Online Casinos

Rolf Oppliger and Jean-Luc Nottaris

University of Berne
Institute for Computer Science and Applied Mathematics (IAM)
Neubrückstrasse 10, CH-3012 Berne, Switzerland

Abstract. Online casinos allow arbitrary users to remotely participate in mental game playing, such as e.g. Mental Poker or Mental Black Jack. Two recent developments have brought the realization of online casinos closer to reality: On the one hand, cryptographic research has come up with practical solutions and corresponding protocols to the problems of how to play games mentally and how to spend money over open networks, and on the other hand, the Internet has become a globally accessible packet-switched (inter)networking infrastructure that provides support for multicast communications. This paper overviews the basic principles of mental games and digital payment schemes, and proposes a model that can be used to set up and run online casinos within a multicast environment, such as e.g. provided by the Internet Multicast Backbone (MBone). The paper also addresses a prototype implementation that has been done at the University of Berne.

1 Introduction

If one area exists where stand-alone computer systems have been far more successful than computer networks and distributed systems, that area is games. We have seen a wide proliferation of computer games during the last decade, and many companies make a good living from selling corresponding software packages. While some of these games are really good at incorporating multimedia streams, such as audio, video, and animations, only few of them also support multi-user and multi-player capabilities. The vast majority of these games are designed to be played by just one person at a time. By contrast, games on computer networks and distributed systems are designed for multi-player use, but in doing so they tend to sacrifice most multimedia features. What they gain, however, is the possibility of global, simultaneous game playing among large numbers of players, and this possibility is required for casinos to go online.

In this paper, the term online casino is used to refer to a casino that offers its service, which is game playing, over open networks, such as e.g. the Internet, to a potentially very wide audience. Thus, an online casino allows arbitrary users to remotely participate in mental game playing, such as e.g. Mental Poker or Mental Black Jack. The primary gaming technology on the Internet today are multi-user environments in which each player must log on a particular host and maintain a TCP/IP connection to this host accordingly. Thus, n connections are required for n participants to play simultaneously, and this may not scale for large n.

Two recent developments have brought the realization of online casinos closer to reality: On the one hand, cryptographic research has come up with practical solutions and corresponding protocols to the problems of how to play games mentally and how to use money over open networks, and on the other hand, the Internet has become a globally accessible packet-switched (inter)networking infrastructure that provides support for multicast communications. The term multicast refers to a technique which allows a single datagram to be passed to selected destinations. It can thus be considered as a generalization of the broadcast and unicast techniques. A datagram is unicast, if it is delivered from one source to one destination, and it is broadcast, if it is delivered to all hosts attached to a network. IP multicast is an extension of local area networking multicast to TCP/IP networks [5, 4]. It provides support for the transmission of an IP datagram to a host group which is a set of hosts identified by a single class D IP destination address [6]. A multicast datagram is delivered to all members of its destination host group with the same best efforts quality of service as regular unicast IP datagrams, i.e., the datagram is not guaranteed to arrive intact at all members of the host group or in the same order relative to other datagrams. The membership of a host group is dynamic, and hosts can join and leave groups at any time. There is no restriction on the location or number of members in a host group, and a host may be a member of more than just one host group at a time. Also, a host need not be a member of a group to send datagrams to it. A host group may be permanent or transient.

- A permanent host group has a well-known, administratively assigned class D IP address. Note that it is the address, not the membership of the host group, that is permanent. At any time a permanent host group may have any number of members, even zero.
- Those class D IP multicast addresses that are not reserved for permanent groups are available for dynamic assignment to transient host groups. These groups exist only as long as they have members.

Internet forwarding of IP multicast datagrams is handled by multicast routers (mrouters), which may be co-resident with, or separate from, normal Internet routers. A host transmits an IP multicast datagram as a local network multicast that reaches all immediately-neighboring members of the destination host group. If the datagram has an IP time-to-live (TTL) value that is greater than 1, the mrouter(s) attached to the local network take responsibility for forwarding it towards all other networks that have members of the destination group. On those other member networks that are reachable within the TTL, an attached mrouter completes delivery by transmitting the datagram as a local multicast. In order to participate in IP multicast, a host must be able to send and receive multicast datagrams. In IPv4, a host communicates its group membership by using the Internet Group Management Protcol (IGMP). In IPv6 (or IPng respectively), the IGMP is incorporated into the Internet Control Management Protcol (ICMP) [10, 11].

The Multicast Backbone (MBone) is an experimental network that is overlaid on the existing IPv4-based Internet to carry IP multicast datagrams [14, 7]. It originated from experiments during IETF meetings in which live audio and video were transmitted around the world. Application tools, such as `vat` (visual audio tool), `nevot` (network voice terminal), `nv` (network video), `ivs` (inria videoconferencing system), `vic` (video conferencing) and `wb` (whiteboard) are publicly available today. They can be used to hold multimedia conference sessions over the MBone. In addition to that, session directory tools, such as `sd` or `sdr`, can be used to announce MBone conference sessions in public, and to have MBone application tools start up according to the session parameters.

This paper overviews the basic principles of mental games and digital payment schemes, and proposes a model that can be used to set up and run online casinos within a multicast environment, such as e.g. the MBone. The paper also addresses a prototype implementation that has been done at the Institute for Computer Science and Applied Mathematics (IAM) of the University of Berne. The rest of the paper is organized as follows: The terminology that is used to subsequently describe the cryptographic protocols is summarized in section 2. The basic principles of mental games and digital payment schemes are overviewed in sections 3 and 4, and a model for online casinos is proposed in section 5. A prototype implementation is addressed in section 6, and conclusions are drawn in section 7.

2 Terminology

A protocol specifies the format and relative timing of messages exchanged between communicating entities. A cryptographic protocol is a protocol that uses cryptography, meaning that all or parts of the messages are encrypted on the sender' side and decrypted on the receiver' side [17]. The following notation is used to describe cryptographic protocols:

- The term K is used to refer to a secret key, which is a key taken from a secret key cryptosystem, whereas the term (k, k^{-1}) is used to refer to a public key pair, which is a key pair taken from a public key cryptosystem. In either case, key subscripts may be used to refer to particular entities.
- The term $\{m\}K$ is used to refer to a message m that is encrypted with the secret key K. The same key is used for decryption; thus, $\{\{m\}K\}K$ equals m.
- Similarly, the term $\{m\}k$ is used to refer to a message m that is encrypted with the public key k. The message can be decrypted only with the corresponding private key k^{-1}; thus, $\{\{m\}k\}k^{-1}$ equals m.
- In a digital signature scheme, a user's private key is used to digitally sign messages, and the corresponding public key is used to verify the signatures. Referring to the OSI terminology, the term $\{m\}k^{-1}$ is used to denote a digital signature giving message recovery, whereas the term $\langle m\rangle k^{-1}$ is used to denote a digital signature with appendix [12]. Note that in the second case, $\langle m\rangle k^{-1}$ in fact abbreviates $m, \{h(m)\}k^{-1}$, with h referring to a collision-resistant one-way hash function.

– In accordance to international standardization, the term $X \ll Y \gg$ is used to refer to a certificate that has been issued by X for Y's long-term public key k_Y. This certificate may conform to any standard in use today, such as e.g. ITU-T X.509 [13].

In addition to that, π_n is used to refer to a permutation of n numbers, L_n to a list of n random numbers, and square brackets $[]^+$ to an iteration of one or more times.

3 Mental Games

Mental game playing is an application that has attracted many cryptographers in the past. The rules for a mental game are just like its regular counterpart, except that players may communicate over a public and open network, such as e.g. the public switched telephone network (PSTN), the integrated services digital network (ISDN), or even the Internet. There are many games that can be played mentally. They range from rather simple games, such as e.g. coin flipping, to more sophisticated games, such as e.g. Mental Poker.

Playing Mental Poker is actually a very difficult problem for several reasons. The foremost reason is that it is impossible from an information-theoretic point of view. This result is due to Shamir, Rivest, and Adleman [18]. The same authors, however, have also proposed a protocol for playing mental poker that relies on the difficulty of inverting certain cryptographic transformations. The protocol was shown to be insecure by Lipton and Coppersmith. Lipton found a way to determine one bit about the messages using the fact that exponentiation modulo n preserves quadratic residuosity. This leakage of partial information was fixed by Goldwasser and Micali in the two-player case, using probabilistic encryption [9]. Unfortunately, their scheme does not extend to a larger number of players. Several protocols that have been proposed so far address the multi-player problem instance. Unfortunately, they all make special assumptions, such as the players' inability to establish secret communications [19] or the existence of a trusted third party [8]. In 1985, Crépeau came up with a protocol for playing Mental Poker that minimizes the effect of player coalitions [2], and one year later, he even presented an enhanced version of the protocol that does not require that players reveal their cards at the end of the game to show that they didn't cheat. This protocol allows players to keep secret their strategy, and to keep up a poker face accordingly [3]. In 1987, Goldreich, Micali, and Wigderson proposed a generalization of this protocol. In fact, they came up with a polynomial-time algorithm that, given as input the description of a mental game with incomplete information and any number of players, produces a protocol for playing the game that leaks no partial information, provided the majority of the players is honest.

This paper does not further elaborate on the theory of mental game playing. Instead, it focuses on Mental Black Jack (also known as "21") as an exemplary (and sufficiently simple) game that can be used to set up and run an online casino within a multicast environment. Black Jack is at present considered as one of the world's most popular games played with cards. The aim of Black Jack, which is played against a bank, is to acquire a total card value as near as possible to 21, without exceeding it. To determine his value, the player must remember that cards from two to nine take their pip or number value. Tens, Jacks, Queens, and Kings count ten each. Aces, at the player's option, count 1 or 11. At start, the banker deals two cards and the player may stand, refusing other cards, or hit, asking for one or more cards to improve his count. If he goes over 21, he busts and loses his bets. If he decides to stand, the banker gives himself one or more cards following the rules of the game. However, the banker must give himself a card if his count totals 16 or less, while he must stand if his count totals 17 or more. If a player's first two cards are a ten or a court card and an ace, he has a Black Jack. He has "21" when a count of 21 is obtained with three or more cards. When the banker has a Black Jack, all players lose their bets except the ones also having a Black Jack, as the bet is called off. In this case, no one loses or wins. The bet can be left for the next deal or can be withdrawn. A Black Jack is paid 3 to 2, except in the event of tie (same count) with the banker. If nobody has a Black Jack, the count which is nearest to 21 wins. If the count is lower than the banker's, bets are lost. If there is a tie, the bet is called off. If 21 is exceeded, the banker wins. If the banker busts, bets win.

There are some variations of the game making it possible to increase the initial bet and hence the winnings, when the initial cards have already been dealt. These variations refer to double-down, split Pair, and insurance. Note that the model proposed in section 5 is designed to handle Mental Black Jack as an exemplary game that can be played mentally within an online casino. The same model is (at least conceptually) able to handle any cryptographic protocol that one may think of and come up with. The variations mentioned above are therefore not further addressed in this paper.

4 Digital Payment Schemes

A digital payment scheme addresses the problem of how to securely spend money over a public and inherently open network, such as e.g. the Internet, and how to securely process financial transactions accordingly. The availability of digital payment schemes is a crucial prerequisite for online casinos to be used and widely deployed. There are many digital payment schemes available today. They can be divided into two major categories:

- Payment schemes that follow the credit card model, where a customer uses his credit card to buy and postpay goods and services. The most important example of this category is the Secure Electronic Transaction (SET) protocol as specified by the major credit card companies and software vendors, including Visa International, MasterCard, and Microsoft.
- Payment schemes that follow the debit card model, where a customer buys and prepays tickets that he can later use to obtain goods and services. In the simplest case, the tickets refer to digital cash. There are three parties

involved in a digital cash system: customers, merchants, and a bank. Every customer and every merchant is assumed to have an account with the bank. The customer withdraws digital cash from his account and spends it with a merchant. The merchant, in turn, deposits the cash he receives from the customer on his account. Digital cash protocols are protocols for the interactions between customers, merchants, and a bank. During the last decade, various digital cash protocols have been proposed to be used either in online or offline systems:

- Online systems require the merchant to communicate with the bank at every sale, whereas
- offline systems require no communication between the merchant and the bank until after the transaction between the merchant and the customer.

From a security point of view, online systems are advantageous. If there is a problem, the bank simply doesn't accept the cash and the customer cannot cheat. Offline systems can't prevent a customer from cheating, but instead detect cheating. Another way is to create a special smart card containing a tamperproof chip (which is often called an observer). The observer keeps a database of all the pieces of digital cash spent by the smart card. If a customer attempts to copy some digital cash and spend it twice, the observer detects the attempt and does not allow the transaction. Since the chip is tamperproof, the customer can't erase the database without permanently damaging the smart card. Digital cash protocols can also be divided along another line. Electronic coins have a fixed value; people using this system need several coins in different denominations. Electronic checks can be used for any amount up to a maximum value and then returned for a refund of the unspent portion. Another distinction is made whether a digital cash protocol preserves the customers' anonymity. The cryptographic techniques that can be used to provide anonymous digital cash are blind digital signatures [1].

Both categories provide suitable payment schemes for online casinos. Payment schemes that follow the debit card model are appropriate for games with small stakes, whereas payment schemes that follow the credit card model better fit the requirements of games with larger stakes. With special regard to online casinos, micropayment schemes provide a viable alternative. Micropayments schemes are a special kind of payment schemes for applications in which each payment is sufficiently small. To support micropayments, efficiency is required; otherwise, the cost of the mechanism will exceed the value of the payments. The area of micropayments has attracted considerable attention recently, and several researchers have proposed corresponding schemes.

It is assumed that there will be many digital payment schemes in the future, and that there must be a negotiation layer on top of the corresponding protocols. This negotiation layer and its protocols are being defined by the Joint Electronic Payment Initiative (JEPI) jointly announced by the CommerceNet and World Wide Web Consortium (W3C). The model proposed next does not require a particular payment scheme, but assumes different payment schemes to coexist simultaneously . Thus, it is up to the user to specify a particular payment scheme (that is supported by the online casino), and to buy and sell chips accordingly.

5 Model

Having introduced the basic principles of mental games and digital payment schemes, this section works towards a synthesis of both in a model for online casinos. The processes involved and the cryptographic protocols are overviewed in the following subsections.

5.1 Process Overview

It is assumed that online casino $\mathcal{C}$ is managed by a casino manager process C, and that C creates and stops table processes $T_i (i = 1, \ldots, n)$ according to user demand. Referring to the real world, C represents the casino manager, whereas each T_i represents the dealer running table i. T_i not only represents the dealer of table i, but the banker (or cashier) as well. In this model, every table has a banker of its own, and payments are considered locally, meaning that players buy and sell chips from the dealer of the table they play at. This approach has been chosen for simplicity, and there is no underlying reason, why a cashier would not be able to serve an entire casino.

In addition to that, it is assumed that C has a public key pair (k_C, k_C^{-1}) of which k_C is publicly available and generally trusted, and each T_i has a public key pair $(k_{T_i}, k_{T_i}^{-1})$ of which k_{T_i} is publicly available but not generally trusted. It is up to C to certify k_{T_i}, and to provide T_i with a corresponding certificate $C \ll T_i \gg$.

Each T_i has chips of size $s_{i1}, \ldots, s_{il_i}$. For every chip size $s_{ir} (r = 1, \ldots, l_i)$ he holds a public key pair $(k_{s_{ir}}, k_{s_{ir}}^{-1})$ of which $k_{s_{ir}}$ is publicly available, certified, and distributed by T_i, whereas $k_{s_{ir}}^{-1}$ is kept private and is used only to sell and buy chips. To keep the protocols sufficiently simple, it is assumed that each T_i has 4 decks of 13 cards each, resulting in a total number of $n = 52$ cards.

Figure 1 illustrates the model for an online casino. It shows the casino manager and table processes, the communication channels between them, as well as the corresponding message flows. With regard to the communication channels, it is assumed that C is assigned a permanent IP class D address, which is denoted as Adr_C, and each T_i is assigned a transient IP class D addresses from a pool. Consequently, there is a multicast channel assigned to C and each T_i. In addition to that, there are several unicast channels between players and table processes.

5.2 Cryptographic Protocols

In principle, a user can enter the casino, join a table, play games at that table, quit it, and leave the casino. The corresponding protocols are overviewed next.

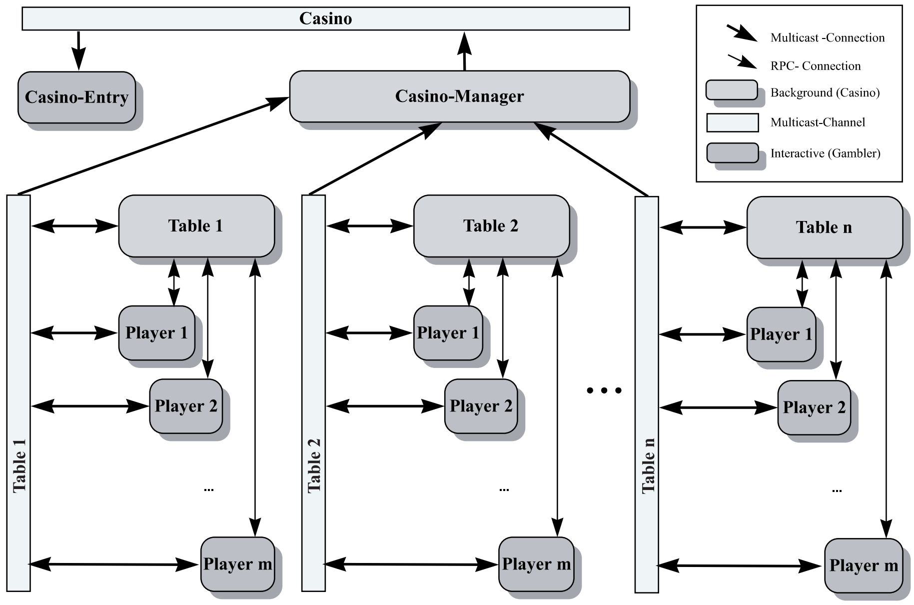

Fig. 1. Model for an online casino

Enter the Casino If user U wants to enter the casino, he has to grab the information that C periodically multicasts to Adr_C. This information includes a description of each table T_i ($i = 1, \ldots, n$) that is up and running, and that U can join and play at. U (or his user agent respectively) can interpret the information locally and render a graphical representation accordingly.

Join a Table If user U wants to join table T_i, he has to randomly select a public key pair (k, k^{-1}) and run the following join table protocol:

$$1 : U \longrightarrow Adr_{T_i} : \text{JOIN\_TABLE\_REQUEST}(U, k)$$
$$2 : T_i \longrightarrow Adr_{T_i} : \text{JOIN\_TABLE\_CONFIRMATION}(\langle U, j, T_i \ll P_{ij} \gg \rangle k_{T_i}^{-1})$$

In step 1, U multicasts a join table request message to Adr_{T_i}. The message includes U and k. T_i grabs the message and decides whether he's going to serve U at table i. If he's going to serve U, he selects a track number $1 \leq j \leq m$ for U, and returns a corresponding join table confirmation message to Adr_{T_i} in step 2. In this case, the message includes U, j, and a certificate $T_i \ll P_{ij} \gg$ for U's public key k. The message is digitally signed by T_i with its private key $k_{T_i}^{-1}$. If T_i is not going to serve U at table i, he simply returns a value for j that is equal to zero.

If $j > 0$, U becomes a legitimate player on track $1 \leq j \leq m$ of table i. In this case, P_{ij} is used to refer to this user, and $(k_{P_{ij}}, k_{P_{ij}}^{-1})$ is used to refer to P_{ij}'s public key pair (formerly denoted as (k, k^{-1})). P_{ij} uses this private key $k_{P_{ij}}^{-1}$ do digitally sign messages that he subsequently sends out.

So far, P_{ij} has only received permission to play at table i. In order to participate, he still has to buy some chips from T_i. It has already been mentioned previously that chips are considered locally, and that they are sold and bought by T_i. Thus, P_{ij} has to use the following protocol to buy a chip of size s_{ir}:

$$\begin{aligned}1: P_{ij} &\longrightarrow T_i \;: \text{BUY\_CHIP\_REQUEST}(\langle s_{ir}, k, params\rangle k_{P_{ij}}^{-1})\\ 2: T_i &\longrightarrow P_{ij} : \text{BUY\_CHIPS\_CONFIRMATION}(\langle \{k\}k_{s_{ir}}^{-1}\rangle k_{T_i}^{-1})\end{aligned}$$

The buy chip protocol can be used by any player at any time. Note that its communication flows are not multicast. Instead, P_{ij} sends a buy chip request message to T_i in step 1, and T_i returns a buy chip confirmation message to P_{ij} in step 2. The buy chip request message includes the chip size s_{ir}, a randomly chosen public key k for this chip, as well as some payment parameters $params$ that are used to handle the financial transaction. The entire message is digitally signed with P_{ij}'s private key, which is $k_{P_{ij}}^{-1}$. The buy chip confirmation message, in turn, includes k digitally signed with $k_{s_{ir}}^{-1}$. Again, the whole message is digitally signed with its originator's private key, which is now $k_{T_i}^{-1}$.

If P_{ij} wants to buy several chips, he (or his user agent respectively) has to iteratively initiate and rerun the buy chip protocol. It is also possible to modify the protocol to support buying several chips in a single protocol run. This possibility, however, does not extend the overall functionality and is therefore omitted in this paper.

Play Games Having joined T_i and bought some chips, $P_{ij} (i = 1, \ldots, n; j = 1, \ldots, m)$ can play multiple games at table i. For every single game g $(g = 1, \ldots)$, T_i has to shuffle the cards, have the players bet, deal the cards, play the game, and pay off. The corresponding protocols are overviewed next.

To shuffle the cards, T_i has to randomly select a permutation $\pi_n^{(g)}$ of n cards, and each player P_{ij} has to randomly select a list of n numbers $L_n^{(ijg)}$ that he has to keep secret. They all commit to their selection by multicasting a one-way hash value of $\pi_n^{(g)}$ and $L_n^{(ijg)}$ to Adr_{T_i}. Thus, T_i multicasts a shuffle card message to Adr_{T_i} that looks like

$$T_i \longrightarrow Adr_{T_i} : \text{SHUFFLE\_CARDS}(\langle T_i, g, h(\pi_n^{(g)})\rangle k_{T_i}^{-1})$$

and each player P_{ij} multicasts a similar message to Adr_{T_i}

$$P_{ij} \longrightarrow Adr_{T_i} : \text{SHUFFLE\_CARDS}(\langle P_{ij}, g, h(L_n^{(ijg)})\rangle k_{P_{ij}}^{-1})$$

Note that the messages are digitally signed by their originators. Before dealing the cards, each player must place his bets. In a real casino, this step refers to the step in which each player puts some chips on the table. In principle, each chip represents a bet. In an online casino, the corresponding protocol can be summarized as follows:

$$
\begin{array}{lll}
1: T_i & \longrightarrow Adr_{T_i} & : \text{BET\_START}(\langle T_i, g, start\rangle k_{T_i}^{-1}) \\
2: P_{ij} & \longrightarrow Adr_{T_i} & : \text{BET\_REQUEST}(\langle P_{ij}, T_i, g, s_{ir}, \{k\}k_{s_{ir}}^{-1}\rangle k_{P_{ij}}^{-1}) \\
3: T_i & \longrightarrow P_{ij} & : \text{BET\_CHALLENGE}(\langle T_i, g, c, N\rangle k_{T_i}^{-1}) \\
4: P_{ij} & \longrightarrow T_i & : \text{BET\_RESPONSE}(\langle T_i, g, c, N\rangle k^{-1}) \\
5: T_i & \longrightarrow Adr_{T_i} & : \text{BET\_STOP}(\langle T_i, g, stop\rangle k_{T_i}^{-1}) \\
6: T_i & \longrightarrow Adr_{T_i} & : \text{BET\_CONFIRMATION}(\langle T_i, g, [(P_{ir}, V_{ir})]^{+}\rangle k_{T_i}^{-1})
\end{array}
$$

In step 1, T_i initializes the protcol run by multicasting a bet start message to Adr_{T_i}. The message includes T_i and g. It is digitally signed with T_i's private key $k_{T_i}^{-1}$. It is now up to each P_{ij} to place his bets. If P_{ij} wants to place a bet, he multicasts a bet request message to Adr_{T_i} in step 2. The message includes P_{ij}, T_i, g, s_{ir}, and $\{k\}k_{s_{ir}}^{-1}$. Again, it is digitally signed with the originator's private key, which is now $k_{P_{ij}}^{-1}$. Note that $\{k\}k_{s_{ir}}^{-1}$ represents a chip of size s_{ir}. To prove legitimate ownership of the chip, P_{ij} has to run a challenge-response handshake with T_i. In step 3, T_i challenges P_{ij} with a bet challenge message that includes a bet counter c and a nonce N. The bet counter is incremented by one for every single bet of P_{ij} in game g, whereas the nonce is randomly chosen by T_i. In step 4, P_{ij} has to provide T_i with a correct response which corresponds to the same message digitally signed with the private key k^{-1} that belongs to the chip's public key k. Protocol steps 2 to 4 can be iterated an arbitrary number of times. After a certain amount of time, T_i multicasts a bet stop message in step 5 and a bet confirmation message in step 6. The first message indicates that bets are going to be accepted only for a certain amount of time, whereas the second message indicates the fact that time has run out and bets are no longer accepted. Both messages are digitally signed by T_i with his private key $k_{T_i}^{-1}$. The bet confirmation message also carries information about the players P_{ir} and their total bet values $V_{ir} (r = 1, \ldots, m)$. Again, the message is digitally signed by T_i.

To deal card d ($d = 1, \ldots, n$) and provide P_{ij} with a card, T_i has to run the following protocol:

$$
\begin{array}{lll}
1: T_i & \longrightarrow Adr_{T_i} & : \text{DEAL\_CARD\_REQUEST}(\langle T_i, g, d\rangle k_{T_i}^{-1}) \\
2: P_{ir} & \longrightarrow Adr_{T_i} & : \text{DEAL\_CARD\_SHIFT}(\langle P_{ir}, g, d, L_n^{(irg)}[d]\rangle k_{P_{ir}}^{-1}) \\
3: T_i & \longrightarrow Adr_{T_i} & : \text{DEAL\_CARD\_SHOW}(\langle T_i, g, d, P_{ij}, \\
\quad : & & : \pi_n^{(g)}[\sum_r L_n^{(irg)}[d](mod\ n - d)]\rangle k_{T_i}^{-1})
\end{array}
$$

In step 1, T_i multicasts a deal card request message to Adr_{T_i}. The message includes T_i, g, and d. It is digitally signed by T_i with $k_{T_i}^{-1}$. It is now up to each player $P_{ir} (r = 1, \ldots, m)$ to multicast a deal card shift message to Adr_{T_i}. This message includes P_{ir}, g, d, and $L_n^{(irg)}[d]$, with $L_n^{(irg)}[d]$ referring to element d in P_{ir}'s random number list $L_n^{(irg)}$. Again, the message is digitally signed with the originator's private key, which is $k_{P_{ir}}^{-1}$. After all players have sent out their messages in step 2, T_i multicasts a deal card show message to Adr_{T_i} in step 3. This message reveals card d to P_{ij}. It includes T_i, g, d, P_{ij}, and $\pi_n^{(g)}[\sum_r L_n^{(irg)}[d](mod\ n - d)]$. The whole message is digitally signed with $k_{T_i}^{-1}$. It is important that T_i drops card d from $\pi_n^{(g)}$ after the protocol run. Thus, the list becomes shorter. Multiple cards can be dealt by running the deal card protocol iteratively.

To play game k, T_i has to run a handshake protocol with each player P_{ij}. The protocol consists of two steps that can be summarized as follows:

$$\begin{array}{l} 1: T_i \longrightarrow Adr_{T_i} : \text{PLAY\_REQUEST}(\langle T_i, g, j, N\rangle k_{T_i}^{-1}) \\ 2: P_{ij} \longrightarrow Adr_{T_i} : \text{PLAY\_CONFIRMATION}(\langle P_{ij}, g, N, hit \mid stand\rangle k_{P_{ij}}^{-1}) \end{array}$$

In step 1, T_i multicasts a play request message to Adr_{T_i}. The message includes T_i, g, j, and a nonce N. It addresses player j and is digitally signed by T_i with $k_{T_i}^{-1}$. It is now up to P_{ij} to grab the message and return a play confirmation message to Adr_{T_i} in step 2. The message has to include P_{ij}, g, N, and either *hit* or *stand*. It is digitally signed by P_{ij} with $k_{P_{ij}}^{-1}$. If P_{ij} returns *hit*, the deal card protocol is relaunched. If the sum of the cards exceeds 21, P_{ij} loses and T_i addresses the next player. If P_{ij} returns *stand*, T_i automatically addresses the next player.

After the game, T_j has to pay off the players. The corresponding protocol is quite similar to the buy chip protocol and can be summarized as follows:

$$\begin{array}{l} 1: T_i \longrightarrow Adr_{T_i} : \text{PAY\_OFF\_ANNOUNCE}(\langle T_i, g, [P_{ij}, W_{ij}]^+\rangle k_{T_i}^{-1}) \\ 2: P_{ij} \longrightarrow T_i \quad : \text{PAY\_OFF\_REQUEST}(\langle g, (k', k'', \ldots)\rangle k_{P_{ij}}^{-1}) \\ 3: T_i \longrightarrow P_{ij} \quad : \text{PAY\_OFF\_CONFIRMATION}(\langle g, (\{k'\}k_{s_{ir}}^{-1}, \{k'\}k_{s_{ir}}^{-1}, \ldots)\rangle k_{T_i}^{-1}) \end{array}$$

In step 1, T_i multicasts a pay off announce message to Adr_{T_i}. The message includes T_i, g, and a list of pairs that contain both P_{ij} and P_{ij}'s wins W_{ij}. The message is digitally signed by T_i with $k_{T_i}^{-1}$. In step 2, it is up to each winning player P_{ij} to send a pay off request message to T_i. The message includes g and a set of public keys $(k', k'', \ldots)$. It is digitally signed by P_{ij} with $k_{P_{ij}}^{-1}$. Finally, T_i returns a pay off confirmation message to each winning player P_{ij} in step 3. Each message includes g and some public keys that are digitally signed by T_i with the private keys that belong to the appropriate chip sizes. Note that messages 2 and 3 are not multicast but addressed directly to T_i or P_{ij} respectively. Also note that at the end of the game, T_i has to publish his initial permutation $\pi_n^{(g)}$, and each player P_{ij} has to publish his list of random numbers $L_n^{(ijg)}$. If they all hash to the values previously announced, everybody has played honestly.

Quit a Table If P_{ij} wants to quit table i, he has to sell his remaining chips to T_i. Again, the corresponding sell chip protocol is similar to the buy chip and pay off protocols. It is thus omitted in this paper.

Leave the Casino Having sold all his chips to T_i, P_{ij} is free to leave the casino without doing any further action. He simply disconnects from the network and becomes user U again.

6 Prototype Implementation

With the proliferation of network and Internet applications, interpreted programming and scripting languages have become popular again. Examples are

Java and Tcl (Tool Command Language). While Java is a full-featured interpreted programming language that gives programmers access to low-level details, such as threaded processes, Tcl is a significantly simpler language that can act as a high-level scripting tool e.g. for linking Java applets. It is possible and very likely that Java will be the programming language of choice for Internet tool programmers, and that Tcl will be the high-level language that novice and sophisticated users alike will use to knit these tools together. Tcl complements Java, and many companies, including Sun Microsystems, actively support both Tcl/Tk and Java. Tcl usually comes with a user-interface toolkit, called Tk, that displays a consistent interface on any platform. This makes Tcl/Tk an ideal candidate for multiplatform development [16]. Tcl-DP is a distributed programming extension to Tcl/Tk that is freely available and well suited for writing client/server applications using sockets. Tcl-DP provides UDP and TCP/IP connection management, remote procedure call (RPC), and a simple distributed object system. As with Tcl, the goal of Tcl-DP is ease of programming for applications, not maximal performance. This makes Tcl/Tk and Tcl-DP an ideal prototyping environment for Internet and MBone applications.

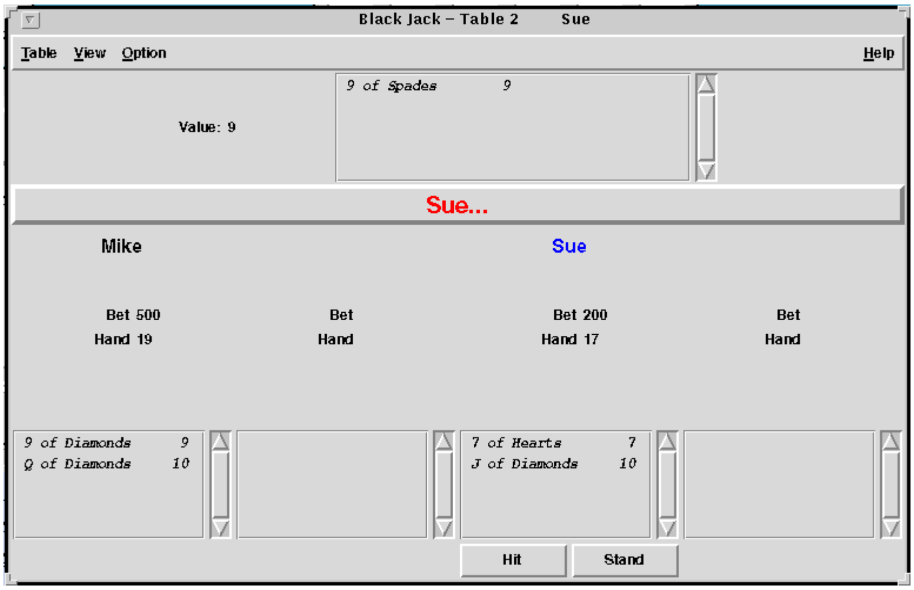

Fig. 2. Graphical user interface for an exemplary table

The model proposed in the previous section has been prototyped at the Institute for Computer Science and Applied Mathematics (IAM) of the University of Berne. Figure 2 shows the graphical user interface after having joined an exemplary table. In a first step, Tcl/Tk and Tcl-DP are used as development environments. In addition to that, SecuDE from GMD is used for cryptographic transformations. Most protocols are implemented in a multicast fashion. For example, the join table protocol consists of two messages that are both multicast. Some protocols, however, combine multicast messages with unicast. In this case, unicast messages are implemented with Tcl-DP's RPC facility. For example, the buy chip request protocol consist of a handshake that is implemented with RPC. In the current implementation of the deal cards protocol, both approaches are being combined, meaning that all messages are multicast, except for the handshake between the dealer T_i and each player P_{ij} which is unicast and implemented with an RPC.

In October 1996, the prototype implementation was released for public scrutiny [15]. The implementation consists of two software modules; one to set up and run a casino (which is the server module), and one to participate as a player (which is the client module). One thing that has not been implemented so far, however, is the use of a digital payment scheme in general, and the support of the outcome of JEPI in particular. Further information about online casinos in general, and the prototype implementation in particular can be found on the World Wide Web (WWW) by following the URL `http://iamwww.unibe.ch/~oppliger/Research/casinos.html`.

7 Conclusions

This paper has overviewed the basic principles of mental games and digital payment schemes, and has proposed a model that can be used to set up and run online casinos within a multicast environment, such as e.g. the Internet Multicast Backbone (MBone). The model can be used to incorporate other games, such as e.g. Mental Poker, American Roulette, Boule, Craps, or Punto Banco. The principles remain the same, although the corresponding protocols may gain in complexity. The paper has also addressed a prototype implementation that has been done at the University of Berne. Early experiments with the prototype implementation have shown that the MBone can serve as a communications infrastructure for online casinos. It is not clear, however, how the prototype behaves if players are widely distributed, and if the network becomes lossy and starts dropping IP packets. With regard to lossy networks, it is assumed that a more realistic approach would be to use a reliable multicast transport service, such as e.g. provided by the Reliable Multicast Protocol (RMP). RMP provides a reliable, ordered, and fault-tolerant transport service from multiple senders to multiple destinations that suits the requirements of online casinos. Another modification that seems to be appropriate and important for online casinos to become widely deployed is the provision of practical solutions to the problem of client module software distribution. It should not be necessary for a user to first download a software module that he can then use for participating in mental game playing. Instead, he should be able to simply click on a button and the system should be able to automatically download and install the module on his local machine. Java applets and Java-aware browsers, such as HotJava from Sun Microsystems or Netscape's Navigator, may provide the technology required to address these issues.

In addition to the technical issues discussed in this paper, there are also legal issues to address when it comes to a wide deployment of online casinos within the MBone. For example, there are states that allow gambling, and others that don't. Obviously, the states that allow gambling make a good living from the taxes they earn. The situation is even more obscure, as there are states that allow gambling only if the players are physically living a certain distance away from the casino they play at. Policies like that are difficult to maintain with regard to online casinos. What does it mean if an online casino is accessible from anywhere? Wouldn't this casino attract the attention of citizens that are coincidentally living in states that prohibit gambling? It is assumed that the proliferation of online casinos will also lead to a world wide harmonization of legislation for gambling.

Acknowledgements

The authors would like to thank Dieter Hogrefe and Hansjürg Mey for their encouragement and support.

References

1. Chaum, D. Achieving Electronic Privacy. *Scientific American* (August 1992), 96 – 101.
2. Crépeau, C. A Secure Poker Protocol that Minimizes the Effect of Player Coalitions. In *Advances in Cryptology — CRYPTO '85* (1986), H. Williams, Ed., Springer-Verlag, pp. 73 – 86.
3. Crépeau, C. A zero-knowledge protocol that achieves confidentiality of the players' strategy or how to achieve an electronic poker face. In *Advances in Cryptology — CRYPTO '86* (1987), A. Odlyzko, Ed., Springer-Verlag, pp. 239 – 247.
4. Deering, S. Host Extensions for IP Multicast. Request for Comments 1112, August 1989.
5. Deering, S., and Cheriton, D. Host Groups: A Multicast Extension to the Internet Protocol. Request for Comments 966, December 1985.
6. Deering, S., and Cheriton, D. Multicast Routing in Datagram Internetworks and Extended LANs. *ACM Transactions on Computer Systems 8*, 2 (1990), 85 – 110.
7. Eriksson, H. MBONE: The Multicast Backbone. *Communications of the ACM 37*, 8 (1994), 54 – 60.
8. Fortune, S., and Merritt, M. Poker Protocols. In *Advances in Cryptology — CRYPTO '84* (1985), G. Blakley and D. Chaum, Eds., Springer-Verlag, pp. 454 – 464.
9. Goldwasser, S., and Micali, S. Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In *Proceedings of ACM Symposium on Theory of Computing* (New York, 1982), ACM Press, pp. 365 – 377.

10. Hinden, R. IP Next Generation Overview. *Communications of the ACM 39*, 6 (1996), 61 – 71.
11. Huitema, C. *IPv6: The New Internet Protocol.* Prentice-Hall, New Jersey, 1996.
12. ISO/IEC. Information Processing Systems — Open Systems Interconnection Reference Model — Part 2: Security Architecture. ISO/IEC 7498-2, 1989.
13. ITU. The Directory — Authentication Framework. Recommendation X.509, Geneva, Switzerland, November 1987.
14. Macedonia, M., and Brutzman, D. MBone Provides Audio and Video Across the Internet. *IEEE Computer 27*, 4 (April 1994), 30 – 36.
15. Nottaris, J. Prototypimplementierung eines virtuellen Casinos in einer Multicast-Umgebung. Diplomarbeit, Universität Bern, 1996.
16. Ousterhout, J. *Tcl and the Tk Toolkit.* Addison-Wesley Publishing Company, Inc., Reading, MA, 1994.
17. Schneier, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* John Wiley & Sons, Inc., New York, NY, 1994.
18. Shamir, A., Rivest, R., and Adleman, L. Mental poker. Technical Report MIT/LCS/TR-125, Massachusetts Institute of Technology (MIT), Cambridge, MA, 1979.
19. Yung, M. Cryptoprotocols: Subscription to a Public Key, the Secret Blocking and the Multi-Player Mental Poker Game. In *Advances in Cryptology — CRYPTO '84* (1985), G. Blakley and D. Chaum, Eds., Springer-Verlag, pp. 439 – 453.

This article was processed using the LaTeX macro package with LLNCS style