環境に応じた暗号プロトコルの動的変更方式の実装・評価

太田　陽基[†]　　清本　晋作[†]　　田中　俊昭[†]

†（株）KDDI 研究所　〒 356–8502 埼玉県ふじみ野市大原 2–1–15

あらまし　多種多様な端末・ネットワークが利用されるユビキタス環境では，環境に応じて暗号プロトコルが動的に変更されることが望ましい．上記の課題を解決するためには，暗号プロトコルの自動生成技術，安全性の高速検証技術，動的実行技術が必要である．本稿では，これらの技術を有する著者らのツールにもとづき，環境に応じた暗号プロトコルの動的変更方式を提案する．また，環境が頻繁に変化する状況として，2 種類の暗号プロトコルの動的変更例を想定する．上記の例に使用されるプロトコルを提案方式に適用することにより，提案方式の有効性を確認する．

Implementation and Evaluation of Dynamic Modification Mechanisms of Cryptographic Protocols Corresponding to Environments

Haruki Ota[†], Shinsaku Kiyomoto[†], and Toshiaki Tanaka[†]

† KDDI R&D Laboratories, Inc.　2–1–15 Ohara, Fujimino-shi, Saitama, 356–8502 Japan

Abstract　In a ubiquitous environment, it is preferable for cryptographic protocols to be modified dynamically in accordance with the environments. In this paper, we propose a dynamic modification mechanism of cryptographic protocols corresponding to the environments, based on the authors' tools to resolve the above problem. Also, we suppose two types of dynamic modification examples to modify the cryptographic protocols dynamically as the situations that change frequently. We showed that this mechanism is valid by applying the protocols used for the above examples to the proposed mechanism.

1　ま え が き

多種多様な端末・ネットワークが利用されるユビキタス社会では，環境に応じて提供される認証プロトコルや鍵交換プロトコルなどの暗号プロトコルが必要不可欠である．しかしながら，ユビキタス環境では，今後さらにサービスの多様化が想定されるため，電子決済などの高度な安全性から公共施設の利用などの属性確認のみの安全性まで，提供サービスによるさまざまなレベルの安全性への対応が求められる．現状端末やサービスごとに個別な技術が用いられており，このような多種多様なサービスへの柔軟な対応はできていなかった．

そこで，上記の課題を解決するためには，以下の 3 つの技術が必要であると考えられる．

(1) 環境に応じて暗号プロトコルを自動的に生成する技術
(2) 安全性要件に従って暗号プロトコルの安全性を高速に検証する技術
(3) 上記の自動生成・高速検証された暗号プロトコルを動的に実行する技術

まず，(1) に関連する技術として，Perrig らはプロトコル設計者が入力した安全性要件からプロトコル定義ファイルを自動的に生成するツールを最初に提案した [1]．以降，いくつかの自動生成ツールが提案されたが，生成の効率性に問題があった．著者らは，性能や安全性要件に従って選択された機能ブロックを用いて，プロトコル仕様を記述した定義ファイルを生成するプロトコル自動生成手法を提案した [2]．著者らのツールにより，Pentium 4 2.6[GHz] プロセッサ及び 2.0[GByte] RAM の PC を用いて，認証・鍵交換プロトコルをすべて 1[s] 以内に自動生成することができた．

次に，(2) に関連する技術として，計算量理論にもとづいて安全性を証明する手法とフォーマル検証にもとづく手法が提案されている．前者の手法として，Bellare らは認証及び鍵交換プロトコルに対して，識別不可能性にもとづく安全性のフォーマルなモデルを最初に導入し [3]，本分野における後に続く多くの研究の基盤となった．後者の手法として，Dolev-Yao モデル [4] などの特化型状態探索にもとづく手法を始めとする多くの手法が提案されてきた．しかしながら，これらの手

法は，プロトコルごとの安全性証明を要する，安全性検証に多くの時間を要する，必ずしも自動化されていない，という問題があった．著者らは，Bellare らのモデルにもとづいて，認証・鍵交換プロトコルの安全性を自動的かつ高速に検証する手法を提案した [5]．著者らのツールにより，Intel Core 2 Duo 3.0[GHz] プロセッサ及び 2.0[GByte] RAM の PC を用いて，著者らの自動生成ツールにより生成された認証・鍵交換プロトコルの安全性をすべて 15[ms] 以内に検証することができた．

最後に，(3) に関連する技術として，暗号プロトコルの実行モジュールを生成するコンパイラが提案されている．Muller らは CAPSL（Common Authentication Protocol Specification Language）などから Java コードを自動的に生成する方式を提案した [6]．以降，いくつかの実行コード生成コンパイラが提案されたが，自動生成ツールにより記述されたプロトコル仕様に合うコンパイラは設計されていない．著者らは，著者らの自動生成ツールにより生成された定義ファイルから C 言語のソースを生成し，C 言語としてコンパイルすることにより実行モジュールを生成する暗号プロトコルコンパイラを開発した [2]．著者らのコンパイラにより，2.4[GHz] プロセッサ及び 1.0[GByte] RAM の PC を用いて，定義ファイルから C 言語のソースをすべて 30[ms] 以内に生成することができた．また，生成された実行モジュールサイズも 100[KByte] 程度と現実のサービスでも実行可能なサイズである．

本稿では，端末の処理性能や通信状況などの環境に応じて，暗号プロトコルを動的に変更する方式を提案する．上記のとおり，提案方式には暗号プロトコルの自動生成，安全性の高速検証及び動的実行の各技術が必要である．そこで，各技術に関して，著者らの自動生成ツール [2]，高速検証ツール [5] 及び暗号プロトコルコンパイラ [2] を動的に実行可能にした暗号プロトコル動的コンパイラ（以下,「動的コンパイラ」と呼ぶ）をそれぞれ使用する．また，環境が頻繁に変化する 2 種類の動的変更例を想定し，提案方式に適用する．

本稿は以下のように構成されている．2 節では，暗号プロトコルの動的変更方式を提案する．3 節では，提案方式に適用される動的変更例を示す．4 節では，3 節における動的変更例に対して，2 節における提案方式の評価結果を示す．5 節では，本稿のまとめを述べる．

2 暗号プロトコルの動的変更方式

2.1 安全性モデル

本小節では，暗号プロトコルの動的変更方式を提案する前に，Bellare らによる認証・鍵交換プロトコルにおける安全性モデルについて説明する．

Bellare らは現実の攻撃をモデル化することにより，計算量理論にもとづいて，認証・鍵交換プロトコルの評価手法を定式化し，いくつかの提案した認証・鍵交換プロトコルの安全性を証明した [3]．Bellare らは上記の証明を行う上で，認証プロトコルの “matching conversation” 及び鍵交換プロトコルの “semantic security” という新しい安全性の概念を導入した．Bellare らは，各概念に対し，認証・鍵交換プロトコルに要求される安全性要件に応じて，現実の攻撃モデルを想定し，次の安全性に関する定式化を行った．

- matching conversation（MC）
  認証プロトコルにおいて，攻撃者が一方のパーティから受信したメッセージをそのままもう一方のパーティへ送ること以外に何もできない．
  - (a) なりすまし攻撃安全（MC-SIA）
    攻撃者がパーティ間の通信を支配しても，認証プロトコルを破ることはできない．
- semantic security（SS）
  鍵交換プロトコルにおいて，攻撃者はセッション鍵に関するどんな情報も得ることができない．
  - (b) 受動的攻撃安全（SS-SPA）
    攻撃者がパーティ間の通信を盗聴しても，鍵交換プロトコルを破ることはできない．
  - (c) 能動的攻撃安全（SS-SAA）
    攻撃者がパーティ間の通信を支配しても，鍵交換プロトコルを破ることはできない．
  - (d) オフライン辞書攻撃安全（SS-RODA）
    攻撃者がパスワード推測用の辞書からオフラインで履歴に一致するパーティのパスワードを探索することはできない．
  - (e) 既知鍵攻撃安全（SS-KKS）
    攻撃者があるセッション鍵を得ることができたとしても，それとは別のセッション鍵を得ることはできない．
  - (f) weak forward secrecy（SS-WFS）
    攻撃者が共通鍵暗号の共通鍵やパスワード，公開鍵暗号の秘密鍵などのパーティ長期鍵を得ることができたとしても，それ以前に共有されたセッション鍵を得ることはできない．

2.2 提案方式の手順

本小節では，暗号プロトコルに対して提案する動的変更方式を示す．

上記の課題を解決するために，著者らの自動生成ツール [2]，高速検証ツール [5] 及び動的コンパイラの各技術を用いた暗号プロトコルの動的変更方式のシステム構成を図 1 に示す．各エンティティの役割は以下のとおりである．

- ユーザ端末
  サービスを受けるユーザの保有する端末であり，動的コンパイラが既に実装されている．
- サービス提供サーバ
  サービスを提供するサービス提供者の保有するサーバであり，動的コンパイラが既に実装

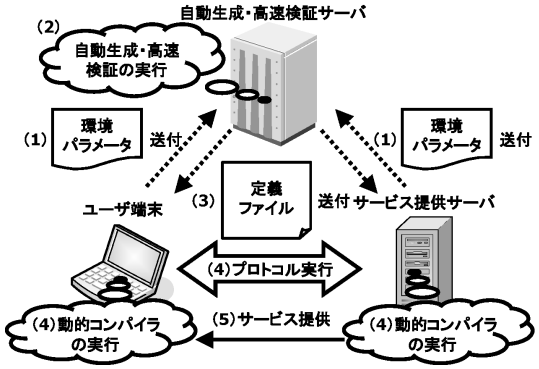

図 1　暗号プロトコルの動的変更方式のシステム構成

Fig. 1　System configuration of dynamic modification mechanisms of cryptographic protocols.

されている．

- 自動生成・高速検証サーバ
  上記のエンティティを支援する信頼できる第三者機関で，自動生成・高速検証の各ツールが実装されている．

また，ユーザ端末やサービス提供サーバに実装されている動的コンパイラのモジュール構成を図 2 に示す．各機能の役割は以下のとおりである．

- フロントエンド GUI アプリ
  フロントエンドの GUI により，相手と定義ファイルの送受信を実行する．
- 制御モジュール
  フロントエンド GUI アプリから受け取った定義ファイルを動的コンパイラモジュールに送る．また，動的コンパイラモジュールやサービスモジュールの制御を行う．
- 動的コンパイラモジュール
  制御モジュールから受け取った定義ファイルにより相手とプロトコルを実行する．

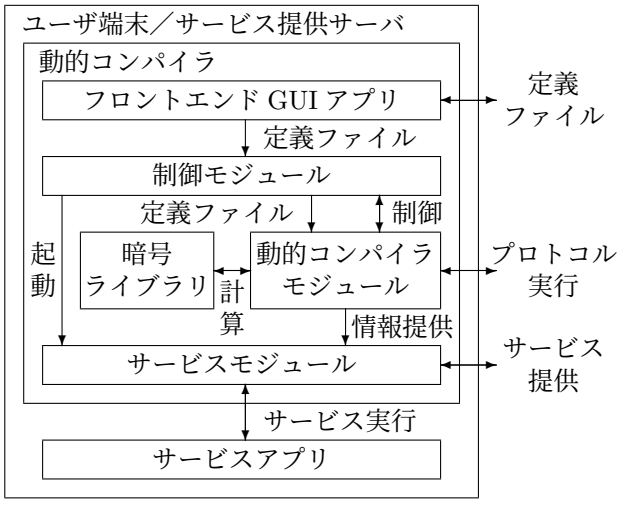

図 2　動的コンパイラのモジュール構成

Fig. 2　Module configuration of dynamic compiler.

- 暗号ライブラリ
  動的コンパイラモジュールにおいて必要な暗号アルゴリズムの計算を行う．
- サービスモジュール
  制御モジュールにより起動され，動的コンパイラモジュールから情報を提供されることにより，相手とのサービス提供を実行する．
- サービスアプリ
  ユーザ端末やサービス提供サーバ側にあり，サービスモジュールから提供されたサービスを実行する．

上記をもとに，提案する動的変更方式の手順を以下に示す．

(1) ユーザ端末とサービス提供サーバは自分の環境パラメータを自動生成・高速検証サーバに送る．ただし，環境パラメータに記載されている主な情報は後に示す．
(2) 自動生成・高速検証サーバは手順（1）において受け取った環境パラメータから両者に最適なプロトコルを定義ファイルとして生成する．定義ファイルにはプロトコル仕様が記載されている．自動生成・高速検証サーバは自動生成されたプロトコルの安全性を検証する．高速検証サーバからは，環境パラメータに記載されている安全性要件を満足しているか否かの検証結果が出力される．このとき，要求されている安全性を満足するまで，自動生成・高速検証サーバは自動生成及び安全性検証を繰り返す．
(3) 自動生成・高速検証サーバは手順（2）において安全性要件を満足した定義ファイルをユーザ端末とサービス提供サーバに送る．
(4) ユーザ端末とサービス提供サーバは自分の動的コンパイラを用いて手順（3）において受け取った定義ファイルを読み込み，プロトコルを動的に実行する．
(5) ユーザ端末とサービス提供サーバは手順（4）において実行されたプロトコルが成功すると，サービスを開始する．

ここで，環境パラメータに記載されている主な情報は以下のとおりである．

- プロトコル種別
  - 認証（AUTH）
  - 鍵交換（KE）
  - 認証付鍵交換（AKE）
- 各端末の処理性能
  - 高処理性能（H）
  - 中処理性能（M）
  - 低処理性能（L）
- 各端末において使用可能な暗号アルゴリズム
  - 共通鍵暗号方式（SKE）
  - 公開鍵暗号方式（PKE）
  - ディジタル署名スキーム（SIG）
  - Diffie-Hellman（DH）

- ハッシュ関数（HF）
- メッセージ認証コードスキーム（MAC）

○ 最大フロー数
- 1
- 2
- 3

○ 最大フローサイズ
- （数値の入力）

○ 安全性要件
- MC-SIA
- SS-SPA
- SS-SAA
- SS-RODA
- SS-KKS
- SS-WFS

注 1：提案方式について，今回は（1）の環境パラメータの送付手順と（3）の定義ファイルの送付手順（点線矢印）を実装していない．そのため，本稿では，環境パラメータ及び定義ファイルについては，ディジタル署名などを利用して安全に送付されることを仮定する．以降，上記の手順についてはオフラインで環境パラメータ及び定義ファイルを受け渡すことにより対応することとし，これらの安全性は議論の対象外とする．また，環境パラメータも定義ファイルも数〜数十 [KByte] 程度のサイズであり，現実的な値であることから，これらの通信時間も議論の対象外とする．

3 動的変更例

本節では，提案方式に適用する暗号プロトコルの 2 種類の動的変更例を示す．

3.1 通信状況等の変化による動的変更例

本小節では，通信状況等の変化による暗号プロトコルの動的変更例を示す．

多種多様な端末がさまざまなネットワークに接続される場合，通信状況や端末の処理負荷状況などが絶えず変化することが想定される．そこで，ユーザ端末がコンテンツ配信サーバからコンテンツ配信サービスを受ける場合において，良好な通信や端末処理が急に混雑して高速用プロトコルを使用できなくなったとき，低速用プロトコルに動的に変更される例及びその逆の例をそれぞれ考える．ここで，高速用及び低速用の各プロトコルの環境パラメータを表 1 に示す．ただし，各記号は前節の環境パラメータ情報の記号を表している．これらの環境パラメータを自動生成・高速検証ツールに入力することにより，高速用及び低速用の各プロトコルが得られる．

高速用プロトコルフローを以下に示す（図 3 参照）．

(1) ユーザ端末とコンテンツ配信サーバはあらかじめ安全な方法で MAC 鍵 K_1 を共有しておく．

表 1　高速用・低速用プロトコルの環境パラメータ
Table 1　Environment parameters of protocols for high speed and low speed.

種類	種別	処理性能	使用可能暗号アルゴリズム	フロー	
				数	サイズ
高速用	AKE	H と H	DH と MAC	3	4096
低速用	AKE	H と L	PKE と MAC	3	1024

種類	MC	SS				
	SIA	SPA	SAA	RODA	KKS	WFS
高速用	○	○	○	—	○	○
低速用	○	○	○	—	○	○

(2) ユーザ端末は乱数 R_1 と x を生成し，コンテンツ配信サーバに DH 公開鍵 g^x とその MAC 値と R_1 を送る．

(3) コンテンツ配信サーバは乱数 R_2 と y を生成し，ユーザ端末に DH 公開鍵 g^y と R_2 と自らの ID である I_2 及びその鍵と R_1 と I_2 の MAC 値を送る．

(4) ユーザ端末はコンテンツ配信サーバに自らの ID である I_1 及び R_2 と I_1 の MAC 値を送る．

(5) ユーザ端末とコンテンツ配信サーバはそれぞれ g^{xy} をセッション鍵 sk とする．

低速用プロトコルフローを以下に示す（図 4 参照）．

(1) ユーザ端末とコンテンツ配信サーバはあらかじめ安全な方法で MAC 鍵 K_1 を共有しておく．

(2) ユーザ端末は乱数 R_1 とテンポラリな公開鍵 Pt_1 を生成し，コンテンツ配信サーバに R_1 と Pt_1 とその MAC 値を送る．

(3) コンテンツ配信サーバは乱数 R_2 と R_3 を生成し，ユーザ端末に R_2 の楕円曲線暗号による暗号文と R_3 と自らの ID である I_2 及び R_1 とその暗号文と I_2 の MAC 値を送る．

(4) ユーザ端末はコンテンツ配信サーバに自らの ID である I_1 及び R_3 と I_1 の MAC 値を送る．

(5) ユーザ端末とコンテンツ配信サーバはそれぞれ R_2 をセッション鍵 sk とする．

3.2 安全性の変化による動的変更例

本小節では，安全性の変化による暗号プロトコルの動的変更例を示す．

多種多様な端末がさまざまなサービスを利用しようとする場合，サービスの種類に応じて安全性レベルが絶えず変化することが想定される．そこで，同じくコンテンツ配信サービスを受ける場合において，コンテンツを試聴する際に使用した低安全プロトコルをコンテンツ購入の際に使用できなくなったとき，高安全プロトコルに動的に変更される例及びその逆の例をそれぞれを考える．こ

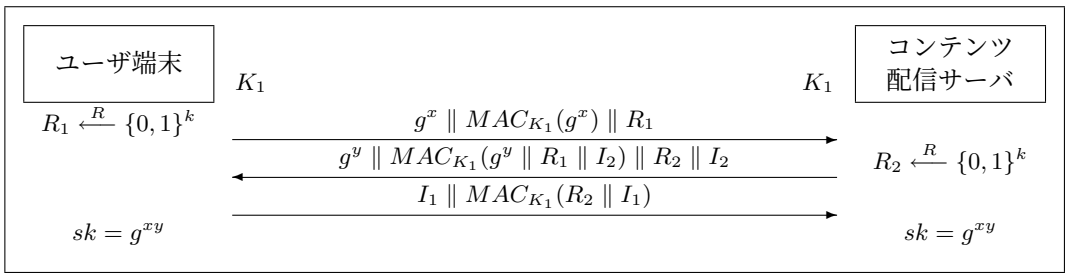

図 3　高速用（高安全）プロトコルフロー

Fig. 3　Protocol flow for high speed (high-level security protocol flow).

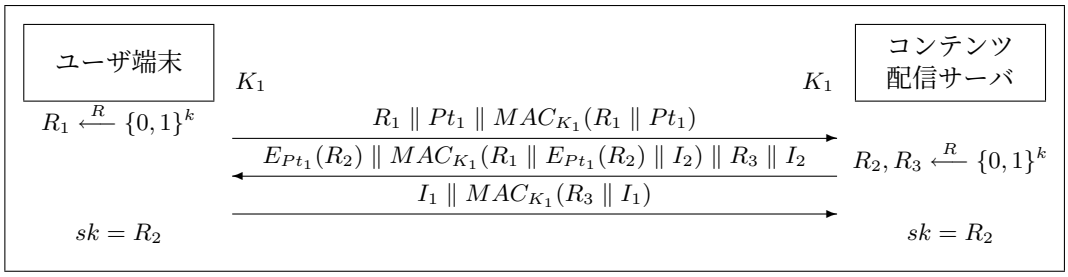

図 4　低速用プロトコルフロー

Fig. 4　Protocol flow for low speed.

こで，低安全及び高安全の各プロトコルの環境パラメータを表 2 に示す．これらの環境パラメータを自動生成・高速検証ツールに入力することにより，低安全及び高安全の各プロトコルが得られる．

低安全プロトコルフローを以下に示す（図 5 参照）．なお，高安全プロトコルフローは高速用プロトコルフローと同じである（図 3 参照）．

(1) ユーザ端末とコンテンツ配信サーバはあらかじめ安全な方法で MAC 鍵 K_1 を共有しておく．
(2) ユーザ端末は乱数 R_1 を生成し，コンテンツ配信サーバに R_1 を送る．
(3) コンテンツ配信サーバは乱数 R_2 と R_3 を生成し，ユーザ端末に R_2 と R_3 と自らの ID である I_2 及び R_1 と R_2 と I_2 の MAC 値を送る．
(4) ユーザ端末はコンテンツ配信サーバに自らの ID である I_1 及び R_3 と I_1 の MAC 値を送る．
(5) ユーザ端末とコンテンツ配信サーバはそれぞれ R_1 と R_2 の MAC 値をセッション鍵 sk とする．

表 2　低安全・高安全プロトコルの環境パラメータ

Table 2　Environment parameters of low-level and high-level security protocols.

種類	種別	処理性能	使用可能暗号アルゴリズム	フロー	
				数	サイズ
低安全	AKE	H と H	DH と MAC	3	4096
高安全	AKE	H と H	DH と MAC	3	4096

種類	MC	SS				
	SIA	SPA	SAA	RODA	KKS	WFS
低安全	○	○	○	—	○	×
高安全	○	○	○	—	○	○

4　評　　価

本節では，前節において示した 2 種類の暗号プロトコルの動的変更例に対して，2 節における提案方式を評価する．

通信状況等及び安全性の変化による動的変更例の各プロトコルを提案方式に適用する．通信状況等の変化によるプロトコルの動的変更例では，高速用プロトコルから低速用プロトコル及びその逆への動的変更が行われる．安全性の変化によるプロトコルの動的変更例では，低安全プロトコルから高安全プロトコル及びその逆への動的変更が行われる．高速用プロトコルと高安全プロトコルは同じプロトコルであるので，低速用，高速用・高安全，低安全の各プロトコルに対する動的変更の処理時間を表 3 に示す．ただし，「自動生成」は 2.2 節における手順（2）にて自動生成サーバが環境パラメータに応じてプロトコルを生成する時間，「検証」は同じく手順（2）にて高速検証サーバが生成されたプロトコルの安全性を検証する時間，「前処理」は同じ節における手順（4）にて動的コンパイラが定義ファイルを読み込みプロトコルを実行するまでの前処理時間をそれぞれ示して

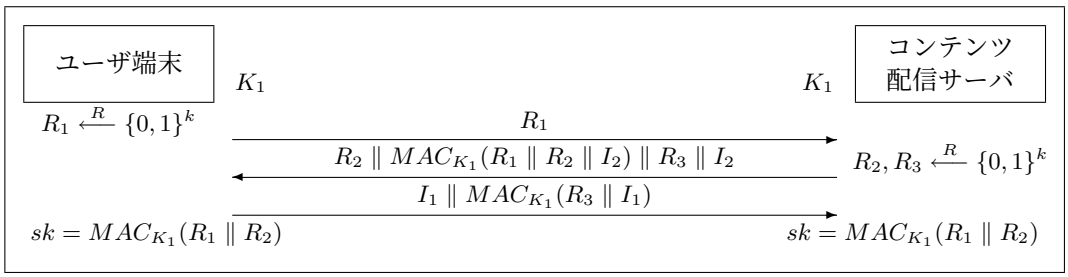

図 5　低安全プロトコルフロー

Fig. 5　Low-level security protocol flow.

表 3　各プロトコルの動的変更処理時間 [ms]

Table 3　Processing time of dynamic modification for each protocol.

種類	自動生成	検証	前処理	全処理
低速用	671	11.868	9.434	692.302
高速用・高安全	812	11.107	9.441	832.548
低安全	890	10.669	9.397	910.066

いる．ただし，自動生成・高速検証サーバは Intel Pentium D 2.8[GHz] プロセッサ及び 2.0[GByte] RAM のスペックであり，ユーザ端末とサービス提供サーバはともに Intel Core 2 Duo 1.2[GHz] プロセッサ及び 1.0[GByte] RAM のスペックである．また，**注 1** にて述べたとおり，環境パラメータと定義ファイルの各送付手順については議論の対象外であるため，今回未実装の手順（1）と（3）の通信時間は含まない．表 3 より，いずれのプロトコルにおいても，提案方式は全処理時間 1[s] 以下と現実の適用に十分な数値を達成している．

また，2009 年暗号と情報セキュリティシンポジウム内の展示セッションに本提案方式のデモシステムを出展した際に調査したアンケートから，図 6 の結果が得られた．本提案方式に関して，9 割弱と高い利用意向があった反面，半数がセキュリティ上の不安を感じたとのことである．その理由として，定義ファイルなどを安全に配送する仕組み，自動生成・高速検証ツールの第三者評価，システム全体の安全性評価が不足していることが挙げられた．

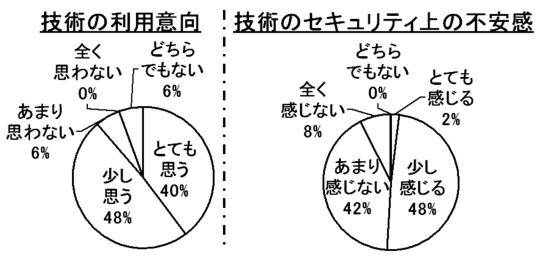

図 6　本提案方式に関するアンケート結果

Fig. 6　Questionnaire results with respect to the proposed mechanisms.

5　ま　と　め

本稿では，著者らの自動生成ツール，高速検証ツール及び動的コンパイラにもとづき，環境に応じた暗号プロトコルの動的変更方式を提案した．環境が頻繁に変化する状況として，通信状況等及び安全性の変化による暗号プロトコルの 2 種類の動的変更例を想定した．上記例の変更後のプロトコルを提案方式に適用した結果，いずれのプロトコルも 1[s] 以内に変更することができた．

今後の課題としては，定義ファイルなどの安全な配送方法や自動生成・高速検証ツールの評価を含めたシステム全体の安全性評価が挙げられる．

文　　献

[1] Perrig, A. and Song, D.: A First Step towards the Automatic Generation of Security Protocols, *Proc. Network and Distributed System Security Symposium (NDSS 2000)*, San Diego, CA, USA, The Internet Society, pp.73–83 (2000).

[2] Kiyomoto, S., Ota, H. and Tanaka, T.: Security Protocol Dynamic Generation and Modification Mechanisms for Ubiquitous Services, *Proc. 11th International Conference on Wireless Personal Multimedia Communications (WPMC'08)*, Lapland, Finland (2008).

[3] Bellare, M., Pointcheval, D. and Rogaway, P.: Authenticated Key Exchange Secure Against Dictionary Attacks, *Advances in Cryptology — EUROCRYPT 2000*, LNCS 1807, Bruges, Belgium, Springer-Verlag, pp.139–155 (2000).

[4] Dolev, D. and Yao, A.: On the Security of Public Key Protocols, *Proc. IEEE 22nd Annual Symposium on Foundations of Computer Science*, Nashville, TN, USA, pp.350–357 (1981).

[5] Ota, H., Kiyomoto, S. and Tanaka, T.: Security Verification for Authentication and Key Exchange Protocols, *International Journal of Computer Science and Network Security*, Vol.9, No.3, pp.1–11 (2009).

[6] Muller, F. and Millen, J.: Cryptographic Protocol Generation From CAPSL, *Technical Report*, No.SRI-CSL-01-07 (2001).