Tutorial

LNCS 3028

Daniel Neuenschwander

Probabilistic and Statistical Methods in Cryptology

An Introduction by Selected Topics

$$|P_X(A(x^{(i-1)}) = x_i) - P_M(A(x^{(i-1)}) = x_i)| = O(\nu(n))$$

Springer

Lecture Notes in Computer Science 3028

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Daniel Neuenschwander

Probabilistic and Statistical Methods in Cryptology

An Introduction by Selected Topics

Springer

Author

Daniel Neuenschwander
Universities of Bern and Lausanne (Switzerland) and
Swiss Ministry of Defense
Section of Cryptology
3003 Bern, Switzerland
E-mail: daniel.neuenschwander@bluewin.ch

Library of Congress Control Number: 2004105111

CR Subject Classification (1998): E.3, G.3

ISSN 0302-9743
ISBN 3-540-22001-1 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Boller Mediendesign
Printed on acid-free paper SPIN: 10998649 06/3142 5 4 3 2 1 0

To Galina

Preface

Cryptology is nowadays one of the most important subjects of applied mathematics. Not only the task of keeping information secret is important, but also the problems of integrity and of authenticity, i.e., one wants to avoid that an adversary can change the message into a fraudulent one without the receiver noticing it, and on the other hand the receiver of a message should be able to be sure that the latter has really been sent by the authorized person (electronic signature). A big impetus on modern cryptology was the invention of so-called public-key cryptosystems in the 1970's by Diffie, Hellman, Rivest, Shamir, Adleman, and others. In particular in this context, deep methods from number theory and algebra began to play a decisive role. This aspect of cryptology is explained in, for example, the monograph "Algebraic Aspects of Cryptography" by Koblitz (1999). The goal of these notes was to write a treatment focusing rather on the stochastic (i.e., probabilistic and statistical) aspects of cryptology. As this direction also consists of a huge literature, only some glimpses can be given, and by no means are we always at the frontier of the current research. The book is rather intended as an invitation for students, researchers, and practitioners to study certain subjects further. We have tried to be as self-contained as reasonably possible, however we suppose that the reader is familiar with some fundamental notions of probability and statistics. It is our hope that we have been able to communicate the fascination of the subject and we would be delighted if the book encouraged further theoretical and practical research.

Let me give my gratitude to my colleagues in the Cryptology Section in the Ministry of Defense of Switzerland for the excellent and stimulating working atmosphere. Many thanks are also due to Werner Schindler from the German "Bundesamt für Sicherheit in der Informationstechnik" for helpful discussions. Furthermore, I am indebted to Springer-Verlag, Heidelberg for the agreeable cooperation. However, the most important thanks goes to my wife Galina for her constant moral support of my scientific activities. Without her asking "How is your book?" from time to time, the latter would certainly not yet be finished!

Bern, February 2004 Daniel Neuenschwander

Contents

Introduction

Background

Cryptology is nowadays considered as one of the most important fields of applied mathematics. Also, aspects from physics and, of course, engineering science play important roles. Classical cryptology consisted almost entirely of the problem of secret keeping. The so-called "Caesar shift code" was just a shift of the alphabet by a certain number of places, e.g., 3 places (then the plaintextletter "a" was encrypted by the ciphertextletter "D", "b" by "E", etc., "w" by "Z", and then "x" by "A", "y" by "B", "z" by "C"). Such a shift code is, of course, trivial to decrypt[1], because one needs to try only 25 possibilities with some groups of subsequent ciphertextletters until one obtains some meaningful plaintext. More general are monoalphabetic substitutions, which are just any permutation of the alphabet. Here, one has $26!-1 \approx 4\cdot 10^{26}$ possibilities, but as the same plaintextletter always corresponds to the same ciphertextletter and vice versa, frequent letters (or pairs/triples of letters) in the ciphertext will with great probability correspond to frequently occurring letters (pairs/triples) in the language in which the plaintext is written, for example the letter "e" in German. For example, the following features of German language support the decryption of monoalphabetic encryptions: If in the ciphertext a triple of consecutive letters occurs several times, then there is a good chance that it corresponds to the plaintext triple "sch"; the plaintext letter "c" is almost always succeeded by "h" or "k", "q" by "u" with hardly any exceptions. In any language (and also with more general cryptosystems) the encryptor should avoid the use of "mots probables" (words from which an adversary can conjecture that they appear in the plaintext, e.g., military terms, "Heil Hitler", etc.). During the Second World War, this danger was often neglected, a mistake that was not the most important, but one of several reasons why enemy codes were decrypted in a decisive measure at that time. In recent years, many documents have been (and still are) found by historians in archives which confirm this fact. In the year 1586, the French diplomat Blaise de Vigenère (1523-1596) found a polyalphabetic code that

[1] In all our subsequent text, the word "decipher" will mean the decoding of a ciphertext by its legitimate receiver, whereas "decrypt" will mean the breaking of the code by an adversary.

D. Neuenschwander: Prob. and Stat. Methods in Cryptology, LNCS 3028, pp. 1-7, 2004.

was thought to be "unbreakable" for centuries. This code will be presented in Section 1.1 of our text, together with the attacks on it found not earlier than in the second half of the 19th and at the beginning of the 20th century. After the spectacular successes in decrypting rotor enciphering machines such as ENIGMA, etc., during the Second World War, in the second half of the 1970s a great impetus on the development of modern cryptology was given by the invention of so-called public-key cryptosystems, in particular the code that is now known under the name "RSA system" (named after the authors who published it, namely "R" for Rivest, "S" for Shamir, and "A" for Adleman). Its detailed working is described in Section 2.1. The only non-trivial ingredient is Fermat's Little Theorem, which was known as a piece of "pure" number theory long before. It turned out since then that number theory and algebra are of decisive importance in modern cryptology, both in cryptography and cryptanalysis, in contrast to the assertion of the English mathematician G. Hardy (1877-1947) that by analyzing primes one "can not win wars"!

Nowadays, not only (classical) algebra and number theory, but also many other fields of mathematics, such as highly advanced topics of algebra and number theory (such as, for example, modern algebraic geometry, elliptic curves), graph theory, finite geometry (see, for example, Walther (1999)), probability, statistics, etc., play a role in cryptography, not to mention the recent (at least theoretical) developments in quantum computing and quantum cryptography (based on quantum mechanics) and all questions on hardware implementation of cryptosystems.

Furthermore, other goals entered into cryptology, namely the task of securization of the integrity and authenticity of a message. This means that (even for a possibly open transmission channel) one wants to avoid the message being changed by some unauthorized person without the receiver noticing it, and, on the other hand, the receiver wants to be sure that really the authorized person was the sender of the message (electronic signature). (In this context, we also mention the (however, already old) concept of steganography, where even the mere fact that a message has been transmitted (not only its contents) is to be kept secret. We will not discuss this subject further.) On the other hand, generalizations to multiparty systems also emerged. Nowadays, network security is a very important problem in practice.

A systematic introduction to the algebraic and number theoretic aspects was given in the Koblitz (1999) book "Algebraic Aspects of Cryptography". The goal of our text will be to give a similar insight into some probabilistic and statistical methods (in its broadest sense, so, for example, also using quantum stochastics) of cryptology. By no means do we claim completeness, only some introductions to certain topics can be given. Important areas, such as for example secret sharing, multi-party systems, zero-knowledge, problems on information transmission channels, linear cryptanalysis, digital fingerprinting, visual cryptography (see, for example, de Bonis, de Santis (2001)), etc., had to be (almost) entirely excluded. For further reading, we recommend that readers consult, in particular, the *Journal of Cryptology* and the various conference proceedings series, e.g., in the Springer Lecture Notes in Computer Science (EUROCRYPT, CRYPTO, ASIACRYPT, AUSCRYPT, INDOCRYPT, FAST SOFTWARE ENCRYPTION, etc.). What is also of interest are the journals *Designs, Codes, and Cryptography*, and *IEEE Transactions on Information Theory*, together with several "computational" periodicals. Sometimes, very important information can also be found in mathematical and stochastic journals/books, though this is rather the exception compared to the specific series devoted more to what is nowadays called "Theoretical Computer Science".

Book Structure

Let us now give a short description of the contents of the present book.
As already mentioned, in Section 1.1 we present the famous classical Vigenère system, which for a long time was believed to be as "secure as possible". Of course, no cryptosystem is absolutely secure in the literal sense of the word, since there is always the possibility of exhaustive search (in many cases, even though no better attack is known, however, also no *proof* that no better attack exists is available up to now). (Somewhat exceptional is quantum cryptography as it is briefly described in Chapter 13. But this is research in progress.) So actually the mere reasonable definition of "security" of a cryptosystem is a non-trivial task. In Section 1.2 we speak about the most natural (but expensive to realize) notion of "perfect secrecy", whereas other security concepts (weaker, but often more easily implementable and testable ones) are discussed in Sections 5.1 (Golomb's conditions, PN-sequences), 5.3 ("perfect pseudo-randomness", which means that a source cannot "efficiently" be distinguished from a truly random source), 5.4 (("almost") ideal local statistics), Chapter 10 ("semantic security", which is a "polynomially bounded" version of perfect secrecy in the sense that one assumes that the adversary has only "polynomial" computational resources), and Chapter 11 ("algorithmic complexity"). Of course, theoretically quite weak but in practice not unimportant is the requirement for maximal linear complexity (see Sections 5.1 and 7.11), if one confines oneself to linear feedback shift registers. A short remark follows about a misleading "intuitive" idea concerning cascade ciphers, against which Massey and Maurer (1993) warned in their paper "Cascade Ciphers: The Importance of Being First".
Chapter 2 is devoted to public-key ciphers, in particular to the RSA system. After the introduction of the RSA system, whose basis is the (probably true and therefore generally supposed) computational difficulty of factoring large integers, we present two of the best-known probabilistic primality tests (the Soloway-Strassen test, which, loosely speaking, tests Euler's criterion for the Legendre-Jacobi symbol, and the Rabin test, which is related to Fermat's

Little Theorem for residue rings modulo a prime). A specially designed probabilistic prime number test for numbers congruent 3 (mod.4) (i.e., candidates for prime factors of so-called Blum integers) has been presented by Müller (2003). In Section 2.4 we prove that in the RSA system, one has a "hard" least significant bit, which means that if ever one finds a probabilistic polynomial time algorithm for calculating the least significant bit of the plaintext from the public key and the ciphertext, then there exists also a probabilistic polynomial-time algorithm for reconstructing the whole plaintext from these data. "Hard bits" have been the subject of much subsequent literature. Another public-key algorithm, the Diffie-Hellman system, will be discussed in Chapter 8. Section 2.5 warns against careless hardware implementation, so that certain internal parameters (e.g., processing time) can be measured by the adversary, and advises on avoiding such attacks. For further reading about the subject of "timing attacks", we also refer to Schindler (2002a). In Section 2.6 we show how somebody can persuade his/her friend that he/she has found an RSA-secret key of somebody else without revealing any information about it, thus giving a first glimpse into the field of zero-knowledge proofs.

Chapter 3 presents Shor's algorithm (for whose invention Shor got the Nevanlinna prize) for factoring numbers with quantum computers. One must admit that up to now, quantum computers have been rather a theoretical concept and not yet producible in a usable way. The latest news about hardware research in this direction is rather pessimistic. Of course, from the viewpoint of users of classical cryptological devices this is reassuring, for if an adversary were really in possession of a quantum computer working on a large scale, then virtually all cryptosystems whose security is based on the "intractability" of the problem of factorizing numbers or the discrete logarithm problem would be breakable in "no" time (more precisely: in linear time, where up to now only behavior (e.g., for the quadratic or the number field sieve) of an order little better than exponential is known). We do not assume that the reader has any preliminary knowledge of quantum theory. All necessary explanations are given in Section 3.2. Shor's algorithm makes use of a result from the theory of continued fractions, which we will present in Section 3.3.

Almost all cryptosystems work with keys, which, as a doctrine (at least in theoretical cryptology), is the only information on the cryptosystem that is assumed to (and can realistically) be kept secret. That is, one always assumes, in order to be on the safe side, that the adversary is in possession of the device that has been used for encryption/deciphering, but he has virtually no information about the key. The most secure way to provide a good key is to generate it with a genuine, physical generator, e.g., radioactive sources with Geiger counters or electronic noise produced by a semiconducting diode (see Chapter 4). For general use, for example, HOT BITS is a source of random bits stemming from beta radiation from the decay of krypton-85, and is available on the Internet. However, physical devices are very slow compared to pseudo-random generators, which we will treat in Chapter 5. Some considerations about possible constructions of good physical random number generators, such as some discussions on their quality due to Zeuner and the author, are the subject of Section 4.2. In Section 4.3 we address the general problem of obtaining random bits that are as unbiased as possible, if the disposable source only produces random bits with a certain bias. We will calculate the "extraction rate" (which indicates in some sense the asymptotical speed of the diminution of the bias per new random bit source, when the final output bit is produced by adding (mod.2) independent biased random bit sources) for rational biases. Interestingly enough, the extraction rate turns out to be independent of the size of the bias b, but to be determined solely by the arithmetic properties of b. However, one finds that the extraction rate is 0 for Lebesgue-almost all biases b.

On the contrary, we speak about pseudo-random generators in the following. In Chapter 5, we present some important examples (linear feedback shift registers (Section 5.1) and combinations thereof (Section 5.5), non-linear feedback shift registers (Section 5.4), shrinking and self-shrinking generators (Section 5.2), and the quadratic congruential generator (Section 5.6)).

Chapter 6 is a brief introduction to the most important notions of information theory as it is of use for us and to the aforementioned problem of authenticity. Section 6.3 is a new unorthodox approach.

In Chapter 7 we give a collection of some of the best-known tests for pseudo-random-number generators, orienting ourselves to a great extent at the tests suggested by Rukhin (2000a,b) and the test-battery used for evaluation of the AES. As is well-known, for a long time, the block cipher "data encryption standard" (DES) has been widely used, but, by using parallelism, it has been possible to break it. Then the NIST (National Institute of Standards and Technology) invited the worldwide cryptologic community to develop an "advanced encryption standard" (AES). The winner of this contest was the algorithm RIJNDAEL designed by Rijmen and Daemen.

Chapter 8 discusses the distribution of keys in the Diffie-Hellman public-key system. In this context, the notion of "strong primes" (primes p that are of the form $p = 2q + 1$ (where q is a prime)) is useful. Namely, it turns out that if the modulus is a strong prime, then the entropy of the Diffie-Hellman key is nearly the maximum possible, which means that it is recommendable to use strong primes as moduli. Similar considerations about bit security as we have in Section 2.4 apply for the Diffie-Hellman system, too. We refer to González Vasco, Shparlinski (2001).

Chapter 9 describes an attack on block ciphers that has become very popular in recent years, namely differential cryptanalysis. Roughly speaking, here the cryptanalyst makes use of cases where "differences/sums" (in the algebraic sense) of pairs of plaintexts leak through to differences/sums of the corresponding pairs of ciphertexts. In an iterative r-round block cipher, with this method it is sometimes possible to guess the r-th round subkey, then the

$(r-1)$-th round subkey, etc., iteratively until the whole key is found. Interestingly enough, although the theoretical results are generally proved under the assumption that the round keys are chosen as i.i.d. (independent and identically distributed), in practice they are experimentally verified (sometimes with even better behavior) if some key schedule algorithm is used. Section 9.2 generalizes distributional results for so-called characteristics (i.e., pairs of differences of plaintext/ciphertext pairs of bitstrings) due to Hawkes and O'Connor to residue rings of arbitrary modulus. Matsui (1994) developed the related concept of linear cryptanalysis, which we have excluded from our presentation.

In Chapter 10 we deal with semantic security. Roughly speaking, semantic security is a polynomially bounded variant of perfect security, i.e., one assumes that the adversary has only polynomially bounded resources.

A notion of "algorithmic complexity" (the so-called "Turing-Kolmogorov-Chaitin complexity", which is — roughly speaking — the length of the shortest program that one must feed to a universal Turing machine to generate as output a given bitstring) is considered in Chapter 11. However, this is of rather theoretical interest, since the algorithmic complexity of a given bitstring is not computable (in the sense of the Church Thesis). It turns out that in the sense of the Haar measure, for almost all bitstrings the algorithmic complexity is equal to the linear complexity, thus here we have a somewhat similar situation as for the extraction rate of biases in Section 4.3. At first glance this contradicts the fact that there are very simply constructed bitsequences with maximal linear complexity (e.g., 00...01), but the abovementioned equivalence is not valid for "effectively constructible" sequences (see the title of the paper of Beth and Dai (1990): "If you can describe a sequence, it can't be random.").

Chapter 12 addresses the problem of collisions and the related "meet-in-the-middle" attack, which has to do with the well-known birthday paradox from probability theory.

Finally, we give a short glimpse into quantum cryptography in Chapter 13. In this situation, the receiver of an encrypted message will immediately detect (with arbitrarily large probability) if an adversary has manipulated the message (maybe even only "measured" it in the quantum-mechanical sense), which in general is of course not the case in classical cryptosystems. However, here also, the technology has not yet been developed far enough. Note that Chapter 13 deals with "genuine" quantum cryptography, whereas in Chapter 3 we showed how to solve a problem of classical cryptography by means of quantum computing.

Finally, a word about giving proper credits should be said: In cryptology, it is even more difficult than in other sciences to know to whom a certain result should really be attributed, since often methods that have been published later have already been developed (at least to a certain extent) before by cryptologists who were not allowed to publish their findings, especially during the time of the Second World War and the Cold War. So, citations of literature in our text should hardly be interpreted as a reference giving a credit to a certain person or group of persons. For example, one sees few Russian names occurring in the cryptological literature however, it turned out that Soviet cryptanalysts have had important successes in, for example, cryptanalysis, too.

In the body of this book, we give few formal citations, in order not to interrupt the smoothness of the presentation too much. Instead, we have included a section "Bibliographical Remarks" at the end of the text.

Chapters and sections with an asterisk treat more specific subjects and can be omitted at first reading.

About Notation and Terminology

Throughout the book, the symbol $\mathbb{B}$ will denote $GF(2) = \mathbb{Z}_2$, the field with the two elements 0 and 1, which will be called "bits" (exception: Section 4.3). Also, for a sequence $x = (x_1, x_2, \ldots)$, the symbol $x^{(n)}$ will mean the finite subsequence consisting of the first n elements: $x^{(n)} = (x_1, x_2, \ldots, x_n)$. The indicator function of the set B will be written as $\mathbf{1}(B)(.)$.

"W.l.o.g." means "without loss of generality". The shorthands "i.i.d." and "a.s." stand for the probabilistic notions "independent and identically distributed" and "almost surely" (i.e., "with probability one"). As already mentioned in the footnote at the beginning, the word "decipher" will mean the decoding of a ciphertext by its legitimate receiver, whereas "decrypt" is the breaking of the code by an adversary.

1 Classical Polyalphabetic Substitution Ciphers

1.1 The Vigenère Cipher

The classical situation in cryptology, which we will consider below, is the following: There are two parties, A (called "Alice" in the jargon) and B (called "Bob"). Alice would like to send a message to Bob by some channel. But this channel is unsecure because in-between the two, there is some adversary ("enemy", eavesdropper) E (called "Eve") who either wants

– to listen in on the message sent from A to B and/or
– to send a message herself to B, asserting that this message comes from A and/or
– to change a message indeed sent by A to B.

All these three attacks should be avoided. The first attack (listening in) concerns the problem of secrecy (or confidentiality), the second that of authenticity, and the third that of integrity. In other words, there are two independent goals: To reach secrecy resp. authenticity/integrity, the output resp. input of the channel from A to B should be exclusive. Of course, there are more general cryptologic situations (multi-party models, secret sharing, zero-knowledge, etc.). But these will not be considered here (except in the short Section 2.6). Also the integrity/authenticity problem will only be addressed in Sections 2.1 (electronic RSA signature) and 6.2 (impersonation attack), and Chapter 12 (meet-in-the-middle attack). Apart from that, in this introductory text we will mainly be concerned with secret keeping.
In this chapter, we will present a classical cryptosystem, the so-called Vigenère cipher, invented in 1586 by the French diplomat Blaise de Vigenère (1523-1596). It belongs to the class of polyalphabetic cryptosystems, which means that the same letter of plaintext is not always encoded by the same letter of ciphertext. This fact is of great importance in general. If a cryptosystem is monoalphabetic, i.e. if every letter of plaintext is always encrypted by the same letter of ciphertext, then statistical properties of the letters of the language in which the plaintext is written automatically leak through to the ciphertext, i.e. (for long enough messages) frequent letters (or m-grams) in the ciphertext correspond to frequent letters (or m-grams) in the plaintext, and by some statistical analysis it is, in general, not too difficult to find the

D. Neuenschwander: Prob. and Stat. Methods in Cryptology, LNCS 3028, pp. 9-15, 2004.

plain-/ciphertext correspondence of frequent letters (m-grams) of the language. To fill in the rest, often some "trial and error" helps (in particular with some additional information about "mots probables" (words that are likely to occur in the message)).
The Vigenère system is very simple and works as follows: Given a keyword, e.g., "PEACE" and the plaintext

OSAMABINLADEN,

then one writes the plaintext and the repeated keyword under each other and "adds" the corresponding letters mod.26 (where A is interpreted as 0, B as 1, etc.) to obtain the ciphertext:

Plaintext	O	S	A	M	A	B	I	N	L	A	D	E	N
Keyword	P	E	A	C	E	P	E	A	C	E	P	E	A
Ciphertext	D	W	A	O	E	Q	M	N	N	E	S	I	N

If Bob knows the key word, he can retrieve the plaintext from the ciphertext simply by subtracting the corresponding letters of the keyword mod. 26. But what cryptanalysis is concerned, one must say that although this system is polyalphabetic as such, always after k places (if k is the length of the keyword) the same substituting alphabet (which is even just a shift of the original alphabet in the sense of its interpretation as elements of $\mathbb{Z}_{26}$) is used. This gives rise to an algebraic method (the so-called Kasiski test) of determining the keyword length up to multiples. Together with the stochastic Friedman test, which yields the order of magnitude of the length of the keyword, one can determine in most cases the actual length of the keyword. If this is known, for every place modulo the length of the keyword, one must replace the letter of the ciphertext that occurs most frequently by some very frequent letter of the language in which the plaintext is written to determine the shift, and then with little routine work one can then (in general) reconstruct the plaintext thus. Let us describe the details: The Kasiski test is named after the Prussian major Friedrich Wilhelm Kasiski (1805-1881), although it had been found nine years before him (but had not been published) by Charles Babbage (1792-1871) in 1854. It rests on the following observation: If a certain word (for example a preposition or a conjunction, etc.) occurs several times in the plaintext and if by chance (which is often quite large) the distance between two such occurrences of the same word is a multiple of the length of the keyword, then this word is encoded both times by the same sequence of letters in the ciphertext. Or - spoken the other way round - if one detects the same subsequences of letters (maybe even short ones, e.g., of length 3) several times in the ciphertext, then the distance between them is quite probably a multiple of the keyword length. Now the second part will be a little more