DISS ETH NO. 20264

SECURITY PROOFS IN NUMBER THEORETIC CRYPTOGRAPHY

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Sciences

presented by

DIVESH AGGARWAL
M.Sc. ETH in Computer Science

born on 29.04.1984

citizen of India

accepted on the recommendation of

Prof. Dr. Ueli Maurer
Prof. Dr. David Basin
Prof. Dr. Dennis Hofheinz

2012

Abstract

Public-key cryptography, also known as asymmetric cryptography, allows two users to communicate securely over an authenticated channel. Unlike symmetric key encryption schemes, public-key encryption schemes do not require an initial exchange of secret keys between the sender and the receiver.

Compared to symmetric cryptosystems, public-key encryption schemes are much harder to design and they rely on specific computational intractability assumptions. Most of these schemes have been based on the hardness of problems in broadly three categories: problems related to factoring integers, problems related to computing discrete logarithms, and more recently various problems in lattices. Since each has its own advantages and disadvantages, cryptosystems are designed based on each of these problems. The true computational complexity of any of these problems is not known. When complexity-theoretic (relative) lower bounds for certain cryptographic problems in a general model of computation seem to elude discovery, a common practice in cryptography is to give proofs of computational security in meaningful restricted models of computation.

The thesis aims at making progress towards determinig the (relation between the) computational complexity of some of these problems. It comprises of the following results.

- We prove that the problem of factoring N which is a product of two primes can be efficiently reduced to solving the generalized RSA problem on $\mathbb{Z}_N$ in the generic ring model of computation, where an algorithm can perform ring operations, inverse ring operations, and test equality. This provides evidence towards the soundness of the RSA encryption and digital signature scheme, in particular showing that under the factoring assumption, they are not vulnerable to certain kinds of cryptanalytic attacks.

  Further, we generalize the above result from RSA to the Strong RSA problem, i.e., we prove that for almost all possible distributions of N (which includes all N relevant in practice), the problem of factoring N can be efficiently reduced to solving the Strong RSA problem on $\mathbb{Z}_N$ in the generic ring model of computation.
- We look at the result of [Sha93] that showed that we can design public-key cryptosystems that are based on multivariate polynomials modulo N. However, we discovered that there is an error in their proof. We point out the exact error in the proof, and identify the conjecture that one needs to resolve to prove the result (assuming it is correct).
- The oracle complexity of a computational search problem is the minimum number of (adaptive) binary queries required to an infinitely powerful oracle, so that the problem can be solved efficiently. Motivated by a comparison between two search problems that are closely related with respect to the algorithms for solving

these problems: factoring integers and discrete logarithms modulo p, we looked at a characerization of the oracle complexity of a problem. We showed that the oracle complexity is (almost) equal to the negative logarithm of the maximum possible success probability in solving the problem by a probabilistic polynomial-time (PPT) algorithm.

This demonstrates, for the first time, the practical relevance of studying PPT algorithms even for problems believed to be hard, and even if the success probability is too small to be of practical interest. With this view, we give probabilistic polynomial time algorithms with the best possible success probability for the learning with errors and lattice problems.

- We improve the bound on the limit to inapproximability for the unique shortest vector problem in lattices. We show that if $\mathtt{GapSVP}_{\gamma} \in$ co-**NP** (or co-**AM**) then $\mathtt{uSVP}_{\sqrt{\gamma}} \in$ co-**NP** (co-**AM** respectively). This improves previously known results from $\mathtt{uSVP}_{n^{1/4}} \in \mathbf{NP} \cap$ co-**AM** to $\mathtt{uSVP}_{(n/\log n)^{1/4}} \in \mathbf{NP} \cap$ co-**AM**, and from $\mathtt{uSVP}_{n^{1/2}} \in \mathbf{NP} \cap$ co-**NP** to $\mathtt{uSVP}_{n^{1/4}} \in \mathbf{NP} \cap$ co-**NP**.

Zusammenfassung

Public-Key-Kryptographie, auch bekannt als asymmetrische Kryptographie, ermöglich es zwei Parteien, sicher über einen authentifizierten Kanal zu kommunizieren. Im Gegensatz zu symmetrischen Verschlüsselungsverfahren benötigen Public-Key-Verschlüsselungsverfahren keinen vorausgehenden Austausch von geheimen Schlüsseln zwischen dem Sender und dem Empfänger.

Im Vergleich zu symmetrischen Verfahren sind Public-Key-Verschlüsselungsverfahren deutlich schwieriger zu konstruieren und beruhen auf Annahmen der Art, dass bestimmte Probleme schwierig sind, das heisst nicht effizient gelöst werden können. Die Probleme, auf denen die meisten dieser Verfahren beruhen, kann man grob in drei Kategorien einteilen: Probleme aus dem Bereich des Faktorisierens von ganzen Zahlen, Probleme aus dem Bereich der Berechnung von diskreten Logarithmen und neuerdings auch verschiedene Probleme aus dem Bereich der Gitter (*Lattices*). Da jedes Problem spezifische Vor- und Nachteile hat, werden basierend auf jedem dieser Probleme kryptographische Verfahren konstruiert. Die genaue Berechnungskomplexität ist allerdings bei keinem dieser Probleme bekannt. Komplexitätstheoretische (relative) untere Schranken für bestimmte kryptographische Probleme in einem allgemeinen Berechnungsmodell scheinen zur Zeit unerreichbar zu sein, und es ist deshalb in der Kryptographie üblich, die berechenmässige Sicherheit in sinnvollen, eingeschränkten Berechnungsmodellen zu beweisen.

Ziel dieser Doktorarbeit ist es, einen Fortschritt in die Richtung zu erzielen, die Berechnungskomplexität von einigen dieser Probleme und die Beziehung zwischen den Komplexitäten zu bestimmen. Sie umfasst die folgenden Resultate:

- Wir beweisen, dass das Problem, N zu faktorisieren (wobei N ein Produkt zweier Primzahlen ist), im generischen Berechnungsmodell für Ringe effizient auf das verallgemeinerte RSA-Problem in $\mathbb{Z}_N$ reduziert werden kann. In diesem Berechnungsmodell kann ein Algorithmus Ring-Operationen, inverse Ring-Operationen, und Tests auf Gleichheit durchführen. Das gibt einen Hinweis auf die Sicherheit der RSA Verschlüsselung und digitaler Signaturverfahren. Insbesondere können diese Verfahren unter der Annahme, dass das Faktorisieren grosser Zahlen schwierig ist, durch bestimmte kryptoanalytische Attacken nicht gebrochen werden.

  Weiterhin verallgemeinern wir das obige Resultat vom RSA-Problem zum "strong" RSA-Problem, das heisst, wir beweisen, dass das Problem, N zu faktorisieren, für fast alle Verteilungen von N (was alle in der Praxis relevanten N einschliesst), im generischen Berechnungsmodell für Ringe effizient auf das Lösen des "strong" RSA-Problems in $\mathbb{Z}_N$ reduziert werden kann.
- Das Resultat von [Sha93] hat gezeigt, dass man Public-Key-Verfahren basierend auf mehrdimensionalen Polynomen modulo N konstruieren kann. Dieser Beweis ist jedoch fehlerhaft. Wir geben

eine präzise Beschreibung des Fehlers, und identifizieren eine hinreichende Vermutung, um das Resultat zu beweisen (angenommen, die Vermutung sei korrekt).

- Die Orakel-Komplexität eines Suchproblems ist die minimale Anzahl (adaptiver) binärer Anfragen an ein unendlich mächtiges Orakel, die benötigt werden, um das Problem effizient zu lösen. Motiviert durch den Vergleich zweier Suchproblemen, für die die bekannten Algorithmen eng verwandt sind, nämlich das Faktorisieren ganzer Zahlen und der diskrete Logarithmus modulo p, haben wir die Orakel-Komplexität von Problemen auf eine mögliche Charakterisierung untersucht. Wir haben bewiesen, dass die Orakel-Komplexität (fast) genau dem negativen Logarithmus der bestmöglichen Erfolgswahrscheinlichkeit entspricht, mit der ein probabilistischer Polynomialzeit-Algorithmus (PPT) das Problem lösen kann.

  Das beweist zum ersten Mal die praktische Relevanz der Untersuchung von PPT Algorithmen auch für Probleme, von denen angenommen wird, dass sie schwierig sind; selbst wenn die Erfolgswahrscheinlichkeit zu gering ist, um für die Praxis relevant zu sein. Vor diesem Hintergrund bechreiben wir PPT Algorithmen mit der bestmöglichen Erfolgswahrscheinlichkeit für Probleme im Bereich von Learning-with-errors und Gittern.

- Wir verbessern die Schranke an die Nicht-Approximierbarkeit des eindeutigen, kürzesten Vektors in einem Gitter. Wir zeigen, dass wenn $\mathtt{GapSVP}_{\gamma} \in$ co-**NP** (oder co-**AM**) dann $\mathtt{uSVP}_{\sqrt{\gamma}} \in$ co-**NP** (beziehungsweise co-**AM**). Dadurch verbessern wir bekannte Resultate von $\mathtt{uSVP}_{n^{1/4}} \in \mathbf{NP} \cap$ co-**AM** zu $\mathtt{uSVP}_{(n/\log n)^{1/4}} \in \mathbf{NP} \cap$ co-**AM**, und von $\mathtt{uSVP}_{n^{1/2}} \in \mathbf{NP} \cap$ co-**NP** zu $\mathtt{uSVP}_{n^{1/4}} \in \mathbf{NP} \cap$ co-**NP**.