IMES DISCUSSION PAPER SERIES

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—

井澤秀益（いざわひでみつ）・五味秀仁（ごみひでひと）

Discussion Paper No. 2016-J-3

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 2016-J-3
2016 年 2 月

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—

井澤秀益（いざわひでみつ）*・五味秀仁（ごみひでひと）**

要　旨

近年、生体認証を活用した次世代認証技術が注目を浴びている。その中でも FIDO（Fast IDentity Online）は、ネットワーク越しの認証に生体認証等を利用するための認証手順を定めた仕様であり、2015 年 12 月末現在で約 250 の団体が関わっており、一部のスマートフォンでは FIDO を活用したサービスがすでに提供されるなど、利活用が始まりつつある。

金融機関においては、海外において FIDO を利用したインターネット・バンキングを提供しているところもあり、今後、他の金融機関においても FIDO を活用する動きが出てくるものと予想される。もっとも、FIDO は 2014 年に策定された新しい仕様であるため、国内における詳細な資料や情報がまだ乏しい状況である。金融機関が FIDO をインターネット・バンキングに活用する際には、FIDO に関する正しい理解を行ったうえで、情報セキュリティの観点から安全性を評価し、自行において適用可能か否かを判断することが重要となる。

そこで本稿では、FIDO の仕組みについて解説を行ったうえで、FIDO をインターネット・バンキングに適用した場合を想定し、その安全性評価を実施し、金融機関が FIDO を導入する際の留意点の考察を行う。

キーワード：生体認証、FIDO、インターネット・バンキング、
安全性評価

JEL classification: L86、L96、Z00

*　日本銀行金融研究所企画役補佐（E-mail: hidemitsu.izawa@boj.or.jp）

**　ヤフー株式会社 Yahoo! JAPAN 研究所上席研究員（E-mail:hgomi@yahoo-corp.jp）

本稿の作成に当たっては、国立研究開発法人産業技術総合研究所主任研究員の大塚玲氏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、日本銀行あるいはヤフー株式会社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

インターネット上においてサービス提供者側（サーバ）が利用者（ユーザ）を認証する仕組みとして、ID とパスワードを利用した方式があるが、それには数々の問題点があると言われている。例えば、①パスワードを記憶しておき、それを入力するのに手間がかかるという利便性面での問題点や、②記憶できるパスワードには限界があるため、それを種々のサービスで使いまわすことにより生じうるパスワードリスト攻撃（IPA・ JPCERT[2014]）のリスクを内包しているという安全性面での問題点、が挙げられる。そこで、パスワードへの依存度を減らすために、生体認証を含む多様な認証手段をインターネット等のオープンなネットワーク上で活用する動きが出てきており、その一つとして、FIDO（Fast IDentity Online）がある。

FIDO は、ネットワーク越しの認証に生体認証等を利用するための認証手順（認証プロトコル）を定めた仕様であり、それを策定した FIDO Alliance は、その理念として、セキュリティ（安全性）と利用者の使いやすさ（利便性）を兼ね備えたものと位置づけている（FIDO Alliance [2014]）。2012 年に FIDO Alliance が発足して以降、同 Alliance に加入する団体は、E コマース運営業者、パソコンベンダ、スマートフォンベンダ、通信キャリア、ソフトウェアベンダ、金融機関等多岐にわたり、2015 年 12 月末現在で約 250 団体が加入している（FIDO Alliance [2015a]）。このため、ここで開発された方式がデファクト標準となる可能性が指摘されており（瀬戸 [2015]）、実際に NTT ドコモ社の一部のスマートフォンにおいては FIDO を活用した同社のサービスが利用可能（NTT ドコモ [2015]）である等、既に利活用が始まっている。

他方、金融機関におけるインターネット・バンキング等の各種リテールサービスにおいても、安全性と利便性の両方を考慮に入れたうえでシステムの構成を考えることが重要である。そのような中、FIDO を活用したインターネット・バンキングの仕組みはその解決策の一つとなる可能性があると考えられ、Bank of America では FIDO を活用したインターネット・バンキングを既に提供している（Bank of America[2015]）。もっとも、金融機関を巡るセキュリティ情勢は厳しさを増しており、インターネット・バンキングにおいては、マルウエアによる情報盗取や自動送金の手口などによる不正送金事例が数多く報告されているほか（大日向[2015]）、今後は、MitB 攻撃（Man-in-the-Browser 攻撃）のように、さらに巧妙なマルウエアによる攻撃が国内の金融機関で猛威をふるう可能性がある。そのような数々の脅威にさらされている中において、FIDO のように、新しい仕組みを金融機関が導入する際には、FIDO に関する正しい理解を行ったうえで、情報セキュリティの観点から安全性を評価することが重要となる。特に、上述のような様々な脅威に対して FIDO がどの程度耐性を持つのかについて検討を行うことは重要である。

そこで、本稿では上述の問題意識のもと、次世代認証技術の一つである FIDO の仕組みを整理し、金融機関が FIDO を導入する際の留意点について考察する。第 2 節で、前提知識となる FIDO の仕組みについて解説し、次に、第 3 節で、FIDO をインターネット・バンキングに適用した場合を想定し、その安全性評価を実施したうえで、金融機関が FIDO を導入する際の留意点の考察を行う。第 4 節は本稿のまとめである。

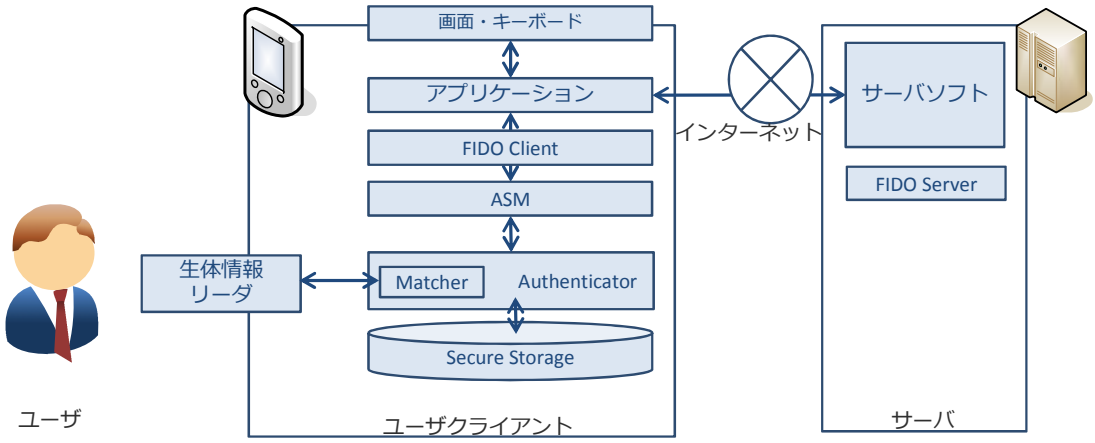
2. FIDO の仕組み

本節では、次世代認証技術として FIDO を取り上げ、その仕組みについて FIDO Alliance [2014]をもとに解説する。なお、本稿においては、FIDO Alliance が定める FIDO UAF（Universal Authentication Framework）ver 1.0 のことを FIDO と呼び、説明を行う。

(1) FIDO の概要

FIDO は、ネットワーク越しの認証に生体認証等を利用するための認証手順（認証プロトコル）を定めた仕様である。FIDO の主な特徴としては、①ユーザのプライバシーに配慮し、生体情報などの「認証に必要な秘匿すべき情報」をサーバに送信・登録しない点、②生体認証に限らず所有物認証や PIN 認証等、多様な認証要素の利用を想定している点、③様々な端末やサービスが存在する状況においても、統一的なプロトコル（FIDO）で認証を行える点、④「ユーザ認証」に限らず、ユーザの意思に基づいた取引であることをサーバで確認する「取引認証」の仕組みも想定している点、が挙げられる（①④の詳細は補論 1 を参照）。

本稿では、FIDO を活用した生体認証手段を備えたクライアント・サーバシステムの一例として、以下の要素から成るシステムを扱う（図表 1、図表 2）。



図表 1. FIDO を活用したクライアント・サーバシステムの一例

	要素名称	主な役割
	サーバ	
1	FIDO Server	Authenticator（後述）の登録を行うほか、ユーザからの検証結果を受け付け、ユーザ認証や取引認証を行うソフトウェア。
2	サーバソフト(*)	ユーザクライアントのアプリケーション（後述）に対してサービスを提供するソフトウェア。
	ユーザクライアント	
3	アプリケーション (*)	サービス提供者が作成する専用アプリケーションや WEB ブラウザ等の、ユーザからの指示を受け付けるためのソフトウェア。
4	FIDO Client	FIDO Server と通信するソフトウェア。
5	ASM (Authenticator Specific Module)	FIDO Client と Authenticator（後述）間で統一的なインターフェースを提供するソフトウェア。
6	Authenticator	生体認証などの所定の認証手段を用いてユーザの検証を実施するモジュール。ユーザの検証の後、必要な情報に対するデジタル署名を生成したり、登録フェーズ（後述）では、公開鍵と秘密鍵のペアを生成する。FIDO においてセキュリティ上の要となる要素。
7	生体情報リーダ(*)	ユーザからの生体情報の入力を受け付ける機器。
8	Matcher(*)	あらかじめ登録されたユーザの生体情報と、生体情報リーダによって取得された生体情報とが合致しているか照合するモジュール。通常は、Authenticator の中に内包されている。
9	Secure Storage(*)	Authenticator が生成したユーザの秘密鍵や、生体情報（テンプレート情報）を安全に保管するための領域。Authenticator に内包される場合もある。

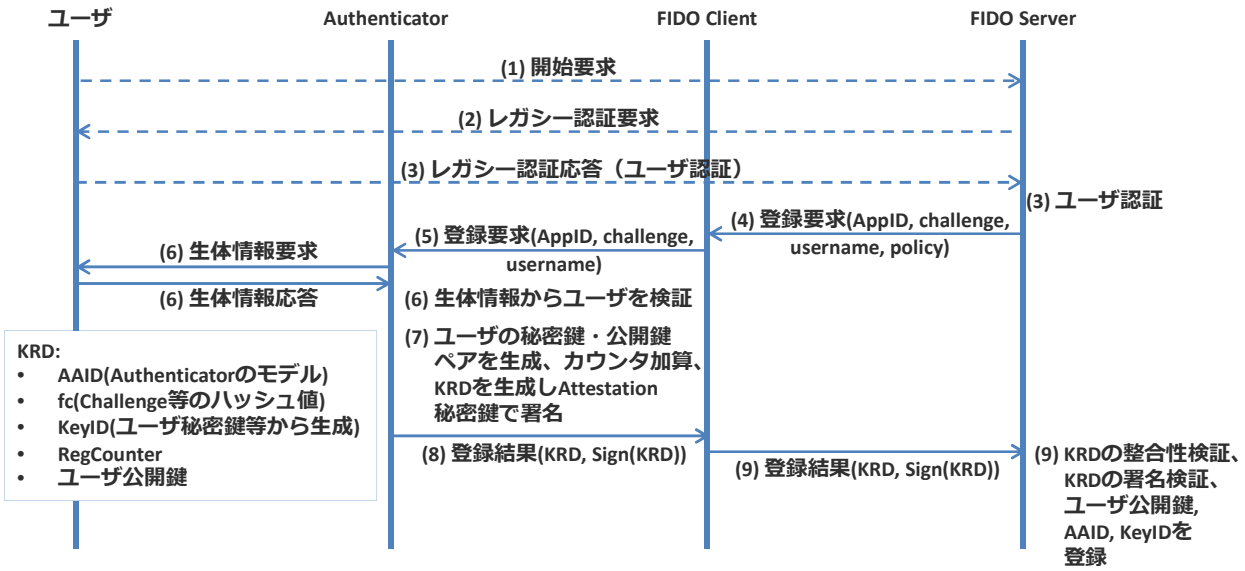
(*) FIDO において仕様が規定されていないものの、実際にシステムを実装する際には必要となる要素。

図表 2. FIDO における主な構成要素

(2) FIDO におけるフロー

FIDO において規定されている登録フェーズ（UAF Registration）と認証フェーズ（UAF Authentication）の説明を行う。登録フェーズは、端末毎・サービス毎に初回にユーザが実施する作業であり、FIDO Server がユーザクライアントの Authenticator の登録を行う。これによりレガシー認証情報（従前使用していた ID・パスワード等の個人を識別する認証情報。詳細は後述）と FIDO で使用するユーザ情報（後述）を紐付けることが可能となる。認証フェーズは、サービスの利用の度にユーザが実施する作業であり、FIDO Server がユーザ認証や取引認証を行う。

イ．登録フェーズにおけるフロー



図表 3. 登録フェーズにおけるフロー

登録フェーズにおけるフローの概要の一例は図表 3 の通り。ここでの前提として、現在は ID・パスワード等によるユーザ認証の仕組み（この認証を本稿では、FIDO における認証と区別するため「レガシー認証」と呼び、そのユーザ認証に必要となる認証情報を「レガシー認証情報」[1]と呼ぶ）が既に運用されている状態で、今次 FIDO を新たに導入するものとする。また、FIDO におけるユーザ認証の手段として生体認証を利用するものとする。なお、下記(1)～(3)は FIDO 仕様に規定されていないものであり（図中点線部分）、ここではその一例を示す。

(1) ユーザは、アプリケーションを通じて利用を希望するサービスのサーバ（FIDO Server）に接続し、登録フェーズの開始を要求する。

(2) FIDO Server は、ユーザに対してレガシー認証方式での認証を要求する。

(3) ユーザは、レガシー認証情報を FIDO Server に応答する。FIDO Server は、その情報をもとにユーザの認証を行い、認証に成功すれば次のステップに移る。

(4) FIDO Server は、FIDO Client に対して登録要求を行う。登録要求は、AppID[2]、

---

1 現在のシステムにおいて、ユーザを認証するために ID・パスワードを利用した方式を使用している場合には、その ID・パスワードがレガシー認証情報となる。本稿ではレガシー認証情報は、FIDO の登録フェーズにのみ用いるものとし、後述する認証フェーズでは用いないものとする。

2 サービスを識別するための ID。

challenge[3]、username、policy[4]等から成る。

(5) FIDO Client は、policy を解釈し、取引を続行可能であると判断すれば、Authenticator に対して登録要求を行う。

(6) Authenticator は、ユーザに対して生体情報を要求し、ユーザは登録フェーズを続行したければ自らの生体情報を生体情報リーダに提示し Authenticator に送信する。Authenticator は、受信した生体情報と、既に登録済の生体情報[5]とを Matcher にて比較することによりユーザを検証する。

(7) Authenticator は、上記(6)でユーザ検証に成功すれば、①後の認証フェーズで使用するためユーザの秘密鍵と公開鍵のペアを生成し、②必要なカウンタ値[6]の加算を行い、③Key Registration Data （KRD）を生成し、Authenticator が所有する登録用秘密鍵（Attestation 秘密鍵）[7]にてデジタル署名する。ユーザの秘密鍵は Secure Storage にて安全に保管される。KRD は、AAID[8]、fc[9]、KeyID[10]、RegCounter、ユーザの公開鍵等から成る。

(8) Authenticator は、デジタル署名付き KRD を登録結果として FIDO Client に返信する。

(9) FIDO Client は、デジタル署名付き KRD を登録結果として FIDO Server に返信する。FIDO Server は、①上記(4)で送信した登録要求の内容と受信した登録結果の内容との整合性を確認し、②KRD のデジタル署名を検証[11]し、③含まれるユーザの公開鍵、AAID、KeyID を当該ユーザのものとして登録する（当該ユーザの Authenticator を登録することを意味している）。これにより、FIDO Server は、AAID と KeyID のペアを FIDO におけるユーザ情報として管理し、上記(3)で受信したレガシー認証情報と紐付けて管理することができる。

---

3 FIDO Server が生成する乱数。

4 ユーザクライアント側で満たすべき能力や仕様に関してサーバ側で定めた情報。

5 Authenticator に生体情報を登録していない場合には、生体情報を登録するプロセス（当該プロセスは FIDO には未規定）に移行する。ここでは、ユーザクライアントのロック解除のためなどで生体情報を Authenticator に登録済のものとして話をすすめる。

6 FIDO では、登録フェーズを行う度毎に加算される RegCounter と、認証フェーズを行う度毎に加算される SignCounter の 2 種類がある。

7 Attestation 秘密鍵は、FIDO 認定を受けた Authenticator とともにユーザクライアントの工場出荷時に埋め込まれる等して配布され、Secure Storage にて安全に保管される。

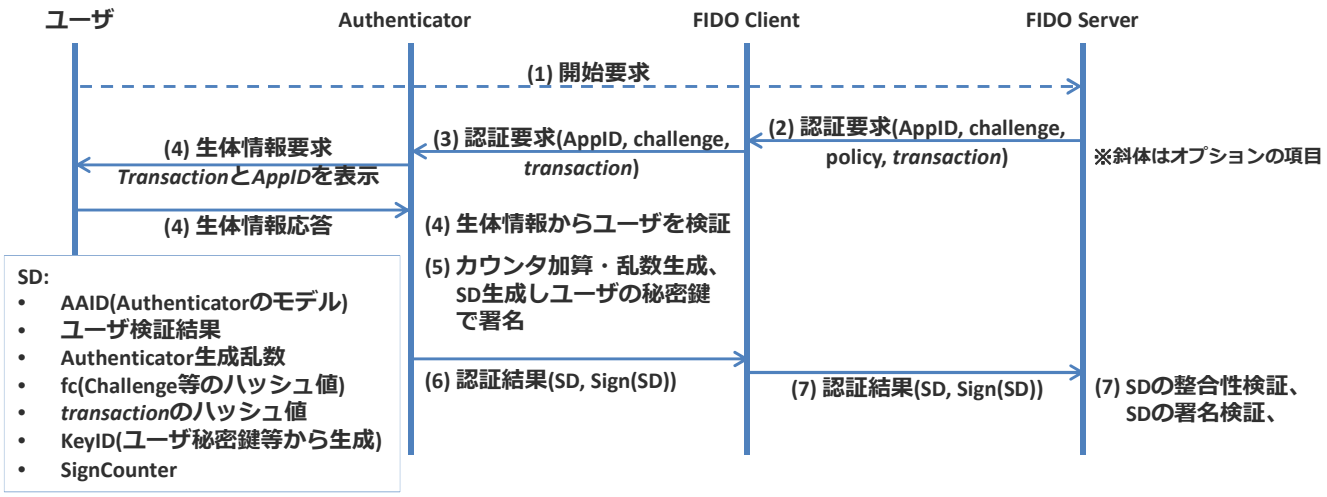
8 Authenticator のモデル毎に固有に付与される ID。

9 AppID や challenge などをまとめた値に対するハッシュ値。

10 ユーザの秘密鍵、appID、username 等のセットに対して固有に付与される ID。

11 デジタル署名を検証するための Attestation 公開鍵は、Metadata サービスと呼ばれる方法で FIDO Alliance から各 FIDO Server に配信される。

ロ．認証フェーズにおけるフロー



図表 4. 認証フェーズにおけるフロー

認証フェーズにおけるフローの概要の一例は図表 4 の通り。ここでの前提として、ユーザは既に登録フェーズを完了しているものとする。このため、レガシー認証情報は認証フェーズにおいては使用しない。なお、下記(1)は FIDO に規定されていないものであり（図中点線部分）、ここではその一例を示す。

(1) ユーザは、アプリケーションを通じて利用を希望するサービスのサーバ（FIDO Server）に接続し、認証フェーズの開始を要求する。この時に、取引内容（transaction[12]）も併せて送信する。

(2) FIDO Server は、FIDO Client に対して認証要求を行う。認証要求に際して、ユーザの認証に加えて、取引内容の確認を要求する場合（取引認証）には、取引内容（transaction）を含めることができる。その場合の認証要求は、AppID、challenge、policy、transaction 等から成る。

(3) FIDO Client は、policy を解釈し、取引を続行可能であると判断すれば、Authenticator に対して認証要求を行う。

(4) Authenticator は、ユーザに対して生体情報を要求し、ユーザは自らの生体情報を生体情報リーダに提示し Authenticator に送信する。ここで、取引内容（transaction）が通信に含まれている場合には、Authenticator は AppID と取引内容（transaction）を画面に表示し、ユーザはそれらを確認したうえで、取引内容（transaction）で取引を継続する意思を表示するため、自らの生体情報を

[12] インターネット・バンキングにおける振込み指図情報や、電子商取引における商品購入情報のこと。

提示する。Authenticator は、受信した生体情報と、既に登録済の生体情報とを Matcher にて比較することによりユーザを検証する。

(5) Authenticator は、上記(4)でユーザ検証に成功すれば、①必要なカウンタ値の加算や乱数を生成し、②Signed Data（SD）を生成し、ユーザの秘密鍵にてデジタル署名する。取引内容（transaction）が通信に含まれている場合の SD は、AAID、ユーザ検証結果、上記①で生成した乱数、fc、KeyID、SignCounter、transaction のハッシュ値等から成る。

(6) Authenticator は、デジタル署名付き SD を認証結果として FIDO Client に返信する。

(7) FIDO Client は、デジタル署名付き SD を認証結果として FIDO Server に返信する。FIDO Server は、①上記(2)で送信した認証要求の内容と受信した認証結果の内容との整合性を確認し、②SD のデジタル署名を検証する。それらに成功すれば、当該取引が正当なユーザからのものであると見なし、認証を完了する。また、取引内容（transaction）が通信に含まれている場合に①②が成功すれば、ユーザの正当性に加えて、取引内容（transaction）が改ざんされておらず、ユーザの意思に基づいたものとみなす。この取引内容（transaction）を上記(4)でユーザが確認し、その結果を(7)で FIDO Server が検証する取引認証の仕組みを「Transaction Confirmation」と FIDO では呼んでいる。

3. 金融機関が FIDO を導入する際の留意点

本節では金融機関が、FIDO を活用したインターネット・バンキングのサービスを提供する際の安全性評価を実施し、そのうえで、金融機関が FIDO を活用する際の留意点を考察する。

(1) 安全性評価の前提

FIDO を活用したインターネット・バンキングにおける安全性評価を実施するにあたり、防御側・攻撃側それぞれに前提をおいて検討する。防御側の前提とは、想定するインターネット・バンキングやそのセキュリティ対策、ユーザクライアント環境や取引フローに関する想定である。また、攻撃側の前提とは、不正送金を実施することを目的とした想定する攻撃者の、攻撃手法や能力に関する想定である。

イ. 防御側の前提

想定する、FIDO を利用したインターネット・バンキングの構成については、FIDO をインターネット・バンキングのサービスに適用している事例（Bank of America[2015]）を一部参考に、以下の前提を置く。

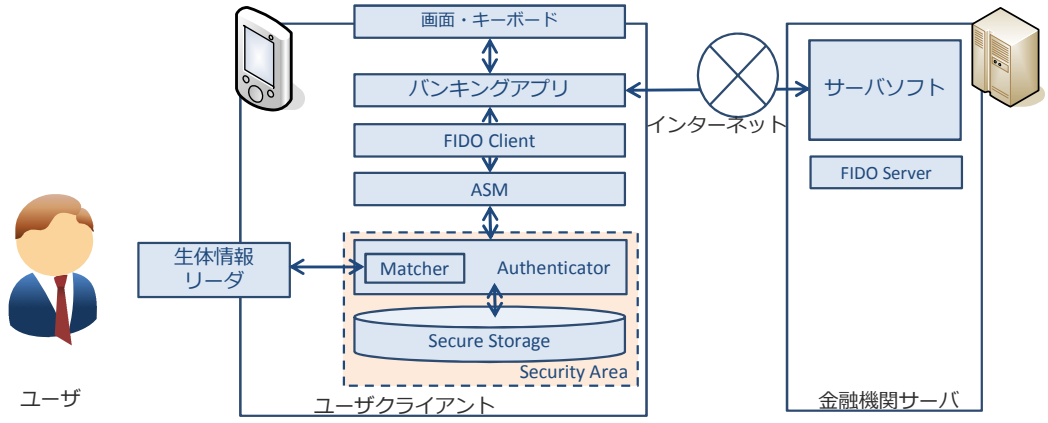
- ユーザは所有する Android スマートフォン[13]（以下、デバイスと呼ぶ）を利用し、金融機関が提供する Android 用アプリケーション（以下、バンキングアプリと呼ぶ）を使用することにより、インターネット回線を経由して金融機関サーバに接続することとする（図表 5）。
- ユーザは 1 種類のデバイスのみを FIDO のインターネット・バンキングの認証に使用する（例えば、スマートフォンと PC の両方を使用した、取引は行わない）こととする。
- 金融機関サーバは堅牢に構築されているため攻撃者からの攻撃を受けないこととする。

図表 5. 想定するインターネット・バンキングの構成

想定するユーザのデバイスについては、FIDO 対応デバイスを提供している事例（NTT DOCOMO [2015]）を一部参考に、以下の前提を置く。

[13] ここでは、後述する TEE の実装が進んでおり、今後インターネット・バンキングのプラットフォームとして普及が予想されるスマートフォンを想定することとし、PC を使った取引は本稿では取り扱わない。

- FIDO Alliance 認定を取得した Authenticator、ASM、FIDO Client がデバイスに既に導入されていることとする。Authenticator におけるユーザ認証の手段としては、生体認証（指紋認証もしくは虹彩認証）を利用することとする。
- ユーザの使用時以外はデバイスがロックされており、生体認証によるロック解除をしなければデバイスを使用できないこととする。またユーザ本人の生体情報のデバイスへの登録は済んでいることとする。
- デバイスはデバッグモードになっておらず、デバイスを PC に接続しても当該デバイスを操作できないこととする。
- Matcher、Authenticator、Secure Storage については、TEE（Trusted Execution Environment（GlobalPlatform [2015]））等の、通常環境とは隔離された領域（以降、Security Area と呼ぶ）に配置されており、同領域内においてはどのようなマルウエアでも活動できないこととする（図表 6）。
- デバイスは、後述する Trusted User Interface を備えていないこととする。



図表 6. 想定するアーキテクチャ

想定するインターネット・バンキングのフローについては、2. (2)の Transaction Confirmation を参考に以下の前提を置く。

（登録フェーズ）…デバイス毎・サービス毎（金融機関毎）に初回実施

Step1. ユーザは、生体情報を提示し、デバイスのロックの解除を行ったうえで、バンキングアプリを立ち上げる[14]。

Step2. ユーザは、レガシー認証情報[15]をバンキングアプリを通じて金融機関サー

[14] バンキングアプリについては、ログインに必要となる ID、パスワードの情報はアプリに保存してあるため、毎回ユーザが入力する必要が無いものとする。

[15] レガシー認証情報は、振込み時に必要となる認証情報（例：ワンタイムパスワード等）を想定し、当該認証情報はワンタイムパスワード生成器等の専用機器で生成されるものとし、専用

バに提示する。金融機関サーバは、登録要求をデバイスに送信する。

Step3. ユーザは、生体情報を生体情報リーダに提示する。Authenticator はユーザの検証を行い、検証に成功すれば登録結果（デジタル署名付き KRD）を金融機関サーバに送信する。金融機関サーバは、登録結果の内容を検証し、問題が無ければレガシー認証情報と FIDO でのユーザ情報（AAID および KeyID）とを紐づけて管理する。

（認証フェーズ）…登録フェーズ終了後、各種取引操作の都度実施

Step1. 登録フェーズの Step1. と同様。

Step2. ユーザは、バンキングアプリを通じて取引内容（transaction）である振込み先や振込み金額情報等を、金融機関サーバに送信する。

Step3. 金融機関サーバは、取引内容（transaction）を含めたうえで、認証要求をデバイスに送信する。ユーザは、画面に表示された取引内容（transaction）等を確認[16]する。

Step4. ユーザは、Step3 の結果、取引を継続したいと希望するときには自らの生体情報を生体情報リーダに提示する。Authenticator はユーザの検証を行い、検証に成功すれば認証結果（デジタル署名付き SD）を金融機関サーバに送信する。金融機関サーバは、認証結果の内容を検証し、問題なければ当該取引を実行する。

ロ．攻撃側の前提

攻撃者が不正送金を実現するため、ユーザのデバイスにアクセスする方法によって、以下の場合分けを行い、安全性評価を実施することとする。

攻撃者がデバイスへアクセスする方法
ケース A：物理アクセス（ネットワーク経由でのアクセスが困難であり、物理的なアクセスに限定される場合） 攻撃例：攻撃者が、ユーザのデバイスを盗取し、デバイスロックの解除を試み、デバイスの操作を行う場合。
ケース B：ネットワークアクセス（物理的なアクセスが困難であり、ネットワーク経由でのアクセスに限定される場合） 攻撃例：攻撃者がユーザのデバイスに対してマルウエアを感染させ、ネットワーク経由で攻撃を実施する場合。

図表 7．攻撃者のデバイスへのアクセス方法の違いによる場合分け

機器には攻撃者はアクセスできないものとする。

[16] 取引内容の確認においては、FIDO で規定されている Transaction Confirmation の仕組みを用いるものとする。取引内容確認メッセージとしては①利用金融機関名、②振込み先口座情報、③振込み金額情報、④振込み日時等が表示されるものとする。

また、想定する攻撃者による攻撃手法としてインターネット・バンキングにおける不正送金の 2 大手口（警察庁 [2013]）である「フィッシングによる手口」と「マルウエアによる手口」を想定するほか、デバイスへの物理アクセスが可能になった時にデバイスをロック解除されるといった「生体認証でのなりすまし」についても想定する。

（手口１：フィッシングによる手口・攻撃者能力の前提）

- 攻撃者は、サイトの見た目を金融機関に似せたサイト（以下、フィッシングサイトと呼ぶ）を構築し、当該サイトに接続してきたユーザから受信した情報を盗取できる[17]こととする。またその結果として攻撃者は、盗取した情報をなりすましの手段として使用することが可能であることとする。

（手口２：マルウエアによる手口・攻撃者能力の前提）

- 攻撃者は、金融機関サーバをマルウエア感染させることはできない一方で、ユーザのデバイスに対してマルウエア感染させることができる[18]こととする。

マルウエアの能力によって以下の場合分けを行う。

マルウエアの種類	マルウエアの能力
偽アプリ型	正規のバンキングアプリとは別のアプリとして独立して存在し[19]、サンドボックス[20]により、正規のバンキングアプリで扱うファイル・メモリや上述 Security Area にアクセスできないものの、最低限のパーミッション[21]はユーザにより許可されているものとする。
凶悪型	サンドボックス機構を回避し、正規のバンキングアプリが扱うファイル・メモリに読み書きができ、攻撃に必要なパーミッションを持つほか、任意のコードの実行が可能[22]。ただし、上述 Security Area に対しては当該能力が無いものとする。

[17] ユーザが、自らのデバイスを使ってフィッシングサイトにアクセスすることを想定するため、上述の「ケース B：ネットワークアクセス」を想定した攻撃手法である。また、フィッシングサイトに接続してきたユーザから「生体情報」を盗取する攻撃も考えられるが、正規の FIDO の仕組みでは生体情報をサーバに送信することは無く、当該攻撃が起こり得る可能性が高いのはマルウエアが介在する場合であるため、当該攻撃は（手口２）と分類する。

[18] 攻撃者がネットワーク経由で、ユーザのデバイスに対してマルウエア感染させることを想定するため、上述の「ケース B：ネットワークアクセス」を想定した攻撃手法となる。

[19] 正規のバンキングアプリ等を再パッケージ（Repackaging）し海賊版として配付するタイプのマルウエアを想定する（大居 [2013]）。

[20] Android 等において備わっている、アプリケーションやそれが扱うデータ同士を論理的に隔離するセキュリティ上の機能。

[21] アプリケーションが、デバイスの重要な機能群（カメラ、電話帳等）にアクセスを許可されること。

[22] 具体例としては、①マルウエアが root 権限を奪取する場合や、②正規バンキングアプリの開発環境が汚染されており、正規バンキングアプリに不正コードが仕込まれている場合、等が挙げられる。①については、root 権限を奪取する Android マルウエアの存在が報告されている（Zhou and Jiang [2012]）。また②については、実際に iPhone の開発環境である Xcode が改ざんされ、それが第三者により配付されたことにより、当該開発環境で作成されたアプリケーショ

図表 8. マルウエアの能力による場合分け

（手口３：生体認証でのなりすまし・攻撃者能力の前提）

- 攻撃者は、生体情報リーダに何らかの情報を提示することによってユーザになりすます攻撃[23]を実施できることとする。

攻撃側の前提のまとめとして、ケース A、B および、手口 1～3 の関係をまとめると下図のようになる。

	手口 1：フィッシング	手口 2：マルウエア	手口 3：生体認証でのなりすまし
ケース A：物理アクセス	想定しない	想定しない	想定する
ケース B：ネットワークアクセス	想定する	想定する	想定しない

図表 9. 攻撃前提におけるケース A、B および手口 1～3 の関係

ハ. 攻撃成功の定義

攻撃者による攻撃成功とは、不正送金を成功させることとし、フェーズ毎に分解すると、以下のいずれかの攻撃に成功することと定義する。

フェーズ	攻撃成功の定義
登録フェーズ	攻撃者が、自身が利用可能な生体情報（攻撃者自身の生体情報や、攻撃者が盗取したユーザの生体情報）を、ユーザのレガシー認証情報と紐づけて、攻撃者のアクセス可能なデバイスで登録に成功すること。
認証フェーズ	攻撃者が、ユーザの意思に反して攻撃者の口座に不正送金を実施すること。 ※なお、登録フェーズにおいて攻撃が成功すれば、攻撃者が利用可能な生体情報と攻撃者のデバイスを使って認証フェーズにて自由な振込みができるため、認証フェーズにおいても自動的に攻撃が成功することとなる。

図表 10. インターネット・バンキングのフロー毎の攻撃成功の定義

ンに不正なコードが埋め込まれた事例（XCodeGhost）が報告されている（Xiao [2015]）。

[23] 攻撃者がユーザになりすます方法の詳細は 3. (3)ニ. 参照。攻撃者が、ユーザのデバイスに対して、成りすますための何らかの情報を提示することを想定するため、上述の「ケース A：物理アクセス」を想定した攻撃方法となる。また、マルウエアによる生体情報盗取等は、マルウエアによる手口（手口２）と分類する。

(2) 安全性評価

上述の前提をもとに、FIDO を活用したインターネット・バンキングにおける安全性評価を実施する。

イ．登録フェーズにおける攻撃

登録フェーズにおいて、ケース毎および攻撃手口毎の攻撃の成否について評価すると図表 11 の通りとなる。なお、攻撃が成功する具体的なシナリオおよび分析手法については、補論 2 に記載する。また、攻撃成立する場合において、その対策手法に関する考察は、3. (3)にて述べる。

アクセス方法 / 攻撃の手口		ケース A：物理アクセス	ケース B：ネットワークアクセス
手口１：フィッシング		―	攻撃成立
手口２：マルウエア	偽アプリ型	―	攻撃成立
	凶悪型	―	攻撃成立
手口３：生体認証でのなりすまし		攻撃不成立	―

図表 11. ケース毎、手口毎の攻撃成否に関する評価（登録フェーズ）

ロ．認証フェーズにおける攻撃

認証フェーズにおいて、ケース毎および攻撃手口毎の攻撃の成否について評価すると図表 12 の通りとなる。なお、攻撃が成功する場合の具体的シナリオおよび分析手法については補論 2 に記載する。また、攻撃成立する場合において、その対策手法に関する考察は、3. (3)にて述べる。

アクセス方法 / 攻撃の手口		ケース A：物理アクセス	ケース B：ネットワークアクセス
手口１：フィッシング		―	攻撃不成立
手口２：マルウエア	偽アプリ型	―	攻撃不成立
	凶悪型	―	攻撃成立
手口３：生体認証でのなりすまし		攻撃成立	―

図表 12. ケース毎、手口毎の攻撃成否に関する評価（認証フェーズ）

(3) 金融機関の FIDO 導入にかかる留意点

上述した安全性評価（図表 11、図表 12）に関していくつか特徴的な点を抽出し、そこから金融機関が FIDO を導入する際の留意点を考察すると以下の通り。

イ．登録フェーズのレガシー認証情報について（ネットワークアクセス）

登録フェーズにおいては、レガシー認証情報が金融機関サーバに送られるが、これはリスクを伴う作業である。なぜなら、攻撃者がユーザのレガシー認証情報を盗取することが可能であれば、盗取した情報を使って攻撃者自身のデバイスで登録

フェーズを実施されてしまう結果、ユーザの口座は攻撃者のコントロール下になってしまうからである。

「ケースB：ネットワークアクセス（攻撃者がユーザのデバイスに対してネットワーク経由でのアクセスを行う場合）」において、攻撃者がレガシー認証情報を盗取する方法としては、①フィッシングサイトに誘導させ、ユーザを騙してレガシー認証情報を盗取する方法、②凶悪型マルウエアを通じて、ユーザが正規バンキングアプリに入力したレガシー認証情報を攻撃者が盗取する方法、③偽アプリ型マルウエアをユーザにインストールさせたうえで、当該アプリに対してレガシー認証情報を入力させ盗取する方法、の3つが考えられる。またその結果として、攻撃者が自身のデバイスを使用し、盗取したレガシー認証情報を使うことにより登録フェーズが不正に行われる。

このようなレガシー認証情報の盗取の多くは、FIDO自身に問題があるというよりは、FIDOに移行する前段階でレガシー認証方式を使っていることに起因する問題である。レガシー認証情報を盗取されないようにすることが、登録フェーズにおけるセキュリティ対策のポイントとなる。

このため、金融機関における留意点としては、以下の事項が考えられる。

(1) ユーザに対して、「レガシー認証情報を正規の金融機関サイト以外で入力しない」旨の注意喚起を行うこと（上述①対策）

(2) ユーザに対して、「(凶悪型マルウエア感染の原因であるOSの脆弱性に対処するため）デバイスのOSを常に最新の状態で使用する」旨の注意喚起を行うこと（上述②対策）

(3) ユーザに対して、「アプリのインストール時に、許可するパーミッションをよく吟味する」旨の注意喚起を行うこと（上述②対策）

(4) 作成する正規バンキングアプリに脆弱性が無いように十分なテストを実施することや、アプリ開発環境が信頼できることを確認すること（上述②対策）

(5) ユーザに対して、「偽アプリをインストールすることが無いように、正規アプリのインストール方法[24]を分かりやすく示す」こと（上述③対策）

(6) アプリストアに偽アプリが出現していないかどうか監視すること（上述③対策）

(7) レガシー認証情報を使用せずに登録フェーズを実施する方式を検討[25]すること

[24] 具体的には、バンキングアプリのインストール時には、①正規の金融機関のWEBサイトからのリンクを辿って公式アプリストアに行き、②公式アプリストアにおけるアプリケーション作成者情報を確認しインストールすることや、③公式アプリストア以外からアプリケーションをインストールしないような設定をデバイスにしておくこと、が挙げられる。

[25] 例えば次のような方法が考えられる。金融機関の窓口等で（行員による本人確認が行われた）

（上述①②③対策）

(8) 普段から、ユーザのデバイスを認証しておき、普段とは異なるデバイスからの登録フェーズは認めない（一定の利用実績のあるデバイスからのみ登録フェーズを認める）仕組みとすること

ロ．登録フェーズのレガシー認証情報について（物理アクセス）

「ケース A：物理アクセス（攻撃者がユーザのデバイスに対して物理的なアクセスを行う場合）」において、攻撃者がレガシー認証情報を盗取する方法は特に見当たらない。これは、本安全性評価の前提では、攻撃者がユーザのデバイスに物理的アクセスができたとしても、レガシー認証情報となるワンタイムパスワード等を生成するための専用機器（脚注 15 参照）にはアクセスできないという前提を置いたためである。前提とは異なるものの、仮に、レガシー認証情報となるワンタイムパスワード等がスマートフォンで生成・保存されるような作りの場合には、攻撃者は、生体認証でのなりすまし（手口 3）やパスコード入力によりデバイスのロックを解除すること等により、レガシー認証情報が盗取される可能性がある。

このため、金融機関における対策・留意点としては、以下の事項が考えられる。

(1) 登録フェーズに必要な、ワンタイムパスワード等のレガシー認証情報は、登録フェーズで使用するデバイス（スマートフォン等）で生成・保存する作りにしないこと

ハ．認証フェーズのマルウエア攻撃について

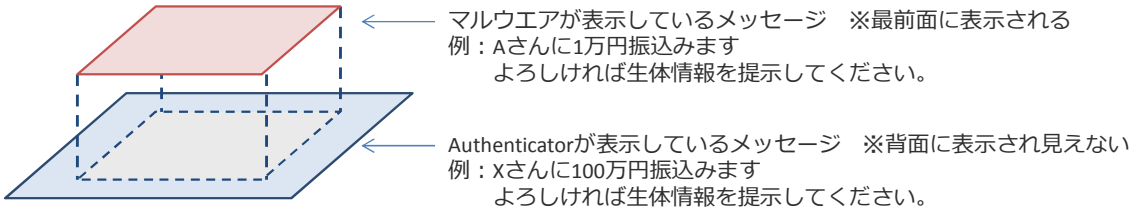
デバイス上のマルウエアは「ケース B：ネットワークアクセス（攻撃者がユーザのデバイスに対してネットワーク経由でのアクセスを行う場合）」において大きな脅威となりえる。認証フェーズにおいては、マルウエアにより、MitB 攻撃に類似した形でユーザの意図せざる取引が行われる可能性がある。その具体的な方法を見ていくために、攻撃者が Transaction Confirmation における攻撃を成功させるための段階の一例を整理すると、攻撃者は(A)取引内容（振込み先と金額情報）を改ざん（例：「A さんに 1 万円振込み」を「X さんに 100 万円振込み」と改ざん）して金融機関サーバに送信できること、(B)ユーザが確認する取引内容確認メッセージを改ざんできること（例：「X さんに 100 万円振込み」との表示を「A さんに 1 万円振込み」との表示に改ざん）、の 2 点が必要となる。

(A)に関しては、凶悪型マルウエアであれば、同マルウエアが正規のバンキング

ユーザが登録フェーズの一部（図表 3 の(1)および(4)～(9)）を実施し、FIDO Server において登録される「FIDO におけるユーザ情報」（AAID および KeyID のペアの値）をユーザが認識できるようにしておく。金融機関の窓口端末（安全な端末）にて「FIDO におけるユーザ情報」と「ユーザの口座情報」とを紐付ける作業を実施する。

アプリのメモリ情報を書き換えるなどして（FFRI [2012]）、ユーザが正規バンキングアプリに入力した取引内容を改ざんし、改ざんした内容を金融機関サーバへ送信させることが可能となる。

(B)に関しては、マルウエアによる「Display Overlay 攻撃[26]」（FIDO Alliance [2014]）により取引内容をユーザが正しく確認できなくなる可能性がある。「Display Overlay 攻撃」は（図表 13）のように、Authenticator が表示している取引内容確認メッセージの上に、マルウエアが表示したメッセージを上から覆い被せて[27]、ユーザにマルウエアが表示した取引内容確認メッセージを見せる攻撃である。こうすることによりユーザは背面の Authenticator が表示したメッセージ（例：X さんに 100 万円振込み）が見えなくなり、取引の改ざんに気付くことができなくなる。



図表 13.「Display Overlay 攻撃」が起こる仕組み

このため、金融機関における対策・留意点としては、イ．で述べた (2)～(6)と同様に、ユーザのデバイスのマルウエア対策を行うことのほか、以下で述べる対策が考えられる。

Display Overlay 攻撃への対策として、「マルウエア感染しても画面の表示内容が信頼できるものであることを確保する仕組みの導入[28]」が考えられる。これを実現する方法として 3 種類紹介する。1 つ目の方法としては、（3. (1)ロ．で述べた防御側前提とは異なるものの）取引内容確認を「別の専用デバイス」で実施するということが考えられる。「別の専用デバイス」がマルウエア感染しない限り、ユーザは取引内容を正しく確認することができる[29]。

---

[26] 本攻撃手法は、ウェブページ上でリンクやボタンなどの要素を偽装・隠ぺいしてクリックを誘う攻撃である「クリックジャック攻撃」（Huang *et al.*[2012]）の一つの応用例であり、より一般的には「UI Redressing Attacks」（Niemietz and Schwenk [2012]）と呼ばれている。

[27] Android における toast の仕組みや、WindowManager における優先度を悪用することにより、マルウエアが表示した通知メッセージを最前面に描画可能となる可能性がある（Richardson [2010]、Niemietz and Schwenk [2012]、Android Developers [2016a]）。

[28] より正確に言えば「ユーザが画面を通じて確認した取引内容（transaction）と Authenticator がデジタル署名した取引内容（transaction）とが同一であることを担保する仕組みを導入」したうえで、「ユーザが取引継続の意思を持ったときにのみデジタル署名が行われること」が必要と言える。

[29] ただしこの場合、取引の意思（取引続行もしくは取引中止）を「別の専用デバイス」に入力

2 つ目の方法としては、バンキングアプリの実装において、Display Overlay を検知する仕組みを導入することが考えられる。Android には OS の標準機能として、「ユーザが（画面上を）タッチした場所に対して、他のメッセージが上から覆いかぶさっていないかを判別する機能[30]」がある。この機能を利用して、バンキングアプリの取引内容確認メッセージにおいて、振込み先や振込み金額が記されている部分をユーザにタッチしてもらい、バンキングアプリにおいて、当該部分に他メッセージが覆いかぶさっていないかを確認するということが考えられる。もっとも凶悪型マルウエアによる攻撃の場合には、メッセージが覆いかぶさっているかを判定するフラグが使用するメモリ領域を、マルウエアにより書き換えられ、判定結果が改ざんされる可能性があり、完全な対策ではない点には留意が必要である。

3 つ目の方法として、デバイスの Trusted User Interface（以降、Trusted UI とよぶ）を活用するという方法がある（GlobalPlatform [2013]）。Trusted UI が備わったデバイスであれば、画面表示機能が Security Area の管理下におかれ、通常 OS 環境から隔離・保護されるため、マルウエアによる Display Overlay 攻撃の影響を受けないとされている（Coombs [2015]）。著者の調べる限り、一部のスマートフォンでは Trusted UI の装備が報じられている（Dyke [2015]）。なお FIDO 仕様では、金融機関サーバが、Authenticator において Trusted UI が使用可能かどうかを確認できる仕組みがある[31]。

このため、金融機関における留意点としては、以下の事項が考えられる。

(1) Trusted UI の今後の実装動向について留意すること

(2) 認証フェーズにおいて、ユーザの Authenticator が Trusted UI を利用可能でないと認識した場合には、その認証フェーズがリスクの高い取引になる可能性を認識すること

ニ．生体認証でのなりすましについて

攻撃者がユーザのデバイスを盗取した場合など、「ケース A：物理アクセス（攻撃者がユーザのデバイスに対して物理的なアクセスを行う場合）」においては、生体認証でのなりすましのリスクも意識する必要がある。生体認証として指紋認証を利用している場合を例にとると、次のような 3 種類の攻撃方法が考えられる。（イ）

し、当該情報を安全に金融機関サーバに送信する必要がある。

[30] 例えば、Android の MotionEvent の仕組みにおいて、FLAG_WINDOW_IS_OBSCURED というフラグが存在する（AndroidDevelopers [2016b]）。

[31] FIDO においては Metadata サービスを通じて、金融機関サーバが Authenticator 毎（AAID 毎）にそれぞれ Trusted UI が利用可能かを確認できる。また、各認証フェーズにおける通信において、ユーザクライアントから金融機関サーバに AAID がデジタル署名付きで送信される。金融機関サーバは、AAID をキーにして「当該取引で使用されている Authenticator が Trusted UI を利用可か否か」を確認することができる。

攻撃者が（システムが誤認することを期待して）自らの指紋を提示する方法、（ロ）攻撃者がユーザの指紋等を何らかの方法で盗取し、それに模した人工物を生体情報リーダに提示する方法[32]、（ハ）攻撃者がユーザの意図に反してユーザに指紋を提示させる方法[33]、などが考えられる。特に、認証フェーズにおいては、攻撃者に生体認証でのなりすましが行われてしまうと、端末のロック解除から取引内容の確認まで攻撃に必要なことが全て実施されてしまうため、大きな問題となりえる。

（ハ）については、技術的に解決することが容易ではないと考えられる。（イ）（ロ）に関しては、「ユーザクライアント側にて個人が検証され、その結果をサーバ側が信頼する」という FIDO のモデル固有の問題点と、「ユーザクライアントにおける生体認証でのなりすましの検知精度」の問題点に分けて考えることができる。

前者の、「FIDO のモデル固有の問題点」に対処すべく、FIDO においては補論 1. (1)で述べるように、ユーザクライアントの検証結果が改ざんされることへの対策や、Authenticator が不正なものに入れ替えられることへの対策を行っている。もっとも、補論 1. (1)の(iii)で述べた、「FIDO Alliance 認定を取得した Authenticator か否かを金融機関サーバにて確認する仕組み」については、FIDO Alliance は、FAR や APCER 等のセキュリティ評価指標[34]を検査しているわけではなく、基本的には FIDO のプロトコルに則ってサーバ・クライアント間で通信ができるかを確認しているに過ぎない点には留意が必要である。

後者の、「ユーザクライアントにおける生体認証でのなりすましの検知精度」の問題点に対処すべく、FIDO においては、金融機関サーバが Metadata サービスを通じて、Authenticator 毎の一部のセキュリティ評価指標（FAR 等）を確認することが可能である。もっとも、それらの評価指標にかかる情報は、基本的には Authenticator ベンダの自己評価値であり、FIDO Alliance が評価したわけではない点には留意が必要である。このため、コモンクライテリア（ISO/IEC 15408）等により第三者評価された Authenticator を使用することや、第三者評価されたセキュリティ評価指標を金融機関が把握することが重要となる。

一般に、生体認証センサに何らかの情報を提示してなりすましを試みる攻撃は、プレゼンテーション攻撃と呼ばれ、その評価については ISO/IEC 30107 において標準化の作業が行われている（新崎[2015]、宇根[2016]）。また、生体認証システムの

---

[32] 実際に当該方法で iPhone の生体認証の突破に成功したと主張している団体がある（Chaos Computer Club [2013]）

[33] 例えば、ユーザが眠っている間に、攻撃者がユーザの指をユーザのデバイスの生体情報リーダに押し付ける方法が考えられる。

[34] FAR（False Acceptance Rate）は「他人を本人」と誤って判定する確率（ISO/IEC 19795-1）。APCER（Attack Presentation Classification Error Rate）は人工物が提示された際に「人間の身体の一部が提示された」と誤って判定する確率（ISO/IEC 30107-1）。

セキュリティ評価に必要なセキュリティ機能要件等を規定する国際標準案（ISO/IEC 19989）が審議されている（山田[2015]、宇根[2016]）。今後、Authenticator にかかる第三者評価が活用されることにより、金融機関が「ユーザクライアントにおける生体認証でのなりすましの検知精度の問題点」に対処可能となる可能性がある。

このため、金融機関における対策・留意点としては、以下の事項が考えられる。

(1) FIDO Alliance による Authenticator の認定はセキュリティ評価指標への認定ではないことを認識すること

(2) Authenticator がコモンクライテリア（ISO/IEC 15408）等の第三者評価を受けたものか調査しておくこと

(3) 第三者によって評価されたものである場合には、Authenticator のセキュリティ評価指標を[35]当該取引のリスクと紐づけることが可能となることを認識すること（例えば、FAR が高い Authenticator からの取引は、FAR が低いものからの取引よりもリスクが高いなど）

(4) デバイスの盗難の届け出がユーザからあった場合には直ちに当該デバイスの取引を中止すること

(5) ユーザの意図に反して生体情報が提示されることを想定し、振込み先情報、振込み時刻等を利用した、振込み操作の「異常検知技術」を導入すること

ホ．生体情報のリプレイ攻撃リスクについて

凶悪型マルウエアに感染したデバイスにおいては、生体情報のリプレイ攻撃リスクが存在する。提示されたユーザの生体情報をマルウエアが、生体情報リーダと Matcher 間（補論 2 の図表 14 の④）において盗取し、それを再送（リプレイ）することにより、マルウエアが生体認証を突破できる可能性があり、ユーザの意思に反して生体認証や、その後続手続きである取引承認のデジタル署名が行われてしまう可能性がある。本インターネット・バンキングのシステムにおける盗取に限らず、他のシステムにおいて生体情報を盗取されたとしても、原理的には本インターネット・バンキングにおいて攻撃が成立する。これら攻撃は、生体情報リーダおよび当該リーダと Matcher との通信経路が Security Area 内に無いことに起因して起こるものである。

---

[35] FIDO においては、登録フェーズ、認証フェーズともに、ユーザクライアントから金融機関サーバに対して AAID がデジタル署名付きで送信される。金融機関は、AAID を確認することにより当該通信における Authenticator のベンダ名とモデル名が確認できる。本文(2)のように金融機関が Authenticator のセキュリティ評価指標をあらかじめ調査しておけば、AAID をキーにして「当該取引で使用されている Authenticator のセキュリティ評価指標」を確認することができる。

このため、金融機関における対策・留意点としては、イ．で述べた凶悪型のマルウエア対策に加えて、以下の事項が考えられる。

(1) ユーザのデバイス（Authenticator）における内部構造を調査し、生体情報リーダ自身および、当該リーダと Matcher の通信経路が Security Area 内にあるか無いかを調査[36]しておくこと

(2) 仮に Security Area 内にない場合には、当該取引はリスクが高くなる可能性を認識しておくこと

へ．振込み限度額のリスク管理について

以上のように、FIDO の仕組みにおいては、生体認証でのなりすましの脅威があるほか、従来のインターネット・バンキングと同様に、マルウエアの脅威が大きいものとなる。マルウエアや生体認証でのなりすましへの留意点は前述の通りであるが、これらのリスクをゼロにすることは難しいと考えられ、万が一攻撃を受けたときの被害を軽減する策や、取引の異常を検知する技術も重要になると考えられる。

このため、金融機関における対策・留意点としては、以下の事項が考えられる。

(1) ユーザが高リスクサービス（例：新規振込み先への振込み操作）を利用する場合には、従来のインターネット・バンキングと同様に、振込み限度額を設定しておくこと

(2) 振込み先情報、振込み金額等を利用した、振込み操作の「異常検知技術」を導入すること

(3) 当該「異常検知技術」を導入する際のリスク値を算出する際に、a)デバイスの種類（OS の種類）によるマルウエア感染リスクの違いや、b)第三者によって評価された Authenticator のセキュリティレベル評価指標（FAR や APCER 等）や、c)デバイスの生体情報リーダ自身および当該リーダと Matcher の通信経路が Security Area 内にあるか否かの情報、d)デバイスで Trusted UI が使用可か否かの情報を利用することができる点に留意すること

[36] なお、FIDO の Metadata サービスにより、各 Authenticator の Matcher が Security Area 内に存在するか否かを金融機関は確認することができる。ただし Metadata サービスでは、生体情報リーダおよび当該リーダと Matcher の通信経路に関して Security Area 内外を確認する方法は無い。

4. まとめ

本稿では、まず FIDO における登録フェーズおよび認証フェーズにおけるフローの解説を行った。次に、FIDO における Transaction Confirmation の仕組みを、スマートフォンを使ったインターネット・バンキングに適用した場合において、攻撃側・防御側の前提条件を置いたうえで、安全性評価を実施した。攻撃者の手口として 3 種類（フィッシング、マルウエア、生体認証でのなりすまし）を想定し、攻撃者がユーザのデバイスに物理的にアクセスする場合とネットワークアクセスする場合を分け、それぞれの場合で攻撃の可否を評価した（図表 11、図表 12）。

本稿では、そうした評価を踏まえ、金融機関が FIDO を導入する際の留意点について考察を実施した。特に重要な点を要約すると、①登録フェーズにおけるレガシー認証情報が盗取されることについて十分注意すること、②従来のインターネット・バンキングと同様に、マルウエアの脅威に対しては引き続き注意する必要があること、③生体認証でのなりすましも脅威となりえること、④このため第三者により評価された Authenticator のセキュリティレベル評価指標等を有効に活用すること、⑤デバイスのセキュリティ対策（Trusted UI や、生体情報リーダが Security Area 内にあるか否か等）に関する情報の収集を行い、「異常検知技術」に活用すること、が挙げられる。

今後、上述⑤については、Trusted UI 搭載デバイスの普及や、上述④については、Authenticator の第三者評価認定の取得が求められよう。ただ、このような課題があるものの、FIDO は EC サイトにおける決済やインターネット上の各種サービスにおけるログインに広く活用されることを想定して策定されているものであり、FIDO Alliance の加盟メンバー数が年々増加している状況に鑑みると、そうした課題の解決とともに、FIDO におけるエコシステムが、ネットワーク越しの生体認証等のデファクト標準になる可能性がある。現在はその移行期にあるとも考えられる。

インターネット・バンキングにおける不正事件の手口は日々巧妙化している。また、FIDO に関しても、FIDO 2.0 の策定に向けた動きがあるが（FIDO Alliance [2015b]、近藤[2015]）、最新の方式だけにその動向には留意が必要である。このように、インターネット・バンキングの不正送金事例に関する国内外の動向や学界の動向に注視しつつ、次世代認証技術の一つである FIDO の動向および、その実用化動向に注意しながら、金融機関におけるインターネット・バンキングの将来像を考えることが今後重要になると考えられる。

参考文献

宇根正志、「生体認証システムにおける人工物を用いた攻撃に対するセキュリティ評価手法の確立に向けて」、IMES Discussion Paper Series、2016 年

大居司、「Android プラットフォームの基本的なセキュリティ機構」、『Android セキュリティ・バイブル 2013』、日経エレクトロニクス/日経コミュニケーション編、2013 年

大日向隆之、「オンラインバンキング不正送金の手口と対策」、日本銀行金融研究所第 16 回情報セキュリティ・シンポジウム　講演資料、2015 年（http://www.imes.boj.or.jp/citecs/symp/16/ref4_oohinata.pdf）

株式会社 FFRI、「Man in the Browser in Android の可能性」、FFRI Monthly Research 2012 年 12 月、（http://www.ffri.jp/research/monthly_research.htm）

警察庁、「不正送金及び不正アクセス等の被害について」、2013 年 12 月 17 日（https://www.antiphishing.jp/news/pdf/apcseminar2013npa.pdf）

近藤裕介、「次世代認証プロトコル FIDO の動向」、Yahoo! JAPAN Tech Blog、2015 年 12 月 17 日（http://techblog.yahoo.co.jp/security/fido-introduction/）

新崎卓、「SC37 Biometrics 標準化報告 WG3 Biometric Data interchange Formats」、JAISA バイオ関係標準化セミナー発表資料、2015 年 10 月

瀬戸洋一、「ビッグデータ時代のバイオメトリクスにおけるプライバシー保護」、『最新自動認識技術 2015（月間自動認識 2015 年 9 月増刊号）』、日本工業出版、2015 年

タオソフトウェア株式会社、『Android Security 安全なアプリケーションを作成するために』、株式会社インプレスジャパン、2012 年

竹森敬祐、「スマートフォンのセキュリティ」、第 13 回日本銀行情報セキュリティ・シンポジウム講演資料、2011 年 10 月 28 日（http://www.imes.boj.or.jp/citecs/symp/13/ref2_takemori.pdf）

独立行政法人情報処理推進機構（IPA）・一般社団法人 JPCERT コーディネーションセンター（JPCERT）、「STOP!! パスワード使い回し!!パスワードリスト攻撃による不正ログイン防止に向けた呼びかけ」、IPA・JPCERT、2014 年 9 月 17 日（https://www.jpcert.or.jp/pr/2014/pr140004.html）

山田朝彦、「SC27（情報セキュリティ）におけるバイオメトリクス関係プロジェクト」、JAISA バイオ関係標準化セミナー、2015 年 10 月

NTT ドコモ、「『FIDO Alliance』に加入―生体情報を使った新しいオンライン認証を提供開始―」、NTT ドコモ報道発表資料、2015 年 5 月 26 日

（https://www.nttdocomo.co.jp/info/news_release/2015/05/26_00.html）

Android Developers, “WindowManager.LayoutParams,” Android Developers, 2016a （ http://developer.android.com/reference/android/view/WindowManager.LayoutParams.html）

Android Developers, “MotionEvent,” Android Developers, 2016b （http://developer.android.com/reference/android/view/MotionEvent.html）

Bank of America, “Bank of America Introduces Fingerprint and Touch ID Sign-in for Its Mobile Banking App,” Bank of America Newsroom, September 15, 2015 （ http://newsroom.bankofamerica.com/press-releases/consumer-banking/bank-america-introduces-fingerprint-and-touch-id-sign-its-mobile-ban）

Rob Coombs, “Securing the Future of Authentication with ARM TrustZone-based Trusted Execution Environment and Fast Identity Online (FIDO),” ARM White Paper, 2015

Chaos Computer Club, “Chaos Computer Club breaks Apple TouchID,” September 21, 2013（https://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid）

Rob Dyke, “The Benefits of Trusted User Interface,” Trustonic Blog, September 2, 2015 （https://www.trustonic.com/news-events/blog/benefits-trusted-user-interface）

FIDO Alliance, “FIDO UAF Complete Specifications,” FIDO Alliance, December 9, 2014 (https://fidoalliance.org/specifications/download/)

FIDO Alliance, “Members: Bringing together an ecosystem,” FIDO Alliance, 2015a (https://fidoalliance.org/membership/members/)

FIDO Alliance, “FIDO Authentication Poised for Continued Growth as Alliance Submits FIDO 2.0 Web API to W3C,” FIDO Alliance, November 19, 2015b (https://fidoalliance.org/fido-alliance-announces-fido-authentication-poised-for-continued-growth-as-alliance-submits-fido-2-0-web-api-to-w3c/)

GlobalPlatform, “GlobalPlatform Device Technology Trusted User Interface API Version1.0,” GlobalPlatform Device Specifications, 2013 （http://www.globalplatform.org/specificationsdevice.asp）

GlobalPlatform, “The Trusted Execution Environment: Delivering Enhanced Security at a Lower Cost to the Mobile Market ,” GlobalPlatform White Paper, 2015

Lin-Shung Huang, Alex Moshchuk, Helen J.Wang, Stuart Schechter, and Collin Jackson, “Clickjacking: Attacks and Defenses,” 21st USENIX Security Symposium, 2012

Marcus Niemietz and Jorg Schwenk, “UI Redressing Attacks on Android Devices,” Blackhat ABU DHABI 2012, 2012

NTT DOCOMO, INC. “FIDO Alliance Seminar in D.C. Case Study: NTT DOCOMO,” FIDO Seminar in D.C. October 5, 2015
（https://fidoalliance.org/wp-content/uploads/NTT_DOCOMO_case_study_FIDO-Seminar-DC_10_05_15.pdf）

N.K. Ratha, J.H. Connell and R.M.Bolle, “Enhancing security and privacy in biometrics-based authentication systems,” IBM systems journal, Vol40, No 3, 2001 pp.614-634

David Richardson, “Tapjacking,” Lookout Mobile Security Blog, 2010
（https://blog.lookout.com/look-10-007-tapjacking）

Claud Xiao, “Novel Malware XcodeGhost Modifies Xcode, Infects Apple iOS Apps and Hits App Store,” paloalto networks blog, September 17, 2015
（ http://researchcenter.paloaltonetworks.com/2015/09/novel-malware-xcodeghost-modifies-xcode-infects-apple-ios-apps-and-hits-app-store/）

Yajin Zhou and Xuxian Jiang, “Dissecting Android Malware: Characterization and Evolution,” IEEE Symposium on Security and Privacy, 2012, pp.95-109

補論 1. FIDO における特徴

ここでは、FIDO における以下の 2 点の特徴について詳しく説明する。

- 利用者のプライバシーに配慮し、生体情報などの「認証に必要な秘匿すべき情報」をサーバに送信・登録しない点
- 「ユーザ認証」に限らず、ユーザの意思に基づいた取引であることをサーバで確認する「取引認証」の仕組みも想定している点

(1) 生体情報などをサーバに送信・登録しない点

FIDO においては、生体情報をサーバに送信・登録せず、「ユーザクライアント側にて個人が検証され、その結果をサーバ側が信頼する」というモデルとなっている。このため、①ユーザクライアントにおける検証結果が攻撃者に改ざんされるリスクや②Authenticator が不正なものに入れ替えられ検証結果が信頼できなくなるリスクがある。

①に関しては、(A)Matcher において生成されるユーザの検証結果情報を改ざんする方法、(B)Authenticator にて生成される SD（2. (2)ロ. 参照）の情報を攻撃者が生成する方法が考えられる。

これらに対して、FIDO では、(i)Authenticator が（攻撃者が容易に介入できない）安全な領域に設置されることを想定したうえで、Authenticator の中にある Matcher にてユーザを検証し、検証に成功したときのみ Authenticator にて SD に対するデジタル署名を付すことが規定されている（上述(A)対策）。また、FIDO では、(ii)ユーザの秘密鍵は Authenticator と同じく安全な Secure Storage にて格納することが想定されているため、攻撃者は、自らが生成した SD に対してデジタル署名を付すことは困難である（上述(B)対策）。なお FIDO においては安全な領域の実装方法の例として、TEE（Trusted Execution Environment（GlobalPlatform [2015]））や Secure Element[37] を例示している。

②に関して、攻撃者が Authenticator を偽物（生体認証でのなりすましの検知精度の悪いもの）に入れ替えてしまい、偽 Authenticator が使用され、検証結果が信頼できなくなるという攻撃方法が考えられる。

これに対して、FIDO では、(iii)「ユーザデバイス側で使用される Authenticator が FIDO Alliance における認定取得製品か否か」ということをサーバ側で確認でき

[37] 外部からの解析に耐えるように設計され、安全にデータを格納し、処理するための演算処理機能を持ったハードウェアの総称。

る仕組みがある。具体的には、登録フェーズにおいて、KRD（2．(2)イ．参照）に対して Authenticator の Attestation 秘密鍵でデジタル署名することにより、FIDO Alliance における認定取得製品であることをサーバ側で確認できる（認定取得製品でなければ Attestation 秘密鍵を所有していないため）。もっとも、FIDO Alliance における認定取得製品は、FAR や APCER 等のセキュリティ評価指標を検査しているわけではなく、FIDO のプロトコルに準拠しているかの検査を行っているだけである点には留意が必要である。

(2) 取引認証の仕組みについて

FIDO においては Transaction Confirmation の仕組みがオプションとして規定されており、取引内容（transaction）に対して、ユーザが取引継続の意思表示を行い、その結果をサーバ側で確認する仕組みがある。

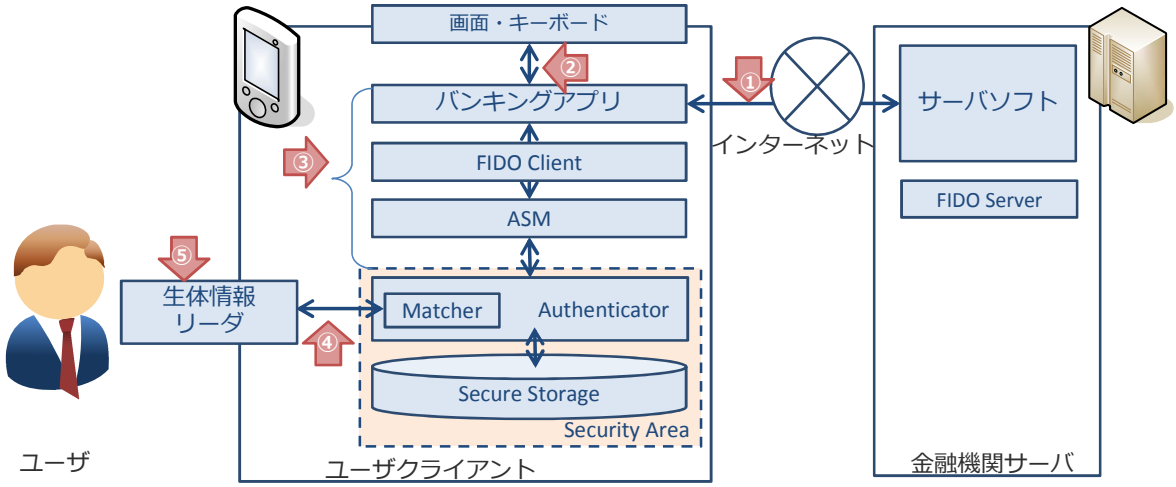
図表 4 において、(1)にてユーザは実行したい取引内容（transaction）をサーバ側に送信し、(4)にてユーザはこれから行われようとしている取引内容（transaction）について画面を通じて確認する。ユーザが取引を続行する意思があれば、自らの生体情報を生体情報リーダに提示して取引を続行する意思表示をする。その後、(5)にて Authenticator は、ユーザ検証に成功した場合に、取引内容（transaction）のハッシュ値をはじめとする情報（SD）に対してデジタル署名を行い、(6)および(7)にて当該情報がサーバ側に送信される。

この Transaction Confirmation の仕組みにより、仮にユーザクライアントがマルウエアに感染しているような状況下にあり、マルウエアによって、(1)で取引内容を改ざんされたり、自動的に攻撃者への送金指図を出されたりする等が行われた場合においても、「ユーザが画面を通じて確認した取引内容（transaction）と、Authenticator がデジタル署名した取引内容（transaction）とが同一であること」と「ユーザが取引継続の意思を持ったときのみにデジタル署名が行われること」の条件が担保できれば、マルウエアによる不正送金は阻止できる（ユーザは不正送金が行われる前に気づくことができる）。もっとも、当該前提条件をマルウエアにより崩されれば攻撃者による不正送金が成功することになり、3．(3)ハ．においてその手法および対策を紹介する。

補論 2. 具体的な攻撃のシナリオ

図表 11 および図表 12 に示した内容において具体的な分析手法および攻撃シナリオ例を以下の通り示す。

具体的な分析手法としては、登録フェーズおよび認証フェーズのそれぞれの Step 毎（3. (1)イ. 参照）に、ケースに応じて想定される攻撃を洗い出したうえで、その具体的な攻撃手口および攻撃箇所を検討する。そして、Step を通じて考えたときに、最終的に 3. (1)ハ. で示した攻撃成功につながるか否かを検証する。なお、攻撃が行える箇所は、Ratha, Connell, and Bolle [2001]を参考に図表 14 の通り整理する。



箇所	攻撃者による攻撃の内容の例
①	ユーザクライアントと金融機関サーバ間の通信を改ざん・盗聴する。 フィッシングサイトを設置し、同通信を改ざん・盗聴する。
②	ユーザに見える表示内容を改ざん・盗聴する。 ユーザが文字入力する内容を改ざん・盗聴する。
③	バンキングアプリ・FIDO Client・ASM の各コンポーネントがやり取りする情報を、改ざん・盗聴する。偽のコンポーネントを利用する。
④	生体情報リーダと Matcher 間のデータを改ざん・盗聴する。
⑤	生体情報リーダに対して、ユーザになりすますための生体情報を提示する。

図表 14. 攻撃が行える可能性がある箇所

(1) 登録フェーズにおける攻撃について

登録フェーズにおける Step 毎（3. (1)イ. 参照）に、想定される攻撃の内容、ケース毎の攻撃の手口、図表 14 の攻撃可能箇所を整理すると、図表 15 の通りとなる。

Step	各 Step における通常時のユーザのフロー	各 Step で想定される攻撃の内容(攻撃者が実施する行動)	「ケース A：物理アクセス」において攻撃を成立させるために必要な手口	「ケース B：ネットワークアクセス」において攻撃を成立させるために必要な手口 ※括弧内の数字は図表 14 の攻撃可能箇所を示す	
Step1	デバイスロックを解除	デバイスロックを不正に解除	不要（注 1）	不要（注 1）	
Step2	レガシー認証情報を入力・送信	レガシー認証情報を盗取	有効な方法無し（注 2）	手口１（フィッシング）によってレガシー認証情報を盗取（①）	手口２（マルウエア）によってレガシー認証情報を盗取（②③）
Step3	生体情報を提示（ユーザ検証）し、登録要求を継続	ユーザ検証を不正に実施	不要（注 1）	不要（注 1）	

図表 15. Step 毎の攻撃手口の整理（登録フェーズ）

（注 1）登録フェーズにおいて攻撃を成立させるために最低限必要な事項としては、「攻撃者が Step2 においてレガシー認証情報を盗取する」ということになる。これができれば、攻撃者は自身のデバイスで登録フェーズを実施でき、攻撃が成功する。本安全性評価では、脚注 15 で述べたとおり、「レガシー認証情報は攻撃者がアクセスできない専用機器で生成する」という前提であるため、レガシー認証情報を盗取するために Step1 や Step3 において攻撃を行う必要は無い。

（注 2）「ケース A：物理アクセス」の場合、利用可能な手口が「手口３（生体認証でのなりすまし）」のみという前提を置いているが（図表 9）、この手口を使ってレガシー認証情報を盗取する有効な方法は見当たらない。ただし攻撃者が、手口３を使ってデバイスのロックを不正に解除し、デバイスにマルウエアを仕込んだうえで、当該デバイスを正規ユーザに返却することにより、ケース B の手口２に帰着させることも考えられる。本稿では、そのような場合はケース B にて論じている。

図表 15 をもとに、ケース毎、攻撃手口毎の攻撃成否を整理すると図表 16 の通りとなる（本文中の図表 11 と同内容）。図表内の【】は、攻撃成立の場合の図表 15 で示した Step および攻撃箇所を示す。なお、図表中の「ー」で示した部分は、図表 9 で示した「想定しない」攻撃手口を指す。

攻撃の手口 ＼ アクセス方法		ケース A： 物理アクセス	ケース B： ネットワークアクセス
手口１：フィッシング		―	攻撃成立（イ） 【Step2①】
手口２： マルウエア	偽アプリ型	―	攻撃成立（ロ） 【Step2②】【Step2③】
	凶悪型	―	攻撃成立（ハ） 【Step2②】
手口３：生体認証でのなりすまし		攻撃不成立	―

図表 16. ケース毎、手口毎の攻撃成否に関する評価（登録フェーズ）再掲

以下に、具体的な攻撃シナリオの例を見ていく。図表 15 で示した Step および攻撃箇所は、【】内に示す。

イ．手口１：フィッシング・ケース B：ネットワークアクセス　の場合

- 攻撃者が、ユーザをフィッシングサイトに誘導し、ユーザに対して言葉巧みに「レガシー認証情報を入力して下さい」とフィッシングサイトから指示を出す。ユーザが、その指示に従えば、攻撃者は（ユーザの）レガシー認証情報と、自らの生体情報を使って、自らのデバイスで登録フェーズを実施でき、攻撃が成功する。【Step2①】

ロ．手口２：偽アプリ型マルウエア・ケース B：ネットワークアクセス　の場合

- 攻撃者は、銀行口座の管理を便利にするためのアプリケーションと称して、偽アプリ型マルウエアを作成する。ユーザが当該マルウエアを立ち上げ、マルウエアがユーザに対して「銀行口座の管理を便利に行うためレガシー認証情報を入力して下さい」と指示を出す。ユーザがその指示に従えば、マルウエアは攻撃者に当該情報を送信する。攻撃者は（ユーザの）レガシー認証情報と、自らの生体情報を使って、自らのデバイスで登録フェーズを実施でき、攻撃が成功する。【Step2②】【Step2③】

ハ．手口２：凶悪型マルウエア・ケース B：ネットワークアクセス　の場合

- ユーザがレガシー認証情報を正規のバンキングアプリに入力し、凶悪型マルウエアがキーロガー等（竹森 [2011]、タオソフトウェア [2012]）を使用し当該情報を盗取し攻撃者に送信する。攻撃者は（ユーザの）レガシー認証情報と、自らの生体情報を使って、自らのデバイスで登録フェーズを実施でき、攻撃が成功する。【Step2②】

(2) 認証フェーズにおける攻撃について

認証フェーズにおける Step 毎（3. (1)イ．参照）に想定される攻撃の内容、ケース毎の攻撃の手口、図表 14 の攻撃可能箇所を整理すると、図表 17 の通りとなる。

Step	各 Step における通常時のユーザのフロー	各 Step で想定される攻撃の内容（攻撃者が実施する行動）	「ケース A：物理アクセス」において攻撃を成立させるために必要な手口 ※括弧内の数字は図表 14 の攻撃可能箇所を示す	「ケース B：ネットワークアクセス」において攻撃を成立させるために必要な手口 ※括弧内の数字は図表 14 の攻撃可能箇所を示す	
Step1	デバイスロックを解除	デバイスロックを不正に解除	手口３（生体認証でのなりすまし）により、攻撃者がデバイスロックを不正に解除（⑤）	不要（注 3）	
Step2	取引内容を送信	取引内容を改ざんして送信	不要（注 1）	手口２（マルウエア）により、攻撃者が取引内容を改ざんして送信（②③）	
Step3	取引内容確認メッセージを確認	偽の取引内容確認メッセージを提示	不要（注 2）	手口２（マルウエア）により、攻撃者が取引内容確認メッセージを偽装（②③）	不要（注 4）
Step4	生体情報を提示（ユーザ検証）	ユーザ検証を不正に実施	手口３（生体認証でのなりすまし）により、攻撃者が不正にユーザ検証を突破（⑤）	不要（注 5）	手口２（マルウエア）により、攻撃者が不正にユーザ検証を突破（④）

図表 17. Step 毎の攻撃手口の整理（認証フェーズ）

（注 1）この場合には、攻撃者がユーザのデバイスに物理的にアクセスでき、Step1 にてデバイスロックを不正に解除できている。このため、攻撃者は、取引内容を改ざんするまでもなく、（攻撃者）自身の口座への送金指図（取引内容）を作成し送信すれば良い。取引内容を改ざんする必要が無いという意味において、（攻撃は）「不要」と記している。

（注 2）この場合には、攻撃者がユーザのデバイスに物理的アクセスでき、Step1 にてデバイスロックを不正に解除できており、Step2 にて攻撃者への送金指図が既に行われている。このため、攻撃者は、偽の取引内容確認メッセージを提示するまでもなく、表示された「（攻撃者）自身の口座への送金確認メッセージ」を確認すれば良い。偽の取引内容確認メッセージを提示する必要が無いという意味において、（攻撃は）「不要」と記している。

（注 3）この場合には、ユーザ本人が通常使用のためにユーザ自身のデバイスのロック解除を実施することが想定される。このため、攻撃者がデバイスロックを不正に解除しなくても、Step2 以降を攻撃者が実施することが可能であり、（攻撃は）「不要」と記している。

（注 4）この場合には、ユーザ本人がデバイスを使用しており、Step2 においてユーザの与り知らないところで取引内容が改ざんされており、次の Step4 においてユーザ検証が不正に実施されるものである。このため、Step3 にて偽の取引内容確認メッセージを提示せずと

も、攻撃者は「攻撃者への振込みの取引内容（transaction）」をユーザに提示すれば良い（次の Step4 にて、ユーザの意図に反してユーザ検証が実施されるため）。このため、攻撃者が偽の取引内容確認メッセージを提示する必要がないことから、（攻撃は）「不要」と記している。

（注 5）この場合には、ユーザ本人がデバイスを使用しており、Step2 においてユーザの与り知らないところで取引内容が改ざんされており、Step3 において同じく取引内容確認メッセージが偽装されている。ユーザからすると、自身の意図する取引内容を送信（Step2）し、取引内容確認メッセージでそれを確認（Step3）しているため、ユーザは疑いを持つことなく、自らの生体情報を提示することにより取引を実行しようとする。このため、ユーザ検証を攻撃者が不正に実行する必要がないことから、（攻撃は）「不要」と記している。

図表 17 をもとに、ケース毎、攻撃手口毎の攻撃成否を整理すると図表 18 の通りとなる（本文中の図表 12 と同内容）。図表内の【】は、攻撃成立の場合の図表 17 で示した Step および攻撃箇所を示す。なお、図表中の「ー」で示した部分は、図表 9 で示した「想定しない」攻撃手口を指す。

アクセス方法 / 攻撃の手口		ケース A：物理アクセス	ケース B：ネットワークアクセス
手口１：フィッシング		—	攻撃不成立（注 1）
手口２：マルウエア	偽アプリ型	—	攻撃不成立（注 2）
	凶悪型	—	攻撃成立（ロ） 【Step2②】【Step3②】 または 【Step2②】【Step4④】
手口３：生体認証でのなりすまし		攻撃成立（イ） 【Step1⑤】【Step4⑤】	—

図表 18. ケース毎、手口毎の攻撃成否に関する評価（認証フェーズ）再掲

（注１）原理的には、攻撃者が、ユーザをフィッシングサイトに誘導し、当該サイト上で「攻撃者へ振込みを行ってください」と指示を出し、ユーザがバンキングアプリを使って振込みを実施してしまうと攻撃が成功する。もっとも本稿では、そのような攻撃手法は、ユーザの意図せざる振込みではないという整理を行い、攻撃不成立としている。

（注２）この場合、攻撃を成立させるためには、正規 Authenticator が改ざんされた取引内容に対してデジタル署名を付す必要があるが、そのためには、正規 FIDO Client と偽バンキングアプリとを接続する必要がある。もっとも FIDO の仕様では、FIDO Client は接続可能なアプリケーションを FacetID と呼ばれる ID で識別できる仕組みがあり、Android においては、FacetID を Context（アプリケーションの環境情報を受け渡すために使用されるもの）から取得する実装例が例示されている。偽アプリ型マルウエアの前提では、サンドボックスの仕組みにより、偽バンキングアプリが正規バンキングアプリの Context を取得することが困難であると考えられるため、攻撃不成立としている。この FIDO における FacetID の仕組みの他、正規金融機関サーバは、偽バンキングアプリとの接続を拒否する仕組みを導入することによっても、同様の攻撃を防ぐことができる。

以下に、具体的な攻撃シナリオの例を見ていく。図表 17 で示した Step および攻撃箇所は、【】内に示す。

イ．手口３：生体認証なりすまし・ケース A：物理アクセス　の場合

- 攻撃者が、ユーザのデバイスのロック解除を、生体認証でのなりすましにより行う。具体的な生体認証でのなりすましの手法については、3．(3)ニ．で詳しく述べる【Step1⑤】。そのうえで攻撃者は、ユーザのデバイスからバンキングアプリを立ち上げ、自口座への振込み指示を行った後、Transaction Confirmation による確認（攻撃者への振込み指示が記されている）を、生体認証でのなりすましにより実施する【Step4⑤】。

ロ．手口２：凶悪型マルウエア・ケース B：ネットワークアクセス　の場合

- ユーザが正規バンキングアプリを立ち上げ、取引内容（例：A さんに 1 万円振込み）を入力するが、凶悪型マルウエアが正規のバンキングアプリのメモリ情報を書き換えるなどして（FFRI [2012]）当該情報を改ざん（例：X さんに 100 万円振込み）して、金融機関サーバに送信する【Step2②】。金融機関サーバは認証要求を正規バンキングアプリに送信し、正規 Authenticator が取引内容確認メッセージを表示（例：X さんに 100 万円振込み確認表示）するが、凶悪型マルウエアはタイミングを見計らい「Display Overlay 攻撃」（3．(3)ハ．参照）により当該画面上に、マルウエアが作成したメッセージ（例：A さんに 1 万円振込み確認表示）を出す【Step3②】。ユーザはマルウエアが作成した最前面のメッセージ内容を確認し、自らの生体情報を提示してしまうと、X さんに 100 万円の振込みが行われてしまう。
- 凶悪型マルウエアは、生体情報リーダと Matcher 間に流れるユーザの生体情報を盗聴し保持しておく。次に、ユーザが正規バンキングアプリを立ち上げ、上述と同じ要領で取引内容を改ざんしたうえで、金融機関サーバに送信する【Step2②】。金融機関サーバは認証要求を正規バンキングアプリに送信し、正規 Authenticator が取引内容確認メッセージを表示（例：X さんに 100 万円振込み確認表示）する。ユーザは不正な振込みであると気づいて取引をキャンセルしようとするが、凶悪型マルウエアは、Matcher にユーザの生体情報を流し込み（所謂、リプレイ攻撃）、それが Matcher で受理されれば、当該取引がユーザの意図に反して（自らの生体情報を提示しないにも関わらず）実行されてしまう【Step4④】。