1F-5

セキュリティプロトコル自動生成手法の検討

清本 晋作† 太田 陽基† 田中 俊昭†
†KDDI 研究所

1 はじめに

セキュリティプロトコルは, 仕様の変更や新たなプロトコルの設計・検証には膨大な時間を要するため, 対策を迅速に施すことが困難であった. また, 利用するサービスごとに動的にプロトコルを生成して実行するといった機能は実現不可能であった.

一方, 認証プロトコルの自動生成を行なう手法の研究は行なわれてきたが [1][2], 多くの方式は, ランダムに多くの候補を生成し, そこから絞り込み処理を行うために処理時間がかかる, 様々なセキュリティ要件に基づいたプロトコル生成が行えない, 等の問題があり実用化には到っていない.

そこで, 本稿では通信相手や環境に応じて, 最適な二者間の認証・鍵共有プロトコルを高速に自動生成する手法を述べる.

2 自動生成手法

自動生成手法の検討課題としては, (1) プロトコル仕様の記述方法, (2) プロトコルに求められる要件の記述方法, (3) 自動生成手法, (4) 自動生成したプロトコルの検証手法, (5) プロトコル自動生成のためのエンティティ間のプロトコル, がある.

本稿では, 提案する認証・鍵共有プロトコルの自動生成手法における課題 (3), (4) についての検討結果を述べる. 提案方式では, 事前定義された雛形をセキュリティ要件等を考慮しながら組み合わせて１つのプロトコルを生成する手法を用い, 高速な自動生成を実現する. 従来の自動生成手法では, プロトコルの生成と, 生成したプロトコルが要件を満足するかどうかの検証は, 別々の処理として行われており, 結果として生成処理に時間を要していた. 提案方式では, プロトコルの生成過程において, セキュリティ要件, 適用環境に関する要件を満足するかどうかを逐次検証しながらプロトコルを生成することにより, 生成処理時間を短縮する.

2.1 処理概要

まず, 第１フェーズにおいて, 双方の計算能力, 通信環境などの環境情報と求められるセキュリティレベル等の要求条件を交換する. セキュリティレベルについては, 双方が, セキュリティポリシーと照らし合わせ受け入れ可能かどうかを判断する. 決定した基本データは, 符号化して交換することで, お互いが同様の基本データを保持していることを確認する. 次に, 第２フェーズとして, 双方のエンティティが交換した情報を基にプロトコルを生成する. プロトコル生成に際しては, 環境情報のうちセキュリティ要件を基にコンポーネントを組み合わせて基本プロトコルを構成するが, はじめに鍵共有プロトコルのコンポーネントを組み合わせてプロトコルを構成し, そのプロトコルに認証プロトコルに使用するコンポーネントを重ね合わせる. そして, 公開鍵の送信などの不足データの追加, 重なりのあるデータの削除, 鍵生成関数の作成というプロセスを経て, プロトコルの雛形を完成させる. 次に, 環境情報のうち, 性能要件に関する情報を利用して, 各データ長等を決定する. 最後に, 生成したプロトコル仕様を符号化して交換することで, お互いが同様のプロトコルを生成したことを確認する.

2.2 自動生成処理

自動生成処理は, 以下の手順で行なう.

1. プロトコルフレームワークの決定
   鍵共有プロトコル, 認証プロトコルの順に図１に示すコンポーネントを組み合わせてプロトコルを構成する. また, 各々のコンポーネント内の変数 X, Y, $Z1$, $Z2$ を, 鍵共有プロトコルパターン, 認証プロトコルパターンから, 基本データを基に選択しプロトコルのフレームワークを決定する.

2. 不足データ追加、重複データ削除
   不足データの追加処理では, 必要なデータの追加処理を行う. 例えば, プロトコルにおいてテンポラリな公開鍵–秘密鍵を含んでいた場合, 公開鍵を相手に送付する必要がある. 従って, テンポラリな公開鍵によって生成されたデータが送信されるフロー以前にテンポラリな公開鍵の送付を追加する. また, Identifier の送付が必要なフローには Identifier を追加する. 一方, 重複データの削除では, 以下のような処理を行う.

   (a) 同型性チェック
   プロトコルにおける各フローの同型性をチェックする. 同型性とは, 同一方向のフローで交換され

Study of Automatic Security Protocol Generation
† Shinsaku KIYOMOTO, Haruki OTA, and Toshiaki TANAKA
KDDI R & D Laboratories Inc.

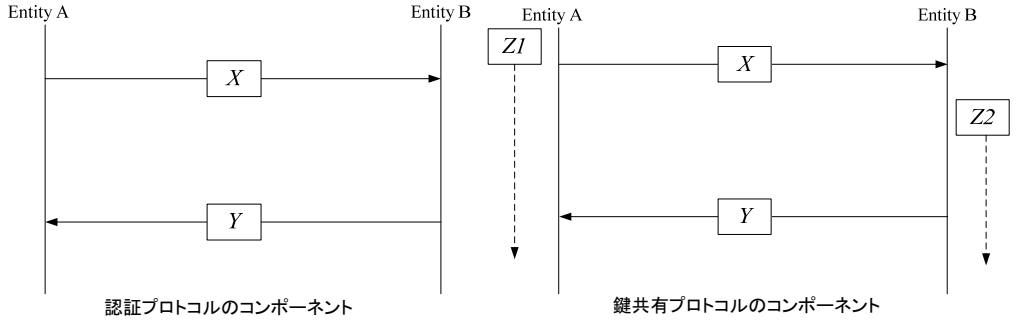

図 1: プロトコル構成用コンポーネント

るデータがほぼ同一のフォーマットになっていることを指す. 具体的には, データの種別が同一, 関数の構成が同一, でかつ作成したエンティティが同じデータを指す. この条件を満たすデータは, 同一とみなし, 先のフローで送付しているデータのみを残し, それ以降のデータを削除する.

(b) データの結合
同一方向のフローにおいて, 同一の鍵を用いて同一エンティティによって生成された関数を抽出し, その関数を結合させる. 例えば, 2 番目のフローで $F(X)$ というデータが送付されており, 4 番目のフローで $F(Y)$ というデータが送付されていた場合（鍵は同一), 第 2 番目のフローのデータを, $F(X\|Y)$ に置き換える。ただし, (c) の関連データの移動ができない場合には, データの結合は行わない. $\|$ はデータの結合を表す.

(c) 関連データの移動
同型性チェックやデータの結合によって, 関数が移動/削除された場合, その関数と同一フローで送付していた他のデータも, 統合先の関数を含むフローへ移動させる. 例えば, 2 番目のフローで X, $F(X)$ というデータが送付されており, 4 番目のフローで X, $F(Y)$ というデータが送付されていた場合（鍵は同一), 第 2 番目のフローの関数が, $F(X\|Y)$ に置き換えるため, Y も第 2 番目のフローに写す. また, X や Y が 1 つ前のフローで受信されていた場合には, 結合先（$F(X\|Y)$ を送るフロー）の 1 つ前のフローに移す.

(d)Known Data の削除
そのエンティティが既知であるデータを受信する場合には, そのデータを削除する. 例えば, A から B へ X というデータを送付している場合について, X を B が既知であれば, X を削除する. ただし, X が関数（の出力）であった場合は, この限りではない.

(e) フロー削除
データが存在しなくなったフローは削除する.

3. 鍵生成関数決定
プロトコルにおいて, 送信したデータ, 送信せずに保持しているデータ, 受信データ, 及び上記のデータから計算可能なデータが既知データとして格納される. 鍵生成関数は, 各エンティティの既知データ及び LL-Key（プロトコル開始以前に保有している秘密鍵）から構成される. 鍵生成関数は, 基本データを条件として, 鍵生成関数パターンの中から選択される.

4. 転送データ詳細、アルゴリズム、データ長の決定
基本データに従って, 各種アルゴリズムを選択し, またデータ長, 鍵長などの各パラメータをカスタマイズする.

3 おわりに

本稿では, コンポーネントの組み合わせによるセキュリティプロトコルの高速自動生成手法について述べた. 今後は, 提案した自動生成手法の実装を行ない, 性能, 利便性等の評価を行う予定である.

謝辞 本研究は, 独立行政法人情報通信研究機構 (NICT) の委託研究「ユビキタスネットワークにおける環境に応じたセキュリティプロトコルの自動生成・カスタマイズ技術に関する研究開発」の一環として行なわれた.

参考文献

[1] A. Perig and D. Song, “Looking for Diamonds in the Desert—Extending Automatic Protocol Generation to Three-Party Authentication and Key Agreement Protocols,” In Proc. of 13th IEEE Computer Security Foundations Workshop, pp.64-76, 2000.

[2] S. N. Foley and H. Zhou, “Towards a framework for autonomic security protocols,” In Proc. of Security Protocol Workshop 2003, pp.49-54, 2003.