Efficient Electronic Gambling: An Extended Implementation of the *Toolbox for Mental Card Games*

Heiko Stamer
University of Kassel, Department of Mathematics and Computer Science
Heinrich-Plett-Straße 40, D-34132 Kassel, Germany
http://www.theory.informatik.uni-kassel.de/~stamer
stamer@theory.informatik.uni-kassel.de

Abstract: There are many wonderful protocols in cryptography which are still waiting for their realization. Here we consider efficient solutions for secure electronic card games. Our contribution seems to be the first known practical implementation that requires no trusted third-party and simultaneously keeps the players' strategies confidential. The provided open source library LibTMCG can be used for creating secure peer-to-peer games and furthermore for some unusual applications, e.g., secure multi-party computation or simple electronic voting schemes.

Keywords: Mental poker, e-gambling, secure electronic card games, verifiable secret shuffle, secure multi-party computation, zero-knowledge proofs.

C. Wolf, S. Lucks, P.-W. Yau (Eds.): WEWoRC 2005, LNI P-74, pp. 1–12, 2005.

1 Introduction

Electronic gambling is fascinating for a large community of players. With the availability of fast communication networks a question related to cryptography has been raised: Is it possible to play a fair card game over a network without the need for a trusted third-party?

The first answer was given by Shamir, Rivest, and Adleman [SRA79]. They have shown that a complete solution is impossible in an information theoretic sense. Fortunately, they also developed a protocol for dealing cards which works under realistic computational assumptions. This protocol was found to be insecure by Lipton [Li81] and Coppersmith [Co85], because it leaks partial information about the cards. Goldwasser and Micali [GM81] used probabilistic encryption to solve this issue. However, an important drawback still remained: In order to detect a cheating player it was necessary to disclose all secrets at the end of each game. The early solutions only covered the classical two player scenario while later proposals appeared for three or more parties [BF83, Yu85]. Nowadays an interesting direction is the design of dropout-tolerant protocols [KKO97, CSD05].

In card games like Poker it is essential to keep the players' strategies confidential. The first solution for this problem was the zero-knowledge protocol suite by Crépeau [Cr87]. But it turned out that an implementation [Ed94] of the original scheme was not at all practical.

Few years later Schindelhauer [Sc98] introduced a more general toolbox which extents the previous work of Crépeau. Roughly speaking, the type of a card is shared among all participating players through bitwise representation by quadratic (non-)residues. Thus the security relies on the well-known *Quadratic Residuosity Assumption (QRA)*. The correctness of the performed operations is assured by interactive zero-knowledge proofs. Unfortunately, the size of the card representation grows linearly in the number of players and logarithmically in the number of different types.

Recently Barnett and Smart [BS03] have proposed a very efficient solution which relies on homomorphic encryption. They describe the abstract concept of a cryptographic primitive denoted as *Verifiable k-out-of-k Threshold Masking Function* and give two possible instantiations. The security can be based either on the *Decisional Diffie-Hellman Assumption (DDH)* or on an assumption related to factoring. The correctness of the most operations is shown by honest-verifier zero-knowledge proofs of knowledge which all have the special soundness property [CDS94]. Moreover, the card encoding is independent of the number of players and almost independent of the number of different types.

1.1 Motivation and Related Work

There is not much known about the software and the quality of randomness used by working Internet casinos [HS97, AT04]. However, almost all online gambling solutions rely on a "trusted server" which is often operated by the casino company itself. Even if they make claims about the security of their systems or random-number generators, one cannot trust such statements due to the lack of physical and third-party auditing. For the future it is not very likely that the existing Internet casinos will use public verifiable software in combination with strong cryptography to guarantee a fair play.

Nevertheless, by looking at the growing possibility of a fast network access for individuals and hence the emerging trend of virtual communities (e.g. Internet gaming) it seems to be a good idea to eliminate the need for trusted third-parties in such distributed environments. In fact, one can easily imagine cases where a trusted third-party is not even available. Of course, we should simultaneously keep in mind that a reasonable level of efficiency is always required.

Now let us briefly review the related work w.r.t. electronic card games. We are aware of only a few real-life implementations [Ed94, Pi02] that fulfill all necessary conditions for a complete solution in the sense of Crépeau [Cr87], i.e. no trusted third-party is needed and rather than other solutions [CDR03], the privacy of the players' strategies has to be preserved. Unfortunately, these first trials were not really practical, e.g., the implementation by Edwards is reported to have taken eight hours to shuffle a deck.

1.2 Contribution and Organization

By incorporating the recently proposed encoding scheme [BS03] we were able to create an efficient implementation of Schindelhauer's toolbox. We have applied some straightforward optimizations to increase the performance and the practicability. Moreover, the development is still ongoing and due to the nature of *Free Software* everybody who likes can use the written source code in his own projects.

The next section is devoted to the details and precise settings of our contribution—the development of the general purpose library LibTMCG. Then we present a working example for the German card game Skat and discuss its practicability. Finally we mention some unusual applications and conclude with new directions in the further development.

2 LibTMCG: A Free Library for Secure Electronic Card Games

The aim of this project is the development of a general purpose library for creating secure card games. The protocols minimize the effect of coalitions and achieve the confidentiality of the players' strategies, i.e. the participants are not required to reveal hidden cards respectively secret keys to show that they did not cheat. All operations are considered in the "honest-but-curious" (aka semi-honest) adversary model. That means, the players follow the protocol properly but they may gather information and share them within a coalition to obtain an advantage in the game. Hence we are not concerned with robustness or availability issues which are hard to solve in an asynchronous setting.

We have implemented the main core of Schindelhauer's toolbox (TMCG), i.e. the important functionalities like masking, shuffle, pickup, and public disclosure of cards and stacks, respectively. Some exotic operations are still missing, e.g., the possibility to insert a card secretly into a stack. The abstract primitive *Verifiable k-out-of-k Threshold Masking Function* (VTMF) of Barnett and Smart has been added recently. Our library only provides the discrete logarithm instantiation for such a VTMF, because the corresponding key generation is easy to realize even in a distributed game environment.

For the sake of completeness we briefly review the two card encodings [Sc98, BS03]:

TMCG: Let $\mathbb{Z}_m^\circ$ be the set of integers from $\mathbb{Z}_m^*$ with a positive Jacobi-Legendre symbol. $\mathbb{QR}_m$ denotes the set of quadratic residues and $\mathbb{NQR}_m$ the set of quadratic non-residues modulo m. Further we define $\mathbb{NQR}_m^\circ := \mathbb{Z}_m^\circ \cap \mathbb{NQR}_m$.

Each player $i \in \{1, \ldots, k\}$ chooses two large prime numbers p_i, q_i for his secret key. The public key consists of $m_i = p_i \cdot q_i$ and an arbitrary element $y_i \in \mathbb{NQR}_{m_i}^\circ$.

We make additional constraints, namely that p_i, q_i are safe primes, $p_i \not\equiv 1 \pmod 8$, and $p_i \not\equiv q_i \pmod 8$, to employ the non-interactive zero-knowledge proof by Gennaro et al. [GMR98]. The first two stages of this proof show that m_i has been created correctly as product of exactly two different primes. With another straightforward argument we assure that y_i is indeed a quadratic non-residue modulo m_i.

Let M be the number of different types of cards. A single card Z is represented by

$$Z = \begin{pmatrix} z_{1,1} & \cdots & z_{1,\lceil \log_2 M \rceil} \\ \vdots & \ddots & \vdots \\ z_{k,1} & \cdots & z_{k,\lceil \log_2 M \rceil} \end{pmatrix}$$

where every $z_{i,j} \in \mathbb{Z}_{m_i}^{\circ}$ encrypts a shared bit of the card type $\tau \in [0, M-1]$. As long as $z_{i,j}$ is randomly and uniformly distributed the corresponding predicate

$$\mathsf{qr}(z_{i,j}, m_i) := \begin{cases} 0 & z_{i,j} \in \mathbb{QR}_{m_i} \\ 1 & \text{otherwise} \end{cases}$$

remains unknown (under the Quadratic Residuosity Assumption) except for the ith player. Only if all k parties agree to reveal these information and hence prove their correctness (with zero-knowledge proofs) accordingly, then the type τ can be computed by the term ($\oplus$ denotes the exclusive-or)

$$\tau = \sum_{j=1}^{\lceil \log_2 M \rceil} 2^{j-1} \cdot \bigoplus_{i=1}^{k} \mathsf{qr}(z_{i,j}, m_i).$$

Note that 1 is always a quadratic residue. Thus an open card is simply given by the binary representation $b_1, \ldots, b_{\lceil \log_2 M \rceil}$ of its type τ and the corresponding matrix

$$\left((y_1^{b_1}, \ldots, y_1^{b_{\lceil \log_2 M \rceil}}), (1, \ldots, 1), \ldots, (1, \ldots, 1) \right).$$

Obviously, $\mathsf{qr}(z_{i,j}, m_i) = 0$ for all $i = 2, \ldots, k$ and $j = 1, \ldots, \lceil \log_2 M \rceil$. The (re-)masking operation of a card Z to Z' is performed element by element by

$$z'_{i,j} = z_{i,j} \cdot r_{i,j}^2 \cdot y_i^{b_{i,j}} \bmod m_i,$$

where $r_{i,j} \in_R \mathbb{Z}_{m_i}^*$ and $b_{2,1}, \ldots, b_{2,\lceil \log_2 M \rceil}, \ldots, b_{k,1}, \ldots, b_{k,\lceil \log_2 M \rceil} \in_R \{0,1\}$ are randomly and uniformly chosen. To keep the type of the former card Z all masking bits of the first row have to be computed by

$$b_{1,j} = \bigoplus_{i=2}^{k} b_{i,j}.$$

The quadratic residuosity property of every $z'_{i,j}$ has to be changed randomly and uniformly, because otherwise it will not hide the permutation of a secret shuffle (stack of cards). The above (re-)masking operation forms a equivalence relation between encoded cards. Therefore it is easy to prove the correctness interactively using the well-known *"cut-and-choose"* methodology [Ra78].

VTMF (discrete logarithm variant): The k parties choose a finite abelian group G in which the Decisional Diffie-Hellman Assumption holds. They agree on a element $g \in G$ of sufficient order q and an appropriate message space $\mathcal{M} \subseteq G$ (possible

types of cards). Further there is the nonce space $\mathcal{R} = \mathbb{Z}_q$ (randomizers) and the encoded cards $\mathcal{C} \subseteq G \times G$ themselves.

Each player generates a secret key $x_i \in_R \mathbb{Z}_q$ and publishes $h_i = g^{x_i}$ along with the zero-knowledge proof of knowledge $PK\{(\alpha_i) : h_i = g^{\alpha_i}\}$. The shared public key is formed from $h = \prod_{i=1}^{k} h_i$. Specifically, this procedure sets up a non-robust threshold cryptosystem [DF90, Pe91] which aims to protect the confidentiality of the types. We suppose that the discovered flaws [GJKR99] regarding the non-uniform generation of the shared secret key $x = x_1 + \cdots + x_k$ does not affect the security of the card encoding.

Again, the main operation is the (re-)masking of cards which is basically a semantical secure ElGamal encryption [El85, TY98], i.e. $\mathcal{M} \times \mathcal{R} \to \mathcal{C}$ resp. $\mathcal{C} \times \mathcal{R} \to \mathcal{C}$:

$$\begin{aligned} (m, r) &\mapsto (c_1 = g^r, c_2 = h^r \cdot m) \\ ((c_1, c_2), r) &\mapsto (c_1' = c_1 \cdot g^r, c_2' = c_2 \cdot h^r) \end{aligned}$$

The correctness is shown by a *Proof of Equality of Discrete Logarithms* [CP93], i.e. $PK\{(\alpha) : c_1 = g^\alpha \wedge c_2/m = h^\alpha\}$ resp. $PK\{(\alpha) : c_1'/c_1 = g^\alpha \wedge c_2'/c_2 = h^\alpha\}$. The verifiable threshold decryption of cards can be performed in an obvious way.

Due to the lack of space the details of the zero-knowledge proofs have been omitted here. The interested reader is referred to the original literature [Sc98, BS03]. We have increased the efficiency of our implementation by the following widely-used optimizations.

TMCG: The primes q_i and p_i are chosen to be congruent 3 modulo 4 (Blum integers). Thus the computation of the modular square roots is feasible with two single exponentiations. Then the Chinese remainder theorem is applied to construct a common square root modulo m_i. Additionally, a small value can be chosen for $y_i \in \mathbb{NQR}_{m_i}^{\circ}$.

VTMF: In general we use the cyclic group $G := \mathbb{QR}_p$ (quadratic residues) of prime order q, where p is a safe prime and $q = (p-1)/2$. The DDH problem is believed to be hard [Bo98] in G, for all p of reasonable cryptographic size, e.g., $\ell_p = 1024$ bit. The choice for this particular subgroup of $\mathbb{Z}_p^*$ was made due to the fact that we can test membership simply by computing the Legendre symbol in time $O((\log_2 p)^2)$.

Further, one can improve some calculations by choosing the generator g as a power of two. In fact, $g = 2^s$ is a quadratic residue modulo p, if $p \equiv 7 \pmod 8$.

VTMF: Many of the involved modular exponentiations can be calculated very efficiently, because the bases are always fixed during a game session. We employ simple precomputed tables to speed up this operation. There exist more flexible tradeoffs between memory usage and computation time [LL94], but unfortunately the danger of a patent violation encumbers their usage in free software projects like ours.

VTMF: The random exponents (masking operation) are shortened to a size of $\ell_r = 160$ bit. Koshiba and Kurosawa [KK04] have shown that under the *Discrete Logarithm with Short Exponent Assumption (DLSE)* the DDH problem is not weakened. In addition the generator has to be shifted to an appropriate size, i.e. $g = 2^{2^{\ell_p - \ell_r}}$.

TMCG, VTMF: The commitments for the proof of the secret shuffle are shortened to a size of $\ell_c = 160$ bit by a cryptographic hash function (e.g. RIPEMD-160). On the other hand, the hash function is used again to turn the proofs of knowledge into non-interactive zero-knowledge proofs (NIZK) using the well-known Fiat-Shamir heuristic. Hence the soundness still holds in the *Random Oracle Model (ROM)*.

Each of the above mentioned optimizations was considered carefully with respect to timing attacks [Ko95]. Therefore we loose some of the gained efficiency.

Our implementation is available as open source library [St05a] released under the GNU General Public License [FSF]. Currently it comprises approximately 7 300 lines of C++ code. The following tables give a short comparison of the computational (Table 1) and communication complexity (Table 2) of both implemented schemes.

Table 1: Comparison of the computational complexity (LibTMCG implementation)

Operation	TMCG $p_i, q_i \equiv 3 \pmod 4$ [Sc98]	VTMF discrete logarithm variant [BS03]
Masking of a card	$= 3k\lceil\log_2 M\rceil\texttt{mulm}$	$= 2\texttt{powm}\langle\ell_r\rangle + 2\texttt{mulm}$
Prover	$= t \cdot 5k\lceil\log_2 M\rceil\texttt{mulm}$	$= 2\texttt{powm}\langle\ell_q\rangle + 5\texttt{mulm}$
Verifier	$= t \cdot 3k\lceil\log_2 M\rceil\texttt{mulm}$	$= 2\texttt{powm}\langle\ell_q\rangle + 2\texttt{powm}\langle\ell_r\rangle + 8\texttt{mulm}$
Decryption of a card	$= 2\lceil\log_2 M\rceil\texttt{mulm}$	$= 1\texttt{powm}\langle\ell_q\rangle + 2\texttt{mulm}$
Prover	$\leq \lceil\log_2 M\rceil((4t+5)\texttt{mulm} + 2\texttt{powm}\langle\ell_m/2\rangle)$	$= 3\texttt{powm}\langle\ell_q\rangle + 1\texttt{mulm}$
Verifier	$\leq \lceil\log_2 M\rceil(2t+2)\texttt{mulm}$	$= 4\texttt{powm}\langle\ell_q\rangle + 4\texttt{mulm}$
Shuffle of a stack $\mathcal{S}$	$= \lvert\mathcal{S}\rvert \cdot 3k\lceil\log_2 M\rceil\texttt{mulm}$	$= \lvert\mathcal{S}\rvert \cdot (2\texttt{powm}\langle\ell_r\rangle + 2\texttt{mulm})$
Prover	$\approx t \cdot \lvert\mathcal{S}\rvert \cdot 3k\lceil\log_2 M\rceil\texttt{mulm}$	$\approx t \cdot \lvert\mathcal{S}\rvert \cdot (2\texttt{powm}\langle\ell_r\rangle + 2\texttt{mulm})$
Verifier	$\approx t \cdot \lvert\mathcal{S}\rvert \cdot 3k\lceil\log_2 M\rceil\texttt{mulm}$	$\approx t \cdot \lvert\mathcal{S}\rvert \cdot (2\texttt{powm}\langle\ell_r\rangle + 2\texttt{mulm})$
k: number of players; M: number of types; t: security parameter (controls the soundness error probability of the interactive zero-knowledge proofs); mulm: modular multiplication et al.; powm$\langle\ell\rangle$: modular exponentiation with exponent of size ℓ		

Table 2: Comparison of the communication complexity (LibTMCG implementation)

Sizes of proofs and representations	TMCG $p_i, q_i \equiv 3 \pmod 4$ [Sc98]	VTMF discrete logarithm variant [BS03]
Card representation	$= k\lceil\log_2 M\rceil\ell_m$	$= 2\ell_p$
Key generation	$= 2(\eta_1 + \eta_2 + \eta_3)\ell_m$	$= \ell_p + \ell_q + \ell_c$
Masking of a card	$= t \cdot k\lceil\log_2 M\rceil \cdot (2\ell_m + 2)$	$= \ell_q + \ell_c$
Decryption of a card	$\leq t \cdot k\lceil\log_2 M\rceil \cdot (5\ell_m + 1)$	$= \ell_p + \ell_q + \ell_c + o(1)$
Shuffle of a stack $\mathcal{S}$	$= t \cdot (\ell_c + 1 + \lvert\mathcal{S}\rvert \cdot (k\lceil\log_2 M\rceil(\ell_m + 1) + \lceil\log_2 \lvert\mathcal{S}\rvert\rceil))$	$= t \cdot (\ell_c + 1 + \lvert\mathcal{S}\rvert \cdot (\ell_r + \lceil\log_2 \lvert\mathcal{S}\rvert\rceil))$
Default cryptographic sizes used in LibTMCG [St05a]: $\ell_m = \ell_p = 1024$ bit, $\ell_q = \ell_p - 1$, $\ell_r = \ell_c = 160$ bit Security parameters that control the soundness error probability of the key generation: $\eta_1 = 16$, $\eta_2 = \eta_3 = 128$		

The parameter t bounds the soundness error probability of the interactive zero-knowledge proofs (by sequential repeating each proof t times). It can be adjusted by the application to achieve a fine granulated tradeoff between the provided security and the efficiency.

Note that the proof sizes given in Table 2 are for a single verification process between two parties. If we want to achieve a k-out-of-k security then the total communication complexity increases by a factor of $O(k^2)$. However, at least the computation of the non-interactive zero-knowledge proofs has to be carried out only once by each prover.

3 SecureSkat: A Cryptographically Secure Card Game

The famous card game Skat [ISPA] is essentially played by three players. The deck consists of $M = 32$ different cards which are shuffled at the start of each single game. Every player receives ten private cards. The *Skat* (remaining two hidden cards of the deck) is given to the player who succeeds in a preceding bidding phase. He can either open it and exchange these two cards privately or leave them untouched on the table. Further, this player proceed alone against a temporarily coalition of the two other participants. The main part of the game is a battle for tricks. Each player, starting with the winner of the last trick, discloses successively one of his private cards in accordance with the rules. All in all, the goal for the single player is to make more than 60 card points by his tricks.

SecureSkat [St05b] is a peer-to-peer network implementation of Skat. The software employs the discrete logarithm VTMF instantiation provided by LibTMCG [St05a]. The negotiation of games and the control during a session is done with a simple ASCII-based protocol. The corresponding messages are transmitted over an IRC network (Internet Relay Chat [OR93]). Additionally, each player establishes an authenticated private channel (Blowfish-128 encrypted by default) with his opponents. These channels will be used to prove the k-out-of-k correctness of the performed card and stack operations. In our adversary model it is not really necessary to encrypt the channel. However, one can think about less coalition-resistant threshold scenarios where such a behavior is required.

3.1 Discussion

Table 3 shows the communication effort of SecureSkat in comparison with an outdated revision of our implementation. The card representation in OpenSkat version ≤ 1.9 was based upon the original quadratic residues encoding without any optimization. Consequently, it produces an enormous communication traffic for reasonable values of the parameter t. Hence the usage was limited to players with a high-speed connection.

This fundamental issue has now been solved due to the discrete logarithm based encoding of Barnett and Smart. The increased amount of computational work (modular exponentiations instead of computing squares) is often negligible on a modern hardware architecture. Further, this effort is partially amortized by the independence of k and M.

Table 3: Compressed network traffic on the private channel (for each single game and player)

t	Soundness error probability	OpenSkat ≤ 1.9 (TMCG) Assumption(s): QRA	SecureSkat [St05b] (VTMF) Assumption(s): DDH, DLSE, ROM
2	≤ 0.25	$\approx$ 3 MByte	$\approx$ 0.64 MByte
4	≤ 0.0625	$\approx$ 5 MByte	$\approx$ 0.72 MByte
8	≤ 0.00390625	$\approx$ 10 MByte	$\approx$ 0.8 MByte
16	≤ 0.00001526	$\approx$ 20 MByte	$\approx$ 1.02 MByte
32	$\leq 2.3284 \cdot 10^{-10}$	$\approx$ 40 MByte	$\approx$ 1.42 MByte
64	$\leq 5.4211 \cdot 10^{-20}$	$\approx$ 80 MByte	$\approx$ 2.18 MByte

On a conventional No-name-PC with an AMD Duron 1.3 GHz processor, running the Debian GNU/Linux 3.1 operating system (GNU Compiler Collection 3.3.4 and GNU Multiple Precision Arithmetic Library 4.1.4), we have made the following measurements for a single instance of SecureSkat. The key generation required 1150 ms of processor time, the shuffle of the deck 150 ms, the dealing of the deck 1650 ms, and the public disclosure of a card 60 ms for the prover and 110 ms for the verifier. The largest amount of time was spent by the interactive proof for the correctness of the secret shuffle (see Table 4a). These values do not include the communication time itself. For convenience we have also listed the required real time of the whole shuffle and dealing phase either over the internal loopback interface (`lo`) or a regular[1] DSL connection (`ppp0`).

Table 4: Measured performance: (a) Proof of the secret shuffle, (b) Shuffle and dealing phase

		$t = 2$	$t = 4$	$t = 8$	$t = 16$	$t = 32$	$t = 64$
(a)		$\approx$ 750 ms	$\approx$ 1500 ms	$\approx$ 3020 ms	$\approx$ 6060 ms	$\approx$ 12200 ms	$\approx$ 24180 ms
(b)	`lo`	$\approx$ 4 sec	$\approx$ 5 sec	$\approx$ 7 sec	$\approx$ 11.5 sec	$\approx$ 21 sec	$\approx$ 40.5 sec
	`ppp0`	$\approx$ 13 sec	$\approx$ 17.5 sec	$\approx$ 27.5 sec	$\approx$ 47.5 sec	$\approx$ 87 sec	$\approx$ 169 sec

4 Unusual Applications

With the help of LibTMCG it was possible to realize two other applications. One can utilize the data representation as cards to obtain working examples for secret voting and secure multi-party computation. These solutions are not very efficient and satisfy not all desired properties. However, they show further scenarios for the usage of our library.

Secret Voting: The following scheme is included in SecureSkat [St05b]: Every participant creates a privately masked card (his vote). These cards are stacked and the result is shuffled by each player. Finally, the stack is disclosed to all participants.

Secure Multi-party Computation: Recall the quite unusual technique for computing arbitrary boolean functions [dBo90, NR98] with a deck of cards. We have implemented the AND-protocol by Stiglic [St01], a copy protocol [CK94] for committed bits, and other helpful tools. They are available in a test program of LibTMCG.

5 Conclusion and Further Development

We have presented an efficient implementation of the *Toolbox for Mental Card Games*. The reduced communication complexity was principally due to the new card encoding scheme of Barnett and Smart. We have gained additional efficiency by some straightforward optimizations. Our contribution shows that meanwhile even proposals are practical

[1] 1024 KBit/sec downstream and 128 KBit/sec upstream

which do not require the disclosure of the players' strategies. The experience with SecureSkat is quite opposite to the previously stated belief [HS97, CDR03, AS05] about the infeasibility of such solutions. However, in games with many participating players or large card decks (e.g. Poker) the currently used techniques may still become very costly.

Finally, we want to mention further improvements in the ongoing development. First, the abstract notation of the VTMF primitive allows the usage of other communication efficient encodings, e.g., prime order groups over elliptic curves [Bo98]. Last but not least, we can replace the expensive proof of the secret shuffle by a more efficient zero-knowledge argument. Some proposals [Ne01, Gr03, Fu04, Wi05, Gr05] have been recently developed in the context of electronic voting and of course, they are applicable to electronic card games as well. Unfortunately, the advanced techniques in this area are covered by patents.

Updating Remark. Our first experiments with the implementation of Groth's shuffle protocol [Gr05] (interactive version) suggest that one can reduce the communication complexity (cf. Table 2, VTMF instantiation) approximately to $(3|\mathcal{S}| + 2)\ell_q + 8\ell_p + 4t$ bits at the soundness error probability of 2^{-t}. Hence the break-even point of such an advanced shuffle proof is around $t = 20$ (for $|\mathcal{S}| = 32$ and default security parameters of LibTMCG). However, a larger amount of savings is achievable by using a smaller subgroup than $\mathbb{QR}_p$. Regarding the computational complexity the situation is quite similar: The measured processor time for the shuffle proof in SecureSkat has been reduced to 1460 ms (cf. Table 4a), where the security level was fixed at 2^{-80}. Concluding, the above results suggest that the recently proposed shuffle proofs are major improvements for mental poker protocols.

Acknowledgment. The author thanks Andreas Klein, Friedrich Otto, Christian Schindelhauer, and the anonymous referees for their valuable comments and discussions.

References

[AS05] A. Askarov and A. Sabelfeld. Secure Implementation of Cryptographic Protocols: A Case Study Of Mutual Distrust. Accepted for the 10th European Symposium On Research In Computer Security (ESORICS 2005). To appear.

[AT04] A. AuYoung and C. Tuttle. Cryptographic Blackjack. Final Project Report CSE 207, University of California at San Diego, Spring 2004.

[BF83] I. Barany and Z. Furedi. Mental poker with three or more players. *Information and Control*, 59(1-3):84–93, 1983.

[Bo98] D. Boneh. The decision Diffie-Hellman problem. In *Proceedings of the Third Algorithmic Number Theory Symposium*, Lecture Notes in Computer Science 1423, pp. 48–63, 1998.

[BS03] A. Barnett and N.P. Smart. Mental Poker Revisited. In *Cryptography and Coding 2003*, Lecture Notes in Computer Science 2898, pp. 370–383, 2003.

[CDR03] J. Castellà-Roca, J. Domingo-Ferrer, A. Riera, and J. Borrell. Practical Mental Poker Without a TTP Based on Homomorphic Encryption. In *INDOCRYPT 2003: Proceedings*, Lecture Notes in Computer Science 2904, pp. 280–294, 2003.

[CSD05] J. Castellà-Roca, F. Sebé, and J. Domingo-Ferrer. Dropout-Tolerant TTP-Free Mental Poker. In *Proceedings of TrustBus 2005*, Lecture Notes in Computer Science 3592, pp. 30–40, 2005.

[CDS94] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *Advances in Cryptology: CRYPTO '94 Proceedings*, Lecture Notes in Computer Science 839, pp. 174–187, 1994.

[CK94] C. Crépeau and J. Kilian. Discreet Solitary Games. In *Advances in Cryptology: CRYPTO '93 Proceedings*, Lecture Notes in Computer Science 773, pp. 319–330, 1994.

[Co85] D. Coppersmith. Cheating at Mental Poker. In *Advances in Cryptology: CRYPTO '85 Proceedings*, Lecture Notes in Computer Science 218, pp. 104–107, 1985.

[CP93] D. Chaum and T.P. Pedersen. Wallet Databases with Observers. In *Advances in Cryptology: CRYPTO '92 Proceedings*, Lecture Notes in Computer Science 740, pp. 89–105, 1993.

[Cr87] C. Crépeau. A zero-knowledge poker protocol that achieves confidentiality of the players' strategy or how to achieve an electronic poker face. In *Advances in Cryptology: CRYPTO '86 Proceedings*, Lecture Notes in Computer Science 263, pp. 239–247, 1987.

[dBo90] B. den Boer. More efficient match-making and satisfiability: the five card trick. In *Advances in Cryptology: EUROCRYPT '89 Proceedings*, Lecture Notes in Computer Science 434, pp. 208–217, 1990.

[DF90] Y. Desmedt and Y. Frankel. Threshold Cryptosystems. In *Advances in Cryptology: CRYPTO '89 Proceedings*, Lecture Notes in Computer Science 435, pp. 307–315, 1990.

[Ed94] J. Edwards. Implementing Electronic Poker: A Practical Exercise in Zero-Knowledge Interactive Proofs. Master's thesis, University of Kentucky, 1994.

[El85] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985.

[FSF] Free Software Foundation, Inc. The GNU General Public License (GPL). Version 2. `http://www.gnu.org/copyleft/gpl.html`

[Fu04] J. Furukawa. Efficient, Verifiable Shuffle Decryption and Its Requirement of Unlinkability. In *Public Key Cryptography - PKC 2004 Proceedings*, Lecture Notes in Computer Science 2947, pp. 319–332, 2004.

[GMR98] R. Gennaro, D. Micciancio, and T. Rabin. An Efficient Non-Interactive Statistical Zero-Knowledge Proof System for Quasi-Safe Prime Products. 5th ACM Conference on Computer and Communication Security, pp. 67–72, 1998.

[GJKR99] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. The (In)security of Distributed Key Generation in DLOG-based Cryptosystems. In *Advances in Cryptology: EUROCRYPT '99 Proceedings*, Lecture Notes in Computer Science 1592, pp. 295–310, 1999.

[GM81] S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proceedings of STOC '82*, pp. 365–377, 1982.

[Gr03] J. Groth. A Verifiable Secret Shuffle of Homomorphic Encryptions. In *Public Key Cryptography - PKC 2003: Proceedings*, Lecture Notes in Computer Science 2567, pp. 145–160, 2003.

[Gr05] J. Groth. A Verifiable Secret Shuffle of Homomorphic Encryptions. Cryptology ePrint Archive, Report 2005/246.

[HS97] C. Hall and B. Schneier. Remote Electronic Gambling. 13th Annual Computer Security Applications Conference, ACM Press, pp. 227–230, 1997.

[ISPA] International Skat Players Association. International Skat and Tournament Order. `http://www.ispaworld.org/`

[KKO97] K. Kurosawa, Y. Katayama, and W. Ogata. Reshufflable and laziness tolerant mental card game protocol. *IEICE Transactions Fundamentals*, E00-A(1), 1997.

[KK04] T. Koshiba and K. Kurosawa. Short Exponent Diffie-Hellman Problems. In *Public Key Cryptography - PKC 2004 Proceedings*, Lecture Notes in Computer Science 2947, pp. 173–186, 2004.

[Ko95] P. Kocher. Cryptanalysis of Diffie-Hellman, RSA, DSS, and other cryptosystems using timing attacks. In *Advances in Cryptology: CRYPTO '95 Proceedings*, Lecture Notes in Computer Science 963, pp. 171–183, 1995.

[Li81] R.J. Lipton. How to Cheat at Mental Poker. Proceedings of the AMS Short Course in Cryptography, 1981.

[LL94] C.H. Lim and P.J. Lee. More Flexible Exponentiation with Precomputation. In *Advances in Cryptology: CRYPTO '94 Proceedings*, Lecture Notes in Computer Science 839, pp. 95–107, 1994.

[Ne01] C.A. Neff. A Verifiable Secret Shuffle and its Application to E-Voting. 8th ACM Conference on Computer and Communications Security, pp. 116–125, 2001.

[NR98] V. Niemi and A. Renvall. Secure Multiparty Computations Without Computers. *Theoretical Computer Science*, 191(1-2):173–183, 1998.

[OR93] J. Oikarinen and D. Reed. RFC 1459: Internet Relay Chat Protocol. Network Working Group, Request for Comments, May 1993.

[Pe91] T. Pedersen. A Threshold Cryptosystem without a Trusted Party. In *Advances in Cryptology: EUROCRYPTO '91 Proceedings*, Lecture Notes in Computer Science 547, pp. 522–526, 1991.

[Pi02] M. Pinna. A Secure Card Game. BA-Thesis, Gonwille & Caius College, University of Cambridge, May 2002.

[Ra78] M. Rabin. Digital signatures. In *Foundations of Secure Computation*, pp. 155–168, Academic Press, NY, 1978.

[Sc98] C. Schindelhauer. Toolbox for Mental Card Games. Technical Report A-98-14, University of Lübeck, 1998.

[SRA79] A. Shamir, R.L. Rivest, and L.M. Adleman. Mental Poker. Technical Report MIT-LCS-TM-125, Massachusetts Institute of Technology, February 1979.

[St01] A. Stiglic. Computations with a deck of cards. *Theoretical Computer Science*, 259(1-2):671–678, 2001.

[St05a] H. Stamer. LibTMCG. A general purpose library for secure card games. `http://savannah.nongnu.org/projects/libtmcg/`

[St05b] H. Stamer. SecureSkat. A cryptographically secure implementation of Skat. `http://savannah.nongnu.org/projects/secureskat/`

[TY98] Y. Tsiounis and M. Yung. On the security of ElGamal-based encryption. In *Public Key Cryptography - PKC 1998: Proceedings*, Lecture Notes in Computer Science 1431, pp. 117–134, 1998.

[Wi05] D. Wikström. A Sender Verifiable Mix-Net and a New Proof of a Shuffle. Cryptology ePrint Archive, Report 2005/137.

[Yu85] M. Yung. Cryptoprotocols: Subscription to a public key, the secret blocking and the multi-player mental poker game. In *Advances in Cryptology: CRYPTO '84 Proceedings*, Lecture Notes in Computer Science 196, pp. 439–453, 1985.