情報処理学会論文誌 Vol.59 No.9 1639–1655 (Sep. 2018)

利便性・安全性・実在性・可用性を兼ね備えたIoTサービス利用者認証

二村 和明[1,2,a)] 矢崎 孝一[1] 伊藤 栄信[1] 坂本 拓也[1] 西垣 正勝[2]

受付日 2017年12月11日, 採録日 2018年6月8日

概要：生活をとりまく様々なIoT機器の利用時には，そのサービス利用の意思表示のため，利用者が物理的にIoT機器にタッチするという行為が必ず発生する．利用者の所有物による物理的なタッチにより，持ち主がサービス利用を希望していることが認証できるが，今後広がるIoTサービスには，これだけでは不十分であり，多数のサービスの中から利用サービスを選び出す手間を省いたり，課金型サービスのために所有物を所持している利用者が正規の所有者であることの確認や，利用者がIoT機器の目の前にいることの確認までできる必要がある．また，このようなIoTサービスでは，サービスによるデータ通信の品質確保が重要になる．そこで本稿では，利用者が普段携帯する生体認証機能を具備するスマートフォンを使い，それを持った利用者がIoT機器に物理的にタッチする動作を，利用しようとしているIoTサービスの自動特定および起動手段，利用者がIoT機器の前に所在することの確認手段として活用し，普段利用する生体認証機能による本人確認手段を活用することで，様々なIoTサービスの認証を一手に引き受けるIoTサービス利用者認証のためのフレームワークを提案する．その際，サービスサーバとIoT機器の間であらかじめ形成されているローカルな通信路に認証プロトコルを安全に収容し，データオフロードに配慮したIoTサービスのインフラを提供可能にする．

キーワード：スマートフォン，IoT機器，認証，クラウドサービス，トラストチェーン，データオフロード，FIDO

User Authentication for IoT Service Combining Convenience, Safety, Substantiality and Availability

Kazuaki Nimura[1,2,a)] Koichi Yasaki[1] Hidenobu Ito[1] Takuya Sakamoto[1] Masakatsu Nishigaki[2]

Received: December 11, 2017, Accepted: June 8, 2018

Abstract: When users use various IoT devices surrounding our daily lives, the user conduct physical touch ineluctably to the devices by a user belonging to signal for starting the use IoT services. By the physical touch using user belongings, it is possible to authenticate the possessor that desired to use the service, however in case of IoT service which would wide spread hereafter, that is not enough, then it would be required to cut out the burden to choose the service that user wanted form the various services, to authentication the user for billing transaction that would be required in addition to the authentication by the possession, and to confirm the user existence in front of the IoT devices. At the same time, caring about quality of service for the network and ensuring the scalability of the service are needed as the infrastructure to accommodate the increasing IoT services. Then in this paper, we propose a framework that compound the action of physical touch to automatic determination and initiation of the service that user trying to use by a smartphone penetrated in our daily life, and to proof of the user existence in front of a IoT device. In that regard, to overcome the issue of shortage of spectrum of the mobile Internet that would worsen by communication of IoT services, we provide a data offloading method targeting for IoT service that detour the payload to local communication channel between smartphone and IoT device by accommodating authentication and services protocols in it safely.

Keywords: smartphone, IoT device, user authentication, cloud service, trust chain, data offload, FIDO

1. はじめに

2020年には，数百億個のIoT [1] 機器がクラウドに接続され，生活や産業を大きく変革するといわれている．様々なIoT機器がネットワークにつながることにより，新たなサービスの展開も期待される．シェアードカーや宅配ボックス [2] などのシェアリングサービスがその一例である．

このようなサービスの利用時には，そのサービス提供開始に対する意思表示を行うため，利用者が利用権限の所有を示すメンバカードなどをかざすような，物理的にIoT機器に触わるという行為が発生する．Suicaがこの代表例である．このときIoT機器側では，「利用者がこの時点・状況でIoT機器に触ったということは，利用者はこのサービスを使いたいに違いない」と判断しサービスの提供を開始する．たとえばシェアードカーでは，利用者の所有物を車のドアノブにタッチした瞬間[*1]に「サービス利用申請 → サービス許諾 → 利用者認証 → ドアのアンロック」がなされると，ワンタッチで非常にスムースなサービス提供が可能となる．今後同様のIoTサービスが多数展開されることを想定すると，サービスごとの開発負担を軽減するためインターオペラビリティが担保された新たな認証フレームワークを提供することでサービス展開を加速することが望ましい．

利用者の所有物による物理タッチは「利用者に複数のサービスが提供されている環境において，利用者がどのサービスの利用を希望しているのかが特定できる」，「利用者がIoT機器の前に所在していることが確認できる」，「所有物の持ち主がサービスの利用を希望していることが認証できる」という観点で，サービス提供を開始するためのトリガとして効果的に機能する．ただし，3つ目の観点に関しては，今後増えてくると考えられる課金型のIoTサービスを想定すると，所有物の盗難紛失や不正アクセスについても考慮する必要がある．すなわち，単なる所有物認証だけでは不十分であり，「所有物を所持している利用者が正規の所有者である」ことが確認できなければならない．このため，何らかの本人確認の追加が必要になるが，利便性と安全性の観点からは生体認証の追加による2要素認証の実現が有用であると考えられる．

また，IoTサービスを提供するためのインフラとしては，サービス品質（QoS）とスケーラビリティの確保が肝要である．現在のモバイルインターネットにおける無線通信量は，すでにサービスの安定提供に支障を来たす状況にあり，Wi-Fiなどへのデータオフローディングが通信キャリアによって積極的に進められている [3]．今後，IoT機器による通信の増大により，通信環境はさらに圧迫される．フィージブルなIoTサービスの実現のためには，データオフロードの仕組みをあらかじめインフラに組み込むことで，サービスの安定提供を可能にしておく必要がある．

そこで本稿では，利用者が普段携帯するスマートフォンを使い，上記を満たすIoTサービスの利用者認証を効率的に提供するフレームワークの確立を目指す．具体的には，スマートフォンを持った利用者がIoT機器に近づき物理的にタッチして利用するという動作を，利用しようとしているIoTサービスを自動特定して起動するための手段，かつ，利用者がIoT機器の前に所在していることを確認するための手段として活用する．さらに，スマートフォンに搭載されている生体認証機能によって，様々なIoTサービスの認証を一手に引き受けることで，ワンストップで利便性の高いIoTサービスの利用を可能にする．その際，サービスサーバとIoT機器の間であらかじめ形成されているローカルな通信路に認証プロトコルを収容し，この通信路を通じてスマートフォンからの認証情報をサービスサーバ側に送ることにより，データオフロードに配慮したIoTサービス提供を可能とする．

提案手法の実装にあたり，利用者認証のための生体情報をスマートフォンの中に閉じ込めることによりシンプルかつ安心な運用が可能となるFIDO（Fast IDentity Online）[4] プロトコルを活用する．

なお，本稿は文献 [5] を基に，提案手法の理論体系を深化させたものである．

2. フレームワークが具備すべき要件と関連技術

ここでは本稿が想定する要件についてまず述べる．その後に関連技術を示して，要件の充足関係について比較を行う．

2.1 クラウドサービスの構成

ここでは本稿が想定するクラウドサービスの構成について述べる．

図 1 (A) は現在のオンラインサービスの概念図を表している．サービスサーバにおいてコンテンツが提供されており，利用者は自身が所持する携帯端末を用いてサービスサーバに直接アクセスし，携帯端末上でサービスを利用する．クラウドサービスでは，サービスサーバがオープンス

1 株式会社富士通研究所
Fujitsu Laboratories Ltd., Kawasaki, Kanagawa 211–8588, Japan
2 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University, Hamamatsu, Shizuoka 432–8011, Japan
a) kazuaki.nimura@jp.fujitsu.com

[*1] 理想的には，利用者が車のドアノブに（所有物をかざすのではなく）手をかけた瞬間にサービスの提供を開始できることが望ましい．それを実現しうる技術の1つが人体通信である．利用者は人体通信モジュールを身に付けておけば，IoT機器に手を触れるだけで，人体通信モジュールからの信号がIoT機器に伝わり，IoT機器のサービスが起動される．しかし，現時点において，人体通信を搭載しているスマートフォンが存在していないため，本稿では人体通信は対象外としている．

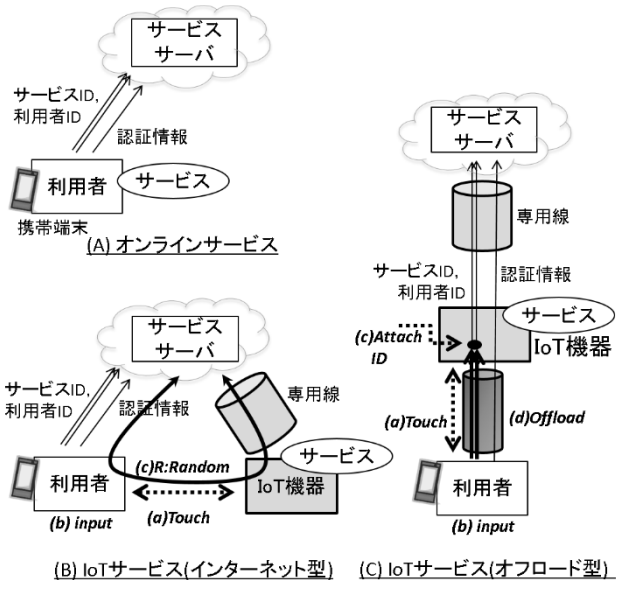

図 1 オンラインサービスと IoT サービスの概念図
Fig. 1 Conceptual diagram of cloud online service and IoT service.

ペースのクラウド上にあるため，携帯端末からサービス ID の指定が必要になる．サービスサーバは，自身が提供するサービスに対応したサービス ID が指定されると，利用者に利用者 ID の入力を促す．サービスサーバは携帯端末の認証情報の検証も行う．利用者が利用者 ID を入力すると，サービスサーバはパスワードなどの認証情報を求める．そこで利用者は，携帯端末を介して利用者の認証を行い，その認証情報（パスワードなど）をサービスサーバに提供する．この認証結果を事前に保持されている認証情報と比較し，認証結果が正しければ，サービスサーバは携帯端末にコンテンツを提供し，利用者はそのサービスを利用する．これらの通信はすべて一般網を経由して行われる．

これに対し，IoT 機器をサービス・コンテンツとしたサービスが IoT サービスである．IoT サービスでは，図 1 (B) あるいは図 1 (C) のように，サービスサーバの指示に従い IoT 機器がサービスを提供する．サービスサーバと IoT 機器間は専用線で接続されている．IoT サービスでは通常，サービスサーバ側で IoT 機器の状態をつねに管理する必要があるため，この専用線は常時接続している安全なローカル通信路であることが前提となる．すなわち，ファームウェアやアプリケーションのオンライン更新も可能であり，セキュリティパッチが速やかに適用されうる．なお，IoT サービスを提供する IoT 機器は，クラウド上のサービスサーバならびに利用者のスマートフォンと連携する必要があるため，両者との通信機能についても有していることが前提となる．

IoT サービスでは，クラウドサービスの構成に対して IoT 機器が加わることになるため，サービス提供の手法を図 1 (A) の形から発展させる必要がある．図 1 (B) は，近接通信によって携帯端末と IoT 機器の間で通信を行いながら，サービス ID，利用者 ID，認証結果などの認証情報については，一般網を用いて携帯端末からサービスサーバへ直接送信する構成であり，本稿ではインターネット型と呼ぶ．これに対して，図 1 (C) は，サービスサーバと IoT 機器が形成している専用線の IoT 機器側に，近接通信によって携帯端末を利用のつどつなぎ，この経路によりサービス ID，利用者 ID，認証結果などの認証情報をサービスサーバに提供する構成を示しており，本稿ではオフロード型と呼ぶ．

2.2 要件

以下では，IoT サービスの要件について整理する．

(a) 現在，利用者は，suica のように専用カードを IoT 機器にかざしたり，Walmart Pay [6] や Alipay [7] のように専用アプリを起動するなどの方法で IoT サービスを利用している．しかし，多様な IoT サービスが提供されている環境においては，サービスの数だけメンバカードやスマートフォンアプリを用意するような方法は，利用者にとっては煩雑となって良策ではない．すなわち，利用者がスマートフォンアプリを意識的に事前起動したり，サービス ID を明に提示したりすることなくサービス提供できることが望ましい【利用サービスの自動特定】．

(b) IoT サービスを利用するにあたっての利用者認証には，パスワード認証，所有物（トークン）認証，生体認証が適用可能である．パスワード認証は広く利用されているが，利用者が利用者 ID と認証情報を入力する手間や，無数に存在しうるサービスごとに異なるパスワードを記憶あるいは管理する必要があり利便性が高いとはいえない [8]．所有物認証は，利用者が所有している物を認証に用いる手法で，スマートフォンとの親和性が高いうえに，サービスに応じた利用者 ID とトークン（スマートフォンに格納されている認証情報）を IoT 機器に同時に伝えることができるため利便性が高いが，スマートフォンの置き忘れ・紛失・盗難の際に問題が生じる．今後の課金型の IoT サービスにおいては，トークンを所持している利用者であることを確認する（本稿では「弱本人確認」と呼ぶ）だけでなく，トークンを所持している利用者が正規の所有者であることまで確認できる（本稿では「強本人確認」と呼ぶ）必要がある．そして，その目的に対し，利用者認証まで可能な生体認証は，利便性と安全性の両面で優れた認証情報入力手段である．以上より，サービスの状況に応じて，所有物認証と生体認証を組み合わせることが可能な多要素認証の実現が必要になる【利用者の強本人確認】．

(c) IoT サービスにおいては，正規の利用者が誤って（あ

るいは不正者の誘導によって）IoT 機器を起動させてしまった際に深刻な問題が発生する場合があるため，これを防ぐ手立てが必要である．1 章では物理タッチを行う利用シーンをあげたが，それが行われない場合どうなるであろうか．たとえば，シェアードカーのドアを遠隔から解錠可能な運用としてしまうと，自宅にいる利用者が誤操作（あるいは不正者の誘導）によって車のドアを開錠して運転可能な状態にしてしまった場合，車の前にいる別の人（あるいは不正者）がその車を運転できてしまう．これに対処するためには，利用者の位置情報を取得して，サービス利用時に利用者が IoT 機器の前にいることを確認することが必要になる．しかしその際，GPS のようにつねに利用者を監視するタイプの検知手段を用いると，プライバシ問題につながる懸念がある．よって，利用者が IoT サービスを受けようとする際に，必要最低限の情報取得のみで，利用者が実際に IoT 機器の前にいることを確認できるようにする必要がある【利用者の所在確認】.

(d) 従来，安定したモバイルインターネット通信の実現のため，Wi-Fi 通信へのデータオフローディングによる負荷軽減が行われている．公共 Wi-Fi 通信に接続させることで [9]，オフロード比率は 6 割にも達しており [10]，この効率化のために，時間的，空間的，通信路的にオフロードすることも考えられている [11]．IoT 機器による通信の増大にともない，さらにモバイルインターネットの帯域が圧迫されることは必然である．このため，IoT サービスと IoT 機器がつながる専用通信路を活用して認証・サービス提供のための通信のデータオフロードをインフラの基本機能の中に組み込んでおくことにより可用性を高めることが必要になる．その際，不正な Wi-Fi アクセスポイントによる不当行為の横行に鑑み，安全なデータオフロード環境が提供されなければならない．すなわち，図 1 (C) のように，サービスサーバと IoT 機器の間であらかじめ形成されている安全なローカル通信路に認証プロトコルを収容するようなインフラ設計が肝要となる【安全なデータオフロード】.

2.3 関連技術

ここでは関連技術を示し，要件との比較を行う．2.2.1〜2.2.3 項は所有物認証の関連技術，2.2.4〜2.2.5 項は生体認証の関連技術である．

2.3.1 IC カード

Suica に代表される IC カードでは，ショッピングや電車に乗るときにワンタッチで所有物認証 [12] を行うことができる．

この IC カード方式と要件との関係についてまとめる．要件 (a) については，利用者が利用したいサービスの IC カードを取り出し，IC カードを IC カードリーダにタッチすることで，サービスを利用する．すなわち，利用サービスの特定は，利用者により行われ，自動的に選択されることはない．要件 (b) については，利用者が IC カードを所持していることは確認できるが，正規の持ち主が IC カードを利用していることまで確認を行わないため，盗難紛失時に他の人から悪用される可能性がある．要件 (c) については，IC カードと IoT 機器の間の接触・非接触タッチにより，利用者が IoT 機器の前に所在していたことが確認できる．要件 (d) については，Suica の場合はデータオフロードのために IC カードシステムの専用通信路を用いている．

2.3.2 オンライン型トークン

スマートフォンによる電子航空券 [13] に代表されるオンライン型トークンは，Web ログインなどで用いられ，オンラインでトークンが発行される．これは，OAuth [14]，OpenID connect [15] など Web ログインの連携やサービス間連携で用いられる．たとえば，サービス会社の Web サイトにログインしてトークンの発行を受け，そのトークンを QR コードなどに変換して会場に入場する際に読み取りを行うことでサービスの利用権の認証が行われる．このトークンを QR コード読み取り装置などの IoT 機器に提示することで利用権の認証を行うことが可能である．

このオンライン型トークンと，要件との関係についてまとめる．要件 (a) については，トークンの利用時に，利用者が利用したいサービスのトークンを選択して取り出し，提示することが必要になる．要件 (b) については，トークンを提示することによりトークン発行を受けた本人であることの確認が行われるが，トークンが第三者にわたると IoT 機器を不正に操作可能であるというリスクがある．要件 (c) については，トークンが提示されることにより，利用者が IoT 機器の前にいることを確認可能である．要件 (d) については，オンライン型トークンでは，オープンな通信路が使用される場合が一般的で，データオフロードのために専用線を使用することは多くない．

2.3.3 IC カード対応スマートフォン

スマートフォンには IC カードに対応可能なものがある．たとえば，Apple pay [16] では，IC カードのプロトコルを変更することなくスマートフォンに搭載し，クレジットカード情報を登録することで，物理タッチのみで少額の支払いを行うことができる．また，Android pay [17] のようにクレジットカード情報をスマートフォンに持たせないタイプや，航空券と連携させたサービス [18] もある．これらは利用者が IC カードを所有しているかどうかをチェックするが，さらに盗難対策を目的として，利用者認証を呼び出すこともできる．

この IC カード対応のスマートフォンをベースとしたサービスと，要件との関係についてまとめる．要件 (a) については，（スマートフォンの中に登録されている複数の IC

カードの中から）利用したいICカードを利用者があらかじめ選択する必要がある．iPhoneにおけるSuica利用のように改札にタッチするだけでチャージされるサービスもある．しかし，利用シーンがスマートフォンとICカードに限定され，サービスから直接認証結果や利用者の存在を確認をするような汎用的な利用を行うことはできない．要件 (b) については，盗難対策用の利用者認証として生体認証を採用することにより，利用者の強本人確認を行うことも可能である．要件 (c) については，ICカード読み取り装置へのタッチにより，機器の前に所在していたことが分かるが，サービス側から直接利用者の所在を確認することはできない．要件 (d) については，決済機器の回線が使われ，オフロードは行われていない．

2.3.4 生体認証のIoT機器搭載

IoT機器ごとに生体認証機能 [19] が取り付けられる場合には，利用者は生体センサによるスキャンによる認証だけで利便性良くIoT機器が利用可能になる．一方，生体情報は利用者が変えることのできないIDでもあるため，その情報が漏れないようにする対策が必要でシステムコストが高くなる．また，IoT機器へのセンサ設置が必要であることもコストに反映され，ユーザインタフェースも機器ごとに異なる場合が多い．

この生体認証がIoT機器に搭載されている場合と，要件との関係についてまとめる．要件 (a) については，IoT機器に専用の生体認証が組み込まれており，利用サービスはこれに紐付いていることから，利用サービスの自動特定を行う必要はない．要件 (b) については，利用者が直接IoT機器を操作するため，強本人確認は行われている．要件 (c) については，利用者は直接IoT機器を操作するため，IoT機器の前にいることを判断できる．要件 (d) については，IoT機器の備えるローカルな通信路にデータオフロードされている．

2.3.5 生体認証のスマートフォン搭載

生体認証機能を搭載したスマートフォンが増えている．また最近は，生体認証を扱う基盤としてFIDOが用いられる場合も多い．FIDOは，オンラインサービスなどへのログインにスマートフォンなどが備える生体認証 [20] を用いて本人確認を追加して認証を行うことができる．この認証器には生体認証にとどまらず，2要素認証の適用なども想定されている．たとえばエンタテイメント性を活用した認証 [21] などを組み込むことで利便性を高めることも可能な仕組みとなっている．

この生体認証付きのスマートフォンでサービスを利用した場合と，要件との関係についてまとめる．要件 (a) については，サービスの利用時には，利用者が何のサービスを受けたいのか選択が必要になる．要件 (b) については，生体認証により強本人確認が行われる．要件 (c) については，IoT機器へのタッチをともなわないため，IoT機器の前にいることは確認できない．要件 (d) については，IoT機器側の通信路へのオフロードは考慮されていない．

ここまで既存技術について述べてきた．これらを組み合わせた様々なサービスが今後展開される可能性があるが，その開発の効率化が望まれる．本稿の貢献は，4つの要件を同時に満たし，IoT機器を活用したシェアードサービスなどを効率的に組み上げるフレームワークを明確化することにある．

3. 提案手法

2章にあげた要件を同時に満たすための提案手法について説明する．このフレームワークにより，様々な利用シーンへの適用をしやすくする．

3.1 要件 (a)：利用サービスの自動特定

IoTサービスでは，IoT機器によってサービスが提供されるため，サービス利用時に利用者が個々のIoT機器と接触することになる．この特徴を活かし，利用者とIoT機器の物理的接触によって，利用者が利用しようとしているIoTサービスを自動特定することができる．このために，IoT機器にサービスIDを設定し，利用者が所有する機器（スマートフォン）を使ってIoT機器からサービスIDを読み取れるようにする．

本稿ではIoTサービスは複数のIoT機器をかかえてサービス提供している前提とし，個々のサービスサーバが管理する複数のIoT機器には同じサービスIDを付番するものとする．そして，利用者とIoT機器の物理的接触の際に，利用者のスマートフォンを用いてIoT機器からサービスIDを取得することで，利用者が利用しようとしているIoTサービスを特定できるようにする．

利用者のスマートフォンには，複数のIoTサービスIDを登録できるようにする．そして，スマートフォンにインストールされているサービスソフトウェア群の中から，各サービスIDに対応するソフトウェアを呼び出すことができるようにする．これにより，サービスを提供するIoT機器に利用者が近づいた際に，利用者のスマートフォンがIoT機器のサービスIDを認識し，対応するサービスソフトウェアを起動することで利用者にサービス提供を開始できる．

なお，IoT機器はIoT機器固有のIDを表すIoT機器IDも備えているため，同一サービスIDであってもIoT機器ごとに異なるソフトウェアが起動されるように利用者のスマートフォンを設定することも可能である．

3.2 要件 (b)：利用者の強本人確認

IoTサービス利用者の強本人確認のため，利用者のスマートフォンを活用した所有物認証と生体認証による2要素認証を行う．所有物認証の目的は，利用者IDと暗号鍵

の所持の確認である．生体認証の目的は，それらが正規の所有者に所持されていることの確認である[*2]．

利用者のスマートフォンは，利用者 ID と暗号鍵を保持している．この暗号鍵を所有物認証のための認証情報として用い，利用者のスマートフォンとクラウド上のサービスサーバの間で公開鍵暗号をベースとした認証プロトコルを実装する．標準プロトコルを採用することで汎用性を高め，サービス，IoT 機器，スマートフォンごとに様々な認証方式をサポートすることを避ける．

また，利用者のスマートフォンは生体認証機能を備えているものとし，生体認証にパスした場合にのみ，スマートフォン内の利用者 ID と暗号鍵が認証プロトコルに従ってサービスサーバに渡される．生体情報（生体認証のための認証情報）は利用者のスマートフォンに閉じ込め，それ以外で使われることはないようにする．IoT 機器やサービスサーバでセンシティブな情報を扱う必要がなくなるため，情報漏えいリスクに対する管理コストを軽減可能である．

IoT 機器はハードウェアリソースが限られている場合があるが，クラウド上のサービスサーバならびに利用者のスマートフォンとの通信機能は確保されている．そこで，IoT 機器には認証のための機能は搭載せず，サービスサーバとスマートフォンの間で所有物認証の認証プロトコルを実行するようにする．これにより，利用者の強本人確認が IoT 機器に非依存となり，利用者は，あらゆる IoT 機器に対して，つねに同じ方法（自身のスマートフォンに生体情報を提示する）で認証を行うことができるようになる．

3.3 要件 (c)：利用者の所在確認

利用者が実際に IoT 機器の前にいることを，必要最低限の情報取得のみで確認できるようにするため，利用者と IoT 機器の物理的接触の中に利用者の所在確認の仕組みを組み込む．2.1 節で述べたように，IoT サービスでは低リソースの IoT 機器であっても（クラウド上のサービスサーバならびに）利用者のスマートフォンとの通信機能は確保されている．この「IoT 機器とスマートフォンの間の通信機能」を近接通信技術によって実装することで，利用者と IoT 機器の物理的接触を「IoT 機器とスマートフォンの間の近接通信の発生」というイベントによって可視化する．

利用者が実際に IoT 機器の前にいること（利用者と IoT 機器の物理的接触）をサービスサーバで確認するためには，IoT 機器とスマートフォンの間の近接通信発生のイベントをサービスサーバに伝える必要がある．これには，図 1 (B) と図 1 (C) に示した 2 つの構成（インターネット型，オフロード型）が考えられる．

図 1 (B) のインターネット型では，IoT 機器とスマートフォンの間の近接通信が発生した際に，乱数 R を発生させて IoT 機器とスマートフォンで共有するようにする．利用者のスマートフォンからサービスサーバに乱数 R を渡し，これと並行して IoT 機器からもサービスサーバに乱数 R を渡す．サービスサーバで 2 つの乱数 R が同じであることを確認することで，利用者の所在確認が完了する．しかし，この方式は後述の 3.4 節の要件 (d) を満たさない．

図 1 (C) のオフロード型では，IoT 機器をサービスサーバとスマートフォンをつなぐ媒体として利用する．2.1 節で述べたように，IoT サービスでは，サービスサーバと IoT 機器の間には安全なローカル通信路があらかじめ開通されている．このため，利用者のスマートフォンからの情報が，この「サービスサーバと IoT 機器の間のローカル通信路」を通じて，サービスサーバに届くこと自体が，利用者の所在確認（スマートフォンと IoT 機器が近接通信によって接続されていること）となる．この方式は，後述の 3.4 節の要件 (d) の要求も同時に満たすためより好ましいと考えられる．

3.4 要件 (d)：安全なデータオフロード

IoT サービス上の通信に対するデータオフロードを実現するために，サービスサーバと利用者のスマートフォンとの間の通信を，2.1 節で述べた「サービスサーバと IoT 機器の間のローカル通信路」に収容する．そこで図 1 (C) のオフロード型の構成を採用し，IoT 機器を「サービスサーバとの通信のためのアクセスポイント」として利用する．IoT サービスを利用する利用者が IoT 機器と物理的に接触した時点で，利用者のスマートフォンが IoT 機器と近接通信で接続され，その後の「スマートフォンとサービスサーバの間の通信（3.2 節で述べたスマートフォンと IoT 機器の間で実行される認証プロトコルの通信を含む）」が「IoT 機器とサービスサーバの間の通信路」に収容される．

利用者はいつも同じ IoT 機器を利用するわけではないため，任意の IoT 機器を介して安全なデータオフローディングが担保されなければならない．このために，スマートフォン内に IoT 機器への経路スイッチが可能なトンネリング機能を設けるとともに，サービスサーバ →IoT 機器 → スマートフォンの間でトラストチェーン [22], [23] による信頼の連鎖を構成するようにする．ここで，「サービスサーバと IoT 機器の間のローカル通信路」は，あらかじめ開通されている安全な通信路であることが前提となっている．また，3.3 節で述べたように，「IoT 機器とスマートフォンの間の通信」を近接通信によって実装することで，利用者と IoT 機器の物理的接触が保証される．このように，あらかじめ確保されているトラスト（サービスサーバ →IoT 機器）に対し，物理的接触に基づくトラスト（IoT 機器 → 機器端末）をそのつど連鎖させることによって，安全な通信経路が毎回動的に確保される（詳細は 5.1.1 項を参照され

[*2] 利便性の観点から生体認証が最適であると考え，本稿では生体認証により利用者認証を行うものとして進めるが，パスワードなど生体認証以外の利用者認証を用いてもよい．

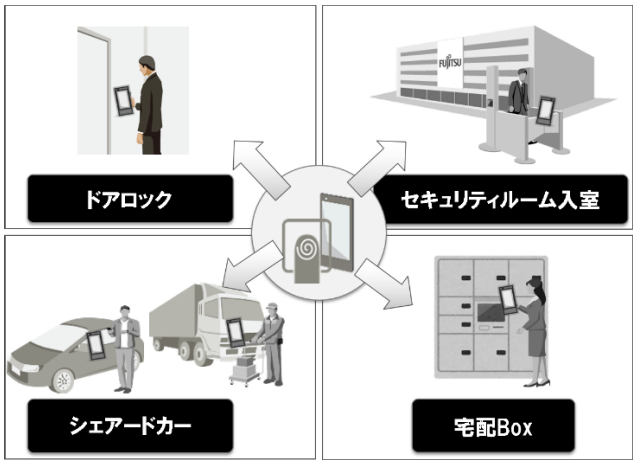

図 2 利用シーン
Fig. 2 Use case.

たい）．

図 1 (B) のインターネット型は，サービスサーバはモバイルインターネット経由で不特定多数の利用者のスマートフォンからアクセスを受付けるため，遠隔地の攻撃者からサービスポートへの不正アクセスや DoS 攻撃を受けるリスクを孕む．図 1 (C) のオフロード型であれば，サービスサーバは IoT 機器経由でのアクセスのみを受付ければよいため，不正者はいずれかの IoT 機器に近接通信によって接続しなければ攻撃を実行できない．サービスサーバは各 IoT 機器ごとにサービスポートのアドレスを事前に設定しておけばよいので，全利用者にサービスポートのアドレスを周知する必要もなくなり，より安全性の高いサービス提供が可能になる．以上の観点から，図 1 (C) のオフロード型のほうが図 1 (B) のインターネット型よりも好ましいと考えられる．

3.5 ユースケース

提案手法の利用シーンを**図 2** に示す．「ドアロック [24], [25], [26]」では，ホテル [27], [28]，民泊，託児所，訪問介護・家事サービスなどにおいて，入退室管理に利用者のスマートフォンによるタッチと生体認証を用いることで，物理的な鍵の受け渡しをすることなく解錠が可能になる．「セキュリティルーム入室」では，利用者のスマートフォンによるタッチと生体認証を用いることで入退室が可能になる．「宅配 Box [29], [30], [31]」では，利用者のスマートフォンによるタッチと生体認証を用いることで，パスワードを利用することなく，本人だけが鍵の解錠を行うことが可能になる．「シェアードカー」では，ドアの解錠 [32], [33], [34] や，カーナビやコックピットなどのパーソナライズが利用者のスマートフォンによるタッチと生体認証により可能になる．

このように，IoT 機器へのタッチによるサービス開始要求と，利用者認証を行うことが必要となる様々なサービスにおいて，利用者が普段利用する利用者のスマートフォンを共通の鍵として用いることにより，安心安全かつ簡易な利用者認証を提供できるようになる．

提案手法では，IoT サービス利用時に，利用者認証と利用者の所在特定が確実に行われることから，利用者が IoT 機器の前にいることを確認したうえでのリスクベース認証 [35] を構成可能である．たとえば，シェアードカーの利用場所が利用者の普段の活動圏と異なるような場合には，追加認証を行うなどの施策をとることが可能である．また，宅配 Box の例では，利用者が宅配 Box から離れている間は，宅配 Box を誰にも使用できないようにロックすることができる．これにより，利用者に対して安心安全なサービス提供ができるようになる．

これらは，図 1 (A) のオンラインサービスとは異なる特長である．提案手法では，利用者が本当に IoT 機器の目の前にいることを物理タッチにより明確に判断することで，物理世界の安心感を取り込んでいる．

さらに，提案方式を Digital Twin [36]，Device Shadow [37]，WoT Servient on Cloud server [38] などと組み合わせて，IoT 機器と IoT 機器からあがってくるデータをクラウドインスタンスに蓄積することで，そのときその場所の IoT 機器の利用履歴を利用者と確実な形で紐付けることができる．ここに利用者の同意を得るための適切な手段を追加することによって，信頼度の高い利用者データをプライバシに配慮した形で第三者に情報提供することが可能になる．たとえばシェアードカーでは，利用者の同意の下で運転履歴情報を利用者と紐付け，保険会社など第三者に提供し，運転履歴に応じた保険料の算出（テレマティクス保険 [39], [40], [41]）などのサービスに結び付けることができる．

4. 実装

提案手法の実装について述べる．まず 4.1 節で，近年スマートフォンへの搭載が行われており，我々が実装に活用する FIDO 仕様についての概要をまとめる．4.2〜4.5 節で，提案手法の実装について説明を行う．これによりサービスサーバ上のサービスやスマートフォン上のアプリケーションから見て，認証を外部サービスとして扱うことが可能な API 化がなされる．4.6 節では，提案手法の登録と認証の動作について説明する．

4.1 FIDO

FIDO アライアンスはパスワード不要のオンライン認証技術を策定する団体で，Google，DoCoMo，VISA，ARM，Bank of America など 250 を超える組織が加入している．FIDO アライアンスが定める仕様には，UAF（Universal Authentication Framework）と呼ばれる仕様があり，これは，生体認証を含めた様々なローカル認証をオンライン

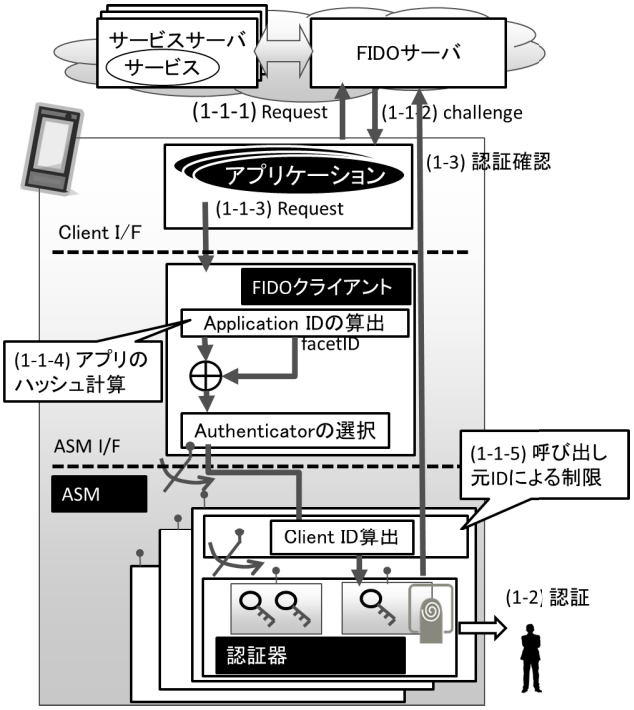

図 3 FIDO UAF プロトコルの処理
Fig. 3 Transaction of FIDO UAF protocol.

サービスが利用するためのプロトコルを規定している．

UAF では，スマートフォン内のセキュア領域で生体認証を処理し，認証が成功した場合に公開鍵暗号の秘密鍵をアクティベートする．FIDO サーバはスマートフォンの秘密鍵に対応する公開鍵だけを持っていればよい．これにより，生体情報やパスワードなどの個人認証情報の管理をオンラインサービス側で行うことが不要になる．

以下では，この FIDO のモジュール構成，登録，認証，およびセキュリティについてまとめる．

4.1.1 モジュール構成

UAF アーキテクチャ [53] は，以下のモジュールから構成される（**図 3**）．

- FIDO サーバ：FIDO サーバは，サービスサーバからの利用者認証要求により，アプリケーションを経由して FIDO クライアントに送出する UAF Request パケットを生成する機能を備える．また，認証が行われた後に送られてくる UAF パケットを解釈し，その中に含まれる署名データの検証を行う機能を備える．この検証が正しければ，その UAF パケットに対応する利用者 ID をサービスサーバに返す．
- FIDO クライアント：FIDO クライアントは，UAF 仕様で規定されているプロトコルに準拠した UAF パケットを解釈し，認証器を制御するための UAF パケットの生成を補助する．この FIDO クライアントは，SDK（Software Development Kit）を通じてアプリケーションの中に組み込まれる場合と，独立したソフトウェアとして組み込まれる場合がある．
- ASM（Authenticator Specific Module）：ASM は，スマートフォンのセキュアな空間に実装された複数の認証器を，FIDO クライアントを介してアプリケーションから利用可能にするためのブリッジ機能と，FIDO クライアントを介してアプリケーションから認証器を呼び出すためのインタフェースを提供する．
- FIDO 認証器（Authenticator）：認証器は，指紋認証など利用者を認証する認証方式ごとに用意されるモジュールである．このモジュールは利用者の生体情報を扱うため，スマートフォン内のセキュア領域で動作させ，利用者の認証結果を鍵のアクティベートに用いる．このモジュールは，ASM からの呼び出しインタフェースのみを備えている．

4.1.2 登録

動作の前に，UAF 仕様に則って生成した公開鍵を，サービスサーバ経由で FIDO サーバが利用するデータベースに登録する．

4.1.3 認証

スマートフォン上のアプリケーションを使って，サービスサーバにログインするときの利用者認証の動作は以下のようになる（図 3）．

(1-1) 認証要求

(1-1-1) Request：スマートフォン上のアプリケーションがサービスサーバ上のサービスに対してサービスログイン要求を発行する．

(1-1-2) Challenge：サービスサーバは FIDO サーバを経由して利用者認証のプロトコルを開始する．FIDO サーバからは UAF パケットが生成・発行される．

(1-1-3) Request：アプリケーションは，認証処理依頼の UAF パケットを受け取ると，FIDO クライアントに認証処理を依頼する．

(1-1-4) アプリのハッシュ計算：FIDO クライアントは，認証処理依頼を受けると，依頼元アプリケーションのハッシュ値を計算して facetID[42] を求め，UAF パケットに追加する．その後，UAF パケットに記載されている Policy [43] を解釈し，サービスサーバが期待する ASM・認証器を選択して認証処理を ASM に依頼する．

(1-1-5) 呼び出し元 ID による制限：依頼を受けた ASM は，呼び出し元の FIDO Client ID のハッシュ計算とチェーンの形成を行う．この Client ID と UAF パケットの内容をもとに，セキュア領域にある認証器の駆動および認証処理の依頼を行う．

(1-2) 認証

認証器は，認証処理依頼を受け取ると，自身の認証方式により利用者認証を実施する．そして認証結果にデジタル署名を付したうえで FIDO サーバに返す．

(1-3) 認証確認

FIDO サーバは，署名データに含まれる KeyID [44] から，

データベースに登録されている公開鍵を取得する．認証結果のデジタル署名を検証し，署名データに含まれる情報から利用者の本人性を確認して，サービスサーバに結果を報告する．

4.1.4 FIDO のセキュリティ

UAF パケットには，リプレイ攻撃を防ぐ仕掛けが組み込まれており，モジュールを通過するごとにパケットに以下の値を追加していく [45]．

- FIDO サーバが設定するチャレンジデータ
- FIDO クライアントがハッシュ計算した FacetID
- NONCE
- SSL セッション情報，サーバ証明書，時刻
- 鍵を特定するための Key ID
- 認証器が生成する利用者の確認結果

4.2 要件 (a)：利用サービスの自動特定

提案手法の実装では，利用サービスの自動特定のために，サービスサーバ，IoT 機器，スマートフォンに対して以下のような機能を搭載する．

サービスサーバには，サービスごとに固有の ID（URL ベースのサービス ID）が割り振られる必要がある．その一貫性を保証するため，本フレームワークでは，FIDO サーバにサービス ID を発行する機能を設け，サービスサーバがサービス ID を FIDO サーバが発行することとした．

サービスサーバは，そのサービスを提供するすべての IoT 機器を把握していなければならない．IoT 機器にはサービス ID が事前に登録されており，このサービス ID を，スマートフォン内のトンネリング機能が読み出せるようになっていなければならない．これを満たすため**図 4** のように，サービスサーバからサービス ID を受け取る機能とサービス ID を発信する機能を IoT 機器内のトンネリング機能の中に付加した．

利用者がサービス利用登録をする際に，スマートフォンにサービスアプリケーションがインストールされるとともにサービス ID が登録される．スマートフォンには，複数のサービス ID を登録できるようになっている．

サービス ID を発信している IoT 機器に，スマートフォンを物理的接触させることで，IoT 機器のサービス ID がスマートフォンによって読み取られる．その際，スマートフォンは，IoT 機器から実際に読み取ったサービス ID が，利用者のスマートフォンに登録された ID のいずれかと一致することを確認する必要がある．それを実現するため，IoT 機器からサービス ID を受け取り，そのサービス ID の一致を確認する機能をスマートフォンの ASM Interface（I/F）内に設ける．一致が確認された場合には，スマートフォンは IoT 機器のサービス ID に紐付いているサービスアプリケーションの起動を行う．

以上により，利用者が物理的に IoT 機器にタッチするだけで，明示的なサービス選択を行うことなく，どの IoT サービスに対する要求があったのか特定することが可能になり，あらゆるサービスを簡易に利用できるようになる．

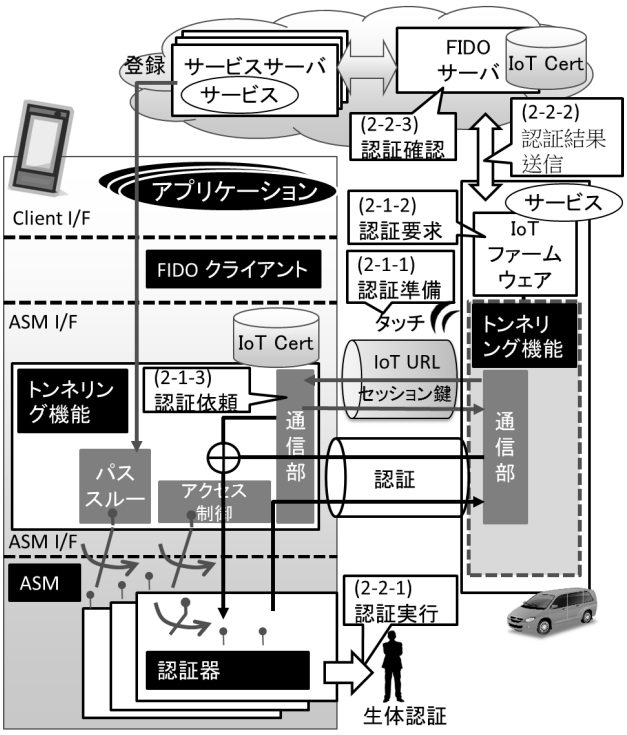

図 4 提案手法の構成
Fig. 4 System configuration of proposed method.

利用者の物理的な IoT 機器へのタッチ検出には Near Field Communication（NFC）[46] や Bluetooth Low Energy（BLE）[47] などの近接通信を用いる．このうち BLE では，ホスト機能とペリフェラル機能のうち，ペリフェラル側がビーコンを発することになっている．提案手法では，サービスを提供する IoT 機器がホストとなるべきであり，利用者認証機能を備えるスマートフォンがペリフェラルとなる．よって，スマートフォン側がビーコン発行を担うことになる．しかしこれは，スマートフォン側がつねづね（物理タッチを行う以前から）ビーコンを発行し続ける必要があるため，スマートフォンのバッテリを消費させてしまう．IoT 機器は電源供給がなされる場合も多く，低消費電力設計されている場合が多いことから，IoT 機器でビーコン発信を担うことでスマートフォン側の負担を減らした手法を実現することが好ましい．

そこで**図 5** に示すように，通常のスマートフォンのビーコン発行処理の前に，ホストとペリフェラルを入れ替えた「IoT 機器がビーコンを発行するモード」を新たに付け加えることで，この課題に対処する．この手法では，普段は IoT 機器がビーコンの発行機能を担い，そこに近づいてきたスマートフォンが，その電波レベルにより IoT 機器を認識する．そして，このビーコン検知をトリガとして，スマートフォンと IoT 機器間でセッション鍵を交換するとと

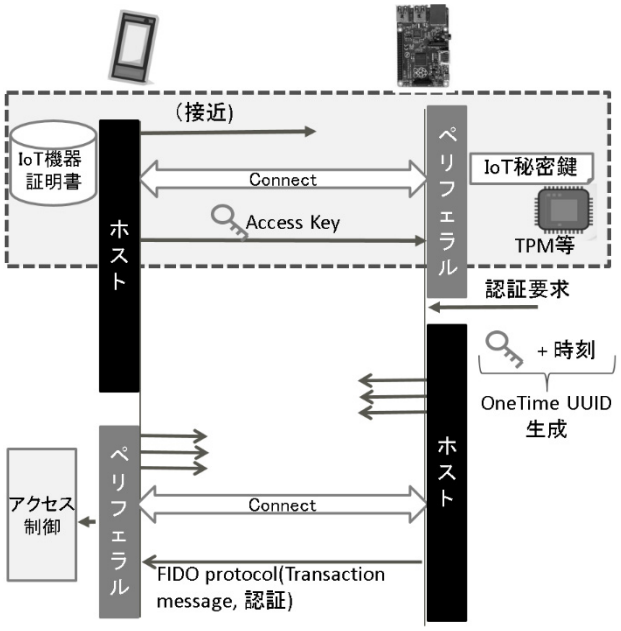

図 5 BLE 方式での自動ペアリング
Fig. 5 Automated pairing for BLE.

もに，それぞれの動作モードを切り替える．すなわち近接検出時は，IoT 機器が BLE のペリフェラルとなり，実際の通信時にはスマートフォンがペリフェラルになり動作する．これにより，スマートフォン側のバッテリ消費を抑えることが可能になる．

4.3 要件 (b)：利用者の強本人確認

利用者の強本人確認のために，利用者がスマートフォンを保持していることを確認する所有物認証とスマートフォンが備える生体認証を使用した FIDO UAF ベースの利用者認証を行う．

具体的には，IoT 機器，スマートフォン，サービスサーバに対して以下のような機能を搭載する．

IoT 機器には，スマートフォンから IoT 機器に対する物理タッチがあった場合に，FIDO サーバに，FIDO UAF ベースの利用者認証を実施するよう依頼する機能を持たせる．そして，FIDO サーバとスマートフォンの間で認証プロトコルに介在し，利用者の強本人確認を補佐する．

スマートフォンには，FIDO サーバからの認証要求に従いサービスに対応する認証器を呼び出して利用者認証を行う機能を持たせる．そして，認証結果を IoT 機器と FIDO サーバを介してサービスサーバに返す機能を持たせる．

サービスサーバには，スマートフォンの認証結果を確認する機能を持たせる．なお，攻撃者に無用な手がかりを与えないために，サービサーバがスマートフォンに返すのは，認証の最終的な可否（と認証成功時の暗号鍵）である．

4.4 要件 (c)：利用者の所在確認

利用者の所在確認のため，IoT 機器，スマートフォン，サービスサーバに対して以下のような機能を搭載する．

IoT 機器には，スマートフォンから IoT 機器に対する物理タッチがあった場合に，IoT 機器側の時刻情報を記録する機能を持たせる．そして，スマートフォンからの認証結果を FIDO サーバに中継する際には，認証結果に IoT 機器の時刻情報と IoT 機器 ID（たとえば MAC アドレス）をハッシュ値を加えて FIDO サーバに返信する．

スマートフォンには，物理タッチが発生した際に，IoT 機器から IoT 機器 ID を取得する機能を持たせる．FIDO サーバからの認証要求があった場合に，認証器を呼び出して UAF ベースの利用者認証を行い，認証結果に IoT 機器 ID とスマートフォンの時刻情報のハッシュ値を付加して，サービスサーバに返す機能を持たせる．

サービスサーバには，受け取った認証情報から，スマートフォンと IoT 機器が同じ IoT 機器 ID と時刻情報を持っていることを確認する機能を持たせる．これらが一致している場合には，IoT 機器とスマートフォンの間で確かに近接通信によって利用者認証が行われたことが分かる．結果情報には，スマートフォンで行われた利用者認証の結果も含まれていることから，IoT 機器と物理タッチしたのは利用者であるという所在確認ができる．

4.5 要件 (d)：安全な通信のオフロード

安全な通信のオフロードのために，IoT 機器経由でスマートフォンと FIDO サーバをつなぐ安全な通信路を自動的に確立する機能を持たせる．具体的には，スマートフォンと FIDO サーバをトンネリングさせるための機能をスマートフォンおよび IoT 機器に設けるとともに，FIDO サーバとスマートフォンの間に中間者がいないことをトラストチェーンで確認できるようにする．以下では，それぞれの機能について説明する．

4.5.1 トンネリング機能

通信のオフロードのために，図 4 に示したように，IoT 機器に物理タッチが行われた際には，自動的に IoT 機器を介した形でサービスサーバとスマートフォンが接続されるようにするトンネリング機能を追加する．

このために IoT 機器には，FIDO クライアントの機能を SDK を用いて組み込んだサービスごとのスマートフォンアプリである FIDO アプリケーションや FIDO クライアントに相当する機能を有する IoT ファームウェアと，IoT ファームウェアからスマートフォン内の認証器を呼び出すことができる機能を有するアクセス制御を配備する．IoT ファームウェアは，スマートフォンから利用者 ID を得て，サービスログイン要求をサービスサーバに伝える．また，FIDO サーバから与えられた UAF パケットをスマートフォンに渡すことで，スマートフォン内の生体認証機能を呼び出す．そしてスマートフォンから認証結果を受け取ると，IoT ファームウェアを経由したことを示す情報を追加して

サービスサーバに渡す．

スマートフォンには，IoT 機器を介してサービスサーバにサービスの利用要求を行うとともに，IoT 機器を介してサービスサーバからの利用者認証の実行依頼を受け付ける機能を設ける．

なお，本フレームワークの FIDO サーバは，従来の FIDO サーバと同様の以下の機能を備える．

- スマートフォンからの認証要求に付加された利用者 ID を使って，FIDO サーバに利用者の生体認証を依頼する機能
- スマートフォン側に，FIDO サーバが付加した認証ポリシ付きで認証実行要求を渡す機能
- スマートフォンの認証結果を FIDO サーバに渡す機能
- FIDO サーバから正当なユーザであるとの結果を得た場合に，その利用者向けのサービスを準備し IoT 機器経由で利用者に提示する機能．また，正当な利用者でない場合にサービスを停止する機能

4.5.2 トラストチェーンの形成

サービスサーバと IoT 機器の間においてあらかじめ確立されているトラストを，物理タッチにより一時的に利用するスマートフォンにまで広げ，これらの情報を UAF プロトコルに収めて，FIDO サーバに吸い上げ検証できるようにする．

具体的には，スマートフォンと IoT 機器の接続を行った際に，IoT 機器のトンネリング機能モジュール（IoT ファームウェア），スマートフォンのトンネリング機能モジュール，スマートフォンの FIDO クライアントのそれぞれにおいて，毎回の通信確立時の属性値（例：物理タッチした時刻，近接通信の MAC アドレス，セッション ID/鍵）を用いたハッシュチェーンを計算することによって，サービスサーバからスマートフォンへのトラストチェーンをそのつど構築する．

4.6 動作

以下では，提案フレームワークの登録および認証について説明する．

4.6.1 登録

サービスの利用登録にあたり，スマートフォンの中で公開鍵と秘密鍵の鍵ペアを生成する．そして，スマートフォンにサービスアプリケーションをインストールして，サービスアプリケーションを用いてサービスサーバに対して登録申請する．このとき，サービスサーバからサービス ID を取得する．

4.6.2 認証

IoT 機器利用時の認証動作は以下のようになる（図 4）．

(2-1) 認証要求

(2-1-1) 認証準備：スマートフォンを IoT 機器に物理的に近づけることによって，スマートフォンは IoT 機器から IoT 機器内のサービス ID を取得する．スマートフォンは，IoT 機器からのサービス ID が，サービス登録時にサービスサーバから取得したサービス ID と一致したならば，接続してよい IoT 機器であると判断する．この結果，スマートフォンと IoT 機器の間に一時的な通信路が生成され，スマートフォンとサービスサーバの間にトラストチェーンが構築される．

(2-1-2) 認証要求：IoT 機器内の IoT ファームウェアは，物理タッチによりスマートフォンから利用者 ID を得る．そして，この利用者 ID とともにサービスログイン要求をサービスサーバに伝える．サービスサーバは，これを受けてサービスの開始要求を FIDO サーバに対して行う．そして，FIDO サーバは利用者認証プロトコルを開始し，認証実行要求の UAF パケットを生成，サービスサーバに発行する．サービスサーバは，この UAF パケットを IoT ファームウェアに渡す．IoT ファームウェアは，与えられた UAF パケットを一時的に生成された通信路を用いてスマートフォンに転送する．

(2-1-3) 認証依頼：スマートフォンの ASM 内の受信機能では，送られてきた UAF パケット内で指定されている Policy を解釈する．そして，サービスサーバから求められている適切な認証器を特定し，それに対して利用者認証を依頼する．このときに，トラストチェーンを構築するための情報として，通信確立時の属性値のハッシュ値をパケットに追加する．

(2-2) 認証

(2-2-1) 認証実行：認証器は，認証実行要求を受け取ると，Policy で指定された認証方式により利用者認証を実施する．そして認証結果を IoT ファームウェアに送信する．

(2-2-2) 認証結果送信：IoT ファームウェアは，認証結果を受け取ると，IoT 機器が持つ通信確立時の属性値のハッシュ値を追加してサービスサーバに渡す．

(2-2-3) 認証確認：サービスサーバは FIDO サーバに結果を渡し，FIDO サーバはその結果を検証し，一時的に生成された通信路の正当性と利用者の強本人性が確認された場合に，アクセスのあった IoT 機器に対して，利用者向けのサービスを提供する．

5. 評価

以下では，提案手法の理論評価と実装して動作評価を行った結果を示す．理論評価では，提案手法が要件を満たしていることと，提案手法が経路攻撃に対する耐性を持っていることを検証するとともに，FIDO で規定する CTAP 仕様（Client to Authenticator Protocol）[48] との比較を行う．動作評価では，ドアロックを例にして検証を行う．

5.1 理論評価

ここでは理論評価として，要件との整合性，セキュリ

ティ，FIDO CTAP との比較を行う．

5.1.1 要件との整合性

ここでは，解決策や実装がそれぞれの要件を満たしていることを確認する．

要件 (a) 利用サービスの自動特定については，スマートフォンによる IoT 機器へのタッチの際に，IoT 機器からサービスを表す ID が与えられることによって，サービスに対応するスマートフォン上のアプリケーションを一意に特定することが可能である．

要件 (b) 利用者の強本確認については，IoT サービス利用者の生体認証登録時に公開鍵暗号により公開鍵と秘密鍵を生成し，IoT 機器利用者の本人確認に生体認証を用いた際に，その認証結果に基づきスマートフォン内の公開鍵をアクティベートしてサービスサーバに渡して検証することで実現している．

要件 (c) 利用者の所在確認は，利用者が IoT 機器の前にいたことを，IoT 機器 ID と時刻情報をスマートフォンと IoT 機器の両方でハッシュ計算した結果と，利用者認証の結果が共にあることを検証することで，利用者が IoT 機器の前にいることが確認可能になっている．遠隔から不正な操作を行った場合には，ハッシュ値が一致しない結果となり，この場合には何らかの問題があったと判断しリクエストを破棄するなどの対処を行うことができる．

要件 (d) 安全な通信のオフロードについては，IoT 機器とスマートフォン間のローカルな通信路を使用し，その通信接続を確立する際に，IoT 機器とスマートフォンのトンネリング機能モジュールにおいて，毎回属性値を用いたハッシュチェーンを計算しトラストチェーンを構築することで安全性を確保している．

実際には，4.1.4 項に示したような，さらなる様々なトラストの強化につながる情報が UAF パケットに付加されていく．

5.1.2 セキュリティ

ここでは，脅威分析によるセキュリティ評価を行う．**図 6** に提案手法で起こりうる攻撃箇所をまとめた．攻撃箇所は全部で 4 つあり，以下ではそれぞれの概要と対策についてまとめた．

① IoT 機器なりすましによるサービスサーバへの攻撃

これは，他の装置などが IoT 機器になりすまし，サービスサーバに不正アクセスする攻撃を指している．

これに対しては，IoT 機器とサービスサーバ間の接続に VPN を使うことでエンドツーエンドの安全な通信を張ることにより対処する．その際，IoT 機器の秘密鍵は TPM（Trusted Platform Module）[49] のような耐タンパ性を持つハードウェア機能を活用することで保護する．また，アクセス時には，IoT 機器とサービスサーバがお互いの公開鍵証明書を確認することで安全な VPN 通信を実現する．

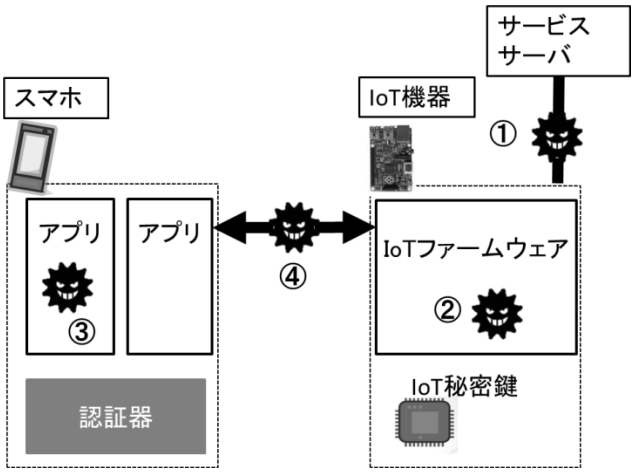

図 6 攻撃に関する検討
Fig. 6 Consideration of attack points.

本稿で想定しているような「サービスサーバと IoT 機器の関係がおおむね固定化されている IoT サービス」の場合には，あらかじめ IoT 秘密鍵に対応する公開鍵証明書をサービスサーバが，サービス秘密鍵に対応する公開鍵証明書を IoT 機器が持つという運用形態をとることができ，アクセス時にお互いが証明書を提示し照らし合わせることで正当性をチェックすることができる．これらにより，IoT 機器とサービスサーバ間で VPN（安全な暗号通信）を担保し，不正者による IoT 機器のなりすましを発生させないようにした．

② IoT 機器上のマルウェア

これは，IoT 機器にマルウェアが侵入してくる [50] 場合の攻撃を指している．攻撃には，正規利用の範囲で発生する問題と，脆弱性の悪用により発生する問題が考えられる．

前者に対する対策としては，コード署名により IoT ファームウェアの正しさをチェックすることで，正規のファームウェア以外が IoT 機器に入らないように対処する．このコード署名は，TPM 上で実行することに加え，IoT ファームウェアのブートストラップ時に毎回チェックを行う．後者は，侵入のために脆弱性を悪用したり，権限昇格のために脆弱性を悪用したりする攻撃となるが，脆弱性を有している IoT ファームウェアに対するアップデートを行うことで対処する．一般的に，しばらくすると誰かが問題に気づきパッチが作成される．そこで，いずれ問題を解決したファームウェアが作られるという前提で，問題が解決されたファームウェアを待ち，アップデートをかけ対処していくアプローチをとる．これは，ゼロデイ攻撃に対する対処と同様のアプローチにあたる．

③ スマートフォン上のマルウェア

これは，IoT サービス用アプリケーションにマルウェアが感染する場合と，他のアプリケーションや OS にマルウェアが侵入する場合の攻撃を指している．

これも，正規利用の範囲で発生する問題と，脆弱性の悪

用により発生する問題が考えられる．前者に対する対策としては，コード署名（アプリ署名）によりアプリケーションの正しさをチェックすることで，正規のアプリケーション以外がスマートフォンに入らないように対処する．後者は，侵入のために脆弱性を悪用したり，権限昇格のために脆弱性を悪用したりしてスマートフォン内の認証器を勝手に操作したり，認証器に格納されている秘密鍵を取得したりする攻撃であるが，脆弱性を有しているアプリケーションに対するアップデートを行うことで対処する．また，アプリケーションが認証器にアクセスする際に，アプリケーションの ID を認証器で確認することにより，正規の IoT サービス用アプリケーション以外からの認証器へのアクセスをブロックするとともに，秘密鍵は TrustZone 内で処理することにより外に漏れることがないように管理する．これらにより，不正者がスマートフォン内の認証器を乗っ取ったり，秘密鍵を使って認証器をエミュレートしたりするようなことができないようにした．

④ **IoT 機器とスマートフォン間の攻撃**

これは，IoT 機器とスマートフォン間の接続に介入する中間者攻撃を指している．本稿で想定している IoT サービスにおいては，サービスサーバと IoT 機器は，システム構築時に両者の関係が構築され，公開鍵証明書をお互いに持ち合うことができる．また，利用者がスマートフォン上のアプリケーションを介してサービスに対する利用者登録をする際に，アプリ署名をサービス提供側に，サービス署名をスマートフォン側に登録することで，サービスサーバとスマートフォンの関係が構築される．しかし，利用者がいつどの IoT 機器を利用するのかは固定されていないことから，IoT 機器とスマートフォンの間に関しては，あらかじめ両者の関係を設定しておくことを期待するのは現実的でない．すなわち，IoT 機器とスマートフォンは，利用者が IoT サービスを利用する時点で初めて対面する形となる．両者は安全なエンドツーエンドの接続を確立するための事前知識を有していないため，中間者攻撃が起こりうる．

この問題に対しては，「中間者が入ると必ず遅延が発生する」という事実 [51], [52] を活用し，中間者攻撃を無効化する方法を採用する．具体的には，トランザクションの際に IoT 機器とスマートフォンの双方が通信時間を計測した結果を与信情報としてサービスサーバに渡し，サービス側でその通信時間の適性（あらかじめ期待された値よりも長い通信時間を要していないか）を検証することで，スマートフォンと IoT 機器が直接つながっていることを確認できるようにした．NFC のような近接無線通信の活用を特徴とする提案手法においては，十分に近い 1 対 1 通信かつ十分に短い瞬時のレスポンスが保証される．ここに中間者による何らかの攻撃が入った場合には，ネットワークのバイパスや悪意のある処理を行うために要する時間が重畳されることになる．著者らはすでに，近接無線通信の所用時間を測定することによって中間者の存在を検知可能であることを，予備実験を通じて確認している．

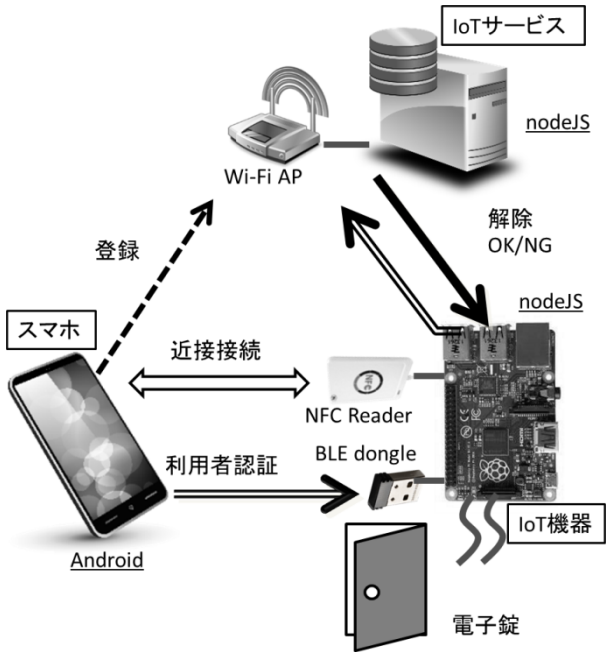

図 7 ドアロック評価環境
Fig. 7 Evaluation environment for door lock.

5.1.3 FIDO CTAP との比較

ここでは FIDO CTAP と，提案手法との比較を行う．CTAP は，PC などの端末のブラウザから多様な認証器を利用するための共通のインタフェース仕様を規定しており，PC などの FIDO Client に対して，UAF [53] に対応したスマートフォンや U2F（Universal Second Factor）[54] デバイスをつないで認証器として利用することができる．

本提案手法では，スマホに FIDO Client と認証器を保持しながら，ASM I/F を拡張することで，もう 1 つの FIDO Client に相当する IoT ファームウェアからスマートフォン内の認証器を利用できるようになっている．

CTAP を使うには，PC と認証器にあたるスマートフォンと認証器のペアリング作業が必要になり，PC にこのためのユーザインタフェースが設けられる．しかし，一般にリソースが限られている IoT 機器では，このユーザインタフェースを設けることが難しい場合もある．そこで提案手法では，認証器としてのスマートフォンから自動ペアリングができるようになっている．

5.2 動作評価

ここでは，提案手法の動作評価を行うために，ドアロックシステムを開発し評価した結果を示す．システムは図 7 に示したように，IoT 機器に代わり利用者認証を行うスマートフォン，物理的な電子錠を制御する IoT 機器，利用者認証の結果を受けドアロックの解除を行う IoT サービス（サーバ機能）から構成される．

図 8 近接接続時/利用者認証時の動作
Fig. 8 Behavior of detection of IoT and authentication.

それぞれの評価環境の構成は以下のとおりである．

IoT サービス：IoT サービスは，ノート PC 上に IoT サービスソフトウェアをインストールすることで構成した．ノート PC は Wi-Fi AP（Access Point）と接続されており，Wi-Fi AP を介してスマートフォンおよび IoT 機器からの通信を受けられるようになっている．PC 上の IoT サービスソフトウェアは，Node.js [55] を用い，その上に Web サービス実装することで，サービスサーバおよび FIDO サーバ機能を構成した．Wi-Fi AP には，スマートフォンの FIDO 機能をサービスサーバに登録する場合と IoT 機器がサービスサーバに利用者確認を行うときにアクセスを行う．

スマートフォン：利用者認証を行うスマートフォンとして Android スマートフォン（認証タイプ：指紋，OS：Android 6.0）を用い，この上に開発したソフトウェアをインストールした．

IoT 機器：ドアロックを制御する IoT 機器は，Raspberry Pi [56] をベースとして，そこに電子錠（ソレノイドロック）[57] を接続した．また，Raspberry Pi 上に Node.js を搭載し，そこから GPIO 制御を行うことで，電子錠の制御を行った．またスマートフォンとの近接検知の方法として NFC を用いた．

図 8 は，近接接続時と利用認証時の動作結果のフローを表している．また**図 9** は，図 8 の①～⑤に対する画面イメージおよびメッセージである．

① スマートフォンを IoT 機器の NFC Reader にタッチすることで近くにいることを伝え，自動ペアリングを行う（図 8・図 9 ①）．

② IoT 機器の検出を行う（図 8・図 9 ②）．

③ 通信路の生成を行う（図 8・図 9 ②→③）が行われる．

①～③までにかかる時間を測定したところ，約 2 秒であった．実際の利用者視点では，スマートフォンを IoT 機器にタッチして手元にもってくるまでに接続は完了しているため，提案手法による追加機能により遅れを感じるなどの違和感はないことを検証できた．

④ 近接接続が終わると，利用者認証の動作が始まる．これはスマートフォンに搭載されている FIDO 機能を IoT サービスから呼び出すことで，利用者認証が行われる．UAF 仕様では，認証時に表示するメッセージをサービスサーバからコントロールすることができる．そこで，利用者により確認を行うことができるようにサービスサーバからメッセージ発行を行い（「You are about to Open 3F office door. Is it OK?」というメッセージを発行），メッセージを受信したスマートフォンに表示されることを確認した．

⑤ 認証結果には，NFC タッチ時刻（2017/2/10/11:17:35），通信路で使用している BLE アドレス（6C:A7:D1:A5:82:90）を署名データに入れて IoT サービスに送信する（図 9）．IoT 機器からも接続時のデータ（NFC タッチ時刻，BLE アドレス）が報告されるので，そのデータと照合して一致すれば IoT 機器に利用 OK 命令を発効し，電子錠の解錠を行う．

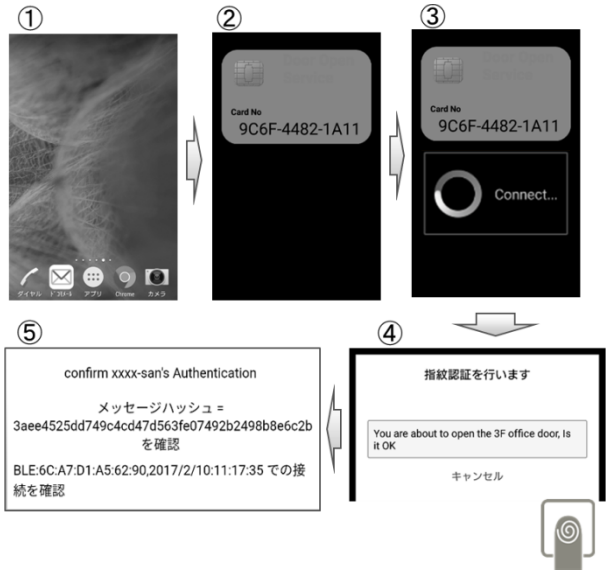

図 9 近接接続時/利用者認証時の画面/メッセージ
Fig. 9 Screen shots of detection of IoT device and authentication/Message of the data.

6. おわりに

IoT サービスが課金型に広がる中で，所有物の持ち主が

サービス利用を希望していることの認証に加えて，所有物を所持している利用者が正規の所有者であることまで確認する必要が生じている．また，IoT サービスを提供するインフラとして，サービス品質の確保が必要になる．これに対して本稿では，スマートフォンを活用し，それ持った利用者が IoT 機器に物理的にタッチする動作を，「利用サービスの自動特定（利用しようとしている IoT サービスの自動特定および起動）」と「利用者の所在確認（利用者が IoT 機器の前に所在することの確認）」の手段として使用するとともに，「利用者の強本人確認（スマートフォンを用いた生体認証よる様々な IoT サービスの認証）」をワンストップで実現可能にする IoT サービス利用者認証のためのフレームワーク手法を提案した．その際，サービスサーバと IoT 機器の間のローカルな通信路を活用し，「安全なデータオフロード（安全性を確保した形でのサービスの通信品質の維持）」に配慮した IoT サービスを提供可能にした．また，提案手法を FIDO プロトコルを活用して実装し，追加処理が 2 秒以内で完了すること，ユーザビリティに影響がないことを示した．

参考文献

[1] IEEE Internet Initiative: Towards a definition of the Internet of Things (IoT), Revision 1 – Published 27 MAY 2015, available from ⟨http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf⟩ (accessed 2017-05).

[2] 「宅配受取ロッカー」，いよいよ JR の首都圏 100 駅に設置スタート，入手先 ⟨http://www.rbbtoday.com/article/2016/05/10/141886.html⟩（参照 2017-02）．

[3] 大和　哲：データオフロードとは，ケータイ Watch，入手先 ⟨http://k-tai.watch.impress.co.jp/docs/column/keyword/477048.html⟩（参照 2017-04）．

[4] FIDO alliance, available from ⟨https://fidoalliance.org/⟩ (accessed 2017-02).

[5] 矢崎孝一，伊藤栄信，坂本拓也，二村和明：スマホ認証を用いた IoT 機器サービスの簡易利用方式，第 76 回コンピュータセキュリティ合同研究発表会 (2017).

[6] Walmart Pay, available from ⟨https://www.walmart.com/cp/walmart-pay/3205993⟩ (accessed 2017-05).

[7] Alipay, available from ⟨https://play.google.com/store/apps/details?id=com.eg.android.AlipayGphone&hl=ja⟩ (accessed 2017-05).

[8] パスワード管理の現実解，日経パソコン，2014.3.24 入手先 ⟨http://itpro.nikkeibp.co.jp/npc/npcs/pdf/140324/tokushu2.pdf⟩（参照 2017-05）．

[9] Puri, P., Singh, M.P.: A survey paper on routing in delay-tolerant networks, *Information Systems and Computer Networks* (*ISCON*) (2013).

[10] Cisco Visual Networking Index：全世界のモバイルデータトラフィックの予測，2016〜2021 年アップデート，入手先 ⟨http://www.cisco.com/c/ja_jp/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf⟩（参照 2017-05）．

[11] 西岡哲朗，木谷友哉，太田　剛，峰野博史：DTN を用いたモバイルデータ 3D オフローディング手法の提案，IPSJ 第 76 回全国大会 (2014).

[12] セキュリティトークン，入手先 ⟨https://ja.wikipedia.org/wiki/セキュリティトークン⟩（参照 2017-05）．

[13] 電子航空券，入手先 ⟨https://ja.wikipedia.org/wiki/電子航空券⟩（参照 2017-02）．

[14] OAuth 2.0, available from ⟨https://oauth.net/2/⟩ (accessed 2017-04).

[15] OpenID Connect, available from ⟨http://openid.net/connect/⟩ (accessed 2017-04).

[16] Apple pay, available from ⟨http://www.apple.com/jp/apple-pay/⟩ (accessed 2017-05).

[17] Android pay, available from ⟨https://www.android.com/intl/ja_jp/pay/⟩ (accessed 2017-05).

[18] JAL タッチ＆ゴー，入手先 ⟨https://www.jal.co.jp/k-tai/appli/touchandgo/⟩（参照 2017-05）．

[19] 生体認証 (Wikipedia)，入手先 ⟨https://ja.wikipedia.org/wiki/生体認証⟩（参照 2017-02）．

[20] 指紋，虹彩，顔認証 · · · どれが安全？スマホ生体認証の「限界」を探る：モバイル決済最前線，入手先 ⟨http://japanese.engadget.com/2017/04/13/payment/⟩（参照 2017-05）．

[21] 藤田真浩，山田眞子，西垣正勝：エンターテイメントを活用したセキュリティ強化：パスワード強化要素を組み込んだゲームの実装とその有効性，情報処理学会論文誌，Vol.57, No.12, pp.2711–2722 (2016).

[22] 宮田　健：地道に “セキュア” を積み重ねることの重要性，2008/6/24，入手先 ⟨http://www.atmarkit.co.jp/fsecurity/special/124interop2008/interop01.html⟩（参照 2017-05）．

[23] Andrew Martin, Trusted Infrastructure 101, Trusted Infrastructure Workshop 2011, June 19–23, 2011, available from ⟨https://www.cylab.cmu.edu/tiw/slides/martin-tiw101.pdf⟩ (accessed 2017-05).

[24] Qrio Smart Lock, available from ⟨https://qrio.me/smartlock/⟩ (accessed 2017-05).

[25] Ninja Lock, available from ⟨https://www.ninjalock.me/⟩ (accessed 2017-05).

[26] キーモバイルシステム「KEYMO」，入手先 ⟨http://www.miwa-lock.co.jp/tec/products/keymo/⟩（参照 2017-05）．

[27] 39 HOTELS PROJECT WITH AKERUN, available from ⟨https://akerun.com/39hotels/⟩ (accessed 2017-05).

[28] ヒルトンのキーレス・チェックインサービス DIGITAL KEY，入手先 ⟨http://hiltonhonors3.hilton.com/ja_JP/hhonors-mobile-app/digital-key.html⟩（参照 2017-05）．

[29] 宅配の再配達の削減に向けた検討の進め方について，国土交通省，平成 27 年 6 月，入手先 ⟨http://www.mlit.go.jp/common/001106424.pdf⟩（参照 2017-05）．

[30] オープン型宅配便ロッカーネットワーク，入手先 ⟨http://packcity.co.jp/vision⟩（参照 2017-05）．

[31] Amazon Locker, available from ⟨https://www.amazon.com/b?node=6442600011⟩ (accessed 2017-05).

[32] THE NEXT GENERATION OF ON-DEMAND CAR RENTAL TECHNOLOGY, available from ⟨http://www.getkeyfree.com/car-rental⟩ (accessed 2017-05).

[33] これがほんとのキーレスだ！ボルボがスマホでロックする完全キーレスを採用，2016/04/20，入手先 ⟨https://carnny.jp/2232⟩（参照 2017-05）．

[34] iVIPER, available from ⟨http://www.kato-denki.com/products/viper/new_iviper/⟩ (accessed 2017-05).

[35] IPA，オンライン本人認証方式の実態調査報告書，2014.8，入手先 ⟨https://www.ipa.go.jp/files/000040778.pdf⟩.

[36] Digital twin, available from ⟨http://iot-jp.com/iotsummary/iottech/digital-twin（デジタルツイン）/.html⟩ (accessed 2017-05).

[37] AWS IoT の Device Shadow，入手先 ⟨http://docs.aws.amazon.com/ja_jp/iot/latest/developerguide/

iot-thing-shadows.html⟩ （参照 2017-05）.
[38] W3C Web of Things (WoT) Architecture, available from ⟨https://w3c.github.io/wot/architecture/wot-architecture.html⟩ (accessed 2017-05).
[39] テレマティクス等を活用した安全運転促進保険等による道路交通の安全，第 9 回自動車関連情報の利活用に関する将来ビジョン検討会（テーマ I），国土交通省，入手先 ⟨https://www.mlit.go.jp/common/001061957.pdf⟩（参照 2017-05）.
[40] 損保ジャパン日本興亜，スマートフォンを活用した「テレマティクス保険」の開発，2017.3.27，入手先 ⟨http://www.sjnk.co.jp/~/media/SJNK/files/news/2016/20170327_1.pdf⟩ （参照 2017-05）.
[41] デロイトトーマツコンサルティング合同会社，トレンドから読み解く保険業界の新しい将来像デジタル時代の保険業界の disruption，入手先 ⟨https://www2.deloitte.com/content/dam/Deloitte/jp/Documents/financial-services/ins/jp-ins-digital-disruption-070116.pdf⟩ （参照 2017-05）.
[42] FIDO AppID and Facet Specification v1.0, available from ⟨https://fidoalliance.org/specs/fido-u2f-v1.0-ps-20141009/fido-appid-and-facets-ps-20141009.html⟩ (accessed 2017-02).
[43] FIDO UAF Policy, available from ⟨https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-protocol-v1.0-ps-20141208.html#processing-rules-for-the-server-policy⟩ (accessed 2017-02).
[44] KeyID, available from ⟨https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-protocol-v1.0-ps-20141208.html#keyid-typedef⟩ (accessed 2017-05).
[45] FIDO Security Reference, available from ⟨http://fidoalliance.org/specs/fido-uaf-v1.0-rd-20140209/fido-security-ref-v1.0-rd-20140209.pdf⟩ (accessed 2017-02).
[46] Near Field Communication (NFC) Technology and Measurements, available from ⟨https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1ma182/1MA182_5E_NFC_WHITE_PAPER.pdf⟩ (accessed 2017-05).
[47] Specification of the Bluetooth System, available from ⟨file:///C:/Users/nimura/Downloads/Core_v4.2.pdf⟩.
[48] FIDO 2.0: Client To Authenticator Protocol, available from ⟨https://fidoalliance.org/specs/fido-v2.0-rd-20161004/fido-client-to-authenticator-protocol-v2.0-rd-20161004.html⟩ (accessed 2017-04).
[49] Trusted Platform Module, available from ⟨https://trustedcomputinggroup.org/⟩ (accessed 2017-02).
[50] ネットワークカメラや家庭用ルータ等の IoT 機器は利用前に必ずパスワードの変更，入手先 ⟨https://www.ipa.go.jp/security/anshin/mgdayori20161125.html⟩（参照 2017-05）.
[51] Vallivaara, V.A., Sailio, M. and Halunen, K.: Detecting man-in-the-middle attacks on non-mobile systems, *CODASPY '14, Proc. 4th ACM Conference on Data and Application Security and Privacy*, pp.131–134 (2014).
[52] 佐藤隼人，宮田純子，加島宜雄：RTT の分散を考慮した中間者攻撃検知手法の提案，IEICE-116, No.71, pp.23–28 (2016).
[53] FIDO UAF Architectural Overview, available from ⟨https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-overview-v1.0-ps-20141208.pdf⟩ (accessed 2017-05).
[54] Universal 2nd Factor (U2F) Overview, available from ⟨https://fidoalliance.org/specs/fido-u2f-v1.1-id-20160915/fido-u2f-overview-v1.1-id-20160915.html⟩.
[55] node.js, available from ⟨https://nodejs.org/ja/⟩ (accessed 2017-04).
[56] RASPBERRY PI, available from ⟨https://www.raspberrypi.org/⟩ (accessed 2017-04).
[57] ソレノイドロック，入手先 ⟨https://www.takigen.co.jp/jp/contents/eyesearch/summary.do?name=L-002⟩（参照 2017-04）.

二村 和明 （学生会員）

1969 年生．1994 年東京電機大学大学院情報通信工学専攻修士課程修了，同年富士通株式会社入社．1997 年富士通研究所勤務．データセキュリティの研究開発に従事．2016 年より静岡大学創造科学技術大学院博士課程．

矢崎 孝一

1996 年大阪府立大学大学院工学研究科電子工学専攻修士課程修了．同年 (株) 富士通研究所入社．データセキュリティの研究開発に従事．

伊藤 栄信

1968 年生．1993 年大阪府立大学大学院工学研究科数理工学専攻博士前期課程修了．同年（株）富士通研究所入社．データセキュリティの研究開発に従事．

坂本 拓也

1973 年生．1997 年神戸大学大学院自然科学研究科情報知能工学専攻博士前期課程修了，同年富士通株式会社入社．2000 年から富士通研究所勤務．データセキュリティの研究開発に従事．

西垣 正勝 （正会員）

1990 年静岡大学工学部光電機械工学科卒業．1995 年同大学大学院博士課程修了．日本学術振興会特別研究員（PD）を経て，1996 年静岡大学・情報助手．同講師，助教授の後，2010 年より同創造科学技術大学院教授．博士（工学）．情報セキュリティ全般，特にヒューマニクスセキュリティ，メディアセキュリティ，ネットワークセキュリティ等に関する研究に従事．2013～2014 年情報処理学会コンピュータセキュリティ研究会主査．2015～2016 年電子情報通信学会バイオメトリクス研究専門委員会委員長．2016 年より日本セキュリティマネジメント学会常任理事．本会フェロー．