SCIS 2004 The 2004 Symposium on Cryptography and Information Security
Sendai, Japan, Jan.27-30, 2004
The Institute of Electronics, Information and Communication Engineers

メンタルポーカーとコイン投げの教育用実装
Educational implementation of mental poker and fair coin flipping

金子 晃* 二瓶典子* 松川悦子† 虫鹿里佳‡
Akira Kaneko Noriko Nihei Yoshiko Matsukawa Rika Mushika

あらまし メンタルポーカーと公平なコイン投げの代表的なプロトコルの例を UNIX 上でソケット通信と Xlib を用いて C 言語および C++ 言語により可視的に実装してみた. これは, 学部の卒業研究における暗号と通信のプログラミングを通した学習の意味で, 及び, 作成したプログラムを暗号の教材として使用するという意味で, 二重の教育目的を目指したものである.

キーワード メンタルポーカー, 公平なコイン投げ, 電話でじゃんけん, ソケット通信, 教育用実装

1 序

暗号プロトコルの代表例であり, 暗号教育においても学生の興味を引きやすいメンタルポーカー (mental poker) と公平なコイン投げ (fair coin flipping) のそれぞれについて, プロトコルの説明を可視化し, かつそれを実際の通信に同期させて行えるようなプログラムの開発を, 学部の卒業研究で取り上げ学生と協力して製作してみた.

メンタルポーカーの場合は暗号化したカードを裏返しのカードで表示し, 復号と同時にカードを表にして表示する等, 実際のゲームにも使えるような可視化を試みた. またコイン投げの場合は, 最も面白く可視化できるプロトコルとして, 可換な二つの暗号を用いる方法を取り上げ. 暗号化したコインは封筒で表示し. 復号に合わせて封筒を取り外す等で, プロトコルの進展を分かりやすく示すようにした. コイン投げの日本版であるじゃんけんについても同様に実装してみた.

作成したプログラムは, 暗号の講義あるいは解説講演において実際に使用してみた. それらの体験についても報告したい.

なお, 実装は `gcc` および `g++` の `version 2.95.3` と `Xlib` の描画函数を用い, FreeBSD および Linux 上で行った. Windows への対応はまだしていないが, Cygwin の X11 エミュレーションモードでは, `sig_t` の定義を

```
typedef void (*sig_t)(int);
```

で追加すれば, 少なくともクライアントの部分は動作する.

* お茶の水女子大学〒 112-8610 文京区大塚 2-1-1 Ochanomizu University 2-1-1, Otsuka, Bunkyo-ku, Tokyo, Japan, kanenko@is.ocha.ac.jp

2 公平なコイン投げ

2.1 プロトコルの説明

公平なコイン投げプロトコルは, メンタルポーカーよりシンプルなので最初に紹介するが, 最初に発表されたのはこちらの方が少し後である [1]. しかもこの文献では. 平方剰余の理論を用い, 実際に安全性が素因数分解の困難さに帰着できる方法によるやや複雑なアルゴリズムを提案している. ここでは, 最初に学ばせるプロトコルとして, より簡単でかつ可視化に最も馴染むものとして, もともとメンタルポーカー [6] で提案された二つの可換な暗号化写像による方法を実装してみた. 使用した可換暗号は RSA 暗号と同等なものである. 以下の説明では, RSA 暗号での役割に従い, ペア鍵に対して便宜上公開鍵, 秘密鍵という言葉を用いるが, どちらもプレイ中は秘匿するので, 暗号にはなっていない. 従ってこの使い方では, 素数を法とする離散対数を用いたオリジナルなメンタルポーカープロトコルによるものと大差はないが, RSA の方がブロック長がやや短くて済む.

① アリス (親) は巨大素数 p, q を生成し, p, q および $n = pq$, $m = \mathrm{LCM}(p-1, q-1)$ をボブ (子) に通知する.

② アリスは $d_a e_a \equiv 1 \bmod m$ を満たす鍵の対 d_a, e_a をランダムに生成する. ボブも同様に d_b, e_b を生成する. 以下. $E_A(x) := x^{e_a} \bmod n$, $D_A(x) := x^{d_a} \bmod n$ 等と記す.

③ アリスは, それぞれ裏 (0), および表 (1) を表すビットにある乱数を連結して二つのメッセージ M_1, M_2 を作り, これらを自分の公開鍵で暗号化した $E_A(M_1)$, $E_A(M_2)$ をシャッフルしてボブに送る.

④ ボブは $E_A(M_i), i = 1, 2$ のどちらかを選び, 自分の公開鍵で暗号化した $E_B(E_A(M_i))$ をアリスに送る.

⑤ アリスはこれに自分の秘密鍵をかませる. 可換性により $D_A(E_B(E_A(M_i))) = E_B(D_A(E_A(M_i))) = E_B(M_i)$ を得るので, これをボブに送り返す.

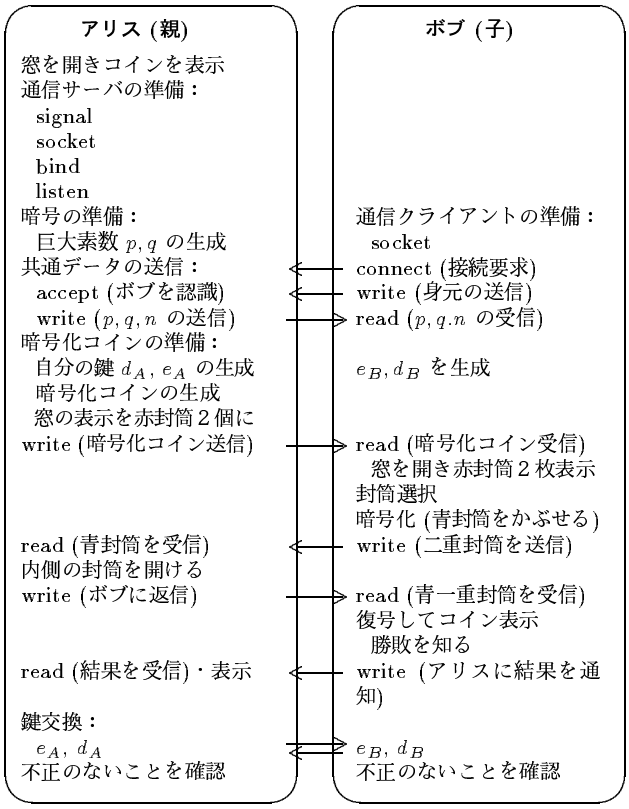
⑥ ボブは自分の秘密鍵で $D_B(E_B(M_i)) = M_i$ を得, M_i の先頭ビットを見て, 自分が選んだのが表か裏かを知る. またこの結果をアリスに送る.

⑦ 最後にアリス, ボブの双方が二つの鍵を知らせ合って, 不正のないことを確認する.

勝敗は, 例えば, ボブが表を引いたらボブの勝ち, 裏を引いたらアリスの勝ちと決めておく.

2.2 実装の流れ

以上の手順をソケット通信, および可視化と組み合わせた実装の流れを概念図により示す. 実際には一回のデータ送信にも read と write を複数組み合わせて使用する必要があるが, ここでは単純化して示している.



最後の不正確認の部分は, 実演の際には二人の画面を聴衆に同時に見せているため, 省略している. 実際に不正の確認まで解説するには, コインや封筒を相手に送ったときに, その控を画面に残すようにするべきであろう.

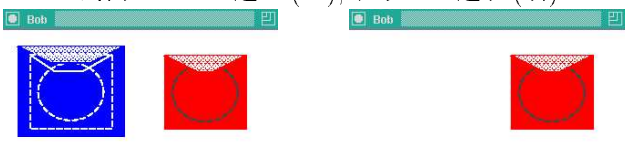
2.3 実行画面の紹介

実行画面は, コインの窓と指令入力・メッセージ出力用のターミナルウインドウの二つから成る. 純粋なゲームとしてはキャラクタベースのターミナルを使うべきではなかろうが, ここでは, 暗号や通信の学習用としての意味も含めて, メッセージだけでなく通信でやりとりしているデータ (多倍長整数を base64 エンコードしたもの) も含め, すべてを表示するようにしている.

アリスの初期画面 (p を見付けたところまで)

ボブがアリスと接続したところ

ボブが封筒の一つを選び (左), アリスに返す (右)



アリスは二重封筒を受け取り (左), 内側の封筒を破る (右)

ボブは受け取った封筒を破り, 結果を知る.

3　じゃんけん

3.1　プロトコルの説明

じゃんけんは, 普通, コイン投げと同じものとして扱われるので, 特にじゃんけんのプロトコルとして書かれたものは無いようである. そこで, 最初にコイン投げと同じくグー, チョキ, パーをアリスが暗号化してボブに送り選ばせる実装を試みたが, それはコイン投げとほとんど同じなので, ここではより実際のじゃんけんに近く, 両者が独立に自ら手を選べるビット預託を用いた実装を紹介する.

① ② 同上

③ **乱数の選択** アリスは乱数 R_A を選び, ボブに通知. ボブも乱数 R_B を選びアリスに通知.

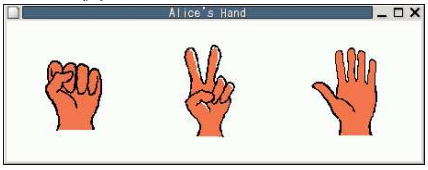
④ **手の選択** アリスは, グー (0), チョキ (1), パー (2) のどれかに自分が選んだ乱数を連結して M_A を作り, 暗号化した $E_A(M_A)$ をボブに送る.
ボブも同様に M_B を作り, $E_B(M_B)$ を相手に送る.

⑤ **鍵の開示** アリスとボブはそれぞれ相手に復号鍵 D_A, D_B を送る.

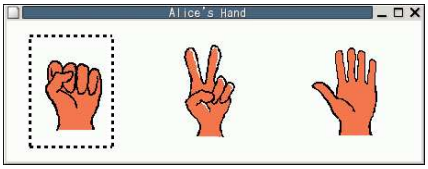
⑥ **復号** アリスとボブはそれぞれ相手の手
$D_B(E_B(M_B)) = M_B$, $D_A(E_A(M_A)) = M_A$
を知り, 自分の手と比べて勝敗を知る.
このとき, 相手から受け取った乱数も確認して不正のないことを確認する.

3.2　実行画面の紹介

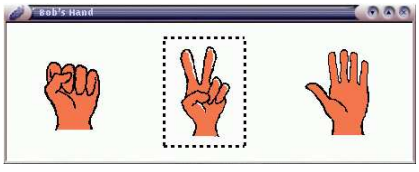
アリス側のじゃんけんの選択窓が開いたところ



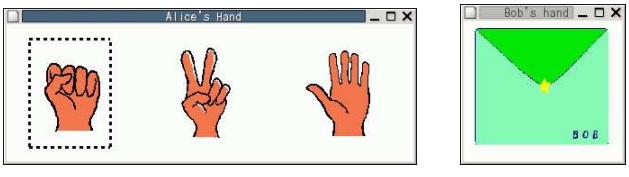
アリスがマウスでクリックして手を選んだところ



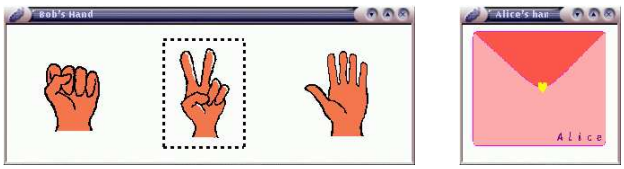
同じくボブが手を選んだところ



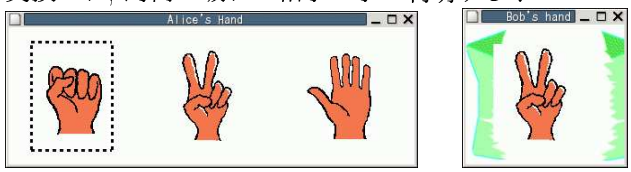
手が選ばれると, 暗号化して相手に送られる
右：アリスのところに届いたボブの封筒↓



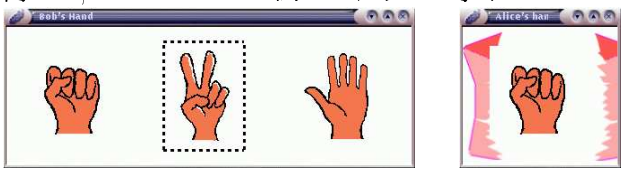
同じくボブのところに届いたアリスの封筒↓



アリスがボブの封筒をクリックすると, 瞬時に復号鍵が交換され, 封筒が破れて相手の手が判明する↓



同じく, ボブのところで開いたアリスの手↓



なお, インターネットで検索するとじゃんけんの類のサイトがいろいろ見付かるが, それらにどの程度暗号が用いられているのかは不明である. 暗号を使わないじゃんけんとしては, i モード用の Java プログラムのソースコードなども出版されている.

4　メンタルポーカー

4.1　プロトコルの説明

メンタルポーカープロトコルは [6] により導入されたもので, カードを裏向けで配る操作をネットワークを通して実現する方法をいう. オリジナルなメンタルポーカープロトコルは２人対戦用にカードを配布するだけのもので, 裸の離散対数が使われていたが, その後, 冪乗演算が平方剰余の 1 ビットを漏洩するのでポーカーゲームには支障が有るというクレームが付き, 種々の改良案が提案されて来た. 初期の歴史については [4] が詳しい. ３人以上でのプレイでは, 更に共謀などの不正を防ぐ必要があり, 実用的ではないが安全性の証明が付いたものとしてマルチパーティプロトコル [5] や零知識証明 [3] により最終的に一般化された .

ここでは, 完全な ElGamal 暗号を用いて, [7] に書かれている３人対戦用のプロトコルを実装してみた. ElGamal 暗号は暗号化によりデータ長が倍になるので, 途中でやりとりされるカードデータの対称性は崩れるが, 教科書に載っているような可換でかつ真の暗号にもなっているものを実現できる. 公開鍵を実際に公開できるので, プレイヤーの相互認証等にも使用可能である.

① アリス (親) は巨大素数 p で, $(p-1)/2$ もまた素数となるようなものを生成し, 奇数 $g < p$ をランダムに選んで p, g をボブとキャロルに通知する.

② アリス, ボブ, キャロルは, それぞれに ElGamal 暗号の秘密鍵, a, b, c および公開鍵 $e_A = g^a$, $e_B = g^b$,

$e_C = g^c$ を用意する. (以下の説明では後者は使わない.)

③ アリスは, ジョーカーも込めて 53 枚のカードデータ $M_0 \sim M_{52}$ を用意し, これに乱数を付け, ランダム置換で順序を変えた後, ElGamal 暗号で暗号化したカードの山 $(g^{k_n}, g^{ak_n}M_n)$, $n = 0, 1, \ldots, 52$ を作り, これをボブにすべて送る.

④ ボブは受け取ったカードの山から自分の手として５枚を選び, 更に ElGamal 暗号化した $(g^{k'_n}, g^{bk'_n+k_n}, g^{k''_n}, g^{bk''_n+ak_n}M_n)$ をアリスに送り返す.

④ アリスはボブから来た５枚に対し, それぞれ自分の復号鍵を用いて $g^{k''_n-ak'_n} = g^{k''_n}/(g^{k'_n})^a$, $g^{b(k''_n-ak'_n)}M_n = g^{bk''_n+ak_n}M_n/(g^{bk'_n+k_n})^a$ を計算し, 得られたペア $(g^{k''_n-ak'_n}, g^{b(k''_n-ak'_n)}M_n)$ をボブに送り返す.

⑤ ボブは通常の ElGamal 暗号の復号手続き $M_n = g^{b(k''_n-ak'_n)}M_n/(g^{k''_n-ak'_n})^b$ により自分の手を得る. また残りのカードの山をキャロルに送る.

⑥ キャロルは受け取ったカードの山から自分の手として５枚を選び, ボブと同様の方法で暗号化してアリスに送り返す.

⑦ アリスはキャロルから来た５枚についている自分の暗号を上と同様に部分復号してキャロルに送り返す.

⑧ キャロルはボブと同様に復号して自分の手を得る. また残りのカードの山から５枚を選びそのままアリスに送る.

⑨ アリスはキャロルから受け取ったカードを復号して自分の手を得る.

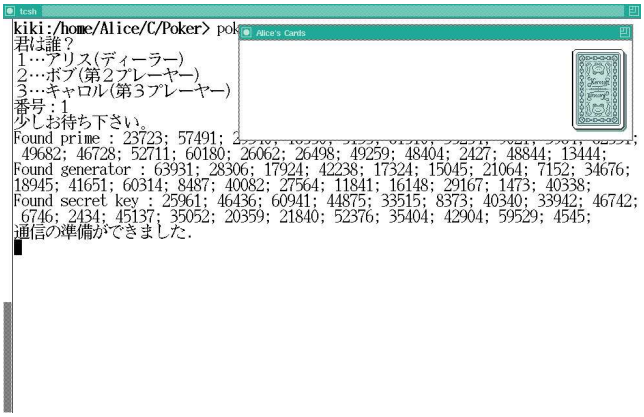
通常のメンタルポーカーと異なっている部分について, 安全性を見ておこう. ボブからアリスへの送信データは通常の ElGamal 暗号と同じものなので, これによりアリスがボブの選んだ手を想像する可能性は, 少なくとも ElGamal 暗号の強度で評価できる. また, アリスからボブへ返される情報は, アリスの公開鍵とボブの秘密鍵からすべて計算可能なので, ここでも余分な情報はもらされていない. また, キャロルがアリスとやることはボブがアリスとやることと全く同じなので, この暗号化方式は, 人数が増えても複雑さは増えない.

以上の手順をソケット通信の手順とともに図式化した表を末尾に添付した. ボブとキャロルの通信にもサーバ-クライアント型を用いているので, 今回はボブもサーバを立ち上げている. 暗号については簡単のため教科書通りの可換な暗号で表記した.

4.2 実行画面の紹介

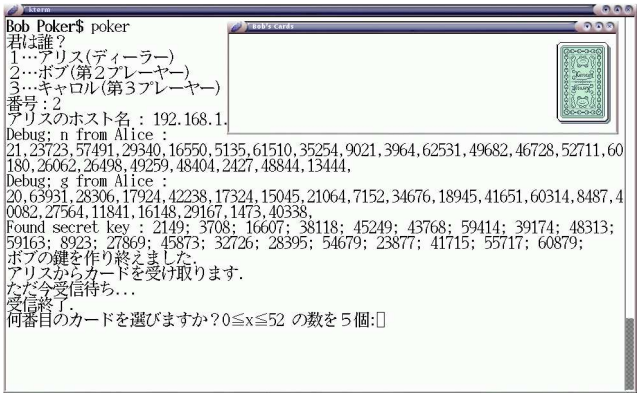
以上の実装の流れを, 実行画面のコピーによって示す:

アリスの初期画面↓



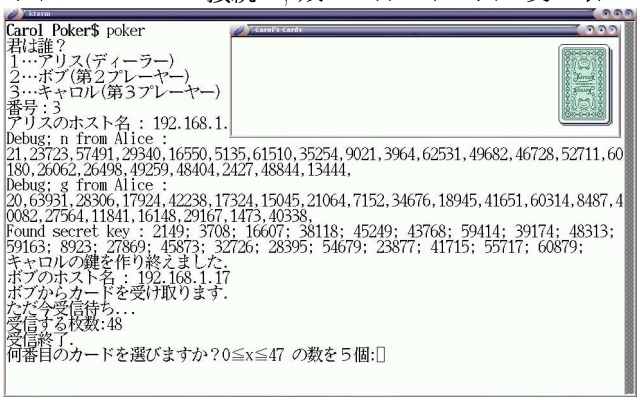
ボブがアリスと接続したところ↓

ここでキャロルもアリスと接続するとゲームが始まる.



ボブが５枚のカードを選び, 暗号化してアリスに送る. アリスが内側の暗号を復号して返すと, ボブが外側を復号してカードが現れる. 以上はほぼ瞬時に進行する.

キャロルはボブと接続し, 残ったカードの山を受け取る.



キャロルは５枚のカードを選びアリスに送って, 同様にカードを得る↓

キャロルに選んでもらった5枚をアリスは復号して自分の手を得る↓

上記実装では, 上に続いて残りのカードの山をボブに返してもう一巡させ, その後各自が手を一枚ずつ開いて勝敗を競い, 実際にゲームを楽しめるようにしてある. このプロトコルは2人のプレイの場合はそれなりの安全性が保たれているが, 3人の場合は秘密の通信路を用いた結託行為に対する対策が全くなされていないので, 大金を賭けるような実用化 (!) には向いていない.

オリジナルのメンタルポーカープロトコルは, 上に示したように最初のカードを配布し終えたところまでを指していたが, ポーカーゲームとしては, その後カードの交換や勝敗の決定までを安全に行う方法がいろいろ提案されてきた. Crépeau [3] 以降は, 結託を防ぎ, かつ戦略も秘匿した実装が可能ということになっているが, 実際にゲームを楽しむ状況下では, 各自が自分の手の裸の情報を知っているので, 最後の勝敗の付け方によっては, 秘密の通信路を用いた情報交換という結託行為の影響を完全には防ぐことができないように思われる.

上記の実装は, 300MHz 程度の Pentium II でも, 鍵の生成にやや時間がかかる程度で, 後は現実的な速度で動いた. (卒業研究の発表時には, 時間内に収まるよう, 鍵長を半分にして実演した.) Schreier[7] は, [2] を引用して, (1996 年の時点で) 零知識証明を用いたメンタルポーカーの安全な実装は Sparc Workstation で3人にカードを配り終えるのに8時間程かかるので, 当分は実際のカードでプレイすることを勧めると書いている. 我々の実装では, GUI はそのままで, プロトコルだけをこれらのより高度なものに置き換えられるので, これらより安全な実装が現在の計算機でどの程度実用的な速さで動作させられるかを確かめてみたいと思っている.

5 使用の感想と今後の課題

取り上げたプロトコルは, 学部の卒業研究として製作し, 教育の現場で使用してみた. 卒業研究のプログラミングとしては非常に適切な題材であったと思われるが, プログラムの教材としての有効性については, まだ工夫の余地がある. これらを実際に学生に見せたときの反応は, 暗号理論を教えられた後の学生と, 簡単な説明だけで見せられた学生とで, 当然ながらかなりの違いがある. 更に, 計算機に対する予備知識があまり無い学生の場合は, 実際にカードやコインの絵を通信でやりとりしていると誤解される恐れが大きいことも分かった. 高校生程度の聴衆の場合は, 離散対数や RSA 暗号等の説明を短時間で納得させるのはなかなか難しかったが, 逆に, “インターネットの世界でこんなに数学が使われていると知って驚いた”という, 期待した以上の (?) 反応が得られてしまった. 少なくとも導入部分で興味を引くという効果は有ったようである.

実際の暗号教育における内容に立ち入った教材として使用する場合は, 面白くなくても生のキャラクターデータのやりとりを直接表示するような原初的なプログラムの方が却って教育的に正しい理解を与えるかもしれないとも感じた. 以上の点を踏まえて, 今後は, 面白くて, かつ仕組みを理解させる助けにもなるような, より改良された視覚化を, コンソール画面の出力データをも工夫しつつ, より高度なプロトコルについて模索してゆきたい.

参考文献

[1] M. Blum,“Coin flipping by telephone: A protocol for solving impossible problems,” Proc. 24-th CompCon, 1982, pp.133–137.

[2] J. Edwards, “Implementing electronic poker: A practical exercise in zero-knowledge interactive proofs,” Master's thesis, Department of Computer Science, University of Kentucky, May, 1994.

[3] C. Crépeau, “A zero-knowledge poker protocol that achieves confidentiality of the players' strategy, or how to achieve an electronic poker face,” Proceedings CRYPTO 86, LN in Computer Sciences **263**, Springer, 1987, pp.239–247.

[4] S. Fortune, M. Merritt,“Poker protocols,” Proceedings CRYPTO 84, LN in Computer Sciences **196**, 1985, pp.454–464.

[5] O. Goldreich, S. Micali, A. Wigderson, “How to play any mental game,” Proc. 19-th ACM Symp. on Theory of Computing, 1987, pp.218–229.

[6] A. Shamir, R. L. Rivest, L. M. Adleman, “Mental poker,” The Mathematical Gardner, Wadsworth, 1981, pp.37–43.

[7] B. Schneier,“Applied Cryptography,” John WIley, 1996.
[8] K. Kurosawa, Y. Katayama, W. Ogata, “Reshuffiable and laziness tolerant mental card game protocol,” IEICE Trans. Fundamentals E00-A, No.1 (1997), 1-7.
[9] 二瓶典子, 松川悦子, 「ElGamal 暗号の実装とゲームへの応用」, お茶の水女子大情報科学科卒業研究, 2001.

メンタルポーカーの手順

(実線はアリスとボブまたはボブとキャロルの通信, 破線はアリスとキャロルの通信を表す.)