携帯端末向けプライバシ保護型 操作履歴ミドルウェアの設計と実装

太田賢，木南克規，中川智尋，土井千章，稲村浩

携帯端末の多様な操作履歴データを活用するためのミドルウェア向けに，携帯端末のリソース制約に対応した 2 つのプライバシ保護方式を提案する．第一に，操作データの収集を必要最小限に維持するため，オンデマンド収集方式は複数アプリケーションからの収集の開始・停止・制限の要求に基づき，必要最小限の収集すべき操作種別の集合と保存期間を動的に決定し，収集制御とデータ消去を行う．さらに時空間ベースの収集ルールを利用した自動収集制御により，収集期間や頻度を限定し，プライバシ保護の向上とリソース消費削減をはかる．第二に，選択的暗号化方式は性能とセキュリティのバランスのため，操作履歴データベースの特定フィールドを部分的に暗号化する手段を提供する．性能評価の結果，ユーザ操作や端末状態の操作データ収集のオーバヘッドは小さいが，センサデータ収集のリソース消費は大きく，オンデマンド収集方式によるリソース消費削減が適用できることが分かった．また，選択的暗号化方式の操作データ記録における暗号化オーバヘッドは 26 ミリ秒以下に収まったが，クエリ時間は暗号化フィールドを参照する場合に実行される復号処理量の影響が大きいことが分かった．

Design and Implementation of Privacy-enhanced Operation History Middleware for Mobile Handset

KEN OHTA† KATSUKI KIMINAMI† TOMOHIRO NAKAGWA† CHIAKI DOI† HIROSHI INAMURA†

This paper proposes two privacy protection functions for middleware collecting sensitive operation data. First, the on-demand collection function dynamically determines the minimum collection set of operation types and their storage period for collection and erasing control. It automatically starts and stops collection based on temporal-spatial-based rules to limit a timeframe and frequency of collection. Second, the selective encryption function encrypts specific fields of sensitive operation data in the local database for balancing performance and security. Our evaluation shows that sensory data collection consumes resource heavily. The on-demand collection function is applicable to limit collection of sensory data. Encryption overhead in recording is less than 26 milliseconds, while query time depends on the number of decryption processing.

1. はじめに

携帯電話は常時ユーザのそばにあり，位置情報等のセンサを備えたパーソナルな多機能デバイスである．現在から過去までのユーザの状況や行動を示す端末の操作履歴は，ユーザインタフェースのカスタマイズ[a]，利用のモニタリング [10]，コミュニケーション支援[b]，過去の活動に基づく情報推薦 [2]等の多様なアプリケーションで活用できる．

本研究は操作履歴を利用するアプリケーションの開発とリソース利用の効率化のため，統合的に操作データの継続的収集を行うミドルウェアの実現を目的とする．ミドルウェア導入により，アプリケーション個別での操作データの収集や送信の機能開発，常駐の実行の必要がなくなり，開発コストと CPU やバッテリ等のリソース消費の削減がはかれる．ミドルウェア実現の課題はパーソナルな操作履歴データのプライバシ保護と，リソース制約のある携帯端末での使い勝手の両立である．多様な操作データの頻繁な収集や暗号化による保存等の処理は応答性の悪化や端末の稼働時間の減少を引き起こす可能性がある．

本稿は携帯端末のリソース制約に対応した，操作履歴ミドルウェアのプライバシ保護機能として，操作データ収集を必要最小限に維持するオンデマンド収集方式と，操作履歴データベースのレコードの指定フィールドのみを暗号化する選択的暗号化方式を提案する．プライバシ保護として文献 1)で示された P1.ユーザへの収集状況の通知，P2.選択肢の提供と許諾の確保，P3.必要最小限のデータ収集，P4.認証や暗号化を含む適切なセキュリティの確保の 4 つの方針に従う．オンデマンド収集方式は P3 に関して収集する操作データ種別，保存期間，収集期間を必要最小限に維持するものである．複数アプリケーションからの収集の開始・停止・制限の要求に基づき，最小限の収集すべき操作種別の集合とそれぞれの保存期間を動的に決定し，収集や消去を制御する．さらに収集期間や頻度を限定するため，アプリケーションから時空間の条件に基づく収集要求を ECA(Event, Condition, Action)ルールの形式として受付け，自動で収集の開始や停止を制御する．また，選択的暗号化方式は，P4 に関してリソース制約のある携帯端末の性能とセキュリティのトレードオフを考慮し，操作データレコードの操作種別，タイムスタンプ，パラメータのフィールドにおいて，保護が要求されたフィールドのみを部分的に暗号化する．クエリが暗号化されたフィールド内容の参照を必要とする場合は復号を行う．

Android ベースのプロトタイプにおいて，P1, P2 の対応のため，収集状況の常時通知及び収集設定機能を実装した．また，Android フレームワーク上に 1. Logcat 監視，

*† 株式会社 NTT ドコモ
NTT DOCOMO, INC.

a NTT ドコモ技術資料，きせかえツールコンテンツ作成ガイド，
http://www.nttdocomo.co.jp/service/imode/make/content/kisekae_tool/

b つながりほっとサポートサービス, http://202.214.192.60/service/communication/tsunagari_hotto_support/

2.BroadcastIntent 監視, 3.EventListener 監視, 4.アプリケーション固有収集, 5. 操作データ生成の 5 つの操作データ収集法を実装した．性能評価の結果，ミドルウェアに起因するバッテリ消費はバッテリ容量の 10%以下，端末の CPU 使用率は 6%に収まることが確認された．ユーザ操作や端末状態の収集に比べて，センサデータ収集のリソース消費は大きく，常時の収集ではなく，収集期間や頻度を限定する必要があることが分かった．提案のオンデマンド収集方式により，時空間的なルールで収集期間を限定可能である．一方，選択的暗号化機能に関して，操作データの記録における暗号化オーバヘッドは 1024 バイトまでの操作データで 26 ミリ秒以内に収まるものの，クエリ時間はクエリが暗号化データを参照する場合に必要な復号処理の影響が大きく，応答性を確保するためにはクエリの範囲の限定や，暗号化するフィールドの限定の必要があることが確認された．

以降，2 章で操作履歴データの形式や種別とミドルウェアの要件について述べる．3 章で提案ミドルウェアの設計を示し，オンデマンド収集方式を提案する．4 章でプロトタイプ実装と性能評価について述べ，5 章でまとめと今後の課題を述べる．

2. 背景と関連研究

2.1 操作履歴データ

操作履歴データは時系列に並んだ操作データレコードの集合である．各レコードは操作種別，タイムスタンプ，操作種別固有の複数のパラメータを含む．本稿は能動的なユーザ操作である端末操作データと受動的な操作であるセンサデータと端末状態データの 3 種の操作データを扱う．

- 端末操作データ：通話，メール送受信，写真撮影，ナビゲーションやゲーム等のアプリケーション起動，ブラウジング，TV 視聴など．通話の操作種別において通話相手や通話時間等がパラメータとなる．
- センサデータ：GPS やセルラ網の基地局による位置，加速度センサ情報など．位置の操作種別において，緯度，経度などがパラメータになる．
- 端末状態データ：バッテリ状態やロック状態，マナーモード状態など．バッテリ状態の操作種別において，バッテリ残量や充電状態等がパラメータとなる．

操作履歴データを活用するアプリケーションの例として，子どものケータイ利用みまもり [10]は様々なアプリケーションの起動履歴やメール送受信，ブラウジング，TV 視聴などの操作種別を利用する．対象アプリケーションの全体利用時間が指定時間を超えた際に通知を行うと共に，日々のアプリケーションの起動回数やメール送信回数等を含む日記をメールでフィードバックすることでリテラシー向上をはかる．

2.2 ミドルウェアの要件

アプリケーション開発とリソース利用の効率化のため，操作履歴アプリケーションが共通的に必要とする機能を機能要件とする．具体的には多様な操作履歴の収集や保存，操作履歴サーバへのアップロードの機能が挙げられる．アップロード機能はサーバ側アプリケーションの対応に必要であり，操作履歴サーバはアップロードされた操作履歴データをサーバ上に保存し，サーバアプリケーションに提供する．

操作データはユーザの行動や状況に関するプライバシ情報を含むことがあり，不正アクセスや情報漏洩等から保護する必要がある．例えば通話の操作種別において通話先情報はプライバシ情報と考えられる．表 1 にユビキタスコンピューティングにおけるデータ収集に関する 6 つのプライバシ保護方針を示す [1]．本稿は P1 から P4 を採用する．本稿ではミドルウェア自身が操作履歴データを他のエンティティと共有するユースケースは扱わないため，P.5 の配布の局所性と P.6 の匿名性と仮名性は各アプリケーションの課題とする．また，P5 の近接性は，ユーザが常に携帯端末を携帯しているものと仮定し，扱わないものとする．身につけるリモコンと携帯端末が離れたことを検知する機能を備えた端末もある[c]．

表 1　プライバシ保護の方針
Table 1　Principles of privacy protection.

番号	方針	説明
P1	ユーザへの収集状況の通知	データを収集していることをユーザに明確に知らせること
P2	選択肢の提供と許諾の確保	ユーザに，収集の可否や程度に関する選択肢を提供すると共に，ユーザの明確な許諾を得ること
P3	必要最小限のデータ収集	明確に規定された目的に沿ったデータのみを収集し，その目的に必要な間だけ保存すること
P4	認証や暗号化を含む適切なセキュリティ	収集データの重要性に従って，ユビキタス環境のリソース制約を考慮しつつ，適切なセキュリティレベルの手段をとること
P5	近接性と局所性	意図しない監視を防止するため，デバイスの所有者がそばにいない場合，収集を停止すること．収集した情報を無制限に配布しないこと
P6	匿名性と仮名性	収集データから個人が特定できないことを保証すること，リンク不能性を確保すること

2.3 関連研究

操作データはユーザと携帯端末のコンテキストを示すものであり，操作履歴ミドルウェアはコンテキスト情報システムとして位置づけられる．文献 2)ではコンテキストを，あるエンティティの状況を特徴付けるあらゆる情報と定義している．

---

cキッズケータイ F-05A, http://www.nttdocomo.co.jp/info/news_release/page/090128_00.html

コンテキスト情報の収集を行うミドルウェアは数多くあり，The Context Toolkit [7]のコンテキスト収集ウィジェットは物理センサの違いを隠蔽し，アプリケーションが使いやすいデータ形式で提供する．Context Phone [8]は位置や携帯電話の利用，近接のデバイスなどの情報を収集する Symbian ベースのミドルウェアである．これらのミドルウェアに提案のオンデマンド収集方式を適用することで，操作データの収集や保存を最小化してプライバシ保護を向上できる．また，時空間の条件に基づく自動の収集制御により，アプリケーション開発の負担を削減できる．

コンテキスト情報システムのプライバシ保護には包括的な対策が必要である．Confab[3]はコンテキスト情報の共有において，出力内容や出力先を含む情報フローのユーザへの提示や，ユーザが共有する情報の粒度や頻度，期間に関する設定や個人情報の扱いに関する共有先への要求を行うことを可能にする．また，PCO[5]はオントロジを用いて共有する相手によって，コンテキスト情報の抽象化を制御可能とする．Confab や PCO は共有制御を扱うのに対し，提案のオンデマンド収集方式は収集・保存時のプライバシ保護を扱うものであり，組み合わせて利用可能である．TaintDroid[6]はプライバシ情報のフローを追跡し，不正なアプリケーションがプライバシ情報を外部送信することを検知するシステムである．TaintDroid の利用により，操作履歴を利用する Android アプリケーションが不適切な振る舞いをしていないかを監視できる．

コンテキスト情報システムのエネルギー効率に関して，EEMMS[4]は階層的なセンサ管理手法によって必要なセンサのみをオンにすると共に適切なセンサの動作周期を制御することで，消費電力を削減する手法を提案している．適応的なロケーションセンシングフレームワーク [9]は消費電力低減のため，代用や抑制機能，ピギーバック，アプリケーションのロケーションセンシング要求の適応などの機能を備えている．

3. 操作履歴ミドルウェアの設計

3.1 アーキテクチャ

操作履歴ミドルウェアは OS やアプリケーションフレームワークから操作データを収集し，保存や消去を行う収集管理モジュール，送信を行うアップロード管理モジュール，操作データの収集状況の確認や設定を行う機能設定モジュール，暗号化して操作データを保存する操作履歴 DB(データベース)から構成される（図１）．ミドルウェアはいくつかの API を提供し，*Collect and Suppress* API は操作データ種別を指定して収集の開始・停止・制限，保存期間の要求を行う．*Record* API はアプリケーション固有の操作データの書き込み， *Query* API は操作履歴データの取得を行う．さらに，サーバアプリケーションは *Upload* API を利用して，プロキシモジュールを介して操作データの即座のアップロード，周期的なアップロードを要求できる．プライバシ保護方針の P1 と P2 に関して各モジュールの機能を説明する．P3，P4 は 3.2，3.3 で述べる．

・P1: ユーザへの収集状況の通知

機能設定モジュールは収集中の操作種別一覧の提示機能と，動作状態をステータスウィンドウ等に表示する機能を備え，ユーザが収集状況を常時確認可能とする．

・P2: 選択肢の提供と許諾の確保

機能設定モジュールは，選択肢としてユーザが操作種別単位で収集の制限や消去する手段を提供する．許諾についてはアプリケーションによって操作データの利用の目的が異なるため，アプリケーション自身がユーザの許諾をとるものとする．

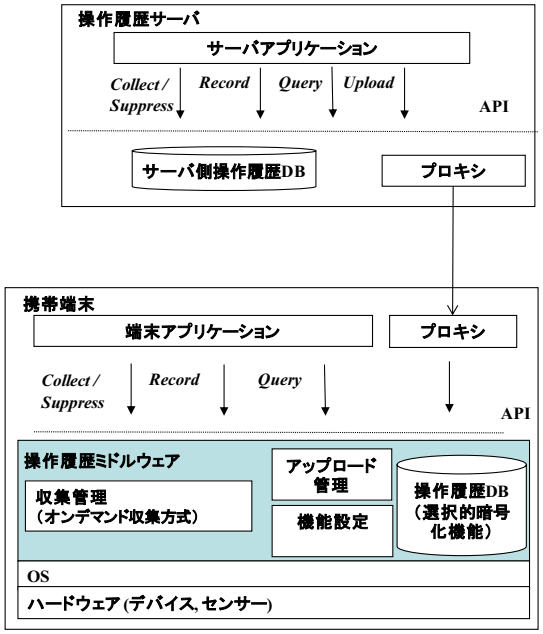

図 1　操作履歴ミドルウェアのアーキテクチャ
Figure 1　Architecture of operation history middleware.

3.2 オンデマンド収集方式

P3 を満たすには，収集する操作データ種別，保存期間，収集期間と頻度を必要最小限とするための機構が必要である．提案方式は複数のアプリケーションからの収集の開始・提示・制限の要求に従って最小限の収集すべき操作種別の集合とそれぞれの保存期間を含む収集設定を動的に決定し，収集の開始・停止の制御と保存期間を過ぎた

操作データを消去することで，必要最小限の収集状態を維持する．収集の停止は収集開始を要求したアプリケーションがその要求をキャンセルするため，収集の制限は収集を要求する他のアプリケーションの存在に関わらず，指定の操作種別の収集を禁止するためのものである．また，収集期間や頻度を制限するため，オンデマンド収集方式はアプリケーションから時空間の条件に基づく収集要求を ECA ルール形式で受付け，収集の開始や停止を制御する．アプリケーションの要求の具体例を示す．

- ユーザインタフェースカスタマイズのアプリケーションは，ユーザの許諾が得られた際に，アプリケーション利用の操作種別の収集と 1 ヶ月の保存期間を要求する．
- 利用モニタリングアプリケーションはアプリケーション利用と位置情報の操作種別について，午前 9 時から午後 17 時までの時間的に限定した収集と 1 日の保存期間を要求する．
- ソーシャルアプリケーションはオフィスにいる際にのみ，近接デバイスの操作種別の収集を要求し，1 週間の保存期間を指定する．
- 機能設定アプリケーションは各操作種別の収集を制限する設定項目をユーザに提供し，設定変更時に収集の制限や解除を要求する．

以下，収集設定を動的に決定する収集・保存管理アルゴリズムと，時空間条件に基づく収集制御を説明する．

3.2.1 収集・保存管理アルゴリズム

本アルゴリズムはアプリケーションが収集関連の要求を発行した際に実行される．図 2 の左に示す擬似コードの通り，操作種別 *i* に関して少なくとも 1 つのアプリケーションが収集を要求している場合，収集可否設定 *C[i]*を true に設定する．true は収集，false は収集停止を示す．ただし，収集要求があっても制限を要求しているアプリケーションが 1 つでもある場合，収集を制限することで競合を解決する．なお，アプリケーション *a* が操作種別 *i* の収集を開始，停止，制限，制限の解除を要求する場合，アプリケーション要求設定 *K[i,a]* はそれぞれ”COLLECT”，“NULL”，“SUPPRESS”，“NULL”に設定されるものとする．

図 2 の右に操作データの保存を最小限にする保存管理の擬似コードを示す．関数 *storage-period* は操作種別 *i* の端末保存期間 *T[i]*とサーバ保存期間 *S[i]*を端末アプリケーションとサーバアプリケーションそれぞれの最大の保存期間要求に設定し，定期的に実行される関数 periodical-erasing がその保存期間に従って不要な操作データを消去する．端末保存期間は端末アプリケーションが利用する端末側操作履歴 DB における保存期間，サーバ保存期間はサーバアプリケーションが利用するサーバ側 DB の保存期間に対応する．*L[a, i]*はアプリケーション *a* からの操作種別 *i* の保存期間要求を示す．ただし，*A* 個のアプリケーションの内，*1* から *A’*が端末アプリケーション，*A’+1* から *A* はサーバアプリケーションを指すものとする．消去の処理において，T[i] ≥ S[i]の場合，T[i]より古い操作種別 *i* の操作データは端末側とサーバ側とも不要であるため消去する．一方，T[i]<S[i]の場合，S[i]よりも古い操作履歴データに加えて，T[i]よりも古い操作データであってかつアップロード済みの操作データを消去する．これはサーバ保存期間以内の操作データは，端末保存期間を過ぎていたとしてもアップロード済みでない場合に保持するためである．これにより，無線の接続性の問題やサーバのダウンによってアップロードが妨げられても，障害からの回復時にアップロードできる．

```
FOR   i =1 to N // N は操作種別数
 flag = 0
 FOR j=1 to A // A はアプリケーション数
  IF K[i,j] = COLLECT THEN
    C[i] = true
  END IF
  IF K[i,j] = SUPPRESS THEN
   flag = 1
  END IF
 END FOR
 IF flag = 1 THEN
  C[i] = false
 END IF
END FOR
```

```
storage-period()
  T[i] = max(L([1,i], …   L[A’,i]) // 端末保存期間
  S[i] = max(L[A’+1,i],…,L[A,i]) // サーバ保存期間
RETURN
periodical-erasing()
      FOR   i =1 to N   // N は操作種別数
      IF S[i] =0 OR T[i] >= S[i] THEN
                  erase-old-data(T[i])
      ELSE IF T[i] < S[i] THEN
                  erase-old-data(S[i])
                  erase-uploaded-data (T[i])
      END IF
    END FOR
RETURN
```

図 2　収集管理(左)と保存管理(右)の擬似コード
Figure 2　Pseudo code of collection and storage management.

3.2.2 時空間条件に基づく収集制御

オンデマンド収集方式は，時空間の条件に基づく ECA ルールベースの収集要求を解釈，実行する機能を提供する．すなわち，アプリケーション自身は時間や位置を監視して収集の開始/停止を制御する必要はなく，ルールを規定して与えればよい．ECA ルールは XML で記述するが，図 3 にその具体例を示す．ルール 59，60 は時間条件の例で，2011 年 3 月 9 日の 7 時に位置情報の収集開始，22 時に収集停止を要求している．一方，ルール 61 は時空間条件の例で，緯度経度が(35.6810737056106, 139.767036437988)の場所から 100 メートル以内であって，3 月 10 日から 3 月 17 日の間であった場合に，位置情報の収集を要求している．

```
<rule id="59“>
  <event><time><eq type="datetime“>2011-03-09T7:00:00</eq></time></event>
   <condition/>
   <action> <logstart kind="LOCATION"/> </action>
  </rule>
  <rule id="60“>
  <event><time><eq type="datetime“>2011-03-09T22:00:00</eq></time></event>
   <condition/>
   <action> <logstop kind="LOCATION"/> </action>
  </rule>
  <rule id="61">
   <event><center lat="35.6810737056106" lon=”139.767036437988” kind="LOCATION”>
    <le type="numerical”>100</le></center></event>
   <condition><and>
    <time><ge type="datetime">2011-03-10T00:00:00</ge></time>
    <time><le type="datetime">2011-03-17T00:00:00</le></time>
   </and> </condition>
   <action> <logstart kind="LOCATION"/> </action>
</rule>
```

図 3　時空間条件に基づく ECA ルールベースの収集要求
Figure 3　ECA-rule-based collection request using temporal-spatial condition.

3.3 選択的暗号化方式

P4 に関してミドルウェアへの脅威は，悪意のあるソフトウェアからのミドルウェアの API や操作履歴 DB(DB と呼ぶ)への不正アクセス，攻撃者や他のユーザによる不正操作，通信路の覗き見・改ざんの 3 つを含む．

DB への不正アクセス防止に対して選択的暗号化方式を提案する．操作データの保存の際，操作データレコードの操作種別，タイムスタンプ，パラメータのフィールドの内，暗号化設定情報において保護が要求されたフィールドのみを部分的に暗号化する．一方，操作種別や時間範囲，パラメータの指定を含むクエリを処理する際，そのクエリが暗号化されたフィールドを参照する場合は復号を行う．

本稿では，性能とセキュリティのバランスを考慮して，パラメータのフィールドを暗号化し，操作種別とタイムスタンプのフィールドは暗号化しない，という暗号化設定情報を仮定する．例えば位置の操作種別の場合，緯度経度は暗号化されるアプリケーション利用の操作種別の場合，アプリケーション名は暗号化され，通話の操作種別において通話相手や通話時間は暗号化される．ただし，それぞれそのタイムスタンプの時刻に何らかのアプリケーションを利用したこと，どこかに通話したことが攻撃者に見られる可能性がある．その懸念を許容できない場合，操作種別も暗号化するような暗号化設定情報を利用する必要がある．

本稿で仮定する暗号化設定情報の場合，操作種別と時間範囲を指定して操作データのレコードを取得するクエリは復号処理なしに応答が可能である．ただし，取得後にそのレコードのパラメータを参照する際には復号の必要がある．一方，パラメータの特定のフィールドのパラメータ値を指定してマッチするレコードを取得するクエリについては応答にあたって復号処理が必要となる．他の例としてタイムスタンプ以外のフィールドを暗号化するような暗号化設定情報の場合，時間範囲のみを指定したレコードの取得は復号処理なしに応答可能であるが，操作種別やパラメータを条件とする場合，復号処理が必要である．

以下に本ミドルウェアのその他のセキュリティ対策を示す．

- 不正アクセス防止：ミドルウェアはパーミッションを持つアプリケーションにのみ API アクセスを許可するアクセス制御を行う．
- 不正操作防止：機能設定モジュールは暗証番号によってユーザを認証し，収集に関する不正な設定を防止する．携帯端末を他ユーザに貸与や譲渡する利用シナリオにおいて，他のユーザが操作履歴データを利用できないようにするため，操作履歴 DB は UIM(User Identity Module)に基づくアクセス制限を行う．操作データの各レコードに UIM の識別子を記録し，クエリを処理する際，装着中の UIM に紐尽くレコードのみを処理対象とする．
- 通信路の保護: アップロード管理モジュールは操作履歴サーバに操作データを送信する際，SSL で端末と操作履歴サーバの間のセキュアな通信チャネルを確立し，操作履歴データの覗き見や改ざんを防止する．

4. 実装と評価

4.1 Android プロトタイプシステム

ミドルウェアを Android 2.2(Froyo)ベースの Nexus One に実装した．互換性の確保のため，Android のプラットフォームの改変を行わず，実装を行っている．操作履歴 DB の選択的暗号化方式の暗号化アルゴリズムは DES の ECB モードを利用した．アップロード管理モジュールは XML ベースのフォーマットで操作データを集約して操作履歴サーバに送信する．Logcat 監視, 2. BroadcastIntent 監視, 3. EventListener 監視, 4.アプリケーション固有収集，5. 操作データ生成の 5 つの収集法を Android フレームワーク上に実装した．表 2 にプロトタイプでサポートしている操作種別の一部を示す．

1. Logcat 監視: スレッドを生成して Logcat を実行し，その出力を保存する．例えばアプリケーションの起動終了や Activity 単位での画面遷移を抽出できる．

2. BrodcastIntent 監視: 携帯端末の状態が変化した際に Android フレームワークから通知される *BroadcastIntent* を監視し，その受信を契機に，Manager や Provider，*intent* の拡張領域から情報を取得する．通知される BroadcastIntent として画面の点灯/消灯，パッケージのインストール，マナーモードの設定，WiFi や Bluetooth の接続状態などがある．Provider の場合，各種 Provider がデータベースに登録しているコンテンツ情報を *ContentResolver* を経由して取得する．例えば，ブラウザのブックマーク情報(Web 閲覧履歴)を取得できる．

3. EventListener 監視: コールバックを登録して，電話の通話状態及び受信強度，GPS の受信状態，各種センサ(加速度/地磁気/傾斜等)等の状態変化通知を受信し，そのリスナーに対応した Manager クラスから情報を取得する．

4. アプリケーション固有収集: アプリケーションから *Record* API を呼び出してアプリケーション固有の操作データを記録する．

5. 操作データ生成:既存の操作種別から新たな操作種別のデータを生成する．本プロトタイプでは，位置情報と加速度センサという操作データを利用し，あらかじめユーザが設定したランドマーク情報への滞在や移動状態を検知し，自宅や職場，通勤中などのプレゼンス情報を生成する機能を実装した．

表 2　操作データ種別と収集法
Table 2　Types of operation data and collection technique.

カテゴリ	種別	詳細	収集法
能動的ユーザ操作:端末操作データ	電話発信，電話着信	発信/着信番号,応答有無,通話時間	1,2,3
	アプリ利用	アプリケーション ID，名前	1
	ブラウジング	アクセス先 URL	1,2
	カメラ撮影	写真の保存枚数	1,2
	スケジュール	予定の件名，時間，場所，内容	2
受動的操作:センサデータ	位置	測位種別，経度，緯度	3
	加速度センサ	出力値 1,2,3(X,Y,Z 軸)	3
	プレゼンス	プレゼンス情報(自宅・職場等)	5
受動的操作： 端末状態データ	電池残量	バッテリレベル，充電状態	2
	電源状態	ブート，シャットダウン	2
	画面点灯	点灯・消灯	2
	マナーモード	マナーモード設定状態	2

4.2 性能評価

プロトタイプで操作データ収集のオーバヘッド，操作データの保存とクエリにかかる選択的暗号化方式のオーバヘッドの評価実験を行った．リソース消費に関して，収集法 1 から 4 を有効にした条件において，以下の 2 点を要件として設定した．

R1.ミドルウェアによるバッテリ消費の増分がバッテリ容量の 10%以下であること

R2. CPU 使用率が常時 10%以下に抑えられること

24 時間，端末の操作をすることなく静止状態において測定を行った．リソース消費に大きく影響すると考えられる GPS と WiFi をオンにした場合（シナリオ 1）とオフにした場合（シナリオ 2）のバッテリ消費と CPU 使用率，メモリ使用量を測定した．バッテリ消費は試験開始時点と終了時点での電池残量表示の差を読み取り，CPU 使用率及びメモリ使用量は top コマンドを使用し，5 分間隔で測定した．ミドルウェアによるバッテリ消費は，ミドルウェアを動作させた場合とさせない場合のバッテリ減少の差分とする．位置の操作種別の収集は最小の時間・距離間隔を要求し，Android フレームワークから可能な限りの高い頻度で位置の通知を受け取るように構成した．また，加速度センサはユーザインタフェース向けの低い感度を設定した．

実験の結果，表 3 に示すとおり，ミドルウェアは要件 R1 と R2 を満たすことが確認された．ミドルウェアの動作によるバッテリ消費は，シナリオ 1 と 2 でそれぞれ 6%と 9%であった．また，CPU 使用率はすべての測定ポイントで 6%以下に収まった．

表 3　操作履歴データ収集によるリソース消費
Table 3　Resource consumption on operation data collection.

デバイス設定 (GPS,WiFi)	バッテリ消費 (減少値)	CPU 使用率 (ピーク値)	メモリ使用量(KB)
シナリオ 1. オン	ミドルウェア動作:90% ミドルウェア非動作: 84% 差分: 6%	app: 1% user: 5% system: 6%	最小：22368 最大：34188 平均：29560
シナリオ 2. オフ	ミドルウェア動作:16% ミドルウェア非動作: 7% 差分: 9%	app: 0% user: 1% system: 1%	最小：21000 最大：28456 平均：21471

さらにリソース消費の分析のため，収集法 1, 2,3 を個別に評価した．収集法 4 のリソース消費はアプリケーションに依存し，収集法 5 は操作データ生成アルゴリズムに依存するため評価対象外とした．表 4 に示す通り，主に端末操作データや端末状態データの収集が可能な収集法 1,2 について，監視の有効状態と無効状態でバッテリ消費と CPU 使用率に差はなく，監視のオーバヘッドは小さいことが確認された．一方，セ

ンサデータの収集が可能な収集法 3 については，GPS と加速度センサのデバイスをオンにした移動状態で 1 時間に 10%のバッテリ減少が見られ，リソース消費への影響が見られた．一方，それぞれをオフにした場合はバッテリ減少が抑えられている．従って，エネルギー節約のため，センサデータの収集の時間範囲や頻度は最小限に限定すべきであるといえる．オンデマンド収集方式を適用することにより，時空間の条件に基づくルールで収集期間や頻度を限定可能である．例えば毎時 10 分間のみセンサデータの収集を行うルールを与えることが考えられる．

操作データの抽出の際の処理は，収集法 1 が Logcat 出力の文字列操作，収集法 2 は Manger クラスや Intent 拡張領域，Provider クラスへのアクセスを含む．いくつかの Manager と Provider のアクセス時間を測定したところ，Manager アクセスの時間は短く，例えば *PackageManager* のアプリケーション名を取得する API 呼び出しを 10000 回行ったところ，全体で 390 ミリ秒であった．一方，Provider アクセス時間はデータベースの件数に依存する．登録件数 166 件のブラウザのブックマーク情報/閲覧履歴の情報取得を 100 回実施したところ処理時間は 1707 ミリ秒であり，1 回あたりの平均情報取得時間は約 17 ミリ秒であった．

表 4　各操作履歴収集法のリソース消費
Table 4　Collection techniques and resource consumption.

収集法	条件	バッテリ残量	CPU 使用率 (ピーク値)
1.Logcat 監視	固定して放置．3 時間測定	収集なしの場合と差はない	左と同様
2.BrodcastIntent 監視	固定して放置．3 時間測定	収集なしの場合と差はない	左と同様
3. EventListener 監視	GPS を収集しつつ移動．1 時間測定	約 8%減少	2%
	加速度情報を収集しつつ移動．1 時間測定	約 3%減少	2%
	GPS・加速度情報を収集しつつ移動．1 時間測定	約 10%減少	3%

次に，選択的暗号化方式に関する暗号化オーバヘッドを評価するため，操作データの記録時間とクエリ時間を測定した．図 4 は書き込むレコードのデータサイズと記録時間を示す．Plain1 は暗号化を行わない 1 つのパラメータを持つ操作データの書き込みを指し，Encryption8 は暗号化を個別に行う 8 つのパラメータを持つ操作データの書き込みを指す．トータルのレコードサイズは 32,256,1024 バイトとしており，パラメータが多いほど 1 つのパラメータフィールドのデータサイズは小さくなる．暗号化を伴わない場合，記録時間は 40 ミリ秒程度であったのに対し，記録時間における暗号化オーバヘッドは，いずれのレコードサイズであっても 26 ミリ秒以下であり，実用性が確認された．

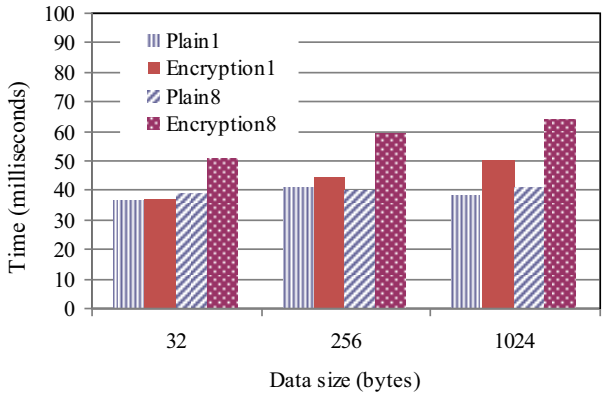

図 4　記録時間に対する暗号化の影響
Figure 4　Effects of Encryption on recording time.

図 5 に，異なるレコード数とパラメータ数におけるクエリ時間に対する暗号化の影響を示す．3 種類のクエリにおける処理時間を比較した．No Parameters は操作種別のみを指定したクエリであり，登録されている全レコードからマッチする操作種別のレコードを抽出する．操作種別に加えて，One Parameter は 1 つのパラメータ値，Two Parameters は 2 つのパラメータ値を指定したクエリである．クエリの処理はまず操作種別の指定により，4000 レコードを初期登録した Plain4000 と Encryption4000 では 78 個のレコードが取得され，10000 レコードを初期登録した Plain10000 と Encryption10000 では 197 個のレコードが取得された．そして取得された各レコードの 1 つあるいは 2 つのパラメータが抽出され，クエリのパラメータ値と比較した結果，一致した場合にそのレコードが抽出される．パラメータ値の比較の際，Encryption4000 と Encryption10000 ではパラメータが復号される．結果として，クエリ時間は復号されるパラメータの数に依存する傾向が見られた．復号処理がない場合，レコード数の増加はクエリ時間に大きな影響は与えていない．なお，No Parameter と One Parameter のクエリ時間の差は，One Parameter と Two Parameters の差よりも大きく，レコードの取得時間が影響していると考えられる．従って，応答性を確保するためには，クエリが暗号化されたフィールドを参照する場合，クエリの時間範囲，すなわち参照するレコ

ード数を限定することが必要不可欠となる．また，暗号化設定情報における暗号化するフィールドをその重要性に基づき，必要最小限にすることが必要と考えられる．

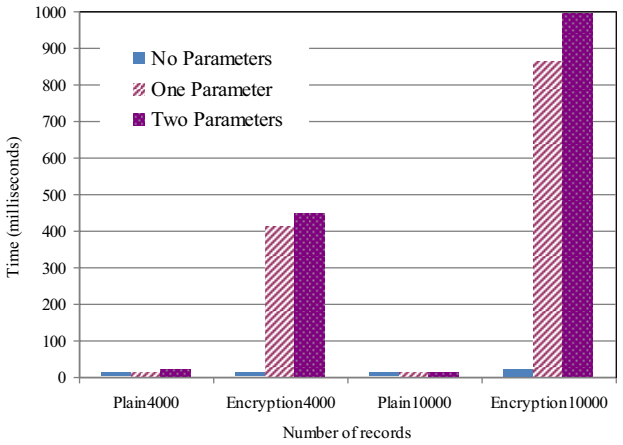

図 5　クエリ時間に対する暗号化の影響

Figure 5　Effects of encryption on query time.

5. まとめと今後の課題

本稿は操作履歴を活用したアプリケーション開発とリソース利用の効率化のため，操作履歴ミドルウェアの設計とプロトタイプを示した．本ミドルウェアは，P1.ユーザへの収集状況の通知，P2.選択肢の提供と許諾の確保，P3.必要最小限のデータ収集，P4.認証や暗号化を含む適切なセキュリティの確保の 4 つのプライバシ保護方針に対応した機能を備える．必要最小限のデータ収集に関して，操作データ種別，保存期間，収集期間や範囲を必要最小限とするためのオンデマンド収集方式を提案した．さらに携帯端末のリソース制約への適応のため，操作履歴データベースのレコードにおける指定フィールドのみの暗号化を行う選択的暗号化方式を実装した．プロトタイプについて，収集処理における監視オーバヘッドや記録及びクエリに関する暗号化のオーバヘッドを評価し，オンデマンド収集方式の適用性や選択的暗号化方式の実用性を確認した．

今後の課題として，様々な操作履歴活用アプリケーションを構築し，ミドルウェアの機能性を評価する．また，携帯端末以外の環境のセンサやデバイスなど，分散した操作データの収集を管理できるようにミドルウェアの拡張を行う．

参考文献

1)　M. Langheinrich, “Privacy by Design: Principles of Privacy-Aware Ubiquitous Systems,” Ubicomp 2001: Ubiquitous Computing, Lecture Notes in Computer Science,　vol. 2201, Springer-Verlag, Berlin, 2001, pp.273–291.

2) Dey, A. K. and Abowd,G. D.,” Toward a Better Understanding of Context and Context-Awareness”, In Proceedings of　the CHI2000 Workshop on The What, Who, Where, and How of Context-Awareness, 2000.

3) J. I. Hong, J. A. Landay, “An architecture for privacy-sensitive ubiquitous computing”, In Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services, pp. 177-189, 2004.

4) Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, N. Sadeh, “A framework of energy efficient mobile sensing for automatic user state recognition”, In Proceedings of the 7th international conference on Mobile systems, applications, and services,　pp. 179-192, June 2009.

5) F. Rahman, M. Hoque, F. A. Kawsar, S. I. Ahamed, Preserve Your Privacy with PCO: A Privacy Sensitive Architecture for Context Obfuscation for Pervasive E-Community Based Applications, In Proceedings of the IEEE Second International Conference on Social Computing (SocialCom) , pp. 41-48, August 2010.

6) William Enck, Peter Gilbert, Byung-Gon Chun, Landon P. Cox , Jaeyeon Jung, Patrick McDaniel, and Anmol N. Sheth. TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones. Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI), October 2010.

7) A. K. Dey and G. D. Abowd, "The Context Toolkit: Aiding the Development of Context-Aware Applications," presented at Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland, 2000.

8) Raento, M., Oulasvirta., A., Petit, R., and Toivonen, H.ContextPhone: A prototyping platform for context-aware mobile applications. IEEE Pervasive Computing Special Issue on The Smart Phone, 4, 2 (2005), 51-59.

9) Z. Zhuang, K. H. Kim, J. P. Singh, “Improving energy efficiency of location sensing on smartphones”, In Proceedings of the 8th international conference on Mobile systems, applications, and services, pp. 315-330, June 2010.

10)　T. Nakagawa, T. Yoshikawa, C. Doi, K.Ohta, C. Noda., and H. Inamura, “Cellphone Usage Support Function based on Operation History,” Proceedings of 5th International Conference on Mobile Computing and Ubiquitous Networking, April, 2010.