RM-012

ストレージの暗号化による仮想プライベートデバイスの提案

Virtual Private Device: Encryption for Storages

関　良明†　　小田　哲†　　小林　鉄太郎†
Yoshiaki Seki　　Satoshi Oda　　Tetsutaro Kobayashi

1．まえがき

携帯電話，PDA，ノート PC に代表されるモバイル端末は，常に起動した状態もしくはストレスなく起動できる簡便性の高さにより，所有者の数がますます増加している．モバイル端末を含むコンピュータと利用者の関係性は，1 台の大型計算機を複数利用者で共有する”one-for-many”モデルから，1 つのコンピュータを 1 人の利用者が所有する”one-for-one”モデル，複数のコンピュータを 1 人の利用者が所有する”many-for-one”モデルを経て， 1 つのストレージを中心に複数のコンピュータを複数の利用者が使い分ける”many-for-many”モデルへ移行していると考えられる（図 1 参照）．また，モバイル端末の高機能化や記憶容量の大容量化が進み，利用者の連絡先やメール，予定表などのパーソナル情報のほか，今まで企業内の物理的にセキュリティが保障された端末で扱われていた業務情報などの重要な情報資産も，モバイル端末に保存して持ち歩くようになってきた．しかし，タスクによって 1 つの端末を選択して利用しているため，端末ごとにそれぞれ異なる情報が蓄積されており，携帯電話，PDA，ノート PC それぞれにデータが分散しがちとなる問題がある．このような問題を解決するためには，全ての情報をネットワークストレージに蓄積する方法[1]や，物理的に情報を携帯する方法がある．

情報をネットワーク上にストレージする場合は，Google の Gmail や Microsoft の Hotmail など，ネットワーク上の大きな容量のストレージが企業から提供されており，個人ユースを中心に盛んに用いられ始めている．しかし，パーソナル情報を含むデータを企業に預けることを懸念する考えもある．また，企業ユースにおいては，ネットワーク上にデータを保存する Amazon の 3S: Simple Storage Solution などのユーティリティコンピューティングサービスが提供されている．しかし，業務情報を一企業に対する信頼のみに基づいて扱うことは望ましくないという考えもある．

一方，情報を物理的に携帯する方式では，ネットワークを利用せずに，USB メモリや外付け HDD，SD カード，コンパクトフラッシュといった小型のストレージデバイス（以降は可搬記憶媒体と標記）を用いて端末間で情報をやり取りする手段があげられる．しかし，可搬記憶媒体はその性質から容易に持ち運びができるため，厳密な管理が非常に困難である．また，可搬記憶媒体の紛失や盗難が発生した際に，十分な時間や環境を使って攻撃されることを考慮したセキュリティ設計が必要となる．そこで，多くの企業では，個人情報保護法における個人情報適正管理や，不正競争防止法における営業秘密の管理などさまざまな観点から，可搬記憶媒体が紛失したり盗難した際の対策として，可搬記憶媒体を利用する際には，指紋認証機能を搭載したデバイスの利用を義務付けたり，パスワードを用いてアクセス制限を行うことが当たり前になっている[2].

「共通の作業を行っている，または共通の目標をもつ人のグループを支援し，共有作業環境へのインタフェースを提供するコンピュータベースのシステム」が，1991 年に Clarence Ellis が示したグループウェアの定義であり[3]，現在最も広く認められている[4]．これまでグループウェアは人の振る舞いに重点を置いて研究されてきた．しかし，本研究ではパーソナル情報や業務情報などの重要な情報資産が保存されているストレージに着目し，「共通の作業を行っている，または共通の目標をもつ人のグループに属する複数の利用者が情報共有に用いているストレージ」を中心としたシンプルなグループウェアを，暗号技術を活用して実現しようとするものである．グループウェアの古典的な分類に従えば，非リアルタイム型（蓄積型）グループウェアに属する．また，グループの規模/構成に基づく分類では，企業での利用を想定したエンタープライズウェアもしくは，家庭での利用を想定したホームウェアと考えられる．

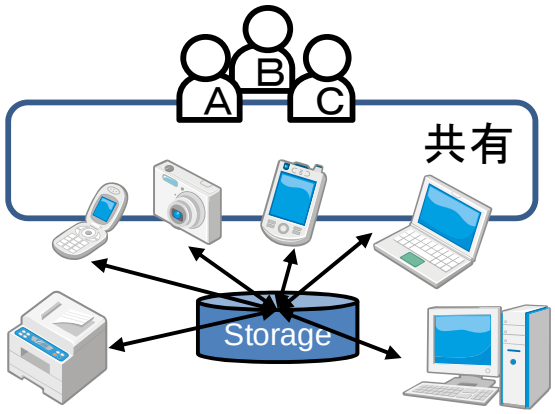

図 1．ストレージを中心とした many-for-many 利用

本稿では，可搬記憶媒体の誰もが読み書き可能なストレージ領域に対して，暗号技術を用いてアクセス制限を行い，あたかも VPN: Virtual Private Network のように，利用できる端末を限定したストレージデバイス（VPD: Virtual Private Device：仮想プライベートデバイス）の実現を考える．VPD を各種アプリケーションと連携することで，読込専用機能，自動バックアップ機能，管理者による利用端末の管理（追加/削除/一覧など）機能などを容易に実現することができる．セキュリティ要件として，その情報にアクセスする権限を持たない端末や利用者（例えば，落とした可搬記憶媒体を拾った人や，盗み見ようとしている人）は，暗号化された状態でしかアクセスできない仕組みを実現す

†日本電信電話（株）情報流通プラットフォーム研究所

る．機能要求としては，権限を持つ端末や利用者であれば，セキュリティ機能がない可搬記憶媒体と同等の操作性（パスワード入力不要であることや，処理速度など）を保つことを目標とする．

以下，2 章で関連研究を主にセキュリティの観点から概観し，3 章で本研究が想定する利用状況を設定し，4 章で実現方式を検討して，可搬記憶媒体を USB メモリに特定したシステムの実現と性能評価を論じる．さらに，5 章で開発システムの適用領域と VPD の可能性を考察する．

2．関連研究

指紋認証機能やパスワードなどを用いて可搬記憶媒体のストレージ領域を暗号化する技術は，既に製品として多数流通している[5]．しかし，本稿で提案する方式は，指紋認証機能やパスワードなどは用いずに，ストレージ領域の暗号化に用いる共通鍵を公開鍵暗号によって管理するものである．この考え方は，インターネット通信においてクライアント/サーバ間の Web アクセスデータを暗号化する仕組みとして，一般的に用いられている SSL(Secure Socket Layer)/TLS(Transport Layer Security)に類似している．しかし，可搬記憶媒体は，CA: Certificate Authority: 認証局などのインフラを利用できないため，鍵管理が課題となる．

そこで，以下では，可搬記憶媒体内部の暗号化に利用する共通鍵ブロック暗号 Camellia（カメリア）と，その鍵の管理に利用する公開鍵暗号方式を用いた鍵カプセル化メカニズム PSEC-KEM，およびデータ改ざんの有無を検証するメッセージ認証方式 HMAC とデジタル署名 ECAO を紹介する．

可搬記憶媒体内部の暗号化に利用できる共通鍵ブロック暗号として，Camellia がある[6],[7]．Camellia は技術的に高い安全性を有するとともに，処理性能と実用性についてもたいへん優れており，低機能型 IC カード（8 ビット CPU）から汎用 PC（32 ビット CPU），サーバ系（64 ビット CPU）まで，実装環境に応じて使用する命令セット，ROM/RAM メモリ使用量を適切に選定して，ソフトウェアを実装できる[8]．ハードウェア実装においても，高速実装はもとよりコンパクトかつ低消費電力型の実装が可能であり[9]，多様な端末と共に利用される可搬記憶媒体に最適な暗号技術である．

鍵の管理に利用する鍵カプセル化メカニズムとして，公開鍵暗号方式を用いた PSEC-KEM: Provably Secure Elliptic Curve encryption with Key Encapsulation Mechanism がある[6],[10]．PSEC-KEM は，PSEC をベースとした鍵配送を目的とする公開鍵暗号である．PSEC は，安全性の高さが数学的に証明された日本国内初の楕円曲線上の離散対数問題に基づいた公開鍵暗号方式で，RSA 暗号などと比較して，短い鍵長でも十分な安全性を確保できる．このため，より高速な実装が可能であり，記憶容量に制限のある可搬記憶媒体に最適な暗号方式である．また，鍵カプセル化メカニズムとしての KEM は，共通鍵暗号で利用するセッション鍵を配送するために使用され，実際のデータやコンテンツの暗号化は，当該セッション鍵を用いた AES や Camellia などの共通鍵暗号からなるデータカプセル化メカニズムによって実行される．

通信経路上で，メッセージの変化や改ざんがないことを確認するメッセージ認証コード MAC を付与する方式として，HMAC: Keyed-Hashing for Message Authentication Code がある．HMAC は共有鍵と暗号学的なハッシュ関数（例えば MD5，SHA-1，SHA-256 など）を用いて構成され，その仕様は，RFC2104 などで規格化されている．また，安全性の高さが数学的に証明された楕円曲線上の離散対数問題ベースのメッセージ回復型デジタル署名として ECAO: Elliptic Curve Abe-Okamoto signature がある[6],[11]．ECAO は，ハッシュ関数の出力がランダムであるという仮定と楕円曲線上の離散対数問題が解読困難であるという仮定のもとで署名の安全性を厳密に評価することができ，国際標準化機関 ISO/IEC15946-4 で規格化されている．

3．利用状況設定

可搬記憶媒体を情報共有の手段として利用する場合，従来の通信やストレージに基づくセキュリティモデルでは考慮されていない事象がおこる．本章では，可搬記憶媒体を暗号化する安全性について，まず利用シーンについて状況設定を行い，セキュリティリスクを議論する．

3.1 利用シーン

ネットワークが利用できない環境において，可搬記憶媒体を利用して複数の端末間で情報を共有することを考える．以下の記述の（　）内は図 2 の端末/人/データの記号を指している．その可搬記憶媒体は限られた端末（a/b/c）でしか利用できないが，限られた端末であれば誰（A/B/C/D）でも自由に情報を読み/書き/追記できるものとする．当然，可搬記憶媒体がアクセスされる順番は予め決めることはできない．このため，端末から書き込んだ内容が必ず特定の人に伝わるとは限らない．

ちょうど可搬記憶媒体を回覧板のように利用して，回覧内容を共有するイメージである．いわゆる回覧板と異なる点は，回覧板は情報を最初に発行したものが明確（x に固定）であり，その情報の中身が書き換わることなく利用者に閲覧されるのに対して，今回の利用シーンでは常に情報が最新のものにアップデートされており（x/x’/x’’/x’’’と遷移），最初の情報から書き換わった状態を閲覧することもありうるという点である．

可搬記憶媒体を利用して情報を共有する場合の特徴を，以下に例とともに列挙する．

（1）可搬記憶媒体にアクセスした時点で，最後に書き込まれた内容のみが読み込まれるため，暗号文の全体を一人の人が書いているわけではない．
例：ある利用者（C）が読み込んだ内容は，他の複数利用者（A/B）が書き加えたものである．

（2）情報がアクセスされる順番か予想できないため，どの時点であっても正しい状態と正しくない状態を区別できなくてはならない．
例：利用者端末（a/b/c）がアクセスする順序は事前に決められず，途中で不正な端末（d）からのアクセスがあるかもしれない．

（3）復号が何度行われても安全でなくてはならない．
例：利用者端末（a/b/c）は毎回復号している．

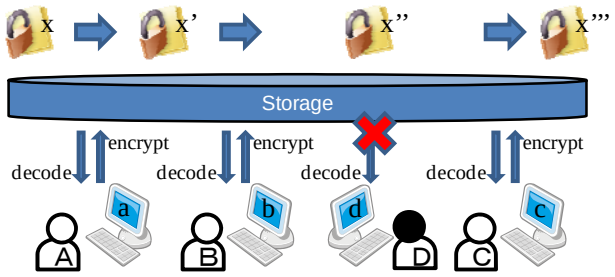

図 2. 可搬記憶媒体ストレージ

3.2 セキュリティリスク

前節で設定した利用シーンにおけるセキュリティリスクについて，情報セキュリティの CIA: Confidentiality / Integrity / Availability（秘匿性/完全性/可用性）[12]に則って議論する．

本スキームには，グループに属する利用者端末，およびグループに属していない人やこのスキームを破ろうとする人（以降は攻撃者と標記）が存在する．議論を簡単にするために，攻撃者は，利用者端末の中へアクセスできないものと仮定する．逆に，可搬記憶媒体に対しては，利用者端末による読み/書き/追記が実行されている期間を除き，攻撃者が可搬記憶媒体に対して自由に読み/書き/追記できると仮定する．これは，実際にグループ内で可搬記憶媒体を運用している際に，可搬記憶媒体が利用者端末間を転々とし，誰の管理下に置かれているわけでもない，極端な例としては，可搬記憶媒体が机の上に放置されている状態を指し，最も管理が緩い状況を想定している．この最も管理が緩い状況においても安全なスキームであれば，より管理された環境で利用されても安全であると言える[13].

- **秘匿性(Confidentiality)**

本スキームにおける秘匿性とは，攻撃者がその可搬記憶媒体に書かれている内容を一切知ることができないことである．攻撃者は，あるタイミングで可搬記憶媒体を搾取し内容を盗み見るだけではなく，日々更新されている情報の差分などの情報を得ながら，可搬記憶媒体に保存されている情報を推測しようとする．このような攻撃に対しても，情報が一切漏れないことを保証すべきである．

- **完全性(Integrity)**

本スキームにおける完全性とは，攻撃者がその可搬記憶媒体に何かしらの情報を書き込んだ場合，それを利用者端末が検知できることである．例えば，記憶された値を書き換えるような改ざんだけでなく，意味のないノイズを書き込んだとしても，それがノイズであると検知できることである．さらに，ある時点で書かれていた正しい暗号文の一部を別な媒体に保存しておき，別の段階で書き戻しても検知できる必要がある．これは，例えば，貯金の残高部分だけを過去の情報に書き戻すような攻撃を想定している．このような攻撃に対しても，改ざんを検知できることを保証すべきである．

- **可用性(Availability)**

本スキームにおける可用性とは，利用者端末が最後に書き込んだ可搬記憶媒体の状態を，可搬記憶媒体の状態に関わらず復元できることである．可搬記憶媒体は，常に利用者端末の管理下にあるとは限らないため，紛失や故障によりデータが読み出せなくなるかもしれない．このような状態になっても最後の状態にリカバリできることを保証すべきである．また，通常，より管理された環境では予防や回復も考慮する必要があるが，ここでは最も管理が緩い状況を想定している．

4. 実現方式の検討

前章で議論した安全性を実現する暗号方式について検討する．本章ではまず，面らの議論[14]に従い，暗号化を行うレイヤについて検討を行い，次にシステムの実現方式について議論する．

4.1 暗号化のレイヤ

ストレージに対して暗号化を施すレイヤは，大きく分けて 2 つあげられる．1 つは ZIP 暗号に代表されるようなファイル単位/ディレクトリ単位のように保護するデータ単位で暗号化する filesystem-level encryption である．もう 1 つは，OS などのシステムも含んだレイヤでディスク全体を暗号化する Full disk encryption である．

前者は，システムに関与せず構成することができるため，開発が容易となる利点がある．また，システム稼動中も復号処理を行わない限り，データが暗号化されたままの状態で保管されているため，意図しない情報漏洩を防ぐ効果も期待できる．

一方，後者は，システムを起動する際に復号処理が必要なため，システムの外に暗号処理を実装する必要がある．その代わり暗号処理が，システムやその上位アプリケーションからは透過的に行われるため，従来利用しているアプリケーションを一切変更する必要がない利点がある．

本稿では，可搬記憶媒体における暗号化について両者の利点を併せ持つように，両者の中間的な性質を持つデバイスドライバレベルでの暗号化の利用を考えた．それを実現するために，Windows におけるフィルタドライバの仕組みを利用した．フィルタドライバは，Windows のカーネルモードで動作するデバイスドライバの一種でデバイスとデバイスドライバの間や，デバイスドライバとアプリケーションの間に位置して動作するプログラムである．このフィルタドライバを利用して，従来のファイルと OS との間に仮

想的なファイルシステムを作り，そのフィルタドライバ内部で暗号化/復号/改ざん検知/バックアップなど様々な処理を実施することで，可搬記憶媒体暗号化システムを構築した．可搬記憶媒体暗号化システムの構成を図 3 に示す．

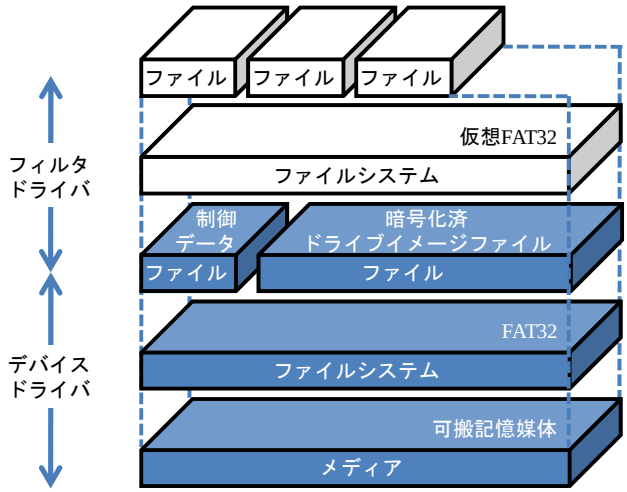

図 3. 可搬記憶媒体暗号化システムの構成

フィルタドライバを経由せずに可搬記憶媒体にアクセスすると，通常の可搬記憶媒体のファイルシステム（例えば FAT32）に暗号化されたドライブイメージを確認することができる．一方，フィルタドライバを経由すると，ドライブイメージが復号されて，かつファイルシステムをマウントした状態で見ることができる．

4.2 システム設計

本節では，可搬記憶媒体を USB メモリに特定して，システム開発した具体例を論述する．USB 暗号メモリを中心とした周辺 PC と，その利用者の関係を図 4 に示す．

ここでは利用者の PC を，以下の通り，暗号ドライブ管理者 PC，ユーザ管理者 PC，利用者 PC の 3 種類に分類し，その総称を利用者端末と定義する．

- **暗号ドライブ管理者 PC**
  - ・暗号ドライブを初期化した利用者が権限を持つ端末．
  - ・暗号ドライブの初期化，リカバリ機能を利用でき，利用者端末の追加と削除，ドライブの利用申請やマウント/アンマウントのすべての機能を利用できる．

- **ユーザ管理者 PC**
  - ・暗号ドライブ管理者 PC から，利用者端末の追加と削除をできる権限を与えられた利用者の端末．
  - ・利用者端末の追加と削除，ドライブの利用申請やマウント/アンマウントが利用できる．

- **利用者 PC**
  - ・一般的なユーザの端末．
  - ・ドライブの利用申請やマウント/アンマウントが利用できる．

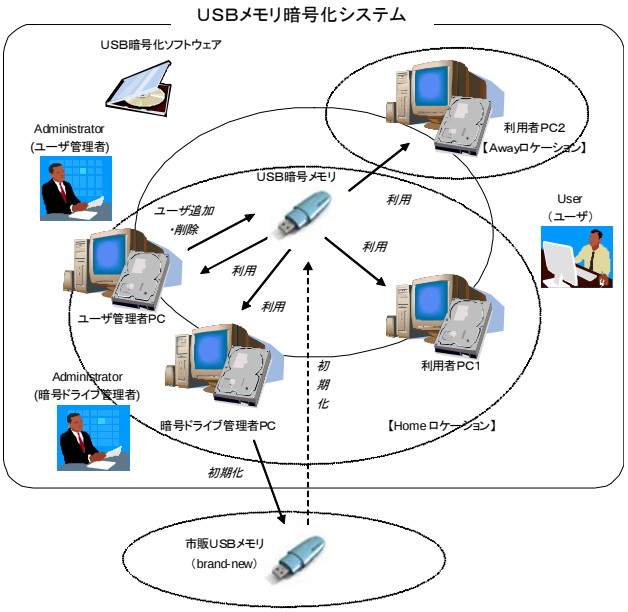

図 4. システム構成の概念図

4.3 利用シーケンス

前節のシステムを使う利用者の画面遷移図を図 5 に示し，一連の流れを以下の通り実現した．ここで，利用者端末やこれから利用者端末にしようとする端末には，暗号化に必要なソフトウェアがインストールされているとする．例えば，図 3 の制御データや暗号化済みドライブイメージファイルと同じレイヤに，暗号化を施さないままソフトウェアのインストーラを格納しておくと，必要に応じてソフトウェアがインストールできる環境を構成できる．そして，ソフトウェアをインストールする際に，利用者端末ごとの秘密鍵を生成する．

- **初期設定**

共通鍵暗号で利用する鍵（以下，データ鍵）を生成し，データ鍵を用いて，ソフトウェアから利用する暗号ドライブのフォーマット（ドライブイメージファイル/制御データファイル/リカバリファイルの作成）を行う．

なお，この初期設定を実施した利用者端末が暗号ドライブ管理者 PC となる．

- **通常利用**

フィルタドライバを介すことにより，ドライブイメージと制御データを通常のドライブとしてマウントする．通常の USB メモリのマウント/アンマウントと同時に行うことで，利用者は通常の USB メモリを操作しているのと同様の操作感で，暗号 USB メモリを利用することができる．フィルタドライバを介してファイルを書き込むと，暗号化処理/改ざん検知のための制御データ HMAC の付与の他，リカバリファイル（暗号ドライブに書き込んだ内容をそのまま HDD にも記録する）の作成も同時に行う．

● 利用者の追加

利用者端末にしたい利用者 PC は，まず利用申請（技術的には，端末の公開鍵を登録）を当該 PC から実施する．暗号ドライブ管理者 PC もしくはユーザ管理者 PC は，この利用申請を受理（技術的には，USB メモリ内に格納されている利用者端末の公開鍵を用いて，利用者端末のみがデータ鍵を導出できるように暗号化）する．

● 利用者の削除

ある利用者 PC を利用者端末から除外し，初期設定と同様の処理を行う．簡易的なセキュリティを実現するためには，利用者の追加処理で既申請を除去する（技術的には，公開鍵で暗号化された暗号文を削除する）のみでよいが，その場合は，利用者端末ではなくなった端末でも，アクセスできる可能性が残る．

● リカバリ

全利用者端末からリカバリファイルを集める．そして書き込まれた時間が古い順にリカバリファイル（リカバリデータの断片）を上書きすることで，最後に利用者端末によって書き込まれた USB メモリの状態を復元する．暗号ドライブ管理者 PC は，このリカバリされたドライブイメージファイルを復号し，USB メモリに書き込まれた中身のファイルを復元する．なお，利用者端末への攻撃者のアクセスは，本稿のスコープ外ではあるが，リカバリファイルは暗号文であるため，暗号文とその鍵を導出できる情報が同じ場所にあることは好ましくない．そこで，本システムでは，鍵を利用者端末側と USB メモリ内部に秘密分散して保存することにより，USB メモリが接続されているときしか，USB メモリ内部のファイルを復号できなくしている．

また，利用者端末が USB メモリと鍵を秘密分散をしていると，USB メモリが失われた時/故障した時に復元できなくなるため，暗号ドライブ管理者 PC のみは，端末の中に完全な形で鍵を保管している（このため，リカバリは暗号ドライブ管理者 PC しか実行できない）．

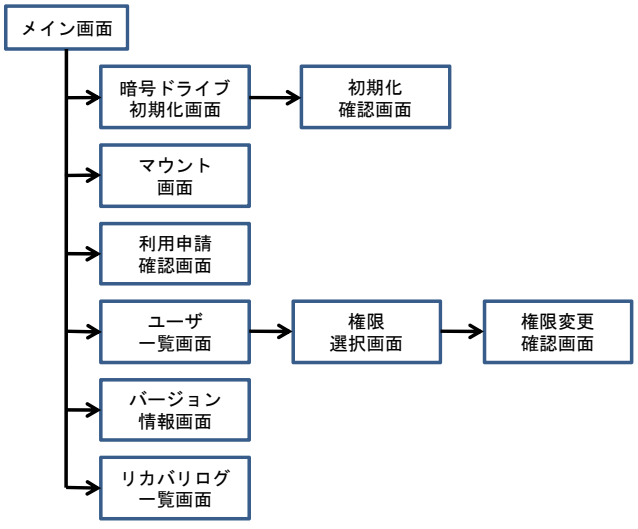

図 5. 利用者の画面遷移図

4.4 ユースケース

前述の利用シーケンスの構成要素となるユースケースの主要なものを以下に示す．

(1) 初めてアプリケーションを起動
- 安全性が数学的に証明された楕円曲線による署名方式 ECAO の鍵を作成し，利用者端末に署名用秘密鍵 SSK，署名用公開鍵 SPK を保存する．

(2) USB メモリを初期設定
- USB メモリを初期設定し，メモリ内に暗号ファイルを作成する．
- 暗号ファイルを作成した利用者端末が暗号ドライブ管理者 PC となる．
- Windows における OS のプロダクト ID など端末を特定する情報のハッシュ値と XOR した PSEC-KEM 秘密鍵 PSK，および PSEC-KEM 公開鍵 PPK を USB メモリに保存する．
- この際，暗号ドライブ管理者 PC は，秘密鍵 SK を分割保存しない．
- 暗号ドライブ管理者 PC は初期化の種類を選択し，暗号ドライブの初期設定を行う．

(3) 暗号ドライブを利用申請
- USB メモリに利用者端末を設定し，利用を申請する．
- 利用申請を行うと，PSEC-KEM および証明書用の利用者端末の公開鍵 PK，秘密鍵 SK を生成し USB メモリと利用者端末のローカル HDD に保存する．このとき，利用者端末の秘密鍵 SK は USB メモリ，ローカル HDD，OS のプロダクト ID がそろって初めて利用者端末の SK となるように秘密分散する（図 6 参照）．

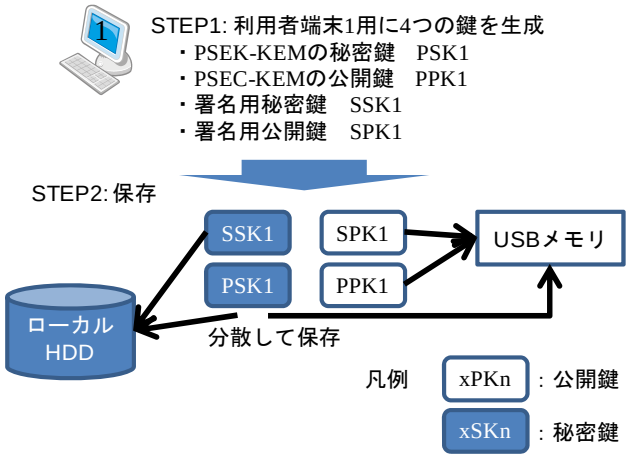

図 6. PSEC-KEM の鍵生成概念図

(4) 暗号ドライブを利用承認
- USB メモリ上に作成された暗号ドライブがマウントできるように，利用の承認を行う．
- 承認は，利用申請された利用者端末の公開鍵 PK に対して，ユーザ管理者 PC の SSK で署名する．

・利用承認時に利用者端末の権限をユーザ管理者 PC か利用者 PC に設定することができる．

(5) 利用者端末を削除
・USB メモリ上に設定された利用者端末を削除する．暗号ドライブ管理者 PC またはユーザ管理者 PC は，削除の方式を選択する．
・通常の利用者端末の削除は USB メモリ上の利用者の公開鍵 PK を削除し，新しい Camellia のデータ鍵で暗号ドライブを再暗号化する．

(6) 暗号ドライブが含まれる USB メモリを利用者端末に接続
・初期化済みの暗号ドライブが含まれる USB メモリを利用者端末に挿入する．
・OS の自動実行で選択できるように，インストーラにてレジストリを変更する．
・自動実行により起動したアプリケーションは USB メモリの暗号ドライブをマウントし，表示を行う．
・OS の自動実行を設定していない場合は，アプリケーションの起動やマウントは手動で実行する．

(7) 暗号ドライブをマウント
・暗号ドライブをマウントする．
・暗号ドライブがライトプロテクトされている場合は，リードオンリーでマウントを行う．

(8) データを書き込む
・暗号ドライブにデータを書き込む．
・リカバリ用のログをローカル HDD に保存する．

4.5 性能評価

暗号ドライブ管理者 PC が USB メモリのストレージ領域を初期化する際に，通常では記憶容量に比例した時間を要する．これは，3.2 節で議論した秘匿性や完全性を守るために，USB メモリ中の情報の有無が判明（＝情報が漏れていると考えられる）しないように，また，ヘッダー情報とそれに対応する内容が一致するように，いったんドライブ内の全てのデータ領域を null で暗号化するからである．

しかし，記憶容量の飛躍的な大容量化に伴い，初期化に要する時間が次第に現実的ではないものとなってきた．一方，ドライブを初期化する際には，通常のフォーマットではなく，ヘッダー情報のみを初期化するクイックフォーマットの利用が増えてきている．そこで，本システムにおいても，通常フォーマットの他に，FAT32 の管理領域のみを書き換えるクイックフォーマットを採用した．

通常フォーマットが全てのデータ領域に暗号処理を行うのに対して，クイックフォーマットを利用した高速フォーマットは，ヘッダー部分のみに暗号処理を実行するため，非常に高速な処理が可能となる．そこで，USB メモリに対して，通常フォーマットと高速フォーマットの 2 つの初期化方式，それぞれが要した時間を測定し比較した結果を表 1 に示す．

表 1. フォーマットの処理時間比較

回数	1	2	3	4	5	平均
通常	602秒	597秒	609秒	614秒	611秒	607秒
高速	22秒	25秒	20秒	28秒	31秒	25秒

- **性能測定環境**
  (1) ハードウェア（PC）
  ・CPU：Intel Core2Duo 2.4GHz
  ・メモリ：2048MB
  ・HDD：SATA160GB
  (2) ソフトウェア（PC）
  ・OS：Windows XP Professional SP3 (32bit 版)
  ・OpenSSL：バージョン 0.9.8.d
  (3) USB メモリ： PATRIOT XT 32GB

- **測定方法**
  ・各フォーマット方法にて 5 回測定し，平均時間により性能の比較を行う．
  ・フォーマットサイズは，2GB(2048MB)で実施．

USB メモリへの実容量分の書き込みが必要な通常フォーマットと比べ，FAT32 の管理領域のみを書き換えることでフォーマットを実現する高速フォーマットの方が，処理時間を圧倒的に短縮できることが測定結果から確認できた．通常フォーマットのように厳密な秘匿性や完全性を求めない（前述の状態をセキュリティリスクととらえない）場合，初期化時において通常のフォーマットではなく，高速フォーマットを利用できる．ただし，高速フォーマットでは，ヘッダー情報とそれに対応する内容が一致するとは限らないため，何も書かれていない（はずの）データ領域における完全性の保証にも不都合が生ずることも留意する必要がある．なお，初期化以外のデータの読み/書き/追記/ユーザ PC の追加と削除は，セキュリティ機能がない場合と比べて有意な性能劣化は見られなかった．

5. 適用領域

本章では，可搬記憶媒体を USB メモリに特定して開発した USB メモリ暗号化システムの適用領域および，ストレージを中心に複数のコンピュータを複数の利用者が使い分ける”many-for-many”モデルにおいて，ストレージの暗号化による仮想プライベートデバイスの可能性を考察する．

5.1 USB メモリ暗号化システムの適用領域

本システムの訴求対象は，以下のような環境の利用者である．
(1) ルールによる情報漏洩対策はあるが，システムによる防止策の集約化が期待できない小規模な組織環境に所属する利用者．
(2) セキュリティ対策は必要だと思っているが，なるべく通常運用の手間を軽減したい利用者．

(3) 限定した端末間の情報共有に USB メモリを利用している利用者.

上記に対して以下の訴求点がある（図 7 参照）.
(1) 通常の利用時には，パスワードの認証，指紋の照合などが不要である.
(2) 悪意ある者に取得されても，128bit 鍵で守られているので安全である.
(3) USB メモリを紛失したり破損したりしても，リカバリが可能である.

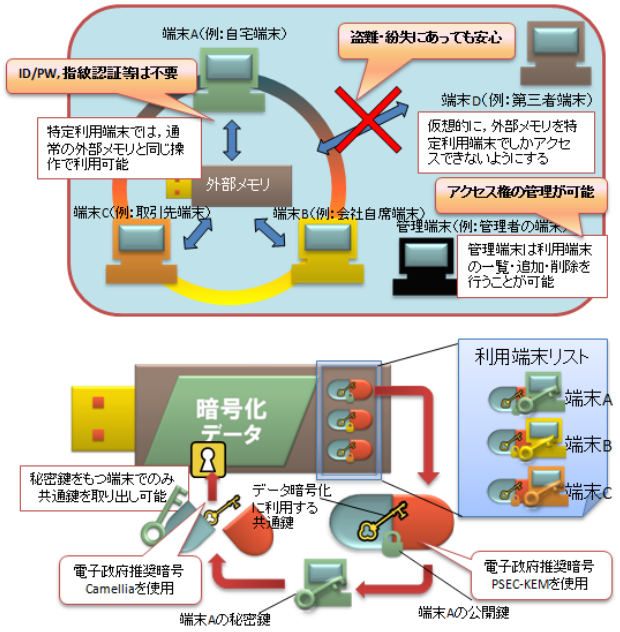

図 7. システムの利用イメージ

具体的な利用ケースを，その背景/既存製品の機能/本システムの特徴とともに以下に 3 つ示す.

● **利用ケース 1：USB メモリの紛失/破損対策**

(1) 背景
- USB メモリの持ち出しが必要だが，今以上にセキュリティ保護の運用に手間をかけられない.
- USB メモリを万一紛失した際のデータの保護に既存製品では不安が残る.
- USB メモリを紛失／破損してしまった場合，データが失われてしまう.

(2) 既存製品の機能
- USB メモリを持ち出す際にはパスワードや指紋認証など，持ち出すための設定や処理が必要である.
- パスワードの保護は回数制限によるロック程度である.
- パスワードが判明してしまえばデータを読み出すことができる.
- 紛失や破損をした場合に持ち出したデータは復元できない.

(3) 本システムの特徴
- 端末を限定することで，USB メモリを利用したい PC で安全にすぐに使える.
- パスワードではなく鍵でデータを保護しており，鍵を持たない悪意ある人の解読攻撃を完全に防げる.
- USB メモリの持ち出しの許可/解除を限定した PC で管理できる．解除する際には，ライトプロテクトを外し，HMAC と署名の検証を行うことにより，持ち出された状態から変更がないことを確認できる.
- USB メモリを持ち出す前の状態にデータを復元できる．なお，高速フォーマットされた暗号ドライブではリカバリファイルが存在しないので，持ち出し許可の設定を行うことができない.

● **利用ケース 2：P2P での情報漏えい対策**

(1) 背景
- P2P によるファイル交換ソフトがインストールされている PC への重要情報の持ち出しは危険であり，対策が必要である.

(2) 既存製品の機能
- 既存の暗号 USB メモリは，パスワードが判明すればどの PC でも接続できる.
- P2P ソフトのある PC から社外秘の情報を漏えいしてしまう.

(3) 本システムの特徴
- 接続できる端末を限定しているため，危険な PC に接続して情報を読み出す危険を予防できる.

● **利用ケース 3：チームでの USB メモリ利用**

(1) 背景
- USB メモリをチームで共有し，情報のやり取りを行う場合，運用ルールによる手順が煩雑になる.

(2) 既存製品の機能
- パスワード暗号であるため，以下の運用が必要となる.
  1. パスワードの設定やメンバーへの周知.
  2. 利用者の増減によるパスワードの再設定.
  3. 情報の保護の鍵となるパスワードが多人数に知れるとともに，パスワードが漏えいする可能性がある.

(3) 本システムの特徴
- 管理者が利用許可等について一括で設定を行う.
- パスワードの周知などの煩雑な運用が不要である.

5.2 ストレージの暗号化による仮想プライベートデバイスの可能性

多数の端末がアクセスできる情報に対して，暗号技術を用いたアクセス制御を実現した．前述の USB メモリなどの可搬記憶媒体に適用すると，あたかも VPN のように利用できる端末を限定したストレージデバイス（仮想プライベートデバイス）を実現することができる.

主な特徴は以下の通り.
(1) 可搬記憶媒体の紛失・盗難による情報漏洩に対して，暗号技術を用いて対策を実現.
(2) 管理者によるアクセス可能な端末の管理（追加・削除・一覧）を実現.

(3) 暗号化を透過的に実現しているため，各種アプリケーションで利用可能.
(4) ネットワークが利用できない環境であっても安全な情報共有を実現.
(5) 電子政府推奨暗号の共通鍵暗号 Camellia，公開鍵暗号 PSEC-KEM を利用.

適用可能な主な領域は以下の通り.
(1) 可搬記憶媒体を利用した複数人による情報の共有.
(2) 可搬記憶媒体を利用して持ち出した情報に対して，利用できる端末を限定.
(3) 異なるネットワーク間での情報の安全な受け渡し.

携帯電話，PDA，ノート PC などのモバイル端末のコモディティ化に伴って，企業内の業務情報やパーソナル情報などを蓄積しているストレージ，特に可搬記憶媒体のセキュリティや利便性向上が，ますます重要になってきている．一方，企業や家庭において，社員や家族は，共通の目標をもって日常の業務や生活を営んでおり，複数の利用者が，1 つの可搬記憶媒体を使って情報共有する場面も多い．USB メモリに特定すれば，前節の利用ケース 1 から 3 などがあげられ，他の小型ストレージデバイスとしては，携帯電話やカメラの可搬記憶媒体のように複数の端末間で，複数の利用者による情報共有に用いられている（図 8 参照）．

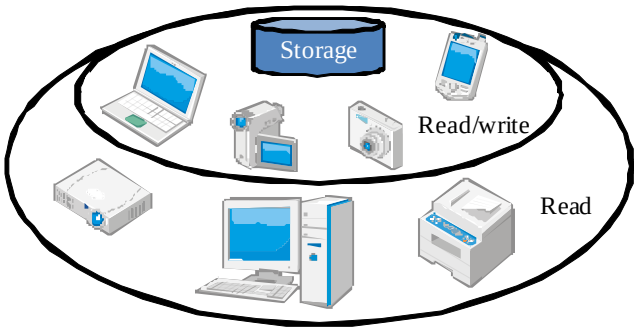

図 8. 機器のアクセス制御

本稿で提案する暗号化による仮想プライベートデバイス技術は，ストレージを中心としたシンプルなグループウェア，特に，指紋認証機能の搭載やパスワード入力が煩雑となるモバイル機器間で，パスワード入力が不要でセキュリティ強度の高い情報共有に大きな貢献が期待できる．

6. まとめ

本稿では，可搬記憶媒体の誰もが読み書き可能なストレージ領域に対して，暗号技術を用いたアクセス制限の利用状況を設定し，実現方式を検討した．ストレージ領域の暗号化に共通鍵ブロック暗号 Camellia を用い，その鍵を管理するために鍵カプセル化メカニズムとして公開鍵暗号方式 PSEC-KEM を用いた．また，データの改ざん検証にメッセージ認証コード HMAC とデジタル署名 ECAO を用いた．ネットワークを利用した通信において通信したい相手だけに情報を伝達する手段として用いられる暗号技術を可搬記憶媒体に適用することにより，あたかも VPN のように，利用できる端末を限定したストレージデバイス（仮想プライベートデバイス）を開発した．

このシステムは，各種アプリケーションと連携することで，読込専用機能，自動バックアップ機能，管理者による利用端末の管理（追加・削除・一覧）機能などを容易に実現することができる．セキュリティ要件として，その情報にアクセスする権限を持たない端末や利用者（例えば落とした可搬記憶媒体を拾った人や，盗み見ようとしている人）は，暗号化された状態でしかアクセスできない仕組みを実現した．機能要求としては，権限を持つ端末や利用者であればセキュリティ機能がない可搬記憶媒体と同等の操作性（パスワード入力不要であることや，処理速度など）を保つことを実現した．

さらに，その適用領域として，具体的な利用ケースを 3 つあげ，ストレージの暗号化による仮想プライベートデバイスのシンプルなグループウェアとしての可能性を考察した．

今後の課題としては，情報家電機器やネットワークサービスと連携した，簡便でセキュリティ強度の高いアプリケーションの開発があげられる．

参考文献

[1] 遠藤大礎, 川原圭博, 浅見徹: “P2P 分散ファイル共有基盤を活用した自己暗号化法によるプライベートストレージ,” 情報処理学会研究報告, 2007-QAI-25(10), 2007.
[2] 独立行政法人 情報処理推進機構(IPA): “情報セキュリティ白書 2008,” 実教出版株式会社, 2008
[3] C. A. Ellis, S. J. Gibbs and G. L. Rein: “Groupware: Some Issues and Experiences,” Communication of the ACM, Vol.34, No.1, pp.38-58, 1991.
[4] 速水治夫, 五百蔵重典, 古井陽之助, 服部哲: “グループウェア,” 森北出版, 2007.
[5] “セキュリティ総覧 2008,” 日経 BP, 保存版 PR 別冊, 2008.
[6] NTT 情報流通プラットフォーム研究所: “NTT R&D セキュリティシリーズ 最新 暗号技術,” 株式会社アスキー, 2006.
[7] “Camellia,”http://info.isl.ntt.co.jp/crypt/camellia/intro.html
[8] 小田哲, 青木和麻呂, 小林 鉄太郎: “Pentium 4 における Camellia の高速実装,” SCIS2006, 2C3-2, Jan. 2006.
[9] 松尾一慶, 阿部公輝: “再構成・拡張可能なプロセッサへのブロック暗号 Camellia の実装,” 情報処理学会研究報告, 2008-CSEC-42(30), 2008.
[10] “PSEC-KEM,” http://info.isl.ntt.co.jp/crypt/psec/index.html
[11] “ECAO,” http://info.isl.ntt.co.jp/crypt/archive/index.html
[12] 宮地充子, 菊池浩明: “情報セキュリティ,” 株式会社オーム社, 2003.
[13] 小田哲, 小林鉄太郎, 関良明: ”ストレージに対する暗号化の一考察,” SCIS2009, 2009.
[14] 面和成: “セキュア VM を支える暗号技術,” 第 2 回セキュア VM シンポジウム～仮想化とセキュリティ～, http://www.securevm.org/svms2-slides/svms2-omote.pdf