社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－DPS－130 (15)
2007－CSEC－ 36 (15)
2007／3／1

小型端末を利用した匿名性を持つ遭遇情報保証技術の提案

堺拓郎 内山彰 中村嘉隆 東野輝夫
大阪大学 大学院情報科学研究科

本稿では，ユーザが小型携帯端末を保持して移動する状況を想定し，匿名性を保持しながら他者との遭遇情報を保証するための技術を提案する．位置情報を利用したサービスを提供する場合，他者との遭遇情報を用いることで，提供する位置情報の信頼性を高めることができる．一方で各ユーザのプライバシーの問題や小型携帯端末の性能が低いという問題も考慮する必要がある．そこでハッシュ関数を用いることで，単体で個人の特定ができないようにした遭遇情報を交換して，遭遇者同士が互いに遭遇した事実の保証を行う方式を考案した．提案方式は匿名性に加えてリンク不能性を満たすため，あるユーザから送信された複数の遭遇情報を基に，それらの遭遇情報が同一のユーザから送信されたことを特定できないようにしている．ハッシュ関数の有用性および提案方式の実現可能性を調べるため，評価実験を行い，小型端末 MOTE 上にハッシュ関数 SHA-1 を実装し，その計算時間を求めた．また電力消費量について公開鍵暗号と比較を行い，約 1/340 に電力消費量を抑えられることが分かった．

Design of a Sensor-based Certification Technique for Anonymous Encountering Information

Takurou Sakai, Akira Uchiyama, Yoshitaka Nakamura and Teruo Higashino
Graduate School of Information Science and Technology, Osaka University

In this paper, we propose a certification technique for anonymous encountering information. In the proposed technique, each mobile user holds a small sensor with a short range wireless device such as RFID tags, and fixed sensors are sparsely distributed in the target area. Each user exchanges and keeps encountering information obtained from the other users or fixed sensors. Encountering information is helpful to enhance reliability of location information for location-based services. However, we need consider privacy of users carefully. For this purpose, anonymity and unlinkability of encountering information is guaranteed by using a hash function. We have implemented the hash function on MOTE and confirmed efficiency of the hash function on computation time and power consumption through the experiments.

1 はじめに

無線通信ネットワークの進歩に伴い，無線端末を利用して個人の位置情報を取得し，応用することが重要視されてきている．人や車を対象とする位置情報を応用したシステムとして，遊園地での迷子発見システム [1]，米国における E-911 サービス [2] などが実際に運用されている．これらのシステムでは，各ユーザの位置情報のみを利用している．これに加えて「誰が」「誰と」「いつ」「どこで」遭遇したかを示す情報（**遭遇情報**）を確実に残すことができれば，トレーサビリティの観点からさまざまな応用が考えられる．

遭遇情報を利用した探索救助システム Cenwits[3] では，原生自然環境保全地区内を移動するユーザが小型端末を保持し，ユーザ同士が遭遇した場合に，遭遇情報を交換・蓄積する．対象地区にはいくつかの固定端末が配置されており，ユーザ端末に蓄積された遭遇情報は固定端末を介してサーバに送信される．そして，ユーザが遭難・負傷した場合には，サーバに集められた遭遇情報を基にユーザの遭難位置を推定することで早期救出を支援する．Cenwits では全てのユーザが信頼できる関係にあることを仮定しており，遭遇情報は平文で送信されるため，信頼できないユーザが存在する環境では，第三者に個人情報を知られてしまう可能性がある．また，交換される遭遇情報は容易に偽証可能であり，証拠性はない．

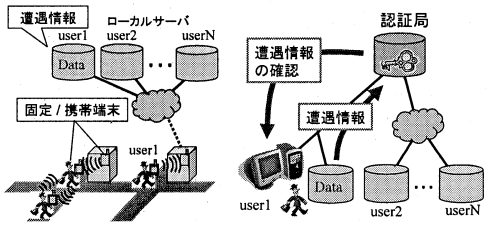
そこで本稿では (i) 遭遇情報単独で個人の特定ができず，かつ (ii) 遭遇情報に証拠性があるような，匿名性を持つ遭遇情報保証技術を提案する．遭遇情報に匿名性を持たせるためには，遭遇情報に端末の ID を付与せず，信頼できる第三者以外には個人の特定ができない遭遇情報を作成することが必要である，また，遭遇情報に証拠性を持たせるためには，端末間での遭遇情報の保証，および信頼できる第三者による遭遇情報の確認が必要である．これらを実現する手法として公開鍵暗号を用いた電子証明書の利用が考えられる．公開鍵暗号には，RSA や RSA よりビット長の短い鍵で同程度の安全性を実現可能な楕円曲線暗号などがあり，本稿で想定するセンサなどの小型端末に実装した例 [4, 5] はあるが，計算量が大きく，即時的通信には向かない．そこで提案方式では計算量の少ないハッシュ関数を利用する．匿名性を持つ ID 照合方法で用いるハッシュ関数を利用して，遭遇情報の匿名化と同時に遭遇情報に証拠性を持たせる．

また匿名化した遭遇情報からでも，位置や時刻などのユーザから提供される複数の情報を元にユーザの遭遇情報を関連付け（リンク）し，ユーザの行動履歴が追跡できてしまう問題がある．これに対して，第三者には複数のアクセスが同一ユーザによるものか判断できないリンク不能性という概念 [6] がある．提案方式では匿名性に加えてリンク不能性を満たすよう設計し，あるユーザから送信された複数の遭遇情報を基に，それらの遭遇情報が同一のユーザから送信されたことを特定できないようにしている．

提案方式が低性能な小型端末上で実現可能であるかを調べるため，MICA2 MOTE[7] 上にハッシュ関数 SHA-1[8] を実装し，計算時間を測定した結果，15$\mu sec.$ で 1 回のハッシュ値計算が可能であることが分かった．さらに，消費電力の見積もりを行った結果，ハッシュ関数 SHA-1 の消費電力は公開鍵暗号 RSA-1024[9] の約 1/340 であることが分かった．

2 提案方式の概要

提案方式では，対象領域にインフラとして固定端末（シード）が分散配置される．複数のユーザが ZigBee や Bluetooth などの近距離無線通信機能を搭載した小型端末を所持して移動し，他ユーザおよびシードと遭遇時に遭遇情報を交換・蓄積する（図 1 (a)）．この時，遭遇情報に偽証がないことの保証（**遭遇情報の保証**）を端末間で行う．ユーザは必要時に蓄積しておいた遭遇情報を信頼できる第三者（**認証局**）に



(a) 遭遇情報の保証　(b) 遭遇情報の確認

図 1: 提案方式の概要

送信して，遭遇情報に偽証がないことの確認（**遭遇情報の確認**）を行う（図 1 (b)）．このときストーカーなどの犯罪行為に悪用される可能性を考慮し，認証局はあらかじめ各ユーザが登録したユーザとの遭遇情報のみを返信する．各ユーザが登録したユーザリストは**フレンドリスト**として保存され，フレンドリストへの登録には相互の同意が必要である．蓄積した遭遇情報はユーザ端末上で保持するが，長時間の利用や端末密度が高い環境では，遭遇情報のデータ量が大きくなり，ユーザ端末上で保持できなくなる場合も考えられる．そこで，各ユーザはホームサーバなどの専用サーバ（**ローカルサーバ**）を保持しているものと仮定し，ユーザは蓄積した遭遇情報をローカルサーバに適宜転送することでメモリスペースが不足する問題を回避する．この時，遭遇情報をローカルサーバに転送するため，一部のシードはインターネットに接続されており，ホットスポットとして機能すると仮定する．ユーザはホットスポット通過時に，蓄積した遭遇情報を自身のローカルサーバに転送可能である．

遭遇情報に匿名性を持たせるためには，遭遇情報に端末の ID を付与せず，認証局以外には個人の特定ができない遭遇情報を作成することが必要である．また，遭遇情報に証拠性を持たせるためには，端末間での遭遇情報の保証，および認証局による遭遇情報の確認が必要である．これらを実現するため，2.1 章で説明するハッシュ関数を用いた匿名性を持つ ID 照合方法 [10, 11, 12, 13] を利用する．

2.1 匿名性を持つ ID 照合方法

ハッシュ関数を用いた ID 照合方法については，これまでに多くの手法が提案されている [10, 11, 12, 13]．これらの手法では，認証局（サーバ）と各端末間で秘密情報を共有しておくことで，他端末には ID を隠しつつ，認証局が ID 照合を行うことができる．文献 [10, 11, 12] では，必要な ID 数を N として認証局の ID 照合に $O(N)$ の計算時間が必要である．文献 [13] では，ID 照合の計算時間を $O(\log N)$ で実現する手法を提案している．いずれの ID 照合方法でも提案方式が要求する匿名性を満たし，提案方式では ID 照合に必要な計算時間はフレンドリストのサイズに依存するため，小さいと仮定できる．従って，シンプルな手法である Randomized Hash Lock 方式 [10] を用いる．Randomized Hash Lock 方式では，認証局が全端末の ID を保持しておき，以下の手順で ID 照合を行う．以降では，X と Y のハッシュ値を $MD[X,Y]$ とする．また，端末 i の ID を id_i とし，id_i は独立で，かつ id_i を知っているのは認証局と端末 i のみであるとする．

1. 各端末 i は乱数 R を生成して，自身の ID と連結したハッシュ値 $MD_i = MD[id_i, R]$ を求め，R とともに認証局へ送る．
2. 認証局では R と保持している各 ID とを連結したハッシュ値を総当たりで計算し，端末から送られてきた MD_i と一致した $MD[id_i, R]$ に対応する id_i が端末の ID となる．

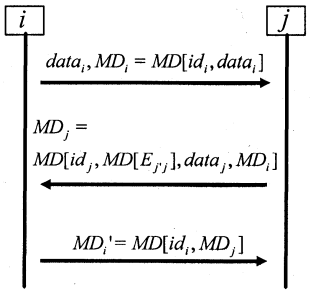


図 2: 基本プロトコル

Randomized Hash Lock 方式で生成されるハッシュ値 MD_i には ID が含まれていないので，明らかに匿名性を満たす．

定理 1 $MD[id_i, X]$ は端末 i が X を保持していたことを保証する．

ハッシュ関数は一方向性関数であるから，$MD[id_i, X]$ が id_i と X 両方を知る者でなければ作成できないことは自明である．仮定より，id_i を知っている端末は i のみである．従って，$MD[id_i, X]$ は端末 i でなければ作成できず，さらに，$MD[id_i, X]$ は X を知らなければ作成できない．以上より，$MD[id_i, X]$ は i が X を保持していたことを保証する．なお，認証局で i が X を保持していたことを保証する場合，$MD[id_i, X]$ 以外に X が必要である．

3 遭遇情報の保証

3.1 基本プロトコル

3.1.1 仮定

端末 i は遭遇時刻，遭遇位置に関する情報 $data_i$ を遭遇相手に提供する．提案方式では以下を仮定する．

仮定 1 端末 i の ID（id_i）を知っているのは i 自身と認証局のみであり，id_i は独立である．

仮定 2 フレンドリストに登録されているユーザ，シード，および自身が提供する遭遇時刻，遭遇位置に関する情報は信頼できる．

3.1.2 手順

端末 i, j が遭遇したときに j が遭遇情報を取得するための基本プロトコルの手順を図 2 に示す．ユーザ j が端末 i との遭遇情報を取得する手順は以下の通りである．

1. 端末 i は遭遇時刻・遭遇位置の情報 $data_i$ と端末 ID（id_i）から生成した $MD_i = MD[id_i, data_i]$ と $data_i$ をブロードキャストする．このとき，i がシードの場合はそのことを示すフラグ s を付与する．
2. MD_i と $data_i$ を受信したユーザ j は自身が最後に遭遇した端末 j' との遭遇情報 $E_{j'j}$ から $MD[E_{j'j}]$ を計算し，$MD[E_{j'j}]$ および端末 ID（id_j），$data_j$，受信情報 MD_i から生成した $MD_j = MD[id_j, MD[E_{j'j}], data_j, MD_i]$ を i に返信する．

図 3: ワームホール攻撃

3. MD_j を受信した i は受信情報と端末 ID（id_i）から生成した $MD'_i = MD[id_i, MD_j]$ を j に返信する．

4. j は遭遇情報 $E_{ij} =< MD'_i, MD_j, MD[E_{j'j}], data_j, MD_i, data_i >$ を保存する．

2.1 章で述べたように，$MD[id_i, X]$ の確認を認証局で行うためには，$MD[id_i, X]$ と X が必要である．E_{ij} には各ハッシュ値（$MD[id_i, X]$ に相当）と，ID と結合してハッシュ関数に入力する情報（X に相当）が含まれている．また，$MD[E_{j'j}]$ は j が最後に遭遇した端末 j' との遭遇情報をハッシュ化した値なので，既に j のメモリに保存されている．従って認証局で E_{ij} は確認可能である．

3.1.3 遭遇位置の保証

提案方式が，文献 [14] で述べられているワームホール攻撃を検出可能なことを示す．ワームホール攻撃では，攻撃者 X, Y が共謀して，X の隣接端末 i からの情報を別チャネルのリンクなどを通じて Y へ転送し，Y がその情報をそのまま隣接端末 j へ送信する（図 3）．ワームホール攻撃によって，実際には隣接していない端末 i, j が遭遇情報 E_{ij} を作成可能である．

以後，j が取得した遭遇情報 E_{ij} について，i がシードであるか，j のフレンドリストに登録されている場合に，E_{ij} は**有効である**と定義する．提案方式では，i がシードであるか，j のフレンドリストに登録されているユーザでない場合には，E_{ij} は認証局での確認時に無効となり破棄される．ゆえに仮定 2 より，有効な E_{ij} について，$data_i, data_j$ はともに正しい．ここで，i がシードであるか，GPS デバイスを保持したユーザである場合，$data_i$ は正確な位置座標 l_i および時刻情報 t_i である．i が GPS デバイスを保持しないユーザである場合，$data_i$ には i が最後に遭遇したシード s_i の位置座標 l_{s_i} と遭遇時刻 t_{s_i}，および s_i と遭遇してからの経過時間 Δt_{s_i} が含まれるものとする．$data_j$ についても，同様に GPS デバイスの有無によって $data_j$ は異なる．また，ユーザの最大移動速度を V_{max}，無線範囲を r とする．簡単のため，全ユーザについて V_{max}, r は共通とするが，ユーザ毎に異なる値を設定することも可能である．

まず，i, j どちらも正確な位置座標 l_i, l_j および時刻情報が取得可能であるとする．遭遇情報 E_{ij} を作成するため，i, j は互いに通信可能な範囲に存在していなければならない．すなわち，$|l_i - l_j| \leq r$ が成り立つ（図 4 (a)）．この制約を満たさない場合は，ワームホール攻撃が行われたと判断できる．制約を満たす場合は，j の位置座標は l_j であることが保証される．

次に，i, j のうち，どちらか一方だけが正確な位置座標および時刻情報を取得可能であるとする．まず，i だけが正確な位置座標 l_i および時刻情報を取得可能な場合を考える．今，中心座標 l，半径 r の円を $Circ(l, r)$ で表すと，j の位置座標は $Circ(l_{s_j}, V_{max}\Delta t_{s_j} + r)$ 内部に存在する．従って，遭遇情報 E_{ij} を作成するための条件から，$|l_i - l_{s_j}| \leq V_{max}\Delta t_{s_j} + 2r$ が成り立つ（図 4 (b)）．この制約を満たさない場合は，ワームホール攻撃が行われたと判断できる．満たす場合は j の位置座標は $Circ(l_i, r)$ と $Circ(l_{s_j}, V_{max}\Delta t_{s_j} + r)$ の共通領域内部であることが保証される．j だけが取得可能な場合は同様に制約 $l_i - l_{s_j} \leq V_{max}\Delta t_{s_j} + 2r$ が成り立ち，制約を満たさない場合はワームホール攻撃が行われたと判断できる．満たす場合は j の位置座標は l_j であることが保証される．

最後に，i, j どちらも正確な位置座標および時刻情報が取得できないとする．この場合，遭遇時の i, j の位置座標はそれぞれ $Circ(l_{s_i}, V_{max}\Delta t_{s_i} + r)$，$Circ(l_{s_j}, V_{max}\Delta t_{s_j} + r)$ の内部に存在する．従って，同様にして $|l_{s_i} - l_{s_j}| \leq V_{max}(\Delta t_{s_i} + \Delta t_{s_j}) + 3r$ が成り立つ（図 4 (c)）．この制約を満たさない場合は，ワームホール攻撃が行われたと判断できる．満たす場合は j の位置座標が $Circ(l_{s_i}, V_{max}\Delta t_{s_i} + 2r)$ と $Circ(l_{s_j}, V_{max}\Delta t_{s_j} + r)$ の共通領域内部であることが保証される．

以上の議論より，以下に示す定理 2 が成り立つ．

定理 2 遭遇情報 E_{ij} が有効ならば，i, j が正確な位置座標および時刻情報を取得可能かどうかに応じて，式 *1* が成り立てば，j の遭遇位置は式 *2* で示す座標，あるいは領域内であることを保証する．

$$\begin{cases} |l_i - l_j| \leq r & (i,j \text{ どちらも可能}) \\ |l_i - l_{s_j}| \leq V_{max}\Delta t_{s_j} + 2r & (i \text{ のみ可能}) \\ |l_{s_i} - l_j| \leq V_{max}\Delta t_{s_i} + 2r & (j \text{ のみ可能}) \\ |l_{s_i} - l_{s_j}| \leq V_{max}(\Delta t_{s_j} + \Delta t_{s_i}) + 3r & \\ & (i,j \text{ どちらも不可能}) \end{cases} \quad (1)$$

$$l_j \in \begin{cases} l_j & (i,j \text{ どちらも可能}) \\ Circ(l_i, r) \cap Circ(l_{s_j}, V_{max}\Delta t_{s_j} + r) & \\ & (i \text{ のみ可能}) \\ l_j & (j \text{ のみ可能}) \\ Circ(l_{s_i}, V_{max}\Delta t_{s_i} + 2r) \cap & \\ \quad Circ(l_{s_j}, V_{max}\Delta t_{s_j} + r) & (i,j \text{ どちらも不可能}) \end{cases} \quad (2)$$

3.1.4 遭遇情報の確認手順

遭遇情報の確認手順を図 5 に示す．端末 i との遭遇情報 $E_{ij} =< MD'_i, MD_j, MD[E_{j'j}], data_j, MD_i, data_i >$ は，以下の手順で確認が行われる．

1. 認証局は，シードの ID リスト，または j のフレンドリストから MD'_i と一致する $MD[id_i, MD_j]$ を見つける（図 5 (a)）．見つからない場合は不正データとして破棄する．

2. $MD[E_{j'j}]$ と直前の遭遇情報 $E_{j'j}$ のハッシュ値が一致するか確認する．一致しない場合は不正データと判断して破棄する．

3. MD_j と $MD[id_j, MD[E_{j'j}], MD_i]$ が一致するか確認する（図 5 (b)）．一致しない場合は不正データと判断して破棄する．

4. 既に id_i が分かっているので，MD_i と $MD[id_i, data_i]$ が一致するか確認する（図 5 (c)）．一致しない場合は不正なデータと判断して破棄する．

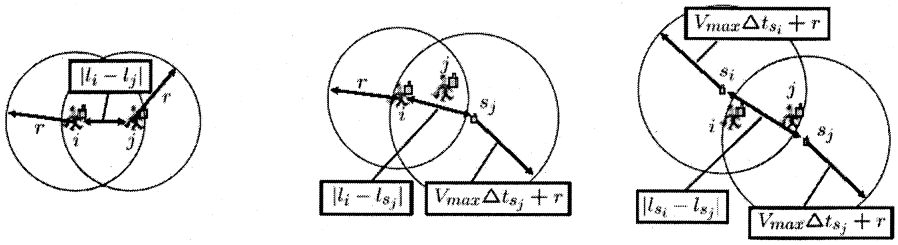
5. $data_i$ と $data_j$ について矛盾がないか式 1 によって確認する（図 5 (d)）．式 1 の制約を満たさない場合は不正データとして破棄する．

端末 i との遭遇情報 E_{ij} について以下の定理 3 が成り立つ．

定理 3 遭遇情報 E_{ij} が有効ならば i, j が正確な時刻情報を取得可能かどうかに応じて，遭遇時刻 t_{ij} は以下の式 *3* の通りであることを保証する．

$$t_{ij} = \begin{cases} t_i \text{ or } t_j & (i,j \text{ どちらも可能}) \\ t_i & (i \text{ のみ可能}) \\ t_j & (j \text{ のみ可能}) \\ t_{s_i} + \Delta t_{s_i} \text{ or } t_{s_j} + \Delta t_{s_j} & (i,j \text{ どちらも不可能}) \end{cases} \quad (3)$$

E_{ij} が有効である時，i はシード，あるいは j のフレンドリストに登録されていなければならないことは自明である．従って定理 2 より，j の遭遇時の位置は式 2 に示す通りである．また，仮定 2 より i, j の時刻情報は信頼できるので，遭遇時刻は式 3 に示す通りである．以上より，定理 3 が成り立つ．



(a) i, j どちらも取得可能　(b) i, j どちらか一方が取得可能　(c) i, j どちらも取得不可能

図 4: 遭遇位置の保証

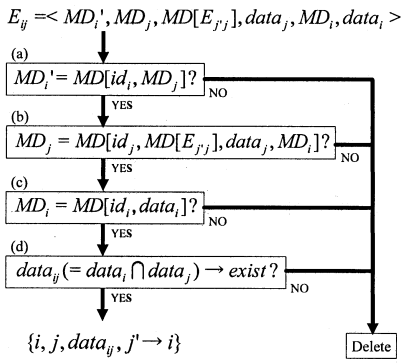


図 5: 遭遇情報の確認

定理 4 i との遭遇情報 E_{ij} はユーザ j がシード k_1 と遭遇し，その後 $k_2, k_3, \ldots, k_n, j', i$ の順に遭遇したことを保証する．

j が順に端末 $k_1, k_2, \ldots, k_n$ と遭遇したとすれば，k_n との遭遇情報 $E_{k_n j}$ には k_{n-1} との遭遇情報のハッシュ値 $MD[E_{k_{n-1}j}]$ が含まれる．$MD[E_{k_{n-1}j}]$ は $E_{k_{n-1}j}$ を知らなければ作成できないので，ユーザ j は少なくとも k_{n-1} と遭遇後に k_n と遭遇したことが保証される．同様にして，遭遇情報 $E_{k_{n-1}j}$ には遭遇情報のハッシュ値 $MD[E_{k_{n-2}j}]$ が含まれ，ユーザ j は少なくとも k_{n-2} と遭遇後に k_{n-1} と遭遇したことが保証される．つまり，遭遇情報 $E_{k_n j}$ は，ユーザ j が k_{n-2}, k_{n-1}, k_n の順に遭遇したことを保証する．これを繰返すと，遭遇情報 $E_{k_n j}$ はユーザ j が $k_1, k_2, \ldots, k_n$ の順に遭遇したことを保証する．ここで k_1 がシードであることを仮定すれば，$E_{k_1 j}$ は信頼できる遭遇位置および遭遇時刻に関する情報 $data_{k_1}$ を含んでいる．ゆえに，演繹的に定理 4 が成り立つ．

3.2 拡張プロトコル

基本プロトコルは送信情報に ID を含まないので匿名性を満たす．しかし，端末 ID が特定できない場合でも，遭遇時刻・遭遇位置の情報などユーザから提供される複数の情報を元にユーザの遭遇情報を関連付け（リンク）し，ユーザの行動履歴を追跡可能である．この問題に対して，端末から提供される情報からユーザの遭遇情報をリンクできないというリンク不能性の概念が重要である．そこでユーザの情報が第三者に対するリンク不能性を満たすよう基本プロトコルを拡張する．

3.2.1 リンク不能性

ISO/IEC 15408 [6] において，リンク不能性は，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．さらに文献 [13] において，リンク不能性の概念は定義 1 のように拡張されている．

定義 1 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，$m \neq n$）の送信元が同一ユーザによるものであると判断できない場合，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．

また文献 [13] では，Randomized Hash Lock 方式においてユーザの情報が第三者に対するリンク不能性を満たすことを示している．Randomized Hash Lock 方式では，ユーザがハッシュ値 $MD[R, id_i]$ と乱数 R を送信する．乱数は送信毎に変化するので，乱数を入力に持つハッシュ値も送信毎に変化する．つまり，ある端末の情報が第三者に対するリンク不能性を満たすためには，その端末が送信する情報を送信毎にランダムに変化させる必要である．

これらを踏まえて，基本プロトコルをユーザの情報が第三者に対するリンク不能性を満たすプロトコルに拡張する．i, j が送信する情報は MD_i', MD_j, MD_i, $data_i$ である．まず $data_i$ を考える．$data_i$ は i がシードの場合とユーザの場合で異なるため，以下では分けて説明する．i がシードの場合，$data_i$ はシード i が提供する遭遇時刻・遭遇位置の情報である．これらの値はランダムに変化しないため，リンク可能である．しかしシードの情報は第三者に対してリンク不能性を満たさなくて良い．i がユーザの場合，$data_i$ はユーザ i の正確な位置・時刻情報，もしくはユーザ i が最後に遭遇したシードの位置・時刻・そこからの経過時間の情報である．これらの値はランダムに変化しないため，リンク可能である．そこで，ユーザ i が端末 ID（id_i）を秘密鍵として $data_i$ を暗号化する．i の秘密鍵でデータ X を暗号化した暗号文を $C[id_i, X]$ とする．得られた暗号文 $C_i = C[id_i, data_i]$ はリンクできず，また第三者は秘密鍵を知らないため復号できない．よって，暗号文は第三者に対するリンク不能性を満たす．

次にハッシュ値について考える．ユーザ i が送信する MD_i の場合，ハッシュ関数の入力値に id_i, C_i を持つ．id_i は端末 ID のため固定の値である．C_i は位置・時刻の情報であり，同時刻ならば値は変わらない．そのため送信毎にハッシュ値が変わらないことが起こりうる．そこで i はハッシュ関数の入力に i が生成した乱数 R_i を加え，$MD_i = MD[id_i, C_i, R_i]$ として，MD_i, C_i, R_i を送信する．これにより，送信毎にハッシュ値をランダムに変化させることができる．同様にして MD_j, MD_i' も乱数を利用することで，ハッシュ値は第三者に対するリンク不能性を満たす．

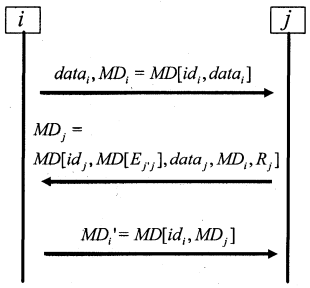


図 6: 拡張プロトコル：シード i との遭遇時

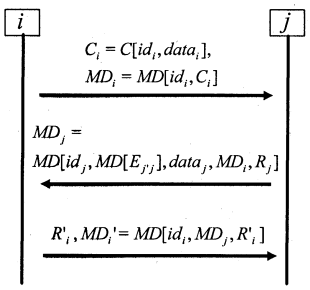


図 7: 拡張プロトコル：ユーザ i との遭遇時

以上のようにすると，シード i の情報は $data_i$ を含むためリンク可能であるが，ユーザ i の情報は MD_i, C_i, MD_i' となり，ハッシュ値と暗号文であるため，第三者に対するリンク不能性を満たす．ユーザ j の情報はハッシュ値 MD_j であるため，第三者に対するリンク不能性を満たす．以上より，シードの情報はリンク可能であるが，ユーザの情報は第三者に対するリンク不能性を満たすプロトコルを実現できる．

3.2.2 手順

拡張プロトコルはユーザ j の遭遇相手 i がシードの場合と，ユーザの場合で手順が異なる．i がシード，あるいはユーザである場合の手順をそれぞれ図 6，7 に示す．

シードとの遭遇 ユーザ j がシード i との遭遇情報を取得する手順は以下の通りである．

1. シード i は遭遇時間・遭遇位置の情報 $data_i$ と端末 ID（id_i）から生成した $MD_i = MD[id_i, data_i]$ と $data_i$ をブロードキャストする．
2. MD_i と $data_i$ を受信したユーザ j は自身が最後に遭遇した端末 j' との遭遇情報 $E_{j'j}$ から $MD[E_{j'j}]$ を計算し，$MD[E_{j'j}]$ と受信情報（MD_i）と $data_j$，端末 ID（id_j）および乱数 R_j から生成した $MD_j = MD[id_j, MD[E_{j'j}], data_j, MD_i, R_j]$ を i に返信する．
3. MD_j を受信した端末 i は受信情報と端末 ID（id_i）から生成した $MD_i' = MD[id_i, MD_j]$ を j に返信する．
4. 端末 j は遭遇情報として $E_{ij} = < MD_i', MD_j, MD_i, MD[E_{j'j}], data_j, data_i, R_j >$ を保存する．

シード i との遭遇情報 E_{ij} には各ハッシュ値に対応するハッシュ関数への入力値が含まれる．従って，シード i との遭遇情報 E_{ij} は認証局で確認可能である．

表 1: 小型センサノード MOTE の性能

	MICA2	MICA2DOT
プロセッサ	ATMega128L	
クロック周波数	7.37 MHz	4 MHz
プログラムメモリ	128 KB	
SRAM	4 KB	
無線帯域	2405 MHz	315/433/915 MHz
フラッシュメモリ	512 KB	

表 2: 1 回の計算あたりの消費電力

アルゴリズム	消費電力
RSA-1024	$304mWs$
ECDSA-160	$22.82mWs$
SHA-1	$5.9\mu Ws/byte$

ユーザとの遭遇 ユーザ j がユーザ i との遭遇情報を取得する手順は，遭遇相手からの遭遇時刻・遭遇位置に関する情報 $data_i$ が i の秘密鍵で暗号化される以外は，シードとの遭遇と同様である．i の秘密鍵は認証局でも保持しており，$C[id_i, X]$ は i または認証局のみで復号可能である．

1. ユーザ i は端末 ID（id_i）を用いて，$data_i$ を暗号化し（$C_i = C[id_i, data_i]$），C_i と id_i から生成した $MD_i = MD[id_i, C_i]$ および C_i をブロードキャストする．
2. MD_i と C_i を受信したユーザ j は自身が最後に遭遇した端末 j' との遭遇情報 $E_{j'j}$ から $MD[E_{j'j}]$ を計算し，$MD[E_{j'j}]$ と受信情報（MD_i），$data_j$，端末 ID（id_j）および乱数 R_j から生成した $MD_j = MD[id_j, MD[E_{j'j}], data_j, MD_i, R_j]$ を i に返信する．
3. MD_j を受信した端末 i は受信情報と端末 ID（id_i）と乱数 R_i' から生成した $MD_i' = MD[id_i, R_i', MD_j]$ と R_i' を j に返信する．
4. 端末 j は遭遇情報として $E_{ij} = < MD_i', MD_j, MD_i, MD[E_{j'j}], data_j, C_i, R_i', R_j >$ を保存する．

ユーザ i との遭遇情報 E_{ij} には各ハッシュ値に対応するハッシュ関数への入力値が含まれる．また認証局は C_i を復号可能であるので，ユーザ i との遭遇情報 E_{ij} は認証局で確認可能である．

拡張プロトコルで交換した遭遇情報の確認方法は，基本プロトコルと同様であるので省略する．また基本プロトコルと同様にして，定理 2, 3, 4 も成り立つ．

4 評価実験

提案方式が低性能な端末上に実装可能かどうかを調べるため，簡単な実験を行った．実装には MICA2 と呼ばれる ZigBee 規格の短距離無線デバイスを搭載した小型センサノード MOTE [7] を用いた．出力サイズ 160bit のハッシュ関数 SHA-1 [8] を MOTE 上に実装したところ，ハッシュ関数の計算時間は，1 回あたり $15\mu sec.$ ($15 \times 10^{-6} sec.$) であることが確認できた．提案方式において各端末が送受信するデータに対して行う主な処理はハッシュ関数を用いた計算である．従って，実用的な速度での遭遇情報作成が可能である．

次に公開鍵暗号およびハッシュ関数による電力消費の比較を行う．文献 [15, 16] において，MICA2DOT を用いたときの，RSA の暗号化アルゴリズム（RSA-1024 [9]），ECC の暗号化アルゴリズム（ECDSA-160 [17]），ハッシュ関数（SHA-1）それぞれの計算による電力消費量が求められている．MICA2 および MICA2DOT の性能は表 4 の通りであ

る．これらの電力消費量を表 4 にまとめる．ハッシュ関数は 1 バイトあたりの電力消費量が求められているため，提案方式の拡張プロトコルにおけるハッシュ関数の計算 1 回あたりの電力消費量を求める．ユーザとの遭遇時に得るハッシュ値の入力ビット長が最大となるのは $MD_j = MD[id_j, MD[E_{j'j}], data_j, MD_i, R_j]$ である．$MD[E_{j'j}]$ はハッシュ値であるから，ハッシュ関数の出力ビット数分のデータサイズになる．まず $E_{j'j} = < MD'_{j'}, MD_j, data_j, MD[E_{knj}], MD_{j'}, data_{j'} >$ から $MD[E_{j'j}]$ を計算時に必要なハッシュ関数の入力データサイズを見積もる．ここで id_j は 32 ビットあれば 2^{32} 通りを表現できるので，id_j のデータサイズは 4 バイトあれば十分と考える．また $data_{j'}$ は時間と位置で構成される．時間は，サービスを 1 日単位で運用すれば十分と仮定して，時を 5 ビット，分を 6 ビット，秒を 6 ビットの計 3 バイトで表現できる．月日を加えることを考慮しても，合計 5 バイトあれば十分である．位置は仮に $10km$ 四方の領域に適用する場合を考えると，16 ビットで表す場合，$10000/2^{16} = 0.15m$ の精度で位置情報を表現できるので，x 座標，y 座標ともに 2 バイトあれば十分である．従って $data_{j'}$ は約 9 バイトとなる．ハッシュ関数は SHA-1 を利用すると，ハッシュ値のサイズは 20 バイトである．まずハッシュ値 $MD[E_{j'j}]$ を生成するために必要な入力データサイズは 98 バイトとなる．$data_j$ は約 9 バイト，MD_i は 20 バイトであり，乱数 R_j のビット長は決まっていないので，ここでは 1 バイトで考えると，MD_j を生成するために必要なデータサイズは 54 バイトとなる．以上より，ハッシュ値 MD_j を生成するためには合計 152 バイトが必要になる．従って，提案方式で 1 ユーザとの遭遇情報を作成する際の消費電力は $5.9\mu Ws/byte \times 152bytes = 897\mu Ws = 0.897mWs$ であり，RSA-1024 および ECDSA-160 の計算に必要な消費電力のそれぞれ約 1/340，1/25 に相当する．以上より，ハッシュ関数の計算時間は非常に短く，消費電力も少ないため，提案方式で想定する低性能な小型端末でも十分実装可能である．

5 まとめ

本稿では，小型端末を利用した匿名性を持つ遭遇情報保証技術を提案した．提案方式では，匿名性およびリンク不能性を持つ遭遇情報の収集プロトコルを考案し，低性能な小型端末上に実装可能な計算量が少ないハッシュ関数を用いて設計した．提案方式が小型端末上で実装可能であることを確認するため，ハッシュ関数として SHA-1 を MOTE 上に実装し，実験を行った．その結果，計算時間は短く，消費電力も RSA を用いた公開鍵暗号と比べて非常に少ないことが確認できた．今後は，シミュレーションによって提案方式の精度やスケーラビリティなどに関する性能評価を行い，MOTE 上に実装することで，低性能な小型端末上で動作可能であることを確認する予定である．

参考文献

[1] *Wi-Fi location system.* `http://www.sandi.co.jp/location/`.

[2] *911 services.* `http://www.fcc.gov/911/enhanced/`.

[3] J. Huang, S. Amjad, and S. Mishra. CenWits: A sensor-based loosely coupled search and rescue system using witnesses. In *Proc. of ACM SenSys*, pp. 180–191, 2005.

[4] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In *Proc. of Workshop on Cryptographic Hardware and Embedded Systems*, pp. 119–132, 2004.

[5] D. J. Malan, M. Welsh, and M. D. Smith. A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In *Proc. of ACM SenSys*, pp. 71–80, 2004.

[6] ISO/IEC 15408. Common criteria for information technology security evaluation - part2: Security functional components, 2006.

[7] *MOTE official page.* `http://www.xbow.jp/motemica.html`.

[8] D. Eastlake, and P. Jones. US secure hash algorithm 1 (SHA1). RFC3174, 2001.

[9] C. K. Koc. High-speed RSA implementation. Technical report, RSA Laboratories, 1994.

[10] S. A. Weis, S. E.Sarma, R. L. Rivest, and D. W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Proc. of Security in Pervasive Computing*, pp. 201–212, 2004.

[11] M. Ohkubo, K. Suzuki, and S. Kinoshita. Hash-chain based forward-secure privacy protection scheme for low-cost RFID. In *Proc. of Symp. on Cryptography and Information Security*, pp. 719–724, 2004.

[12] 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. コンピュータセキュリティシンポジウム（CSS）論文集, pp. 139–144, 2004.

[13] 野原康伸, 井上創造, 馬場謙介, 安浦寛人. 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. 暗号と情報セキュリティシンポジウム（SCIS）予稿集, pp. 1567–1572, 2005.

[14] Y.-C. Hu, A. Perrig, and D.B. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *Proc. of IEEE INFOCOM*, pp. 1976–1986, 2003.

[15] K. Piotrowski, P. Langendoerfer, and S. Peter. How public key cryptography influences wireless sensor node lifetime. In *Proc. of ACM SASN*, pp. 169–176, 2006.

[16] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *Proc. of IEEE PerCom*, pp. 324–328, 2005.

[17] D. Hankerson, A. Menezes, and S. Vanstone. *Guide to Elliptic Curve Cryptography.* Springer-Verlag, 2004.