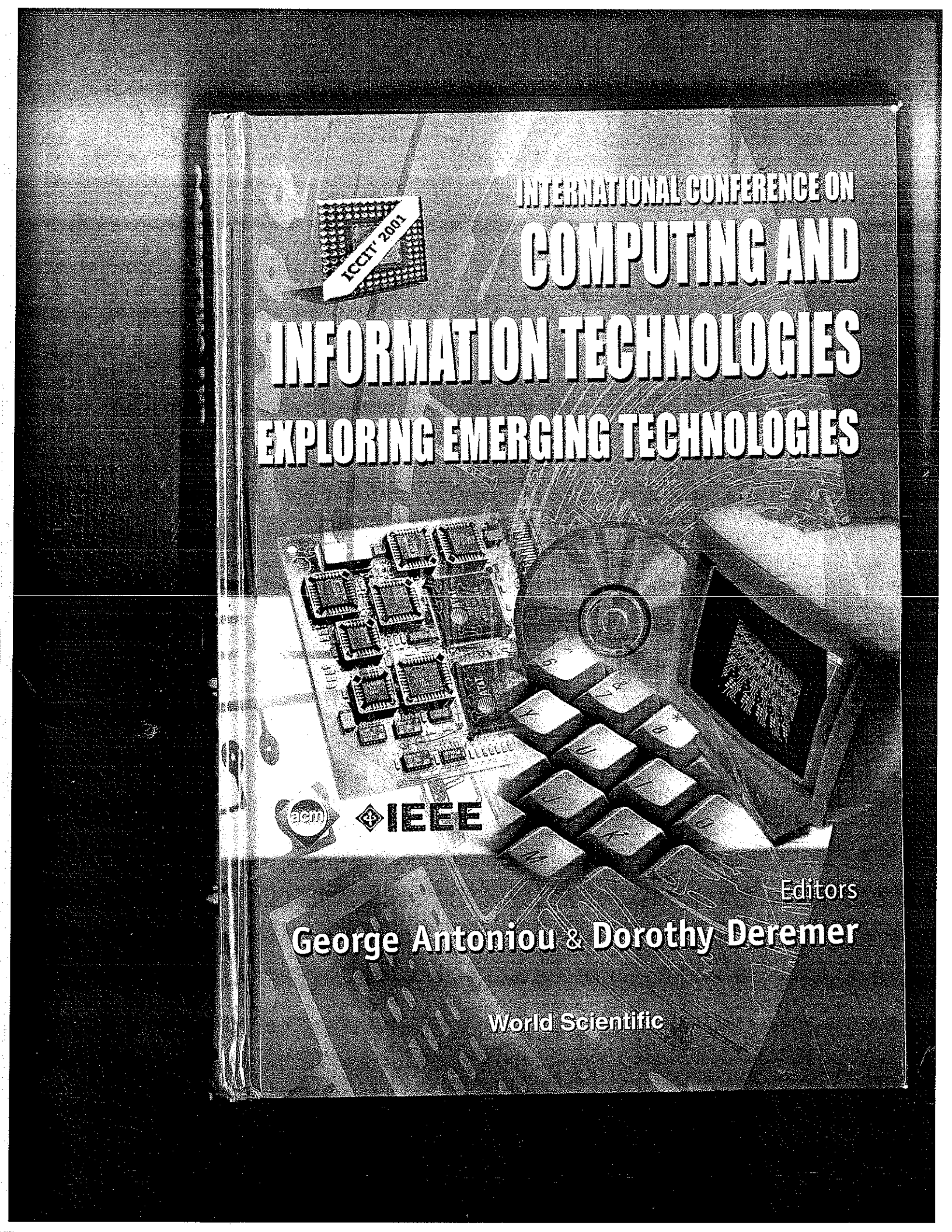
ICCIT' 2001
INTERNATIONAL CONFERENCE ON
COMPUTING AND
INFORMATION TECHNOLOGIES
EXPLORING EMERGING TECHNOLOGIES
acm
IEEE
Editors
George Antoniou & Dorothy Deremer
World Scientific

INTERNATIONAL CONFERENCE ON

COMPUTING AND INFORMATION TECHNOLOGIES

EXPLORING EMERGING TECHNOLOGIES

Montclair State University, NJ, USA

12 Oct 2001

Editors

George Antoniou
Dorothy Deremer

Montclair State University

World Scientific
New Jersey • London • Singapore • Hong Kong

PUBLIC KEY ENCRYPTION AND TRANSPARENCY IN INTERNET CASINOS

H.M. HUBEY
Department of Computer Science, Montclair State University, Upper Montclair, NJ 07043 USA
E-mail: hubeyh@mail.montclair.edu

P. B. IVANOV
International Science and Technology Center, 9 Luganskaya Street, P.O. Box 25, Moscow, 115516, Russia
E-mail: ivanov@istc.ru

Different games have different characteristics. One of the standard characteristics is that of transparency of the state of the game; for example, chess is an open game; whereas card games usually are not. The second characteristics are the transparency of the protocols: In games such as the slot-machines, the players have no choice. Complete trust in the house is required. At the other end of the spectrum is craps where the rules for tossing the dice are quite explicit and it is virtually impossible to cheat. Backgammon belongs to the latter type except the Internet or the computer version.

1 Introduction

Backgammon is a game very easy to learn to play badly and hard to master. Probability calculations are much easier for a machine than a human, however, heuristics are quite easy. There are excellent players who cannot even read (literally) or

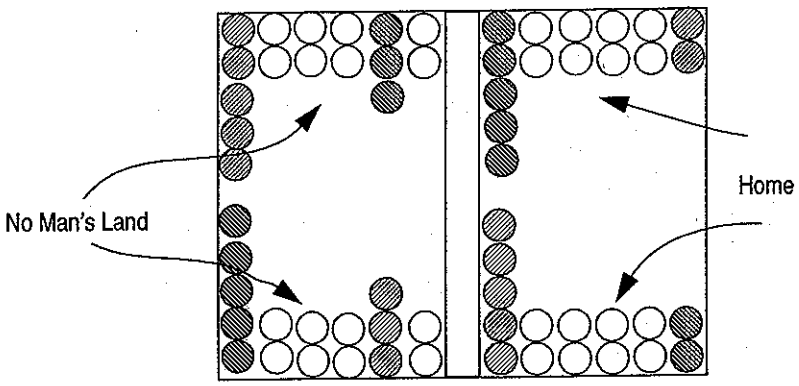

Figure 1: *Backgammon Board:* Each player moves the pieces into his home board (moving in opposite directions) and bears off (takes them off).

have only 5th grade educations. Effectively speaking players move their pieces in

opposite directions. The board is 'folded over' so that it takes the shape of Fig (1). The players try to bring all their pieces into their homeboards after which they bear off (pick them all up). Two or more pieces on a slot is called a point or making a point. A single piece occupying a slot is called a blot. One cannot put a piece on any of the opponents points. If a piece lands on a blot of the opponent that piece goes off the board (on the bar) where it goes into the opponents homeboard according to the toss of the dice. Entrance back into the game is not automatic.

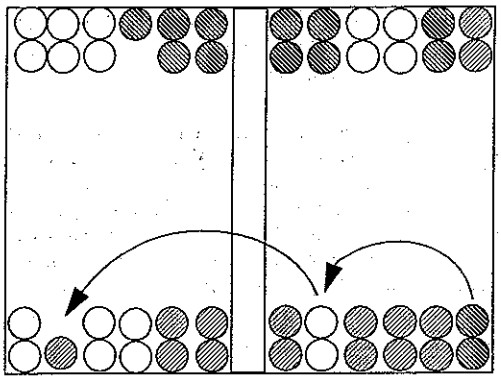

Figure 2: *An Example of a Miracle Roll:* The checkered piece gets a 46 and hits the opponent's piece which effectively seals the fate of the game. There are many types of miracle rolls.

The game is a stochastic game, and the computations are difficult for human beings, however the heuristics are quite easy. If we group the numbers on the die into two sets H (high) and L (low), then 50% of the time, we get HL or LH. Obviously HH and LL occur 25% of the time. The average roll moves a piece 8 pips (8 spaces). In live plays with actual players it is possible to exert control over what numbers come up when the dice are thrown. In actual play, a match can be anywhere from 5 to 11 points (or games) so as to make luck insignificant. Furthermore the rating system works over many matches. In the course of a single game, there occur what might be called bad luck streaks or good luck streaks, but they are generally of short duration. Someone who has been on the receiving end of a short bad luck streak may cheat only once, and reverse the course of the game.

Indeed, it is possible even in fair games for the roll of the dice to have the same effect. For those who want to practice the effect of this, a rule similar to chess may be annunciated, the *Hubey Rule*[1]. The better player may spot the poor player a roll or two, similar to giving up a rook or a queen. In other words, when

the player is desperately stuck, he may opt to use his option to select any specific roll of the dice.

In Internet games there are advantages and disadvantages. On the whole it is possible to fix up things so that the advantages outweigh the disadvantages. One of the advantages is that it is not necessary to keep account of the rolls, or to depend on the players to stay honest. The software keeps track of it all. There is no way the other player (unless it is the machine owned and operated by the house) can affect the dice roll. Secondly, there is no way for the player to bear off more pieces, or put pieces in the wrong place or disagree about the value of the dice. In games in which the dice are thrown by a computer program there are two main problems. The first problem is that the random numbers (actually pseudo random numbers) may be seriously correlated [1]. However this will have the general effect of making all games take on a certain coloring; there might be more streaks in computerized games because of the correlation, but it will be fair to every player over the long run.

2 Miracles and the Duck Rule

The second problem is more serious; there is no way for the player to know what algorithm is being used by the house and if the toss is really "fair", and this not only in the case of a player playing against the house machine but especially during such games. This is similar to playing the one-arm bandits in casinos. It is probably because people are already habituated to taking what is offered that this game persists. There is no way for a player to know if the one-arm bandit is being manipulated by someone or if it has a program that makes things look random. In other games such as craps or blackjack there is much more openness to the algorithm and rules of the game. There are very few ways for the house to manipulate the outcome. But in backgammon tournaments when the player plays against a machine there are serious trust and security holes. In such a case, the rule is; *if it looks like a duck, walks like a duck, and quacks like a duck, it is probably a duck.*

There are many ways in which a software can manipulate the outcome of the game, and most of these are based on "magic rolls". These are "lucky rolls" that save the player. Unfortunately, if all the luck in its various manifestations is on one side, then the duck rule (see above) applies. It is difficult enough to test a sequence of integers for randomness, and it is even more difficult to show that for software that combines together to toss the die and play the game. In that case, the *duck rule* applies. There are many ways to manipulate the game.

1) **Constant drain:** this is when one side consistently throws bigger numbers without suffering any bad consequences. The "luck" here is spread out over the whole game. Someone has to win, so it's hard to tell if the rolls are really random.

2.1) **Miracle rolls**: In Figure 2, the CH (crosshatched) player rolls 46 and hits the LI blot, whereas LI might roll several times and not even be able to get out of the opponents homeboard, although his chances of getting out are greater than that of CH. If these miracle rolls do not occur to both sides, the duck rule applies.

2.2) Throw miracle doubles and overtake the opposition over and over. This can also be hidden cleverly by making the opponent also get doubles, but either low ones or even better, useless or harmful doubles. When the statistics are tallied, both players have the same average number of doubles. This is also a good technique if the doubles come right at the end, where they do little good, for example if the player bears off the last blot with a 66.

2.3) Another clever and nice trick is to lose every once in a while but only 1 point at a time, and to win big. This can be done in a variety of ways.

2.4) When the machine hits 70-80% of the player's pieces that land in the no man's zone when it should be in the neighborhood of 30% and the player can only hit the machine's no-man's-zone pieces only about 20% of the time, and at opportune moments the duck rule applies.

2.5) The *Mexican Standoff* miracle: Often, when the game reaches a critical point players' points are left in the other's zones and neither wants to run first and break his point and leave a blot. Whoever throws the first 6 (usually) is a victim of the Mexican Standoff, especially if he gets hit more than the average.

3.1) **Bearing off:** One trick that will surely ruin anyone is to let the player seem to be winning until almost the end. Arrange all the pieces in the machine's home board so that it is almost airtight, and then force the player to leave a blot while bearing off, and of course, hit it!

3.2) Throw bigger numbers at each roll, and doubles while bearing off. This must be done if the other tricks have been overused.

3.3) Throw big doubles near the end and take the game. It also helps if the player has a "hole" in his homeboard while bearing off, say at point 3. Then if every roll has a 3 in it, such as 31, 34, 36, the player slowly falls behind.

4) Finally, if all of these have been overused, it does not hurt to cheat once or twice directly select the best roll for that play.The best thing that Internet backgammon sites can do is to allow the *Hubey rule* to be implemented so players can

see for themselves that cheating only once per game vastly improves their chances.

5) And over all, the last trick, win by a score such as 9-8, 9-7, 9-8, etc. enough times and win many games with a single roll or a single blot to give the impression that the player only needs to improve an iota.

After 40-50 matches like this, it becomes impossible for a player to trust in the laws of probability theory. In such a case, it is quite likely that laws of probability are not operating, and that the program is cheating, according to the duck rule. Even if it is not cheating, it is important not to give even an impression of cheating. If the probability for the machine to win p=0.9, the probability of winning 10 games in a row is only $(0.9)^{10}$=0.3487, and probability of winning 30 straight games is $(0.9)^{30}$=0.424. If p=0.95, then the results are $(0.95)^{10}$=0.5987, and $(0.95)^{30}$=0.2146. With even p=0.99 we obtain $(0.99)^{10}$=0.9044 and $(0.95)^{30}$=0.7397. More sophisticated probability calculations can be made relatively easily.

The only method of dispelling feelings that the machines are cheating is to employ open standards and protocols. First, there should be a set of robust random number generators to choose from [2,3,4,5]. There are many kinds of tests for random numbers [3]; there is no excuse for using ones that fail simple tests such as the serial correlation tests [1]. The best random number generators are congruential and usually have some seed or modulus that is a prime. This number can be set for every session. In the days of broadband access and 1 GHz CPUs there is no excuse for not using this. There are rules that even children employ to minimize cheating. For example, when choosing up sides, two players will simultaneously display 1, 2 or 3 fingers. They are all added up, and if the sum is even one player wins, and if odd the other. The key here is that the result is computed from both inputs not from a single one. A protocol resembling this can be created for Internet gaming. The key is the *Diffie-Hellman key exchange*.

3 The Diffie-Hellman Exchange

For any prime number p, a primitive root ρ is such that $\rho \mod p$, $\rho^2 \mod p$,... $\rho^{n-1} \mod p$ are all distinct and consist of the integers 1 to p-1 in some permutation. With p a publicly known prime number, and an integer ρ that is a primitive

root of p, the human player selects a random integer $X_h < p$ and computes $Y_h = \rho^{X_p} \ mod \ p$. The machine selects $X_m < p$ and computes $Y_m = \rho^{X_m} \ mod \ p$. Both the machine and the human keep the X value private and make the Y value available publicly to the other side. The human player computes the key as $K_h = Y_m^{X_h} \ mod \ p$. The machine computes the key as $K_m = Y_h^{X_m} \ mod \ p$. From the rules of modular arithmetic both of these are equivalent [6].

4 Simple Rules to make the game transparent, and open

This protocol can be used to generate several algorithms to make the game open to inspection by anyone and show that it is an honest game. One way would be to generate the integers $\{X_h, X_m\}$ at each step (by the human player's CPU and the bot of the Internet game provider respectively) using a random number generator. Then they can use a simple algorithm to compute $r = (K \ mod \ 5) + 1$, where the various parameters are explained below. Then using the exact same random number generator then can use r to generate the other die number. This can be repeated for every roll. *Even if one side cheats in the generation of the random number set $\{X_h, X_m\}$ they still cannot control the final output.*

5 Other alternatives

Another way to proceed, after the initial exchange of the key, would be to make sure that the number generation is transparent to both the bots and the human players, (or if the day should arrive, to bots that compete against each other). There must be input into this process of random number generation from both the player (actually his/her CPU) and the Internet backgammon provider who also provides the bots. Random number sequences are generally of the type

$$r_{n+1} = \alpha r_n \ mod \ M \qquad (1)$$

M should be a large prime. Good choices for the parameters α and M can be found in [2,3]. A good choice for M, given by Lehmer, is $M = 2^{31} - 1$, a Mersenne prime number. From a possible choice of more than 2 million integers, there are only a handful for the multiplier α, that pass several tests of random-

ness [2]. The minimal standard random number generator uses $\alpha = 16807$. This multiplier can be set as the default and the others can be given to the players as choices should they wish to change.

The algorithm follows: both the player and the machine generate the initial random numbers r_0^m, and r_0^p, using some random generator function $r_0 = f_j(\sigma, \tau, k)$ where f_j is the jth acceptable random generator function. These are supplied by the Internet game service provider. These functions should be coded in an open language such as Java which can be examined by millions of competent people, and it should be certified to be working correctly by organizations devoted to Internet gaming. The seed σ can be chosen in a number of ways. It can be entered by the player or it could be a function of the timer register, thus $\sigma = F(\tau)$, and k is some parameter chosen for this session. This parameter can be one of several to add more randomness and choices to the process. Then these random numbers are exchanged between the player and the machine. For security reasons, these exchanges should also use the Diffie-Hellman key exchange algorithm. After the exchange each side computes two new random numbers. In general the next set of random numbers are generated as

$$r_{n+1} = g_1(r_n^m, r_n^p, k) \text{ and } r_{n+2} = g_2(r_n^m, r_n^p, k) \quad (2)$$

It should be noted that both machines (the player's and the bot's) use the same two functions g_1 and g_2 and the same parameters to compute the same numbers. To make sure there are no errors, these numbers should be exchanged and compared although with the reliability of optical fiber connections the need for this decreases in time. The most important part of the algorithm as can be seen is that as in the childrens' algorithm of choosing up, both inputs determine the random numbers.

For practical reasons the numbers generated internal by the player's CPU should be displayed separately and the two integers to be played should be displayed on the game board. It should be possible for the player at any time to check the calculations, and just as importantly, the algorithm should be coded in a language like Java which can be compiled by the player.

There should be move toward developing a standard API so that the fairness of the game can be made obvious to millions of programmers worldwide. It is in the interest of the Internet game service providers to make this available to the

users. The Internet is a different kind of a world than other aspects of life. Millions of people might crank slot machine arms mindlessly but it won't work on the Internet, at least not for long. Indeed, it will even influence the future of slot machines in casinos.

6 Conclusions

The number of ways in which one can cheat at backgammon is even more than this especially in the days of AI and supercomputers. A simple random number generator checking program cannot spot sophisticated cheating techniques. What we need are either (i) a metric that determines how "lucky" one side is, or (ii) a clearly open version of how the random numbers are generated. The "luck" part can be worked out in the future, however the easiest way to approach this is to create an open environment. The concept of Open Gaming, is aan idea whose time has passed. It is time for games of chance to use open standards and fit into the general scheme of things on the Internet and partake in the future which the Internet has spawned.

Notes

1. The **Ivanov Rule** says that "If you don't want it, just don't play".

References

1. Hubey, H.M., and A. Gutierrez, *Testing Random Numbers Via Cumulants,* Proceedings of the Twenty-third Annual Pittsburgh Conference on Modeling and Simulation, May (1991), pp. 147-152.
2. Park, S. and K. Miller, *Random Number Generators: Good Ones Are Hard to Find*, CACM, vol. 31(1988), No.10, 1192-1200.
3. Bratley, P., B. Fox, and L. Schrage (A Guide to Simulation, Springer-Verlag, 1983)
4. MacLaren, M and G. Marsaglia, *Uniform Random Number Generators*, J. ACM, vol.12(1965), no.1, 83-89.
5. Marsaglia, G. *Random Numbers Fall Mainly in the Planes*, Proc. NAS, vol. 61(1968), 25-28.
6. Stallings, Wm, Cryptography and Network Security, (Prentice-Hall, Englewood Cliffs, 1999)