3-7　量子暗号における誤り検出と認証

3-7 Error Detection and Authentication in Quantum Key Distribution

山村明弘　　石塚裕一
YAMAMURA Akihiro and ISHIZUKA Hirokazu

要旨

量子暗号（つまり量子通信を利用した鍵交換スキーム）において未加工鍵の誤り検出と共有鍵の認証は、重要な問題である。量子暗号における未加工鍵における誤り検出と共有鍵の認証に関する実用的方法を提案することが本論文の目的である。ブール関数の近傍衝突がない特性に関する幾つかの概念を導入し、近傍衝突が起こらない関数と Reed-Solomon 符号などの誤り訂正符号に基づく方法を提案する。ブール関数の近傍衝突に関する性質はハッシュ関数の性質に密接に関連している。我々はまた、広く利用されている暗号ハッシュ関数 SHA-1 と MD5 が近傍衝突が起こらない特性を満たしているか否か、計算機による実験により検証する。

Detecting errors in a raw key and authenticating a private key are crucial for quantum key distribution schemes. Our aim is to propose practical methods for error detection and authentication in quantum key distribution schemes. We introduce several concepts about neighborhood collision free properties of Boolean functions, which are closely related to hash functions, and propose methods based on neighborhood collision free functions and error correcting codes such as Reed-Solomon code. We also examine whether or not widely used cryptographic hash functions SHA-1 and MD5 satisfy the neighborhood collision free property by computation experiments.

[キーワード]
量子暗号，誤り検出と訂正，近傍衝突，ハッシュ関数
Quantum cryptography, Error detection and correction, Neighborhood collision, Hash functions

1　まえがき

量子暗号は広く研究され（[1][2][8]）、実用化レベルに近づきつつある。物理的干渉が全くない研究室のような理想的な環境において、量子暗号は、情報理論的安全性な秘密鍵の共有手法を提供する。盗聴者 Eve の量子チャネルへの非合法なアクセスは、Heisenberg の不確定性原理により、Alice によって送られたフォトンのビットパターンを乱すので、Alice と Bobは、量子チャネルによるデータ転送後の誤り率を測定することにより、Eve の介在を検出することができる。誤り測定は古典チャネルを介した通信によって実施することができる。現実的な環境では、量子チャネルによるデータ転送においては、物理的な誤りが不可避的に起こる。Eve は、Alice と Bob が共有する秘密鍵に関して、ほんの少量の情報を得たいだけかもしれない。Eve にとって最善の攻撃戦略は、全データ転送量と比較して少量のデータを量子チャネルから盗聴し、結果として生じるデータの損傷を、量子チャネルやその他の周辺機器から不可避的に生じる物理的誤りと、Alice と Bob に見せかける手法であるだろう。この攻撃により、Eve は Alice と Bob が共有する秘密鍵の情報の一部を得ることができるかもしれない。このようなシナリオにおいては、誤りが起きるビットは、それ以外のビットよりも Eve の介在が疑わしいので、Eve が共有

鍵の部分情報を得ることを防ぐために廃棄されるべきである。量子暗号が情報理論的安全性を実現するためには以下の二つの段階が必要不可欠である。一つ目は、量子チャネルのデータ転送における誤り発生率を低下させることである。これは、光ファイバ、単一光子源生成機器、アバランチフォトダイオードなど物理的装置の改良に依存する。誤り率は、量子データ転送の距離にも依存する。つまり、距離が長いチャネルほど、誤り率は高くなる。二つ目は、未加工鍵における誤りを効率的に検出(更に訂正)し、漏えい情報を取り除き、Alice と Bob の共有鍵の認証を行うことである。本論文の目的は 2 番目の方法について、実用的な方法を提案することである。

まず、量子暗号の一般的なスキームについて簡単に説明する(より詳細な情報は、[6]の 2 章を参照)。最初に、Alice は(十分に長い)ランダムなビット列を生成し、このランダムビット列に対応するフォトンパルスを量子チャネルを通して送信する。このとき、ベースと偏光は、ランダムに決定される。Bob もランダムビット列を生成し、自分のランダムビット列に対応して決定されたベースで受信したフォトンパルスを測定する。このようにして Alice と Bob はそれぞれ、未加工鍵と呼ばれるビット列を得る。ここで Bob の未加工鍵と Alice の未加工鍵は全く違うものであることに注意したい。Bob は Alice がどのベースを選んだか知らないので、Alice と同じベースを選ばない限り、Alice の未加工鍵のビットを知ることができないからである。古典チャネルを通してベースの選択を確認することにより、Bob の未加工鍵に存在する誤りを見積もることができ、シフティド鍵を得る(このプロセスは、「シフティング」と呼ばれる。)。Eve が妨害しない限り、誤り率は物理的な装置の品質によりあらかじめ決められた値以下になる。もし Alice から Bob への量子チャネルのデータ転送を、Eve が大量に盗聴した場合、このEve の介在はこの段階で検出される。Alice と Bob は、あらかじめ決められた値より誤り率が高いことを発見するからである。盗聴するための Eve の最善の攻撃戦略は、量子チャネルの総データ転送量のうちごく少量のデータを盗聴することである。すると Eve に漏えいした情報は、多く見積もっても物理的デバイスにより発生する誤り率以下の情報になる。

次に、誤りは取り除くか又は訂正されなければならない。誤り訂正処理の後、Alice と Bob は、リコンシル鍵と呼ばれる、同一の鍵を持つ。漏えいした情報は無視できるほど小さいものであったとしても、Eve は量子チャネルと古典的チャネルの間の通信を盗聴して、リコンシル鍵の部分的な情報を持つ可能性がある。

三番目に、Eve に漏えいした情報は、プライバシー増幅を利用することにより、大幅に削減される。プライバシー増幅とは、リコンシル鍵において少数のビットを犠牲にすることにより、それに比べて指数関数的に増量して Eve に漏えいした情報を削減する手法である([3][4][10])。プライバシー増幅は、t-レジリエント関数[4]((N, J, K)関数[7]としても知られる。)を利用することにより、実行することができる。結果として生じた鍵は、秘密鍵と呼ばれる。

最後に、Alice と Bob は自分たちの秘密鍵の整合性を確認し、認証された秘密鍵を得る。量子暗号における典型的なプロセスを図 1 に示す。

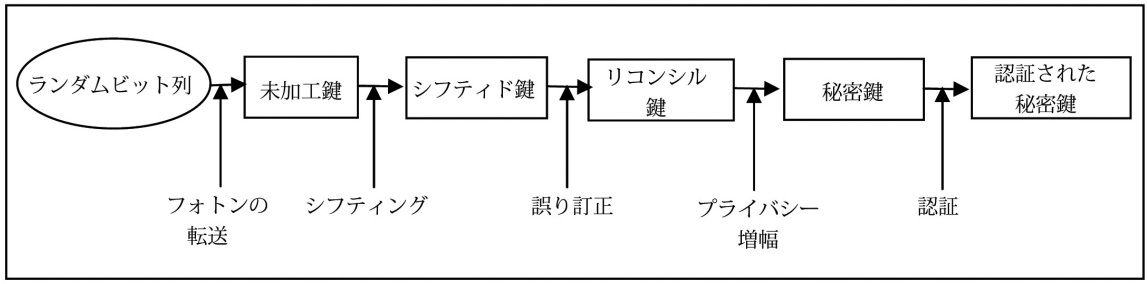


図 1 量子暗号におけるデータ処理

大局的又は局所的に近傍衝突がない関数の概念を紹介し、計算機実験により、SHA-1[9]とMD5[12]が近傍衝突がない特性を満足することを示す。未加工鍵の誤りを検出する方法と、近隣衝突がない関数を利用して量子暗号における秘密鍵の認証をする方法を提案する。この方法により、図1における誤り検出（訂正）と認証手順を実現できる。

2　誤り訂正方法

この節では、[4]と[5]における誤り訂正方法を簡単に説明する。AliceとBobは、量子暗号におけるシフティング処理後の、シフティド鍵を持っていると仮定する。Aliceがシフティド鍵rを持っている場合、Bobはシフティド鍵$r \oplus e$を持つ。ここで、$\oplus$はビット単位の排他的論理和を意味し、eは発生した誤りを意味する。eのハミング重みは、量子チャネルのデータ転送における物理的誤り発生率に依存し、近距離データ転送では、比較的低くなることが最近の物理的実験で明らかになっている。物理的な誤り発生率は、量子チャネルによる総通信量における誤りが起こったビットの比率である。もっとも理想的な条件下では、e=0となり、よって、AliceとBobは同一の鍵を持ち、Eveはその鍵に関する情報を一切持たない。現実的な状況下では、物理的誤りは不可避的に、一定の比率で起こるものであるが、それは非常にまれである。よって、eのハミング重みは、誤り率に比例し、0よりわずかに大きいだけである。したがって、eの大部分のビットが0であると仮定できる。同一の秘密鍵を共有するために、AliceとBobは誤りビットを取り除く必要がある。特に、彼らがこの同一の秘密鍵を、対称鍵暗号を利用した暗号通信における秘密鍵として利用するならば、同一の認証された秘密鍵を共有することが不可欠である。

最初に、Bennett、Bessette、Brassard、Salvail、Smolin[5]による、誤り訂正方法を説明する。Aliceは自分のシフティド鍵をブロックに分ける。Bobも、Aliceが実行した方法と同じ方法で自分のシフティド鍵をブロックに分割する。つまり、Aliceがシフティド鍵rを持ち、rを$r=r_1r_2 \cdots r_n$に分割した場合、Bobはシフティド鍵$r \oplus e$を持ち、$r \oplus e$を$r \oplus e=(r_1 \oplus e_1)(r_2 \oplus e_2) \cdots (r_n \oplus e_n)$に分割する。このとき、$e=e_1e_2 \cdots e_n$は誤りビットを表す。次に、Aliceは$r_i$の各ブロックのパリティを計算し、古典的チャネルを利用して、そのすべてをBobに送る。Eveは古典的チャネルを盗聴することができるので、このブロックのパリティを得ることができる。各ブロックのパリティは、1ビットの情報として考えられるので、各ブロック当たり1ビットの情報が漏えいしているとAliceとBobは想定する。Bobは自分のシフティド鍵に対応するブロックのパリティを計算し、Aliceから送られたパリティと比較する。比較した結果、すべてが一致した場合は、AliceとBobはたぶん同一の鍵を持つことになる。そうではない場合は、Aliceのブロックと Bobのブロックで最低1か所異なるはずである。このような場合、AliceとBobはパリティが異なるブロックを更に短いブロックに分け、異なるパリティがなくなるまで処理を続ける。Eveに漏えいした情報を無意味にするために、どの段階においても、AliceとBobは各ブロックの同じ位置から1ビットを消去する。処理を数回繰り返すことにより、最終的にAliceとBobは高い確率で同一の鍵を共有することになる。この方法の短所は以下のようなものである。AliceとBobが同一のリコンシル鍵を得る保証がない。多数のビットを無駄にし、相当な計算を必要とする。未加工鍵の生成処理において、AliceとBobは、リコンシル鍵を構築するために必要なビット数を、理論的に予測することができない、つまり、誤り訂正の効率性を理論的に予測することはかなり難しい。

2番目にBennett、Brassard、Robert[4]らの方法の一つを説明する。Aliceは、古典的チャネルを利用して、自分のシフティド鍵のハッシュ値を送ることを提案している。Bobは、自分のシフティド鍵のハッシュ値を同じように計算する。Bobはこの二つのハッシュ値を比較する。二つのハッシュ値が同一である場合、彼らは同一のリコンシル鍵を共有する。そうではない場合は、Bobは自分のシフティド鍵の数ビットを反転し、変更した鍵のハッシュ値を計算し、Aliceのシフ

ティド鍵のハッシュ値と一致するか否かチェックする。Alice のシフティド鍵のハッシュ値と一致するハッシュ値を持つ鍵を発見するまで、Bob はこの処理を続ける。Bob は、ビット列内で誤りが起こった位置を発見するために、誤りを検出するまで、基本的に全数探索を実行する。この手法は、ビット回転と呼ばれる。この方法の欠点は、Bob は膨大な計算を実行せねばならず、古典的チャネルを利用して転送したハッシュ値は、Eve にも大量の情報を与えることである。誤り率が非常に低く、ビット列が短いという、非常に制限された条件下でのみ、全数探索は実行できる。そうでなければ、この処理は不可能である。Alice が、誤り訂正符号により、自分のシフティド鍵を符号化し、符号化されたシフティド鍵の冗長部分だけを送るという方法も、[4] では提案されている。この方法の欠点もまた、符号化されたシフティド鍵の冗長部分は、Eve に相当の情報を与えることである。この方法には幾つかの欠点があるが、これらの欠点は、**4** で我々が示すように、改善することができる。

3 近傍衝突がない関数

H は Z_2^l から Z_2^k への Boolean 関数であるとする。直感的には、二つのハミング距離が小さいビット列をハミング距離が大きい二つのビット列に写像するときに H は近傍衝突がないといわれる。ビット列 x_1 と x_2 のハミング距離とは x_1 と x_2 で異なるビットを持つ位置の総数である。ビット列 x のハミング重みは、x とゼロ（つまり 0 のみから成るビット列）の間のハミング距離である。安全な通信のためには十分ではないが、この特性は、すべての（対称鍵、非対称鍵）暗号関数において満たされるべきものである。Boolean（ハッシュ）関数 H が衝突がないというのは、$r_1 \neq r_2$ であり、かつ $H(r_1)=H(r_2)$ であるビット列 r_1 と r_2 を発見することが（計算量的に）難しいことである。言い換えれば、$r_1 \neq r_2$ であり、かつ $H(r_1)$ と $H(r_2)$ のハミング距離が 0 であるビット列 r_1 と r_2 を発見することが難しいときに、Hは衝突がない。この概念は、以下のように、一般化される r と s のハミング距離を d(r, s) によって記す。ここで、$r, s \in Z_2^l$ である。$t \in Z_2^l$ に対して、集合 $\{s \in Z_2^l \mid d(s, t) < i\}$ は半径 i の t の近傍と呼ばれ、N(t, i) によって表記される。幾つかの近傍衝突に関する性質を定義する。H は、Z_2^l からへの Boolean 関数とする。

- $s, t \in Z_2^l$ で $H(s) \in N(H(t), j)$ となるもの、同値な条件として $H(t) \in N(H(s), j)$（又は $N(\{H(s), j/2\} \cap N(H(t), j/2)$ は空ではない）、を発見することが困難である場合、H は大局的に j- 近傍衝突がない関数である。
- すべての $u \in Z_2^l$ に対して、$s, t \in N(u, i)$ で $H(s) \in N(H(t), j)$ となるもの、同値な条件として $H(t) \in N(H(s), j)$（又は $N(H(s), j/2) \cap N(H(t), j/2$ は空ではない）、を発見することが困難である場合、H は i- 近傍局所的に j- 近傍衝突がない関数である。
- $H(s)=H(t)$ である $s, t \in Z_2^l$ を発見することが困難である場合、H は大局的に衝突がない関数である。
- すべての $u \in Z_2^l$ に対して、$H(s)=H(t)$ である、$s, t \in N(u, i)$ を発見することが困難である場合、H は i- 近傍局所的に衝突がない関数である。

これらの概念は、本論文における誤り検出及び認証システム構築において、非常に重要な役割を担う。この困難さの概念は、文脈に依存し、情報理論的又は計算量理論的のどちらにも解釈することができる。大局的に衝突がない特性は、暗号学的ハッシュ関数における無衝突性と一致する。大局的に j- 近傍衝突がない関数は、i- 近傍において局所的に j- 近傍衝突がない関数であり、大局的に j- 近傍衝突がない関数は大局的に衝突がない関数であり、大局的に衝突がない関数は、j- 近傍において局所的に衝突がない関数と i- 近傍において局所的に j- 近傍衝突がない関数は、i- 近傍において局所的に衝突がない関数であることは、簡単に分かる。逆は必ずしも真ではない。概念間の関係に関しては、図 2 を参照してほしい。

例えば、よいブロック暗号は強いアバランチ効果を示し、よって、少ない段数においても大局的に近傍衝突がない特性を満たす。大局的に近傍衝突がない特性は、アバランチ効果の一般化として考えられる。**5** で、SHA-1 と MD5 は大局的に近傍衝突がない特性を確かめる計算機

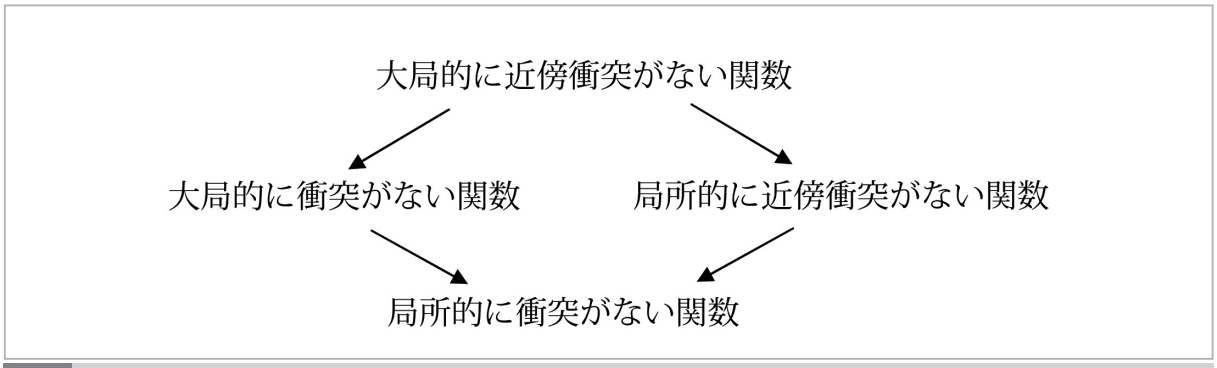


図2　衝突がない関数の相関図

実験の結果を示す。この実験で、SHA-1 は 43-近傍衝突がない特性を持ち、MD5 は 34- 近傍衝突がない特性を持つことが示されたが、理論的に厳密に証明することは難しい。

4　局所的に近傍衝突がない関数を利用した誤り検出

2 で説明した方法で、誤りを検出し、訂正するためには、多数のビットを浪費し、ランダム置換の反復など膨大な計算量が必要である。さらに、最終ステージにおいて、認証された秘密鍵の構築に成功するために、必要なビット数、つまり未加工鍵の長さを予測することは難解である。簡単に理論的に必要なビット数をあらかじめ予測することができるような、簡単で効率的な方法を発明することが望まれる。シフティド鍵の誤り検証のために、局所的に近傍衝突がない関数を利用する。

量子チャネルにおけるデータ転送の物理的誤り発生率は $\varepsilon>0$ であると仮定する。Alice と Bob はシフティング処理後のシフティド鍵に対して、ランダム置換を実行する必要があることに注意する。この処理が実行された場合は、誤りはランダムである、つまり誤りは Bob のシフティド鍵に一様に分布していると推測することができる。Eve が秘密鍵の特定の位置にあるビットを盗聴し（Eve の攻撃戦略に従って）、Alice と Bob はランダム置換を実行しない場合には、誤りは特定の位置に偏って出現する、つまり、誤りは Bob のシフティド鍵に偏って分布する。誤り測定処理後、Alice と Bob は、それぞれ自分たちのシフティド鍵 r と s を持つ。ここで、r, s $\in Z_2^{\,l}$ である。$r \oplus s$ は誤りビットパターンを示し、そのハミング重みはおおよそ $\varepsilon \times l$ と推定される。$0<\varepsilon<1$ と $0<\alpha<1$ は定数で α は ε より十分大きいと仮定する。H は $Z_2^{\,l}$ から $Z_2^{\,k}$ への局所的に近傍衝突がない関数とする。ランダムかつ一様に $Z_2^{\,l}$ から異なるビット列のペア (r_1, r_2) で r_1 と r_2 のハミング距離が $\varepsilon \times l$ より小さい又は等しいものを選んだときに、事象 $d(H(r_1), H(r_{12})) \leqq \alpha \times k$ の起こる確率を $\theta(H, \varepsilon, \alpha)$ とする。$0<\varepsilon \ll \alpha<1$ であるような定数 ε と α に対して、$\theta(H, \varepsilon, \alpha)$ が無視できる場合に Boolean 関数 H は局所的に近傍衝突がないと考えることができる。

誤り検出方法の基本的な考え方について説明する。H は局所的に近傍衝突がない関数であり、$\theta=\theta(H, \varepsilon, \alpha)$ は小さいと仮定する。このことは、$r \neq s \in Z_2^{\,l}$ で $d(r, s)<\varepsilon \times l$ に対して $d(H(r), H(s))<\alpha \times k$ である確率は無視できることを意味する。$r \oplus s$ のハミング重みが、$\varepsilon \times l$ より小さいと仮定する。よって、$r \neq s$ である場合、$H(s)$ は $N(H(r), \alpha \times k)$ には存在せず、同様に、H の局所的に近傍衝突がない特性により、$H(r) \oplus H(s)$ のハミング重みは、$\alpha \times k$ より大きい。もし $r=s$ ならば、$H(r)=H(s)$ であり、$H(r) \oplus H(s)$ のハミング重みは 0 である。

Alice と Bob はそれぞれ、自分のシフティド鍵の部分として、t と $t \oplus e$ を所有していると仮定する。ここで、$t, e \in Z_2^{\,k}$ であり、e のハミング重みはおおよそ $\varepsilon \times k$ である。$H(r) \oplus t$ と $H(s) \oplus (t \oplus e)$ の間のハミング距離は、$(\{H(r) \oplus t\} \oplus (H(s) \oplus (t \oplus e))=(H(r) \oplus H(s)) \oplus e$ により与えられる。よって、$r=s$ である場合、ハミング距離はおおよそ $\varepsilon \times k$ であり、さもなけれ

ばα×kより大きい。もし($\varepsilon+\alpha$)k／2を閾値として設定すれば、H(r)⊕tとH(s)⊕(t⊕e)の間のハミング距離が($\varepsilon+\alpha$)k／より大きいか、小さいかを検査することにより、Bobはr=sであるか否か決定することができる。

誤りの存在を発見するためのこの基準を、誤りが起こった正確なビット位置を発見するための幾つかの方法と結び付ける。次の節で幾つかの方法について述べる。最初の三つの方法の違いは、資源の消費(計算、量子チャネルのデータ転送、古典チャネルのデータ転送)にある。この違いは、計算、量子通信、古典的通信におけるトレードオフ関係があることを示している。

4.1 方法

lが要請されるリコンシル鍵のサイズであるとする。HはZ_2^lからZ_2^kへの局所的に近傍衝突がない関数であり、確率θ(H, ε, α)は無視でき、$\varepsilon \ll \alpha$とする。AliceとBobはHを利用すると仮定する。Hは秘密にする必要がないことに注意する。よって、EveもHを利用することができる。AliceとBobは、シフティング処理において、最初に2l+kビットのシフティド鍵を構築する。AliceとBobはそれぞれ、自分のシフティド鍵として、2l+kビットのバイナリ列rとr⊕eを持つ。ここで、eは誤りを表す。eのハミング重みは、おおよそ$\varepsilon \times |e| = \varepsilon \times l$である。基本的な考え方は、AliceとBobは自分たちのシフティド鍵のl+kビットを犠牲にして、Eveに一切情報を漏えいすることなく、eの誤りビットを探知することである。そのとき、彼らはrを共有し、rが自分たちのリコンシル鍵であることに同意する。

Aliceは自分のシフティド鍵としてrを所有し、$r=r_1r_2r_3$であると仮定する。ここで、$r_1, r_2 \in Z_2^l$であり、$r_3 \in Z_2^k$である。Aliceはハッシュ値H(r_1)を計算し、古典的チャネルを利用して、$r_1 \oplus r_2$とH(r_1)⊕r_3をBobに送る。Eveは古典的チャネルを盗聴できる。Bobは、自分のシフティド鍵としてr⊕eを所有し、r⊕e=($r_1 \oplus e_1$)($r_2 \oplus e_2$)($r_3 \oplus e_3$)である。ここで、$e=e_1e_2e_3$かつ$e_1, e_2 \in Z_2^l$かつ$e_3 \in Z_2^k$である。Bobは$r_1 \oplus r_2$とH(r_1)⊕r_3を受け取る。このようにして、Bobは$r_1 \oplus e_1$, $r_2 \oplus e_2$, $r_3 \oplus e_3$, $r_1 \oplus r_2$, H(r_1)⊕r_3を所有する。彼はハッシュ値H($r_1 \oplus e_1$)を計算する。次に、彼は($r_1 \oplus r_2$)⊕($r_2 \oplus e_2$)=$r_1 \oplus e_2$と($r_1 \oplus e_2$)⊕($r_1 \oplus e_1$)=$e_1 \oplus e_2$を計算する。ビット列$e_1 \oplus e_2$は、ビット列e_1e_2に関する相当な情報を含む。Bobは(H(r_1)⊕r_3)⊕($r_3 \oplus e_3$)=H(r_1)⊕e_3と(H(r_1)⊕e_3)⊕(H(r_1)⊕e_1)=H(r_1)⊕(H(r_1)⊕e_1)⊕e_3を計算する。e_1が1を含まない場合、つまり、$r_1=r_1 \oplus e_1$である場合、H(r_1)=H($r_1 \oplus e_1$)を得る。この場合、H(r_1)⊕(H(r_1)⊕e_1)⊕$e_3=e_3$である。よって、H(r_1)⊕(H(r_1)⊕e_1)⊕e_3のハミング重みは、高い確率で、($\alpha+\varepsilon$)k/2より小さい。一方、e_1が1を含む場合、高い確率で、H(r_1)⊕(H(r_1)⊕e_1)⊕e_3は($\alpha+\varepsilon$)k/2より大きい。よって、H(r_1)⊕(H(r_1)⊕e_1)⊕e_3のハミング重みは($\alpha+\varepsilon$)k/2より大きいか小さいという閾値の基準により、e_1=0であるか否か決定することができる。

e=0であるならばAliceとBobはサイズlの同一の鍵r_1を構築したことになる。もしH(r_1)≠H($r_1 \oplus e_1$)であるならば、Bobは情報$e_1 \oplus e_2$からe_1を推定する(ビット回転)。彼は、情報$e_1 \oplus e_2$に対応する、$r_1 \oplus e_1$からビット回転させたビット列のハッシュ値を計算し、H(r_1)⊕e_3と比較する。Bobは最終的にH(r_1)=H($r_1 \oplus e_1 \oplus e'$)であるe'を発見する。(厳格に言えば、H(r_1)⊕e_3とH($r_1 \oplus e_1 \oplus e'$)の間のハミング距離が($\alpha+\varepsilon$)k/2より小さいe')。Hは局所的に近傍衝突がないので、彼がe'≠e_1でありH(r_1)=H($r_1 \oplus e_1 \oplus e'$)であるものを発見することは考えられない。したがって、高い確率でe'=e_1となり、Bobは量子データ転送において起こるすべての誤りを検出することができる。AliceとBobはこれらの誤りビットe_1=e'を削除又は訂正することができ、長さがlよりわずかに短いリコンシル鍵r_1'を構築することができる(誤りが検出される場合)。AliceとBobは誤りを訂正し(削除ではない)、誤りビットを再利用する場合は、サイズが正確にlであるリコンシル鍵r_1を共有するということに注意する。プライバシーを増幅し、彼らは敵の情報を自由に減らすことができる。

この方法の安全性に関して簡単に述べる。シフティド鍵の構築処理が信頼できる場合、Eve

は古典的チャネルを通した通信から情報を得ることができる。よって、Eve は $r_2 \oplus r_2$ と $H(r_1) \oplus r_3$ の情報のみ得ることができる。量子暗号のメカニズムにより、r_1, r_2, r_3 は互いに独立なランダムなビット列である。r_1 と $H(r_1)$ は、r_2 と r_3 をそれぞれ犠牲にすることにより、Vernam 暗号[14]としても知られるワンタイムパッドにより暗号化されていると考えることができる。このことから、ワンタイムパッドは情報理論的に安全な秘密通信を提供するので[13]、Eve は実質的に何も情報を得られないということを意味する。しかし、物理的実装の問題のため、Eve が少量の情報を得るための余地が残る。もし Eve が総データ転送量のうちのごく少量のデータを盗聴し、攻撃が成功し、リコンシル鍵 r_1 の一部の情報を得ることができる場合、その情報量は、最大 $2\varepsilon \times l$ ビットであると見積もることができる。この漏えいした情報は、プライバシー増幅処理により、取り除くことができる。

4.2 方法 2

Bob の計算力は高いと想定し、トレードオフとして Bob に大量の計算を要求することにより、量子データ転送量を減らすための方法を述べる。量子チャネルを利用したデータ転送は、古典的チャネルを利用したデータ通信と計算処理よりも負担が大きい。よって、Bob に豊富な計算資源がある場合は、Bob に大量の計算の実行を要求することは妥当なことと考えられる。前述のように、H は、Z_2^l からから Z_2^k への局所的に近傍衝突がない関数であり、確率 $\theta(H, \varepsilon, \alpha)$ は無視できる。ここで、$\varepsilon \ll \alpha$ である。

Alice は、自分のシフティド鍵として r_1 r_2 を所有する、ここで、$r_1 \in Z_2^l$ と $r_2 \in Z_2^k$ である。一方、Bob は、自分のシフティド鍵として、$(r_1 \oplus e_1)$ $(r_2 \oplus e_2)$ を保持すると想定する。ここで、e_1 と e_2 は誤りを表す。Alice はハッシュ値 $H(r_1)$ を計算し、古典的チャネルを利用して、Bob に $H(r_1) \oplus r_2$ を送る。通信は、ワンタイムパッドにより暗号化されていると考えることができる。転送されたビット量は、定数 k である。Bob は $H(r_1 \oplus e_1)$ と $(H(r_1) \oplus r_2) \oplus (r_2 \oplus e_2) = H(r_1) \oplus e_2$ を計算する。もし $H(r_1) = H(r_1 \oplus e_1)$ であれば、$H(r_1 \oplus e_1) \oplus H(r_1) \oplus e_2 = e_2$ であり、そのハミング重みはおおよそ $k \times \varepsilon$ である。もし $H(r_1) \neq H(r_1 \oplus e_1)$ であれば、H は局所的に近傍衝突がないので、$(r_1 \oplus e_1) \oplus H(r_1) \oplus e_3$ のハミング重みは、おおよそ $k \times \alpha$ である。α は ε よりはるかに大きいので、$H(r_1) \oplus e_2$ のハミング重みが $(\alpha + \varepsilon)k/2$ より小さいならば $H(r_1) = H(r_1 \oplus e_1)$ あり、それ以外は $H(r_1) \neq H(r_1 \oplus e_1)$ と高い確率でなると結論できる。もし $H(r_1) \neq H(r_1 \oplus e_1)$ であれば、Bob は $r_1 \oplus e_1$ の $\varepsilon \times l$ ビットまで、ランダムに回転させ、そのハッシュ値を計算し、$H(r_1) \oplus r_2$ と比較する。Bob は全数探索により、最終的に e_1 を発見する。e_1 はおおよそ $\varepsilon \times l$ ビットの 1 を持つので、Bob は $r_1 \oplus e_1$ をおおよそ $\varepsilon \times l$ ビットの回転が必要である。明らかに Bob の計算タスクは、r_1 の長さと誤り率 ε に依存する。

量子チャネルと古典的チャネルのデータ転送量について議論する。方法 1 では、Alice と Bob は、長さ l ビットのリコンシル鍵を生成するために、サイズ 2l+k のシフティド鍵を生成しなければならなかった。量子データ転送量は、2l+k に比例する。古典的データ転送量は、l+k である。他方、方法 2 では、量子データ転送量は、l+k に比例し、古典的データ転送量は、k である。

もう一つの方法 2 の利点は、Eve に漏えいする可能性がある情報量が方法 1 と比較して減少することである。総通信量（量子と古典的）が方法 1 より少ないからである。方法 1 では、Eve は最大 $\varepsilon \times (2k+l)$ ビットを盗聴することができると見積もられるが、方法 2 では、最大 $\varepsilon \times (k+l)$ ビットである。

方法 2 の欠点は、Bob に大量の計算を要求することである。ε は小さく、構築された鍵が小さい場合は、Bob の計算はデスクトップパソコンで実行できる。しかし、ε が大きく、鍵の長さが長い場合、計算は実行不能になる。

4.3 方法 3

我々は方法 1 と方法 2 の中間的な手法を与える。H は Z_2^l から Z_2^k への局所的に近傍衝突がない関数で、確率 $\theta(H, \varepsilon, \alpha)$ が無視できるものであり、$\varepsilon \ll \alpha$ とする。Alice は自分のシフティド鍵として、$r_1 r_2 r_3 r_4$ を所有し、ここで、$r_1, r_2, r_3 \in Z_2^{l/2}$ と $r_4 \in Z_2^k$ である。同様に、Bob は自

分のシフティド鍵として、$(r_1 \oplus e_1)(r_2 \oplus e_2)(r_3 \oplus e_3)(r_4 \oplus e_4)$を所有し、$e_1, e_2, e_3 \in Z_2^{l/2}$であり、$e_4 \in Z_2^k$である。列 $e_1\ e_2\ e_3\ e_4$は誤りを表す。Alice と Bob はリコンシル鍵 $r_1\ r_2$を共有するものと考える。ビット列 $e_1\ e_2$はおおよそ $\varepsilon \times l$ ビットの 1 を含む。Alice は $r_1 \oplus r_2 \oplus r_3$ と $H(r_1\ r_2) \oplus r_4$を計算し、古典的チャネルを利用して、それを Bob に送る。Bob は $(r_1 \oplus r_2 \oplus r_3) \oplus (r_3 \oplus e_3) = r_1 \oplus r_2 \oplus e_3$ と $(H(r_1\ r_2) \oplus r_4) \oplus (r_4 \oplus e_4) = H(r_1\ r_2) \oplus e_4$を計算する。彼は $(r_1 \oplus e_1) \oplus (r_2 \oplus e_2) = r_1 \oplus r_2 \oplus (e_1 \oplus e_2)$を計算し、次に $(r_1 \oplus r_2 \oplus e_3) \oplus (r_1 \oplus r_2 \oplus (e_1 \oplus e_2)) = e_1 \oplus e_2 \oplus e_3$を計算する。もし $r_1\ r_2$が $(r_1 \oplus e_1)(r_2 \oplus e_2) = (r_1\ r_2) \oplus (e_1\ e_2)$と等しいならば、$H(r_1\ r_2) \oplus e_4$と $H((r_1 \oplus e_1)(r_2 \oplus e_2))$の間のハミング距離はおおよそ $\varepsilon \times k$ である。一方、もし $r_1\ r_2$が $(r_1 \oplus e_1)(r_2 \oplus e_2))$と等しくないならば、$H(r_1\ r_2) \oplus e_4$と $H((r_1 \oplus e_1)(r_2 \oplus e_2))$ の間のハミング距離は $\varepsilon \times k$ より大きくなる。α は ε よりはるかに大きいので、$H(r_1\ r_2) \oplus e_4$と $H((r_1 \oplus e_1)(r_2 \oplus e_2))$の間のハミング距離が $(\varepsilon + \alpha)k/2$ より大きい又は小さいという閾値の基準により、Bob は $e_1\ e_2 = 0$ であるか否か決定することができる。もし $H(r_1\ r_2) = H((r_1 \oplus e_1)(r_2 \oplus e_2))$ ならば、Alice と Bob はリコンシル鍵 $r_1\ r_2$を共有する。もし $H(r_1\ r_2) \neq H((r_1 \oplus e_1)(r_2 \oplus e_2))$ ならば、Bob は情報 $e_1 \oplus e_2 \oplus e_3$ (ビット回転) を利用して、$e_1\ e_2$を推測する。この方法により、$e_1\ e_2$を発見することは、方法 2 より明らかに簡単であるが、方法 1 よりは難しい。

Alice と Bob が長さ l のリコンシル鍵を共有するためには、$r_1\ r_2$は長さ l である必要がある。$|r_1| = |r_2| = |r_3| = l/2$ であり、$|r_4| = k$ であることに注意する。よって、Alice と Bob は長さ $3l/2+k$ のシフティド鍵を生成しなければならない。k を無視すれば、彼らはリコンシル鍵の長さの約 $3l/2$ の長さのビット列を生成する必要がある。一方、方法 1、方法 2 においてはそれぞれサイズ 2l と l のシフティド鍵が必要になる。

4.4 誤り訂正符号を利用する方法

誤り訂正符号を利用する方法について簡単に述べる。H は Z_2^1 から Z_2^k への局所的に近傍衝突がない関数であり、確率 $\theta(H, \varepsilon, \alpha)$ は無視できるものであり、$\varepsilon \ll \alpha$ とする。Alice と Bob のシフティド鍵の誤りを訂正するために、Alice は古典的誤り訂正符号により、自分のシフティド鍵を符号化し、符号化されたシフティド鍵の冗長部分だけを転送することを考えるかもしれない。しかし、冗長部分には、Alice のシフティド鍵の情報を大量に含むので、Eve にその情報がわたることを防ぐためには、冗長部分を暗号化する必要がある。

ワンタイムパッドにより、冗長部分を暗号化することを提案する。Alice は $r_1\ r_3$を自分のシフティド鍵として所有する、ここで、$r_1 \in Z_2^1$、$r_3 \in Z_2^k$とする。Bob は $(r_1 \oplus e_1)(r_3 \oplus e_3)$をシフティド鍵として所有する、ここで、$e_1 \in Z_2^1$、$e_3 \in Z_2^k$とする。Alice は、誤り訂正符号 C による r_1の符号化された語の冗長部分 ($C(r_1)$により表される) を計算する。Bob がシフティド鍵 $r_1 \oplus e_1$と $C(r_1)$の正しいビットの大部分を所有している場合、誤りビット列 e_1を検出し、訂正できる。Alice は $C(r_1) \oplus e_3$を送り、$C(r_1)$はワンタイムパッドにより暗号化されるので、Eve が盗聴できるとしても、実質的な情報は一切 Eve に与えられない。Bob は $C(r_1) \oplus e_3 \oplus (r_3 \oplus e_3) = C(r_1) \oplus r_3$を計算できる。よって、誤り率が十分に小さい場合は、Bob は C の誤り訂正能力により、誤りビットを訂正することができる。例えば、我々の目的のために、Reed-Solomon 符号[11]を利用することができる。この符号は、ランダム誤りを訂正する能力を持つからである。Alice と Bob はシフティング処理後、自分たちのシフティド鍵に対して、ランダム置換を実行したので、誤りはシフティド鍵全体に一様に分布していると仮定できることに注意する。

4.5 認証

リコンシル鍵生成後、Alice と Bob はプライバシー増幅を実行し、自分たちの秘密鍵を得る。次に、その秘密鍵の整合性を確認する。秘密鍵を認証するために、今まで説明してきたのと同じ考え方を利用することができる。既存の方法では、原理的に事前に認証された秘密鍵を共有することが必要であるが、我々の方法では事前

に共有する必要がないことに注意する。プライバシー増幅処理後、Alice は自分の秘密鍵 r_1 を所有し、Bob は自分の秘密鍵 r_1' を所有すると仮定する。ここで、$r_1, r_1' \in Z_2^1$ である。未加工鍵を作成するとき、Alice と Bob はそれぞれ余分にシフティド鍵 r_3 と $r_3 \oplus e_3$ を生成する。ここで、$r_3 \in Z_2^k$ であり、e_3 は誤りを表す。Alice は $H(r_1) \oplus r_3$ を Bob に送る。この通信は、ワンタイムパッドによる暗号化として考えられるので、Eve は実質的に全く情報を得られない。Bob は $H(r_1) \oplus r_3$ と $H(r_1 \oplus e_1)$ のハミング距離が閾値 $(\varepsilon + \alpha)k/2$ より小さいか否かを検証する。小さい場合は、$r_1 = r_1 \oplus e_1$ であり、$e_1 = 0$ である。そうでない場合は、$r_1 \neq r_1 \oplus$ である。この認証方法は、誤り訂正処理後に適用できる。我々はあらゆる誤り訂正とプライバシー増幅方法の後に、この方法が利用できることにも注意する。

4.6 実験結果

我々の誤り検出方法を実装するために、具体的な局所的に近傍衝突がない関数が必要である。計算機による実験により、SHA-1[9] と MD5[12] は局所的に近傍衝突がない特性を満たすことを示す。もし関数 H が局所的に近傍衝突がない特性を満たせば、小さいハミング距離を持つビット列 x_1, x_2 に対して、$H(x_1)$ と $H(x_2)$ のハミング距離は高い確率で、比較的大きくなると期待される。実験では、ハミング距離がそれぞれ 1、10、20 を持つビット列のペア (x_1, x_2) を N=100,000,000 個、ランダムに選ぶ。次にペア $(H(x_1), H(x_2))$ のハミング距離の頻度を計算する。もし H が暗号学的に安全なハッシュ関数であるならば、H が正規分布を示すことが推測される。もし標準偏差が比較的小さい場合、つまり多くのサンプルが平均値に近いハミング距離を持つ場合、その関数は、近傍衝突がないよい関数であるという結果できる。

ここでは SHA-1 を Z_2^{512} から Z_2^{160} への関数として考える。つまり、我々の実験では、SHA-1 の定義域を Z_2^{512} に限定する。平均値は 80 になり、大部分のペア (x_1, x_2) に対して、ハミング距離 $d(H(x_1), H(x_2))$ は 80 に近くなると推測する。実際、ハミング距離 1 (10, 20) を持つ、10,000,000 個のサンプルによる SHA-1 に関する我々の実験では、平均値は約 80、標準偏差は 6.3、$d(H(x_1), H(x_2))$ の最小値は 44、$d(H(x_1), H(x_2))$ の最大値は 115 であった。統計値に関しては表 1 を、ヒストグラムに関しては図 4 と図 5 を参照してほしい。我々の実験は、偏差が十分小さいことを示している。よって、SHA-1 は近傍衝突についてよい特性を持ち、よって、ハミング距離 1 のビット列のペアの大部分はハミング距離が 80 である列に写像される。例えば、$\alpha = 1/4$ と設定する。すると、すべての誤り率 $0 < \varepsilon < \alpha$ に対して、確率 $\theta(H, \varepsilon, \alpha)$ は無視できる。この場合の閾値は約 $(\varepsilon + (1/4)) \times 180)/2$ である。

MD5 は Z_2^{512} から Z_2^{128} への関数として考える。平均値は 64 になり、大部分のペア (x_1, x_2) に対して、ハミング距離 $d(H(x_1), H(x_2))$ は64に近くなると推測する。ハミング距離 1 (10, 20) を

表3 SHA-1 と MD5 の実験における統計値

アルゴリズム	ID	#data	平均値	標準偏差	ハミング距離 最大値	ハミング距離 最小値
SHA-1	1	10^8	80.000029	6.327076	115	44
	10	10^8	80.004204	6.334314	109	49
	20	10^8	79.994482	6.321717	111	47
MD5	1	10^8	63.999359	5.656389	95	34
	10	10^8	63.998326	5.658194	93	38
	20	10^8	63.995178	5.655455	92	37

表4 SHA-1 のハミング距離ヒストグラム

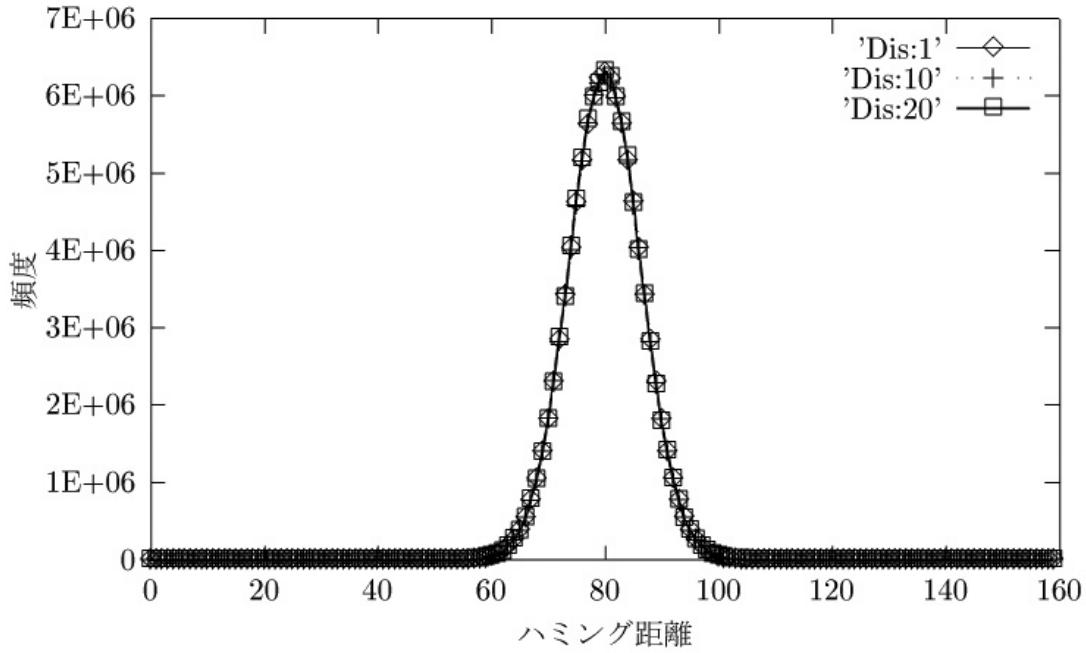
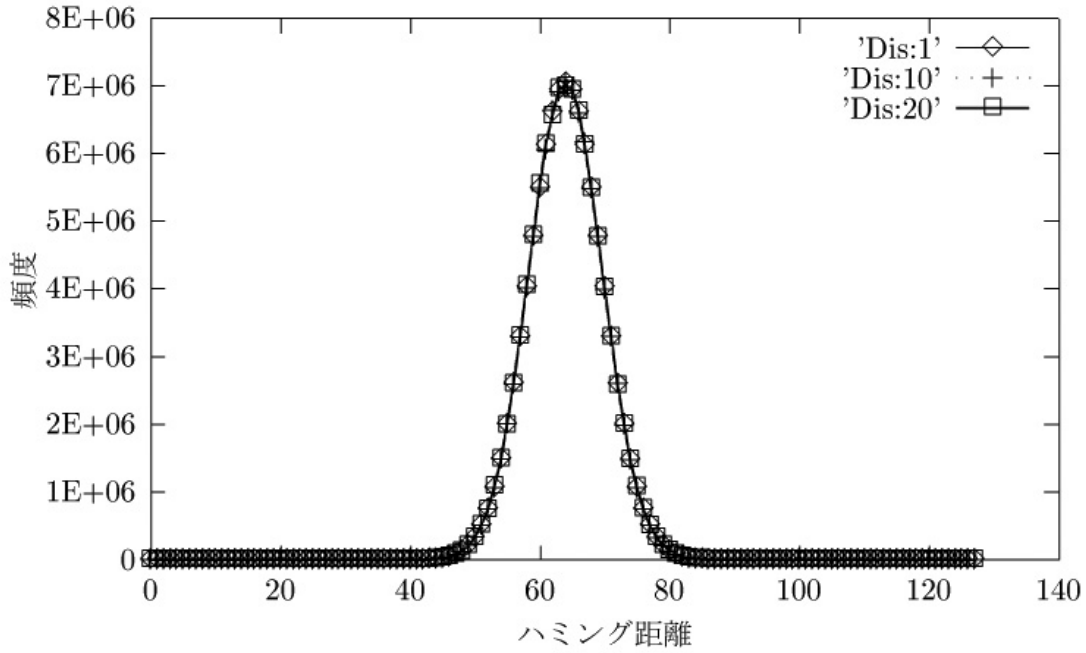


表5 MD5 のハミング距離ヒストグラム



持つ、10,000,000 個のサンプルによる MD5 に関する我々の実験では、平均値は 64、標準偏差は 5.6、$d(H(x_1), H(x_2))$ の最小値は 34、$d(H(x_1), H(x_2))$ の最大値は 95 であった。統計値に関しては表 1 を、ヒストグラムに関しては図 4 と図 5 を参照してほしい。我々の実験は、偏差が十分小さいことを示している。よって、MD5 は近傍衝突についてよい特性を持ち、ハミング距離 1 のビット列のペアの大部分はハミング距離が 64 である列に写像される。例えば、α =1/4 と設定

する。すべての誤り率 $0<\varepsilon<\alpha$ に対して、確率 $\theta(H, \varepsilon, \alpha)$ は無視できる。この場合の閾値は約 $(\varepsilon+(1/4))\times128)/2$ である。

図 4 と図 5 において、Dis：1、Dis：10、Dis：20と標示されたグラフは、ハミング距離 1、10、20 のヒストグラムを示す。

5 むすび

本論文は量子暗号における量子通信以外の部分の技術を開発することを目的とするものであった。今後は誤り検出・訂正と認証だけではなく、プライバシー増幅について研究を進めていくことが量子暗号の更なる発展に寄与すると考える。

参考文献

1 C.H.Bennett and G.Brassard, "Quantum cryptography : Public-key distribution and coin tossing", Proc. Int. Conf. on Computers, Systems and Signal Processing, Bangalore, India, pp.175-179, 1984.
2 C.H.Bennett, "Quantum Cryptography Using Any Two Nonorthogonal States", Phys. Rev. Lett., Vol.68, pp.3121-3124, 1992.
3 C.H.Bennett, G.Brassard, C.Crepeau, and U.M.Maurer, "Generalized privacy amplification", IEEE Trans. Information Theory, Vol.41, pp.1915-1923, 1995.
4 C.H.Bennett, G.Brassard, and J.M.Robert, "Privacy amplification by Public Discussion", SIAM J Comput., Vol.17, pp.210-229, 1988.
5 C.H.Bennett, F.Bessette, G.Brassard, L.Salvail, and J.Smolin, "Experimental Quantum Cryptography", J.Cryptology, Vol.5, pp.3-28, 1992.
6 D.Bouwmeester, A.Ekert, and A.Zeilinger, "The Physics of Quantum Information", Springer-Verlag, Berlin Heidelberg New York, 2000.
7 B.Chor, O.Goldreich, J.Hastad, J.Freidmann, S.Rudich, and R.Smolensky, "The Bit Extraction Problem or t-resilient Functions", 26th IEEE Symp. Foundations of Computer Science, pp.396-407, 1985.
8 A.K.Ekert, "Quantum Cryptography Based on Bell's Theorem", Phys. Rev. Lett. Vol.67, No.6, pp.661-663, 1991.
9 FIPS 180-1 : Secure Hash Standard, Federal Information Processing Standard (FIPS), Publication 180-1, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., April, 1995.
10 U.M.Maurer, "Secret Key Agreement by Public Discussion from Common Information", IEEE Trans. Information Theory, Vol.39, pp.733-742, 1993.
11 I.S.Reed and G.Solomon, "Polynomial Codes over Certain Finite Fields", J.Soc. Indust. Appl. Math. Vol.8, pp.300-304, 1960.
12 R.L.Rivest, "The MD5 Message-digest algorithm", Request for Comments (RFC) 1321, Internet Activities Board, Internet Task Force, April, 1992.
13 C.E.Shannon, "Communication Theory of Secrecy Systems", Bell Syst. Tech. J., Vol.28, pp.656-715, 1948.
14 G.S.Vernam, "Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Communications", J.Amer. Inst. Elect. Eng., Vol.55, pp.109-115, 1926.
15 H.Zbinden, H.Bechmann-Pasquinucci, N.Gisin, and G.Ribordy, "Quantum Cryptography", Applied Physics B, Vol.67, pp.743-748, 1998.

16 A.Yamamura and H.Ishizuka, "Detecting errors and authentication in quantum key distribution", Information Security and Privacy (ACISP2001), LNCS 2119, Springer-Verlag, pp.260-273, 2001.

山村明弘（やまむらあきひろ）
情報通信部門セキュリティ基盤グループ
プリーダー　Ph.D.
暗号理論、情報セキュリティ

石塚裕一（いしづかひろかず）
三菱電機株式会社情報技術総合研究所
主席研究員
量子情報処理、量子暗号など