n チャネルメッセージ伝送方式による暗号化通信

木下研究室　　渡邉 優司 (200402931)

1　はじめに

従来の公開鍵暗号方式では公開鍵の正当性を証明するために認証局のような信頼できる第三者機関が必要であった (図 1)。それに対して n チャネルメッセージ伝送方式では事前の鍵が不要なため第三者機関も必要ない。そこで本研究では n チャネルメッセージ伝送方式に着目し、より効率の良い新しい暗号化通信プロトコルを提案することを目的とした。

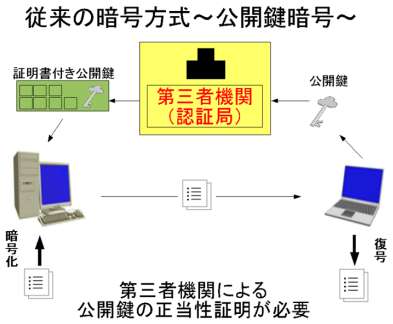

図 1: 公開鍵暗号方式

2　n チャネルメッセージ伝送方式

n チャネルメッセージ伝送方式は文書を n 本の通信路を使用して安全に送信する暗号化通信方式である。もし n 本のうちの何本かに文書を盗聴・改ざんする敵が潜んでいても、残りの通信路の情報を用いて文書を復号することができる (図 2)。次の 2 つの条件を満たす n チャネルメッセージ伝送方式を PSMT(Perfectly Secure Message Transmission) と呼ぶ。

1. 敵は送信メッセージに関する情報を何も得られない。(盗聴耐性)
2. 受信者がメッセージを正しく受信できる確率が 100 %である。(改ざん耐性)

また、送信者が受信者に 1 回送信するだけで済む方式を 1-round 方式、送信者と受信者が相互に r 回やり取りを行う方法を r-round 方式と呼ぶ。

PSMT は 1993 年に Dolev らによって提案された。彼らは、敵が n 本の通信路のうち t 本に潜んでいるとしたときに PSMT プロトコルが存在するための必要十分条件は、1-round 方式では $n \geqq 3t+1$、2-round 方式では $n \geqq 2t+1$ であることを証明し、また、それぞれ通信量が $O(n)$、$O(2^n)$ のプロトコルを提案した。その後 2-round 方式の通信量、計算量は改善され、2008 年に Kurosawa らは通信量 $O(n)$、計算量 $O(n^3)$ となるプロトコルを提案した。一方、1-round 方式は、通信量が $O(n)$、計算量が多項式時間となるプロトコルを Dolev らが最初に提案しており、既にそれが $n \geqq 3t+1$ における最善のプロトコルであった。PSMT においては、$n \geqq 3t+1$ でなければ使えない 1-round 方式よりも、$n \geqq 2t+1$ で使える 2-round 方式のほうが優れている。そこで、1-round 方式における必要十分条件を $n \geqq 2t+1$ に改善するために生まれたのが次に述べる ASMT(Almost Secure Message Transmission) である。

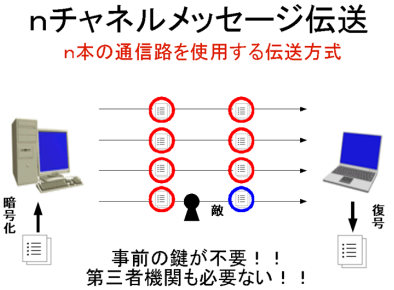

図 2: n チャネルメッセージ伝送方式

3　ASMT

ASMT の安全性の定義は以下のとおりである。

1. 敵は送信メッセージに関する情報を何も得られない。(盗聴耐性)
2. 受信者がメッセージを正しく受信できる確率が 1-δ 以上である。(改ざん耐性)
3. 受信者が正しく受信できない確率が δ 以下であり、そのとき受信者は failure を出力できる。(失敗検知能力)
4. 敵が t 本の通信路を遮断しても受信者は残りの通信路で得た情報だけからメッセージを受信できる。(遮断耐性)

ASMT は 2004 年に Srinathan らによって提案されたが、そのプロトコルには間違いがあった。その後 2007 年に Kurosawa らによって厳密に定義された。そのなかで $n = 2t+1$ のときの通信効率の限界が示され、限界に近い通信量で通信できるプロトコルが提案された。しかし、そのプロトコルは計算量が指数関数的であるという問題点がある。そこで本研究では計算量が多項式時間となる新しいプロトコルを提案する。

4　提案プロトコル (basic)

提案プロトコルのポイントはハッシュ関数を用いる点である。通信路の数を n、敵の数を t、送りたい秘密を S とすると、まず送信者は $f(0) = S$ となる t 次関数 $f(x)$ をランダムに生成する。次にハッシュ値 $H(f(1))$ ～$H(f(n))$ を計算する。そして各チャンネル ch-i に $f(i)$ と $H(f(1))$～$H(f(n))$ をセットにして送る。すると敵は盗聴した情報からは何も分からないが、受信者は受信した値 $f'(1)$～$f'(n)$ とハッシュ値 $H(f(1))$～$H(f(n))$ を上手く利用することで秘密 S を復号することができる。

5　おわりに

上記の提案プロトコル (basic) によって計算量を多項式時間にすることができたが、通信量はあまりよくない。そこで上記プロトコルをさらに改良することで通信効率の限界に近い通信量に改善することができた。また提案プロトコル (basic) をプログラミング実装した結果、正しく動作することが検証できた。