リンク不能性を実現し大規模RFIDシステムに適用可能なID照合プロトコル

野原, 康伸
九州大学大学院

井上, 創造
九州大学大学院

馬場, 謙介
九州大学大学院

安浦, 寛人
九州大学大学院

出版情報：Proc. of the 2005 Symposium on Cryptography and Information Security(SCIS2005) || 3 || p1567-1572, pp.1567-1572, 2005-01. 電子情報通信学会 情報セキュリティ研究会
バージョン：
権利関係：

SCIS 2005 The 2005 Symposium on Cryptography and Information Security
Maiko Kobe, Japan, Jan.25-28, 2005
The Institute of Electronics, Information and Communication Engineers

リンク不能性を実現し大規模RFIDシステムに適用可能なID照合プロトコル
Unlinkable ID Matching Protocol for Large-scale RFID Systems

野原　康伸 * 井上　創造 * 馬場　謙介 * 安浦　寛人 *
Yasunobu NOHARA Sozo INOUE Kensuke BABA Hiroto YASUURA

あらまし 近年急速に，IC カードや RFID タグといったデバイスの普及が進んできている．しかし，第三者がユーザに無断でデバイスの ID を読み取ることにより，ユーザの履歴情報を収集し，個人の行動を追跡できてしまうという問題がある．この問題を解決する方法として，デバイスの出力を毎回可変にする Randomized Hash Lock 方式や ID 照合方式がある．しかしながら，これらの方式はデバイス数 N に対して，1 回の ID 解決毎にサーバ側でハッシュ計算を $O(N)$ 回必要とし，大規模システムに適用するには問題があった．本稿で提案する方式は，ID を分割し，各部分 ID についてそれぞれ ID 解決を行うことにより，サーバ側で必要なハッシュ計算を $O(\log N)$ 回に抑えることができる．

キーワード RFID セキュリティ, リンク不能性 (リンク不能度), K 段 ID 照合方式

1　はじめに

近年急速に，IC カードや RFID タグといったデバイス (**ID デバイス**) が普及してきている．しかし，第三者がユーザに無断でデバイスの ID を読み取ることにより，ユーザの履歴情報を収集し，個人の行動を追跡できてしまうという問題がある．

これに対し，第三者やサービス提供者には複数のアクセスが同一ユーザによるものか判定できないというリンク不能性の概念 [1][4] がある．第三者に対するリンク不能性を実現する方法として，デバイスの出力をハッシュ関数を用いて毎回可変にする Randomized Hash Lock 方式 [2] や Extended Hash-chain 方式 [3]，ID 照合方式 [4] がある．しかしながら，これらの方式はデバイス数 N に対して，サーバ側で $O(N)$ 回のハッシュ関数の計算を 1 回の ID 解決毎に必要とし，デバイス数 N の大きな大規模システムに適用するには問題があった．

本稿では，第三者に対するリンク不能性を実現し，大規模システムにも適用可能な ID 照合方式 (**K 段 ID 照合方式**) を提案する．提案方法は，ID を K 分割し，各部分 ID についてそれぞれ ID 解決を行うことにより，1 回の ID 解決毎にサーバ側で必要なハッシュ関数の計算を $O(\log N)$ 回に抑えることができる．このため，大規模システムにも適用することが可能である．

本稿の構成は以下の通りである．2 章でリンク不能性とその定量化手法，関連研究について述べる．3 章で提案する K 段 ID 照合方式について述べ，4 章で提案方法の評価を行う．最後に 5 章で本稿をまとめる．

* 九州大学大学院, 〒 816-8580 福岡県春日市春日公園 6-1, Kyushu University, 6-1 Kasuga-koen, Kasuga-shi Fukuoka, Japan, {nohara,sozo,baba,yasuura}@c.csce.kyushu-u.ac.jp

2　リンク不能性

近年，急速に ID デバイスの普及が進みつつあるが，ID デバイスによる消費者プライバシの侵害に対して懸念が集まってきている．ID デバイスによるプライバシ問題として，ID 追跡によるロケーションプライバシ問題がある [2]．この問題は，第三者が様々な場所でユーザに無断でデバイスの ID を読み，ID を元にユーザの行動履歴を関連付け (リンク) し，ユーザの行動が追跡されてしまうというものである．

ID 追跡問題を解決するためには，第三者が自由にデバイスの出力を読めたとしても，ユーザの行動履歴をリンクできないというリンク不能性の概念が重要となる．

2.1　リンク不能性の定義

ISO/IEC 15408 において，リンク不能性 (Unlinkability) は，ユーザが複数の資源あるいはサービスを使用するとき，他人がそれらを一つにリンクできないようにして使用できることを保証する性質と定義されている [1]．我々は文献 [4] において，リンク不能性の概念を一つの ID デバイスで複数のサービスを受けられるというマルチサービス環境にまで広げ，以下のように定義した．

ユーザ A の持つ ID デバイスから主体 X が取得した n 番目の情報 (ID 情報や利用履歴を含む) を I^n_{AX} としたときに，主体 X が I^n_{AX} と I^m_{AX}(ただし，$m \neq n$) の送信元が同一のユーザによるものであると判定できない場合，

ユーザ A の情報は，主体 X に対してリンク不能性を有するという．また，主体 X と主体 Y が協力して，それぞれが持つ情報を持ち寄った場合に，I^n_{AX} と I^k_{AY} の送信元が同一のユーザによるものであると判定できない場合，**ユーザ A の情報は，主体 X と主体 Y 間に対してリンク不能性を有する**という．

以下，システムのリンク不能性の比較を行うため，リンク不能性の定量化について議論する．

2.2 リンク不能性の定量化 — リンク不能度

本節では，N 個の ID デバイスが存在するシステムにおけるリンク不能性の定量化：**リンク不能度**(Degree of unlinkability) について述べる．まず，以下のようなゲームを考える (図 1 参照)．

デバイス X, Y は，全 ID デバイス N 個の中から重なりなくランダムに選ばれた 2 つの ID デバイスである．R は 1 ビットの乱数発生器であり，出力 b として 0 若しくは 1 を出力する．セレクタは，b の値によって，デバイス X の時刻 t_1 における出力 ($b = 0$)，またはデバイス Y の時刻 t_2 における出力 ($b = 1$) を選択して出力する．攻撃者 A は，デバイス X, Y に任意の入力 (それぞれ，I_1, I_2 とする) を加えることができるが，デバイス X の時刻 t_0 における出力 (O_1) とセレクタの出力 (O_2) のみ観測可能であり，X, Y, b がどのような値であるかは分からない．攻撃者 A は，デバイスの入出力結果 I_1, I_2, O_1, O_2 に基づいて b の出力を当てる．攻撃者の答えを b' とする．

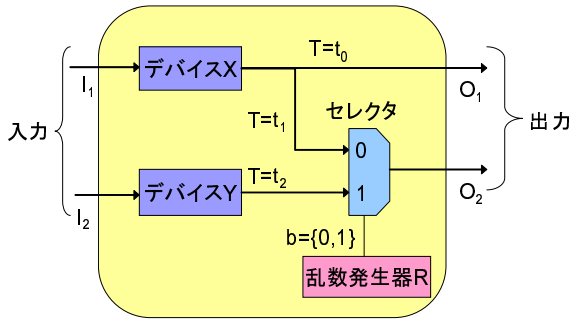

図 1: リンク不能度ゲーム

ここで，攻撃者 A の正解率 $Pr(b' \leftarrow A, b = b')$ を用いて、A の利得 $Advantage(A)$ を (1) 式のように定義する．

$$Advantage(A) = |Pr(b' \leftarrow A, b = b') - \frac{1}{2}| \qquad (1)$$

さらに，攻撃者 A に対するシステムのリンク不能度 $d(A)$ を以下のように定義する．

$$\begin{aligned} d(A) &= 1 - \frac{Advantage(A)}{MAX(Advantage)} \\ &= 1 - 2 \cdot |Pr(b' \leftarrow A, b = b') - \frac{1}{2}| \qquad (2) \end{aligned}$$

攻撃者が全てのデバイスの出力を完全に識別することができる場合は，$Advantage(A) = \frac{1}{2}$ であり，$d(A) = 0$ となる．また，攻撃者がデバイスの出力を全く識別することができない場合は，$Advantage(A) = 0$ であり，$d(A) = 1$ となる．$Advantage(A)$ が小さいほど，攻撃者はデバイスの出力を識別できないといえ，リンク不能度 $d(A)$ は高くなる．$0 \leq d(A) \leq 1$ である．

$d(A) = 1$ の場合を特に，**システムは攻撃者 A に対してリンク不能性を有する**という．

2.3 関連研究

第三者 (Third Party) に対するリンク不能性を実現する方式，すなわち $d(TP) = 1$ であるような ID 保護方式として，ハッシュ関数を用いた Randomized Hash Lock 方式 [2] や Extended Hash-chain 方式 [3]，ID 照合方式 [4] がある．

ハッシュ関数を用いた ID 保護方式は，ID デバイスに実装する回路が暗号回路よりも回路規模の小さなハッシュ回路で済む．このため，デバイスコストの制約が厳しい RFID タグにも適した方式である．

2.3.1 Randomized Hash Lock 方式

ID デバイス i は id_i を，サーバは全 ID デバイスの ID の集合 $ID = \{id_i\}(1 \leq i \leq N)$ を保持しておく．ID デバイス i は以下の手順で，ID 情報をサーバに送る．

STEP1: ID デバイス i は，乱数 R を生成し，サーバに $H = H(R||id_i)$ と R を送る[1][2].

STEP2: サーバは，$H(R||id_i)$ を ID に含まれる全ての $id_i(1 \leq i \leq N)$ について計算する．ID デバイスから送られてきた H と一致した $H(R||id_i)$ に対応する id_i がデバイスの ID となる．

2.3.2 Extended Hash-chain 方式

フォワードセキュア性という，デバイス内の秘密情報が取り出されたとしても，過去の ID 情報からデバイスの追跡ができないという性質を有した ID 保護方式である．

ID デバイス i は id_i と乱数の種 cs_1 と認証回数 k を，サーバは全 ID デバイスの ID の集合 $ID = \{id_i\}(1 \leq i \leq N)$ と cs_1 を保持しておく．ID デバイス i は以下の手順で，ID 情報をサーバに送る．

STEP1: ID デバイス i はサーバに $H = H(id_i||cs_k)$ と k を送る．ID デバイス i は，$cs_{k+1} \leftarrow G(cs_k), k \leftarrow k+1$ とし[3]，cs_k は破棄する．

STEP2: サーバは，$cs_k = G^{k-1}(cs_1)$ を計算し，$H(id_i||cs_k)$ を ID に含まれる全ての id_i について計算する．ID デバイスから送られてきた H と一致する $H(id_i||cs_k)$ に対応する id_i がデバイスの ID となる．

---

[1] $H(X)$ は，X のハッシュ値を表す．
[2] $X||Y$ は，X と Y の結合を表す．
[3] $G(X)$ は，X のハッシュ値を表す．ただし，$H(X) \neq G(X)$.

2.3.3 ID 照合方式

マルチサービス環境において，サービス提供者毎に使用する ID を変えることにより，サービス提供者が結託したとしても，ユーザの行動履歴をリンクできない (サービス提供者間に対するリンク不能性を有する) という ID 保護方式である．また，ID 解決だけではなく，相互認証も同時に行う方式である．

まず，ユーザとサービス提供者に割り当てられる情報について説明する (図 2 参照)．

発行者は，PID(Personal IDentifier) と呼ばれる各ユーザに固有の長い ID を ID デバイスに格納し，ユーザに配布する．PID の一部分のビット列のことを subPID とよび，この subPID が各サービス提供者に割り当てられる．

ユーザ i の PID(PID_i) のうち，サービス提供者 j に割り当てられた subPID(sid_{ij}) は，$sid_{ij} = f(PID_i, ID_j)$ で与えられる．ここで，ID_j はサービス提供者 j を特定するための ID であり，f は PID_j から ID_j に対応するアドレスの sid_{ij} を取り出す関数である．

subPID は，ある同一のサービス提供者 j において重複せず，唯一となるように割り当てられる．すなわち，任意の $i \neq i'$ 及び j について，$sid_{ij} \neq sid_{i'j}$ である．また，それぞれの subPID は，サービス提供者ごとに異なる．すなわち，任意の i 及び $j \neq j'$ について，$sid_{ij} \neq sid_{ij'}$ である．

サービス提供者 j に対して subPID のリスト $SID_j = \{sid_{1j}, sid_{2j}, \cdots, sid_{nj}\}$ が割り当てられる．

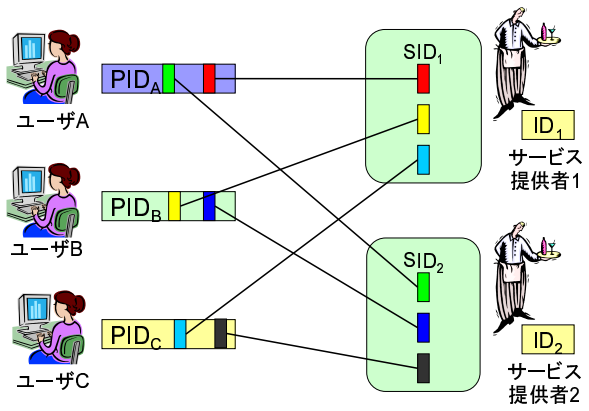

図 2: ユーザとサービス提供者が保有する情報

ユーザとサービス提供者は，以下の ID 照合プロトコルにより，ID 情報の交換と相互認証を行う (図 3 参照)．その後，ID デバイスとサーバ間でサービスが行なわれる．

STEP1: ID デバイス i は，サービス提供者のサーバに認証要求を出す

STEP2: サーバ j は，自分のサービス番号 ID_j と乱数 R_s を ID デバイスに対して送信する．

STEP3: ID デバイス i は，ID_j と自分の持つ PID_i から，当該のサービス提供者 j に対応した $sid_{ij} = f(PID_i, ID_j)$ を取り出す．また乱数 R_u を生成する．サービス提供者 j のサーバには，$H_u = H(R_s||R_u||sid_{ij})$ と R_u を送信する．

STEP4: サービス提供者 j のサーバは，$H(R_s||R_u||sid_{ij})$ の計算を SID_j に含まれる全ての sid_{ij} について計算する．ID デバイスから送られてきた H_u と一致する $H(R_s||R_u||sid_{ij})$ に対応する sid_{ij} がデバイスの ID となる．

STEP5: サービス提供者 j のサーバは，sid_{ij} と R_u をハッシュ化した $H_s = H(sid_{ij}||R_u)$ を ID デバイス i に送信する．

STEP6: ID デバイス i は，サーバから送られてきた H_s が $H(sid_{ij}||R_u)$ と一致するか検証する．

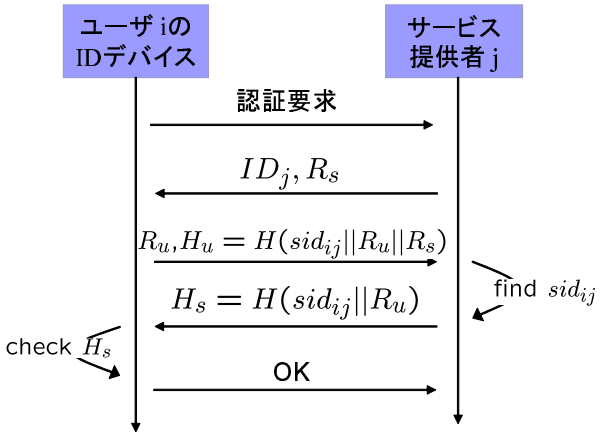

図 3: ID 照合プロトコル

2.3.4 関連研究の比較と問題点

ハッシュ関数を使用した上記 3 つの ID 保護方式を比較すると表 1 のようになる．

	Randomized Hash Lock 方式	Extended Hash-chain 方式	ID 照合方式
サーバでのハッシュ回数	$O(N)$	$O(N)$	$O(N)$
デバイスの構成	ハッシュ +乱数生成	ハッシュ +RAM	ハッシュ +乱数生成
フォワードセキュア性	×	○	×
サービス間リンク不能性	×	×	○
相互認証	×	×	○

表 1: ハッシュ関数を利用する ID 保護方式の比較

3 つの ID 保護方式を比べると，フォワードセキュア性やサービス提供者間のリンク不能性の有無などの違いはあるが，サーバ側で ID 解決のために，全てのデバイ

スの $id_i(1 \leq i \leq N)$ についてハッシュの計算 $H(id_i||R)$ を行う点で共通する．すなわち，サーバで 1 回の ID 解決に必要なハッシュ関数の使用回数は $O(N)$ である．このため，N が増えるとサーバの負担が増大する．

あらかじめ，サーバ側でデバイスの生成する可能性のある全て乱数 R の値 (M 個とする) に対応した $H(ID_i||R)$ を計算した表を作っておけば，ID 解決は表から R と $H(ID_i||R)$ に対応した ID_i を検索すれば済むため，サーバへの負荷は減少する．しかし，$N \times M$ という莫大なメモリー空間を必要とする問題がある．

よって，ハッシュ関数を用いる既存の方式は，デバイス数 N の多い大規模システムにおいて適用することがサーバの負担から考えて容易ではなかった．

3 提案方法 — K 段 ID 照合方式

本章では，我々が提案する K 段 ID 照合について述べる．まず，計算量削減のための 3 つのアイデアについて述べ，ID の生成方法とプロトコルについて説明する．

3.1 計算量削減のためのアイデア

サーバでの計算量削減のための 1 つ目のアイデアは，グルーピングである．まず，各デバイスにそれぞれグループ ID を割り当てて，デバイスを α 個ごとのグループに分けておく．ID 交換時に，デバイスがハッシュ値と合わせて，グループ ID もサーバに送ることで，ID の検索範囲を α 個に縮小し，サーバでのハッシュ関数の計算量を α 回に削減する．しかし，生のまま送られるグループ ID によって，第三者に対するリンク不能度が悪化する．

そこで，2 つ目のアイデアは，グループ ID のハッシュ化である．グループ ID を，既存方法 [4] の ID と同様に乱数と一緒にハッシュ化して送れば，第三者はグループ ID を取り出すことができなくなり，第三者に対するリンク不能度は悪化しなくなる．サーバでのハッシュの計算量は，1 つ目のアイデアの場合よりもグループ ID の数: $\frac{N}{\alpha}$ 個分増えた $\frac{N}{\alpha} + \alpha$ 回になってしまうが，既存方法の計算量 N 回よりも，計算量は減少している．

3 つ目のアイデアは，上記 2 つのアイデアの再帰的適用である．“グループ ID 自体を新たなグループ ID を使ってグルーピングし，デバイスが新たなグループ ID を乱数と一緒にハッシュ化して送る” ということを繰り返し適用していくことで更なる計算量の削減を狙うものである．

この 3 つのアイデアを元に K 段 ID 照合方式は，ID の生成方法に工夫を行い，その性質を利用することで ID 照合で必要なサーバの計算量を削減する．

3.2 ID の生成方法と割当て

深さ K のラベル付き木 (labeled tree) を考える (図 4 参照)．木は N 個の葉を持ち，各葉が各デバイス $i(1 \leq i \leq N)$ に対応している．ある同じ親を持つ子達には，それぞれユニークなラベルが付けられる．根から葉に向かって進んで行った場合の各ラベルを集めたものが，その葉の持つ ID(sid_{ij}) であり (e.g. a2bX@)，各デバイス i に付与される．以下，デバイス i の $k(1 \leq k \leq S_i)$ 番目のラベルを $sid_{ij}[k]$ と表記する．ただし，S_i はデバイス i の葉の深さであり，$1 \leq S_i \leq K$ である．サービス提供者 j はラベル付き木を木構造のまま保存しておく．

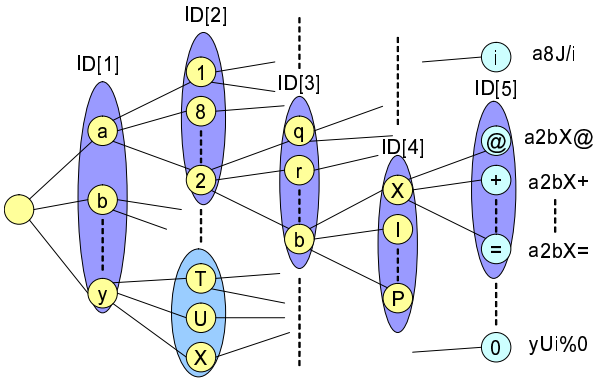

図 4: 提案手法における ID 生成方法

3.3 K 段 ID 照合プロトコル

K 段 ID 照合プロトコルは，通常の ID 照合プロトコル (2.3.3 節参照) の STEP3,4 を以下のように変更する．

STEP3: ID デバイス i は，ID_j と自分の持つ PID_i から，当該のサービス提供者に対応した $sid_{ij} = f(PID_i, ID_j)$ を取り出す．また乱数 R_u を生成する．デバイス i は，$(R_u, H_u[1], H_u[2], \cdots, H_u[K])$ をサービス提供者 j のサーバに送る．ただし，$H_u[k] = H(R_s||R_u||sid_{ij}[k])$ $(1 \leq k \leq S_i)$ であり，$S_i < K$ の場合，$H_u[l] =$ 乱数 R_l $(S_i + 1 \leq l \leq K)$ とする[4]．

STEP4: サービス提供者 j のサーバは，以下の手順を実行する

STEP4-1: $X \leftarrow$ ラベル付き木の根．$k \leftarrow 1$.

STEP4-2: X の子に付けられた各ラベル L について，$H(R_s||R_u||L)$ を計算し，送られてきた $H_u[k]$ と一致するものを探す．$X \leftarrow$ 一致したラベルを持つ子．

STEP4-3: X が葉ならば STEP5 へ．そうでなければ $k \leftarrow k + 1$ として STEP4-2 に戻る．

4 提案手法の評価

本章では K 段 ID 照合方式における

1. サーバでのハッシュ計算の使用回数

[4] 葉の深さの違いによってデバイスが特定され，第三者に対するリンク不能性が悪化しないように，乱数をパディングして送る．

2. 第三者，ユーザに対するリンク不能度

について議論する．

4.1 サーバでのハッシュ関数の使用回数

本節では，1 回の ID 解決毎にサーバで必要なハッシュ関数の使用回数について議論する．

サーバにおけるハッシュ関数の使用回数は，ラベル付き木の構造

1. 葉の数 N
2. 木の深さ K
3. 各節点から出る枝の数

に依存する．そこで，木の深さ K と，各接点から出る枝の数について決定する．

4.1.1 各接点から出る枝の数

本稿では，各接点から出る枝の数が全て等しく同じ

$$\alpha = N^{\frac{1}{K}} \tag{3}$$

であるとする．すなわち，ラベル付き木が完全 α 分木 (complete α-tree) であるとする．

このとき，K 段 ID 照合における 1 回の ID 解決には，ID の探索範囲が α 個の ID 解決を K 回繰り返すことになる．よって，サーバで必要なハッシュ関数の使用回数 $g_K(N)$ は，(4) 式のようになる．

$$g_K(N) = KN^{\frac{1}{K}} \tag{4}$$

なお，ラベル付き木を完全 α 分木にすることが，ハッシュ関数の使用回数を最小化することになるかは未証明である．

4.1.2 K の決定

木の深さ K を決定するに当たって，以下の二つの要素を考慮する必要がある．

1. リンク不能度
2. コストの最小化

まず，リンク不能度について考える．$\alpha_1 = A$ とすると，あるデバイスと ID がほとんど同じ ($sid_{ij}[K]$ のみ異なる) デバイスが $A-1$ 個存在することになる．これらのデバイスは，同一のグループ内のデバイスであれば，$A-1$ 人にまで絞り込みを行うことができるため，リンク不能度に影響を与える．$k=1$ 以外の α_k についても同様である．よって，α として，ある閾値 T_α を超えるように設定を行う必要がある．すなわち，$\alpha = N^{\frac{1}{K}} \geq T_\alpha$ を満たすように $K \leq \log_{T_\alpha} N$ とする必要がある．なお，T_α とリンク不能度 $d(user)$ の関係については，次節で詳しく議論する．

それでは，コストの最小化について考える．サーバでのハッシュ関数の使用回数 $g_K(N) = KN^{\frac{1}{K}}$ をコストとすると，以下の最適化問題が得られる．

$$\min_K KN^{\frac{1}{K}} \ (s.t\ K \leq \log_{T_\alpha} N)$$

これを解くと，$K = \log_{T_\alpha} N$ のときに $g_K(N) = KN^{\frac{1}{K}}$ は最小値 $\log_{T_\alpha} N$ をとる．すなわち，サーバでのハッシュ関数の計算回数は，$O(\log N)$ 回となる．既存の方式 [2][3][4] の計算量 $O(N)$ と比べて，提案方式は大幅に計算量が減少しており，大規模なシステムにも適用可能であることが分かる．なお，このときのデバイスで必要なハッシュ関数の使用回数は，$K = \log_{T_\alpha} N$ 回であり，既存方式 [2][3][4] の 1 回に比べて増加している．

サーバにおけるハッシュ関数の使用回数 $g_K(N) = KN^{\frac{1}{K}}$ と ID デバイスにおけるハッシュ関数の使用回数 K の間にはトレードオフの関係がある．よって，現実のシステムに用いる K を決定するには，$g_K(N)$ と K に重み係数 c_1, c_2 をそれぞれ掛けた和をコストとした以下の最適化問題を解く必要があると考えられる．

$$\min_K c_1 \cdot KN^{\frac{1}{K}} + c_2 \cdot K \quad (s.t.\ K \leq \log_{T_\alpha} N) \tag{5}$$

現実のシステムに応じた c_1, c_2 を求めての K の導出は今後の課題である．

4.2 K 段 ID 照合におけるリンク不能度

本節では，K 段 ID 照合における第三者とユーザに対するリンク不能度 $d(TP), d(user)$ を導出する．

K 段 ID 照合における攻撃者が設定可能な入力 I_1, I_2 は，

- $I_1 = (ID_{j1}, R_{s1})$
- $I_2 = (ID_{j2}, R_{s2})$

である．本稿では，攻撃者は I_1, I_2 とも同じ値 (ID_j,R_s) を送るものとする．このときの攻撃者が観測可能な出力は以下のようになる．

- $O_1 = (R_{u1}, H_{u1}[1], H_{u1}[2], \cdots, H_{u1}[K])$
- $O_2 = (R_{u2}, H_{u2}[1], H_{u2}[2], \cdots, H_{u2}[K])$

なお，以下の議論では $2^L \gg N$ が成り立っているとする．ただし，L は部分 ID の長さである．

4.2.1 $d(TP)$ の導出

攻撃者が第三者の場合を考える (A=TP)．攻撃者は全 N 個の ID のうち 1 個も ID を持っていない．そして，ハッシュ関数 $H(X)$ の性質，すなわち “$H(X)$ の入力長 L の多項式時間で出力 $H(X)$ から入力 X を求めることはで

きない” という性質から，攻撃者は O_1, O_2 からまったく情報を得ることができない．よって，次式が成り立つ．

$$d(TP) = 1 \tag{6}$$

4.2.2 $d(user)$ の導出

攻撃者がユーザの場合 ($A = user$) を考える．攻撃者は，自己の持つ ID(sid_{ij}) を用いて以下の方針で b' を決定するものとする．

1. 自己の持つ sid_{ij} を用いて $m = 1, 2$ のそれぞれの場合における $H(sid_{ij}[k], R_{um}, R_s)$ $(k = 1, \cdots, K)$ を計算し，出力 O_m の $H_{um}[k]$ と比較する．
2. $H(sid_{ij}[k], R_{um}, R_s)$ と $H_{um}[k]$ が $k = 1, \cdots, \beta$ まで一致した場合，$C_m = \beta$ とする．そうでない場合，$C_m = 0$ とする．
3. $C_1 = C_2$ の場合 $b' = 0$ とし，そうでない場合 $b' = 1$ とする．

上記の攻撃方針を採用した場合，$b \neq b'$ となるのは，$C_1 = C_2$ かつ $b = 1$ の場合だけである．よって，攻撃者の正解率 $Pr = Pr(b' \leftarrow user, b = b')$ は，

$$\begin{aligned} Pr &= 1 - \frac{1}{2}\sum_{i=0}^{K-1}\left(\frac{\alpha^i(\alpha-1)}{N}\right)\left(\frac{\alpha^i(\alpha-1)-1}{N-1}\right) \\ &= 1 - \frac{\alpha-1}{2N(N-1)}\sum_{i=0}^{K-1}\left((\alpha-1)\alpha^{2i} - \alpha^i\right) \\ &= \frac{1}{2} + \frac{N+1}{N(\alpha+1)} \end{aligned}$$

となる．よって，(7),(8) 式が成り立つ．

$$Advantage(user) = |Pr - \frac{1}{2}| = \frac{N+1}{N(\alpha+1)} \tag{7}$$

$$d(user) = 1 - \frac{2(N+1)}{N(\alpha+1)} \tag{8}$$

既存の ID 照合方式 ($K = 1$ 段 ID 照合方式) の場合の $Advantage(user) = \frac{1}{N}$ に比べて，K 段 ID 照合の $Advantage(user)$ は，$\frac{N+1}{\alpha+1} = \frac{N+1}{\sqrt[K]{N}+1}$ 倍大きくなっている．すなわち，K 段 ID 照合ではユーザに対するリンク不能度 $d(user)$ が小さくなっていることが分かる．

ユーザに対するリンク不能度 $d(user)$ をある一定の閾値 T_d 以上に保つためには (8) 式より，

$$T_\alpha = \frac{2(N+1)}{N(1-T_d)} - 1 \tag{9}$$

とすればよい．

5 おわりに

本稿では，ID を K 個に分割し，各部分 ID について ID 解決を行う K 段 ID 照合方式を提案した．提案手法は，サーバにおける 1 回の ID 解決に必要なハッシュ関数の計算量を $O(\log N)$ に抑えることができ，大規模システムにも適用可能である．今後は，ID の長さと認証・リンク不能性の関係について詳細に検討したい．また，ID デバイスとサーバの総コストを現実のモデルに当てはめたときにも，サーバにおけるハッシュ関数の計算量を $O(\log N)$ にできるのか検討を行っていきたい．

謝辞

本研究は，平成 14-18 年度科学研究費補助金学術創成研究・課題番号 14GS0218 および，平成 15-16 年度科学研究費補助金若手研究・課題番号 15700100 によるものである．

参考文献

[1] “ISO/IEC 15408 - INTERNATIONAL STANDARD Information technology - Security techniques - Evaluation criteria for IT security - Part2: Security functional requirements”, Dec.1999.

[2] Stephan A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels,“Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems”, International Conference on Security in Pervasive Computing 2003, LNCS 2802, pp.201-212, 2004.

[3] 大久保 美也子, 鈴木 幸太郎, 木下 真吾,“Hash-Chain Based Forward-Secure Privacy Protection Scheme for Low-Cost RFID”, SCIS2004 論文集, 2004 年 1 月

[4] 野原 康伸, 井上 創造, 安浦 寛人,“マルチサービス環境に適したリンク不能性を実現する ID 管理方法”, CSS2004 論文集, pp.139-144, 2004 年 10 月