Vol.2013-DPS-154 No.3
Vol.2013-CSEC-60 No.3
2013/3/14

近傍デバイスを利用したコンテキスト依存セキュリティシステムの提案

大久保 隆夫[1,a)]　海野 雪絵[1,b)]　野田 敏達[1,c)]　金谷 延幸[1,d)]

概要：位置情報センサを持つ携帯端末の普及により，時間や場所限定のサービスやアクセス権など，利用者のコンテキストに依存したサービス提供が注目されつつある．しかし，サービス提供側がコンテキスト改竄などの不正行為を防止したい場合，耐タンパー性等を持たない端末ではその安全性を保証できない．本稿では，信頼できるデバイスを利用者の近傍に配置することを想定して，そのデバイスにセンサーデータの取得とコンテキスト生成を代行させることで，デバイスとペアリングした任意の端末 – サービス間でコンテキスト依存サービスを安全に利用できるシステムを提案する．提案システムでは，コンテキストの含まれる情報をアクセスチケットに利用することで，Kerberos 等ではできなかったコンテキスト依存のセキュリティ制御が可能になっている．

1. はじめに

GPS や加速度センサを搭載したスマートフォンの急速な普及に伴い，これらの端末のセンサー情報に基づいた利用者のコンテキスト (文脈) に応じたサービス (コンテキスト依存サービス) が注目されつつある．アーティストとカフェの期間限定コラボレーションで，期間中に店の写真を撮影し，twitter でツイートした利用者から抽選で景品が当たる，など時間や位置を限定して提供されるサービスも増えている[*1]．また，サービスの価値が高いなどの場合や，また職場向けのデータ共有などのように，利用者がコンテキストを偽造や改竄するなどの「ずる」を防止しなければならない場合もある．一般には，耐タンパー性のハードウェアを搭載した端末で，通信やデータの保証を行う解決手段があるが，端末の種類が制限されてしまうという問題がある．

本稿では，耐タンパーハードウェアとセンサーを持つデバイスを利用者の近傍に配置することを前提として，これらのデバイスが採取したセンサー情報を利用者のコンテキスト構築と保証に用いることで，任意の利用者端末とサービス提供者間で，安全なコンテキスト依存サービスの提供を可能にするシステムを提案する．また，富士通研究所の携帯セキュリティアプライアンス技術を用いて提案システムを実現した例を示し，システムの安全性について述べる．

[1] (株) 富士通研究所
Fujitsu Laboratories limited, 4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki 211–8588, Japan
[a)] okubo@jp.fujitsu.com
[b)] unno.yukie@jp.fujitsu.com
[c)] noda.bintatsu@jp.fujitsu.com
[d)] kanaya.nobuyuki@jp.fujitsu.com
[*1] http://megweb.jp/?p=info&id=17044

2. 背景と課題

2.1 コンテキスト依存サービスとその脅威

携帯端末のセンサを用いてコンテキスト依存サービスを行う一般的な構成を**図 1** に示す．利用者の携帯端末では，端末のセンサーから得られる情報をもとにコンテキスト情報を構築する．利用者は構築されたコンテキスト情報を提供を受けたいサービスを提供するサーバに送信する．サーバ側では，受信したコンテキスト情報が，サービス提供の条件を満たしているかを検査し，満たしていれば，サービスを提供する．サービス提供の条件となるコンテキスト情報の例を**表 1** に示す．

コンテキスト依存サービスには，利用者を信頼し，特に厳密なコンテキスト検証を行わないモデルもあるが，下記のような場合にはコンテキスト改竄やなりすましにより被害が発生することが想定されるため，サービスへのアクセス許可を与える前に，コンテキストの完全性などの検証を行う必要性が生じる．

- サービス自体が高い (金銭的な) 価値を持つもの
- 利用者を限定してサービスを行う必要がある
- 個人情報や著作権管理情報など，不用意に第三者に漏洩してはならない情報資産を扱う

コンテキスト依存サービスに対する脅威としては，図 1 に示す通り次に挙げるものが想定される．

表 1 コンテキスト情報に基づく条件記述の例
Table 1 An Example of Context Condition Description.

10 時から 12 時まで	$(time, (10,0) \leq t \leq (12,0))$
北緯 35.679575 以上 35.683305 以下	$(location, (35.679575 \leq lat \leq 35.683305)$
かつ東経 139.765098 以上 139.769518 以下	$\wedge(139.765098 \leq long \leq 139.769518))$
無線 LAN アクセスポイント 74.125.228.35 圏内	$(network, (gw = (74.125.228.35)))$

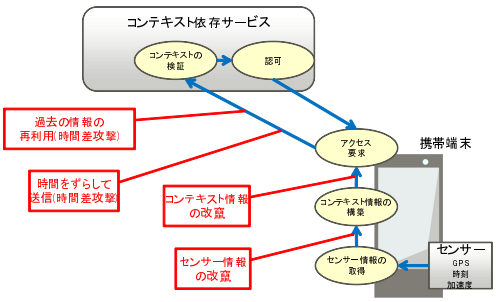

図 1 携帯端末センサを用いたコンテキスト依存サービス
Fig. 1 Context service using sensors on mobile devices.

- センサー情報の改竄
- コンテキスト情報の改竄
- 時間をずらして送信する時間差攻撃
- 過去のコンテキスト情報の再利用 (リプレイ) 攻撃

上記の攻撃は，第三者だけでなく，利用者自身が不正に利益を得る目的で行う可能性があり，サービスによっては利用者自身の不正も排除する必要がある．生成した第三者認証局などでコンテキスト情報に電子署名やタイムスタンプを付与する手法では，コンテキスト情報生成前の情報の改竄や，リプレイ攻撃には対応できない．

3. 提案システム

本稿では，上記の問題を解決するため，信頼できるデバイス (Trusted Devide:TD) が利用者の近傍にあることを前提とし，TD と利用者の端末が最初に安全な通信を確立することで，センサー情報の採取，完全性確保を TD に移譲することで，完全性の確保された安全なコンテキスト依存サービスの提供を任意の端末で可能とするシステムを提案する．

本提案では，TD の存在を前提とするが，TD は以下の性質を持つものとする．

- 耐タンパーハードウェアを搭載し，鍵等を安全に格納できる
- 外部からのインタフェースが制限されており，アプリケーションやセンサ情報の機密性や完全性が確保されている

TD は画面出力のみを持ち，利用者の入力インタフェースを持たない，単純なデバイスでよい．本提案で想定する TD の形態は次の 2 種類である．

(1) 利用者が常に携行するタイプ (腕時計など)．後述の PSER はこちらを想定している
(2) 特定の地点に固定されており，動かせないもの

前者と後者では，TD の利用形態が異なる．前者は，携行している利用者個人のみが TD を用いているのに対し，後者では，不特定多数の利用者が 1 つの TD を利用することになる．後者の想定する利用シーンとしては，特定のイベント会場に TD が設置され，イベント参加者全員が TD を利用するなどの利用が想定される．

上記の前提において，利用者は自身の端末 (スマートフォン，タブレット，PC 等) と TD をペアリングさせることにより，サービス提供を受ける際のセンサー情報の取得やコンテキスト情報の作成，完全性保証のための電子署名，タイムスタンプを TD に移譲することが可能になる．

安全性，利便性を保証するためには，TD と端末間のペアリングは下記の条件を満たす必要がある．

- 近傍にある端末としかペアリングできない
- ペアリングのための通信の安全性が保証されており，なりすましや盗聴に対して保護されている
- TD は任意の端末とのペアリングが可能

上記の条件を満たす具体的なペアリング手法については，文献 [8],[7] などに基づく既知の技術を利用するものとし，本稿では対象外とする．例えば，近傍に限定された通信としては Wi-fi や Bluetooth，NFC[5] などがある．

提案方式では，端末の代わりに TD のセンサー情報を用い，構築した情報に署名する．署名した結果を端末が受けとり，コンテキスト依存サービスに利用することで，デバイスとペアリングした任意の端末において安全にコンテキスト依存サービスの利用が可能になる．

また，ペアリングの際に TD，端末間で交換する乱数情報を利用し，リプレイ，時間差攻撃の防止を行う．最初に TD による端末の認証 (ペアリング) を行い，TD 上に表示されたワンタイムの乱数を端末/利用者に通知する．利用者が TD 発行の乱数と，利用者側の乱数 (または秘密文字列) を合わせ端末に入力し，TD に送信する．TD では，コンテキスト情報への署名時に上記乱数 (TD 乱数+利用者乱数) を埋めこむ．次に端末がサービスにアクセス要求する際に，利用者に上記乱数を再入力させ，サービス側で検証を行う．

Vol.2013-DPS-154 No.3
Vol.2013-CSEC-60 No.3
2013/3/14

4. PSER による実現

PSER は，筆者らが提案している，端末からセキュリティ機能を分離した個人用のセキュリティアプライアンスである [8]． PSER は，端末で使用する暗号鍵，暗号機能，ウイルス検知機能などを安全に管理し，これらの機能を端末に提供したり，端末にかわって実行する．PSER では，デバイス自体が耐タンパー技術や API の制限により，セキュリティ機能の専用デバイス化されており，デバイスの安全な機能提供が保証される．PSER の管理さえ適切に行っておけば，本アプライアンスを利用するどんな OS のどんな端末からでも，適切なセキュリティ機能を安全に利用することが可能となる．

PSER は利用者に特化した常時携行型のデバイスであるため，本提案方式の携行型 TD として用いることができる．また，固定の TD として，複数利用者を扱うことも可能である．

PSER を用いた実現方法には大別して，アプライアンス型と，プロキシ型の 2 つがある．以下ではそれぞれの構成と手順について述べる．

4.1 アプライアンス型

アプライアンス型の提案システム構成を**図 2** に示す．手順を**図 3** および次に示す．

(1) TD がワンタイムの乱数 R_{td} を発行
$R_{td} = random()$

(2) TD が画面上に R_{td} を表示

(3) 端末がサービスからコンテキスト条件 C_{cond} とサービスの証明書 S_{cert} を取得
$S_{cert}, C_{cond} = getContextCondition()$

(4) サービスの正当性を証明書検証により行う
$verifyService(S_cert)$

(5) 端末が TD にサービスアクセスに必要なコンテキスト情報を要求
その際 R_{td}，および利用者自身の乱数 R_{user} の連結 R_{pair}，S_{cert} を入力，端末が TD に送信する
$R_{user} = random()$
$R_{pair} = concat(R_{td}, R_{user})$
$requestContext(R_{pair}, S_{cert})$
(a) TD は入力された値と R_{td} を比較，一致していればペアリングが成立
$verifyRand(R_{td}, getRtd(R_{pair}))$
(b) TD は TD のセンサーから情報 S_{td} を収集 (位置，時間など)
(c) TD は S_{td} に基づきコンテキスト情報 C_{td} を構築，その際 TD の時刻情報 T_{td} を含める
$C_{td} = context(S_{td}, T_{td})$

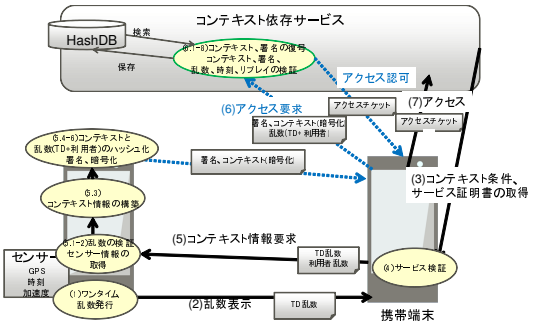

図 2 提案システム構成 (アプライアンス型)
Fig. 2 Architecture of the proposed system (appliance type).

(d) TD は R_{pair} のハッシュ値 H_{rand} を取得
$H_{rand} = hash(R_{pair})$
(e) TD の署名鍵 K_{td} で C_{td} と H_{rand} に署名を付与 $(Sign, Cxt)$
$Sign, Cxt = sign(K_{td}, concat(C_{td}, H_{rand}))$
(f) $Sign, Cxt$ をサービスの公開鍵 S_{pk} で暗号化
$C_{sign}, C_{cxt} = encrypt(Sign, Cxt, S_{pk} = S_{cert}.getpk())$

(6) TD は端末に S_{cxt} 返信

(7) 端末はサービスにアクセスを要求する．その際，R_{pair} と S_{cxt} を送信する
$requestService(R_{pair}, C_{sign}, C_{cxt})$ サービス側では以下の検証を行う
(a) C_{sign}，C_{cxt} をサービスの秘密鍵 S_{sk} で復号
$Sign, Cxt = decrypt(C_{sign}, C_{cxt}, S_{sk})$
(b) コンテキストはサービスの要求 C_{cond} を満たすか
$verifyContext(Cxt, C_{cond})$
(c) 署名が正しいか
$verifySignature(Sign)$
(d) 端末から送信された乱数のハッシュ値を計算し，S_{cxt} に含まれる H_{rand} とが一致するか
$compare(hash(R_{pair}), getHashRand(Cxt))$
(e) H_{rand} が既に認可されたものの中にないか (HashDB を参照)
$ifContains(H_{rand}, HashDB)$
(f) コンテキスト中の時刻情報が，サービスの現在時刻から一定時間以上経過していないか
$verifyTime(getTime(Cxt), getCurrentTime())$
(g) 認可後，H_{rand} を HashDB に保存
検証が正しければアクセスを認可し，端末にアクセスチケット A_{ticket} を送信する

(8) サービスにアクセスする．その際端末はアクセスチケットを送信する $access(A_{ticket})$

4.2 プロキシ型

プロキシ型の提案システム構成を**図 4** に示す．手順を

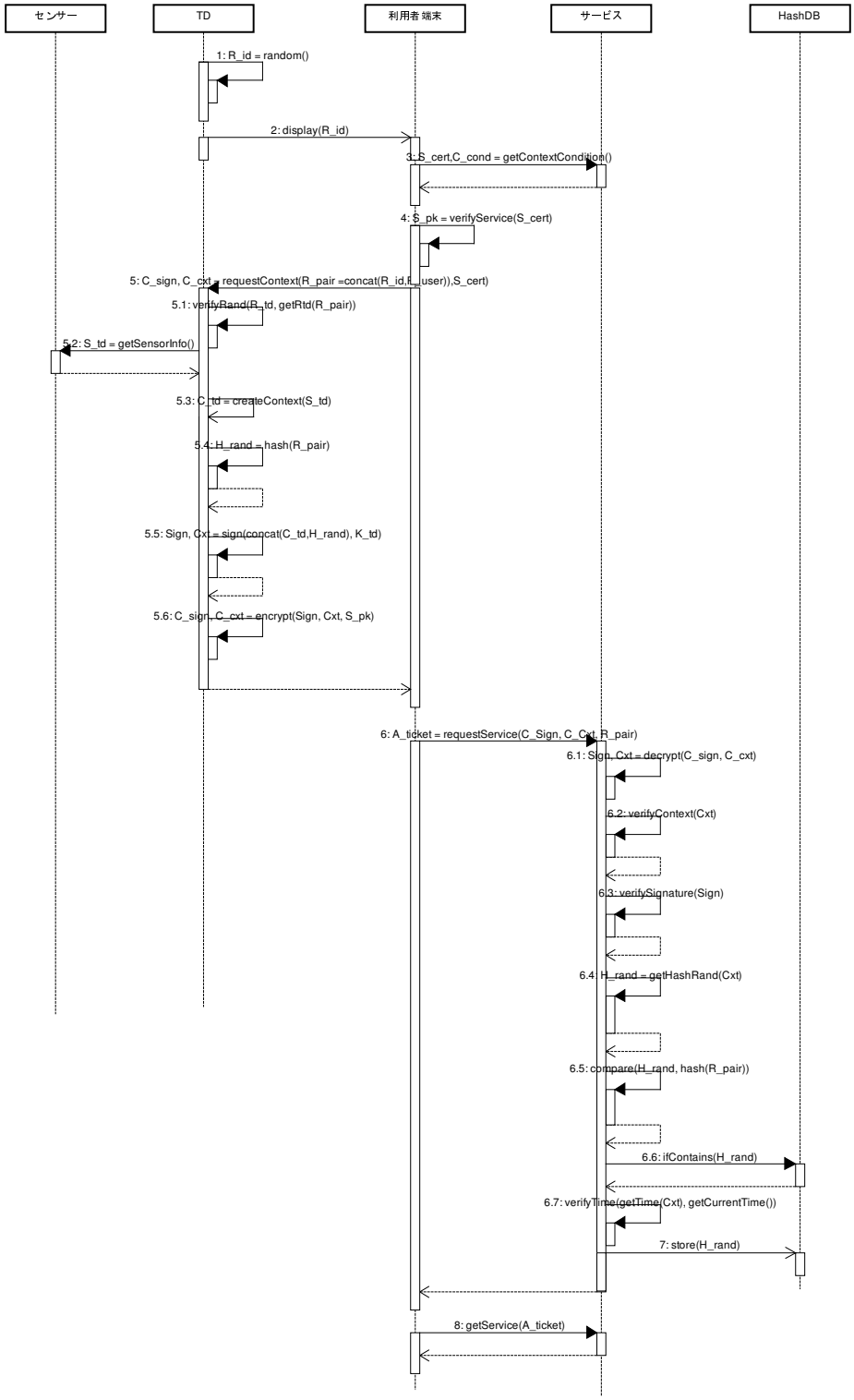

図 3 手順 (アプライアンス型)
Fig. 3 Sequence of the proposed system (appliance type).

図 5 および次に示す.

(1) TD がワンタイムの乱数 R_{td} を発行
$R_{td} = random()$

(2) TD が画面上に R_{td} を表示

(3) 端末がサービスからコンテキスト条件 C_{cond} とサービスの証明書 S_{cert} を取得
$S_{cert}, C_{cond} = getContextCondition()$

(4) サービスの正当性を証明書検証により行う
$verifyService(S_cert)$

(5) 端末が TD にサービスアクセスに必要なコンテキスト情報を要求
その際 R_{td}, および利用者自身の乱数 R_{user} の連結

R_{pair}，S_{cert}, サービスアドレス S_{addr} を入力，端末が TD に送信する
$R_{user} = random()$
$R_{pair} = concat(R_{td}, R_{user})$
$requestContext(R_{pair}, S_{cert}, S_{addr})$
(a) TD は入力された値と R_{td} を比較，一致していればペアリングが成立
$verifyRand(R_{td}, getRtd(R_{pair}))$
(b) TD は TD のセンサーから情報 S_{td} を収集 (位置，時間など)
(c) TD は S_{td} に基づきコンテキスト情報 C_{td} を構築，その際 TD の時刻情報 T_{td} を含める
$C_{td} = context(S_{td}, T_{td})$
(d) TD は C_{td} と R_{pair} のハッシュ値 H_{rand} を取得
$H_{rand} = hash(R_{pair})$
(e) TD の署名鍵 K_{td} で C_td と H_{rand} に署名を付与 $(Sign, Cxt)$
$S_{cxt} = sign(K_{td}, concat(C_{td}, H_{rand}))$
(f) $Sign, Cxt$ をサービスの公開鍵 S_{pk} で暗号化
$C_{sign}, C_{cxt} = encrypt(Sign, Cxt, S_{pk} = S_{cert}.getpk())$
(g) TD は S_{addr} が示すサービスにアクセスを要求する．その際，R_{pair}，C_{sign}，C_{cxt} を送信する
$requestService(R_{pair}, C_{sign}, C_{cxt})$ サービス側では以下の検証を行う
(i) C_{sign}，C_{cxt} をサービスの秘密鍵 S_{sk} で復号
$Sign, Cxt = decrypt(C_{sign}, C_{cxt}, S_{sk})$
(ii) コンテキストはサービスの要求 C_{cond} を満たすか
$verifyContext(Cxt, C_{cond})$
(iii) 署名が正しいか
$verifySignature(Sign)$
(iv) 端末から送信された乱数のハッシュ値を計算し，S_{cxt} に含まれる H_{rand} とが一致するか
$compare(hash(R_{pair}), getHashRand(Cxt))$
(v) H_{rand} が既に認可されたものの中にないか (HashDB を参照)
$ifContains(H_{rand}, HashDB)$
(vi) コンテキスト中の時刻情報が，サービスの現在時刻から一定時間以上経過していないか
$verifyTime(getTime(Cxt), getCurrentTime())$
(vii) 認可後，H_{rand} を HashDB に保存
(viii) 検証が正しければアクセスを認可し，TD にアクセスチケット A_{ticket} を送信する
検証が正しければアクセスを認可し，TD にアクセスチケット A_{ticket} を送信する
TD から端末にアクセスチケット A_{ticket} を送信する
(6) 端末からサービスにアクセスする．その際端末はアクセスチケット A_{ticket} を送信する
$access(A_{ticket})$

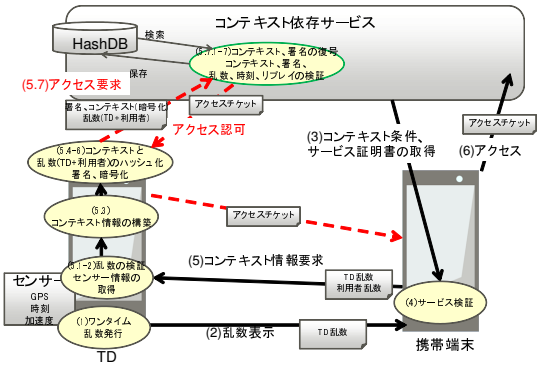

図 4 提案システム構成 (プロキシ型)
Fig. 4 Architecture of the proposed system (proxy type).

5. 議論

5.1 効果

本提案方式の利点は，次の 2 点である．

- 任意の端末で利用可能
  従来，耐タンパーハードウェア等によりセンサー情報の保証が困難だった端末を用いても，デバイスを利用して，センサー情報に基づいた安全なコンテキスト依存サービスの提供が可能になる．
- 利用者識別，認証が不要
  本提案方式は利用者の認証を行わずに，コンテキストのみによる検証を可能としている．したがって，必要のない限り，コンテキストを保持している本人認証をする必要がなく，トラッキング等によるプライバシー問題の回避や，利用者認証実装の省力化が可能になる．逆に，利用者の識別認証が機能要件上必須になる場合には，利用者識別認証と併用してもよい．

5.2 リプレイ攻撃に対する安全性

本提案方式では，従来の署名，タイムスタンプでは検出できなかった，端末サイドでのセンサーデータの偽造検出を実現し，かつ従来の署名，タイムスタンプでは検出できなかった，時間差，リプレイ攻撃の検出も可能になっている．

- 取得したコンテキスト情報を，正規利用者のサービス利用後に正規利用者/第三者が利用してアクセス要求した場合
  一度認可された乱数 R_{pair} は，HashDB に記録されるため，検証時に検出され再利用できない．
- 取得したコンテキスト情報を，正規利用者のサービス利用前に第三者が窃取してアクセス要求した場合
  第三者は TD の乱数 R_{td} と正規利用者 (コンテキスト

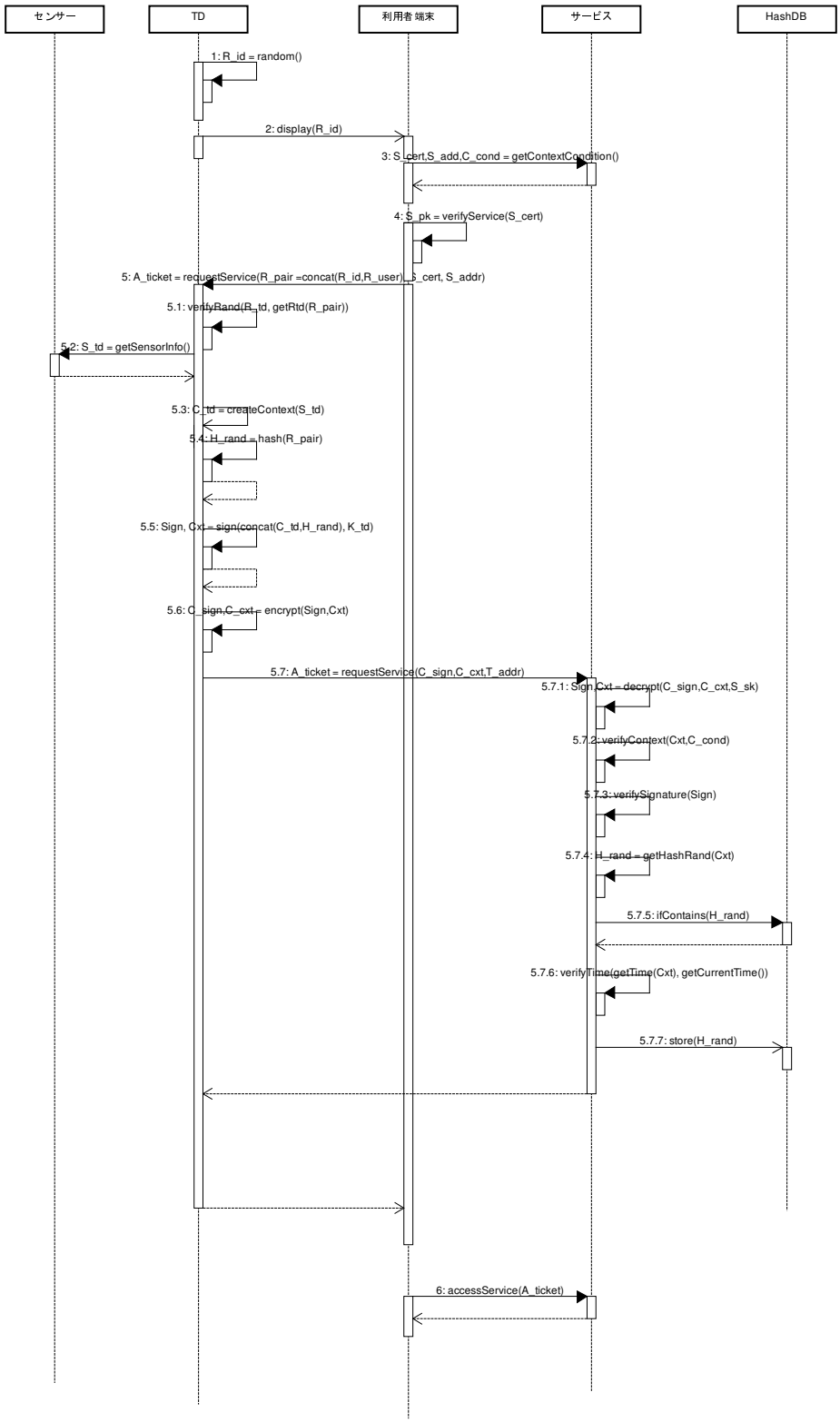

図 5 手順 (プロキシ型)
Fig. 5 Sequence of the proposed system (proxy type).

情報請求を行った利用者) の乱数 R_{user} の双方を知らなければ，認可されない．

- コンテキスト情報作成から時間をおいて利用する場合検証時に，サービスはコンテキスト情報中の時刻情報を抽出し，現在時刻との差を検証する．一定時間以上経過したものは認可されない．

5.3 コンテキストに関する議論

本提案方式が適用できるコンテキストの記述能力は，3節に挙げた利用形態によって異なる．(1)TD を利用者が携行する場合，TD に対応する利用者は単独のため，過去のコンテキストやセンサー情報を蓄積することができ，この情報を用いて要求するコンテキストの記述が可能になる．しかし，(2) 固定した場所に置かれた TD では，TD と利用者が 1 対多のため，利用者または端末の識別が必要になる．前節の適用例において，ペアリングの際に利用者が TD に入力している乱数の代わりに，固定の識別 ID を入力させ，TD 側で識別 ID に応じたコンテキストの蓄積を行う必要がある．しかし，この方式では，1 つの TD 内で収集可能なコンテキストしか蓄積，利用できない．仮に利用者が別の場所にいたという履歴が必要である場合，(1) 携行型ではこの情報を TD 内に蓄積し利用できるが，(2) では別の TD の情報が必要になってしまう．また，複数の利用者の情報を ID が収集，追跡できてしまうプライバシーの問題もある．

5.4 コンテキストなりすまし以外の脅威

本提案は，コンテキストのなりすましに注目した対策であるが，実現にあたり，コンテキストなりすまし以外の脅威も想定し，実現手法を示す．コンテキストなりすまし以外の脅威としては以下のものがある．

- 【脅威 1】サービスによるコンテキストの悪用
  第三者が，サービスのふりをして利用者からコンテキスト，署名と乱数を取得し，別のサービスへ転送してサービス要求を行う．
- 【脅威 2】利用者によるコンテキストの第三者への譲渡
  TD 近傍のコンテキストを満たす利用者がコンテキスト情報取得後，乱数とコンテキスト情報を第三者に譲渡する．

前節の実現では，最初に端末がサービスを証明書検証により認証し，かつサービスの公開鍵で暗号化しているため，そのコンテキスト情報は他のサービスへは転用できない．従って，第三者のサービスなりすましによる悪用 (【脅威 1】) を防ぐことができる．

利用者によるコンテキストの第三者への譲渡は，前節のプロキシ型では，サービスへのアクセス要求が TD からしか発信できないため，譲渡されたコンテキストの二次利用 (【脅威 2】) を防ぐことができる．一方，アプライアンス型では二次利用を防止できないため，一般的にはプロキシ型の方がより安全と言える．ただしプロキシ型は不特定多数利用では，サービスと特定の TD 間にトラフィックが集中するため，注意が必要である．

本提案は，TD 側の安全性は保証するが，利用者端末側は耐タンパーなどの技術を必須としていないため，利用者端末がマルウェアに感染するなどして遠隔操作され，その場にいない第三者がサービスを享受できてしまう脅威が存在する．具体的には，次のような脅威が想定される．

- 【脅威 3】利用者が気付かずに，端末がサービスの中継点として利用される
- 【脅威 4】利用者が意識的に第三者に協力して，サービスの中継点となる
- 【脅威 5】端末から TD，利用者乱数などの入力情報が漏洩する

【脅威 3】については，端末-TD 間のペアリングを利用者に意識させるようにする (ペアリング時の端末からの乱数入力は，機械的に行う方法もあるが，利用者による手入力にするなど) ことにより，意図しないサービス利用を利用者に気付かせることが期待できる．一方，【脅威 4】については利用者自身が協力しているためそれは期待できない．【脅威 2】同様，DRM 技術などを用いてサービスの二次利用を防止する策が別途必要になると考えられる．一方，【脅威 5】に対しては，端末内に乱数情報を保持しないこと以外に有効な策はなく，端末がこのようなマルウェアには感染しないことを前提に置くしかない．

6. 関連研究

事前にアクセスチケットを取得する認証方式としては，Kerberos[4] がある．Kerberos は，得られたアクセスチケットをもとに複数のサービスにシングルサインオンする認証方式であり，アクセスチケットに基づきサービスにアクセスする本提案方式に類似している．しかし Kerberos のアクセスチケットは利用者のコンテキストに関する情報を持たないため，サービス提供側で，アクセスチケットの完全性や正当性の検証は可能だが，コンテキストの妥当性 (サービスが要求する条件を満足するか) の検証は困難である．したがって，完全性や正当性の検証に加え，コンテキストの検証も行える本提案方式の方が Kerberos よりもコンテキスト依存サービスにおけるセキュリティに適している．

コンテキストに基づいたアクセス制御など，初めからコンテキストに着目したセキュリティ制御手法もいくつか提案されている [3][6][1][2]．Covington らは，各コンポーネントから得られるセンサー情報に基づいたコンテキストによるアクセス制御を提案している [3] が，各情報を安全に通信するために，コンポーネントの認証を行っている．一方，本提案では，利用者の端末自体の認証自体は必須ではない．

7. おわりに

本稿では，PSER などの安全なデバイスを利用者の近傍に配置することで，コンテキスト依存サービスの安全な利用を可能にするシステムを提案した．提案システムを用いれば，利用者の端末の限定や利用者認証を行わなくても，

コンテキスト情報のみに基づいたセキュリティ制御を実現できる．また，端末の利用者自身によるコンテキストの偽造や改竄，リプレイ攻撃等も検出可能である．

提案システムが扱えるコンテキスト情報は，現在は位置情報や時刻情報などのセンサー情報の単純な組み合わせを想定している．具体的な自由度について，特に 5 節で述べた履歴を含むコンテキスト情報の扱いや，ロジックや抽象的なコンテキスト概念の扱いについては今後の検討課題である．また，端末の遠隔操作による不正なサービス利用に対する検討も今後，進める必要がある．

参考文献

[1] Al-Rabiaah, S. and Al-Muhtadi, J.: ConSec: Context-Aware Security Framework for Smart Spaces, *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pp. 580 –584 (2012).
[2] Chung, M., Choi, J., Yang, S. and Rhyoo, S.-K.: Context-Aware Security Services in DAA Security Model, *Advanced Language Processing and Web Information Technology, 2008. ALPIT '08. International Conference on*, pp. 424 –429 (2008).
[3] Covington, M. J., Long, W., Srinivasan, S., Dev, A. K., Ahamad, M. and Abowd, G. D.: Securing context-aware applications using environment roles, *Proceedings of the sixth ACM symposium on Access control models and technologies*, SACMAT '01, New York, NY, USA, ACM, pp. 10–20 (2001).
[4] Kohl, J. and Neuman, C.: *RFC1510 The Kerberos Network Authentication Service (V5)*, Internet Engineering Task Force (1993).
[5] NFCForum: White Paper. The Keys to Truly Interoperable Communications (2006).
[6] Rahnama, H., Jamshidi, S., Johns, S. and Shepard, A.: CAMPUS: context aware mobile platform for uniformed security, *Proceedings of the 13th international conference on Ubiquitous computing*, UbiComp '11, New York, NY, USA, ACM, pp. 489–490 (2011).
[7] 海野雪絵，野田敏達，大久保隆夫，金谷延幸：Web アプリから利用者端末内情報へのコンテキストアウェアなアクセス制御手法の提案，情報処理学会第 60 回コンピュータセキュリティ研究会研究報告 (2013).
[8] 野田敏達，海野雪絵，金谷延幸：個人用セキュリティアプライアンスの提案，情報処理学会第 60 回コンピュータセキュリティ研究会研究報告 (2013).