3-6　プライバシ確保型IPトレースバック技術の実現に向けて

3-6　Toward Realizing Privacy-Preserving IP-Traceback

野島　良
NOJIMA Ryo

要旨

インターネットの急速な発展に伴い、コンピュータウィルスやDOS攻撃等、ネットワークのセキュリティに関する問題が大きく取り上げられるようになってきた。IPトレースバック技術は重要な対策技術となりうるが、プライバシの問題等があり普及しなかった。

そこで本稿においては、IPトレースバックにおいてプライバシ確保を実現する方法について考える。特に、我々が研究・開発した紛失秘密鍵暗号のIPトレースバックへの有効性を示す。

The IP-traceback technology enables us to trace widely spread illegal users on Internet. However, to deploy this attractive technology, some problems have been remained unsolved. One of the biggest issues among them is the *privacy* problem. That is, there is a possibility of tracing not only the illegal users but also the *legal* ones.

In this paper, we show, by using the modern cryptography, the solution to the above problem. Especially, the effectiveness of our oblivious symmetric encryption to the privacy-preserving IP-traceback is introduced.

[キーワード]
IPトレースバック，プライバシ確保，紛失秘密鍵暗号
IP-traceback, Privacy-preserving, Oblivious symmetric encryption

1　はじめに

1.1　背景

インターネットの急速な発展に伴い、コンピュータウィルスやDOS攻撃等、ネットワークのセキュリティに関する問題が大きく取り上げられるようになってきた。そうした問題の中でも、我々は、DOS攻撃を行った不正ユーザを追跡する技術、すなわち**IPトレースバック技術**にこれまで注目してきた。IPトレースバック技術は非常に有用な技術とされているが、不正ユーザだけではなく、正当なユーザのプライバシをも暴露してしまう可能性がある。そこで我々は、プライバシを確保可能なIPトレースバックについても同時に研究・開発を行ってきた。

IPトレースバックとプライバシ確保型IPトレースバックに関する問題は、次のように簡単化することができる。まず、2人のユーザ（アリスとボブ）を考える。アリスはIPアドレスの集合$A = \{a_1, \ldots, a_n\}$を、ボブはIPアドレスaを保持しているものとする。ボブの目的は、Aの中にaが含まれているかどうか調べる事である。この問題は、ボブがaをアリスに送り、アリスがAの中にaが含まれているかどうかを調べる事により解決可能になる。実際にIPトレースバックでは、同様のことが行われる。一方、**プライバシ確保型**のIPトレースバックにおいては、問題が若干複雑になる。この技術を実現するためには、アリスがAを漏らさずに、そしてボブがaを漏らさずに、aがAに含まれているか調べることが必要になる。この一見解決不可能な問題を、我々は、紛失秘密鍵暗号という技術を開発・応用することにより解決した。本稿では、この紛失秘密鍵暗号を紹介する。

1.2 関連研究

先に考えた問題は、暗号プロトコルの研究分野で広く注目されてきた問題、**秘匿共通集合計算問題**（図1）の特殊な場合になっている。**1.1** と同じように書くと次のようになる：

> アリスとボブは、それぞれ秘密の集合 S_A、S_B を保持している。彼らは、S_A、S_B の共通集合のみを知りたい。しかしながら、相手にそれ以外の一切の集合の要素を教えることは避けたい。

例えば、$S_A = \{1, 345, 787, 88\}$、$S_B = \{9893, 3232, 89, 345\}$ とする。$S_A \cap S_B = \{345\}$ であるため、アリスは $\{1, 787, 88\}$ を漏らさずに、ボブは $\{9893, 3232, 89\}$ を漏らさないように、$S_A \cap S_B = \{345\}$ のみを取得可能にしたい。従って、この問題は、本来解きたい問題の一般化になっていると言える。

この問題に対する一般的な解決方法は、フリードマンらにより提案された[1]。

2 既存方式

2.1 加法に関して準同型性を有する公開鍵暗号方式

関連研究の紹介をするため、加法に関して準同型性を有する公開鍵暗号（図2）の紹介からはじめる。

公開鍵暗号方式においては、暗号化する（公開）鍵 pk と復号する（秘密）鍵 sk が異なる。秘密鍵 sk を保有するユーザ（受信者）は pk のみを公開する。メッセージ送信者は pk を使いメッセージを暗号化して、受信者におくる。受信者は sk を使い暗号文を復号し、メッセージを得る。ここで、メッセージ m の暗号文を $\mathrm{Enc}(m)$、あるいは $\mathrm{Enc}(sk, m)$ と記述することにする。準同型性を有する暗号系においては、秘密鍵 sk なしで、$\mathrm{Enc}(m_1)$、$\mathrm{Enc}(m_2)$ から $\mathrm{Enc}(m_1+m_2)$ を得ることが可能である。このような性質を持つ暗号系として、例えばパエリア暗号[2]やエルガマル暗号などがある。

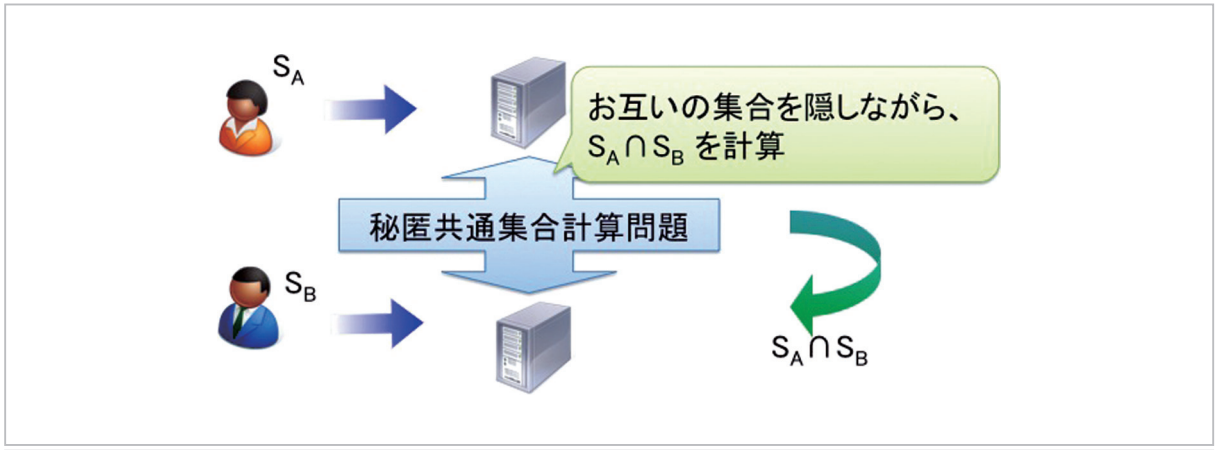

図1 秘匿共通集合計算問題

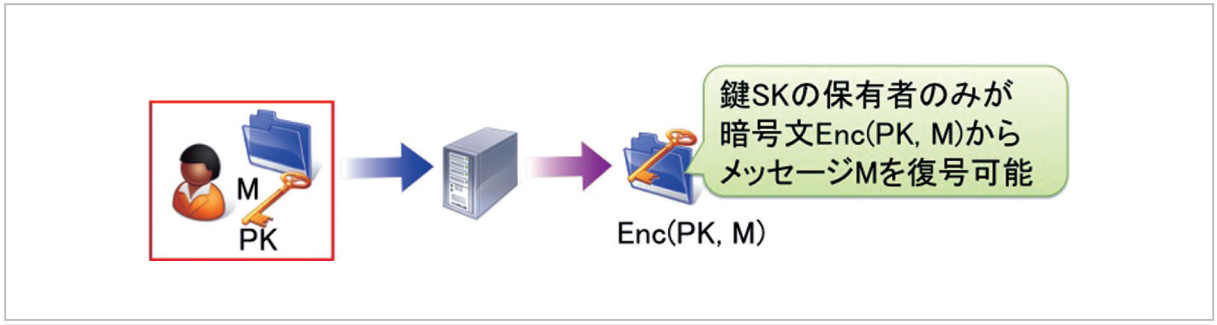

図2 公開鍵暗号の説明

2.2 既存研究の構成

はじめに（秘匿化していない）共通集合計算プロトコルの構成方法について考え、続けてその方式の秘匿化を行う。

普遍集合 U（$|U|=N$）とし、その部分集合に対するベクトル表現を考える。すなわち、S を普遍集合 U の部分集合とし、V を長さ N のベクトルとすると、$x \in S$ ならば $V[x-1]=1$、$x \notin S$ ならば $V[x-1]=0$ と定義する。例えば、$U=\{1, 2, 3, 4, 5\}$、$S=\{1, 3, 5\}$ ならば、S のベクトル表現 V は、$V=[1, 0, 1, 0, 1]$ となる。

方式1

入力：アリスの入力 S_A、ボブの入力 S_B

ステップ1：アリスは S_A をベクトル V_A に変換し、ボブに送る。

ステップ2：ボブは V_A と S_B から共通集合を出力する。

準同型性暗号を使い、方式1を秘匿化した方式に変換する。

秘匿共通集合計算プロトコル（方式1）

入力：アリスの入力 $S_A(V_A)$、pk、
ボブの入力 S_B (V_B)、pk、sk

ステップ1：ボブがアリスに $\mathrm{Enc}(V_B[0])$、$\mathrm{Enc}(V_B[1])$、…、$\mathrm{Enc}(V_B[N-1])$ を送る。

ステップ2：アリスは各 i に関して、$c_i=\mathrm{Enc}(r_i(V_B[i]-V_A[i])+i)$ を計算し、$\{(i, c_i)\}_i$ をボブに送る。ただし、r_i は各 i 毎に選ばれる乱数とする。

ステップ3：ボブは送られてきた暗号文を復号し、その要素が S_B に含まれる場合、共通集合に含まれるものとして、出力する。

通信量理論において、普遍集合のサイズが N であった場合、通信のコストは $\Omega(N)$ となることが知られているが、集合のサイズ n が $n\log N<N$ を満たす場合に特化すると、次のような効率的なプロトコルを構成することができる。

先ほどと同じように秘匿化されていない方式から考える。

方式2

入力：アリスの入力 S_A、ボブの入力 S_B

ステップ1：アリスはボブに S_A の各要素を送る。

ステップ2：ボブは S_A と S_B の共通集合を出力する。

この方式の通信量は $n\log N$ となる。すなわち、$N>n\log N$ ならば、こちらの方式の方が方式1よりも、時間計算量、通信計算量ともに効率が良くなる。

この方式を秘匿化した方式として、下記を考えることができる。

秘匿共通集合計算プロトコル（方式2）

入力：アリスの入力 $S_A=\{a_1, \ldots, a_n\}$、pk、
ボブの入力 $S_B=\{b_1, \ldots, b_n\}$、pk、sk

ステップ1：ボブはアリスに $\mathrm{Enc}(b_1)$、$\mathrm{Enc}(b_2)$、…、$\mathrm{Enc}(b_n)$ を送る。

ステップ2：アリスは各 i、j に関して、$\mathrm{Enc}(r_{ij}(b_i-a_j)+a_j)$ を送る。ここで、r_{ij} は乱数である。

ステップ3：ボブは送られてきた暗号文を復号し、その平文が S_B に含まれる場合、共通集合に含まれるものとして出力する。

この方式の通信量は $\mathrm{O}(n^2)$ となり、必ずしも効率的であるとはいいきれない。これに対する解決策は、フリードマンらにより提案された[1]。すなわち、彼らは集合の多項式表現を使い通信量を $\mathrm{O}(n)$ にまで引き下げることに成功した。

この方式に**バケットアロケーション**というテクニックを適用すると、時間計算量が大幅に改善される。

3 提案方式1

フリードマンらの方式は時間計算量が n に対して線形ではなく、現実社会で利用した際に満足できるものにはなっていない。そこで野島、門林[3]は、計算量が $\mathrm{O}(n)$ となる方式を提案した。

3.1 ブラインド署名

ブラインド署名は、2者（署名者、申請者）間の暗号プロトコルである。署名者は、署名鍵 sk と検証鍵 vk を、申請者はメッセージ M、vk をもっている。このプロトコルを使うと、互いに情報 sk、

Mを漏らす事なく、申請者が電子署名Sig(sk, M)(Sig(M)と省略することもある)を得る事ができる(図3、図4)。

3.2 方式

秘匿共通集合計算プロトコル
(Nojima-Kadobayashi)

入力: アリスの入力$S_A=\{a_1, \ldots, a_n\}$、pk、vk、
ボブの入力$S_B=\{b_1, \ldots, b_n\}$、vk

ステップ1: アリスはボブにH(Sig(a_1))、H(Sig(a_2))、…、H(Sig(a_n))を送る。Hはハッシュ関数である。

ステップ2: ボブとアリスは、ブラインド署名を動かす。アリスの入力はsk、ボブの入力は、b_1、…、b_nである。このプロトコルにより、ボブはH(Sig(b_1))、H(Sig(b_2))、…、H(Sig(b_n))を得る事が可能となる。

ステップ3: ボブは、H(Sig(a_1))、H(Sig(a_2))、…、H(Sig(a_n))とH(Sig(b_1))、H(Sig(b_2))、…、H(Sig(b_n))を比較することにより共通集合を得る(図5)。

ブラインド署名として、Chaumのブラインド署名[4]を利用することができる。この方式の計算量はO(n)となり、非常に効率的である。

図3 電子署名の説明

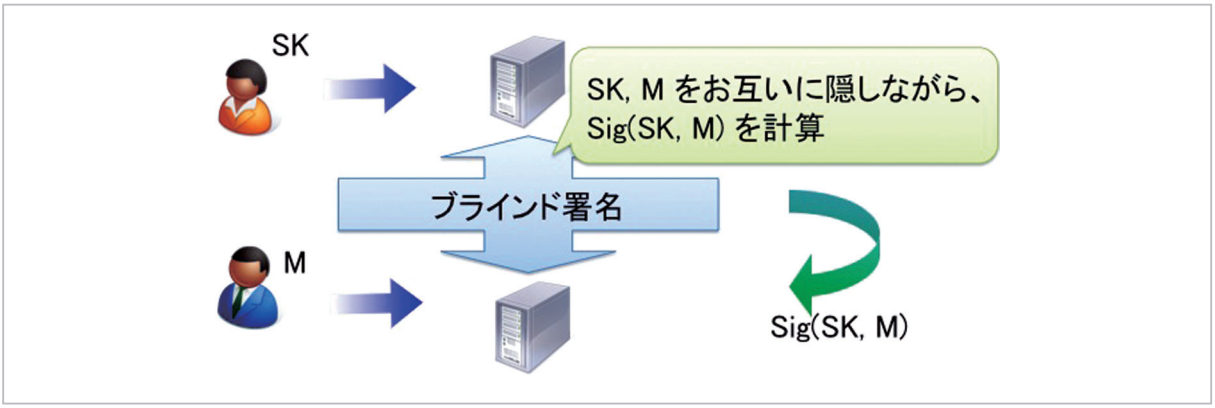

図4 ブラインド署名の説明

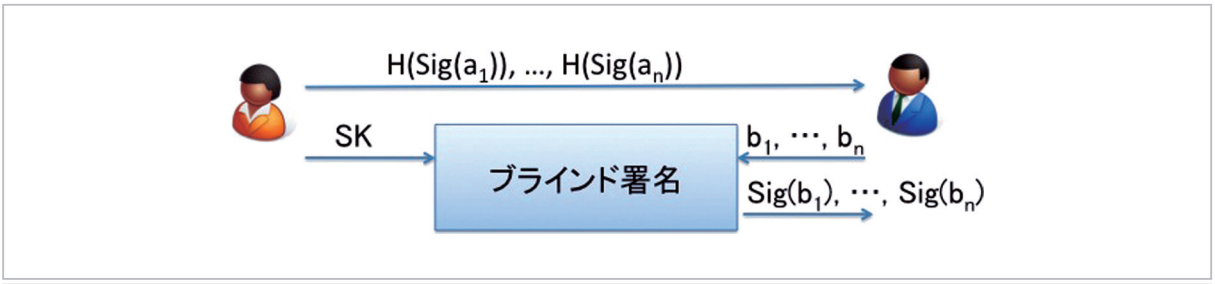

図5 ブラインド署名を使った秘匿共通集合計算プロトコル

3.3 IPトレースバックへの適応性

これまで紹介してきた暗号プロトコルにおいては、べき乗剰余を集合のサイズnに比例する数だけ計算しなければならない。しかしながら、IPトレースバックにおいては、そもそもnはパケットの数であり、その数だけべき乗剰余計算をさせるのは現実的に不可能であると考えられる。

そこで**4**においてはブラインド署名を使った方式を改良し、より現実的な方式を提示する。

4 提案方式2

4.1 秘密鍵暗号

秘密鍵暗号においては、秘密鍵skを使いメッセージMを暗号化することができる。この暗号化されたメッセージを$\mathrm{Enc}(sk, M)$と表す。ここで秘密鍵skを保有する人だけが、$\mathrm{Enc}(sk, M)$からMを取り出すことが可能となる。逆に、skを保有していない人はMに関する情報を一切得る事ができない（図6）。秘密鍵暗号として代表的なものに、DESとAESがある。

4.2 紛失秘密鍵暗号

紛失秘密鍵暗号プロトコル（以降、OEP）は、2者（アリス、ボブ）間の暗号プロトコルである。

アリスは秘密鍵暗号の秘密鍵skを、ボブはメッセージMを保有している。このプロトコルは、互いの情報skとMを秘密にしたまま暗号文$C=\mathrm{Enc}(sk, M)$を計算することを可能にする。ここで、もちろんCを得られるのはボブであり、アリスはCに関する情報を一切得る事ができない（図7）。我々は、秘密鍵暗号DESのOEPの設計・開発に成功した。方式の詳細については後ほど紹介する。

4.3 IPトレースバックへの応用

プライバシ確保型IPトレースバック技術において、アリスとボブは、互いの情報を隠しながら、aが$A=\{a_1, \ldots, a_n\}$に含まれているかどうかを検証する必要があった。この問題は、OEPを使うと簡単に解決できる（図8）。

(1) アリスは、秘密鍵暗号の秘密鍵skを選び、$\mathrm{Enc}(sk, a_1)$、…、$\mathrm{Enc}(sk, a_n)$をボブに送る。

(2) ボブは、OEPを使い$\mathrm{Enc}(sk, a)$を得る。そ

図6 秘密鍵暗号の説明

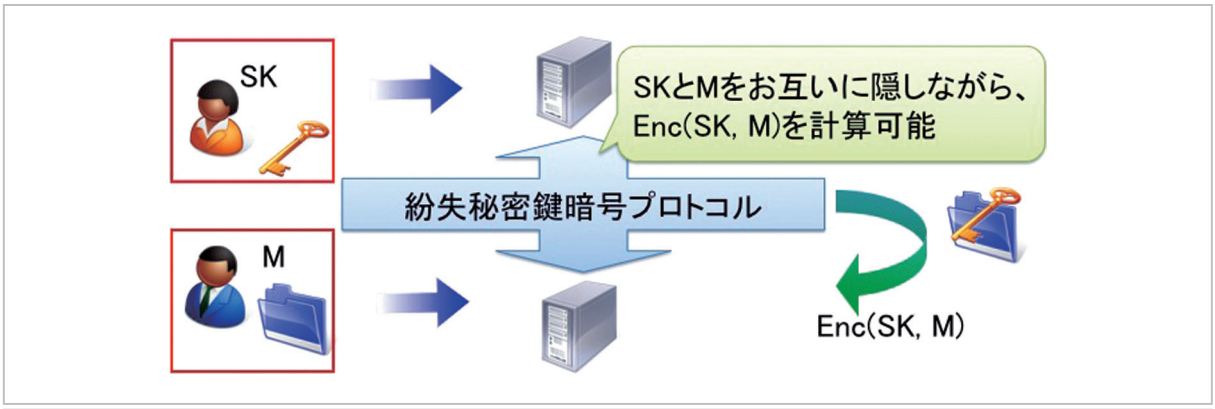

図7 紛失秘密鍵暗号プロトコルの説明

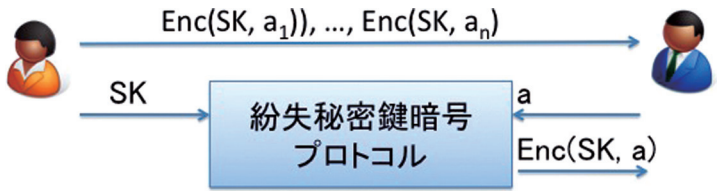

図８ 紛失秘密鍵暗号プロトコルを使った解決策

して、Enc(*sk*, a_1)、…、Enc(*sk*, a_n)の中に、Enc(*sk*, *a*)と同じになるものがあった場合、*a*が*A*に含まれていると判定する。

OEPを使うことにより、*sk*と*a*を互いに隠せるため、ボブの秘密情報である*a*がアリスに漏れる事はない。さらに、*sk*がボブに漏れないので、*n*個の暗号文からアリスの秘密情報*A*が漏れることもない。

このプロトコルにおけるOEPの役割は提案方式1のブラインド署名と同じである。本プロトコルの利点は、アリスがa_1、…、a_nの暗号文を計算するときに、べき乗剰余計算をしなくても良い点である。従って、IPトレースバックのように、パケット数*n*が膨大なときに適している。

4.4 拡張

IPトレースバックでは、パケットのハッシュ値の保存方法として、ブルームフィルタを利用することが多い。ここでは、ブルームフィルタを使った方式にOEPを適用する方法を紹介する。

ブルームフィルタベースのIPトレースバックの秘匿化[5]

プライバシ確保型IPトレースバック技術において、アリスとボブは、お互いの情報を隠しながら、*a*が$A=\{a_1, \ldots, a_n\}$に含まれているかどうかを検証する必要があった。ここで紹介する方式においては、アリスの集合はブルームフィルタに保存されている。ブルームフィルタで用いる配列の長さを2^mとし、使われるハッシュ関数を$H(x)=G(Enc(sk, x))$で定義する。Gは疑似乱数生成器であり、$H_1(x)$は$H(x)$の最初の*m*bit、…、$H_k(x)$は$H(x)$の最後の*m*bitとする。従って、*k*個のハッシュ関数が利用される。

(1) アリスは、秘密鍵暗号の秘密鍵*sk*を選び、$H(x)=G(Enc(sk, x))$を使いa_1、…、a_nを保存する。その暗号化されたブルームフィルタをボブに送る。

(2) ボブは、OEPを使いハッシュ値$G(Enc(sk, a))$を得る。そして、暗号化されたブルームフィルタと比較することにより、$a \in \{a_1, \ldots, a_n\}$かどうかを検証する。

安全性は、ブルームフィルタを使わない方式と同じ理由で保たれる。すなわち、OEPを使うことにより、*sk*と*a*が漏れないため、結果としてボブの秘密情報である*a*がアリスに漏れる事はない。逆に*sk*がボブに漏れないので、ブルームフィルタからアリスの秘密情報*A*が漏れることもない。

5 具体的構成方法[6]

5.1 準備：Modified ElGamal

DESベースの紛失秘密鍵暗号（Oblivious-DES）を構成するために、DDH仮定を応用した暗号方式（Modified ElGamal）を考える。

$G=<g>=<h>$を位数が素数*q*であるような群とする。この群の上では、DDH仮定が成り立つものとする。秘密鍵と公開鍵をそれぞれ以下のように定義する。

$$sk=x$$
$$pk=(q, g, h, g^x)$$

*x*は、Z_qからランダムに選ばれた元である。暗号化アルゴリズムと復号アルゴリズムを次のように定義する。

$$Enc(m; r)=(g^r, g^{rx}h^m)$$
$$Dec(c_1, c_2)=c_2/c_1^x$$

rは、Z_qからランダムに選ばれた元とする。従って、

$$\mathrm{Dec}(\mathrm{Enc}(m; r)) = h^m$$

が成り立つ。この方式のメッセージ空間は小さい（対数空間）が、oblivious-DESの設計には十分である。実際に、必要となるメッセージ空間は、

$$\{0, 1, \ldots, 15\}$$

と非常に小さい。実装においては、復号アルゴリズムが事前に、リスト$(h^0, h^1, \ldots, h^{15})$を用意しておくとよい。

5.2 提案方式

Oblivious-DESを設計するために、まず、Private-indexingプロトコルを設計する。Private-indexingプロトコルという概念は、NaorとNissimにより考案された[7]。彼らの設計においては、Oblivious Transferがブラックボックスとして利用される。一方、我々は、Modified ElGamalを使い具体的に構成する。この手法を利用する利点は大きく、効率面が格段にあがる。すなわち、Oblivious Transferを使うと安全性を保つためにGenericなZKが必要となるが、Modified ElGamalを使うと効率的なΣプロトコルのみで構成可能となる。

5.2.1 Modifed ElGamalを用いたPrivate-indexingプロトコル

d個の暗号文の列

$$\mathrm{Enc}(m_1), \ldots, \mathrm{Enc}(m_d)$$

を

$$\mathrm{Enc}_d(m_1, \ldots, m_d)$$

と記述する。

DBをインデックス0から$2^{d_1}-1$まである配列とする。DBのi番目の値をDB$[i]$と書くことにする。このとき、全てのiについて、DB$[i] \in \{0, 1\}^{d_2}$である。

π_1、π_2を長さd_1のビット列、π_3、π_4を長さd_2のビット列とすると、我々が実現したいプロトコルは、関数

$$f_{DB}(\pi_1, \pi_2) = (\pi_3, \pi_4)$$

と定義できる。ここで、

$$\pi_3 + \pi_4 = \mathrm{DB}[\pi_1 + \pi_2]$$

を満たす。すなわち、

入力π_1に対して送信者Sがπ_3を得る
入力π_2に対して受信者Rがπ_4を得る

Rが公開鍵・秘密鍵対(pk, sk)を、Sが公開鍵pkを保有しているものとする。この前提条件の下で、我々のPrivate-indexingプロトコルは次のようになる。

1. Rは$\mathrm{Enc}_{d_1}(\pi_2)$をSに送る。ここで、この暗号文列は、0か1の暗号文で構成されている。
2. Sは、$\mathrm{Enc}([\pi_1+\pi_2])$を$\mathrm{Enc}_{d_1}([\pi_1+\pi_2])$から計算する。
   次に、Sは長さd_2のビット列π_3をランダムに選び、すべての$0 \leqq i \leqq 2^{d_1}-1$について、
   $C^{(i)} = \mathrm{Enc}\}(r_i([\pi_1+\pi_2-[i]]+[\mathrm{DB}[i]+\pi_3])$
   を送る。ここでr_iはランダムに選ばれる。
3. Rは暗号文を復号し、
   $\pi_4 = \mathrm{DB}[\pi_1+\pi_2]+\pi_3$
   を得る。

5.2.2 Oblivious DESの基本的な構成方法

2人のパーティS、Rがいるものとする。SがDESの秘密鍵k、Rが64bitの平文mを持っている。このプロトコルのゴールは、Rが暗号文$\mathrm{DES}(k, m)$を得ることである。Private-indexingを使うと、このプロトコル（Oblivious DES）を下記のように構成することができる。

初期フェーズ

RとSは次のように動く。

- 入力mに対して、Rは(sk, pk)を生成し、pkを送信する。続けて、
  $m' = (m_1', \ldots, m_{64}') = \mathrm{IP}(m)$
  を計算する。さらに、
  $R_L^{(0)} = (m_1', \ldots, m_{32}')$、
  $R_R^{(0)} = (m_{33}', \ldots, m_{64}')$
  とおく。
- 入力kに対して、Sは副鍵$k^{(1)}$、…、$k^{(16)}$を計算する。
  ここで、
  $S_L^{(0)} = (0, \ldots, 0)$、
  $S_R^{(0)} = (0, \ldots, 0)$
  とおく。

iラウンド目（$1 \leqq i \leqq 16$）

- Sは、
  $E(S_R^{(i-1)}) + k^{(i)} = (\alpha_1, \ldots, \alpha_{48}) = (\beta_1, \ldots, \beta_8)$
  を計算し、Rは
  $E(R_R^{(i-1)}) = (\alpha_1', \ldots, \alpha_{48}') = (\beta_1', \ldots, \beta_8')$
  を計算する。ここで、α_i、$\alpha_i' \in \{0, 1\}$、$\beta_l = (\alpha_1, \ldots,$

α_8)、…、$\beta_8=(\alpha_{43}, \ldots, \alpha_{48})$、$\beta_1'=(\alpha_1', \ldots, \alpha_8')$、…、$\beta_8'=(\alpha_{43}', \ldots, \alpha_{48}')$とする。

- SとRは下記で定義されるPrivate-indexingを並列に動かす：

$$f_{S_1}(\beta_1, \beta_1')=(\gamma_1, \delta_1)$$
$$\cdots$$
$$f_{S_8}(\beta_8, \beta_8')=(\gamma_8, \delta_8)$$

ここで、S_1、…、S_8はS-boxである。次のような変数を考える：

$$\varepsilon=(\gamma_1, \ldots, \gamma_8)$$
$$\zeta=(\delta_1, \ldots, \delta_8)$$

- Sは、

$$S_R^{(i)}=P(\varepsilon)+S_L^{(i-1)}$$
$$S_L^{(i)}=S_R^{(i-1)}$$

と置き換え、

Rは、

$$R_R^{(i)}=P(\zeta) \oplus R_L^{(i-1)}$$
$$R_L^{(i)}=R_R^{(i-1)}$$

と置き換える。

最終フェーズ

- Sは

$$\eta=FP((S_L^{(16)}, S_R^{(16)}))$$

を送る。

- ηを受け取ったRは、

$$DES_k(m)=FP((R_L^{(16)}, R_R^{(16)})) \oplus \eta$$

を出力する。

6 おわりに

本稿においては、プライバシ確保型IPトレースバックを紹介した。特に、DESをベースとした紛失秘密鍵暗号の構成を提案し、そのプライバシ確保型IPトレースバックへの有効性を示した。実際に我々は、DESベースの紛失秘密鍵暗号の実装にも成功した。今後の研究課題としては、AESをベースとした効率的な紛失秘密鍵暗号を設計・実装することがある。

参考文献

1 Michael J. Freedman, Kobbi Nissim, and Benny Pinkas, "Efficient Private Matching and Set Intersection," EUROCRYPT 2004, 1–19.

2 Pascal Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," EUROCRYPT 1999, 223–238.

3 Ryo Nojima and Youki Kadobayashi, "On the Construction of the Set-Intersection Protocol from Blind Signatures," ISEC 2008–9, 57–60.

4 David Chaum, "Blind Signatures for Untraceable Payments," CRYPTO 1982, 199–203.

5 Ryo Nojima and Youki Kadobayashi, "Cryptographically Secure Bloom-Filters," Transactions on Data Privacy 2(2), 131–139, 2009.

6 Ryo Nojima and Youki Kadobayashi, "Oblivious Symmetric Key Encryption and Its Application to IP-Traceback," ISEC 2010-103, 199–203.

7 Moni Naor and Kobbi Nissim, "Communication preserving protocols for secure function evaluation," STOC 2001, 590–599.

（平成23年6月15日 採録）

野島　良（のじま　りょう）
ネットワークセキュリティ研究所
セキュリティ基盤研究室主任研究員
博士（工学）
暗号理論