推進するための議論のたたき台として利用することを目的に、ＡＲＩＢ高度無線通信研究委員会モバイルコマース部会に提供する。

1. 課題ア　モバイルアクセスシステムの技術仕様の策定

1.1. 概要

1.1.1. 対象範囲

本研究の対象範囲を図 1-1 に示す。図 1-1 に示すように、複数のサービス提供機関が、携帯電話端末利用者の耐タンパデバイスへの認証情報や個人情報などの ID 情報の書き込こむ。また、書き込んだ ID 情報を読み込んでサービス提供に利用する。このような、耐タンパデバイスへの ID 情報の書き込みと読み込みを安全かつ容易に行うことを本実証事業の対象範囲とする。

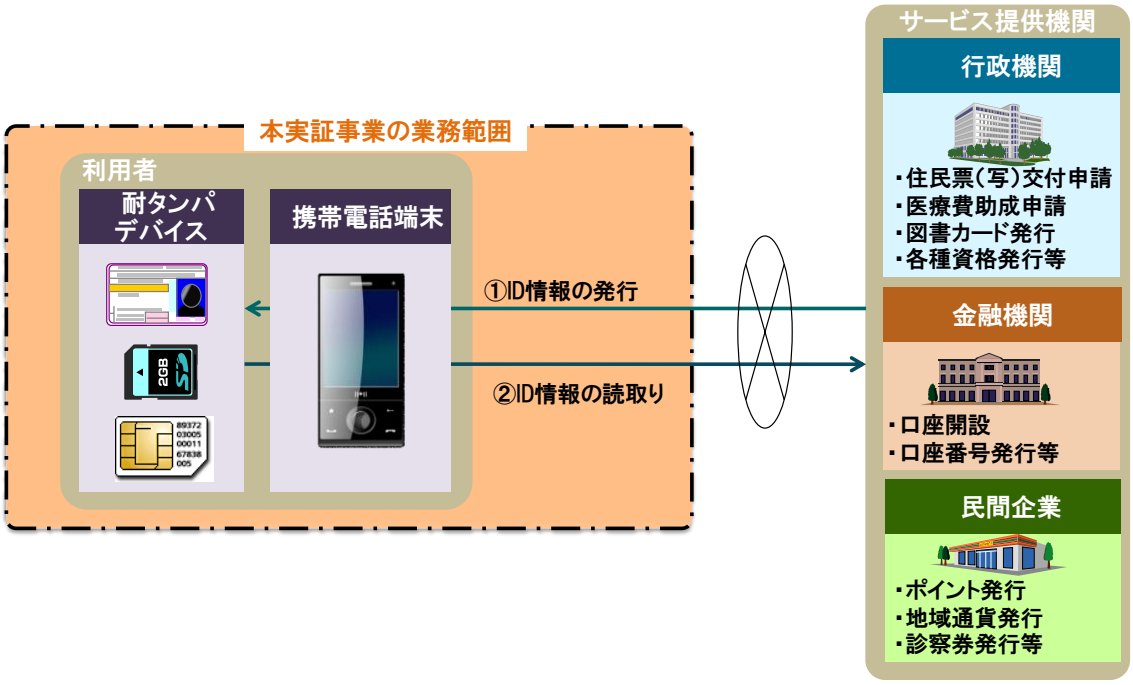

図 1-1 本実証事業の業務範囲

1.1.2. 現状の課題

複数のサービス提供機関が、携帯電話端末利用者の耐タンパデバイスへの ID 情報の書き込みや読み込みを行おうとする場合、現状では図 1-2 に示すように、サービス提供機関ごとに携帯アプリを開発・運用する必要がある。また、利用者は各サービス提供機関が提供する携帯アプリを個別にダウンロード・インストールする必要がある。さらに今後は携帯電話端末の OS のオープン化が進むことが想定されるため、携帯アプリのセキュリティを確保する仕組みも必要となる。

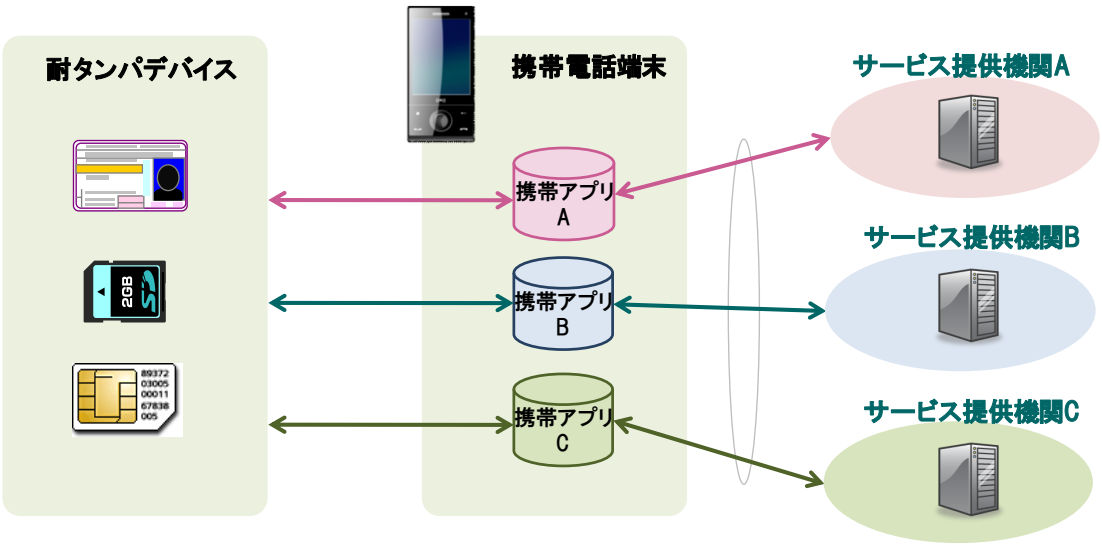

図 1-2 業務範囲の現状と課題

1.1.3. 解決方法

上記現状の課題を解決するために、図 1-3 に示す構成のモバイルアクセスシステムを提案する。具体的には、サービス提供機関が耐タンパデバイスにアクセスする際に共通的に利用できるモバイルアクセスサーバおよび携帯電話端末上の共通アプリからなるモバイルアクセスシステムを提案する。

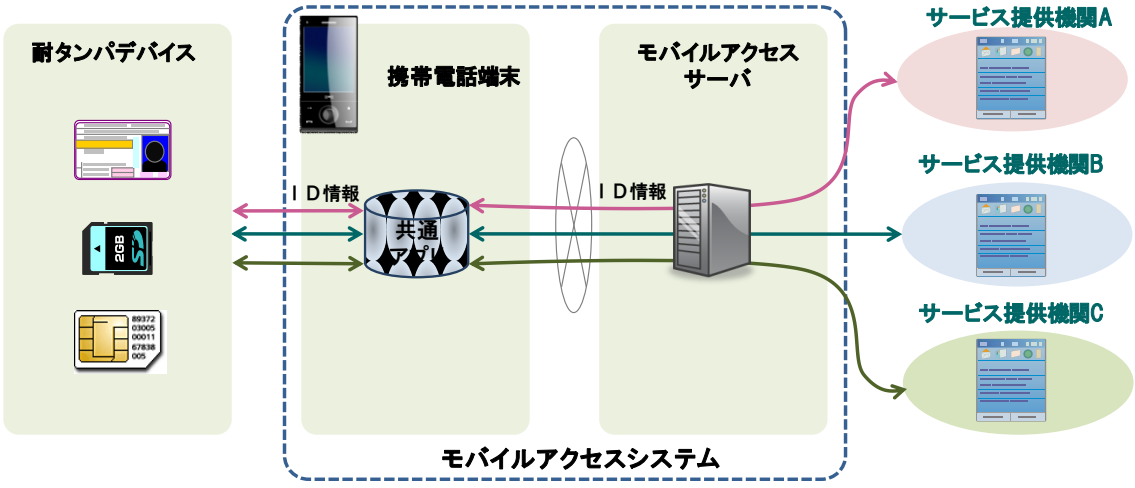

図 1-3 現状課題の解決方法（提案方式）

具体的には、サービス提供機関は、耐タンパデバイスに対する命令（コマンド）を生成しモバイルアクセスサーバに通知する。モバイルアクセスサーバは、共通アプリを経由して、耐タンパデバイスとのセキュアな通信路を確立する。(具体的には、モバイルアクセスサーバと耐タンパデバイスが共有するセッション鍵を使って安全な通信路（セキュアチャネル）を張る)。モバイルアクセスサーバは、確立された安全な通信路を使ってサービス提供機関から通知されたコマンドを、共通アプリを経由して耐タンパデバイスに送信する。共通アプリは、耐タンパデバイスの複数種類の差異を吸収し、モバイルアクセスサーバからのセキュアチャネル上のコマンドを正しく耐タンパデバイスに届けることを行う。このときに不正なサービス提供機関がコマンドを発行できないような仕組みを組み込む。また携帯電話端末もオープン端末を想定しているため、共通アプリは不正者の攻撃の対象になるという前提を置き、鍵などの秘密情報を持たせない設計とする。

このようなモバイルアクセスシステムを導入することにより、耐タンパデバイスの ID 情報を格納・参照のための複数サービス提供機関が共通的に利用できる仕組みをシステムとして利用することで、サービス提供機関が個別に携帯アプリを開発しなければならないという負担を減らすことが期待できる。また、サービス提供機関ごとに個別の携帯アプリを開発する方式では、サービスごとに利用者は携帯アプリをダウンロードする必要があるが、共通アプリであればダウンロードの手間は省ける。さらに、共通アプリを用いることによってユーザインタフェースなど統一化され、利用者の操作性を向上させることが期待できる。

1.2. 前提条件

以下に課題アの仕様の検討におけるシステムの前提条件を示す。なお、ID 体系の仕組みやその管理方法に関する仕様、個別のサービスに関する仕様は、本章の範囲外とする。

■端末に関する前提条件

- データ通信が行える端末を前提とする。音声通話だけしかできない端末は対象としない。
- 今後は携帯電話端末の OS のオープン化が進むことが想定される。そのような OS 上でも安全に耐タンパデバイスへアクセスできるようにするため、携帯電話端末上ではマルウェア等が動作する可能性があり、必ずしも安全性が確保されるとは限らない。つまり耐タンパデバイスへアクセスする鍵などの秘密情報の管理を正しく行うことができない場合があるという前提を置いたうえで仕様の検討を行うものとする。
- ブラウザから携帯電話端末内のアプリが起動できるものとする。
- 逆に、携帯電話端末内のアプリからブラウザを起動できるものとする。
- 耐タンパデバイスにアクセスできる機能を有するのもとする。

■耐タンパデバイスに関する前提条件

- 携帯電話端末を使ってデータの送受信ができるものとする。

- 耐タンパデバイス内の処理は、安全に行えるものとする。つまり、耐タンパデバイス上で動作するアプリケーションは、正当なサービス提供機関によって作成され、正当な方法で耐タンパデバイス内へロードされ、その動作も正しく動くものとする。
- 耐タンパデバイスは、マルチアプリケーション対応とし、複数のサービス提供機関が相乗りできるものとする。異なるサービス提供機関のＩＣカードアプリケーションはファイアウォールで適切に守られているとする。
- 平成２１年度の総務省の調査研究「携帯電話から電子行政サービス等へのアクセス技術の調査研究」で対象とされた①国が発行する公的 IC カードを携帯電話にかざして利用する公的 IC カード方式におけるフルサイズ IC カード(ISO14443 Type A/Type B)、②携帯電話端末に挿入可能なデバイスを国が発行し携帯電話端末に挿入して利用する携帯電話向け公的カード方式における IC チップを搭載したフラッシュメモリ型デバイス、③携帯電話端末内に国が発行する情報を書き込み利用する公的認証情報方式における UICC (Universal Integrated Circuit Card) を前提とする。
- 携帯電話端末に対して OTA(Over the Air)で耐タンパデバイスへアクセスするための唯一の世界標準である GlobalPlatform に準拠した IC カードを前提とする。

■ネットワークに関する前提条件

- サービス提供機関とモバイルアクセスサーバはインターネットに接続されるため、携帯電話端末とサービス提供機関間、携帯電話端末とモバイルアクセスサーバ間は、モバイル網以外のオープンなネットワークを通ることになる。このため、必ずしも安全性が確保されるとは限らないものとする。 つまり、携帯電話端末とサービス提供機関間、携帯電話端末とモバイルアクセスサーバ間は、ネットワーク上のデータの盗聴や改ざんの恐れがあるとする。
- サービス提供機関とモバイルアクセスサーバ間の通信は VPN や専用線などで保護されるため安全であるとする。

■サービス提供機関に関する前提条件

- 偽造や改ざん、漏洩などから守るべき何らかの価値を有する情報（ID 情報）を携帯電話端末に接続された耐タンパデバイスに対して付与し、また、耐タンパデバイスから読み込み、その ID 情報を利用することをおこなうサービス提供機関を対象とする。
- サービス提供機関の動作は、運用も含め正しく安全に行われるものとする。
- サービス提供機関とモバイルアクセスサーバは事前の契約に基づいて鍵の共有などを行っているものとする。

■モバイルアクセスサーバに関する前提条件

- モバイルアクセスサーバ内の動作は、運用も含め正しく安全に行われるものとする。
- サービス提供機関とモバイルアクセスサーバは事前の契約に基づいて鍵の共有などを行っているものとする。

1.3. セキュリティ要件

前述の前提条件のもとで、提案システムは、以下に示すセキュリティ要件を満たす必要がある。

(1) 不正な携帯電話端末アプリケーションへの対応
  (1-1) 悪意のある携帯電話端末アプリケーションが耐タンパデバイス内のセキュアデータへのアクセスの防止

(2) 通信路の安全性の確保
  (2-1) サービス提供機関―(共通アプリ)―モバイルアクセスサーバ間の通信路の安全性の確保
  (2-2) モバイルアクセスサーバー(共通アプリ)ー耐タンパデバイス間の通信路の安全性の確保
  (2-3) サービス提供機関ーモバイルアクセスサーバ間の通信路の安全性の確保

(3) 成りすまし防止
  (3-1) サービス提供機関の成りすましの防止
  (3-2) モバイルアクセスサーバの成りすましの防止

1.4. 全体システム構成

本実証事業のシステムの全体構成を**図 1-4** に示す。**図 1-4** に示すように、行政機関や、金融機関や民間企業なども含めて複数のサービス提供機関が、携帯電話端末を使う利用者に対して種々のサービスを提供することを想定している。今回対象とするサービスは、何らかの価値を有する情報（ID 情報）を利用者に付与して、その ID 情報を利用することを前提としたサービスを対象とする。ID 情報は、利用者の携帯電話端末からアクセスできる耐タンパデバイスに保存するものとする。

耐タンパデバイスへの情報の書き込み、および読み込みには、通常、サービス提供機関が個別に耐タンパデバイスの自身の領域に対してセキュアなチャネルを構築し、そのチャネルを経由してのみ読み書きが可能となる。今回の提案では、複数のサービス提供機関への負担を軽減するために、前記耐タンパデバイスへの情報の読み書きを代行するモバイルアクセスサーバをモバイルアクセスシステム側に用意し、サービス提供機関の負担を軽減する。

また、耐タンパデバイスと直接データの送受信を行う携帯アプリに関しても、従来であれば個々のサービス提供機関が自身のサービスのために携帯アプリを個別に開発する必要があったが、今回の提案では、複数のサービス提供機関が共通的に利用できる共通アプリで処理することとする。

また、耐タンパデバイスへの ID 情報の読み書きに対する結果通知サービスや、耐タンパデバイスからの ID 情報の読み込みを本人認証に利用したのちの実際のサービスなどは、Web ベースで提供されることを想定している。よって、携帯電話端末内での共通アプリと

ブラウザの連携、モバイルアクセスシステム内でのモバイルサイトとモバイルアクセスサーバの連携を実現することで安全なサービス提供の基盤を実現する。

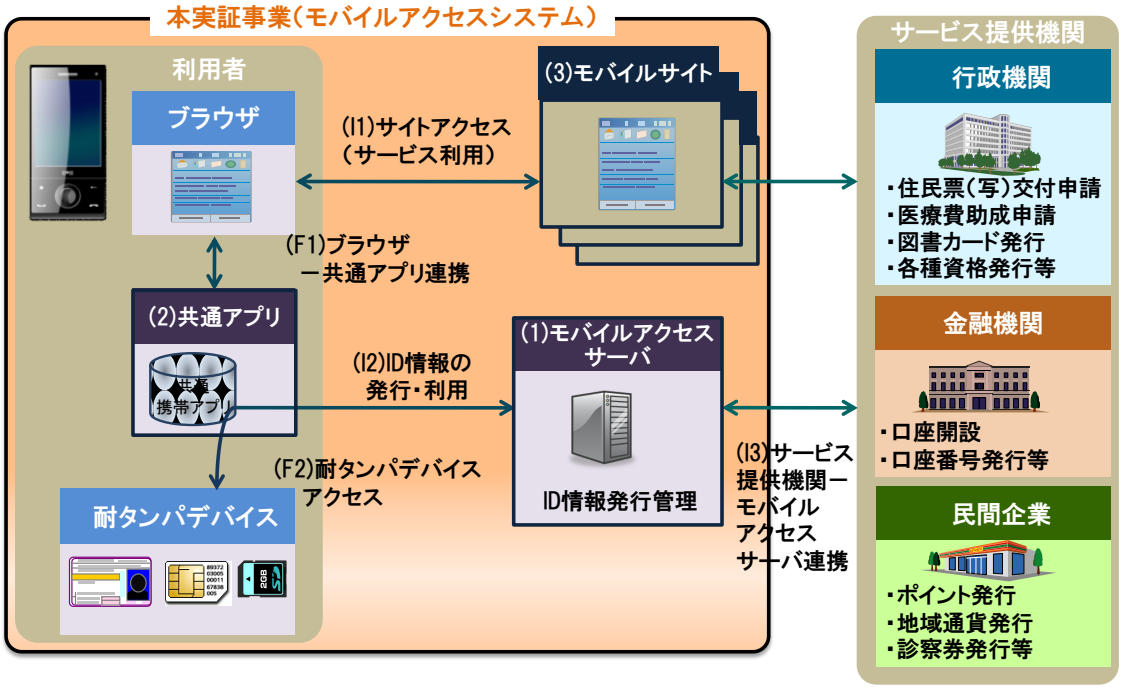

図 1-4 本実証事業のシステムの全体構成

エンティティの説明

- サービス提供機関:行政機関や金融機関、民間企業などの ID 情報を利用者に付与して、その ID 情報を利用することを前提としたサービスを提供する機関
- モバイルサイト：サービス提供機関ごとに構築する携帯電話端末向けサイト
- モバイルアクセスサーバ：サービス提供機関との契約に基づいて、利用者の耐タンパデバイスとセキュアな通信路を確保し暗号化した IC カードコマンドの送受信を行うサーバ。
- 共通アプリ：モバイルアクセスサーバと通信を行い、受信したデータを耐タンパデバイスに送信し、耐タンパデバイスからのレスポンスをモバイルアクセスサーバに返信する携帯アプリ。共通アプリ自身は秘密情報を保持しない設計とすることでオープンな携帯電話端末においても安全性を確保する。
- 耐タンパデバイス：IC チップを搭載したデバイス。携帯電話端末と非接触 IC 通信（NFC）で通信を行うフルサイズの IC カードや、IC チップを搭載したフラッシュメモリ型のデバイス、UICC（Universal Integrated Circuit Card）などを想定する。

具体的には、下記の流れで処理が実行される。

1. 利用者が携帯電話端末内のブラウザでモバイルサイトにアクセスする（図 1-5 の(I1)）。
2. モバイルサイトからのレスポンスによってブラウザから共通アプリが起動される（図 1-5 の(F1)）。

3. 共通アプリからモバイルアクセスサーバに接続し、モバイルアクセスサーバから ID 情報の発行を受ける（図 1-5 の(I2)）。
4. 発行する ID 情報の管理はサービス提供機関が行う。当該 ID 情報をサービス提供機関からモバイルアクセスサーバに通知する。（図 1-5 の(I3)のようにサービス提供機関とモバイルアクセスサーバ間で直接通知する方法と、(I1)と(I2)のインタフェースを使って共通アプリを経由して通知する方法がある）。
5. 上記ステップで発行された ID 情報を共通アプリが耐タンパデバイスに書き込む（図 1-5 の(F2)）。
6. 書き込み後の結果通知は、モバイルアクセスサーバを経由して、さらにブラウザを経由して、モバイルサイトに通知され、最終的にブラウザの画面でユーザに通知される。

課題アでは、図 1-5 の破線で示した範囲を検討範囲とする。破線の矩形が新規に導入されるサブシステム（機能）であり、破線の楕円が、新規に定義するインタフェースである。

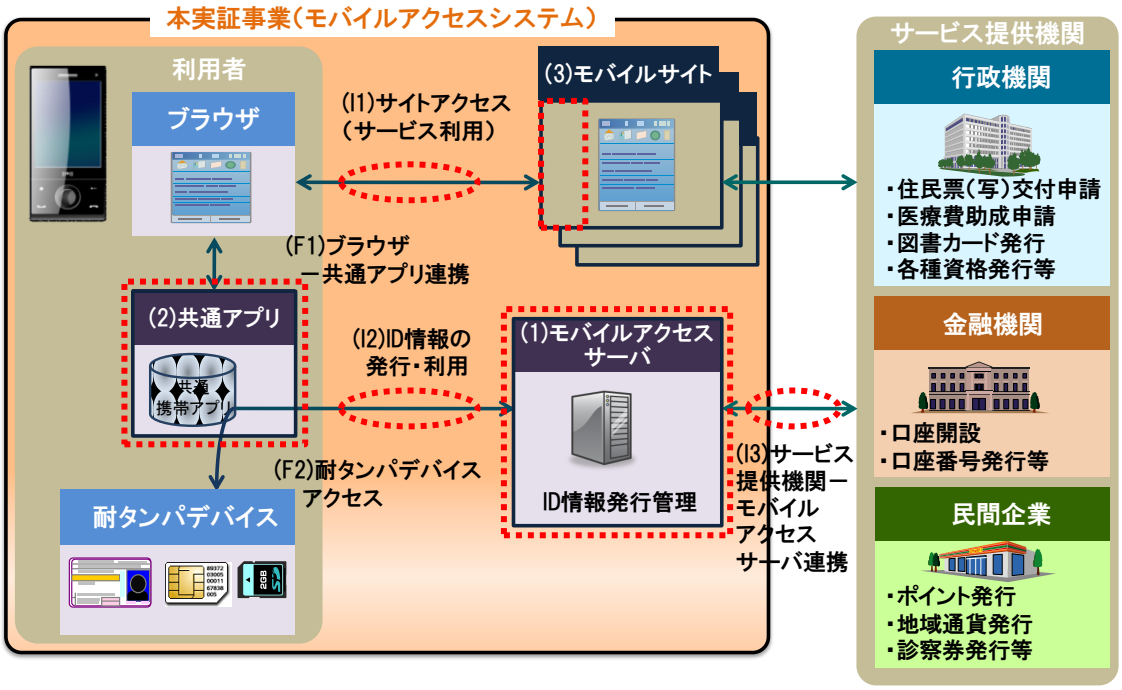

図 1-5 課題アの検討範囲

なお、ブラウザと共通アプリの連携に関しては、現状利用できる携帯電話端末の機能に準ずる。また、耐タンパデバイスと共通アプリのインタフェースに関しては、耐タンパデバイスの種類によって使われるデバイスドライバなどが変わる事が想定されるためハードウェアレベルでの規定は行わない。さらに、モバイルサイトとサービス提供機関は同一機関が運営すると考えられるため、モバイルサイトーサービス提供機関間のインタフェースの規定は行わない。

1.5. システム概要

1.5.1. 機能の概要

以下に、各エンティティの機能の概要を記述する。

（1）モバイルアクセスサーバ（全機能）（**図 1-5** の(1)）

- 受付処理機能：
  サービス提供機関から送信された情報(サービス事業者 ID、受付番号、APDU 生成年月日、APDU 実行順序、APDU コマンド)を受け取り、情報が正しい場合は、受け取った情報を DB に登録する。
- 共通アプリアクセス機能：
  サービス提供機関から共通アプリ経由で転送されるデータが本当に正しいサービス提供機関から送信されたデータなのかを確認する。さらに、携帯電話端末内の共通アプリを経由して、耐タンパデバイスとセキュアセッションを確立し、携帯電話端末内の共通アプリに対して暗号化されたコマンドを送受信し、結果をサービス提供機関に返信する。

（2）共通アプリ（全機能）（**図 1-5** の(2)）

- APDU 転送機能：
  モバイルアクセスサーバから受信した暗号化されたコマンドを耐タンパデバイスへ送信し、耐タンパデバイスからのレスポンスをモバイルアクセスサーバに返信する。

（3）モバイルサイト（一部機能）（**図 1-5** の(3)）

- ID 情報発行機能：
  耐タンパデバイスに送信したいコマンドをモバイルアクセスサーバに移譲し、かつ、携帯電話端末のブラウザを経由して、共通アプリを起動させ、耐タンパデバイスに ID 情報を送信する。
- 処理結果受信機能：
  モバイルアクセスサーバから耐タンパデバイス内での処理結果を受信し、返される処理結果が本当に正しいモバイルアクセスサーバから送信されたデータなのかを確認する。

1.5.2. インタフェースの概要

以下に、各インタフェースの概要を記述する。

（1）共通アプリ起動インタフェース（**図 1-5** の I1 および F1）

- JavaScript で共通アプリを起動する。
- 要求元：モバイルサイト（ブラウザ経由)、応答先：共通アプリ
- 通信形式：実行時パラメータ

（２）共通アプリ連携インタフェース（**図 1-5** の I2 および F2)

- 耐タンパデバイスに対してセキュアにアクセスするために共通アプリとモバイルアクセスサーバが通信する。
- 要求元：共通アプリ、要求先：モバイルアクセスサーバ
- 通信形式：HTTPS（XML）

（３）APDU 移譲インタフェース（**図 1-5** の I3)

- 耐タンパデバイスに対して実行する APDU コマンドをサービス提供機関からモバイルアクセスサーバに移譲する。
- 要求元：サービス提供機関、要求先：モバイルアクセスサーバ
- 通信形式：HTTPS（XML）

（４）APDU 実行結果通知インタフェース（**図 1-5** の I3)

- 耐タンパデバイスに対して実行した APDU コマンドの結果をモバイルアクセスサーバからサービス提供機関に通知する。
- 要求元：モバイルアクセスサーバ、要求先：サービス提供機関
- 通信形式：HTTPS（XML）

（５）再アクセスインタフェース（**図 1-5** の I1,I2 および F1)

- モバイルアクセスサーバから共通アプリに処理終了を通知し、共通アプリからサービス提供機関へ再アクセスする。
- 要求元：モバイルアクセスサーバ（共通アプリ経由、ブラウザ経由)、要求先：サービス提供機関
- 通信形式：HTTPS

1.5.3. プロトコルの概要

以下に各エンティティ間のデータの流れ（プロトコル）に関して記述する。**図 1-6** は、サービス提供機関による耐タンパデバイスとの ID 情報の書き込み、及び読み込みの際のデータの流れを示す簡易的な図である。

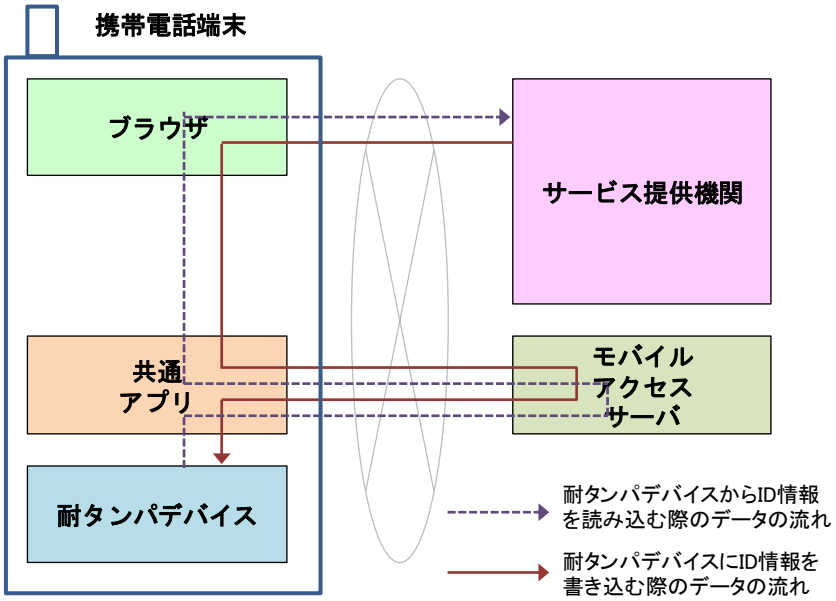

図 1-6 データの流れを示す簡略図

図 1-6 に示す実線の矢印が、サービス提供機関から耐タンパデバイスへ送信する ID 情報の流れである。サービス提供機関はブラウザ経由で共通アプリを起動し、その後、共通アプリからモバイルアクセスサーバに接続され、モバイルアクセスサーバは、共通アプリ経由で耐タンパデバイスに ID 情報を送信する。その際に、モバイルアクセスサーバと耐タンパデバイスはセキュアな通信路を確立する。点線の矢印は、耐タンパデバイスからモバイルアクセスサーバへ送信する ID 情報の流れである。サービス提供機関は、ブラウザ経由で共通アプリを起動し、その後、モバイルアクセスサーバ経由で共通アプリから耐タンパデバイスにアクセスし、ID 情報をモバイルアクセスサーバに送信する。モバイルアクセスサーバは、共通アプリ、ブラウザ経由でサービス提供機関に ID 情報を送信する。

図 1-7 にデータの流れを表わす。さらに詳細なフローを示す。

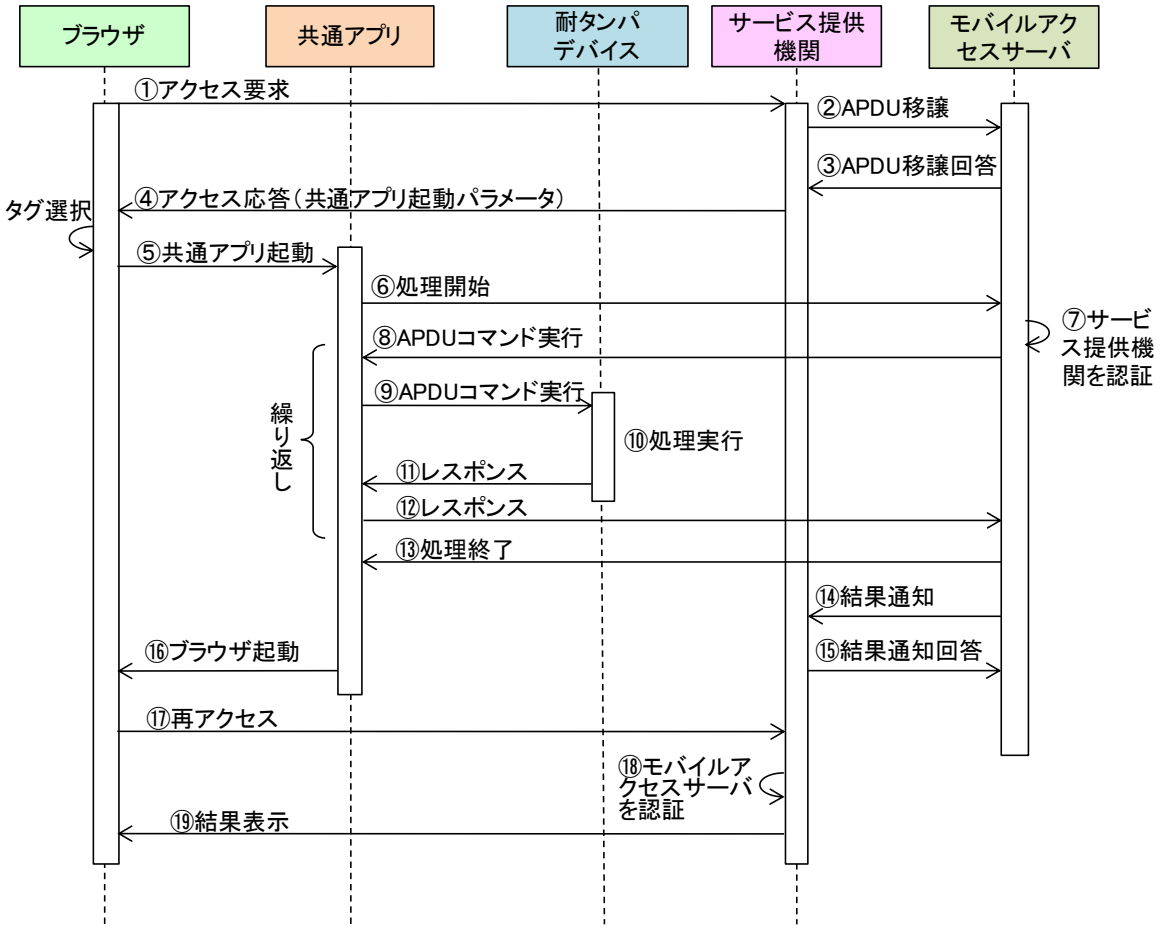

図 1-7 データの流れを示す詳細図

まず、利用者が携帯電話端末のブラウザ経由でサービス提供機関にアクセスする①。サービス提供機関からモバイルアクセスサーバに対して、耐タンパデバイスに送るべきAPDU コマンドを送信し②、回答を受信する③。

次にサービス提供機関から、アクセス応答として、ブラウザに対して共通アプリ起動パラメータを送信する④。ブラウザは、共通アプリを起動し、サービス提供機関から受信したデータを共通アプリに渡す⑤。

耐タンパデバイスにアクセスするための秘密情報を保持しない共通アプリは、そのままでは耐タンパデバイスにアクセスできないため、モバイルアクセスサーバに処理開始要求をを送信する⑥。④、⑤、⑥でサービス提供機関からモバイルアクセスサーバに送信されるデータは暗号化されている。モバイルアクセスサーバは、共通アプリから転送された処理開始要求データが正しいサービス提供機関から送信された要求データだということを確認する⑦。モバイルアクセスサーバは、耐タンパデバイス内のサービス提供機関の管理下の領域の IC カードアプリケーションに送信するための APDU コマンドを共通アプリに返信する⑧。共通アプリは受信した APDU コマンドを耐タンパデバイスに転送する⑨。耐タンパデバイスは、APDU コマンドに従って耐タンパデバイス内で処理を実行し⑩、結果をレスポンスデータとして共通アプリ経由でモバイルアクセスサーバに返す⑪⑫。APDU コ

マンドは複数回実行されることが想定されるため⑧～⑫が繰り返される。なお、この APDU コマンド送受信の初期の段階で、モバイルアクセスサーバと耐タンパデバイス内のサービス提供機関の管理下の領域の IC カードアプリケーションとの間で、セキュアセッションの確立（暗号通信を行うための鍵共有）が行われ、以降の APDU コマンドは安全な通信路内で送受信される。よって APDU コマンドは共通アプリを経由するが共通アプリはその内容を見ることはできない。

サービス提供機関は耐タンパデバイスでの処理結果をモバイルアクセスサーバから受信し⑬、受信結果をモバイルアクセスサーバに返信する⑭。

モバイルアクセスサーバは、共通アプリに対して処理終了通知を送信し⑮、共通アプリはブラウザを起動する⑯。起動されたブラウザでサービス提供機関に再度アクセスする⑰。⑬、⑯、⑰でモバイルアアクセスサーバからサービス提供機関に送信されるデータは暗号化されている。サービス提供機関は共通アプリからブラウザ経由で転送されたデータが正しいモバイルアクセスサーバからのデータであるか否かを確認する⑱。最後にサービス提供機関からブラウザに対して結果を表示させる⑲。

なお、**図 1-7** では、②、③と⑬、⑭のステップにおいて、直接サービス提供機関とモバイルアクセスサーバの間で、APDU コマンドの送信と処理結果の受信を行っているが、②の APDU コマンドの移譲の代わりに、④～⑥のそれぞれの送信データに移譲すべき APDU コマンドを含めることもできる。その場合には、⑬、⑭の処理結果受信の代わりに、⑮～⑰のそれぞれの送信データに処理結果を含める。

また、⑬、⑭の処理結果通知処理は、⑧～⑫の繰り返しが終わった後に一括して行っているが、⑫のレスポンスを受け取る都度、サービス提供機関に処理結果を通知しても良い。

1.6. セキュリティ対策

本節では、2.3 節で示したセキュリティ要件に対する対策を示す。

(1) 不正な携帯電話端末アプリケーションへの対応

(1-1)

【要件】悪意のある携帯電話端末アプリケーションが耐タンパデバイス内のセキュアデータへのアクセスの防止

【対策】耐タンパデバイスへアクセスするためには、耐タンパデバイスと相互認証を行ったうえで安全な通信路を確保（セキュアセッションの確立）するようにし、共有鍵を持たない携帯電話端末アプリケーションは、耐タンパデバイスにアクセスできないようにした。また、共通アプリも共有鍵を持たず、モバイルアアクセスサーバ側に共有鍵を持たせることで、共通アプリがマルウェアやウィルスに感染しても共有鍵が漏洩することがないような設計にした。

(2) 通信路の安全性の確保

(2-1)

【要件】サービス提供機関―(共通アプリ)―モバイルアクセスサーバ間の通信路の安全性の確保

【対策】**図 1-7** のデータの流れを示す詳細図の④、⑤、⑥のサービス提供機関からモバイルアクセスサーバへ送信されるデータおよび、⑬、⑯、⑰でモバイルアアクセスサーバからサービス提供機関に送信されるデータは暗号化される。よって安全性が確保されるとは限らない共有アプリを経由してもデータは漏えいしない。

(2-2)

【要件】モバイルアクセスサーバー(共通アプリ)ー耐タンパデバイス間の通信路の安全性の確保

【対策】モバイルアクセスサーバと耐タンパデバイス間は、GlobalPlatform 仕様に基づく相互認証および暗号通信を行うため安全性は確保される。

(2-3)

【要件】サービス提供機関ーモバイルアクセスサーバ間の通信路の安全性の確保

【対策】2.2 節のセットワークに関する前提条件で示したように、サービス提供機関ーモバイルアクセスサーバ間の通信路の安全性は確保されているという前提を置いている。

(3) 成りすまし防止

(3-1)

【要件】サービス提供機関の成りすましの防止

【対策】**図 1-7** のデータの流れを示す詳細図の⑦で示したように、共通アプリを経由してモバイルアクセスサーバが受信したデータには、サービス提供機関の署名が付与されており、モバイルアクセスサーバはその署名を検証することで、正しいサービス提供機関から送信されたデータだということを確認できる。

(3-2)

【要件】モバイルアクセスサーバの成りすましの防止

【対策】**図 1-7** のデータの流れを示す詳細図の⑱で示したように、共通アプリを経由してサービス提供機関が受信したデータには、モバイルアクセスサーバの署名が付与されており、サービス提供機関はその署名を検証することで、正しいモバイルアクセスサーバから送信されたデータだということを確認できる。

1.7. 機能の詳細

本節では、共通アプリ、モバイルアクセスサーバ、サービス提供機関の各エンティティの機能に関して詳細を記述する。なお、**図 1-7** に示した全体フローと本節で詳述する機能

は下図のような対応関係になっている。

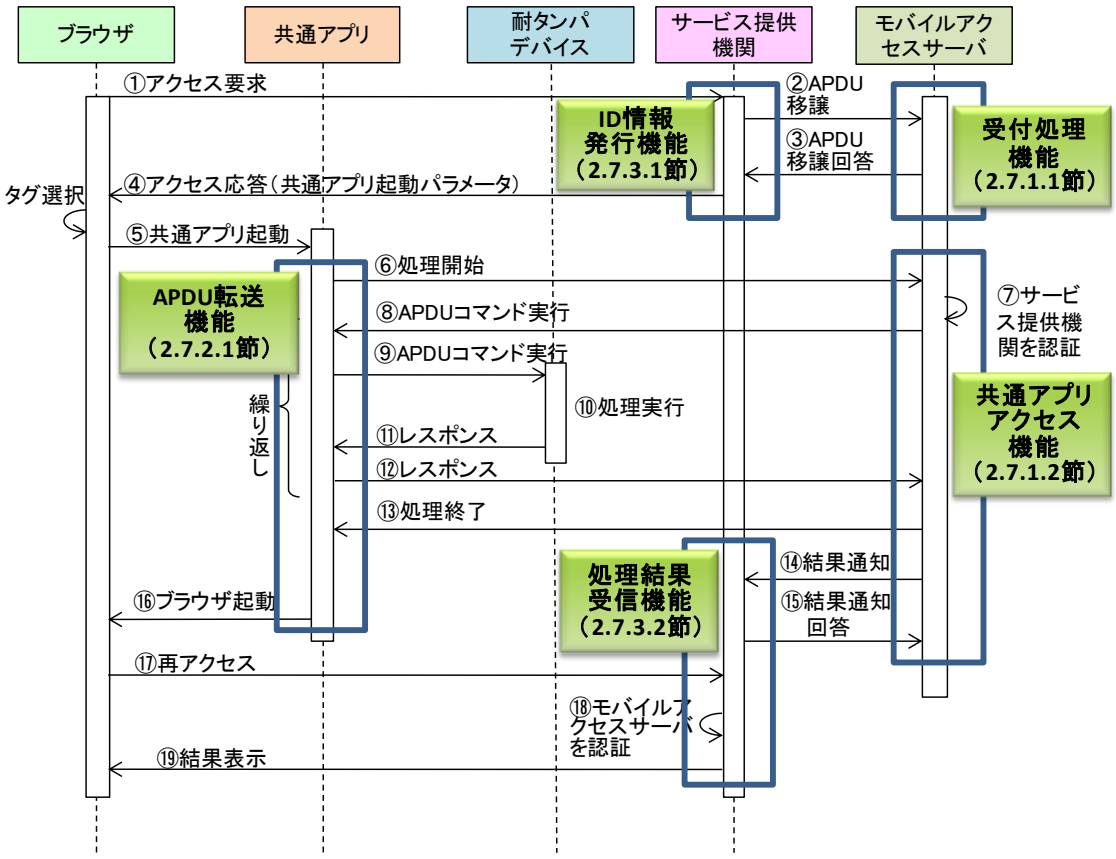

図 1-8 本節の構成と、各機能の対応関係

1.7.1. モバイルアクセスサーバの機能

モバイルアクセスサーバは、サービス提供機関から APDU の移譲を受ける受付処理機能、および、共通アプリに対して、処理開始、セキュアセッション確立、APDU コマンド送信、処理終了を送信する共通アプリアクセス機能を有する。

前提として、モバイルアクセスサーバとサービス提供機関はセキュアな通信が確立されているものとする。

1.7.1.1. 受付処理機能

サービス提供機関から送信された情報(サービス事業者 ID、受付番号、APDU 生成年月日、APDU 実行順序、APDU コマンド)を受け取り、情報が正しい場合は、受け取った情報を DB に

登録する。受付処理機能の処理フローを以下に示す。

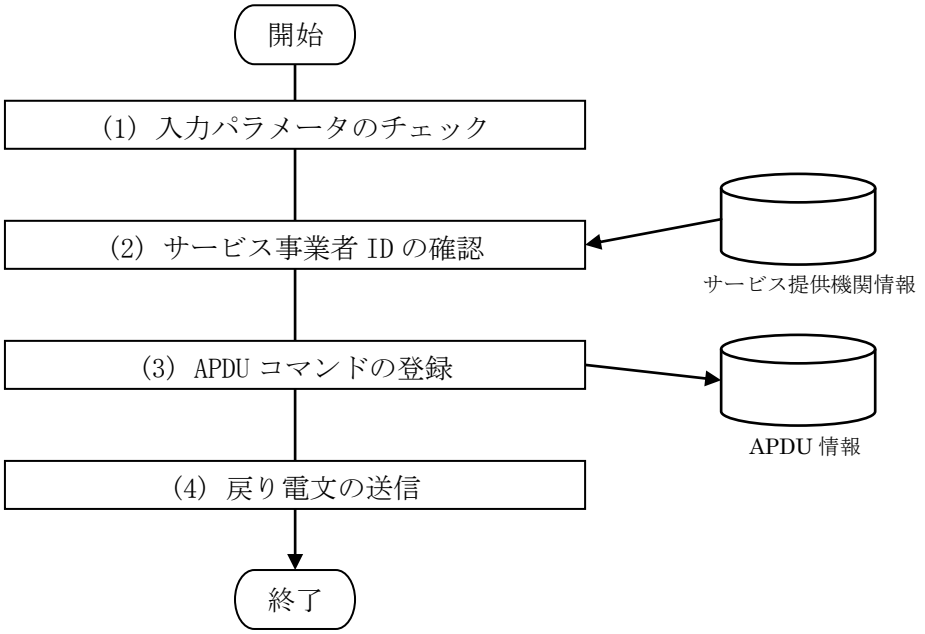

図 1-9 受付処理機能の処理フロー

(1) 入力パラメータのチェック

入力パラメータ(サービス事業者 ID、受付番号、APDU 生成年月日、APDU 実行順序、APDU コマンド)のチェック（文字数などのチェック）を行う。

また､APDU 実行順序と APDU コマンドのペアが 1 つ以上指定されていること､1 番目の APDU コマンドが SELECT であること､APDU 実行順序が 1 からの通番であること等の相関チェックを行う。

入力チェックエラーの場合、処理を中断して(4)の戻り電文の設定処理を実行する。

(2) サービス事業者 ID の確認

受信したサービス提供機関 ID が契約関係にあるサービス提供機関であることを確認する。

(3)APDU コマンドの登録

受信した APDU コマンドを保管する。APDU コマンドが複数ある場合は、その数分登録処理を実行する。

(4) 戻り電文の送信

サービス提供機関に、戻り電文（処理ステータス（00:正常終了、02：アプリエラー)、エラー情報、ADU 受付年月日）を送信する。

1.7.1.2. 共通アプリアクセス機能

共通アプリに対して、処理開始、セキュアセッション確立、APDU コマンド送信、処理終了を送信する。また、共通アプリから実行結果を受信してサービス提供機関に送信する。

共通アプリアクセス機能の処理フローを以下に示す。下図に示すように、共通アプリから受け取った処理 ID を判定して処理を分岐する。

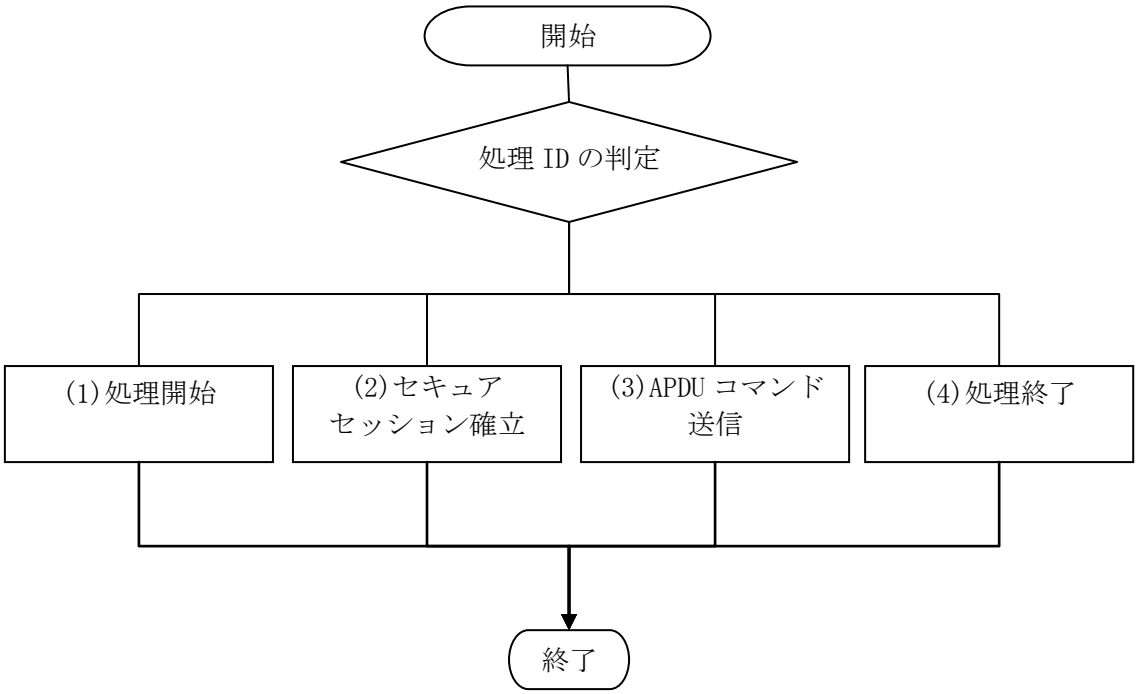

図 1-10 共通アプリアクセス機能の処理フロー

- **処理開始**

処理 ID が処理開始(001)の場合、初期処理を行い耐タンパデバイスへの接続要求を送信する。処理フロー及び処理詳細を以下に示す。

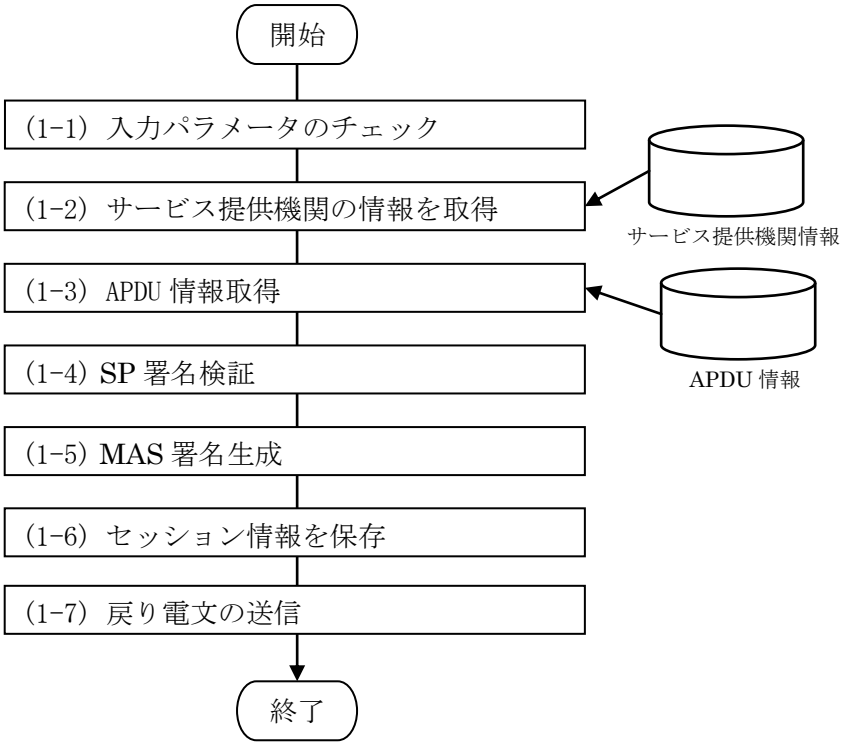

図 1-11 処理開始における処理フロー

(1-1) 入力パラメータのチェック

入力パラメータ（サービス事業者 ID、受付番号、SP 署名）のチェックを行う。

(1-2) サービス提供機関の情報を取得

サービス事業者 ID から、登録されているサービス提供機関の情報（サービス提供機関のの URI,鍵,鍵のバージョン）を取得する。

(1-3) APDU 情報取得

サービス事業者 ID と受付番号から APDU 情報（APDU 生成年月日、APDU 受付年月日、処理ステータス）を取得する。

(1-4) SP 署名検証

受信したサービス事業者 ID、受付番号と取得した APDU 生成年月日からハッシュ値を算出する（SHA 方式でハッシュ値を算出する）。

次に、SP 署名の復号を行う。具体的には、受信した SP 署名を秘密鍵で復号する。

最後に、SP 署名の検証を行う。具体的には、算出したハッシュ値と SP 署名を復号した値

を比較する。

(1-5) MAS 署名生成

受信したサービス事業者 ID、受付番号と取得した APDU 受付年月日を使用して MAS 署名を生成する（SHA 方式でハッシュ値を取得したものを RSA 方式で暗号化する)。

(1-6) セッション情報を保存

サービス提供機関の情報（サービス提供機関のの URI,鍵,鍵のバージョン）や共通アプリから受け取ったサービス事業者 ID、受付番号と生成した MAS 署名をセッション情報として保存する（これらの情報は以降処理で使用する)。

(1-7) 戻り電文の送信

共通アプリに対して、戻り電文（処理 ID（101：Connect 要求戻り)、104:処理終了（エラー発生時)、MAS 署名（エラー発生時)）を送信する。

(2) セキュアセッション確立

処理 ID が Connect 結果(002)の場合、またはレスポンス APDU(003)でかつセキュアセッションの確立前の場合、セキュアセッションの確立処理を行う。また、セッションからセキュアセッション確立状態の情報を取得して処理を分岐する。SELECT コマンド実行後は処理 ID がレスポンス APDU(003)である為、セッションに保持したセキュアセッション確立状態から次に実行する処理を判定する。処理フロー及び処理詳細を以下に示す。

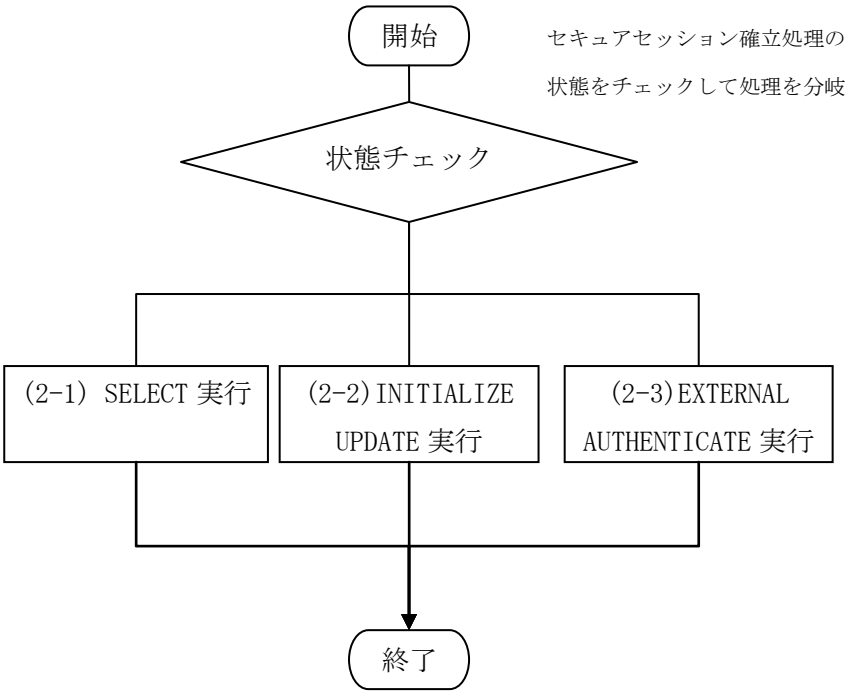

図 1-12 セキュアセッション確立における処理フロー

(2-1) SELECT コマンド処理

セキュアセッションの状態を確認し、セッション確立前であれば、SELECT コマンドを送信する。処理フロー及び処理詳細を以下に示す。

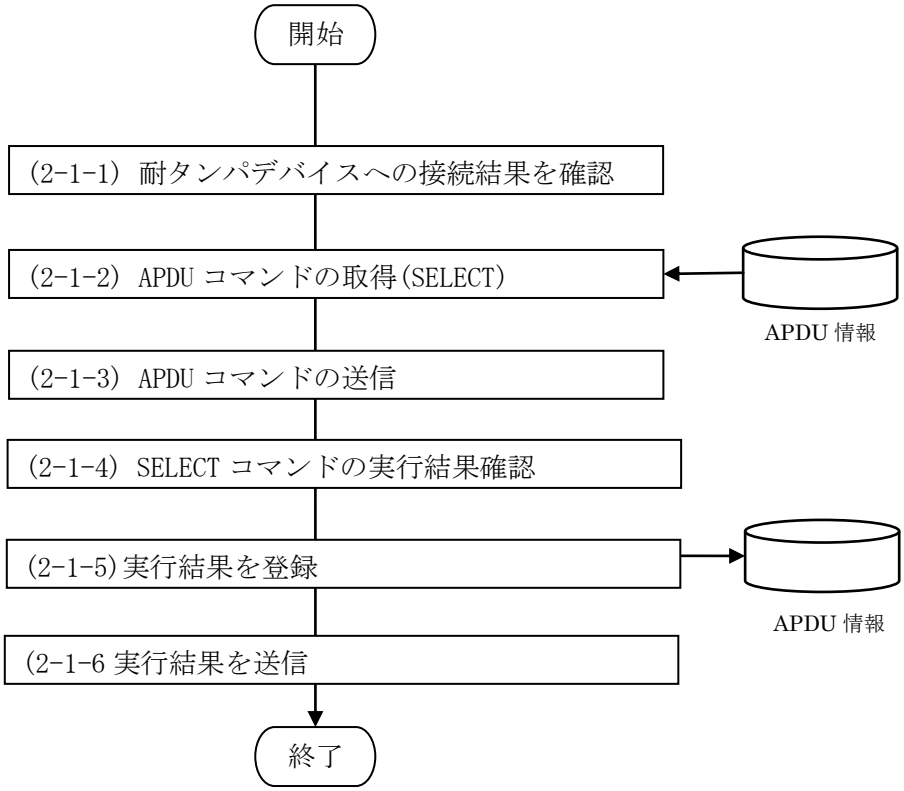

図 1-13 SELECT コマンド処理における処理フロー

(2-1-1) 耐タンパデバイスへの接続結果を確認

耐タンパデバイスへの接続結果が正常であることを確認する。

(2-1-2) APDU コマンドの取得

実行順序が 1 番目の APDU コマンド(SELECT)を取得する。その後、セキュアセッション確立状態を SELECT コマンド実行中に更新する。

(2-1-3) APDU コマンドの送信

耐タンパデバイスに SELECT コマンドを送信する。

(2-1-4) SELECT コマンドの実行結果確認

APDU コマンド実行結果が正常(SW1 が 0x90、SW2 が 0x00)であることを確認する。

(2-1-5) APDU コマンド(SELECT)の実行結果を登録

APDU コマンド(SELECT)の実行結果を保管する。また、現在実行中の APDU コマンド(APDU

実行順)を 1 に設定する。

(2-1-6) APDU コマンド(SELECT)の実行結果を送信

サービス提供機関に APDU コマンドの実行結果（受付番号、APDU 実行順序、APDU レスポンス番号、APDU レスポンス、処理ステータス）を送信する。なお、本ステップで実行結果を送信せず、(4)の終了処理でまとめてサービス提供機関に送信することも可能である。

(2-2) INITIALIZE UPDATE コマンド処理

次に、INITIALIZE UPDATE コマンドを実行する。処理フロー及び処理詳細を以下に示す。

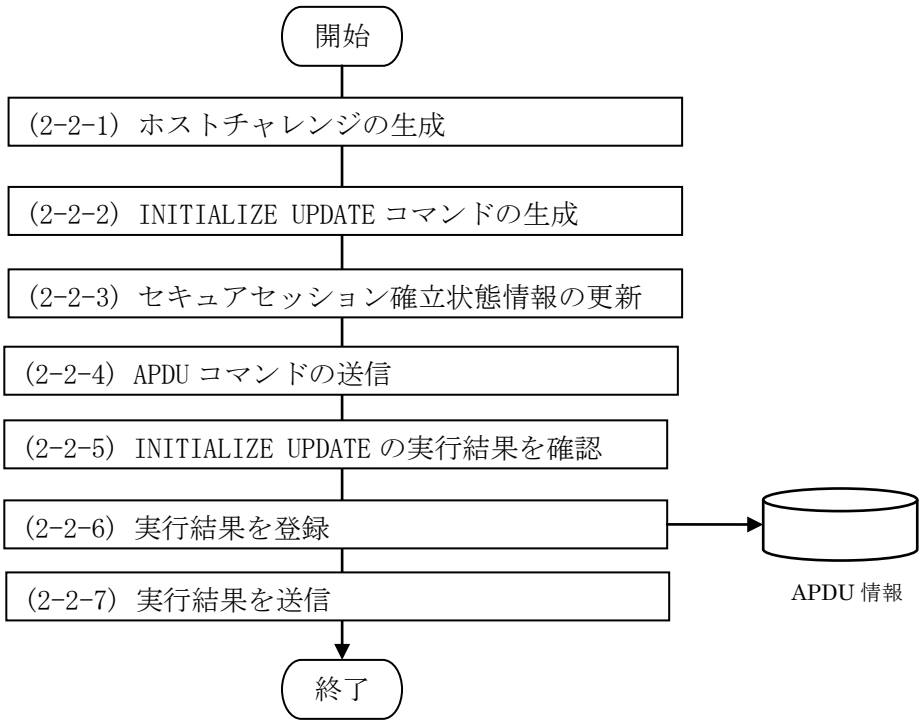

図 1-14 INITIALIZE UPDATE コマンド処理における処理フロー

(2-2-1) ホストチャレンジ(認証乱数)の生成

8 バイト長のホストチャレンジを生成する。また、作成したホストチャレンジはセッションに保存する。この値は、カード認証演算(INITIALIZE UPDATE のレスポンス判定)で使用する。

(2-2-2) INITIALIZE UPDATE コマンドの生成

INITIALIZE UPDATE コマンド(GlobalPlatform 仕様に準拠)を生成する。

(2-2-3) セキュアセッション確立状態情報の更新

セッション情報に保持したセキュアセッション確立状態を INITIALIZE UPDATE コマンド実行中に更新する。

(2-2-4) APDU コマンドの送信

耐タンパデバイスに INITIALIZE UPDATE コマンドコマンドを送信する。

(2-2-5) INITIALIZE UPDATE の実行結果を確認

エラー種別が正常(00)、および APDU コマンド実行結果が正常(SW1 が 0x90、SW2 が 0x00)であることを確認する。

SW1 が 0x61 または 0x6C の場合には、レスポンスデータが存在する場合の為、GET RESPONSE コマンドを送信してデータの取得を行う。

(2-2-6) APDU コマンド(INITIALIZE UPDATE)の実行結果を登録

APDU コマンド(INITIALIZE UPDATE)の実行結果を保管する。

(2-2-7) APDU コマンド(INITIALIZE UPDATE)の実行結果を送信

サービス提供機関に APDU コマンドの実行結果（受付番号、APDU 実行順序、APDU レスポンス番号、APDU レスポンス、処理ステータス）を送信する。なお、本ステップで実行結果を送信せず、(4)の終了処理でまとめてサービス提供機関に送信することも可能である。

(2-3) EXTERNAL AUTHENTICATE コマンド処理

INITIALIZE UPDATE コマンドの実行結果を確認する。実行結果が正常の場合、EXTERNAL AUTHENTICATE コマンドを生成する。処理フロー及び処理詳細を以下に示す。

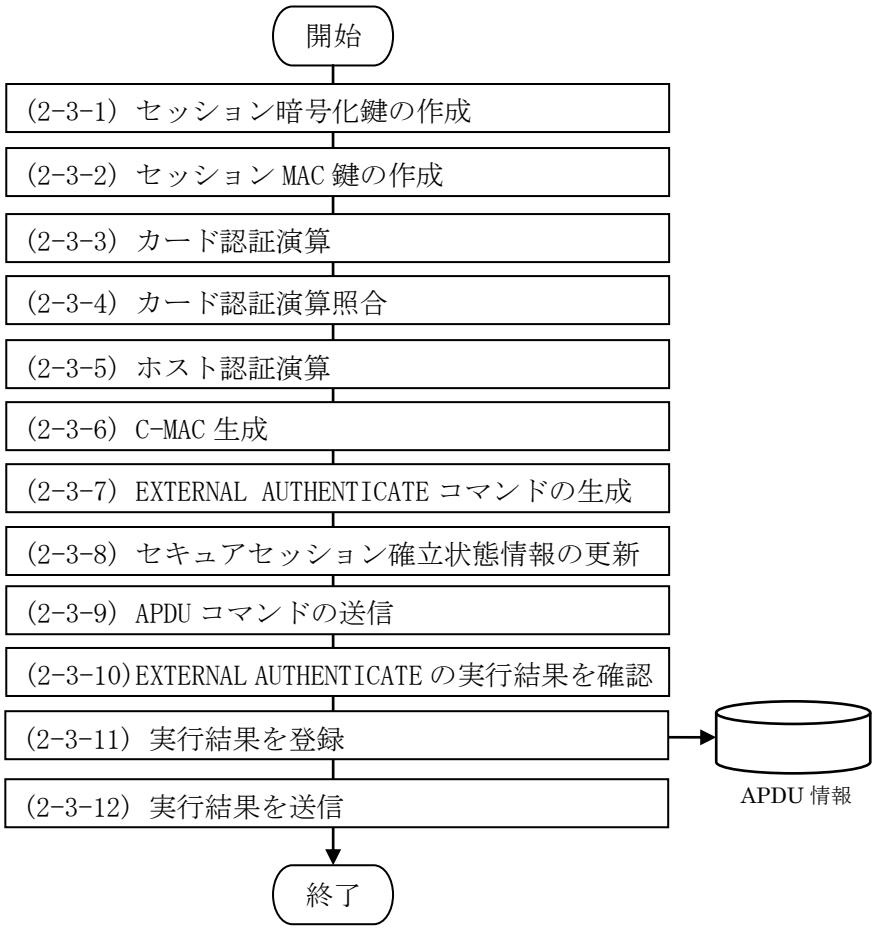

図 1-15 EXTERNAL AUTHENTICATE コマンド処理における処理フロー

(2-3-1) セッション暗号化鍵の作成

セッション情報からベース鍵情報を取得する。ベース鍵からセッション暗号化鍵を作成する。また、作成した鍵はセッション情報に保存する。Data 領域の暗号化と APDU レスポンスの復号化で使用する。

(2-3-2) セッション MAC 鍵の作成

ベース鍵からセッション MAC 鍵を作成する。また、作成した鍵はセッション情報に保存する。C-MAC 作成で使用する。

(2-3-3) カード認証演算

セッション暗号化鍵、ホストチャレンジ、カードチャレンジの情報からカード認証演算結果を算出する。ホストチャレンジはセッションから取得する。また、カードチャレンジ

はAPDUレスポンスから取得する。

(2-3-4) カード認証演算照合

算出したカード認証演算結果と共通アプリから受け取ったカード認証演算結果が正しいことを確認する。

(2-3-5) ホスト認証演算

セッション暗号化鍵、ホストチャレンジ、カードチャレンジの情報からホスト認証演算結果を算出する。

(2-3-6) C-MAC生成

セッションMAC鍵とAPDUコマンドからC-MACを生成する。また、作成したC-MACはセッション情報に保存する。次にC-MACを作成する時に使用する。

(2-3-7) EXTERNAL AUTHENTICATEコマンドの生成

EXTERNAL AUTHENTICATEコマンド（GlobalPlatform仕様に準拠）を生成する。

(2-3-8) セキュアセッション確立状態情報の更新

セッションに保持したセキュアセッション確立状態を EXTERNAL AUTHENTICATE コマンド実行中に更新する。

(2-3-9) APDUコマンドの送信

耐タンパデバイスにEXTERNAL AUTHENTICATEコマンド送信する。

(2-3-10) EXTERNAL AUTHENTICATEコマンドの実行結果を確認

EXTERNAL AUTHENTICATEコマンドの実行結果を確認する。エラー種別が正常(00)、およびAPDUコマンド実行結果が正常(SW1が0x90、SW2が0x00)であることを確認する。

SW1が0x61または0x6Cの場合には、レスポンスデータが存在する場合の為、GET RESPONSEコマンドを送信してデータの取得を行う。

(2-3-11) APDUコマンド(EXTERNAL AUTHENTICATE)の実行結果を登録

APDUコマンド(EXTERNAL AUTHENTICATE)の実行結果を保管する。

(2-3-12) APDUコマンド(EXTERNAL AUTHENTICATE)の実行結果を送信

サービス提供機関にAPDUコマンドの実行結果（受付番号、APDU実行順序、APDUレスポンス番号、APDUレスポンス、処理ステータス）を送信する。なお、本ステップで実行結果を送信せず、(4)の終了処理でまとめてサービス提供機関に送信することも可能である。

(3) APDU コマンド送信

処理 ID がレスポンス APDU(003)であり、セキュアセッション確立状態が、EXTERNAL AUTHENTICATE コマンド実行中、またはセキュアセッション確立成功の場合、実行する APDU コマンドを送信する。EXTERNAL AUTHENTICATE コマンド実行中の場合は、その結果を確認する。処理フロー及び処理詳細を以下に示す

。

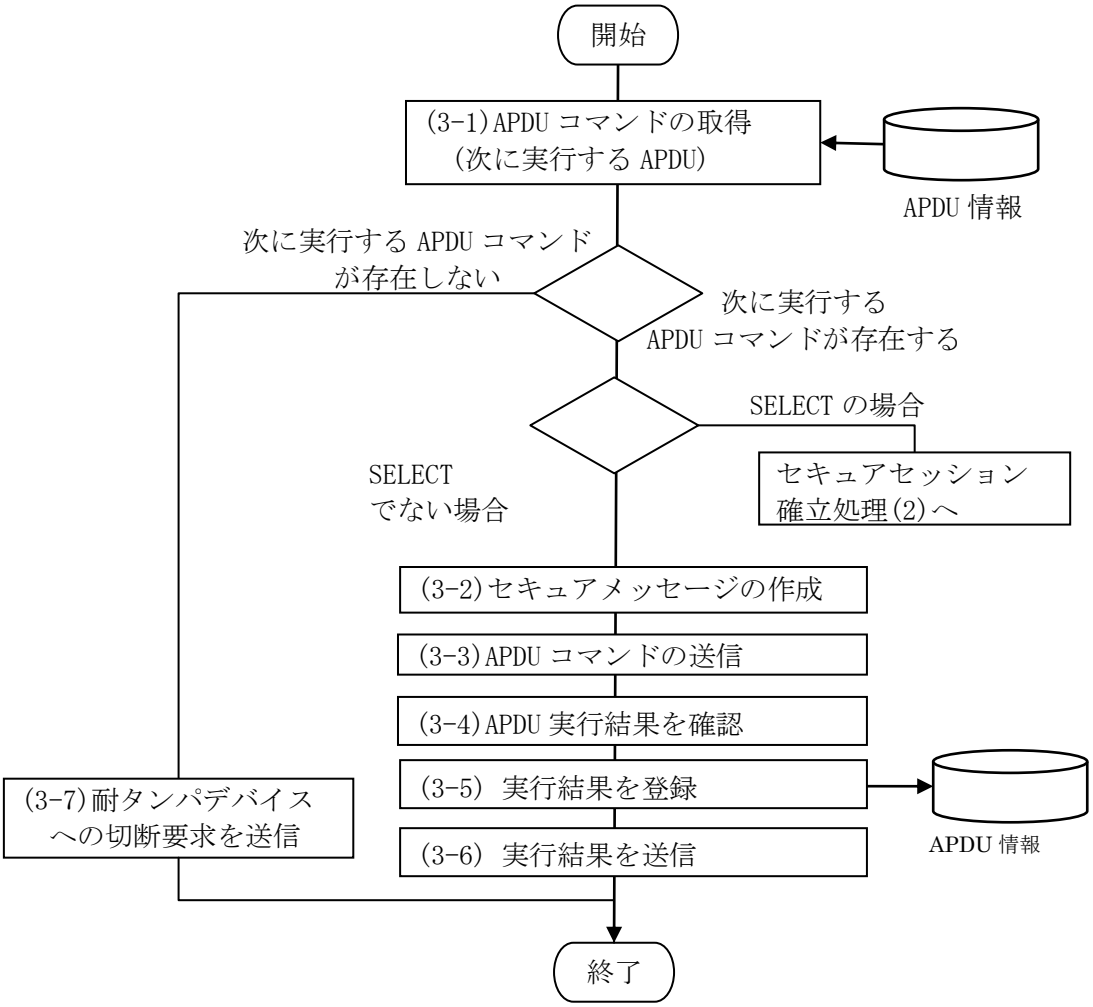

図 1-16 APDU コマンド送信における処理フロー

(3-1) APDU コマンドの取得

次に実行する APDU コマンドを取得する。

(3-2) セキュアメッセージの作成

次に実行する APDU コマンドが存在し、その APDU コマンドが SELECT ではない場合、セキュアメッセージの作成を行う。具体的には、C-MAC の生成と Data 領域の暗号化を行う(SELECT コマンドの場合は、(2)のセキュアセッション確立処理を実行する)。

・C-MAC の作成

セッション MAC 鍵と APDU コマンドから C-MAC を作成する。また、セッションに保持してある C-MAC を作成した C-MAC に更新する。セッション MAC 鍵はセッション情報から取得する。

・Data 領域の暗号化

セッション暗号化鍵で Data 領域を暗号化する。セッション暗号化鍵はセッション情報から取得する。

・セキュアメッセージを作成

セキュアメッセージ(GlobalPlatform 仕様準拠)を作成する。

(3-3) APDU コマンドの送信

耐タンパデバイスに対してセキュアメッセージ化した APDU コマンドを送信する。

(3-4) APDU コマンドの実行結果を確認

APDU コマンドの実行結果を確認する。エラー種別が正常(00)、および APDU コマンド実行結果が正常(SW1 が 0x90、SW2 が 0x00)であることを確認する。

SW1 が 0x61 または 0x6C の場合には、レスポンスデータが存在する場合の為、GET RESPONSE コマンドを送信してデータの取得を行う。

(3-5) APDU コマンドの実行結果を登録

APDU コマンドの実行結果を保管する。

(3-6) APDU コマンドの実行結果を送信

サービス提供機関に APDU コマンドの実行結果（受付番号、APDU 実行順序、APDU レスポンス番号、APDU レスポンス、処理ステータス）を送信する。なお、本ステップで実行結果を送信せず、(4)の終了処理でまとめてサービス提供機関に送信することも可能である。

(3-7) 耐タンパデバイスへの切断要求を送信

ステップ(3-1)で、次に実行するAPDUコマンドが存在しない場合（APDUコマンドが全て実行済みの場合)、耐タンパデバイスへ切断の要求を行う。

(4) 処理終了

処理IDがDisConnect結果(004)の場合、処理終了のステータスを送信する。処理フロー及び処理詳細を以下に示す。

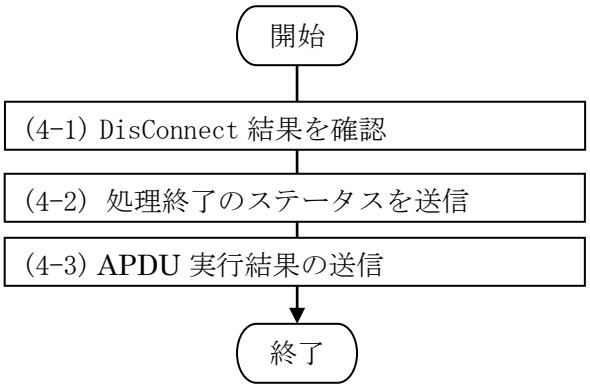

図 1-17 処理終了における処理フロー

(4-1) DisConnect結果を確認

耐タンパデバイスの切断結果の確認を行う。

(4-2) 処理終了のステータスを送信

共通アプリに対して戻り電文（処理ID（104：処理終了)、MAS署名）を送信する。

(4-3) APDU実行結果の送信

サービス提供機関に対して、APDU処理の結果（受付番号、APDU実行順序、APDUレスポンス番号、APDUレスポンス、処理ステータス）を送信する。

1.7.2. 共通アプリの機能

共通アプリは、耐タンパデバイスとの接続、切断を行い、モバイルアクセスサーバから受信したAPDUコマンドを耐タンパデバイスに送信し、また、その結果をモバイルアクセスサーバに送信する機能を有する。

ブラウザから起動され、モバイルアクセスサーバと連動して耐タンパデバイスとの接続、切断を行い、モバイルアクセスサーバから受信したAPDUコマンドを耐タンパデバイスに送信する。モバイルアクセスサーバから処理終了要求を受信すると、ブラウザを立ち上げ処理を終了する。

1.7.2.1. APDU転送機能

共通アプリのAPDU転送機能の処理フローを以下に示す。

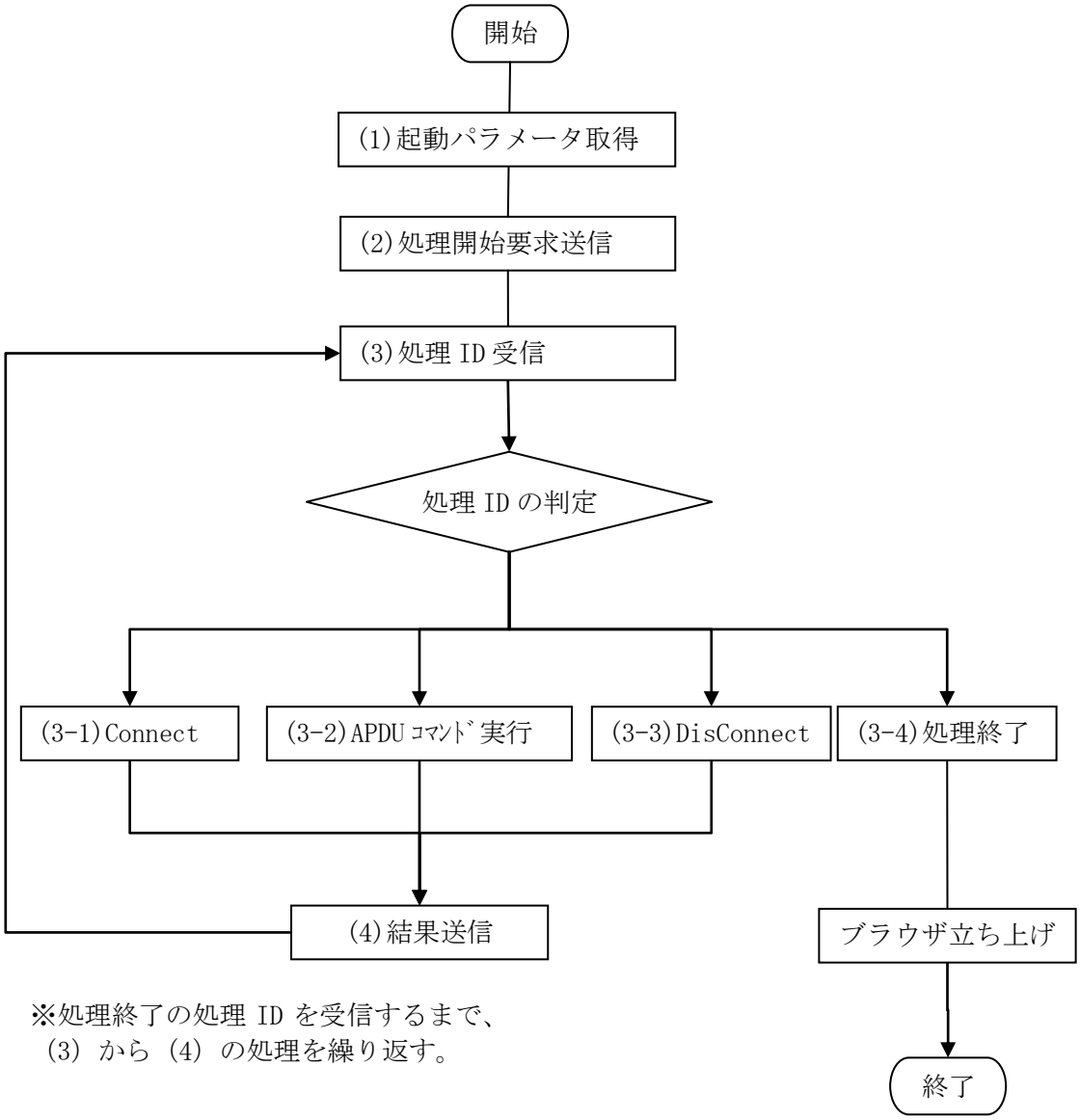

図 1-18 共通アプリの APDU 転送機能の処理フロー

(1) 起動パラメータ取得

共通アプリは、起動パラメータから、サービス事業者 ID、受付番号、戻り URL、モバイルアクセスサーバ URL、SP 署名を取得する。ここで、SP 署名とはサービス提供機関で生成した署名値である。

(2) 処理開始要求送信

モバイルアクセスサーバに、送信電文（処理 ID（001:処理開始）、エラー種別（00:正常）、

サービス事業者 ID、受付番号、SP 署名）を送信する。

(3) 処理 ID 受信

モバイルアクセスサーバから処理 ID を受信する。処理 ID の値によって処理を分岐し、処理終了要求を受信するまで処理を繰り返す。

(3-1) Connect(処理 ID：101)の場合

耐タンパデバイスに接続(Connect)する。

モバイルアクセスサーバに Connect 結果（処理 ID（002：Connect 結果)、エラー種別（00：正常)、カード情報（00:SD カード,01:UIM カード,02:非接触 IC カード)）を送信する。

(3-2) APDU コマンド実行(処理 ID：102)の場合

APDU が SELECT コマンドの場合には、SELECT 処理を実行し、モバイルアクセスサーバに電文（処理 ID（003：レスポンス APDU)、エラー種別（00：正常)、SW1,SW2（0x9000)、取得したレスポンス APDU）を送信する。

APDU が SELECT コマンド以外の場合には、受信した APDU コマンドをそのまま耐タンパデバイスに送信する。

その後、モバイルアクセスサーバに電文（処理 ID（003：レスポンス APDU)、エラー種別（00：正常)、取得した SW1,SW2、取得したレスポンス APDU）を送信する。

(3-3) DisConnect(処理 ID：103)の場合

耐タンパデバイスとの接続を解除（DisConnect）する。

モバイルアクセスサーバに電文（処理 ID（004：DisConnect 結果)、エラー種別（00：正常)）を送信する。

(3-4) 処理終了(処理 ID：104)の場合

「戻り URL + MAS 署名」を指定してブラウザを起動し、処理を終了する。ここで、MAS 署名とはモバイルアクセスサーバで生成した署名値である。

1.7.3. サービス提供機関の機能

サービス提供機関は、耐タンパデバイスに ID 情報を送信する ID 情報発行機能、および、モバイルアクセスサーバから耐タンパデバイスでの処理結果を受信する機能を有する。

なお、前提として、サービス提供機関とモバイルアクセスサーバはセキュアな通信が確立されていることとする。

1.7.3.1. ID 情報発行機能

耐タンパデバイスへ ID 情報を発行する処理フローを以下に示す。ただし「任意」と書かれた処理はサービスに依存する任意の処理である。

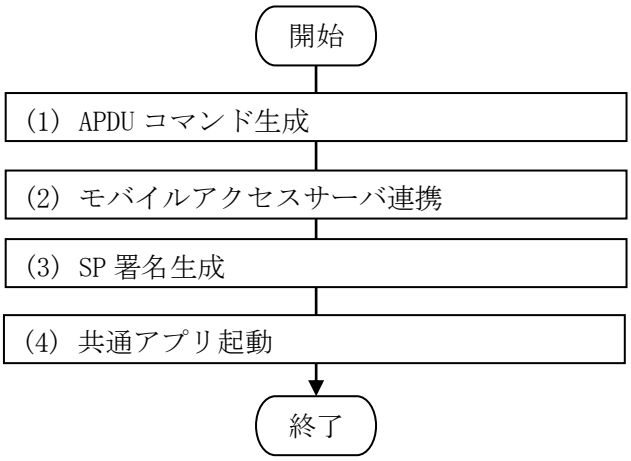

図 1-19 耐タンパデバイスへの ID 情報発行における処理フロー

(1) APDU コマンド生成

任意の APDU コマンドを生成する。

(2) モバイルアクセスサーバ連携

モバイルアクセスサーバに APDU 情報(サービス事業者 ID、受付番号、APDU 生成年月日、APDU 実行順序、APDU コマンド)を送信する。(実行する APDU の数だけ APDU 実行順序と APDU コマンドのペアが存在する)。

(3) SP 署名生成

サービス事業者 ID、受付番号、APDU 生成年月日を使用して SP 署名を生成する。具体的には、SHA 方式でハッシュ値を取得したものを RSA 方式で暗号化する。

(4) 共通アプリ起動

ブラウザに対して共通アプリ起動パラメータ（サービス事業者 ID、受付番号、戻り URL、モバイルアクセスサーバ URL、SP 署名）を送信し、共通アプリを起動する。

1.7.3.2. 処理結果受信機能

耐タンパデバイスから処理結果を受信する処理フローを以下に示す。ただし「任意」と書かれた処理はサービスに依存する任意の処理である。

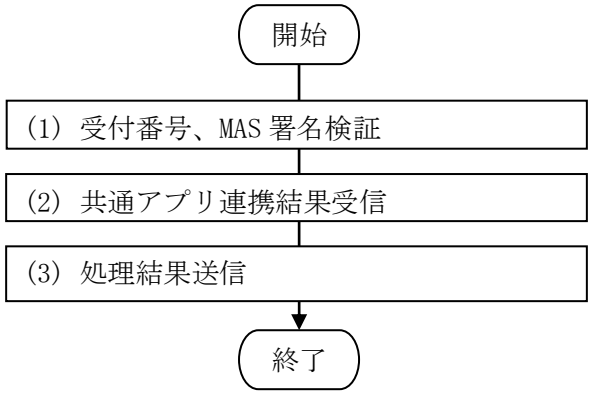

図 1-20 耐タンパデバイスからの処理結果受信における処理フロー

(1) 受付番号、MAS 署名検証

ブラウザから送信された GET パラメータから受付番号、MAS 署名を取得し、検証する。具体的には、まず、サービス事業者 ID、受信した受付番号、APDU 受付年月日からハッシュ値を算出する（SHA 方式でハッシュ値を算出する）。次に、受信した MAS 署名を秘密鍵で復号する。算出したハッシュ値と MAS 署名を復号化した値を比較する。値が異なった場合は、エラー画面を表示する。

(2)共通アプリ連携結果受信

モバイルアクセスサーバから APDU コマンドの実行結果（受付番号、APDU 実行順序、APDU レスポンス番号、APDU レスポンス、処理ステータス）を受信する。

(3) 処理結果送信

モバイルアクセスサーバに処理結果（成功、失敗）を送信する。

1.8. インタフェースの詳細

本節では、ブラウザ、共通アプリ、耐タンパデバイス、サービス提供機関、モバイルアクセスサーバの各エンティティ間のインタフェースに関して詳細を記述する。なお、図 1-7 に示した全体フローと本節で詳述するインタフェースは下図のような対応関係になっている。

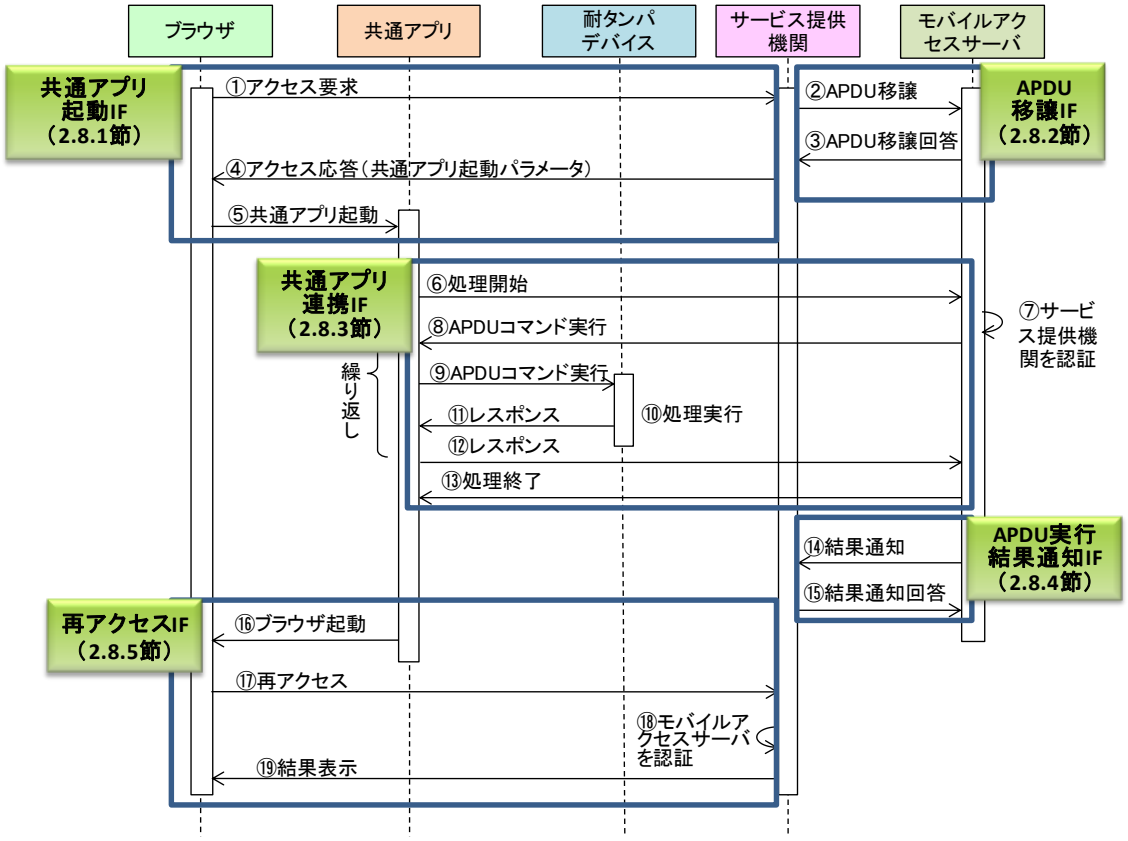

図 1-21 エンティティ間のインタフェース

本システムで使用する通信 IF を下の表に示す。

表 1-1 インタフェース一覧

#	IF 名	概要	要求元	応答先	通信形式
1	共通アプリ起動	JavaScript で共通アプリを起動する。	ブラウザ	共通アプリ	実行時パラメータ

#	IF 名	概要	要求元	応答先	通信形式
2	APDU 移譲	耐タンパデバイスに対して実行する APDU コマンドをサービス提供機関からモバイルアクセスサーバに移譲する。	サービス提供機関	モバイルアクセスサーバ	HTTPS／※XML
3	共通アプリ連携	耐タンパデバイスに対してセキュアにアクセスするために共通アプリとモバイルアクセスサーバが通信する IF。	共通アプリ	モバイルアクセスサーバ	HTTPS／※XML
4	APDU 実行結果通知	耐タンパデバイスに対して実行した APDU コマンドの結果をモバイルアクセスサーバからサービス提供機関に通知する。	モバイルアクセスサーバ	サービス提供機関	HTTPS／※XML
5	再アクセス	共通アプリからサービス提供機関の処理結果画面へ接続する IF	共通アプリ	サービス提供機関	HTTPS

1.8.1. 共通アプリ起動インタフェース

共通アプリを起動する際の起動パラメータ（Android OS の場合）を以下に示す。

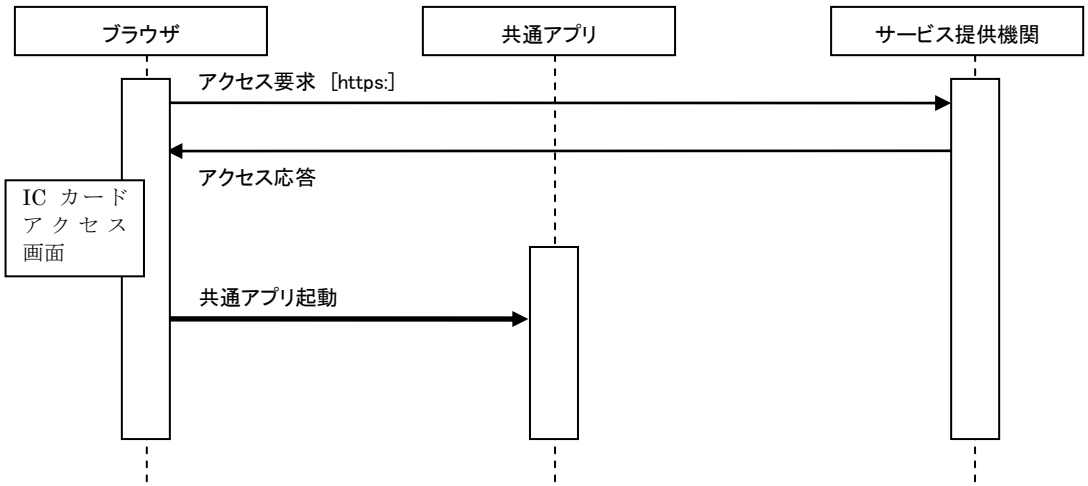

図 1-22 共通アプリ起動シーケンス図

表 1-2 共通アプリ起動時の要求例

```
function startCommonApl(param){
var hrefString="commonApl: " + param;
location.href=hrefString;
}
```

表 1-3 共通アプリ起動時のパラメータ

#	項目名	パラメータ名	属性	桁数	説明
1	起動パラメータ	param	varchar	-	共通アプリの起動パラメータ

表 1-4 起動パラメータの構成

spId=XXXXXXXX&rcptNum=XXXXXXXXXXXXXXXX&rtnUrl=XXXXXXXXXXXXXXX&masUrl=XXXXXXXXXXXXXXX&spSign=XXXXXXXXXXXXXXXX

※spId=、&rcptNum=、&rtnUrl=、masUrl、spSign が必ず存在すること。

表 1-5 起動パラメータの構成要素の詳細

#	項目名	パラメータ名	属性	桁数	説明
1	サービス事業者 ID	spId	char	8	サービス事業者 ID
2	受付番号	rcptNum	char	16	サービス提供機関が採番した受付番号(日付+8 桁の番号)
3	戻り URL	rtnUrl	varchar	-	戻り URL
4	モバイルアクセスサーバ URL	masUrl	varchar	-	モバイルアクセスサーバの URL
5	SP 署名	spSign	varchar	256	Base64 エンコードした SP 署名

1.8.2. APDU 移譲インタフェース

耐タンパデバイスに対して実行する APDU コマンドをサービス提供機関からモバイルアクセスサーバに移譲する通信インタフェースを以下に示す。

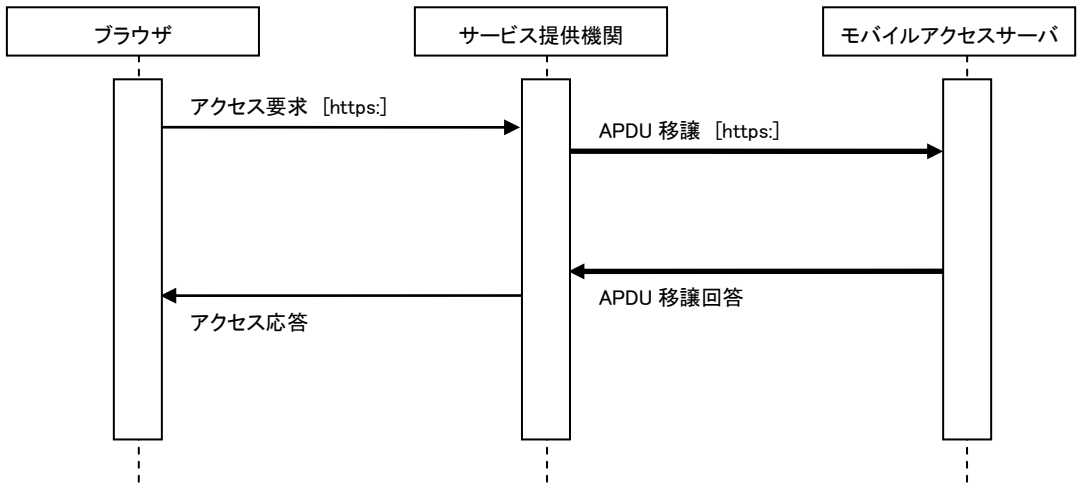

図 1-23 APDU 移譲シーケンス図

表 1-6 APDU 移譲時の要求例

```
POST /xxxxx/xxxx/xxxxx HTTPS/1.1
Host: xxxxx.com
Content-Type: application/xml
Content-Length: バイト数

<?xml version="1.0" encoding="UTF-8" ?>
<request>
  <strSpId>XXXX</strSpId>
  <strRcptNum>XXXX</strRcptNum>
  <strApduGeneDate>XXXX</ strApduGeneDate >
  <apduCmnds>
    <apduCmnd>
      <iApduOrder>XXXX</iApduOrder>
      <byApduCmndAry>XXXX</byApduCmndAry>
    </apduCmnd>
    ・・・<apduCmnd>は実行する APDU の数分定義する
  </apduCmnds>
</request>
```

表 1-7 APDU 移譲時の要求 IF

#	項目名	パラメータ名	属性	桁数	説明
1	サービス事業者 ID	strSpId	char	8	サービス事業者 ID
2	受付番号	strRcptNum	char	16	サービス提供機関が採番した受付番号 (日付+8 桁の番号)
3	APDU 生成年月日	strApduGeneDate	char	14	APDU を生成した年月日 (YYYYMMDDhhmmss)
4	APDU 実行順序	iApduOrder	varchar	3	APDU コマンドの実行順序
5	APDU コマンド	byApduCmndAry	varchar	※説明参照	Base64 エンコードした APDU コマンド ※桁数は生成した APDU の約 3 分の 4 倍

※実行する APDU の数だけ APDU 実行順序と APDU コマンドのペアが存在する

表 1-8 APDU 移譲時の応答例

```
HTTPS/1.1 200 OK
Content-Type: application/xml
Content-Length: バイト数

<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <strProcStatus>XXXX</strProcStatus>
  <strErrInfo>XXXX</strErrInfo>
  <strApduRcptDate>XXXX</strApduRcptDate>
</response>
```

表 1-9 APDU 移譲時の応答 IF

#	項目名	パラメータ名	属性	桁数	説明
1	処理ステータス	strProcStatus	char	2	処理ステータス
2	エラー情報	strErrInfo	varchar	20	エラー情報
3	APDU 受付年月日	strApduRcptDate	char	14	APDU を受け付けた年月日 (YYYYMMDDhhmmss)

1.8.3. 共通アプリ連携インタフェース

耐タンパデバイスに対してセキュアにアクセスするために共通アプリとモバイルアクセスサーバ間でデータを送受信する通信インタフェースを以下に示す。

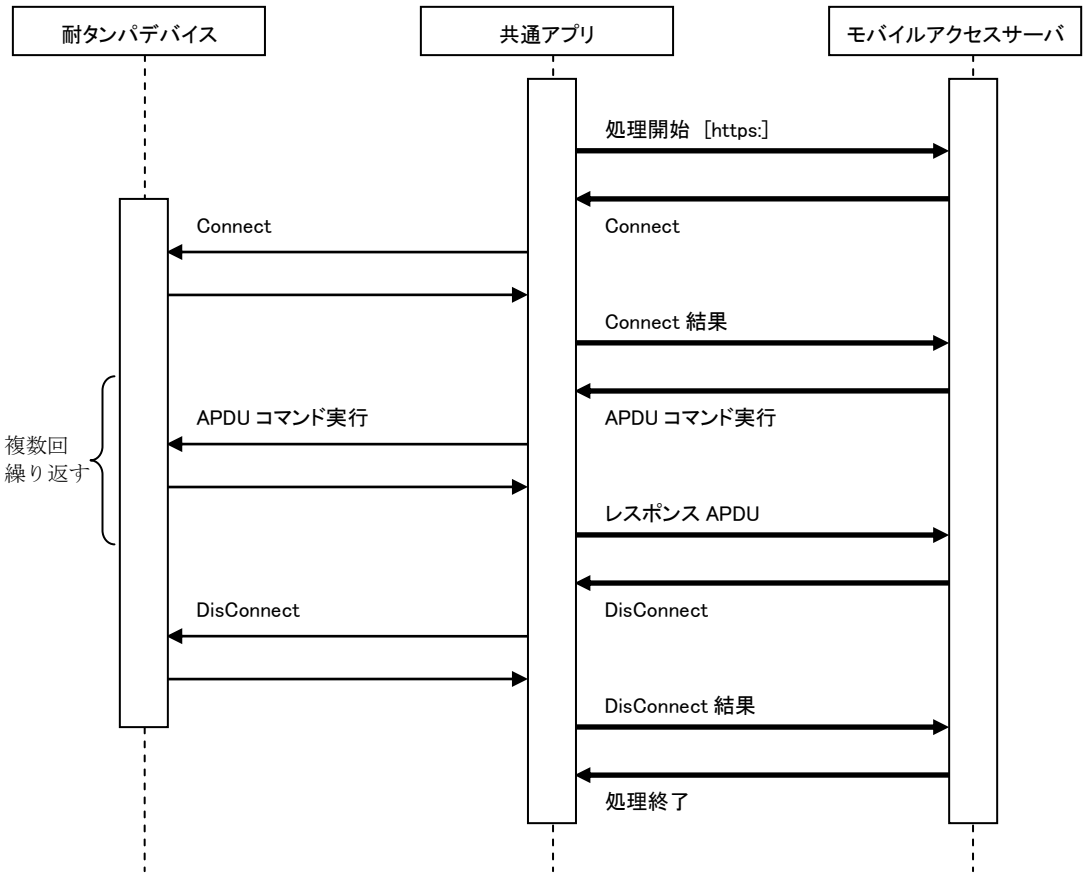

図 1-24 共通アプリ連携シーケンス図

表 1-10 共通アプリ連携時の要求例

```
POST /xxxxx/xxxx/xxxxx HTTPS/1.1
Host: xxxxx.com
Content-Type: application/xml
Content-Length: バイト数

<?xml version="1.0" encoding="UTF-8" ?>
<request>
   <strProcId>XXXX</strProcId>
   <strProcStatus>XXXX</strProcStatus>
   <strSpId>XXXX</strSpId>
   <strRcptNum>XXXX</strRcptNum>
   <byRespSWAry>XXXX</byRespSWAry>
   <byRespApduDataAry>XXXX</byRespApduDataAry>
   <strConnectCard>XXXX</strConnectCard>
   <strErrInfo>XXXX</strErrInfo>
   <strSpSign>XXXX</strSpSign >
</request>
```

表 1-11 共通アプリ連携時の要求 IF

#	項目名	パラメータ名	属性	桁数	説明
1	処理 ID	strProcId	char	3	処理 ID
2	処理ステータス	strProcStatus	char	2	処理ステータス
3	サービス事業者 ID	strSpId	char	8	サービス事業者 ID ※未設定の場合あり
4	受付番号	strRcptNum	char	16	サービス提供機関が採番した受付番号 (日付+8 桁の番号) ※未設定の場合あり
5	SW1,SW2	byRespSWAry	char	3	Base64 エンコードした APDU コマンド実行結果 ※未設定の場合あり
6	レスポンス APDU	byRespApduDataAry	varchar	※説明参照	Base64 エンコードしたレスポンス APDU ※未設定の場合あり ※桁数はレスポンス APDU の約 3 分の 4 倍
7	カード情報	strConnectCard	char	2	Connect したカードの情報 ※未設定の場合あり
8	エラー情報	strErrInfo	varchar	20	エラー情報
9	SP 署名	strSpSign	varchar	256	Base64 エンコードした SP 署名 ※処理 ID：001 の場合のみ設定する

表 1-12 共通アプリ連携時の要求 IF における処理 ID

#	コード	説明
1	001	処理開始
2	002	Connect 結果
3	003	レスポンス APDU
4	004	DisConnect 結果

表 1-13 共通アプリ連携時の要求 IF におけるカード情報

#	コード	説明
1	00	SD カード
2	01	SIM カード

表 1-14 共通アプリ連携時の応答例

```
HTTPS/1.1 200 OK
Content-Type: application/xml
Content-Length: バイト数

<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <strProcIdRtn>XXXX</strProcIdRtn>
  <byApduAry>XXXX</byApduAry>
  <byApduAry>XXXX</byApduAry>
</response>
```

表 1-15 共通アプリ連携時の応答 IF

#	項目名	パラメータ名	属性	桁数	説明
1	処理 ID(戻り)	strProcIdRtn	varchar	3	処理 ID(戻り)
2	APDU コマンド	byApduAry	varchar	※説明参照	Base64 エンコードした APDU コマンド ※桁数は APDU コマンドの 3 分の 4 倍 ※未設定の場合あり
3	MAS 署名	strMasSign	varchar	256	Base64 エンコードした MAS 署名 ※処理 ID(戻り)：104 の場合のみ設定する

表 1-16 共通アプリ連携時の応答 IF における処理 ID(戻り)

#	コード	説明
1	101	Connect
2	102	APDU コマンド実行
3	103	DisConnect
4	104	処理終了

1.8.4. APDU 実行結果通知インタフェース

耐タンパデバイスに対して実行した APDU コマンドの結果をモバイルアクセスサーバからサービス提供機関に通知する通信印インタフェースを以下に示す。

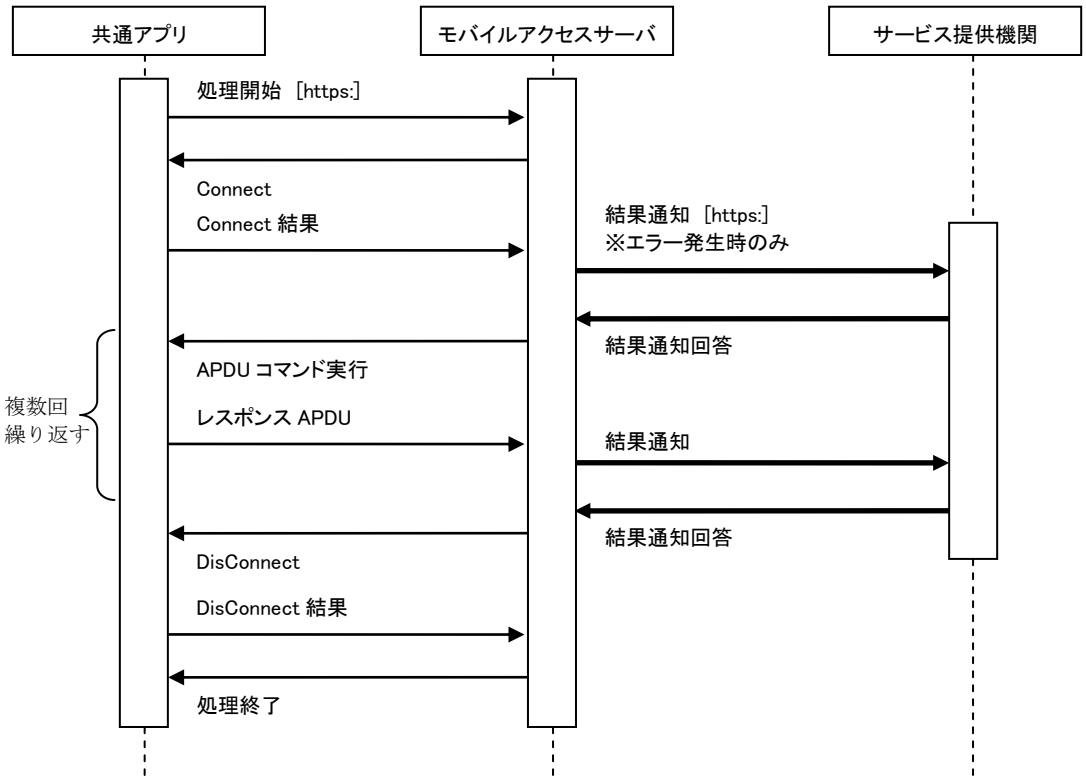

図 1-25 APDU 実行結果通知シーケンス図

表 1-17 APDU 実行結果通知時の要求例

```
POST /xxxxx/xxxx/xxxxx HTTPS/1.1
Host: xxxxx.com
Content-Type: application/xml
Content-Length: バイト数

<?xml version="1.0" encoding="UTF-8" ?>
<request>
   <strRcptNum>XXXX</strRcptNum>
   <iApduOrder>XXXX</iApduOrder>
   <iRespApduNum>XXXX</iRespApduNum>
   <byRespSWAry>XXXX</byRespSWAry>
   <byRespApduDataAry>XXXX</byRespApduDataAry>
   <strProcStatus>XXXX</strProcStatus>
   <strErrInfo>XXXX</strErrInfo>
</request>
```

表 1-18 APDU 実行結果通知時の要求 IF

#	項目名	パラメータ名	属性	桁数	説明
1	受付番号	strRcptNum	char	16	サービス提供機関が採番した受付番号(日付+8 桁の番号)
2	APDU 実行順序	iApduOrder	varchar	3	APDU コマンドの実行順序
3	APDU レスポンス番号	iRespApduNum	varchar	3	APDU レスポンス番号
4	SW1,SW2	byRespSWAry	char	3	Base64 エンコードした APDU コマンド実行結果 ※未設定の場合あり
5	レスポンス APDU	byRespApduDataAry	varchar	※説明参照	Base64 エンコードしたレスポンス APDU ※未設定の場合あり ※桁数はレスポンス APDU の約 3 分の 4 倍
6	処理ステータス	strProcStatus	char	2	処理ステータス
7	エラー情報	strErrInfo	varchar	20	エラー情報 ※未設定の場合あり

表 1-19 APDU 実行結果通知時の応答例

```
HTTPS/1.1 200 OK
Content-Type: application/xml
Content-Length: バイト数

<?xml version="1.0" encoding="UTF-8" ?>
<response>
  <strResult>XXXX</strResult>
</response>
```

表 1-20 APDU 実行結果通知時の応答 IF

#	項目名	パラメータ名	属性	桁数	説明
1	処理結果	strResult	varchar	5	処理結果 true：正常、false：異常

1.8.5. 再アクセスインタフェース

共通アプリからサービス提供機関の処理結果画面へ接続するインタフェースを以下に示す。

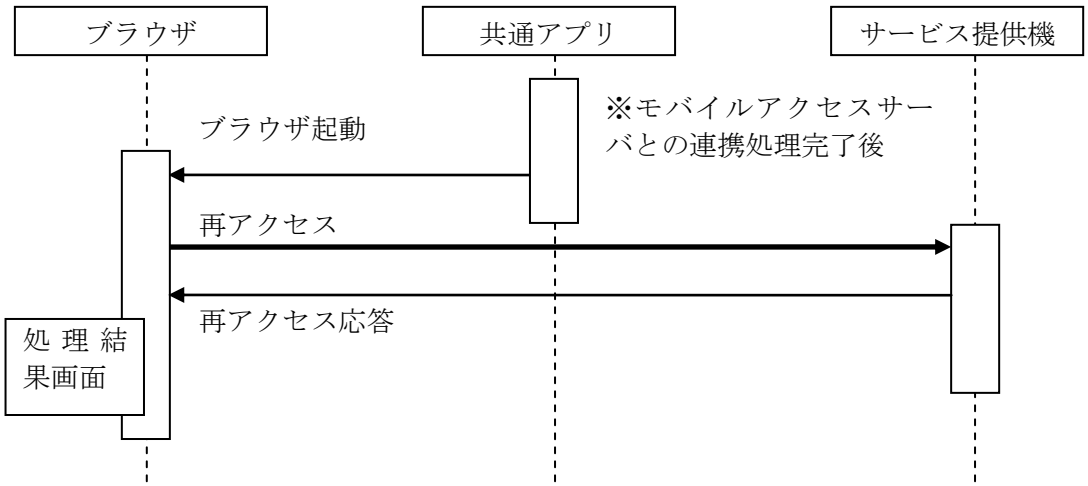

図 1-26 再アクセスシーケンス図

表 1-21 再アクセス時の要求例

https://(戻りURL)?rcptNum=XXXXXXXXXXXXXXXX&masSign=XXXXXXXXXXXXXXXX

※rcptNum=、masSign=が必ず存在すること。

表 1-22 再アクセス時における要求の構成要素

#	項目名	パラメータ名	属性	桁数	説明
1	戻り URL	-	varchar	-	サービス提供機関の処理結果画面の URL
2	受付番号	rcptNum	char	16	サービス提供機関が採番した受付番号 (日付+8 桁の番号)
3	MAS 署名	masSign	varchar	256	Base64 エンコードした MAS 署名

1.9. まとめ

課題アでは、サービス提供機関が携帯電話端末利用者の耐タンパデバイスへ ID 情報の書き込みと読み込みを安全かつ容易に行うことを対象範囲とした検討を行った。

上記対象範囲に関する現状の課題として、複数のサービス提供機関が、携帯電話端末利用者の耐タンパデバイスへの ID 情報の書き込みや読み込みを行おうとする場合、いままではサービス提供機関ごとに携帯アプリを開発・運用する必要があった。また、利用者は各サービス提供機関が提供する携帯アプリを個別にダウンロード・インストールする必要があった。さらに今後は携帯電話端末の OS のオープン化が進むことが想定されるため、携帯アプリのセキュリティを確保する仕組みも必要となっていた。

このような課題を解決するために、モバイルアクセスシステムを提案した。具体的には、

サービス提供機関が耐タンパデバイスにアクセスする際に共通的に利用できるモバイルアクセスサーバおよび携帯電話端末上の共通アプリからなるモバイルアクセスシステムの提案を行った。

課題アで提案したモバイルアクセスシステムを導入することにより、耐タンパデバイスの ID 情報を格納・参照のための複数サービス提供機関が共通的に利用できる仕組みをシステムとして利用することで、サービス提供機関が個別に携帯アプリを開発しなければならないという負担を減らすことが期待できる。また、サービス提供機関ごとに個別の携帯アプリを開発する方式では、サービスごとに利用者は携帯アプリをダウンロードする必要があるが、共通アプリであればダウンロードの手間は省ける。さらに、共通アプリを用いることによってユーザインタフェースなど統一化され、利用者の操作性を向上させることが期待できる。