3 Factorization with Quantum Computers: Shor's Algorithm

3.1 Classical Factorization Algorithms

The most famous classical factorization algorithms are the Quadratic Sieve (QS) and the Number Field Sieve (NFS). Though being subexponential, they are not polynomial. The QS is the fastest general-purpose factorization algorithm for numbers with less than 110 digits, whereas the NFS has the same property for numbers with more than 110 digits (see Schneier (1996), p.256). Recently, RSA-576 with a number of 174 decimal digits was factorized by Franke from the Universoty of Bonn with the aid of the NFS. The NFS was also used to factorize the Mersenne number $2^{757}-1$ (with 288 decimal digits) by the Internet project NESNET (about 5 months of computing time on up to 120 machines was necessary). Further limits on the factorization of large numbers can be found on the Internet site CiteSeer. For particular types of numbers to be factorized, many specially designed algorithms have been developed, which in these cases are faster than the above-mentioned ones. A new direction of cryptanalysis would be the possible use of quantum computers istead of classical Turing machines. Up to now, quantum computing has been more or less only a theoretical concept based on the superposition principle of quantum mechanics. Beyond some basic experiments, nobody has really an idea how to realize physical quantum computers working efficiently. However, if one day quantum computers could be built, this would have dramatic consequences for cryptology. Namely, in the second half of the 1990s, Peter Shor showed that on a quantum computer, large numbers can be factorized in linear (with respect to the length of the binary expansion of the number) time! So in this case, the RSA and all related systems would be worthless against an adversary who has a quantum computer at his disposal. Shor's method is in fact a hybrid algorithm in the sense that it consists of four components, one being done by quantum computing and three others (a little trick based on the Euclidean algorithm from elementary number theory, Fourier transform and continued-fraction approximation) that can be done on a classical computer. (Note that the Fourier transform component can, but need not be done on a quantum computer.) This algorithm will be explained in Section 3.4. We note that Shor has developed another algorithm for solving the discrete logarithm problem on quantum computers. Here, we will not discuss that, but the principles are similar.

D. Neuenschwander: Prob. and Stat. Methods in Cryptology, LNCS 3028, pp. 37-45, 2004.

Note that in contrast to Chapter 13, where we will present the ideas of quantum cryptography, here we use quantum computers to cryptanalyze classical cryptosystems.
We remark that there is a new approach due to Hungerbühler, Struwe (2003), who suggest the heat flow as a cryptographic system that resists also attacks by quantum computers. It is based on the second principle of thermodynamics (increase of entropy). The evolution problem for the heat equation is a well-posed initial-value problem, which can be solved very precisely by numerical methods, whereas the evolution problem in backward time is ill-posed and numerical methods for solving the heat equation for negative time are inherently unstable.

3.2 Quantum Computing

Let us now give a short introduction to quantum computing, which rests on a non-Kolmogorovian type of probability, namely quantum stochastics. In the following, we will present some basic facts on quantum mechanics and quantum computing. In quantum physics, the state of a quantum system is described by a vector in a (complex) Hilbert space H. It is customary to write such a state vector as a column vector, or - in the jargon of quantum physics - as "*ket* vector" $|\psi\rangle$. The corresponding line vector is written as $\langle\psi|$ and called "*bra* vector". The squared norm of the vector, or - in other words - the scalar product of the vector with itself is then written as $\langle\psi|\psi\rangle$, which becomes a *bracket*. We now come to the process of measurement in quantum mechanics. As a principle, in quantum mechanics measurements of observables are described by Hermitian operators A acting on the underlying Hilbert space H. If the system is in an eigenstate of A, then the measurement with the operator A just reproduces the state, multiplied with a real number (since A is Hermitean). If the system is not in an eigenstate, then the outcome of the measurement will collapse to one of the observables (corresponding to eigenstates (eigenvalues)) of A, but what is important is that it cannot be predicted in advance to which one. Only probabilites can be indicated, which correspond to the principle of superposition. The result of any measurement of a quantum system described by the state vector $|\psi\rangle$ is always one of the eigenvalues of the operator A, corresponding to the observable being measured. If the system is in an eigenstate of A, then the outcome of the measurement is just the corresponding eigenvalue of this eigenstate. In general, the system will be in some general state ϕ. Then we may represent ϕ as a complex linear combination with respect to a basis $\{\psi_i\}_i$ of eigenstates of A:

$$|\phi\rangle = \sum_i \omega_i |\psi_i\rangle, \tag{3.1}$$

where the ω_i are called the probability amplitudes. If w.l.o.g. we assume that $\sum_i |\omega_i|^2 = 1$, then $|\omega_i|^2$ is interpreted as the probability that the system is in