

## **Generating Truly Random Numbers!**

When coming up with the specification for our random number generation algorithm, we were faced by many dilemmas.

Do we build this in-house? Do we outsource? Is it going to be fast enough? How will we make sure the cards dealt are unique to each table - there can be dozens of games running on at the same time! How are we going to make this truly random?

### **Mission Objective:**

We wrote down a list of the requirements we had and then came up with our mission objective and goals for this software algorithm. Our objective was to come up with a shuffling algorithm which would ensure that the seeds used in the random number generator are truly random and, given the constraints of a real-time poker game, next to impossible to predict.

The seeds generated from our software algorithm system would then be plunked down into the Standard Template Library function `random_shuffle`, which would shuffle all 52 cards of a standard card-deck using the provided seed. After the shuffling procedure was complete, the according cards would be dealt out

### **The Procedure:**

To come up with a good shuffling algorithm, we decided two things are required. First, you need a good pseudo random number generator. Second, you need a high quality source of entropy. In designing our system, we gave serious consideration to using a hardware entropy source but concluded it isn't good enough and not as reliable as we'd like.

### **Implementation:**

When designing a shuffling card algorithm, one of the most important issues is making sure you have valid random entropy sources. The debate became: "how do we pull good sources of entropy from a software driven system?" What we came up with was: "Why not pull random entropy sources from each client connected to the local table?" By gathering entropy sources from each client, we could mesh this into the main entropy seed; our reasoning was, with a couple of clients we'd be able to generate a very good source of entropy per table.

Now the question was, what source of entropy would be read from each client's computer. After short debate we decided the best source of entropy we could get from each client would be the user's mouse position (since it's always moving) and the current millisecond clock time on their computer's system-time. To ensure that the data is very random and our entropy seeds are "fresh with live data", we collect these "seeds of entropy" several times per minute. Now that we've generated a good source of entropy for the seed associated with each poker table, we must find a pseudo random number generator that is extremely fast and provides us with random numbers of very high quality.

### **Finding The Right Random Number Generator:**

We decided the best pseudo random number for our needs would be a Mersenne Twister . The Mersenne Twister is a pseudorandom number generator which was developed in early 1997 and provides fast generation of very high quality random numbers.

The Mersenne Twister takes input data which should be quite random and it will return back a true random number (which will be used to shuffle all 52 cards). Basically the more random your input data is, the better your seed is going to come out. This will result in a better shuffled deck.

So before a new deck is shuffled, we use a Mersenne Twister. The data we provide it with is all this new entropy seed data we've collected from all the clients and the server's local chipset Timer Stamp Counter.

Doing this helps to ensure that the cards dealt out are truly random and not weighted in any direction; and next to impossible to guess. And we don't just shuffle the deck once! The Timer Stamp Counter is constantly changing several times per nanosecond, so we have the deck shuffled over 10 times to ensure that the cards you get truly are random to your poker-table - and well shuffled!

### **Conclusion:**

By revealing our shuffling algorithm, we hope that you can get a better understanding of some of the measures we use to ensure the integrity of your online poker gaming experience.

We hope you enjoyed this brief description of the shuffling algorithm procedure we use. Thanks, and we're looking forward to see you on our tables!

And now for our plug!

Please come visit us at <http://www.crystalpoker.net> , download our software, tell your friends & family and play a few hands of poker!