二者間データ通信を安全に行うプロトコルの自動生成法

Construction method of Security Protocols for Data Exchange between Two Participants

† 佐藤 直人 Naoto Sato
‡ 萩原 茂樹 Shigeki Hagihara
‡ 米崎 直樹 Naoki Yonezaki

† 日立製作所システム開発研究所
Hitachi, Ltd., Systems Developemnt Laboratory
n-sato@sdl.hitachi.co.jp

‡ 東京工業大学大学院情報理工学研究科計算工学専攻
Department of Computer Science, Graduate School of Information Science and Engineering,
Tokyo Institute of Technology
{hagihara,yonezaki}@fmx.cs.titech.ac.jp

近年のインターネット上での重要な取引の増加に伴い、安全なプロトコルが必要不可欠となっている。本稿では二者間データ通信のためのセキュリティプロトコル自動生成法を提案する。これは、プロトコルの構造から安全性を導出する推論規則を利用し、安全なプロトコルのみを生成する手法である。また、プロトコルに対するコスト評価を行うことで、使用環境に適したより良いプロトコルを生成する。さらに、他の自動生成方式と比較することで、本稿で提案する自動生成法の利点についても議論する。

1 はじめに

個人情報や商取引に関する重要データがネットワークを通じて伝達されるようになった現在、これらの通信上のセキュリティを守るためには、暗号化技術が必須であるが、それだけでは十分ではない。たとえ、暗号化技術が完全であっても、伝送手順であるプロトコルに欠陥があればメッセージのすり替えや成り済まし攻撃を受ける可能性がある。よって、プロトコルの安全性を形式的に検証する技術が必要である。しかし、検証のコストを考えると、これまでは非形式的に行われてきたプロトコルの設計を形式的に行うアプローチの方が現実的である。

そこで本稿では、二者間でデータ通信を行うためのセキュリティプロトコルの自動生成法を提案する。この生成法では安全性が保証されているメッセージ構造を元に、安全なプロトコルのみを生成する。また、プロトコルに対する静的なコスト評価を行うことで、使用環境に適したセキュリティプロトコルを選別する。

以下ではまず前提条件を示した後、プロトコルの構造から安全性を推論するための推論規則を導入する。そしてこれを利用した自動生成手続きを示す。さらにこの推論規則の健全性をストランド空間 [3] における意味に基づいて証明する。最後に、他の自動生成法との比較も行う。

2 準備

本稿で自動生成の対象としているプロトコルの性質や、想定する攻撃者の振る舞い、プロトコルが満たすべき安全性等を定義する。

2.1 生成対象プロトコル

- 2 者間でデータをやり取りするためのプロトコルを生成する。データとは、ナンスや参加者の名前とは異なるメッセージで、これをお互いに通達することをプロトコルの目的とする。
- 参加者は受信したデータを返送することはない。
- データは必ず交互にやり取りする。
- １つの送受信メッセージに対して、同じ鍵で二度以上の暗号化、署名操作は行わない。
- ナンスは必ずセッションに対応付けて使用する。（自分の作成したナンスとセッションとの対応を、相手に認識させる）

- レスポンダは、ステップ 1 のメッセージからイニシエータの名前を知ることが出来る。
- 正当な参加者がセッションごとに作成するナンスは新規なものであり、以前に使用されたものを使用することはない。（ナンスの新規性）
- 参加者は一度に複数のセッションで通信を行うことが出来る。

2.2 攻撃者のモデル

本稿では、攻撃者の振る舞いや、その知識を以下のように想定する。

- 攻撃者は、通信される全てのメッセージを盗聴することが出来る。
- 攻撃者は、通信される全てのメッセージを他のメッセージとすり替えることが出来る。
- 攻撃者は対象とするネットワークにおいて、参加者として通信を行うことが出来る。
- 攻撃者は全ての参加者の公開鍵を知っている。
- セッション開始前に、攻撃者がそのセッション参加者の秘密鍵を知っていることはない。
- 攻撃者は、元々知っている知識や攻撃によって得た知識を元にメッセージを偽造し、正しいメッセージとすり替えることが出来る。

2.3 安全性

以下の 3 つの安全性を満たすプロトコルを生成する。

データの秘密性 秘密にすべきデータについて、データ作成者の意図した相手以外にそのデータは知られることのない性質。

データ認証 受信したデータが意図した相手からのものであることを受信者が確認できる性質。

受信確認 セッションで送信した全てのデータが、意図した相手に正しく受信されたことを送信者が確認できる性質。

2.4 プロトコルの仕様

生成するプロトコルの仕様は以下の項目からなる。仕様は自動生成器に対する入力として与えられる。

- やり取りするデータの数。
- 秘密にしたいデータ（何番目のデータを秘密にするか）。
- イニシエータとレスポンダのどちらからデータを送信し始めるか。

3 自動生成法

本節では、プロトコルの自動生成法について述べる。初めに自動生成に利用する論理を与え、その後これを利用した自動生成手続きについて述べる。また、プロトコルに対するコスト評価についても述べる。

3.1 論理

本節では、プロトコルの自動生成に用いる論理を与える。この論理を用いることで、前節であげた安全性の検証を行うことができる。

3.1.1 用語

A, B: それぞれ、イニシエータ、レスポンダを表す定数。

S_i, R_i: ステップ i での送信者、受信者を表すメタ変数。（ステップ i でイニシエータが送信者、レスポンダが受信者であるとき、$S_i = A, R_i = B$ となる。レスポンダが送信者、イニシエータが受信者であるとき、$S_i = B, R_i = A$ となる。）i が明らかの場合は、i を省略する。

X: 参加者を表すメタ変数

N_X: X が作成したナンスを表す定数。

K_X: X の公開鍵を表す定数。

K_X^{-1}: X の秘密鍵を表す定数。

M_i: ステップ i で送受信されるメッセージを表す定数。

D_i: ステップ i で送受信されるデータを表す定数。

M: メッセージを表すメタ変数。

N: ナンスを表すメタ変数。

K: 鍵を表すメタ変数。

ナンス、データ、参加者の名前をアトミックメッセージと呼ぶ。

3.1.2 式

式は、プロトコルの構造を表す式と、性質を表す式に分けることができる。プロトコルの構造を表す式は、[5] から引用した。

プロトコルの構造を表す式 $M_1 \sqsubseteq M_2$: M_1 が M_2 の部分メッセージであることを表す。ここで、メッセージ中で暗号化や署名に用いられる鍵は部分メッセージではない。以下のように帰納的に定義される。

- $M \sqsubseteq M$
- $M \sqsubseteq M'$ ならば、$M \sqsubseteq \{M'\}_K$
- $M \sqsubseteq M_1$ ならば、$M \sqsubseteq M_1 M_2$ $(M \sqsubseteq M_2 M_1)$

$M_1 \sqsubseteq_N M_2$: M_1 は M_2 の部分メッセージで、かつ、N が M_2 の部分メッセージであるとき、N は M_1 のみに出現することを表し、以下のように定義される。

- $M \sqsubseteq_N M$
- $M \sqsubseteq_N M'$ ならば、$M \sqsubseteq_N \{M'\}_K$
- $M \sqsubseteq_N M_1$ かつ $N \not\sqsubseteq M_2$ ならば、$M \sqsubseteq_N M_1 M_2$ $(M \sqsubseteq_N M_2 M_1)$

$i \vdash M_1 \downarrow_R M_2$: ステップ i で R が M_1 を受け取った際、M_1 から M_2 という情報を取り出せる。

$i \vdash R$ *knows* N: ステップ i 以前のメッセージから、R は N を知ることができる。

$i \vdash N_S \leftrightarrow S$: ステップ i で S は、自分の名前 S と自分が作成したナンス N_S を一緒に暗号化するか、S を N_S で暗号化して、相手に送信する。

プロトコルの性質を表す式 $i \vdash N$ *is not extracted*: イントルーダはステップ i までに送受信されたメッセージから N を得られない。

$i \vdash N$ *is not known*: N の作成者がステップ i まで実行した時点で、N はイントルーダに知られておらず、N の作成者とその意図した相手のみが知っている。

$i \vdash M$ *is addressed to* N_R: M_i のサブメッセージ M は、N_R の作成者 R への宛先情報を含んでおり、かつイントルーダによって偽造されることはない。

$i \vdash R$ *recognize* (N_S, N_R): N_R を作成した受信者 R は、ナンス N_S が意図した通信相手の作成したもので、セッションに対応したナンスであることをステップ i で認識する。（R は S を認証する）

$i \vdash M$ *is guaranteed*: M_i のサブメッセージ M は、イントルーダによる攻撃に対して耐性がある。つまりステップ i の受信者は、M は意図した通信相手が意図したステップで作成したものであることを確認出来る。

D_i *is warranted*: ステップ i の受信者は、D_i は意図した通信相手が意図したステップで送信したものであることを確認できる。（データ認証）

D_i *is secret*: データ D_i はその作成者の意図する相手以外に知られることはない。（データの秘密性）

3.1.3 推論規則

プロトコルの構造を表す式を導出する推論規則 プロトコルの構造を表す式は以下の規則により導出される。これらの規則は、[5] で導入された。

(DA1)M から M を取り出すことが出来る。

$$\frac{}{i \vdash M \downarrow_R M}$$

(DA2)M_1 から M_2 を取り出すことができ、M_2 から M_3 を取り出すことが出来るならば、M_1 から M_3 を取り出すことが出来る。

$$\frac{i \vdash M_1 \downarrow_R M_2 \quad i \vdash M_2 \downarrow_R M_3}{i \vdash M_1 \downarrow_R M_3}$$

(DA3)M から M_1 と M_2 の連接メッセージが取り出すことが出来るならば、M から M_1 を取り出すことが出来、かつ M から M_2 を取り出すことが出来る。

$$\frac{i \vdash M \downarrow_R M_1 M_2}{i \vdash M \downarrow_R M_j}$$

(DA4)R は K_R^{-1} を知っているため、$\{M\}_{K_R}$ から M と（情報として）R を取り出すことが出来る。

$$\frac{}{i \vdash \{M\}_{K_R} \downarrow_R R \ \wedge \ i \vdash \{M\}_{K_R} \downarrow_R M}$$

(DA5)R は K_S を知っているため、$\{M\}_{K_S^{-1}}$ から M と（情報として）S を取り出すことが出来る。

$$\frac{}{i \vdash \{M\}_{K_S^{-1}} \downarrow_R S \ \wedge \ i \vdash \{M\}_{K_S^{-1}} \downarrow_R M}$$

(DA6)R が N を知っていれば、$\{M\}_N$ から M と（情報として）N を取り出すことが出来る。

$$\frac{i \vdash R \ knows \ N}{i \vdash \{M\}_N \downarrow_R N \ \wedge \ i \vdash \{M\}_N \downarrow_R M}$$

(Knows1) イニシエータ A は N_A を初めから知っている。同様にレスポンダ B は N_B を初めから知っている。

$$\frac{}{i \vdash A \ knows \ N_A \ \wedge \ i \vdash B \ knows \ N_B}$$

(Knows2)i 以前のステップ j のメッセージから、R_i は N を取り出すことが出来る。

$$\frac{\exists j < i \ \ j \vdash M_j \downarrow_{R_i} N}{i \vdash R_i \ knows \ N}$$

(B1) 送信者 S_i は、ナンス N_{S_i} と名前 S_i をメッセージ $\{M\}_{K_{R_i}}$ に含めて相手に送信する。ここで、tag_i はタグを表しており、メッセージがステップ i 用であるという情報を持つ。仮にプロトコルの中で $\{M\}_{K_{R_i}}$ という形のメッセージが一意的であるならば、tag_i は必要ない。他の規則でも同様である。

$$\frac{i \vdash M_i \downarrow_{R_i} \{M\}_{K_{R_i}} \quad i \vdash M \downarrow_{R_i} N_{S_i} \quad i \vdash M \downarrow_{R_i} S_i \quad i \vdash M \downarrow_{R_i} tag_i}{i \vdash N_{S_i} \leftrightarrow S_i}$$

(B2) 送信者 S_i は、名前 S_i を含むメッセージをナンス N_{S_i} で暗号化して相手に送信する。

$$\frac{i \vdash M_i \downarrow_{R_i} \{M\}_{N_{S_i}} \quad i \vdash \{M\}_{N_{S_i}} \downarrow_{R_i} N_{S_i} \quad i \vdash M \downarrow_{R_i} S_i \quad i \vdash M \downarrow_{R_i} tag_i}{i \vdash N_{S_i} \leftrightarrow S_i}$$

プロトコルの性質を表す式を導出する推論規則　プロトコルの性質を表す式は、以下の規則により導出される。

(NN) ナンス N がサブメッセージとして含まれなければ、イントルーダに漏れることはない。

$$\frac{i-1 \vdash N \textit{ is not extracted} \quad N \not\sqsubseteq M_i}{i \vdash N \textit{ is not extracted}}$$

(NS) ナンス N_{S_i} を相手の公開鍵 K_{R_i} で暗号化すれば、相手 R_i 以外に漏れることはない。

$$\frac{i-1 \vdash N_{S_i} \textit{ is not extracted} \quad \{M\}_{K_{R_i}} \sqsubseteq_{N_{S_i}} M_i \quad i \vdash M \downarrow_{R_i} tag_i}{i \vdash N_{S_i} \textit{ is not extracted}}$$

(NR) ナンス N_{R_i} をその作成者の公開鍵 K_{R_i} で暗号化すれば、R_i 以外に漏れることはない。

$$\frac{i-1 \vdash N_{R_i} \textit{ is not extracted} \quad \exists j < i \quad j \vdash N_{R_i} \leftrightarrow R_i \quad \{M\}_{K_{R_i}} \sqsubseteq_{N_{R_i}} M_i \quad i \vdash M \downarrow_{R_i} tag_i}{i \vdash N_{R_i} \textit{ is not extracted}}$$

(K0) メッセージを送受信していなければ、ナンスは漏れない。

$$\frac{}{0 \vdash N \textit{ is not extracted}}$$

(KL) 全てのメッセージから N が漏れることがなければ、セッションを通して N は秘密である。

$$\frac{k \vdash N \textit{ is not extracted} \quad k \text{ は最終ステップ}}{k \vdash N \textit{ is not known}}$$

(K1) ステップ i まで実行した時点で N_{R_i} が秘密であり、ステップ $i+1$ のメッセージが N_{R_i} とタグを含む構造であれば、ステップ $i+1$ のメッセージがイントルーダによって偽造されることはない。よって R_i がステップ i までしか実行しなければ、相手 S_i もステップ i までしか実行することが出来ない。i までのメッセージから N_{R_i} は漏れないため、R_i がステップ i まで実行した時点では N_{R_i} は秘密となる。

$$\frac{i \vdash N_{R_i} \textit{ is not extracted} \quad i+1 \vdash M_{i+1} \downarrow_{R_{i+1}} \{M\}_{K_{R_{i+1}}} \quad i+1 \vdash M \downarrow_{R_{i+1}} N_{R_i} \quad i+1 \vdash M \downarrow_{R_{i+1}} tag_i}{i \vdash N_{R_i} \textit{ is not known}}$$

(K2)(K1) と同様

$$\frac{i \vdash N_{R_i} \textit{ is not extracted} \quad i+1 \vdash M_{i+1} \downarrow_{R_{i+1}} \{M\}_{K^{-1}_{S_{i+1}}} \quad i+1 \vdash M \downarrow_{R_{i+1}} N_{R_i} \quad i+1 \vdash M \downarrow_{R_{i+1}} tag_i}{i \vdash N_{R_i} \textit{ is not known}}$$

(K3)(K1) と同様

$$\frac{i \vdash N_{R_i} \textit{ is not extracted} \quad i+1 \vdash M_{i+1} \downarrow_{R_{i+1}} \{M\}_{N_{R_i}} \quad i+1 \vdash M \downarrow_{R_{i+1}} N_{R_i} \quad i+1 \vdash M \downarrow_{R_{i+1}} tag_i}{i \vdash N_{R_i} \textit{ is not known}}$$

(KK) ナンス N が、ステップ $i+1$ まで実行した時点で秘密ならば、ステップ i の時点でも秘密である。

$$\frac{i+1 \vdash N \textit{ is not known}}{i \vdash N \textit{ is not known}}$$

(AD1) 秘密の N_{R_i} を含み R_i の公開鍵で暗号化された M を作成できるのは、通信相手は N_{R_i} を作成した R_i であると信じている S_i のみ。

$$\frac{i \vdash N_{R_i} \textit{ is not known} \quad i \vdash M_i \downarrow_{R_i} \{M\}_{K_{R_i}} \quad i \vdash M \downarrow_{R_i} N_{R_i} \quad i \vdash M \downarrow_{R_i} tag_i}{i \vdash \{M\}_{K_{R_i}} \textit{ is addressed to } N_{R_i}}$$

(AD2)N_{R_i} と R_i を含み S_i の署名がなされた M を作成できるのは、通信相手は N_{R_i} を作成した R_i であると信じている S_i のみ。

$$\frac{i \vdash M_i \downarrow_{R_i} \{M\}_{K^{-1}_{S_i}} \quad i \vdash M \downarrow_{R_i} N_{R_i} \quad i \vdash M \downarrow_{R_i} R_i \quad i \vdash M \downarrow_{R_i} tag_i}{i \vdash \{M\}_{K^{-1}_{S_i}} \textit{ is addressed to } N_{R_i}}$$

(AD3)R_i を含み秘密の N_{R_i} で暗号化された M を作成できるのは、通信相手は N_{R_i} を作成した R_i であると信じている S_i のみ。

$$\frac{i \vdash N_{R_i} \textit{ is not known} \quad i \vdash M_i \downarrow_{R_i} \{M\}_{N_{R_i}} \quad i \vdash M \downarrow_{R_i} R_i \quad i \vdash M \downarrow_{R_i} tag_i}{i \vdash \{M\}_{N_{R_i}} \textit{ is addressed to } N_{R_i}}$$

(AD4) 既に S_i が N_{R_i} とその持ち主の名前 R_i の対応を認識しているのであれば、秘密の N_{R_i} で暗号化

された M を作成できるのは、通信相手は N_{R_i} を作成した R_i であると信じている S_i のみ。

$$\frac{i \vdash N_{R_i}\ \textit{is not known}\ \ \exists j < i\ \ j \vdash N_{R_i} \leftrightarrow R_i \quad i \vdash M_i \downarrow_{R_i} \{M\}_{N_{R_i}}\ \ i \vdash M \downarrow_{R_i} tag_i}{i \vdash \{M\}_{N_{R_i}}\ \textit{is addressed to}\ N_{R_i}}$$

(AD5)M を作成したのは、通信相手は $N^1_{R_i}$ を作成した R_i であると信じている S_i であり、M に $N^2_{R_i}$ が含まれるならば、M を作成できるのは通信相手は $N^2_{R_i}$ を作成した R_i であると信じている S_i のみである。

$$\frac{i \vdash M\ \textit{is addressed to}\ N^1_{R_i} \quad i \vdash M \downarrow_{R_i} N^2_{R_i}}{i \vdash M\ \textit{is addressed to}\ N^2_{R_i}}$$

(REC1)M を作成できるのは、通信相手を N_{R_i} を作成した R_i であると信じている S_i のみであるため、受信者 R_i は、N_{S_i} はそのような S_i の作成したナンスであることを認識する。

$$\frac{\begin{array}{c}(\text{ステップ } i \text{ 以前に } R_i \text{が送信した全ての } N_{R_i} \text{について})\\ i \vdash M\ \textit{is addressed to}\ N_{R_i}\ \ i \vdash M \downarrow_{R_i} N_{S_i}\end{array}}{i \vdash R_i\ \textit{recognize}\ (N_{S_i}, N_{R_i}) \wedge i \vdash M\ \textit{is guaranteed}}$$

(REC2)(REC1) によって S_i が N_{R_i} が意図した相手のナンスであることを確認した後、N_{R_i} を返送することによって、R_i は N_{S_i} が S_i の作成したナンスであることを確認できる。

$$\frac{\begin{array}{c}i \vdash M\ \textit{is addressed to}\ N_{R_i}\\ \exists j < i\ \ j \vdash S_i\ \textit{recognize}\ (N_{R_i}, N_{S_i})\end{array}}{i \vdash R_i\ \textit{recognize}\ (N_{S_i}, N_{R_i}) \wedge i \vdash M\ \textit{is guaranteed}}$$

(GS1) 受信者がセッションとの対応を認識する送信者の秘密のナンス N_{S_i} と、タグを含む暗号化メッセージは、意図した通信相手が意図したステップで送信したものであることを確認できる。i が最終ステップの場合は $i+1 \vdash M'\ \textit{is guaranteed}$ の条件は必要ない。

$$\frac{\begin{array}{c}i \vdash N_{S_i}\ \textit{is not known}\ \ \exists j\ \ j \vdash R_i\ \textit{recognize}\ (N_{S_i}, N_{R_i})\\ i \vdash M_i \downarrow_{R_i} \{M\}_{K_{R_i}}\ \ i \vdash M \downarrow_{R_i} N_{S_i}\\ i \vdash M \downarrow_{R_i} tag_i\ \ i+1 \vdash M'\ \textit{is guaranteed}\end{array}}{i \vdash \{M\}_{K_{R_i}}\ \textit{is guaranteed}}$$

(GS2) 受信者がセッションとの対応を認識する送信者のナンス N_{S_i} と、タグを含む署名メッセージは、意図した通信相手が意図したステップで送信したものであることを確認できる。

$$\frac{\begin{array}{c}i \vdash M_i \downarrow_{R_i} \{M\}_{K^{-1}_{S_i}}\ \ i \vdash M \downarrow_{R_i} N_{S_i}\\ i \vdash M \downarrow_{R_i} tag_i\ \ \exists j\ \ j \vdash R_i\ \textit{recognize}\ (N_{S_i}, N_{R_i})\end{array}}{i \vdash \{M\}_{K^{-1}_{S_i}}\ \textit{is guaranteed}}$$

(GS3) タグを含み、受信者がセッションとの対応を認識する送信者のナンス N_{S_i} で暗号化されたメッセージは、意図した通信相手が意図したステップで送信したものであることを確認できる。i が最終ステップの場合は $i+1 \vdash M'\ \textit{is guaranteed}$ の条件は必要ない。

$$\frac{\begin{array}{c}i \vdash N_{S_i}\ \textit{is not known}\ \ i \vdash M_i \downarrow_{R_i} \{M\}_{N_{S_i}}\\ i \vdash M \downarrow_{R_i} tag_i\ \ i \vdash \{M\}_{N_{S_i}} \downarrow_{R_i} N_{S_i}\\ \exists j\ \ j \vdash R_i\ \textit{recognize}\ (N_{S_i}, N_{R_i})\ \ i+1 \vdash M'\ \textit{is guaranteed}\end{array}}{i \vdash \{M\}_{N_{S_i}}\ \textit{is guaranteed}}$$

(GR1) 送信者がセッションとの対応を認識した、受信者の秘密のナンス N_{R_i} と、タグを含む暗号メッセージは、意図した通信相手が意図したステップで送信したものであることを確認できる。

$$\frac{\begin{array}{c}i \vdash N_{R_i}\ \textit{is not known}\ \ i \vdash M_i \downarrow_{R_i} \{M\}_{K_{R_i}}\\ i \vdash M \downarrow_{R_i} N_{R_i}\ \ i \vdash M \downarrow_{R_i} tag_i\\ \exists j < i\ \ j \vdash S_i\ \textit{recognize}\ (N_{R_i}, N_{S_i})\end{array}}{i \vdash \{M\}_{K_{R_i}}\ \textit{is guaranteed}}$$

(GR2) 送信者がセッションとの対応を認識した、受信者のナンス N_{R_i}、受信者の名前 R_i、タグを含む署名メッセージは、意図した通信相手が意図したステップで送信したものであることを確認できる。

$$\frac{\begin{array}{c}i \vdash M_i \downarrow_{R_i} \{M\}_{K^{-1}_{S_i}}\ \ i \vdash M \downarrow_{R_i} N_{R_i}\\ i \vdash M \downarrow_{R_i} tag_i\ \ i \vdash M \downarrow_{R_i} R_i\\ \exists j < i\ \ j \vdash S_i\ \textit{recognize}\ (N_{R_i}, N_{S_i})\end{array}}{i \vdash \{M\}_{K^{-1}_{S_i}}\ \textit{is guaranteed}}$$

(GR3) タグを含み、送信者がセッションとの対応を認識した受信者のナンス N_{R_i} で暗号化されたメッセージは、意図した通信相手が意図したステップで送信したものであることを確認できる。

$$\frac{\begin{array}{c}i \vdash N_{R_i}\ \textit{is not known}\ \ i \vdash M_i \downarrow_{R_i} \{M\}_{N_{R_i}}\\ i \vdash M \downarrow_{R_i} tag_i\ \ \exists j < i\ \ j \vdash S_i\ \textit{recognize}\ (N_{R_i}, N_{S_i})\end{array}}{i \vdash \{M\}_{N_{R_i}}\ \textit{is guaranteed}}$$

(WAR)M_i のサブメッセージである M は、意図した通信相手が意図したステップで作成したものであることを確認できるため、M_i に含まれる D_i も意図した通信相手が意図したステップで作成したデータであることを受信者は確認できる。

$$\frac{i \vdash M\ \textit{is guaranteed} \quad i \vdash M \downarrow_{R_i} D_i}{D_i\ \textit{is warranted}}$$

(SEC1)D_i は一度しか送受信されないため、相手の公開鍵で暗号化すればイントルーダに漏れることはない。

$$\frac{\{M\}_{K_{R_i}} \sqsubseteq_{D_i} M_i}{D_i \ is \ secret}$$

(SEC2)D_i が送信者の作成した秘密のナンス N_{S_i} で暗号化されていれば、イントルーダに漏れることはない。(k は最終ステップ)

$$\frac{k \vdash N_{S_i} \ is \ not \ known \quad \{M\}_{N_{S_i}} \sqsubseteq_{D_i} M_i}{D_i \ is \ secret}$$

(SEC3)D_i が認証した相手の秘密のナンス N_{R_i} で暗号化されていれば、イントルーダに漏れることはない。(k は最終ステップ)

$$\frac{k \vdash N_{R_i} \ is \ not \ known \ \ \{M\}_{N_{R_i}} \sqsubseteq_{D_i} M_i \quad \exists j < i \ \ j \vdash S_i \ recognize \ (N_{R_i}, N_{S_i})}{D_i \ is \ secret}$$

3.1.4 安全性を表す式

本節では、本論理においてどのように 2.3 節の安全性を表現しているかを述べる。

データの秘密性 $D_i \ is \ secret$ が導出されればステップ i で送受信されるデータ D_i の秘密性が保証される。

データ認証 やり取りされる全てのデータ D_i について $D_i \ is \ warranted$ が導出されれば、データ認証が満たされる。

受信確認 受信確認を満たすためには、データのやり取りが終わった後（データを含まない）確認メッセージを送受信すればよい [4]。よってやり取りしたいデータが D_k まで存在する場合、$k+1 \vdash M \ is \ guaranteed$ が導出されれば受信確認が満たされる。

3.2 プロトコルに対するコスト評価

本稿では、以下のコスト評価基準に基づきプロトコルを評価し、使用環境に適したプロトコルを選別する。

- メッセージの暗号化コスト
- メッセージの署名操作コスト
- メッセージの量的コスト
- 攻撃の検出時期コスト

やり取りされるメッセージの暗号化コストや量的コストはプロトコルの評価基準として一般的であるが、本稿ではこれに加え、攻撃が検出される時期に対してコスト基準を導入する。

本稿で生成するプロトコルは 2.3 節の安全性を満たしているため、攻撃者によってメッセージすり替え等の攻撃が行われたとしても、セッション参加者は必ずこれを検出することが出来る。しかし、その検出時期はプロトコルごとに異なり、全ての攻撃を即座に検出できるプロトコルもあれば、ある特定の攻撃に関しては後のステップでしか検出できないプロトコルもある。このような違いをコスト基準として導入した。本稿では、より早く攻撃を検出できるプロトコルの方が良いプロトコルであるという立場をとる。

3.3 自動生成法

本節では 3.1 節で示した論理を利用した、プロトコルの自動生成法を示す。まず初めに生成に必要な入力について述べ、その後プロトコルの生成手続きを示す。

3.3.1 入力

まず、2.4 節で挙げたプロトコルの仕様を以下のように形式化し、入力として与える。やり取りするデータの数とデータを送信し始める側が決定されれば、そのために必要なプロトコルの最小ステップ数が決定されるため、これをプロトコルのステップ数とする。これによりデータが送受信されるステップも決定される。

⟨step, Data, Secret⟩：プロトコルの仕様

step: プロトコルのステップ数

Data: データを送受信するステップ番号の集合

Secret: 秘密にしたいデータが送受信されるステップ番号の集合。Data の部分集合

さらに 3.2 節のコスト基準に対するコスト重みを入力として与えることで、3.2 の基準のうちどの基準を重視するかを指定することが出来る。つまり、コスト重みによってプロトコルが使用される環境が表現される。

3.3.2 生成手続き

プロトコルの生成手続きについて述べる。この生成法では、仕様と安全性を両方とも満たす構造を持

つメッセージを元に、安全なプロトコルのみを生成する。
(1) メッセージ生成器によりステップ i のメッセージ候補をコスト昇順に自動生成する。このメッセージ候補はプロトコルの仕様を満たしている。
(2) 生成されたメッセージ候補から、ステップ i に関する安全性の式が導出されるかを、推論規則を用いて調べる。候補メッセージをコスト昇順に調べることで、安全性の式が導出される最小コストメッセージをステップ i の送受信メッセージとして採用する。
(3) 全てのステップの送受信メッセージを決定し、プロトコルが生成される

以上の手続きにより、一つの安全なプロトコルが生成されるが、相手を認証するステップ等、仕様として与えられていない条件を変更することで、一つの入力から複数の安全なプロトコルが生成される。本生成法では、得られる可能性のあるプロトコルを全て生成する。その後プロトコル全体に対するコスト評価を行い、最小コストプロトコルを出力する。

4 本生成法の正当性

本節では、本生成法により生成されたプロトコルは、安全性を満たすことを示す。つまり、3.1 節で示した推論規則の健全性をストランド空間 [3] における意味に基づき証明する。

4.1 ストランド空間

本稿で扱うストランド空間の定義は [3] に従う。ストランドとは符号付メッセージの列である。ストランドはスレッドのようなもので、参加者によって順番に送受信されるメッセージの列を表す。符号付メッセージは $+M$ と $-M$ のどちらかであり、$+M$ によってメッセージ M の送信を表し、$-M$ によってメッセージ M の受信を表す。ストランドに現れるこれらの符号付メッセージをノードと呼ぶ。例えば $\langle +M_1, -M_2 \rangle$ はストランドであり、初めのノードで M_1 を送信し、二番目のノードで M_2 を受信する。また、ストランドの持つノードの総数をそのストランドの高さと呼ぶ。

また、攻撃者でない正当な参加者の振る舞いを表すストランドをレギュラーストランド、攻撃者の振る舞いを表すストランドをイントルーダストランドと呼ぶ。また、イニシエータの振る舞いを表すストランドをイニシエータストランド、レスポンダの振る舞いを表すストランドをレスポンダストランドと呼ぶ。

イントルーダストランドは、攻撃者が行える全ての振る舞いを考慮する必要があるため、攻撃者の行える基本動作の組み合わせで定義する。基本動作はアトミックメッセージの生成、メッセージの消失、メッセージの二重化、連結、分解、鍵使用、暗号化、復号化（署名）であり、これらを組み合わせることにより、2.2 節で定義した攻撃者の可能な行為を全て表現できる。

生成されるプロトコルのストランド空間について、以下の変数を定義する。

$a, b, \cdots$ ： 参加者の名前を表す変数
$n_a, n_b, \cdots$ ： $a, b, \cdots$ の作成したナンスを表す変数
$d_1, d_2, \cdots$ ： やり取りされるデータを表す変数

パラメータを利用してストランドを表記することも出来る。例えば、イニシエータの名前 a、レスポンダの名前 b、イニシエータのナンス n_a、レスポンダのナンス n_b、やり取りするデータ d_1, d_2 であるセッションにおける、高さ i のイニシエータストランドを $INI(a, b, n_a, n_b, d_1, d_2)_i$、レスポンダストランドを $RES(a, b, n_a, n_b, d_1, d_2)_i$ と表記する。さらに $*$ を利用した省略形を定義する。

$$INI(a, b, n_a, *) = \bigcup_{n_b} INI(a, b, n_a, n_b)$$

$*$ は任意の値を表しており、どのような値が入っても構わない。本稿ではしばしば、やり取りするデータを省略したストランド表記を用いるが、その場合もデータの値は任意である。

4.2 ストランド空間における式の意味

式に対して、ストランド空間における意味を与える。プロトコルの構造を表す論理式の持つ意味は、そのような構造を持つプロトコルのストランド空間を考慮しているという意味でしかないため、ここではプロトコルの性質を表す式に対してのみ意味を与える。

$i \vdash N$ *is not extracted*: N がイニシエータのナンスの場合を考える。高さ $i+1$ 以上の $INI(a, b, n_a, *)$ が存在せず、かつ高さ $i+1$ 以上の $RES(*, b, n_a, *)$ も存在しなければ、n_a を知っている可能性のあるレギュラーストランドは $INI(a, b, n_a, *)$ と $RES(*, b, n_a, *)$ のみであり、かつ $+n_a$ をノードとして持つイントルーダストランドは存在しない。N がレスポンダのナンスの場合も同様。

$i \vdash N$ *is not known*: N がイニシエータのナンスの場合を考える。高さ $i+1$ 以上の $INI(a, b, n_a, *)$ が存在しないならば、n_a を知っている可能性のあるレギュ

ラーストランドは $INI(a, b, n_a, *)$ と $RES(*, b, n_a, *)$ のみであり、かつ $+n_a$ をノードとして持つイントルーダストランドは存在しない。N がレスポンダのナンスの場合も同様。

$i \vdash M$ *is addressed to* N_R: ステップ i ではイニシエータが受信側とする。i 番目のノードに、M を含むメッセージの受信ノードを持つ $INI(a, b, n_a, *)_i$ が存在するならば、i 番目のノードに M を含むメッセージの送信ノードを持つ、高さ i 以上の $RES(a, b, n_a, *)$ が存在する。レスポンダが受信側の場合も同様。

$i \vdash R$ *recognize* (N_S, N_R): ステップ i ではイニシエータが受信側とする。$INI(a, b, n_a, n_b)_i$ が存在するならば、高さ i 以上の $RES(a, b, n_a, n_b)$ が存在する。レスポンダが受信側の場合も同様。

$i \vdash M$ *is guaranteed*: ステップ i ではイニシエータが受信側とする。i 番目のノードに、M を含むメッセージの受信ノードを持つ $INI(a, b, n_a, n_b)_i$ が存在するならば、i 番目のノードに M を含むメッセージの送信ノードを持つ、高さ i 以上の $RES(a, b, n_a, n_b)$ が存在する。
※ただし、a による b の認証が i より後のステップ j で行われる場合、この式の意味は以下のようになる
ステップ i ではイニシエータが受信側とする。i 番目のノードに、M を含むメッセージの受信ノードを持つ $INI(a, b, n_a, n_b)_j$ が存在するならば、i 番目のノードに M を含むメッセージの送信ノードを持つ、高さ j 以上の $RES(a, b, n_a, n_b)$ が存在する。レスポンダが受信側の場合も同様。

D_i *is warranted*: ステップ i ではイニシエータが受信側とする。i 番目のノードに、d_i を含むメッセージの受信ノードを持つ $INI(a, b, n_a, n_b, d_i)_i$ が存在するならば、i 番目のノードに d_i を含むメッセージの送信ノードを持つ、高さ i 以上の $RES(a, b, n_a, n_b, d_i)$ が存在する。加えて、受信者はステップ i の送受信メッセージから d_i を取り出すことが出来る。レスポンダが受信側の場合も同様。

D_i *is secret*: ステップ i ではイニシエータが送信側とする。i 番目のノードに d_i を含むメッセージの送信ノードを持つ $INI(a, b, n_a, n_b, d_i)$ が存在するならば、d_i を知る可能性のあるレギュラーストランドは $RES(*, b, *, *, d_i)$ のみで、かつ $+d_i$ をノードとして持つイントルーダストランドは存在しない。レスポンダが受信側の場合も同様。

4.3 推論規則の健全性

3.1 節の推論規則が、式に与えられたストランド空間における意味を保存することを証明する。これにより、定理 1 が示され推論規則の健全性が示される。

対象プロトコルの条件より、考慮するストランド空間では以下の性質が成立すると仮定する。これは、正当な参加者がセッションで作成するナンスは新規なもので、以前に使用されたものを使用することはないという意味である。

仮定 1 (ナンスの新規性) あるストランド空間にレギュラーイニシエータストランド $INI(a, b, n_a, n_b, d_1, \cdots)$ と、レギュラーイニシエータストランド $INI(a', b', n_a, n'_b, d'_1, \cdots)$ が存在するならば、$a = a', b = b', n_b = n'_b, d_1 = d'_1, \cdots$ が成立する。同様に、あるストランド空間にレギュラーレスポンダストランド $RES(a, b, n_a, n_b, d_1, \cdots)$ と、レギュラーレスポンダストランド $RES(a', b', n'_a, n_b, d'_1, \cdots)$ が存在するならば、$a = a', b = b', n_a = n'_a, d_1 = d'_1, \cdots$ が成立する。

(K1) の証明： N_{R_i} はイニシエータが作成したナンスであるとすると、ステップ i ではイニシエータが受信側となる。今、$INI(a, b, n_a, *)_i$ が存在するとする。仮にまだ $RES(*, b, n_a, *)$ の高さが i 以下であるとすると、$i \vdash N_{R_i}$ *is not extracted* より、n_a を知っている可能性のあるレギュラーストランドは $INI(a, b, n_a, *)_i$ と $RES(*, b, n_a, *)$ のみであり、かつ $+n_a$ をノードとして持つイントルーダストランドは存在しない。ここで、ステップ $i+1$ ではイニシエータが送信側であり、ステップ $i+1$ で送受信されるメッセージにはタグと秘密の n_a が含まれるため、イントルーダがこのメッセージの送信ノードを持つことはない。よって $RES(*, b, n_a, *)$ の高さが $i+1$ 以上になるためには、高さ $i+1$ 以上の $INI(a, b, n_a, *)$ が存在しなければならない。つまり、高さが $i+1$ 以上の $INI(a, b, n_a, *)$ が存在しないならば、高さ $i+1$ 以上の $RES(*, b, n_a, *)$ は存在しないことになる。今、$INI(a, b, n_a, *)$ の高さは i であることと $i \vdash N_{R_i}$ *is not extracted* より、n_a を知っている可能性のあるレギュラーストランドは $INI(a, b, n_a, *)$ と $RES(*, b, n_a, *)$ のみであり、かつ $+n_a$ をノードとして持つイントルーダストランドは存在しない。

(GR1) の証明： ステップ i ではイニシエータが受信側とする。i 番目のノードに、$\{M\}_{K_a}$ を含むメッセージの受信ノードを持つ $INI(a, b, n_a, n_b)_i$ が存在

するとする。この時、$i \vdash N_{R_i}$ *is not known* より $INI(a,b,n_a,n_b)_i$ 以外に n_a を知っている可能性があるのは $RES(*,b,n_a,*)$ のみであり、$i \vdash M \downarrow_{R_i} N_{R_i}$ より $RES(*,b,n_a,*)$ が存在する。さらに M が a の公開鍵で暗号化されていることから $RES(a,b,n_a,*)$ が存在し、$i \vdash M \downarrow_{R_i} tag_i$ より高さ i 以上の $RES(a,b,n_a,*)$ が存在することになる。ここで $j < i$. $j \vdash S_i$ *recognize* (N_{R_i}, N_{S_i}) より、高さ j 以上の $INI(a,b,n_a,*)$ が存在することになるが、ナンスの新規性より $INI(a,b,n_a,*)$ は $INI(a,b,n_a,n_b)$ であるため、$RES(a,b,n_a,n_b)$ が存在しなければならない。以上より i 番目のノードに $\{M\}_{K_a}$ を含むメッセージの送信ノードを持つ、高さ i 以上の $RES(a,b,n_a,n_b)$ が存在する。これ以外の規則に対する証明は、本稿では省略する。

定理 1 全ての D_i について D_i *is warranted* が導出されるとする。また、やり取りするデータが D_k まで存在し、ステップ k ではイニシエータが受信側であるとする。この時 $INI(a,b,n_a,n_b,d_1,d_2,\cdots,d_k)_k$ が存在するならば、高さ k 以上の $RES(a,b,n_a,n_b,d_1,d_2,\cdots,d_k)$ が存在する。逆にステップ k ではレスポンダが受信側であるとすると、$RES(a,b,n_a,n_b,d_1,d_2,\cdots,d_k)_k$ が存在するならば、高さ k 以上の $INI(a,b,n_a,n_b,d_1,d_2,\cdots,d_k)$ が存在する。

これはストランド空間における agreement プロパティである [3]。agreement プロパティは、あるストランドに対してそのパラメータと高さが一致した相手ストランドの存在を保証するものであり、認証の形式化のひとつである。

5 関連研究との比較

本節では関連研究との比較を行う。関連研究の特徴に触れながら、本稿で提案した自動生成法の利点について言及する。

5.1 Automatic Protocol Generator

[1] にて提案された自動生成法で、2 者間で認証を行いセッション鍵を秘密共有するためのプロトコルを自動生成する。まず、プロトコル生成器により、入力として与えられた仕様を満たすプロトコルを大量に生成し出力する。次に、得られたプロトコル群をプロトコル自動検証器にて検証し、安全性を満たすプロトコルを選別する。自動検証器には Athena[2] を使用する。そして、安全性を満たしたプロトコルのみを出力する。この生成法では、大量のプロトコルに対して検証を行う。これに対し本稿で提案した自動生成法では、安全性が保証されているメッセージ構造を元に設計を行うため、安全なプロトコルのみを効率的に生成することが出来る。

5.2 束縛関係に基く検証法

これは [5] にて提案された。推論規則によって、プロトコルの構造から直接的にプロトコルの秘密性や認証を検証する。束縛関係とは、ナンス等のメッセージ間の対応関係のことである。ある参加者が相手を認証するということは、相手の名前やナンスを自分の名前やナンスと対応づけることと見なせるため、この対応関係に基づいて認証を検証する。束縛関係に関する推論規則では、本稿で提案した推論規則と同様、プロトコルの静的な構造から安全性を検証することができる。よって、これを利用してプロトコルの自動生成を行うことも可能である。しかし束縛関係に基く推論規則を用いた場合では、生成されないようなプロトコルが存在する。束縛関係に基く推論規則では、ナンスの秘密性に関する導出を行った後にこれを利用して認証を導くことや、ナンスの秘密性に対する要件が厳しすぎるという理由から、ナンスを秘密共有しないプロトコルやセッション途中までしかナンスを秘密保持しないようなプロトコルは生成されない。本稿の手法でも、生成されないプロトコルが存在するが、本手法ではより厳密にナンスの秘密性について調べており、そのようなプロトコルでも生成出来るものがある。

6 まとめ

本稿では、二者間データ通信のための安全なプロトコルの自動生成法を提案した。これにより、使用環境に適した安全なプロトコルを効率的に自動生成出来るようになった。また、関連研究との比較を行い、本生成法の特徴について述べた。近年ではインターネット上において様々なサービスが行われており、求められる安全性も様々である。例えば、web サーバに対して過剰に接続し、サーバを動作不能にするサービス拒否攻撃と呼ばれる攻撃も多く報告されており、このような攻撃に対する耐性も求められている。このような、秘密性、認証以外の安全性も取り入れることが今後の課題である。

参考文献

[1] Adrian Perrig Dawn Song. Looking for diamonds in the dessert — extending automatic protocol generation to three-party authentication and key distribution. In *Proc. of IEEE Computer Security Foundation Workshop*, 2000.

[2] S.Berezin D.Song and A.Perrig. Athena: a novel approach to efficient automatic security protocol analysis. *Journal of Computer Security*, Vol. 9, No. 1, pp. 47–74, 2001.

[3] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Strand spaces: Proving security protocols correct. *Journal of Computer Security*, Vol. 7, pp. 191–230, 1999.

[4] 根岸和義, 米崎直樹. セキュリティプロトコルの一貫性および正常終了一致の同一参加者による複数セッションを考慮した検証法. 情報処理学会論文誌, Vol. 41, No. 8, pp. 2281–2290, 2000.

[5] 萩谷昌己, 竹村亮, 高橋孝一, 斉藤孝道. 束縛関係に基づく認証プロトコルの検証. コンピュータソフトウェア, Vol. 20, No. 3, pp. 17–29, 2003.