情報処理学会論文誌 Vol. 50 No. 9 2218–2227 (Sep. 2009)

切替自在な暗号アーキテクチャ

宮崎真悟[†1] 石川千秋[†2]
越塚登[†1,†2] 坂村健[†1,†2]

本稿では，ユビキタス情報社会における状況認識型の暗号基盤機構,「切替自在な暗号アーキテクチャ」を提案する．本機構は，時事，状況，環境に最適な暗号の選定と，選定した暗号への円滑な切替えを自動化する．これにより，暗号切替えにともなうサービス停止時間を最小限化し，意図したセキュリティ水準を永続的に維持しながら，シームレスなサービス運用が可能となる．本提案は，昨今の著名な暗号アルゴリズムの危殆化にともなう暗号の移行問題や，暗号利用規制が異なる地域・国間の移動にともなう機器内の暗号管理に対して，最適な暗号への自在切替えという技術特性から解決策を与える．

Flexibly Switchable Cryptographic System Architecture

Shingo Miyazaki,[†1] Chiaki Ishikawa,[†2]
Noboru Koshizuka[†1,†2] and Ken Sakamura[†1,†2]

In this paper, we propose the flexibly switchable cryptographic system architecture, which is a cryptographic foundation architecture with a context-awareness in a ubiquitous information society. The architecture smartly selects the most suitable cryptographic module according to the current period, situation and environment and smoothly switches the current module into the selected one. This allows service systems of being protected to reduce a switching time of stopping the service to a minimum and seamlessly produce the service as permanently preserving the security level required. Our proposal of flexibly switching to the most suitable cipher gives an approach to solve cryptographic problems on a migration against compromising of cryptosystems and a cryptographic maintenance of devices transferred to another area or country with a security policy different from the one of the current domain.

†1 東京大学大学院
The University of Tokyo
†2 YRP ユビキタス・ネットワーキング研究所
YRP Ubiquitous Networking Laboratory

1. はじめに

アナログからデジタルデータへの移行や，RFID，センサノード，組み込み機器といった多種多様なコンピュータの生活空間への大量な浸透により，巨大化するデジタル情報空間．そのデジタル情報量は，5 年間で 10 倍の成長速度で，2011 年には 1.8 ZB（1.8×10^{21} Bytes）に達するという予測[1] もある．この急激な成長にともない，情報セキュリティやプライバシ保護に対する企業の責任はますます大きくおしかかり，そのためのコンプライアンスが重要な統治課題となる．

こうした膨張するデジタル情報に対して，意図するセキュリティ水準に最適な情報保護を行い，健全な情報サービスを展開するため，情報セキュリティ技術の中核である暗号技術をいかに適切に利用できるかが重要である．しかし，暗号技術は，それ自体が複雑で高度な専門性を有する技術であることに加え，時事，状況，環境でその取扱いが変化する．

現在，世界がかかえる深刻な課題が，暗号の移行問題である．2004 年以降，MD5 や SHA-1 といった著名な暗号アルゴリズムに関して，相次いで安全性の問題が指摘された．こうした事態を受け，米国立標準技術研究所（NIST）は，世界中の情報システムで現在主流となっている暗号アルゴリズムを 2011 年以降，米国連邦政府機関のシステムで使用しない方針を示した．NIST は，暗号技術に関する実質的なお墨付きを与えている機関であり，世界中に大きな波紋を広げている．

これは暗号の 2010 年問題とも呼ばれ，日本国内でも電子政府や各業界へ大きな影響を与えている．総務省では,「公的個人認証サービスにおける暗号方式等の移行に関する検討会」を開催し，公的個人認証サービスにおける暗号アルゴリズムの移行スケジュール案[9] をまとめている．宇根ら[8] は，本問題が金融分野へ与える影響や対応を考察している．日本データ通信協会では，タイムビジネスの審査基準として，2006 年 4 月以降，ハッシュ関数を SHA-1 から SHA-2 へと移行させた[10]．

暗号を切り替える他の状況として，暗号技術の利用規制が異なる地域・国への移動がある．この際，移動元・先の事情に合わせ，所持機器内の暗号を入れ替えたり，現地で調達したりする必要がある．またユビキタスコンピューティング環境では，接続する多種多様な情報サービスや通信ノードに応じて，利用できる最適な暗号へ切り替える必要もある．

こうした背景の中，本稿では，多種多彩な暗号の中から時事・環境・状況に最適な暗号を自動選定し，自在に切り替えて利活用するための暗号基盤,「切替自在な暗号アーキテクチャ」を提案する．

2. 問題点と既存技術

暗号技術を用いたセキュリティシステムでは，暗号モジュールの保守性と利便性を高め，情報サービスの品質を持続的に保証する必要がある．そのためには，アプリケーションと暗号モジュールとのシステム構造上の関係性が重要である．また，意図するセキュリティ水準を持続的に保持するにあたって，各時点で，好適と評価できる暗号方式の選定や暗号モジュールへの切替え作業を人的に対応するコストやリスクをどう最小限化するかが課題である．

CryptoAPI（Microsoft），CDSA[*1]（Open Group），JCA[*2]（Sun Microsystems）といった著名なセキュリティアーキテクチャでは，いずれも 3 階層のアーキテクチャモデルである．暗号モジュール（サービスプロバイダ），アプリケーションとの間に，サービスプロバイダを束ねるサービスマネージャを仲介させる．サービスマネージャは，暗号プリミティブな機能を隠蔽し，アプリケーションに目的ベースの高度なセキュリティ機能 API を提供する．これにより，暗号モジュールの利用に関する利便性や保守性が飛躍的に向上する．

栃窪ら[4] は，ネットワークを介して，機器内の暗号方式をネットワークを介して更新可能な暗号認証システムを提案した．システムで利用している暗号方式の安全性が低下した場合に，新たな暗号方式へ更新して，システムのセキュリティ水準を一定に保つことができる．

本提案アーキテクチャは，上記高抽象度の API 提供，ネットワークを介した安全な暗号配信機能に加え，時事，環境，状況に最適な暗号モジュールを自動選定する機構をあわせ持つ（**表 1** 参照）．これにより，暗号技術トレンドの変化，機器の移動，アプリケーションからの暗号処理要求に応じた最適な暗号を選定，更新/処理実行を自動化し，暗号資産や暗号処理の履歴情報を一元管理する暗号基盤機能を提供する．本稿で，暗号資産とは，暗号鍵や暗号システムパラメタ，暗号モジュール，暗号モジュールの付帯情報といった暗号に関する情報やモジュール一式をいう．

3. 提案アーキテクチャ

3.1 設 計 要 件

提案する切替自在な暗号アーキテクチャの概念図を**図 1** に示す．本アーキテクチャの設計要件は，以下のとおりである．

(1) 最適な暗号を自動選定　暗号処理要求に対して，暗号トレンドや機器環境情報，暗号エンジン情報に基づき，最適な暗号エンジンを一意に自動選定．速度，安全性，省リソースを優先といった抽象的な暗号選定条件を指定可能．

(2) 円滑な暗号切替え　暗号の詳細を隠蔽化した高抽象度のセキュリティAPI を提供し，上位アプリケーションの都度改変を必要としない，円滑な暗号切替えを提供．暗号の処理カテゴリさえ入力すれば，要求に最適な暗号エンジンを自動選定．

表 1　従来暗号基盤技術との比較
Table 1　The comparison between previous technologies of cryptograhic foundation and our proposal.

技術名	暗号インタフェースの抽象化	暗号のネットワーク更新	最適な暗号の自動選定
一般的な暗号ライブラリ	–	–	–
CryptoAPI（Microsoft）	対応	–	–
CDSA（Open Group）	対応	–	–
JCA（Sun Microsystems）	対応	–	–
リニューアル可能な暗号認証システム[4]	–	対応	–
本提案方式	対応	対応	対応

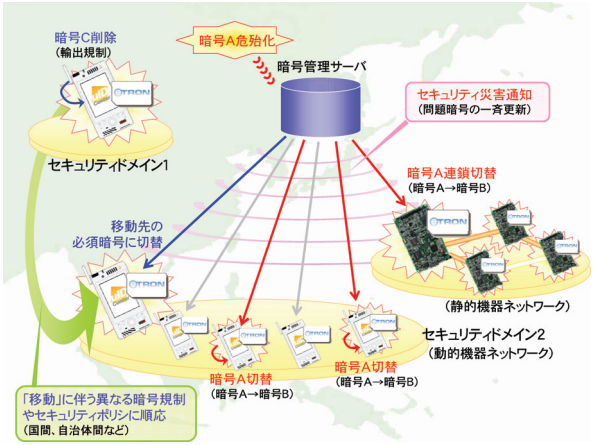

図 1　切替自在な暗号アーキテクチャの概念図
Fig. 1　Concept of the flexibly switchable cryptographic system architecture.

⋆1 Common Data Security Architecture
⋆2 Java Cryptographic Architecture

(3) 暗号資産や暗号処理情報の ID 管理　暗号エンジンやそのプロファイル情報，暗号鍵情報等のすべてに ID を付与し，実体を集中管理．アプリケーションが鍵やパラメータ情報等を意識せずに容易に暗号処理を実行できる．また各暗号処理に対して発行される暗号処理 ID を指定し，過去に実行した暗号処理条件を参照したり，同一条件（同じ暗号エンジンや暗号鍵）で暗号処理を実行可能．

(4) 組み込み機器で軽量動作　各設備に静的に設置された静的な制御機器や人が持ち歩く動的な携帯機器等，資源制約が厳しい組み込み機器でも軽量動作．

上記特徴を有する本アーキテクチャの主要な技術要素である，暗号管理ドメイン，暗号評価標準記述，最適暗号の自動選定機構について，次節より順次説明する．

3.2　暗号管理ドメイン

切替自在な暗号アーキテクチャでは，セキュリティポリシの管理単位となる暗号管理ドメインを定める．暗号管理ドメインは，暗号に関して共通のセキュリティポリシで管理された領域で，大きく，物理的な領域と，非物理的な領域の 2 種類に分類される．物理的な領域は，輸出入や利用の規制等暗号の管理ポリシが異なる地域，国，自治体といった境界である．非物理的な領域は，各種フォーマットや利用暗号等を独自に定め，運用する，医療，製造といった各産業界の情報システムやサービスが形成する領域である．

本アーキテクチャでは，ある暗号管理ドメインへの新規参加や，異なる暗号管理ドメイン間の移動にともなう，機器内にある暗号資産の最適な設定や入れ替えを最大限自動化する．たとえば，移動にともない，使用不可となる暗号モジュールの削除，移動先で必須となる暗号の取得や設定である．なお，各暗号管理ドメインには，当該ドメインの暗号資産を管理する暗号管理サーバ（後述）が少なくとも 1 つ存在する．

3.3　暗号評価標準記述

3.3.1　暗号エンジンと暗号評価標準記述

本アーキテクチャでは，暗号アルゴリズムが実装された暗号処理実体を，暗号エンジンとよぶ．暗号エンジンには，ソフトウェア実装された暗号ライブラリや暗号コプロセッサと一体化した暗号モジュールも含まれる．

この暗号エンジンに対して，定量的な評価値や属性を示す暗号評価標準記述というデータが必ず 1 つ以上存在する．暗号評価標準記述には，その 1 つ 1 つに識別子が付与されている．暗号評価標準記述の例を**図 2** に示す．評価情報には，安全性・速度・リソース使用度に関する定量評価値，評価者や評価日，属性情報には開発ベンダ情報，当該評価記述 ID や対象とする暗号エンジン ID，暗号鍵長に関する利用条件等が含まれる．

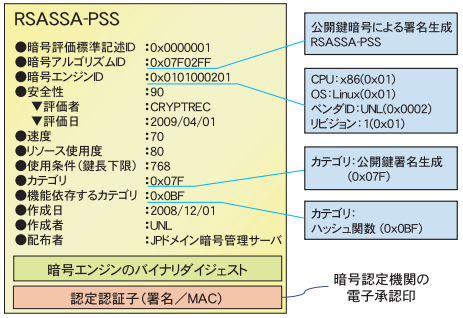

図 2　暗号評価標準記述例（RSASSA-PSS）
Fig. 2　Sample of a cryptographic evaluation profile (RSASSA-PSS).

この暗号評価標準記述は，これを基として最適な暗号選定や暗号保守管理が自動化されるため，本アーキテクチャでは重大な役割を果たす．よって，NIST [⋆1] や CRYPTREC [⋆2] といった信頼ある暗号評価機関の認定署名を格納するデータ構造になっている．また各項目の評価値設定方法に関しては，本システムを運用する国やサービスの評価指針に委ねる．そのため，本稿では評価項目やその標準記述書式のみを提供し，その具体的評価方法に関しては言及しない．各評価方法で設定された評価項目にて動作する暗号基盤アーキテクチャの提案を本稿の対象とする．

なお，本アーキテクチャでは，暗号エンジンと暗号評価標準記述の組を暗号パッケージとよぶ．

3.3.2　単体型と複合型

暗号評価標準記述には，単体型と複合型の 2 種類がある．ある 1 つの暗号エンジンに対する暗号評価標準記述を単体型の暗号評価標準記述という．一方，それぞれ単体型の暗号評価標準記述を有する複数の暗号エンジンが複合的に機能して，ある 1 つの暗号アルゴリズムを形成する場合がある．このとき，複数の単体型暗号評価標準記述を束ねる，当該暗号エンジン群に対する暗号評価標準記述を複合型の暗号評価標準記述という．

暗号評価標準記述に関する単体型と複合型の違いを，RSASSA-PSS-SHA256 の一実装を

⋆1 National Institute of Standards and Technology
http://www.nist.gov/
⋆2 Cryptography Research and Evaluation Committees
http://www.cryptrec.jp/index.html

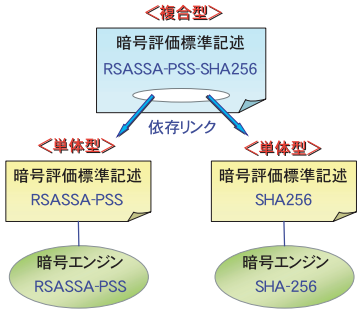

図 3　暗号評価標準記述例（RSASSA-PSS-SHA256）
Fig. 3　Sample of a cryptographic evaluation profile (RSASSA-PSS-SHA256).

例として**図 3** に示す．RSASSA-PSS-SHA256 は，それぞれ単体型の暗号評価標準記述をもつ RSASSA-PSS と SHA256 の暗号エンジンからなっている．暗号エンジンの集合である RSASSA-PSS-SHA256 全体に対する暗号評価標準記述が複合型である．

特に，公開鍵暗号処理は，こうした複合的な暗号エンジンの組合せで形成される．複合型の暗号評価標準記述は，こうした暗号エンジンの組合せからなる暗号機能に対して，最適な暗号エンジンの選定や管理を容易にするためのものである．複合型の暗号評価標準記述は，ある暗号エンジン単体への直接関係を持たないことから，単体型と違って暗号エンジン ID がないといった記載項目の違いがある．

3.4　最適暗号の自動選定機構

3.4.1　暗号選定機構の概要

本アーキテクチャは，状況，環境，上位システムの要求といったさまざまな要件に対して最適な暗号エンジンを自動選定する機構を有する．ここで，上位システムとは，暗号処理や暗号資産の一元管理を本機構へ依頼する，各種のミドルウェアやアプリケーションソフトウェアを含んでいる．暗号処理を実施する機器と暗号を管理するサーバとが必要に応じて連携し，暗号評価標準記述の内容を基に最適な暗号選定を実行する．

暗号エンジンの選定に必要な入力や，その出力はおおむね以下に示す情報である．ただし，初期登録時，更新時，暗号処理要求時といった暗号の選定を必要とする状況の違いにより，詳細は異なる．

入力：

- **暗号指定情報**
  暗号処理や暗号エンジンの指定情報．
- **暗号評価標準記述**
- **暗号エンジン選定ポリシ**
  複数の選定候補が存在した場合に，その中から最適なものを一意に絞る最終選定条件を指定する規約（各端末ごとに設定可能）．
- **搭載暗号エンジンリスト**
  当該機器や相手機器が保有する暗号エンジンのリスト．
- **ハードウェアプロファイル**
  CPU，利用可能な RAM サイズ，暗号コプロセッサの搭載状況等，動作機器に関する情報．
- **暗号処理オプション**
  安全性優先，速度優先，リソース使用度優先といった暗号処理に対する上位システムの要求．
- **パラメータ指定**
  暗号鍵の長さ等を指定．

出力：（主にいずれかを出力する）

- 選定された暗号エンジン群のリスト．
- 上位システムの要求を満たすエンジンが存在しない場合，不足暗号エンジン群のリスト．
- 暗号処理オプションを満たすものは存在しない．

ここで，暗号指定情報は，暗号処理や暗号エンジンの指定情報である．その指定の抽象度合によって，大きく 3 種類に分類される．抽象度の高い順にカテゴリ指定，暗号アルゴリズム指定，暗号エンジン指定である．カテゴリ指定は，上位システムが，公開鍵暗号による署名生成や乱数生成といった暗号処理目的で分類されたカテゴリのみを指定し，暗号処理を要求する抽象度の高い指定形式である．この暗号指定に関する抽象度が高いほど，暗号の切替えにともなう改変といった上位システム側への影響は少ない．たとえば，カテゴリ指定であれば，上位システムは同一暗号カテゴリに属す暗号エンジンを同じインタフェースで利用できるからである．

3.4.2　選定手順と実装形態

選定手順の主な流れは，以下のとおりである．

Step.1：暗号エンジン群から上位システムからの要求条件を満たす暗号エンジンを検索し，

候補となる暗号エンジンリストおよびその組合せを抽出する．その処理は，入力にある暗号指定情報により詳細手順は異なるが，おおむね以下のとおりである．

(1a)　暗号指定情報（カテゴリ，暗号アルゴリズム ID，暗号エンジン ID）をキーとして暗号エンジンを検索する．候補となる暗号エンジンが存在する場合は，(1b) へ．存在しない場合は，不足暗号エンジン群をリスト出力して終了する．

(1b)　処理 (1a) で抽出した各暗号エンジンに対して，対応する単体型の暗号評価標準記述を参照し，機能依存するカテゴリの有無を調査．機能依存するカテゴリの数だけ，Step.1 のカテゴリ指定を再帰的に呼び出す．この結果，リストに存在する暗号エンジンの組合せが 1 つでもあれば，これを存在する暗号エンジンリストとして出力．1 つも存在しなければ，不足カテゴリリストを出力．

Step.2：暗号評価標準記述，ハードウェアプロファイル，暗号処理オプション，パラメータ指定を用いて，Step.1 の出力リストから条件を満たす暗号エンジン群とその組合せを選定する．具体的には，安全性や速度，リソース使用度の点数評価，ベンダや作成日付等の条件一致検証を実施する．さらに，その結果を暗号エンジン選定ポリシにより，最適な暗号エンジン群とその組合せ方を 1 件に絞りこむ．暗号エンジン選定ポリシには，安全性，速度，リソースの優先順位を指定可能で，たとえば選定の優先順位がこの順番である場合，暗号評価標準記述の各定量値（図 2 参照）に基づき，次のような絞り込みを実施する．

(2a)　安全性の定量値が最も高い組合せのエンジン ID 群を出力．結果が複数存在する場合は，処理 (2b) へ．

(2b)　処理 (2a) 結果の中で，処理速度が定量値で最も高速なものを選定．結果が複数存在する場合は，処理 (2c) へ．

(2c)　処理 (2b) 結果の中で，リソース使用度の定量値が最も低いものを選定，出力．この際，ベンダ指定や暗号アルゴリズムの作成日付（最も古いものを選定）も指定可能である．

暗号プリミティブでも，ブロック暗号でハッシュ関数を形成，またはハッシュ関数で擬似乱数生成を形成する場合がある．この場合，処理要素として呼び出す暗号カテゴリの暗号エンジンを一意に選定するために，処理 (1b) で機能依存するカテゴリ有として，再帰的に選定処理を実行する．

上記の暗号選定手順を上位システムの要求のたびに毎回実施するのではなく，その要求と対応する最適な暗号エンジン群とを紐付けてキャッシュ化し，その後の選定処理を高速化するという実装が考えられる．また資源制約の厳しい組み込み機器では，選定処理をサーバ側へ委ね，その要求とエンジンとの関係性のみを管理するという実装も可能である．これは，上位システムへの応答処理性能の高速化に加え，選定に必要不可欠な暗号評価標準記述の管理をサーバへ委ね，機器の資源を節約するという効果がある．機器やネットワーク環境といったさまざま利用形態に応じて，最適な実装形態が存在する．

3.5　暗号切替えと暗号鍵管理

異なる暗号エンジンに切り替えた際，暗号処理に必要な暗号鍵をどう対応させて管理するかが重要である．

バグフィックスや最適化版，異なる暗号ベンダ実装といった暗号アルゴリズム自体が変わらない場合には，切り替えた暗号エンジンでも同じ暗号鍵を継続して利用可能である．異なる暗号アルゴリズムへ切り替えた場合には，新たな暗号鍵を生成する．端末内の暗号鍵 DB 内に過去同一の暗号アルゴリズムに対する暗号鍵を保有している場合には，この暗号鍵を指定して暗号処理を実行することもできる．

暗号化も復号も同じ端末で実行する場合には，暗号処理時に暗号鍵を生成して当該端末内の鍵管理 DB に登録する．他端末と暗号処理を実行する場合，必要に応じて鍵共有や公開鍵の登録，公開鍵証明書の取得を行う．鍵共有の手段として，相手機器との鍵共有を実行して接合機器に共有鍵を出力する，eTRON セキュリティデバイス[7] を利用する．

暗号管理ドメインの移動にともない，規制暗号エンジンを削除する場合も，必ずしも鍵削除を実行する必要はない．再度，利用可能なドメインに移動した際に，削除した暗号エンジンに切り替え，暗号鍵 DB 内に保存していた暗号鍵を指定する．こうした暗号鍵の利用や削除にともなう管理方式の選択は，暗号管理ドメインやアプリケーションのポリシに従う．

4.　切替自在な暗号システム

4.1　システム構成

本提案アーキテクチャのシステム構成図を**図 4** に示す．大きく 3 種類で構成され，暗号管理サーバ，ノード端末機器，当該機器に接合する耐タンパ性セキュリティデバイスである．本システムにおける各構成機器の役割や機能概要は以下のとおりである．

(1) 暗号管理サーバ　膨大な暗号パッケージを管理し，管理ドメイン下にあるノード端末機器に対して，当該ドメインのセキュリティポリシに従い，最適な暗号切替えにともなう多様な暗号保守を行う．

(2) ノード端末機器　PC や組み込み機器等，多種多様な形態で複数存在する．本システム

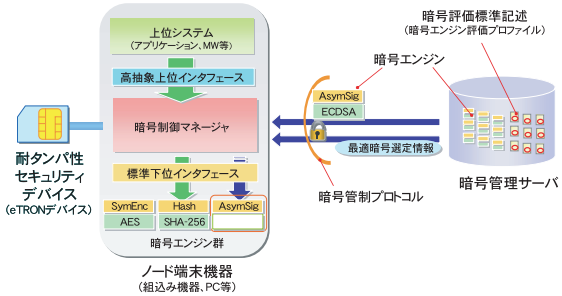

図 4　切替自在な暗号システム実装例
Fig. 4　Implementation of the flexibly switchable cryptographic system.

で最適保守された暗号資産を駆使し，規定水準の情報セキュリティを必要とする多様サービスを実行する．

(3) 耐タンパ性セキュリティデバイス　耐タンパ性を有したセキュリティチップを搭載し，提供する物理インタフェースでノード端末機器に接合する．暗号管理サーバの暗号保守や当該機器の暗号資産管理をより堅牢化する．

ノード端末機器として，PC よりも資源制約の厳しいユビキタス・コミュニケータや T-Engine[2] といった組み込み機器のスペックや実処理検証をふまえ，本アーキテクチャの各仕様を定めている．ユビキタスコンピューティング環境で十分に適用できる機構を提供するため，当該環境を形成する多種多様な機器で動作する必要があるからである．

4.2　ノード端末機器

ノード端末機器の構成は 3 層構造で，暗号を利用する上位システム（アプリケーションやミドルウェア），暗号処理や資産を一括管理する暗号制御マネージャ，暗号処理実体である暗号エンジンからなる．

このうち，暗号制御マネージャは，上位システムからの要求に対して，暗号管理サーバと適宜連携して，最適な暗号エンジンの自動選定を行い，暗号処理を実行する．実行した暗号処理の実行条件（暗号エンジン，暗号鍵等）を DB 記録し，暗号処理 ID を発行して情報参照や同一条件での暗号処理を提供する．また，新規登録，配信や更新，削除といった暗号パッケージの基本管理に加え，ドメイン移動にともなう，移動元・先の暗号管理ポリシに合わせた機器内の暗号資産管理も行う．暗号資産には，暗号パッケージのほか，暗号鍵も含まれる．

4.3　耐タンパ性セキュリティデバイス

ユビキタスコンピューティング環境を想定した耐タンパ性セキュリティデバイスとして，UIM 型の eTRON デバイスを評価プラットフォームとした．eTRON[3),6)] は，ユビキタスコンピューティング環境を想定した情報セキュリティアーキテクチャである．その構成要素である eTRON デバイスは，ユビキタスコンピューティング環境で通信ノードと peer-to-peer の暗号通信路を確立し，情報を安全に授受する機能を所持する．この機能を駆使して，暗号管理サーバからノード端末機器への暗号資産の配信をより堅牢化する．

4.4　暗号管制プロトコル

4.4.1　プロトコルスイート

暗号制御マネージャと暗号管理サーバは，暗号管制プロトコルという，本システム独自の通信プロトコルを介して，暗号パッケージの選定や保守を行う．暗号管制プロトコルスイートは，以下のようなプロトコルで構成される．

(1) 暗号パッケージ初期登録　ノード端末機器が現在所属している暗号管理ドメインで必須となる暗号パッケージが存在しない場合に当該パッケージを取得するプロトコル．輸出規制のため，暗号を現地調達する場合等，機器内に暗号パッケージがまったく存在しない場合にも実行する．

(2) 暗号パッケージ配信　初期登録後，暗号制御マネージャの要求条件を満たす暗号パッケージを配布するプロトコル．

(3) 暗号パッケージ更新　暗号危殆化にともなう，暗号管理サーバからの更新要求通知やノード端末機器からの定期的な更新確認で，更新情報を取得し，更新が必要である暗号パッケージを取得するプロトコル．

(4) 暗号パッケージ更新通知　バージョンアップ等，暗号管理サーバ上で管理された暗号パッケージが更新された際に，更新暗号リストを暗号制御マネージャに送信するプロトコル．

(5) 暗号パッケージ選択　暗号制御マネージャで暗号エンジンを選択できない場合や暗号管理サーバにあるすべての暗号パッケージを対象として最適な暗号を選択したい場合に，暗号制御マネージャの要求条件を満たす暗号パッケージを暗号管理サーバで選定するプロトコル．

(6) 暗号管理ドメイン移動　現在所属している暗号管理ドメインから別の暗号管理ドメインへ移動する場合に，持ち出しを禁止されている暗号パッケージの削除等を実行するプロトコル．

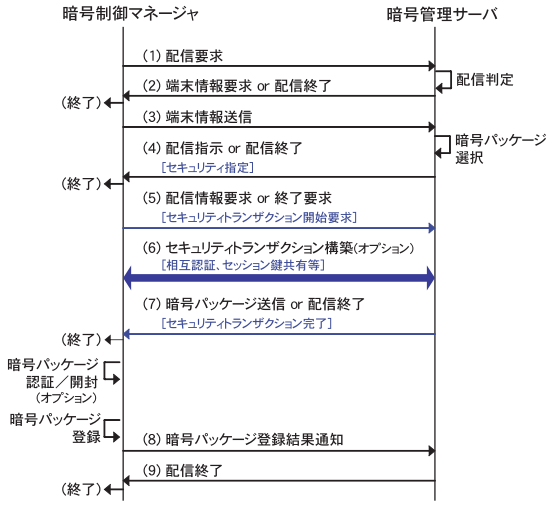

図 5　暗号パッケージ配信プロトコル
Fig. 5　Cryptographic package distribution protocol.

(7) 暗号パッケージリスト取得　暗号マネージャまたは暗号管理サーバに存在する暗号パッケージのリストを相手機器へ配布するプロトコル．

暗号管制プロトコルでは，基本的に暗号制御マネージャから暗号管理サーバに対する要求を行い，処理を実行する．ただし，暗号パッケージ更新通知等，一部暗号管理サーバから起動するプロトコルも存在する．プロトコルスイートの例として，暗号パッケージ配信プロトコルのシーケンスフローを**図 5** に示す．

4.4.2　セキュリティトランザクション

本システムでは，各暗号管制プロトコルの主要部で，暗号管理ドメインで定めるセキュリティクラスに応じたセキュリティ通信路を構築する．具体的には，適宜セキュリティハードウェアを利用して，暗号管理サーバの認証やセッション鍵共有を実行し，通信の機密化やデータ認証を行う．本システムでは，これら一連のセキュリティ通信をセキュリティトランザクションとよぶ．各セキュリティクラスの規定概要を**表 2** に示す．

表 2　セキュリティトランザクションのセキュリティクラス
Table 2　Security class of security transactions.

クラス	相手認証	データ認証		暗号化
	処理主体	鍵種別	処理主体	処理主体
0	—	—	—	—
1	—	固定鍵	CCM	—
2	CCM	セッション鍵	CCM	—
3	CCM	セッション鍵	CCM	CCM
4	SD	セッション鍵	CCM	—
5	SD	セッション鍵	CCM	CCM
6	SD	セッション鍵	SD	SD

CCM：暗号制御マネージャ，SD：セキュリティデバイス

5.　各利用シーンでのシステム動作

5.1　システム稼動開始時

ある暗号管理ドメインで切替自在な暗号システムサービスを始動する際，次のような準備を行う．利用可能な暗号やその鍵長，暗号配信方法といった本ドメインの暗号管理ポリシを規定する．本サーバ自身の暗号鍵生成や公開鍵証明書の取得も行う．その後，暗号評価機関の承認を得た，暗号エンジンと暗号評価標準記述を暗号管理サーバ DB に登録する．暗号エンジンを評価し，暗号評価標準記述への認定署名を行う機構や，異なる暗号管理ドメインを運用管理する機構，公開鍵証明書発行の機構が実運用では重要である．運用モデルは，多様な形態が想定され，本稿では記載を割愛する．

5.2　ノード端末機器のサービス利用開始時

ノード端末機器で本サービスを利用開始する際，次のような処理を行う．暗号制御マネージャといった本システム利用に必要なクライアントシステムをノード端末機器に搭載した後，識別情報となる端末 ID を含む必要情報を暗号管理サーバへ送出して利用登録を行う．その後，暗号パッケージ初期登録プロトコルを実行し，本暗号管理ドメインに必須となる暗号パッケージを取得する．暗号管理サーバは，各ノード端末ごとに暗号パッケージの配信状況を管理する．必須暗号パッケージ配信時のセキュリティは，各暗号管理ドメインで規定するポリシによる．eTRON デバイスには，入力された暗号化データをデバイス内で復号し，接合した機器に提供する機能を有したデバイスが存在する．本デバイスを利用して，暗号パッケージをまったく保有しないノード端末機器に，安全に必須暗号パッケージを配信することもできる．また，必要に応じて鍵生成の暗号パッケージを取得し，暗号鍵の生成や鍵証

明書の取得を実行する．

5.3　暗号処理実行時

上位アプリケーションから上位インタフェースを介して暗号処理を要求された場合，暗号制御マネージャは次のような処理を行う．要求内容に応じて，詳細手順は異なるが，おおむね以下のとおりである．

Step.1：（暗号エンジン選定） 過去の暗号処理履歴を適宜参照し，入力された処理要求条件に見合う暗号エンジンを，ノード端末機器内に保有した暗号エンジンの中から選定する．条件に見合う暗号エンジンが存在しない場合は，暗号パッケージ選択プロトコルや暗号パッケージ配信プロトコルを実行し，必要な暗号パッケージを取得する．

Step.2：（鍵取得） 暗号処理に必要な暗号鍵データは，暗号制御マネージャの鍵 DB に既登録データ，または鍵生成エンジンを実行して新規に作成した暗号鍵を指定する．

Step.3：（暗号処理実行） 選定した暗号エンジンと取得した暗号鍵データで暗号処理を実行し，結果データをアプリケーションへ出力する．

Step.4：（暗号処理 ID 発行） 本暗号処理実行の暗号処理 ID を発行し，アプリケーションへ出力する．暗号処理 ID には，利用した暗号エンジンや鍵情報といった本処理に関する情報を紐づけて暗号処理 DB で管理する．以降，本暗号処理 ID を指定して，処理情報を参照したり，同一の条件で異なるデータへ暗号処理を実行したりすることもできる．

5.4　暗号危殆化時や暗号評価情報の変更時

暗号危殆化や暗号評価情報の変更が発生した際，次のような処理を行う．更新の動機には，2 種類ある．1 つは，暗号管理サーバが暗号パッケージ更新通知プロトコルを実行して，関係するノード端末機器へいっせいに更新情報を通知する場合である．通知情報には，必須・推奨・任意といった更新の重要度を示す更新ランクが含まれる．「必須」は暗号危殆化による緊急かつ最重要の更新，「任意」は暗号評価情報の若干修正といったシステムに大きな支障を来さない更新である．ノード端末機器は更新情報を受信後，必要に応じて暗号パッケージ配信プロトコルや暗号パッケージ更新プロトコルを実行し，暗号パッケージを取得する．もう 1 つは，ノード端末機器の暗号制御マネージャが暗号パッケージ更新プロトコルを実行して，定期的に更新情報を暗号管理サーバへ問い合わせる場合である．必要に応じて新規暗号パッケージを取得する．

5.5　異なる暗号管理ドメインへの移動時

ノード端末機器が暗号管理ドメイン A から異なる暗号管理ドメイン B へ移動する際，次のような処理を行う．移動前，ノード端末機器の暗号制御マネージャは暗号管理ドメイン A の暗号管理サーバと暗号管理ドメイン移動プロトコルを実行し，暗号管理ドメイン B へ持ち出しが禁止されている暗号パッケージを暗号管理ドメイン A 内ですべて削除する．移動後，暗号制御マネージャは暗号管理ドメイン B の暗号管理サーバと暗号パッケージ初期登録プロトコルを実行し，本ドメインで必須となる暗号パッケージを取得する．

表 3　暗号管理サーバの仕様
Table 3　Specification of the cryptographic management server.

暗号管理サーバ（DELL Dimension 8250）
CPU：Pentium4 2.53 GHz
Memory：1 GB
HDD：120 GB
OS：Linux Fedora Core 4 kernel-2.6
DB：PostgreSQL 8.0.3-1

表 4　静的ノード端末の仕様
Table 4　Specification of the static terminal.

静的ノード端末（T-Engine/SH7727）
CPU：内部クロック 96 MHz（SH3-DSP）
フラッシュメモリ：8 MB
SDRAM：64 MB
OS：PMC T-Kernel (R1.14)

表 5　動的ノード端末の仕様
Table 5　Specification of the dynamic terminal.

動的ノード端末（ユビキタス・コミュニケータ：UC）
CPU：32 bit RISC チップ
ディスプレイ：VGA（480 × 640 ピクセル）
通信機能：IEEE802.11b
外形寸法：144 × 76 × 15 mm
重量：196 g

6. 試作システム

切替え自在な暗号システムの試作を以下のとおり実施した．暗号管理サーバ，静的ノード端末，動的ノード端末，セキュリティデバイスを，それぞれ PC-Linux（**表 3**），T-Engine/SH7727（**表 4**），ユビキタス・コミュニケータ（**表 5**），eTRON/16-AE45X（**表 6**）で試作した．なお，暗号管理サーバと T-Engine/SH7727 との通信は，10BASE（HUB：

表 6　セキュリティデバイスの仕様
Table 6　Specification of the security device.

セキュリティデバイス (eTRON/16-AE45X)
CPU：16 bit (AE45X)
EEPROM：32 KB + 4 KB
ROM：128 KB
RAM：4 KB
インタフェース：ISO/IEC 7816, ISO/IEC 18092

表 7　カテゴリと暗号エンジン実装（例）
Table 7　Category and implementation of cryptographic engines (example).

カテゴリ	サブカテゴリ		アルゴリズム名	カテゴリ	サブカテゴリ	アルゴリズム名
共通鍵暗号 暗号化 復号 MAC 生成 MAC 検証	ブロック暗号	暗号コア	DES DES-EDE-2key DES-EDE-3key Hierocrypt-L1 Hierocrypt-3 AES	ハッシュ関数		MD5 SHA-1 SHA-224 SHA-256 SHA-384 SHA-512 RIPEMD-128 RIPEMD-160
		暗号化モード	ECB CBC CFB OFB CTR	HMAC		RFC2104
		MAC モード	ISO9797-ALG1 ISO9797-ALG3 OMAC XCBC-MAC	公開鍵暗号	暗号化/復号	RSAES-PKCS#1v1.5 RSAES-OAEP ElGamal
		パディング	no-padding ISO9797-M1 ISO9797-M2 RFC2409 PKCS#5		署名生成 署名検証	RSASSA-PKCS#1v1.5 RSASSA-PSS DSA ECDSA(Fp)
	ストリーム暗号		RC4		鍵生成	RSA ElGamal DSA ECDSA(Fp)
マスク生成関数	ハッシュ関数		エンジン使用 エンジン内蔵	乱数生成		ANSI X9.42 FIPS180-2 A3.2 FIPS180-2 A3.2 Rev.

Allied Telesis MR820TR 経由）である．

また暗号エンジンは，PC-Linux，T-Engine/SH7727，ユビキタス・コミュニケータの各プラットフォームに対して，**表 7** に示す全暗号アルゴリズムを実装した．

表 8 に性能測定値を示す．具体アルゴリズム名は無指定で暗号カテゴリのみを入力し，最適な暗号エンジンを自動選定して，鍵生成を実行，その出力暗号鍵で暗号処理を実行した結果[⋆1]である．

7. おわりに

本稿では，多種多様な暗号から，時事・状況・環境やアプリケーション要求に最適な暗号を選定し，自在に切り替えて利活用する切替え自在な暗号アーキテクチャを提案した．本提案は，情報システムから危殆化した暗号を事後的に新暗号へ移行する方法ではなく，異なる暗号へ柔軟に切り替える機構を最初から組み込んでおく方法である．NIST では，次世代ハッシュ関数である SHA-3 を決めるコンテスト[⋆2]が開催されている．より高度化する情報社会に対して，情報システムと暗号技術との関係性がどうあるべきか，最良の策を業界全体で考案・評価する必要がある．

表 8　鍵生成＋暗号処理の性能
Table 8　Performance of key generation & cryptograhic process.

暗号指定	選定結果	T-Engine	UC
公開鍵暗号署名	ECDSA-SHA-1	10,431	8,142
HMAC 生成	RFC2104-SHA-1	4,102	3,421
共通鍵暗号化	AES-CBC-PKCS	2,087	1,346

（単位：msec）

本提案アプローチの実運用化にともなう大きな課題は，単体型または複合型の暗号エンジンに対する適切な安全性の定量化である．今回，試作システムの暗号評価標準記述には暫定値を入力した．Lenstra-Verheul の安全性指標[5] といったさまざまな見解を基に，業界全体で規定できることが望ましい．また，どんな環境でも復号できるように，利用が想定される複数の暗号管理ドメインポリシ情報から共通に規制問題のない暗号化エンジンを選定する機構の検討も今後の課題である．

謝辞　本技術は，独立行政法人情報通信研究機構から YRP ユビキタス・ネットワーキング研究所が助成を受け，研究開発したものである．また本稿執筆にあたり東芝ソリューション株式会社 IT 技術研究所の研究員各位には，貴重な技術支援をいただいた．関係各位のご支援に深謝する．

参考文献

1) Gantz, J.F., Chute, C., Manfrediz, A., Minton, S., Reinsel, D., Schlichting, W. and Toncheva, A.: The Diverse and Exploding Digital Universe, White Paper (2008).
2) Koshizuka, N. and Sakamura, K.: T-Engine Project: The Open Platform Project for Ubiquitous Computing, *1st International Conference on Ubiquitous Computing* (*ICUC 2003*), IEEE, pp.185–190 (2003).

⋆1 入力データは 23 Bytes で，ECDSA は 192 bit セキュリティ．

⋆2 http://www.csrc.nist.gov/groups/ST/hash/sha-3/index.html

3) 越塚　登，坂村　健：eTRON：Entity and Economy TRON，第 19 回情報処理学会コンピュータセキュリティ研究会，pp.61–66 (2002).
4) 栃窪孝也，岡田光司，遠藤直樹，岡本栄司：リニューアル可能な暗号認証システムの検討，情報処理学会論文誌，Vol.41, No.8, pp.2121–2128 (2000).
5) Lenstra, A.K. and Verheul, E.R.: Selecting Cryptographic Key Sizes, *Journal of Cryptology*, Vol.14, No.4, pp.255–293 (2001).
6) Sakamura, K. and Koshizuka, N.: The eTRON Wide-Area Distributed-System Architecture for E-Commerce, *IEEE MICRO*, Vol.21, No.6, pp.7–13 (2001).
7) 宮崎真悟，石川千秋，鵜坂智則，小俣三郎，越塚　登，坂村　健：組込み機器に秘密共有機能を提供する SIM カード型セキュアチップの開発，第 66 回全国大会講演論文集（1），pp.27–28 (2004).
8) 宇根正志，神田雅透：暗号アルゴリズムにおける 2010 年問題について，Discussion Paper, No.2005-J-22 (2005).
9) 総務省地域力創造グループ地域情報政策室：暗号アルゴリズムの移行スケジュール案，公的個人認証サービスにおける暗号方式等の移行に関する検討会 (2008).
10) 日本データ通信協会：タイムビジネス認定基準：時刻認証業務（デジタル署名を使用する方式），タイムビジネス信頼・安心認定制度 (2006).

(平成 20 年 12 月 1 日受付)
(平成 21 年　6　月 4 日採録)

宮崎　真悟（正会員）

1974 年生．1999 年九州大学大学院システム情報科学研究科情報工学専攻修士課程修了．同年（株）東芝入社．2002 年から 2007 年にかけて，YRP ユビキタス・ネットワーキング研究所研究員．現在，東京大学大学院学際情報学府総合分析学コース博士課程．ユビキタスコンピューティング，情報セキュリティ技術に関する研究開発に従事．平成 11 年度論文賞，第 62 回全国大会大会優秀賞受賞．

石川　千秋

1955 年生．1980 年東京大学大学院理学系研究科修士課程修了．1983 年よりパーソナルコンピュータ，ワークステーション，組み込み機器のソフトウェア開発，海外のソフトウェアのローカライゼーションに従事．2002 年より YRP ユビキタス・ネットワーキング研究所研究員．ユビキタスコンピューティング，セキュリティ，コンピュータ言語，ソフトウェア開発環境，オペレーティングシステム，フリーソフトウェアの研究に従事．

越塚　　登（正会員）

1966 年生．1994 年東京大学大学院理学系研究科博士課程修了．博士（理学）．1994 年東京工業大学大学院情報理工学研究科助手，1996 年東京大学大学院人文社会系研究科助教授，1999 年同大学情報基盤センター助教授，2006 年同大学大学院情報学環助教授，現在，同准教授．2002 年より YRP ユビキタス・ネットワーキング研究所副所長を兼務．専門は情報科学．トロンプロジェクトに参加し，以後ヒューマンインタフェースやユビキタスコンピューティング，セキュリティ，組み込みシステムの研究に取り組んできた．IEEE-CS，ACM 各会員．

坂村　　健（正会員）

1951 年生．東京大学大学院情報学環教授．工学博士．1984 年からオープンなコンピュータアーキテクチャTRON を構築．現在，TRON はユビキタス環境を実現する重要な組み込み OS として世界で多数使われている．さらに，コンピュータを使った電気製品，家具，住宅，ビル，都市，ミュージアム等広範なデザイン展開を行っている．2002 年 1 月より YRP ユビキタス・ネットワーキング研究所長を兼任．『ユビキタスとは何か』，『変われる国，日本へ』等著書多数．IEEE フェロー，ゴールデンコアメンバー．第 33 回市村学術賞特別賞，2001 年武田賞，2003 年紫綬褒章，2006 年日本学士院賞受賞．