DRAFT: The CrypTool Script: Cryptography, Mathematics and more
Background reading for the free eLearning program CrypTool (version 1.4.00)

Overview about the Content of the CrypTool Script

In this CrypTool script you will find predominantly mathematically oriented information on using cryptographic procedures. The main chapters have been written by various authors (see appendix A.2) and are therefore independent from one another. At the end of most chapters you will find literature and web links. The first chapter explains the principles of symmetric and asymmetric encryption and describes shortly the current decryption records of modern symmetric algorithms. Because of didactical reasons the second chapter gives an exhaustive overview about paper and pencil encryption methods. Big parts of this script are dedicated to the fascinating topic of prime numbers (chap. 3). Using numerous examples, modular arithmetic and elementary number theory (chap. 4) are introduced and applied in an exemplary manner for the RSA procedure. By reading chapter 5 you'll gain an insight into the mathematical ideas and concepts behind modern cryptography. Chapter 6 gives an overview about the status of attacks against modern hash algorithms and is then shortly devoted to digital signatures, which are an essential component of e-business applications. The last chapter describes elliptic curves: they could be used as an alternative to RSA and in addition are extremely well suited for implementation on smartcards. Whereas the eLearning program CrypTool motivates and teaches you how to use cryptography in practice, the script provides those interested in the subject with a deeper understanding of the mathematical algorithms used – trying to do it in an instructive way. If you are alrady a little bit familiar with this field of knowledge you can gain a fast overview about the functions delivered by CrypTool using the menu tree (see appendix A.1). The authors would like to take this opportunity to thank their colleagues in the company and at the universities of Frankfurt, Gießen, Siegen, Karlsruhe and Darmstadt. As with the eLearning program CrypTool, the quality of the script is enhanced by your suggestions and ideas for improvement. We look forward to your feedback. You will find the current version of CrypTool at the web sites http://www.cryptool.org, http://www.cryptool.com or http://www.cryptool.de. The contact people for this free open-source program are listed at the web site and in the “readme” file delivered within the CrypTool package.

Contents

5 The Mathematical Ideas behind Modern Cryptography 5.1 5.2

Preface to the 7th Edition of the CrypTool Script

Starting in the year 2000 this script became a part of the CrypTool package. It is designed to accompany the program CrypTool by explaining some mathematical topics in more detail, but still in a way which is easy to understand. In order to also enable developers/authors to work together independently the topics have been split up and for each topic an extra chapter has been written which can be read on its own. The later editorial work in TeX added cross linkages between different sections and footnotes describing where you can find the according functions within the CrypTool program (see menu tree in appendix A). Naturally there are many more interesting topics in mathematics and cryptography which could be discussed in greater depth – therefore this is only one of many ways to do it. The rapid spread of the Internet has also lead to intensified research in the technologies involved, especially within the area of cryptography where a good deal of new knowledge has arisen. This edition of the script adds some topics, but mainly updates areas (e.g. the summaries of topical research areas): • the search for the largest prime numbers (generalized Mersenne and Fermat primes) (chap. 3.6, 3.4.1), • the factorisation of big numbers (RSA-200) (chap. 4.11.4), • progress in cryptanalysis of hash algorithms (chap. 6.1.2) and • a list of movies or novels, in which cryptography or number theory played major role (see appendix A.3); and where primes are used as hangers (see curiouses in 3.8.4).

The first time the document was delivered with CrypTool was in version 1.2.01. Since then it has been expanded and revised in almost every new version of CrypTool (1.2.02, 1.3.00, 1.3.02, 1.3.03, 1.3.04 and now 1.4.00). I'd be more than happy if this also continues in the further open-source versions of CrypTool. I am deeply grateful to all the people helping with their impressive commitment who have made this global project so successful. Especially I would like to acknowledge the English language proof-reading of this script version done by Richard Christensen and Lowell Montgomery. I hope that many readers have fun with this script and that they get out of it more interest and greater understanding of this modern but also very ancient topic. Bernhard Esslinger Frankfurt (Germany), March 2006

Introduction – How do the Script and the Program Play together?
This script This document is delivered together with the program CrypTool. The articles in this script are largely self-contained and can also be read independently of CrypTool. Chapters 5 (Modern Cryptography) and 7 (Elliptic Curves) require a deeper knowledge in mathematics, while the other chapters should be understandable with a school leaving certificate. The authors have attempted to describe cryptography for a broad audience – without being mathematically incorrect. We believe that this didactical pretension is the best way to promote the awareness for IT security and the readiness to use standardised modern cryptography. The program CrypTool CrypTool is a program with an extremely comprehensive online help enabling you to use and analyse cryptographic procedures within a unified graphical user interface. CrypTool was developed during the end-user awareness program at Deutsche Bank in order to increase employee awareness of IT security and provide them with a deeper understanding of the term security. A further aim has been to enable users to understand the cryptographic procedures. In this way, using CrypTool as a reliable reference implementation of the various encryption procedures, you can test the encryption implemented in other programs. CrypTool is currently been used for training in companies and teaching at school and universities, and moreover several universities are helping to further develop the project. Acknowledgment At this point I'd like to thank explicitly six people who particularly contributed to CrypTool. Without their special talents and engagement CrypTool would not be what it is today: • Mr. Henrik Koy • Mr. J¨rg-Cornelius Schneider o • Dr. Peer Wichmann • Prof. Dr. Claudia Eckert, Mr. Thomas Buntrock and Mr. Thorsten Clausius. Also I want to thank all the many people not mentioned here for their hard work (mostly carried out in their spare time).
Bernhard Esslinger Frankfurt (Germany), March 2006

1

Encryption Procedures

(Bernhard Esslinger, May 1999, Updates Dec. 2001, Feb. 2003, June 2005) This chapter introduces the topic in a more descriptive way without using too much mathematics. The purpose of encryption is to change data in such a way that only an authorised recipient is able to reconstruct the plaintext. This allows us to transmit data without worrying about it getting into unauthorised hands. Authorised recipients possess a piece of secret information – called the key – which allows them to decrypt the data while it remains hidden from everyone else. One encryption procedure has been mathematically proved to be secure, the One Time Pad. However, this procedure has several practical disadvantages (the key used must be randomly selected and must be at least as long as the message being protected), which means that it is hardly used except in closed environments such as for the hot wire between Moscow and Washington. For all other procedures there is a (theoretical) possibility of breaking them. If the procedures are good, however, the time taken to break them is so long that it is practically impossible to do so, and these procedures can therefore be considered (practically) secure. The book of Bruce Schneier [Schneier1996] offers a very good overview of the different algorithms. We basically distinguish between symmetric and asymmetric encryption procedures.

1.1

Symmetric encryption[1]

For symmetric encryption sender and recipient must be in possession of a common (secret) key which they have exchanged before actually starting to communicate. The sender uses this key to encrypt the message and the recipient uses it to decrypt it. All classical methods are of this type. Examples can be found within the program CrypTool, in chapter 2 (Paper and Pencil Encryption Methods) of this script or in [Nichols1996]. Now we want to consider more modern mechanisms. The advantages of symmetric algorithms are the high speed with which data can be encrypted and decrypted. One disadvantage is the need for key management. In order to communicate with one another confidentially, sender and recipient must have exchanged a key using a secure channel before actually starting to communicate. Spontaneous communication between individuals who have never met therefore seems virtually impossible. If everyone wants to communicate with everyone else spontaneously at any time in a network of n subscribers, each subscriber must have previously exchanged a key with each of the other n − 1 subscribers. A total of n(n − 1)/2 keys must therefore be exchanged.

---

[1] With CrypTool you can execute the following modern symmetric encryption algorithms (using the menu path Crypt \ Symmetric (modern)): IDEA, RC2, RC4, DES (ECB), DES (CBC), Triple-DES (ECB), Triple-DES (CBC), MARS (AES candidate), RC6 (AES candidate), Serpent (AES candidate), Twofish (AES candidate), Rijndael (official AES algorithm).

The most well-known symmetric encryption procedure is the DES-algorithm. The DES-algorithm has been developed by IBM in collaboration with the National Security Agency (NSA), and was published as a standard in 1975. Despite the fact that the procedure is relatively old, no effective attack on it has yet been detected. The most effective way of attacking consists of testing (almost) all possible keys until the right one is found (brute-force-attack). Due to the relatively short key length of effectively 56 bits (64 bits, which however include 8 parity bits), numerous messages encrypted using DES have in the past been broken. Therefore, the procedure can now only be considered to be conditionally secure. Symmetric alternatives to the DES procedure include the IDEA or Triple DES algorithms. Up-to-the-minute procedure is the symmetric AES standard. The associated Rijndael algorithm was declared winner of the AES award on October 2nd, 2000 and thus succeedes the DES procedure. More details about the AES algorithms and the AES candidates of the last round can be found within the online help of CrypTool[2] .

1.1.1 New results about cryptanalysis of AES

Below you will find some results, which have recently called into question the security of the AES algorithm – from our point of view these doubts practically still remain unfounded . The following information is based on the original papers and the articles [Wobst-iX2002] and [Lucks-DuD2002]. AES with a minimum key length of 128 bit is still in the long run sufficiently secure against bruteforce attacks – as long as the quantum computers aren't powerful enough. When announced as new standard AES was immune against all known crypto attacks, mostly based on statistical considerations and earlier applied to DES: using pairs of clear and cipher texts expressions are constructed, which are not completely at random, so they allow conclusions to the used keys. These attacks required unrealistically large amounts of intercepted data. Cryptanalysts already label methods as "academic success" or as "cryptanalytic attack" if they are theoretically faster than the complete testing of all keys (brute force analysis). In the case of AES with the maximal key length (256 bit) exhaustive key search on average needs 2255 encryption operations. A cryptanalytic attack needs to be better than this. At present between 275 and 290 encryption operations are estimated to be performable only just for organizations, for example a security agency. In their 2001-paper Ferguson, Schroeppel and Whiting [Ferguson2001] presented a new method of symmetric codes cryptanalysis: They described AES with a closed formula (in the form of a continued fraction) which was possible because of the "relatively" clear structure of AES. This formula consists of around 1000 trillion terms of a sum - so it does not help concrete practical cryptanalysis. Nevertheless curiosity in the academic community was awakened. It was already known, that the 128-bit AES could be described as an over-determined system of about 8000 quadratic equations (over an algebraic number field) with about 1600 variables (some of them

---

[2] CrypTool online help: the index head-word AES leads to the 3 help pages: AES candidates, The AES winner Rijndael and The Rijndael encryption algorithm.

are the bits of the wanted key) – equation systems of that size are in practice not solvable. This special equation system is relatively sparse, so only very few of the quadratic terms (there are about 1,280,000 are possible quadratic terms in total) appear in the equation system. The mathematicians Courtois and Pieprzyk [Courtois2002] published a paper in 2002, which got a great deal of attention amongst the crypto community: The pair had further developed the XLmethod (eXtended Linearization), introduced at Eurocrypt 2000 by Shamir et al., to create the so called XSL-method (eXtended Sparse Linearization). The XL-method is a heuristic technique, which in some cases manages to solve big non-linear equation systems and which was till then used to analyze an asymmetric algorithm (HFE). The innovation of Courtois and Pieprzyk was, to apply the XL-method on symmetric codes: the XSL-method can be applied to very specific equation systems. A 256-bit AES could be attacked in roughly 2230 steps. This is still a purely academic attack, but also a direction pointer for a complete class of block ciphers. The major problem with this attack is that until now nobody has worked out, under what conditions it is successful: the authors specify in their paper necessary conditions, but it is not known, which conditions are sufficient. There are two very new aspects of this attack: firstly this attack is not based on statistics but on algebra. So attacks seem to be possible, where only very small amounts of cipher text are available. Secondly the security of a product algorithm[3] does not exponentially increase with the number of rounds. Currently there is a large amount of research in this area: for example Murphy and Robshaw presented a paper at Crypto 2002 [Robshaw2002a], which could dramatically improve cryptanalysis: the burden for a 128-bit key was estimated at about 2100 steps by describing AES as a special case of an algorithm called BES (Big Encryption System), which has an especially ”round” structure. But even 2100 steps are beyond what is achievable in the foreseeable future. Using a 256 bit key the authors estimate that a XSL-attack will require 2200 operations. More details can be found at: http://www.cryptosystem.net/aes http://www.minrank.org/aes/ So for 256-AES the attack is much more effective than brute-force but still far more away from any computing power which could be accessible in the short-to-long term. The discussion is very controversial at the moment: Don Coppersmith (one of the inventors of DES) for example queries the practicability of the attack because XLS would provide no solution for AES [Coppersmith2002]. This implies that then the optimization of Murphy and Robshaw [Robshaw2002b] would not work.

---

[3] A cipher text can be used as input for another encryption algorithm. A cascade cipher is build up as a composition of different encryption transformations. The overall cipher is called product algorithm or cascade cipher (sometimes depending whether the used keys are statistically dependent or not). Cascading does not always improve the security. This process is also used within modern algorithms: They usually combine simple and, considered at its own, cryptologically relatively unsecure single steps in several rounds into an efficient overall procedure. Most block ciphers (e.g. DES, IDEA) are cascade ciphers. Also serial usage of the same cipher with different keys (like with Triple-DES) is called cascade cipher.

1.1.2

Current status of brute-force attacks on symmetric algorithms (RC5)

The current status of brute-force attacks on symmetric encryption algorithms can be explained with the block cipher RC5. Brute-force (exhaustive search, trial-and-error) means to completely examine all keys of the key space: so no special analysis methods have to be used. Instead, the cipher text is decrypted with all possible keys and for each resulting text it is checked, whether this is a meaningful clear text. A key length of 64 bit means at most 264 = 18,446,744,073,709,551,616 or about 18 trillion (GB) / 18 quintillion (US) keys to check[4] . Companies like RSA Security provide so-called cipher challenges in order to quantify the security offered by well-known symmetric ciphers as DES, Triple-DES or RC5[5] . They offer prizes for those who manage to decipher cipher texts, encrypted with different algorithms and different key lengths, and to unveil the symmetric key (under controlled conditions). So theoretical estimates can be confirmed. It is well-known, that the “old” standard algorithm DES with a fixed key length of 56 bit is no more secure: this was demonstrated already in January 1999 by the Electronic Frontier Foundation (EFF). With their specialized computer Deep Crack they cracked a DES encrypted message within less than a day[6] . The current record for strong symmetric algorithms unveiled a key 64 bit long. The algorithm used was RC5, a block cipher with variable key size. The RC5-64 challenge has been solved by the distributed.net team after 5 years[7] . In total 331,252 individuals co-operated over the internet to find the key. More than 15 trillion (GB) / 15 quintillion (US) keys were checked, until they found the right key. This makes clear, that symmetric algorithms (even if they have no cryptographical weakness) using keys of size 64 bit are no more appropriate to keep sensible data private. Similar cipher challenges are there for asymmetric algorithms (please see chapter 4.11.4).

1.2

Asymmetric encryption[8]

In the case of asymmetric encryption each subscriber has a personal pair of keys consisting of a secret key and a public key. The public key, as its name implies, is made public, e.g. in a key directory on the Internet.

---

[4] With CrypTool you can also try brute-force attacks of modern symmetric algorithms (using the menu path Analysis \ Symmetric Encryption (modern)): here the weakest knowledge of an attacker is assumed, he performs a ciphertext-only attack. To achieve a result in an appropriate time with a single PC you should mark not more than 20 bit of the key as unknown.

[5] http://www.rsasecurity.com/rsalabs/challenges/secretkey/index.html

[6] http://www.rsasecurity.com/rsalabs/challenges/des3/index.html

[7] http://distributed.net/pressroom/news-20020926.html

[8] With CrypTool you can execute RSA encryption and decryption (using the menu path Crypt \ Asymmetric). In both cases you must select a RSA key pair. Only in the case of decryption the secret RSA key is necessary: so here you are asked to enter the PIN.

If Alice[9] wants to communicate with Bob, then she finds Bob's public key in the directory and uses it to encrypt her message to him. She then sends this cipher text to Bob, who is then able to decrypt it again using his secret key. As only Bob knows his secret key, only he can decrypt messages addressed to him. Even Alice who sends the message cannot restore plaintext from the (encrypted) message she has sent. Of course, you must first ensure that the public key cannot be used to derive the private key. Such a procedure can be demonstrated using a series of thief-proof letter boxes. If I have composed a message, I then look for the letter box of the recipient and post the letter through it. After that, I can no longer read or change the message myself, because only the legitimate recipient has the key for the letter box. The advantage of asymmetric procedures is the easy key management. Let's look again at a network with n subscribers. In order to ensure that each subscriber can establish an encrypted connection to each other subscriber, each subscriber must possess a pair of keys. We therefore need 2n keys or n pairs of keys. Furthermore, no secure channel is needed before messages are transmitted, because all the information required in order to communicate confidentially can be sent openly. In this case, you simply have to pay attention to the accuracy (integrity and authenticity) of the public key. Disadvantage: Pure asymmetric procedures take a lot longer to perform than symmetric ones. The most well-known asymmetric procedure is the RSA algorithm[10] , named after its developers Ronald Rivest, Adi Shamir and Leonard Adleman. The RSA algorithm was published in 1978. The concept of asymmetric encryption was first introduced by Whitfield Diffie and Martin Hellman in 1976. Today, the ElGamal procedures also play a decisive role, particularly the Schnorr variant in the DSA (Digital Signature Algorithm).

1.3

Hybrid procedures[11]

In order to benefit from the advantages of symmetric and asymmetric techniques together, hybrid procedures are usually used (for encryption) in practice. In this case the data is encrypted using symmetric procedures: the key is a session key generated by the sender randomly[12] that is only used for this message. This session key is then encrypted

In order to describe cryptographic protocols participants are often named Alice, Bob, . . . (see [Schneier1996, p. 23]). Alice and Bob perform all 2-person-protocols. Alice will initiate all protocols and Bob answers. The attackers are named Eve (eavesdropper) and Mallory (malicious active attacker). 10 The RSA algorithm is extensively described in chapter 4.10 and later within this script. The RSA cryptosystem can be executed in many variations with CrypTool (using the menu path Individual Procedures \ RSA Cryptosystem \ RSA Demonstration). The topical research results concerning RSA are described in chapter 4.11. 11 Within CrypTool you can get a visualization of this technique using the menu path Crypt \ Hybrid: this dialogue shows the single steps and its dependencies with concrete numbers. Here the asymmetric algorithm RSA and the symmetric algorithm AES are used. 12 An important part of cryptographically secure techniques is to generate random numbers. Within CrypTool you can check out different random number generators using the menu path Indiv. Procedures \ Generate Random Numbers. Using the menu path Analysis \ Analyse Randomness you can apply different test methods for

9

using the asymmetric procedure and transmitted to the recipient together with the message. Recipients can determine the session key using their secret keys and then use the session key to encrypt the message. In this way, we can benefit from the easy key management of asymmetric procedures and encrypt large quantities of data quickly and efficiently using symmetric procedures.

1.4

Further details

Beside the information you can find in the following chapters, many other books and on a good number of websites the online help of CrypTool also offers very many details about the symmetric and asymmetric encryption methods.

random data to binary documents. Up to now CrypTool has concentrated on cryptographically strong pseudo random number generators. Only the integrated Secude generator involves a ”pure” random source.

References

[Coppersmith2002] Don Coppersmith, Re: Impact of Courtois and Pieprzyk results, 2002-09-19, “AES Discussion Groups” at http://aes.nist.gov/aes/ [Courtois2002] Nicolas Courtois, Josef Pieprzyk, Cryptanalysis of Block Ciphers with Overdefined Systems of Equations, received 10 Apr 2002, last revised 9 Nov 2002. A different version, so called compact version of the first XSL attack, was published at Asiacrypt Dec 2002. http://eprint.iacr.org/2002/044 [Ferguson2001] Niels Ferguson, Richard Schroeppel, Doug Whiting, A simple algebraic representation of Rijndael, Draft 2001/05/1, http://www.xs4all.nl/~vorpal/pubs/rdalgeq.html [Lucks-DuD2002] Stefan Lucks, R¨diger Weis, u Neue Ergebnisse zur Sicherheit des Verschl¨sselungsstandards AES, in DuD Dec. 2002. u [Nichols1996] Randall K. Nichols, Classical Cryptography Course, Volume 1 and 2, Aegean Park Press 1996; or in 12 lessons online at http://www.fortunecity.com/skyscraper/coding/379/lesson1.htm [Robshaw2002a] S.P. Murphy, M.J.B. Robshaw, Essential Algebraic Structure within the AES, June 5, 2002, Crypto 2002, http://www.isg.rhul.ac.uk/ mrobshaw/rijndael/rijndael.html [Robshaw2002b] S.P. Murphy, M.J.B. Robshaw, Comments on the Security of the AES and the XSL Technique, September 26, 2002, http://www.isg.rhul.ac.uk/ mrobshaw/rijndael/rijndael.html [Schmeh2003] Klaus Schmeh, Cryptography and Public Key Infrastructures on the Internet, John Wiley & Sons Ltd., Chichester 2003. A considerable, up-to-date, easy to read book, which also considers practical problems such as standardisation or real existing software. [Schneier1996] Bruce Schneier, Applied Cryptography, Protocols, Algorithms, and Source Code in C, Wiley 1994, 2nd edition 1996. [Wobst-iX2002] Reinhard Wobst, Angekratzt - Kryptoanalyse von AES schreitet voran, in iX Dec. 2002, plus the reader’s remark by Johannes Merkle in iX Feb. 2003.

Web links

1. AES or Rijndael Cryptosystem http://www.cryptosystem.net/aes http://www.minrank.org/aes/
2. AES Discussion Groups at NIST http://aes.nist.gov/aes
3. distributed.net: “RC5-64 has been solved” http://distributed.net/pressroom/news-20020926.html
4. RSA: “The RSA Secret Key Challenge” http://www.rsasecurity.com/rsalabs/challenges/secretkey/index.html
5. RSA: “DES Challenge” http://www.rsasecurity.com/rsalabs/challenges/des3/index.html
6. Further Links can be found at the CrypTool Homepage http://www.cryptool.org

Edgar Allan Poe: A Few Words on Secret Writing, 1841 Few persons can be made to believe that it is not quite an easy thing to invent a method of secret writing which shall baffle investigation. Yet it may be roundly asserted that human ingenuity cannot concoct a cipher which human ingenuity cannot resolve.

2

Paper and Pencil Encryption Methods

(Christine St¨tzel, April 2004; Updates: B.+C. Esslinger, June 2005) o The following chapter provides a broad overview of paper and pencil methods13 each with references to deeper information. All techniques that people can apply manually to en- and decipher a message are embraced by this term. These methods were and still are especially popular with secret services, as a writing pad and a pencil – in contrast to electronical aids – are totally unsuspicious. The first paper- and pencilmethods already arose about 3000 years ago, but new procedures were developed during the past century, too. All paper and pencil methods are a matter of symmetric methods. Even the earliest encryption algorithms use the basic principles such as transposition, substitution, block construction and their combinations. Hence it is worthwhile to closely consider this “ancient” methods especially under didactic aspects. Methods to be successful and wide-spreaded had to fulfill some attributes which are equally required for modern algorithms: • Exhaustive description, almost standardisation (including special cases, padding, etc.). • Good balance between security and usability (because methods being too complicated were error-prone or unacceptably slow).

2.1

Transposition ciphers

Encrypting a message by means of transposition does not change the original characters of this message, only their order is modified (transposition = exchange)14 . 2.1.1 Introductionary samples of different transposition ciphers

• Railfence15 [Singh2001]: The characters of a message are alternately written in two (or more) lines, creating a zigzag pattern. The resulting cihpertext is read out line by line. This is more a children’s method.

Always added with links to further information. Another name uesed for transposition is permutation. 15 In CrypTool you can simulate this method under the menu Crypt \ Symmetric (classic) \ Permutation: for a railfence with 2 lines use as key “B,A” and accept the default settings (only one permutation, where your input is done line-by-line and the ouput is taken column-by-column). Using the key “A,B” would start the zigzag pattern below in the way, that the first letter is written into the first line instead of the second line.
14

13

Plaintext16 : an example of transposition nxm lota s oiin a e a pe fr np sto Table 1: Railfence cipher Ciphertext17 : NXMLO TASOI INAEA PEFRN PSTO • Scytale18 [Singh2001]: This metod was probably used since 600 B.C. – a description of how it operated is not known from before Plutarch (50-120 B.C.). A long strip of paper is wrapped around a wooden cylinder and then the message is written along the length of this strip. The ciphertext is produced by unwinding the strip. • Grille [Goebel2003]: Both parties use identical stencils. Line by line, their holes are filled with plaintext that is read out column by column to produce the ciphertext. If there is plaintext left, the procedure is repeated19 . • Turning grille [Savard1999]: The German army used turning grilles during WW120 . A square grille serves as a stencil, a quarter of its fields being holes. The first part of the message is written on a piece of paper through these holes, then the grille is rotated by 90 degrees and the user can write down the second part of the message, etc. But this method does only work, if the holes are chosen carefully: Every field has to be used, and no field may be used twice, either. The ciphertext is read out line by line. In the example for a turning grille in the following table you can write 4 times 16 characters of the cleartext on a piece of paper:

If the alphabet only uses 26 letters, we write the plaintext in small letters and the ciphertext in capital letters. The letters of the cleartext are – as used historically – grouped within blocks of 5 letters. It does not matter if the (constant) blocklength is different or no blank is inserted. 18 The result of this encryption method meets the one of a simple columnar transposition. In CrypTool you can simulate this method under the menu Crypt \ Symmetric (classic) \ Permutation: For the Scytale within the dialog box only the first permutation is used. If the wood has e.g. 4 angles use as key “1,2,3,4”. This is equivalent to write the text horizontally in blocks of 4 letters in a matrix and to read it out vertically . Because the key is in an in ascending order, the Scytale is denoted as an identical permutation. And because writing and read-out is done only once it is a simple (and no double) permutation. 19 This method cannot be simulated with a pure column transposition. 20 The turning grille was already invented in 1881 by Eduard Fleissner von Wostrowitz. A good visualization can be found under www.turning-grille.com.
17

16

20

O O -

O -

O O -

O O O

O O O

O -

O O -

O O -

Table 2: A 8 * 8 turning grille

2.1.2

Column and row transposition21

- Simple columnar transposition [Savard1999]: First of all, a keyword is chosen, that is written above the columns of a table. This table is filled with the text to be encrypted line by line. Then the columns are rearranged by sorting the letters of the keyword alphabetically. Afterwards the columns are read out from left to right to build the ciphertext22 . Plaintext: an example of transposition K a x p o r s s i E n a l f a p i o Y e m e t n o t n

Table 3: Simple columnar transposition Transposition key: K=2; E=1; Y=3. Ciphertext: NALFA PIOAX PORSS IEMET NOTN • AMSCO [ACA2002]: The characters of the plaintext are written in alternating groups of one respectively two letters into a grille. Then the columns are swapped and the text can be read out. • Double column transposition [Savard1999] : Double columnar transposition was fre-

21

Most of the following methods can be simulated in CrypTool under the menu Crypt \ Symmetric (classic) \ Permutation. 22 Using CrypTool: Choose a key for the 1st permutation, input line by line, permute and output column by column.

quently used during WW2 and during the Cold War. Two simple columnar transpositions with different keys are executed successively23 . • Column transposition, General Luigi Sacco [Savard1999]: The columns of a table are numbered according to the letters of the keyword. The plaintext is entered line by line, in the first line up to column number one, in the second line up to column number two, etc. Again, the ciphertext is read out in columns. Plaintext: an example of transposition C 1 a n a o s o i O 5 e m f p s o L 2 x p t i n U 6 M 3 N 4

l r t

e a

n

Table 4: Columnar transposition (General Luigi Sacco) Ciphertext: ANAOS OIEMF PSOXP TINLR TEAN • Column transposition, French army in WW1 [Savard1999]: After executing a simple columnar transposition, diagonal rows are read out. • Row transposition [Savard1999]: The plaintext is divided into blocks of equal length and a keyword is chosen. Now the letters of the keyword are numbered and permutation is done only within each block according to this numbering24 . 2.1.3 Further transposition algorithm ciphers

• Geometric figures [Goebel2003]: Write the message into a grille following one pattern and read it out using another. • Union Route Cipher [Goebel2003]: The Union Route Cipher derives from Civil War. This method does not rearrange letters of a given plaintext, but whole words. Particularly sensitive names and terms are substituted by codewords which are recorded in codebooks together with the existing routes. A route determines the size of a grille and the pattern that is used to read out the ciphertext. Aditionally, a number of filler words is defined.

Using CrypTool: Choose a key for the 1st permutation, input line by line, permute and output column by column. Then choose a (different) key for the 2nd permutation, input line by line, permute and output column by column. 24 Using CrypTool: Choose a key for 1st permutation, input line by line, permute column by column and output line by line.

23

- Nihilist Transposition [ACA2002]: Insert the plaintext into a square grille and write the same keyword above the columns and next to the lines. As this keyword is sorted alphabetically, the contents of the grille are rearranged, too. Read out the ciphertext line by line. Plaintext: an example of transposition W a m f s t O n p t p i R e l r o o D x e a s n S a o n i D s e a n x O p p t i n R o l r o e S i o n a W s m f t a

W O R D S

D O R S W

Table 5: Nihilist transposition25

Ciphertext: SPOIS EPLOM ATRNF NIOTX NEAA
- Cadenus [ACA2002]: Cadenus is a form of columnar transposition that uses two keywords. The 1st keyword is used to swap columns. The 2nd keyword is used to define the initial letter of each column: this 2nd keyword is a permutation of the used alphabet. This permutation is written on the left of the first column. Afterwards, each column is moved (wrap-around) so that it begins with the letter, which is in the the same line as the key letter of the first keyword within the second keyword. Ciphertext is read out line by line. See table 6. Plaintext: cadenus is a form of columnar transposition using a keyword Ciphertext: SAASR PIFIU LONNS KTGWN EDOOA TDNNU IISFA OMYOC ROUCM AERRS

After filling the matrix with the cleartext you get the left block. After switching rows and columns you get the right block 26 Within the 2nd block of three chars those chars are printed bold which are at the top of the 3rd block after applying the 2nd key word.

25

	A	D	X	K	C	W	N	S	Y	E	D	T	U	B	R	G	H
K	c	e	s	a	r	f	l	n	t	n	o	t	n	i	a	y	r
E	a	n	i	f	m	c	u	a	r	s	s	i	u	n	k	w	d
Y	d	u	s	o	o	o	m	r	a	p	i	o	s	g	e	o	-
E	a	n	i	f	m	c	u	a	r	s	s	i	u	n	k	w	d
K	c	e	s	a	r	f	l	n	t	n	o	t	n	i	a	y	r
Y	d	u	s	o	o	o	m	r	a	p	i	o	s	g	e	o	-
E	s	s	i	u	n	k	w	d	a	n	i	f	m	c	u	a	r
K	a	r	f	l	n	t	n	o	t	n	i	a	y	r	c	e	s
Y	a	p	i	o	s	g	e	o	d	u	s	o	o	o	m	r	-

Table 6: Cadenus[26]

2.2 Substitution ciphers

2.2.1 Monoalphabetic substitution ciphers

Monoalphabetic substitution assigns one character of the ciphertext alphabet to each plaintext character. This mapping remains unchanged during the whole process of encryption. • General monoalphabetic substitution / Random letter pairs[27] [Singh2001]: The substitution occurs by a given assignment of single letters. • Atbash[28] [Singh2001]: Replace the first letter of the alphabet by the last letter of the alphabet, the second one by the last but one, etc. • Shift cipher, for example Caesar cipher[29] [Singh2001]: Plaintext alphabet and ciphertext alphabet are shifted against each other by a determined number of letters. Using the Caesar cipher means shifting letters about three positions.

---

[27] This cipher can be simulated in CrypTool under the menu Crypt \ Symmetric (classic) \ Substitution / Atbash.

[28] This cipher can be simulated in CrypTool under the menu Crypt \ Symmetric (classic) \ Substitution / Atbash.

[29] In CrypTool this method can be find at three different places in the menu tree: - Crypt \ Symmetric (classic) \ Caesar / ROT13 - Analysis \ Symmetric Encryption (classic) \ Ciphertext only \ Caesar - Indiv. Procedures \ Visualization of Algorithms using ANIMAL \ Caesar.

Plaintext: three positions to the right Ciphertext: WKUHH SRVLWLRQV WR WKH ULJKW • Substitution with symbols [Singh2001], for instance the so-called “freemason cipher”: Each letter is replaced with a symbol. • Variants: Fill characters, intentional mistakes [Singh2001]. • Nihilist Substitution[30] [ACA2002]: Insert the alphabet into a 5x5-matrix and replace every letter of the message with the two corresponding digits. The resulting numbers are written into a grille. For this purpose, a keyword is chosen and placed above the columns of the grille. Its letters are substituted by numbers, too. The ciphertext results from adding the numbers of the plaintext and the numbers of the keyword. Numbers between 100 and 110 are transformed to numbers between 00 and 10, so that each letter is represented by a two-digit number. See table 7. Plaintext: an example of substitution Ciphertext: 58 53 85 88 54 96 78 72 85 56 63 65 47 44 65 49 46 68 47 55 69 56 53

---

[30] An animation of this Nihilist method can be find in CrypTool at the menu item Indiv. Procedures \ Visualization of Algorithms using ANIMAL \ Nihilist.

Matrix

1 2 3 4 5

1 S O E L V K (35) a (58) x (88) p (78) o (56) u (47) t (49) u (47) o (56)

2 U N F M W

3 B A G P X

4 T C H Q Y Y (54) e (85) m (96) e (85) s (65) s (65) t (68) i (69)

5 I D K R Z

Table

E (31) n (53) a (54) l (72) f (63) b (44) i (46) t (55) n (53)

Table 7: Nihilist Substitution

• Coding [Singh2001]: In the course of time, codebooks were used again and again. A codebook assigns a codeword, a symbol or a number to every possible word of a message. Only if both parties hold identical codebooks and if the assignment of codewords to plaintext words is not revealed, a successful and secret communication can take place. • Nomenclature [Singh2001]: A nomenclature is an encryption system that is based upon a ciphertext alphabet. This alphabet is used to encrypt the bigger part of the message. Particularly frequent or top-secret words are replaced by a limited number of codewords existing besides the ciphertext alphabet. • Map cipher [ThinkQuest1999]: This method constitutes a combination of substitution and steganography[31] . Plaintext characters are replaced by symbols which are arranged in a map following certain rules • Straddling Checkerboard [Goebel2003]: A 3x10-matrix is filled with the letters of the

[31] Instead of encrypting a message, pure steganography tries to conceal its existence.

used alphabet and two arbitrary digits or special characters as follows: The different letters of a keyword and the remaining characters are written into the grille. The columns are numbered 0 to 9, the second and the third line are numbered 1 and 2. Each plaintext character is replaced by the corresponding digit, respectively the corresponding pair of digits. As “1” and “2” are the first digits of the possible two-digit-numbers, they are not used as single digits. See table 8. Plaintext: an example of substitution 0 K B P 1 C Q 2 F S 3 E G T 4 Y H U 5 W I V 6 O J X 7 R L Z 8 D M . 9 A N /

1 2

Table 8: Straddling Checkerboard with password “Keyword”

Ciphertext: 91932 69182 01736 12222 41022 23152 32423 15619 Besides, “1” and “2” are the most commonly used digits, but this feature is removed by the following technique. It is ostentatious, how often the numbers 1 and 1 appear, but this will be fixed with the following version. • Straddling Checkerboard, variant [Goebel2003]: This variant of the straddling checkerboard was developed by Sovjet spies during WW2. Ernesto (Ch¥) Guevara and Fidel Castro e allegedly used this cipher for their secret communication. A grille is filled with the alphabet (number of columns = length of keyword), and two arbitrary digits are chosen as reserved to indicate the second and third line of a 3x10-matrix (see above). Now the grille is traversed column by column and the single letters are transfered row by row into the matrix: For a faster encryption, the eight most common letters (ENIRSATO) are assigned the digits from 0 to 9, the reserved 2 digits are not assigned. The remaining letters are provided with combinations of digits one after another and are inserted into the grille. See table 9. Plaintext: an example of substitution Ciphertext: 04271 03773 33257 09343 29181 34185 4

Grille

K A J T 1 T J X

E B L U 2 E B G

Y C M V 3 L P

W F N X 4 N U Z

O G P Z 5 O Y H

R H Q . 6 R C Q

D I S / 7 M . 8 I V D 9 S W /

Matrix

3 7

0 A K F

Table 9: Variant of the Straddling Checkerboard

– Ch¥ Guevara Cipher: A special variant is the cipher used by Ch¥ Guevara (with e e an additional substitution step and a slightly changed checkerboard): ∗ The seven most frequent letters in Spanish are distributed in the first row. ∗ Four instead of three rows are used. ∗ So one could encrypt 10 ∗ 4 − 4 = 36 different characters.

• Tri-Digital [ACA2002]: A keyword with ten letters is used to create a numeric key by numbering its letters corresponding to their alphabetical order. This key is written above the columns of 3x10-matrix. This matrix is filled line by line with the alphabet as follows: The different letters of a keyword are inserted first, followed by the remaining letters. The last column is left out. Plaintext characters are substituted with numbers, the number of the last column is used to seperate words. • Baconian Cipher [ACA2002]: Assign a five-digit binary code to every letter and to 6 numbers or special characters (for example 00000 = A, 00001 = B, etc.) and replace the plaintext characters with this binary code. Now use a second, unsuspicious message to hide the ciphertext inside of it. This may happen by upper and lower case or italicized letters: e.g. all letters of the unsuspicious message below a binary “1” are capitalised. See table 10. message ciphertext unsuspicious message Baconian Cipher F 00101 itisw itIsW I 01000 arman aRman G 00110 thesu thESu H 00111 nissh niSSH T 10011 ining IniNG

Table 10: Baconian Cipher

2.2.2

Homophonic substitution ciphers

Homophonic methods constitute a special form of monoalphabetic substitution. Each character of the plaintext alphabet is assigned several ciphertext characters. • Homophonic monoalphabetic substitution32 [Singh2001]: Each language has a typical frequency distribution of letters. To conceal this distribution, each plaintext letter is assigned several ciphertext characters. The number of ciphertext characters assigned depends on the frequency of the letter to be encrypted. • Beale cipher [Singh2001]: The Beale cipher is a book cipher that numbers the words of a keytext. These numbers replace the cleartext letters by the words' initial letters. • Grandpr¥ Cipher [Savard1999]: A square grille with 10 columns (other layouts are pose sible, too) is filled with ten words. The initial letters should result in an eleventh word. As columns and rows are numbered from 0 to 9, letters can be replaced by two-digit numbers. It is obvious that with the table having a hundred fields, most letters can be represented by more than one number. You should keep in mind that those ten words have to contain all letters of the plaintext alphabet. • Book cipher: The words of a message are substituted by triples "page-line-position". This method requires a detailed agreement of which book to use, especially regarding the edition (layout, error correction, etc.). 2.2.3 Polygraphic substitution ciphers

Polygraphic techniques do not work by replacing single characters, but by replacing whole groups of characters. In most cases, these groups are digramms, trigramms or syllables. • "Great Chiffre" [Singh2001]: This cipher was used by Louis XIV. and was not solved until the end of the nineteenth century. Cryptograms consisted of 587 different numbers, every number representing a syllable. The inventors of the "Great Chiffre" (Rossignol, father and son) constructed additional traps to increase security. For example, a number could assign a different meaning to or delete the preceding one. • Playfair33 [Singh2001]: A 5x5-matrix is filled with the plaintext characters. For example, the different letters of a keyword are inserted first, followed by the remaining letters. The plaintext is divided into pairs, these digraphs are encrypted using the following rules: 1. If both letters can be found in the same column, they are replaced by the letters underneath. 2. If both letters can be found in the same row, take the letters to their right.

32 33

This cipher can be simulated in CrypTool under the menu Crypt \Symmetric (classic)\ Homophone. In CrypTool you can call this method under the menu Crypt \ Symmetric (classic) \ Playfair.

3. If both letters of the digraph are in different columns and rows, the replacement letters are obtained by scanning along the row of the first letter up to the column where the other letter occurs and vice versa. 4. Double letters are treated by special rules, if they apear in one diagraph. They can be seperated by a filler, for example. See table 11. Plaintext: plaintext letters are x encrypted in pairs K R F M T E D G N U Y A H P V W B I Q X O C L S Z

Table 11: 5x5-Playfair Ciphertext: SHBHM UWUZF KUUKC MBDWU DURDA VUKBG PQBHC M • Trigraphic Playfair: A 5x5-matrix is filled with the alphabet (see above) and the plaintext is divided into trigraphs. Trigraphs are encrypted according to the following rules: 1. Three equal letters are substituted by three equal letters. It is the letter on the right underneath the original letter. 2. A trigraph with two different letters is encrypted like a digraph in Playfair. 3. If a trigraph contains three different characters, very complex rules come into effect. See [Savard1999] • Substituting digraphs by symbols [Savard1999]: Giovanni Battista della Porta, 15th century. He created a 20x20-matrix that contained one symbol for every possible combination of letters (his alphabet did not comprise more than twenty letters). • Four square cipher [Savard1999]: This method is similar to Playfair, because it is based on a system of coordinates whose four quadrants are each filled with the alphabet. The layout of letters can differ from quadrant to quadrant. To encipher a message, act in the following way: Look up the first plaintext letter in the first quadrant and the second one in the third quadrant. These two letters are opposite corners of a rectangle and the ciphertext letters can be found in quadrant number two and four. See table 12. Plaintext: plaintext letters are encrypted in pairs Ciphertext: MWYQW XQINO VNKGC ZWPZF FGZPM DIICC GRVCS

d	r	u	a	g	Q	Z	P	V	U
w	q	v	l	c	T	H	M	S	A
x	e	h	b	o	B	N	I	K	F
y	k	p	z	f	L	D	Y	W	R
m	i	s	n	t	E	X	C	O	G
E	C	R	F	B	v	s	n	b	c
P	V	M	W	N	q	t	r	l	z
T	I	A	Y	D	i	u	d	w	k
O	Q	G	H	X	p	o	x	m	a
L	Z	U	S	K	g	h	y	f	e

Table 12: Four Square Cipher

• Two square cipher [Savard1999]: The two square cipher resembles the four square cipher, but the matrix is reduced to two quadrants. Are both letters of the digraph part of the same row, they are just exchanged. Otherwise, the plaintext letters are considered as opposite corners of a rectangle and substituted by the other vertices. Quadrants can be arranged horizontal and vertical. • Tri square cipher [ACA2002]: Three quadrants are filled with the same alphabet. The first plaintext letter is looked up in the first quadrant and can be encrypted with every letter of that column. The second plaintext letter is looked up in the second quadrant (diagonally across) and can be encrypted with every letter of that row. Between these two ciphertext characters, the letter at the intersection point is set. • Dockyard Cipher [Savard1999]: Used by the German navy during WW2. 2.2.4 Polyalphabetic substitution ciphers

Concerning polyalphabetic substitution, the assignment of ciphertext characters to plaintext characters is not static, but changes during the process of encryption (depending on the key). • Vigen`re34 [Singh2001]: Each plaintext character is encrypted with a different ciphertext e alphabet that is determined by the characters of a keyword (the so-called Vigen`re-Tableau e serves auxiliary means). If the plaintext is longer than the key, the latter is repeated. See table 13. – Interrupted key: The key is not repeated continously, but starts again with every new word of the message. – Autokey [Savard1999]: After using the agreed key, use the message itself as a key. See table 14.

34

In CrypTool you can call this method under the menu Crypt \ Symmetric (classic) \ Vigen`re. e

Plaintext: Key: Ciphertext:

the KEY DLC

alphabet KEYKEYKE KPNREZOX

is YK GC

changing EYKEYKEY GFKRESRE

- A B C D E F G H I J K L M N O P Q R S T U VWXY Z A A B C D E F G H I J K L M N O
P Q R S T U VWXY Z B B C D E F G H I J K LM N O P Q R S T U VW XY ZA C C D E F G H
I J K LM N O P Q R S T U VW X Y ZA B D D E F G H I J K LM N O P Q R S T U VW X Y
ZA BC E E F G H I J K LM N O P Q R S T U VW X Y Z A BCD F FG H I J K LM N O P Q R
S T U VW X Y Z A BCD E G G H I J K LM N O P Q R S T U VW X Y Z A B CD E F H H I J
K LM N O P Q R S T U VW X Y Z A B C D E FG I I J K LM N O P Q R S T U VW X Y Z A B
C D E FGH J JK LM N O P Q R S T U VW X Y Z A B C D E FGH I K K LM N O P Q R S T U
VW X Y Z A B C D E F GH I J ... ... Table 13: Vigen`re-Tableau e Plaintext:
Key: Ciphertext: the KEY DLC alphabet THEALPHA TSTHLQLT is BE JW changing TISCHANG
VPSPNIAM

Table 14: Autokey

– Progressive key [Savard1999]: The key changes during the process of encryption. With every repetition, the characters of the keyword are shifted about one position. “KEY” becomes “LFZ”. – Gronsfeld [Savard1999]: Variant of Vigen`re that uses a numeric key. e – Beaufort [Savard1999]: Variant of Vigen`re, the key is subtracted, not added. The e ciphertext alphabets may be written backwards. – Porta [ACA2002]: Variant of Vigen`re with only 13 alphabets. As a consequence, e two letters of the keyword are assigned the same ciphertext alphabet and the first and the second half of the alphabet are reciprocal. – Slidefair [ACA2002]: This method can be used as a variant of Vigen`re, Gronsfeld or e Beaufort. Slidefair does encrypt digraphs according to the following rules: Look up the first letter in the plaintext alphabet above the tableau. Then look up the second one in the row belonging to the corresponding keyword letter. These two letters make up opposite corners of an imaginary rectangle. The letters at the two remaining corners substitute the digraph.

• Superposition – Book cipher: A keytext (for example out of a book) is added to the plaintext. – Superposition with numbers: A sequence or a number of sufficient length (for example pi) is added. • Phillips [ACA2002]: The alphabet is filled into a square table with 5 columns. Seven more tables are generated by first shifting the first row one position towards the bottom, then shifting the second row towards the bottom. The plaintext is divided into blocks of five which are encrypted with one matrix each. Letters are substituted by the ones on their right and underneath. • Ragbaby [ACA2002]: Construct an alphabet with 24 characters. Then number the plaintext characters, starting the numeration of the first word with “1”, the numeration of the second one with “2” and so forth. Number 25 corresponds to number 1. Each letter of the message is encrypted by shifting it the corresponding postions to the right. alphabet: KEYWORDABCFGHILMNPSTUVXZ Plaintext: Numbering: Ciphertext: t h e a l p ha b e t i s c han g i n g 1 2 3 2 3 4 5 6 7 8 9 3 4 4 5 6 7 8 9 10 11 ULO CPVP IMCO NX IP IZTX Y X Table 15: Ragbaby

2.3

Combining substitution and transposition

In the history of cryptography one often comes across combinations of the previous mentioned methods. • ADFG(V)X[35] [Singh2001]: ADFG(V)X-encryption was developed in Germany during WW1. The alphabet is filled into a 5x5- or 6x6-matrix, and columns and rows are marked with the letters ADFGX and V, depending on the size of the grille. Each plaintext character is substituted by the corresponding pair of letters. Finally, a (row-) transposition cipher is performed on the resulting text. • Fractionation [Savard1999]: Generic term for all kinds of methods that encrypt one plaintext character by several ciphertext characters and then apply a transposition cipher to this ciphertext so that ciphertext characters originally belonging to each other are separated. – Bifid/Polybius square/checkerboard [Goebel2003]: Bifid encryption is the basic form of fractionation. A 5x5-matrix is filled with the plaintext alphabet (see Playfair

---

[35] In CrypTool you can call this method under the menu Crypt \ Symmetric (classic) \ ADFGVX.

encryption), rows and columns are numbered, so that each cleartext character can be substituted by a pair of digits. Mostly the plaintext is divided into blocks of equal length. The length of blocks (here 5) is another configuration parameter of this cipher. Block-by-block all line numbers are read out first, followed by all numbers naming the columns. To obtain the ciphertext, the digits are pairwise transformed into letters again. The numbers can be any permutation of (1,2,3,4,5), which is one key of configuration parameter of this cipher. Instead of numbering rows and columns, a keyword can be used, too. See table 16. 1 4 2 3 5 Plaintext: Rows: Columns: combi 41342 33211 2 K R F M T 4 E D G N U 5 Y A H P V 1 W B I Q X tutio 55521 24213 3 O C L S Z nandt 34345 45442 ransp 44333 25435 ositi 13252 33121

nings 32323 41443

ubsti 54352 41321

Table 16: Bifid 41342 32323 54352 55521 34345 44333 13252 33211 41443 41321 24213 45442 25435 33121 Ciphertext: BNLLL UPHVI NNUCS OHLMW BDNOI GINUR HCZQI – Trifid [Savard1999]: 27 characters (alphabet + 1 special character) may be represented by a triple consisting of the digits 1 to 3. The message to be encrypted is divided into blocks of three and the relevant triple is written underneath each plaintext character as a column. The resulting numbers below the plaintext blocks are read out line by line and are substituted with the corresponding characters. • Bazeries [ACA2002]: The plaintext alphabet is filled into a 5x5-matrix column by column, a second matrix is filled line by line with a keyword (a number smaller than a million) followed by the remaining letters of the alphabet. Then the message is divided into blocks of arbitrary length and their characters' order is inverted. Finally, each letter is substituted – according to its position in the original matrix – by its counterpart in the second matrix. See table 17. Plaintext: combining substitution and transposition Keyword: 900.004 (nine hundred thousand and four)

a b c d e com moc TMA bini inib LBLD ngs sgn CRB

f g h i k ub bu DY

l m n o p

q r s t u

v w x y z

N D A K V tiona anoit NBMLP

I R F L W

E T B M X ndt tdn PKB

H O C P Y

U S G Q Z sposi isops LCMXC ti it LP on no BM

stitu utits YPLPC

ran nar BNO

Table 17: Bazeries

- Digrafid [ACA2002]: To substitute digraphs, the following table is used (to simplify matters, the alphabet is used in its original form). Look up the first letter of the digraph in the horizontal alphabet and write down the column number. Then look up the second letter in the vertical alphabet and write down the corresponding line number. Between these two numbers, the number at the intersection point is set. Afterwards, the tripel are written vertically underneath the digraphs that are arranged in groups of three. The three digit numbers arising horizontally are transformed back into digraphs. Remark: This cipher only works with complete blocks of 3 pairs of cleartext characters. For a complete description, it is necessary to explain how sender and receiver handle texts which fill in the last block only 1-5 characters. The possibilities range from ignoring a last and incomplete block to padding it with randome characters or with characters predefined in advance. See table 18.

1 A J S

2 B K T

3 C L U

4 D M V

5 E N W

6 F O X

7 G P Y

8 H Q Z

9 I R .

1 4 7 A B C D E F G H I

2 5 8 J K L M N O P Q R

3 6 9 S T U V W X Y Z . tr 2 8 9 BA

1 2 3 4 5 6 7 8 9 an 1 2 5 HQ sp 1 8 7 RP os 6 6 1 X. it 9 3 2 FT io 9 2 6 AO

co 3 2 6 LI

mb 4 4 2 KB

in 9 2 5 FN

in 9 2 5 .C

gs 7 3 1 BY

ub 3 7 2 EB

st 1 9 2 SU

it 9 3 2 I.

ut 3 9 2 BK

io 9 2 6 RN

na 5 4 1 KD

nd 5 4 4 FD

Table 18: Digrafid

- Nicodemus [ACA2002]: First of all, a simple columnar transposition is carried out. Before reading out the columns, the message is encrypted aditionally by Vigen`re (all letters of a e column are enciphered with the corresponding keyword letter). The ciphertext is read out in vertical blocks. See table 19. Plaintext: combining substitution and transpositio Ciphertext: SMRYX MLSCC KLEZG YSRVW JSKDX RLBYN WMYDG N

K	c	b	i	s	s	t	i	a	t	n	o	t
E	o	i	n	u	t	u	o	n	r	s	s	i
Y	m	n	g	b	i	t	n	d	a	p	i	o
E	o	i	n	u	t	u	o	n	r	s	s	i
K	c	b	i	s	s	z	i	a	t	n	o	t
Y	m	n	g	b	i	t	n	d	a	p	i	o
E	S	M	R	Y	X	Y	S	R	V	W	W	M
K	M	L	S	C	C	J	S	K	D	X	Y	D
Y	K	L	E	Z	G	R	L	B	Y	N	G	N

Table 19: Nicodemus

2.4 Further methods

- “Pinprick encryption” [Singh2001]: For centuries, this simple encryption method has been put into practice for different reasons. During the Victorian Age, for example, small holes underneath letters in newspaper articles marked the characters of a plaintext, as sending a newspaper was much more cheaper than the postage on a letter.
- Stencil: Stencils (Cardboard with holes) are also known as “Cardinal-Richelieu-Key”. Sender and receiver have to agree upon a text. Above this text, a stencil is laid and the letters that remain visible make up the ciphertext.
- Card games [Savard1999]: The key is created by means of a pack of cards and rules that are agreed upon in advance. All methods mentioned in this paragraph are designed as paper- and pencilmethods, i.e. they are applicable without electronical aid. A pack of cards is unsuspicious to outsiders, shuffling the deck provides a certain amount of coincidence, cards can be transformed into numbers easily and a transposition cipher can be carried out without any further aid.
  - Solitaire (Bruce Schneier)[36] [Schneier1999]: Sender and receiver have to own a deck of cards shuffled in the same manner. A key stream is generated that has to consist of as many characters as the message to be encrypted. The algorithm to generate the key is based on a shuffled deck of 54 cards (Ace, 2 - 10, jack, queen, king in four suits and two jokers). The pack of cards is held face up: 1. Swap the first joker with the card beneath it. 2. Move the second joker two cards down. 3. Now swap the cards above the first joker with those below the second one.

[36] In CrypTool you can call this method under the menu Crypt \ Symmetric (classic) \ Solitaire.

4. Look at the bottom card and convert it into a number from 1 to 53 (bridge order of suits: clubs, diamonds, hearts, spades; joker = 53). Write down this number and count down as many cards starting with the top card. These cards are swapped with the remaining cards, only the bottom card remains untouched. 5. Look at the top card and convert it into a number, too. Count down as many cards starting with the top card. 6. Write down the number of the following card. This card is converted into your first keystream character. As we need numbers from 1 to 26 to match the letters of our alphabet, clubs and hearts correspond to the numbers 1 to 13, diamonds and spades to 14 to 26. If your output card is a jocker, start again. For each keystream character you like to generate, these six steps have to be carried out. This procedure is – manually – very lengthy (4 h for 300 characters, dependant on your excercise) and requires high concentration. Encryption takes place by addition modulo 26. Encryption is relatively fast compared to the key stream generation. – Mirdek (Paul Crowley) [Crowley2000]: Even though this method is quite complicated, the author provides a very good example to illustrate the procedure. – Playing Card Cipher (John Savard) [Savard1999]: This algorithm uses a shuffled deck of 52 cards (no joker). Separate rules describe how to shuffle the deck. A keystream is created via the following steps: 1. The pack of cards lies in front of the user, top down. Cards are turned up and dealt out in a row until the total of the cards is 8 or more. 2. If the last card dealt out is a J, Q or K, write down its value, otherwise write down the sum of the cards dealt out (a number between 8 and 17). In a second row, deal out that number of cards. 3. The remaining cards are dealt out in rows under the second row. The first one ends under the lowest card of the top row, the second one under the next lowest card, and so on. If there are two identical cards, red is lower than black. 4. The cards dealt out under step 3 are collected column by column, starting with the column under the lowest card. The first card that is picked up becomes the bottom card (face up). 5. The cards dealt out in step 1 and 2 are picked up, beginning with the last card. 6. The deck is turned over, the top card is now the bottom card (face down). Afterwards, steps 1 to 6 are repeated twice. To generate a keystream character, write down the first card not being J, Q or K. Count down that number of cards. The card selected has to be between 1 and 10. Now repeat these steps beginning with the last card. These two numbers are added and the last digit of the sum is your keystream character. • VIC cipher [Savard1999]: This is a highly complicated but relatively secure paper- and pencilmethod. It has been developed and applied by Sovjet spies. Amongst other things,

the user had to create ten pseudo-random numbers out of a date, the first words of a sentence and any five-digit number. A straddling checkerboard is part of the encryption, too. A detailed description can be found under [Savard1999].

References

[ACA2002] American Cryptogram Association, Length and Standards for all ACA Ciphers, 2002. http://www.cryptogram.org/cdb/aca.info/aca.and.you/chap08.html#

[Bauer1995] Friedrich L. Bauer, Entzifferte Geheimnisse, Springer, 1995.

[Bauer2000] Friedrich L. Bauer, Decrypted Secrets, Springer 1997, 2nd edition 2000.

[Crowley2000] Paul Crowley, Mirdek: A card cipher inspired by “Solitaire”, 2000. http://www.ciphergoth.org/crypto/mirdek/

[DA1999] Data encryption page of the ThinkQuest Team 27158 for ThinkQuest 1999 (no update since 1999, no search possibility), 1999. http://library.thinkquest.org/27158/

[Goebel2003] Greg Goebel, Codes, Ciphers and Codebreaking, 2003. http://www.vectorsite.net/ttcode.htm

[Nichols1996] Randall K. Nichols, Classical Cryptography Course, Volume 1 and 2, Aegean Park Press 1996; or in 12 lessons online at http://www.fortunecity.com/skyscraper/coding/379/lesson1.htm

[Savard1999] John J. G. Savard, A Cryptographic Compendium, 1999. http://www.hypermaths.org/quadibloc/crypto/jscrypt.htm

[Schmeh2004] Klaus Schmeh, Die Welt der geheimen Zeichen. Die faszinierende Geschichte der Verschl¨sselung, u W3L Verlag Bochum, 1. Auflage 2004.

[Schneier1999] Bruce Schneier, The Solitaire Encryption Algorithm, version 1.2, 1999. http://www.schneier.com/solitaire.html

[Singh2001] Simon Singh, Geheime Botschaften. Die Kunst der Verschl¨sselung von der Antike bis in die Zeiten u des Internet, dtv, 2001. [ThinkQuest1999] ThinkQuest Team 27158, Data Encryption, 1999. http://library.thinkquest.org/27158/

3

Prime Numbers

(Bernhard Esslinger, May 1999, Updates Nov. 2000, Dec. 2001, June 2003, May 2005, March 2006) Albert Einstein[37] : Progress requires exchange of knowledge.

3.1

What are prime numbers?

Prime numbers are whole, positive numbers greater than or equal to 2 that can only be divided by 1 and themselves. All other natural numbers greater than or equal to 2 can be formed by multiplying prime numbers. The natural numbers N = {1, 2, 3, 4, · · · } thus comprise • the number 1 (the unit value) • the primes and • the composite numbers. Prime numbers are particularly important for 3 reasons: • In number theory, they are considered to be the basic components of natural numbers, upon which numerous brilliant mathematical ideas are based. • They are of extreme practical importance in modern cryptography (public key cryptography). The most common public key procedure, invented at the end of the 1970's, is RSA encryption. Only using (large) prime numbers for particular parameters can you guarantee that an algorithm is secure, both for the RSA procedure and for even more modern procedures (digital signature, elliptic curves). • The search for the largest known prime numbers does not have any practical usage known to date, but requires the best computers, is an excellent benchmark (possibility for determining the performance of computers) and leads to new calculation methods on many computers (see also: http://www.mersenne.org/prime.htm). Many people have been fascinated by prime numbers over the past two millennia. Ambition to make new discoveries about prime numbers has often resulted in brilliant ideas and conclusions. The following section provides an easily comprehensible introduction to the basics of prime numbers. We will also explain what is known about the distribution (density, number of prime numbers in particular intervals) of prime numbers and how prime number tests work.

---

[37] Albert Einstein, German physicist and Nobel Prize winner, Mar 14, 1879 – Apr 14, 1955.

3.2

Prime numbers in mathematics

Every whole number has a factor. The number 1 only has one factor, itself, whereas the number 12 has the six factors 1, 2, 3, 4, 6, 12. Many numbers can only be divided by themselves and by 1. With respect to multiplication, these are the "atoms" in the area of numbers. Such numbers are called prime numbers. In mathematics, a slightly different (but equivalent) definition is used. Definition 3.1. A whole number $p \in N$ is called prime if $p > 1$ and p only possesses the trivial factors ± 1 and $\pm p$. By definition, the number 1 is not a prime number. In the following sections, p will always denote a prime number. The sequence of prime numbers starts with 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, $\cdots$. The first 100 numbers include precisely 25 prime numbers. After this, the percentage of primes constantly decreases. Prime numbers can be factorised in a uniquely trivial way: $5 = 1 \cdot 5$, $17 = 1 \cdot 17$, $1, 013 = 1 \cdot 1, 013$, $1, 296, 409 = 1 \cdot 1, 296, 409$.

All numbers that have 2 or more factors not equal 1 are called composite numbers. These include $4 = 2 \cdot 2$, $6=2\cdot 3$

as well as numbers that look like primes, but are in fact composite: $91 = 7 \cdot 13$, $161 = 7 \cdot 23$, $767 = 13 \cdot 59$.

Theorem 3.1. Each whole number m greater than 1 possesses a lowest factor greater than 1. This is a prime number p. Unless m is a prime number itself, then: p is less than or equal to the square root of m. All whole numbers greater than 1 can be expressed as a product of prime numbers – in a unique way. This is the claim of the 1st fundamental theorem of number theory (= fundamental theorem of arithmetic = fundamental building block of all positive integers). Theorem 3.2. Each element n of the natural numbers greater than 1 can be written as the product $n = p1 \cdot p2 \ldots pm$ of prime numbers. If two such factorisations $n = p1 \cdot p2 \cdot \cdots \cdot pm = p1 \cdot p2 \cdots pm$ are given, then they can be reordered such that $m = m$ and for all i: $pi = pi$. ($p1 , p2 , \ldots , pm$ are called the prime factors of n).

In other words: each natural number other than 1 can be written as a product of prime numbers in precisely one way, if we ignore the order of the factors. The factors are therefore unique (the expression as a product of factors is unique)! For example, 60 = 2 · 2 · 3 · 5 = 22 · 31 · 51 . And this – other than changing the order of the factors – is the only way in which the number 60 can be factorised. If you allow numbers other than primes as factors, there are several ways of factorising integers and the uniqueness is lost: 60 = 1 · 60 = 2 · 30 = 4 · 15 = 5 · 12 = 6 · 10 = 2 · 3 · 10 = 2 · 5 · 6 = 3 · 4 · 5 = · · · . The following section is aimed more at those familiar with mathematical logic: The 1st fundamental theorem only appears to be obvious . We can construct numerous other sets of numbers (i.e. other than positive whole numbers greater than 1), for which numbers in the set cannot be expressed uniquely as a product of the prime numbers of the set: In the set M = {1, 5, 10, 15, 20, · · · } there is no equivalent to the fundamental theorem under multiplication. The first five prime numbers of this sequence are 5, 10, 15, 20, 30 (note: 10 is prime, because 5 is not a factor of 10 in this set – the result is not an element of the given basic set M). Because the following applies in M : 100 = 5 · 20 = 10 · 10 and 5, 10, 20 are all prime numbers in this set, the expression as a product of prime factors is not unique here.

3.3

How many prime numbers are there?

For the natural numbers, the primes can be compared to elements in chemistry or the elementary particles in physics (see [Blum1999, p. 22]). Although there are only 92 natural chemical elements, the number of prime numbers is unlimited. Even the Greek, Euclid38 knew this in the third century B.C. Theorem 3.3 (Euclid39). The sequence of prime numbers does not discontinue. Therefore, the quantity of prime numbers is infinite.

Euclid, a Greek mathematician of 4th and 3rd century B.C. He worked at the Egyptian academy of Alexandria and wrote “The Elements”, the most well known systematically textbook of the Greek mathematics. 39 The common usage of the term does not denote Euclid as the inventor of the theorem rather; the true inventor is merely not as prominent. The theorem has already been distinguished and proven in Euclid’s Elements (Book IX, theorem 20). The phraseology is remarkable due to the fact that the word infinite is not used. The text reads as followed O¥ π ω τ οι α ιϑμο` πλε¥ους ε`σ` παντ ος τ ο˜ π οτ εϑ¥ντ ος πλ¥ϑ ους π ω τ ων α ιϑμ˜ ν, ι ˜ ` ι ι ιι ` υ ε η ¥ ` ω the English translation of which is: the prime numbers are more than any previously existing amount of prime numbers.

38

His proof that there is an infinite number of primes is still considered to be a brilliant mathematical consideration and conclusion today (proof by contradiction). He assumed that there is only a finite number of primes and therefore a largest prime number. Based on this assumption, he drew logical conclusions until he obtained an obvious contradiction. This meant that something must be wrong. As there were no mistakes in the chain of conclusions, it could only be the assumption that was wrong. Therefore, there must be an infinite number of primes! Euclid’s proof by contradiction goes as follows: Proof Assumption: There is a finite number of primes.

Conclusion: Then these can be listed $p_1 < p_2 < p_3 < \cdots < p_n$, where n is the (finite) number of prime numbers. p_n is therefore the largest prime. Euclid now looks at the number $a = p_1 \cdot p_2 \cdots p_n + 1$. This number cannot be a prime number because it is not included in our list of primes. It must therefore be divisible by a prime, i.e. there is a natural number i between 1 and n, such that p_i divides the number a. Of course, p_i also divides the product $a - 1 = p_1 \cdot p_2 \cdots p_n$, because p_i is a factor of $a - 1$. Since p_i divides the numbers a and $a - 1$, it also divides the difference of these numbers. Thus: p_i divides $a - (a - 1) = 1$. p_i must therefore divide 1, which is impossible. Contradiction: Our assumption was false. Thus there is an infinite number of primes (Cross-reference: overview under 3.8.5 of the number of prime numbers in various intervals).

Here we should perhaps mention yet another fact which is initially somewhat surprising. Namely, in the prime numbers sequence $p_1, p_2, \cdots$, gaps between prime numbers can have an individually determined length n. It is undeniable that under the n succession of natural numbers $(n + 1)! + 2, \cdots, (n + 1)! + (n + 1)$, none of them is a prime number since in order, the numbers $2, 3, \cdots, (n + 1)$ are comprised respectively as real divisors. ($n!$ means the product of the first n natural numbers therefore $n! = n * (n - 1) * \cdots * 2 * 1$).

3.4

The search for extremely large primes

The largest prime numbers known today have several millions digits, which is too big for us to imagine. The number of elementary particles in the universe is estimated to be “only” a 80-digit number (See: overview under 3.8.7 of various orders of magnitude / dimensions).

3.4.1

The 10 largest known primes (as of March 2006)

The following table contains the ”top ten” record primes and a description of its particular number type40 : 1 2 3 4 5 6 7 8 9 10 11 Definition 230,402,457 − 1 225,964,951 − 1 224,036,583 − 1 220,996,011 − 1 213,466,917 − 1 28, 433 · 27,830,457 + 1 26,972,593 − 1 5, 359 · 25,054,502 + 1 23,021,377 − 1 22,976,221 − 1 1, 372, 930131,072 + 1 Decimal Digits 9,152,052 7,816,230 7,235,733 6,320,430 4,053,946 2,357,207 2,098,960 1,521,561 909,526 895,932 804,474 When 2005 2005 2004 2003 2001 2004 1999 2003 2001 2001 2003 Description Mersenne, 43rd known Mersenne, 42nd known Mersenne, 41st known Mersenne, 40th known Mersenne, M-39 Generalized Mersenne Mersenne, M-38 Generalized Mersenne Mersenne, M-37 Mersenne, M-36 Generalized Fermat41

Table 20: The 11 largest known primes and its particular number types (as of March 2006) The largest known prime is a Mersenne prime. This prime was found by the GIMPS project (chapter 3.4.2). Within the largest known primes there are also numbers of the type generalized Mersenne number (chapter 3.6.2) and generalized Fermat numbers (chapter 3.6.5). 3.4.2 Special number types – Mersenne numbers and Mersenne primes

Almost all known huge prime numbers are special candidates, called Mersenne numbers42 of the form 2p − 1, where p is a prime. Not all Mersenne numbers are prime: 22 − 1 = 3 23 − 1 = 7 25 − 1 = 31 27 − 1 = 127 11 − 1 = 2, 047 = 23 · 89 2 ⇒ prime ⇒ prime ⇒ prime ⇒ prime ⇒ NOT prime!

Even Mersenne knew that not all Mersenne numbers are prime (see exponent p = 11). A prime Mersenne number is called Mersenne prime number. However, he is to be thanked for the interesting conclusion that a number of the form 2n − 1 cannot be a prime number if n is a composite number:

40

41

An up-to-date version can be found in the internet at http://primes.utm.edu/largest.html. 17 1, 372, 930131,072 + 1 = 1, 372, 930(2) + 1 42 Marin Mersenne, French priest and mathematician, Sep 08, 1588 – Sep 01, 1648.

Theorem 3.4 (Mersenne). If 2n − 1 is a prime number, then n is also a prime number. Proof The theorem of Mersenne can be proved by contradiction. We therefore assume that there exists a composite natural number n (with real factorisation) n = n1 · n2 , with the property that 2n − 1 is a prime number. From (xr − 1)((xr) s−1 + (xr)s−2 + · · · + xr + 1) = ((xr)s + (xr)s−1 + (xr)s−2 + · · · + xr) −((xr)s−1 + (xr)s−2 + · · · + xr + 1) = (xr)s − 1 = xrs − 1, we conclude 2n1 n2 − 1 = (2n1 − 1)((2n1)n2 −1 + (2n1)n2 −2 + · · · + 2n1 + 1). Because 2n − 1 is a prime number, one of the above two factors on the right-hand side must be equal to 1. This is the case if and only if n1 = 1 or n2 = 1. But this contradicts our assumption. Therefore the assumption is false. This means that there exists no composite number n, such that 2n − 1 is a prime. Unfortunately this theorem only applies in one direction (the inverse statement does not apply, no equivalence): that means that there exist prime exponent for which the Mersenne number is not prime (see the above example 211 − 1, where 11 is prime, but 211 − 1 not). Mersenne claimed that 267 − 1 is a prime number. There is also a mathematical history behind this claim: it first took over 200 years before Edouard Lucas (1842-1891) proved that this number is composite. However, he argued indirectly and did not name any of the factors. Then Frank Nelson Cole43 showed in 1903 which factors make up this composite number: 267 − 1 = 147, 573, 952, 589, 676, 412, 927 = 193, 707, 721 · 761, 838, 257, 287. He admitted to having worked 20 years on the factorisation (expression as a product of prime factors)44 of this 21-digit decimal number! Due to the fact that the exponents of the Mersenne numbers do not use all natural numbers, but only the primes, the experimental space is limited considerably. The currently known Mersenne

43 44

Frank Nelson Cole, American mathematician, Sep. 20, 1861 – May 26, 1926. Using CrypTool you can factorize numbers in the following way: menu Indiv. Procedures \ RSA Cryptosystem \ Factorisation of a Number. CrypTool can factorize in a reasonable time numbers no longer than 250 bit. Numbers bigger than 1024 bits are currently not accepted by CrypTool. The current factorization records are listed in chapter 4.11.4.

prime numbers have the exponents[45] 2; 3; 5; 7; 13; 17; 19; 31; 61; 89; 107; 127; 521; 607; 1, 279; 2, 203; 2, 281; 3, 217; 4, 253; 4, 423; 9, 689; 9, 941, 11, 213; 19, 937; 21, 701; 23, 207; 44, 497; 86, 243; 110, 503; 132, 049; 216, 091; 756, 839; 859, 433; 1, 257, 787; 1, 398, 269; 2, 976, 221; 3, 021, 377; 6, 972, 593; 13, 466, 917; 20, 996, 011; 24, 036, 583; 25, 964, 951; 30, 402, 457. Thus 43 Mersenne prime numbers are currently known. For the first 39 Mersenne prime numbers we know that this list is complete. The exponents until the 40th Mersenne prime number have not yet been checked completely[46] . The 19th number with the exponent 4, 253 was the first with at least 1, 000 digits in decimal system (the mathematician Samual Yates coined the expression titanic prime for this; it was discovered by Hurwitz in 1961); the 27th number with the exponent 44, 497 was the first with at least 10, 000 digits in the decimal system (Yates coined the expression gigantic prime for this. These names are now long outdated).

M-37 – January 1998 The 37th Mersenne prime, 23,021,377 − 1 was found in January 1998 and has 909,526 digits in the decimal system, which corresponds to 33 pages in the newspaper!

M-38 – June 1999 The 38th Mersenne prime, called M-38, 26,972,593 − 1 was discovered in June 1999 and has 2, 098, 960 digits in the decimal system (that corresponds to around 77 pages in the newspaper).

M-39 – December 2001 The 39th Mersenne prime, called M-39, 213,466,917 − 1, was published at December 6, 2001 – more exactly, the verification of this number, found at November 14, 2001 by the Canadian student Michael Cameron, was successfully completed. This

---

[45] The following page from Landon Curt Noll contains all Mersenne primes including its date of discovery and its value as number and as word: http://www.isthe.com/chongo/tech/math/prime/mersenne.html. Also see: http://www.utm.edu/.

[46] The current status of the check can be found at: http://www.mersenne.org/status.htm. Hints, how the primality of a number can be checked, are in chapter 3.5, prime number tests.

number has about 4 million decimal digits (exactly 4,053,946 digits). Trying only to print this number (924947738006701322247758 · · · 1130073855470256259071) would require around 200 pages in the Financial Times. Right now (May 2005) all prime exponents smaller than 13.466.917 have been tested and doublechecked (see home page of the GIMPS project: http://www.mersenne.org): so we can be certain, that this is really the 39th Mersenne prime number and that there are no smaller undiscovered Mersenne primes (it is common usage to use the notation M-nn not until it is proven, that the nn-th known Mersenne prime is really the nn-th Mersenne prime).

GIMPS The GIMPS project (Great Internet Mersenne Prime Search) was founded in 1996 by George Woltman to search for new largest Mersenne primes (http://www.mersenne.org). Further explanations about this number type can be found under Mersenne numbers and Mersenne primes. Right now the GIMPS project has discovered nine largest Mersenne primes so far, including the largest known prime number at all. The following table contains these Mersenne record primes47,48 : Definition 230,402,457 − 1 225,964,951 − 1 224,036,583 − 1 220,996,011 − 1 213,466,917 − 1 26,972,593 − 1 23,021,377 − 1 22,976,221 − 1 21,398,269 − 1 Decimal Digits 9,152,052 7,816,230 7,235,733 6,320,430 4,053,946 2,098,960 909,526 895,932 420,921 When December 15, 2005 February 18, 2005 May 15, 2004 November 17, 2003 November 14, 2001 June 1, 1999 January 27, 1998 August 24, 1997 November 1996 Who Curtis Cooper/Steven Boone Martin Nowak Josh Findley Michael Shafer Michael Cameron Nayan Hajratwala Roland Clarkson Gordon Spence Joel Armengaud

Table 21: The largest primes found by the GIMPS project (as of March 2006) Dr. Richard Crandall discovered the advanced transform algorithm used by the GIMPS program. George Woltman implemented Crandall’s algorithm in machine language, thereby producing a prime-search program of unprecedented efficiency, and that work led to the successful GIMPS project. On June 1st, 2003 a possible Mersenne prime was reported to the GIMPS server, which was checked afterwards as usual, before it was to be published. Unfortunately mid June the initiator

47 48

An up-to-date version can be found in the internet at http://www.mersenne.org/history.htm. Always, when a new record is published in the respective forums the same and often ironic discussions start: Is there a deeper sense? Can this result be applied for anything useful? The answer is, that we don’t know it yet. In fundamental research one cannot see at once how it brings mankind forward.

49

and GIMPS project leader George Woltman had to tell, that two independent verification runs proved the number was composite. This was the first false positive report of a client in 7 years. Now more than 130,000 volunteers, amateurs and experts, participate in the GIMPS project. They connect their computers into the so called “primenet”, organized by the company entropia.

3.4.3

Challenge of the Electronic Frontier Foundation (EFF)

This search is also spurred on by a competition started by the non-profit organisation EFF (Electronic Frontier Foundation) using the means of an unknown donator. The participants are rewarded with a total of 500,000 USD if they find the longest prime number. In promoting this project, the unknown donator is not looking for the quickest computer, but rather wants to draw people’s attention to the opportunities offered by cooperative networking http://www.eff.org/coopawards/prime-release1.html The discoverer of M-38 received 50,000 USD from the EFF for discovering the first prime with more than 1 million decimal digits. The next prize of 100,000 USD offered by EFF is for a proven prime with more than 10 million decimal digits. According to the EFF rules for their prizes they offer in the next stage 150,000 USD for a proven prime with more than 100 million decimal digits. Edouard Lucas (1842-1891) held the record for the longest prime number for over 70 years by proving that 2127 − 1 is prime. No new record is likely to last that long.

3.5

Prime number tests

In order to implement secure encryption procedures we need extremely large prime numbers (in the region of 22,048 , i.e. numbers with 600 digits in the decimal system!). If we look for the prime factors in order to decide whether a number is prime, then the search takes too long, if even the smallest prime factor is enormous. Factorising numbers using systematic computational division or using the sieve of Eratosthenes is only feasible using current computers for numbers with up to around 20 digits in the decimal system. The biggest number factorized into its 2 almost equal prime factors has 200 digits (see RSA-200 in chapter 4.11.4). However, if we know something about the construction of the number in question, there are extremely highly developed procedures that are much quicker. These procedures can determine the primality attribute of a number, but they cannot determine the prime factors of a number, if it is compound. In the 17th century, Fermat[49] wrote to Mersenne that he presumed that all numbers of the form f (n) = 22 + 1 are prime for all whole numbers n ≥ 0 (see below, chapter 3.6.4).
n

---

[49] Pierre de Fermat, French mathematician, Aug 17, 1601 – Jan 12, 1665.

As early as in the 19th century, it was discovered that the 29-digit number f (7) = 22 + 1 is not prime. However, it was not until 1970 that Morrison/Billhart managed to factorise it. f (7) = 340, 282, 366, 920, 938, 463, 463, 374, 607, 431, 768, 211, 457 = 59, 649, 589, 127, 497, 217 · 5, 704, 689, 200, 685, 129, 054, 721
7

Despite Fermat was wrong with this supposition, he is the originator of an important theorem in this area: Many rapid prime number tests are based on the (little) Fermat theorem put forward by Fermat in 1640 (see chapter 4.8.3). Theorem 3.5 (“little” Fermat). Let p be a prime number and a be any whole number, then for all a ap ≡ a mod p. This could also be formulated as follows: Let p be a prime number and a be any whole number that is not a multiple of p (also a ≡ 0 mod p), then ap−1 ≡ 1 mod p. If you are not used to calculate with remainders (modulo), please simply accept the theorem or first read chapter 4 “Introduction to Elementary Number Theory with Examples”. What is important here is that this sentence implies that if this equation is not met for any whole number a, then p is not a prime! The tests (e.g. for the first formulation) can easily be performed using the test basis a = 2. This gives us a criterion for non-prime numbers, i.e. a negative test, but no proof that a number a is prime. Unfortunately Fermat’s theorem does not apply – otherwise we would have a simple proof of the prime number property (or to put it in other words, we would have a simple prime number criterion).

Pseudo prime numbers Numbers n that have the property 2n ≡ 2 mod n but are not prime are called pseudo prime numbers (i.e. the exponent is not a prime). The first pseudo prime number is 341 = 11 · 31.

Carmichael numbers There are pseudo prime numbers n that pass the Fermat test an−1 ≡ 1 mod n with all bases a which are relatively prime to n [gcd(a, n) = 1], despite these numbers n are not prime: These numbers are called Carmichael numbers. The first of these is 561 = 3 · 11 · 17. Sample: The number to be tested is 561. Because 561 = 3 · 11 · 17 it is: The test condition a560 mod 561 = 1 is satified for a = 2, 4, 5, 7, · · · , but not for a = 3, 6, 9, 11, 12, 15, 17, 18, 21, 22, · · · . This means the test condition must not be satisfied for multiples of the prime factors 3, 11 or 17. The test applied for a = 3 results in: 3560 mod 561 = 375. The test applied for a = 5 results in: 5560 mod 561 = 1.

Strong pseudo prime numbers A stronger test is provided by Miller/Rabin50 : it is only passed by so-called strong pseudo prime numbers. Again, there are strong pseudo prime numbers that are not primes, but this is much less often the case than for (simple) pseudo prime numbers or for Carmichael numbers. The smallest strong pseudo prime number base 2 is 15, 841 = 7 · 31 · 73. If you test all 4 bases, 2, 3, 5 and 7, you will find only one strong pseudo prime number up to 25 · 109 , i.e. a number that passes the test and yet is not a prime number. More extensive mathematics behind the Rabin test delivers the probability that the number examined is prime (such probabilities are currently around 10−60). Detailed descriptions of tests for finding out whether a number is prime can be found on Web sites such as: http://www.utm.edu/research/primes/mersenne.shtml http://www.utm.edu/research/primes/prove/index.html

50

In 1976 an efficient probabilistic primality test was published by Prof. Rabin, based on a number theoretic result of Prof. Miller from the year before. Prof. Miller worked at the Carnegie-Mellon University, School of Computer Science. Prof. Rabin, born in 1931, worked at the Harvard and Hebrew University.

3.6

Overview special number types and the search for a formula for primes

There are currently no useful, open (i.e. not recursive) formulae known that only deliver prime numbers (recursive means that in order to calculate the function the same function is used with a smaller variable). Mathematicians would be happy if they could find a formula that leaves gaps (i.e. does not deliver all prime numbers) but does not deliver any composite (non-prime) numbers. Ideally, we would like, for the number n, to immediately be able to obtain the n-th prime number, i.e. for f (8) = 19 or for f (52) = 239. Ideas for this can be found at http://www.utm.edu/research/primes/notes/faq/p n.html. Cross-reference: the table under 3.8.6 contains the precise values for the nth prime numbers for selected n. For “prime number formulae” usually very special types of numbers are used. The following enumeration contains the most common ideas for “prime number formulae”, and what our current knowledge is about very big elements of the number series: Is their primality proven? If their are compound numbers could their prime factors be determined? 3.6.1 Mersenne numbers f (n) = 2n − 1 for n prime

As shown above, this formula seems to deliver relatively large prime numbers but - as for n = 11 [f (n) = 2, 047] - it is repeatedly the case that the result even with prime exponents is not prime. Today, all the Mersenne primes having less than around 4,000,000 digits are known (M-39): http://perso.wanadoo.fr/yves.gallot/primes/index.html 3.6.2 Generalized Mersenne numbers f (k, n) = k · 2n ± 1 for n prime and k small prime

For this first generalisation of the Mersenne numbers there are (for small k) also extremely quick prime number tests (see [Knuth1981]). This can be performed in practice using software such as the Proths software from Yves Gallot http://www.prothsearch.net/index.html. 3.6.3 Generalized Mersenne numbers f (b, n) = bn ± 1 / The Cunningham project

This is another possible generalisation of the Mersenne numbers. The Cunningham project determines the factors of all composite numbers that are formed as follows: f (b, n) = bn ± 1 for b = 2, 3, 5, 6, 7, 10, 11, 12

(b is not equal to multiples of bases already used, such as 4, 8, 9). Details of this can be found at: 53

http://www.cerias.purdue.edu/homes/ssw/cun 3.6.4 Fermat numbers51 f (n) = 22 + 1
n

As mentioned above in chapter 3.5, Fermat wrote to Mersenne regarding his assumption, that all numbers of this type are primes. This assumption was disproved by Euler (1732). The prime 641 divides f (5)52 . f (0) = 22 1 f (1) = 22 2 f (2) = 22 3 f (3) = 22 4 f (4) = 22 5 f (5) = 22 6 f (6) = 22 f (7) =
7 22 0

+ 1 = 21 + 1 + 1 = 22 + 1 + 1 = 24 + 1 + 1 = 28 + 1 + 1 = 216 + 1 + 1 = 232 + 1 +
1 = 264 + 1

=3 =5 = 17 = 257 = 65,537 = 4,294,967,297 = 641 · 6,700,417 =
18,446,744,073,709,551,617 = 274,177 · 67,280,421,310,721 128 + 1 = (see page 51)
+1=2

→ → → → → →

prime prime prime prime prime NOT prime!

→ NOT prime! → NOT prime!

Within the project “Distributed Search for Fermat Number Dividers” offered by Leonid Durman there is also progress in finding new monster primes: http://www.fermatsearch.org/ This website links to other webpages in Russian, Italian and German. The discovered factors can be compound integers or primes. On February 22, 2003 John Cosgrave discovered • the largest composite Fermat number to date and • the largest prime non-simple Mersenne number so far with 645,817 decimal digits. The Fermat number f (2, 145, 351) = 2(2 is divisible by the prime p = 3 ∗ 22,145,353 + 1 At that time this prime p was the largest known prime generalized Mersenne number and the 5th largest known prime number at all. This work was done using NewPGen from Paul Jobling’s, PRP from George Woltman’s, Proth from Yves Gallot’s programs and also the Proth-Gallot group at St. Patrick’s College, Dublin. More details are in http://www.fermatsearch.org/history/cosgrave record.htm/
2,145,351)

+1

The Fermat prime numbers play a role in circle division. As proven by Gauss a regular p-edge can only be constructed with the use of a pair of compasses and a ruler, when p is a Fermat prime number. 52 Surprisingly this number can easily be found by using Fermat’s theorem (see e.g. [Scheid1994, p. 176])

51

3.6.5

Generalized Fermat numbers53 f (b, n) = b2 + 1

n

Generalized Fermat numbers are more numerous than Mersenne numbers of a equal size and many of them are waiting to be discovered to fill the big gaps between the Mersenne primes already found or still undiscovered. Progress in number theory made it possible that numbers, where the representation is not limited to the base 2, can be tested at almost the same speed than a Mersenne number. Yves Gallot wrote the program Proth.exe to investigate generalized Fermat numbers. Using this program at February 16, 2003 Michael Angel discovered the largest of them till then with 628,808 digits, which at that time became the 5th largest known prime number: b2 + 1 = 62, 722131,072 + 1. More details are in http://primes.utm.edu/top20/page.php?id=12
17

3.6.6

Carmichael numbers

As mentioned above in chapter 3.5 not all Carmichael numbers are prime. 3.6.7 Pseudo prime numbers

See above in chapter 3.5. 3.6.8 Strong pseudo prime numbers

See above in chapter 3.5. 3.6.9 Idea based on Euclid’s proof p1 · p2 · · · pn + 1

This idea is based on Euclid’s proof that there are infinite many prime numbers.
2·3 + 1 2·3·5 + 1 2·3·5·7 + 1 2·3· · ·11 + 1 2 · 3 · · · 13 + 1 2 · 3 · · · 17 + 1
=7 = 31 = 211 = 2, 311 = 59 · 509 = 19 · 97 · 277 → → → → → → prime prime prime
prime NOT prime! NOT prime!

53

The base of this power is no longer restricted to 2 . n Even more generic would be: f (b, c, n) = bc ± 1

3.6.10

As above but −1 except +1: p1 · p2 · · · pn − 1 2·3−1 2·3·5−1 2 · 3···7 − 1 2 · 3 · · · 11 − 1 2 · 3 · · · 13 − 1 2 · 3 · · · 17 − 1 =5 = 29 = 11 · 19 = 2, 309 = 30, 029 = 61 · 8, 369 → → → → → → prime prime NOT prime! prime prime NOT prime!

3.6.11

Euclidean numbers en = e0 · e1 · · · en−1 + 1 with n ≥ 1 and e0 := 1

en−1 is not the (n − 1)th prime number, but the number previously found here. Unfortunately this formula is not open but recursive. The sequence starts with

e1 e2 e3 e4 e5 e6 e7 e8

=1+1 = e1 + 1 = e1 · e2 + 1 = e1 · e2 · e3 + 1 = e1 · e2 · · · e4 + 1 = e1 · e2 · · · e5 + 1 = e1 · e2 · · · e6 + 1 = e1 · e2 · · · e7 + 1

=2 =3 =7 = 43 = 13 · 139 = 3, 263, 443 = 547 · 607 · 1, 033 · 31, 051 = 29, 881 · 67, 003 · 9, 119, 521 · 6, 212, 157, 481

→ → → → → → → →

prime prime prime prime NOT prime! prime NOT prime! NOT prime!

e9 , · · · , e17 are also composite, which means that this formula is not particularly useful. Remark: However, what is special about these numbers is that any pair of them does not have a common factor other than 154 . Therefore they are relatively prime. 3.6.12 f (n) = n2 + n + 41

This sequence starts off very promisingly, but is far from being a proof.

54

This can easily be shown via the following greatest common divisor (gcd) rule gcd(a, b) = gcd(b − b/a , a) (see page 136): We have for i < j: gcd(ei , ej) ≤ gcd(e1 · · · ei · · · ej−1 , ej) = gcd(ej − e1 · · · ei · · · ej−1 , e1 · · · ei · · · ej−1) = gcd(1, e1 · · · ei · · · ej−1) = 1.

56

f (0) = 41 f (1) = 43 f (2) = 47 f (3) = 53 f (4) = 61 f (5) = 71 f (6) = 83 f (7) = 97 . . . f (33) = 1, 163 f (34) = 1, 231 f (35) = 1, 301 f (36) = 1, 373 f (37) = 1, 447 f (38) = 1, 523 f (39) = 1, 601 f (40) = 1681 = 41 · 41 f (41) = 1763 = 41 · 43

→ → → → → → → → → → → → → → → → →

prime prime prime prime prime prime prime prime prime prime prime prime prime prime prime NOT prime! NOT prime!

The first 40 values are prime numbers (which have the obvious regularity that their difference starts with 2 and increases by 2 each time), but the 41th and 42th values are not prime numbers. It is easy to see that f (41) cannot be a prime number: f (41) = 412 +41+41 = 41(41+1+1) = 41·43.

3.6.13

f (n) = n2 − 79 · n + 1, 601

This function delivers prime numbers for all values from n = 0 to n = 79. Unfortunately f (80) = 1, 681 = 11 · 151 is not a prime number. To this date, no function has been found that delivers more prime numbers in a row. On the other hand, each prime occurs twice (first in the decreasing then in the increasing sequence), which means that the algorithm delivers a total of 40 difference prime values (the same ones as delivered by the function in chapter 3.6.12). f (0) = 1, 601 f (1) = 1, 523 f (2) = 1, 447 f (3) = 1, 373 f (4) = 1, 301 f (5) = 1, 231 f (6) = 1, 163 f (7) = 1, 097 f (8) = 1, 033 f (9) = 971 f (10) = 911 f (11) = 853 f (12) = 797 f (13) = 743 f (14) = 691 f (15) = 641 f (16) = 593 f (17) = 547 f (18) = 503 f (19) = 461 f (20) = 421 f (21) = 383 f (22) = 347 f (21) = 383 f (22) = 347 f (23) = 313 f (24) = 281 f (25) = 251 f (26) = 223 f (27) = 197 → prime f (28) = 173 f (29) = 151 f (30) = 131 f (31) = 113 f (32) = 97 f (33) = 83 f (34) = 71 f (35) = 61 f (36) = 53 f (37) = 47 f (38) = 43 f (39) = 41 f (40) = 41 f (41) = 43 f (42) = 47 f (43) = 53 ··· f (77) = 1, 447 f (78) = 1, 523 f (79) = 1, 601 f (80) = 41 · 41 f (81) = 41 · 43 f (82) = 1, 847 f (83) = 1, 933 f (84) = 43 · 47 → prime prime prime prime prime prime prime prime prime prime prime prime prime prime prime prime prime prime prime NOT prime! NOT prime! prime prime NOT prime!

3.6.14

Polynomial functions f (x) = an xn + an−1 xn−1 + · · · + a1 x1 + a0

(ai in Z, n ≥ 1)

There exists no such polynomial that for all x in Z only delivers prime values. For a proof of this, please refer to [Padberg1996, p. 83 f.], where you will also find further details about prime number formulae.

This means there is no hope in looking for further formulae (functions) similar to that in chap. 3.6.12 or chap. 3.6.13. 3.6.15 Catalan’s conjecture55

Catalan conjectured that C4 is a prime: C0 C1 C2 C3 C4 = 2, = 2C0 = 2C1 = 2C2 = 2C3 − 1, − 1, − 1, − 1, · · ·

(see http://www.utm.edu/research/primes/mersenne.shtml under Conjectures and Unsolved Problems). This sequence is also defined recursively and increases extremely quickly. Does it only consist of primes? C(0) = 2 C(1) = 22 − 1 C(2) = 23 − 1 C(3) = 27 − 1 C(4) = 2127 − 1 =3 =7 = 127 = 170, 141, 183, 460, 469, 231, 731, 687, 303, 715, 884, 105, 727 → → → → → prime prime prime prime prime

It is not (yet) known whether C5 and all higher elements are prime, but this is not very likely. In any case, it has not been proved that this formula delivers only primes.

3.7

Density and distribution of the primes

As Euclid discovered, there is an infinite number of primes. However, some infinite sets are denser than others. Within the set of natural numbers, there is an infinite number of even, uneven and square numbers. The following proves that there are more even numbers than square ones: • the size of the nth element: The nth element of the even numbers is 2n; the nth element of the square numbers is n2 . Because for all n > 2: 2n < n2 , the nth even number occurs much earlier than the nth square number. Thus the even numbers are distributed more densely and we can say that there are more even numbers than square ones.

---

[55] Eugene Charles Catalan, Belgian mathematician, May 5, 1814–Feb 14, 1894. After him the so-called Catalan numbers A(n) = (1/(n + 1)) ∗ (2n)!/(n!)2 = 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, 742900, 2674440, 9694845, ... are named.

• the number of values that are less than or equal to a certain maximum value x in R is: √ There are [x/2] such even numbers and [x] square numbers. Because for large x the value x/2 is much greater than the square root of 2, we can again say that there are more even numbers.

The value of the n-th prime P (n) Theorem 3.6. For large n: The value of the n-th prime P (n) is asymptotic to n · ln(n), i.e. the limit of the relation P (n)/(n · ln n) is equal to 1 if n tends to infinity. For n > 5, P (n) lies between 2n and n2 . This means that there are fewer prime numbers than even natural numbers but more prime numbers than square numbers56 . The number of prime numbers P I(x) The definition is similar for the number of prime numbers P I(x) that do not exceed the maximum value x: Theorem 3.7. P I(x) is asymptotic to x/ln(x). This is the prime number theorem. It was put forward by Legendre57 and Gauss58 but not proved until over 100 years later. Cross-reference: The overview under 3.8.5 shows the number of prime numbers in various intervals. These formulae, which only apply when n tends to infinity, can be replaced by more precise formulae. For x ≥ 67: ln(x) − 1, 5 < x/P I(x) < ln(x) − 0, 5 Given that we know P I(x) = x/ ln x only for very large x (x tending towards infinity), we can create the following overview: x ln(x) x/ln(x) P I(x)(counted) P I(x)/(x/ln(x)) 103 6.908 144 168 1.160 106 13.816 72, 386 78, 498 1.085 109 20.723 48, 254, 942 50, 847, 534 1.054 For a binary number59 x of the length of 250 bits (2250 is approximately = 1.809251 ∗ 1075) it is: P I(x) = 2250 /(250 · ln 2) is approximately = 2250 /173.28677 = 1.045810 · 1073 .

56 57

Please refer to the table 3.8.6 Adrien-Marie Legendre, French mathematician, Sep 18, 1752 − Jan 10, 1833. 58 Carl Friedrich Gauss, German mathematician and astronomer, Apr 30, 1777−Feb 23, 1855. 59 Number written in the binary system consists only of the digits 0 and 1.

We can therefore expect that the set of numbers with a bit length of less than 250 contains approximately 1073 primes (a reassuring result?!). We can also express this as follows: Let us consider a random natural number n. Then the probability that this number is prime is around 1/ ln(n). For example, let us take numbers in the region of 1016 . Then we must consider 16 · ln 10 = 36, 8 numbers (on average) until we find a prime. A precise investigation shows: There are 10 prime numbers between 1016 − 370 and 1016 − 1. Under the heading How Many Primes Are There at http://www.utm.edu/research/primes/howmany.shtml you will find numerous other details. Using the following Web site: http://www.math.Princeton.EDU/~arbooker/nthprime.html you can easily determine P I(x). The distribution of primes displays several irregularities for which no “system” has yet been found: On the one hand, many occur closely together, like 2 and 3, 11 and 13, 809 and 811, on the other hand large gaps containing no primes also occur. For example, no primes lie between 113 and 127, 293 and 307, 317 and 331, 523 and 541, 773 and 787, 839 and 853 as well as between 887 and 907. For details, please see: http://www.utm.edu/research/primes/notes/gaps.html This is precisely part of what motivates mathematicians to discover its secrets. Sieve of Eratosthenes An easy way of calculating all P I(x) primes less than or equal to x is to use the sieve of Eratosthenes. In the 3rd century B.C., he found an extremely easy, automatic way of finding this out. To begin with, you write down all numbers from 2 to x, circle 2, then cross out all multiples of 2. Next, you circle the lowest number that hasn’t been circled or crossed out (3) and again cross out all multiples of this number, etc. You only need to continue until you reach the largest number whose square is less than or equal to x. Apart from 2, prime numbers are never even. Apart from 2 and 5, prime numbers never end in 2, 5 or 0. So you only need to consider numbers ending in 1, 3, 7, 9 anyway (there are infinite primes ending in these numbers; see [Tietze1973, vol. 1, p. 137]). You can now find a large number of finished programs on the Internet - often complete with source code - allowing you to experiment with large numbers yourself (see chapter 3.6). You also have access to large databases that contain either a large number of primes or the factorisation of numerous composite numbers.

3.8

Notes about primes

Further interesting topics regarding prime numbers This chapter 3 didn’t consider other number theory topics such as divisibility rules, modulus calculation, modular inverses, modular powers, modular roots, Chinese remainder theorem, Euler Phi function or perfect numbers. Some of these topics are considered in the next chapter (chapter 4). The following notes list some interesting theorems, conjectures and open questions about primes, but also some quaint things and overviews. 3.8.1 Proven statements / theorems about primes

• For each number n in N there are n consecutive natural numbers that are not primes. A proof of this can be found in [Padberg1996, p. 79]. • Paul Erd¨s60 proved: Between each random number not equal to 1 and its double, there o is at least one prime. He was not the first to prove this theorem, but proved it in a much simpler manner than those before him. • There is a real number a such that the function f : N → Z where n → a3 only delivers primes for all n (see [Padberg1996, p. 82]). Unfortunately, problems arise when we try to determine a (see below). • There are arithmetic prime sequences of arbitrary length61 . In 1923 the famous British mathematician Godfrey Harold Hardy62 compiled the conjecture, that there are arithmetic sequences of arbitrary length, which consist of primes only. This conjecture was proven in 2004 by two young American mathematicians. At some point every school child learns about arithmetic number series. These are sequences of numbers, for which the difference between any 2 consecutive numbers is equal or constant (an arithmetic sequence must have at least three elements but can also have indefinitely many). In the sample sequence 5, 8, 11, 14, 17, 20 the difference between the serie’s elements is 3 and the length of the sequence is 6. Arithmetic series have been known for millennia and one would think they have no more secrets. They get more interesting again, if we impose additional constraints on the series elements - as the prime example shows. E.g. 5, 17, 29, 41, 53 is an arithmetic prime series which consists of 5 elements and the difference between the elements is always 12.
n

61

Paul Erd¨s, Hungarian mathematician, Mar 26, 1913–Sep 20, 1996. o Sources: - http://primes.utm.edu/glossary/page.php?sort=ArithmeticSequence Original source - German magazine GEO 10 / 2004: “Experiment mit Folgen” - http://www.faz.net “Hardys Vermutung – Primzahlen ohne Ende” by Heinrich Hemme (July 06, 2004) 62 Godfrey Harold Hardy, British mathematician, Feb 7, 1877–Dec 1, 1947.

60

62

The sequence is not extendable - the next would be 65, but 65 is not prime (65 is the product of 5 and 13). How many elements are possible within an arithmetic prime number sequence? Around 1770 the French Joseph-Louis Lagrange and the British Edward Waring investigated this question. In 1923 the famous British mathematician Godfrey Harold Hardy and his colleague John Littlewood theorized, that there is no upper limit for the number of elements. But they could not prove this. In 1939 more progress was achieved. The Dutch mathematician Johannes van der Corput was able to prove that there are infinitely many different arithmetic prime number sequences with exactly three elements. Two examples are 3, 5, 7 and 47, 53, 59. The longest arithmetic prime number sequence known today contains 23 elements. # of elements 22 First element 11,410,337,850,553 Distance 4,609,098,694,200 When 1993 Discovered by Paul A. Pritchard, Andrew Moran, Anthony Thyssen Markus Frind Markus Frind, Paul Jobling, Paul Underwood

22 23

376,859,931,192,959 56,211,383,760,397

18,549,279,769,020 44,546,738,095,860

2003 2004

Table 22: The longest arithmetic prime number sequences (as of May 2005) As a team, the two young[63] mathematicians Ben Green and Terence Tao, were able in 2004 to prove Hardy’s conjecture, which had puzzled mathematicians for over 80 years: It states, that for any arbitrary length there exists an arithmetic prime number series. Additionally they managed to prove, that for any given length there are infinitely many different series. Green and Tao intended to proof that there are infinitely many arithmetic sequences of length four. For this they considered sets of numbers consisting of primes and so called “near primes”. These are numbers with a small set of divisors like numbers which are the product of two primes - these numbers are called “half primes” . Thus they managed to considerably simplify their work because about near primes there already existed a lot of useful theorems. Finally they discovered that the results of their theorem were far more reaching than they had assumed and so they were able to prove Hardy’s conjecture. Any one who believes that it is easy to use Green’s and Tao’s 49 page proof to compute arithmetic prime number series of arbitrary length will soon become disappointed, because

---

[63] Hardy wrote in his memoirs in 1940, that mathematics - more than all other arts and sciences - is an activity for young people. At that time 27-years-old Ben Green from the University of British Columbia in Vancouver and 29-year-old Terence Tao from the University of California in Los Angeles seem to confirm Hardy.

the proof is non-constructive. It is a so called proof of existance. This means that these mathematicians have shown “only” that these series exist, but not how to find them in practice. This means that in the set of the natural numbers there is e.g. a series of one billion primes, which all have the same distance; and there are infinitely many of them. But these sequences lie extremely far beyond the numbers we usually use (“far outside”). If someone wants to discover such sequences he should consider the following thought. The length of a sequence determines the minimal common distance between the single primes of the sequence. Given a sequence with 6 elements the distance between them has to be 30 or a multiple of 30. The number 30 results as the product of all primes smaller than the length of the sequence. So its the product of all primes smaller than 6: 2 ∗ 3 ∗ 5 = 30. If you look for a sequence with 15 elements, then the common distance is at least 2∗3∗5∗7∗11∗13 = 30.030. This means that the length of an arithmetic prime sequence can be arbitrary big, but the distance between the elements cannot be any arbitrary number. E.g. there is no arithmetic prime sequence with the distance 100, because 100 cannot be divided by 3. If you take the sequences above (with the lengths of 22 and 23) and look at the factors of their distances, you get: 4, 609, 098, 694, 200 = 23 ∗ 3 ∗ 52 ∗ 7 ∗ 11 ∗ 13 ∗ 17 ∗ 19 ∗ 23 ∗ 1033 18, 549, 279, 769, 020 = 22 ∗ 3 ∗ 5 ∗ 72 ∗ 11 ∗ 13 ∗ 17 ∗ 19 ∗ 23 ∗ 5939 44, 546, 738, 095, 860 = 22 ∗ 3 ∗ 5 ∗ 7 ∗ 11 ∗ 13 ∗ 17 ∗ 19 ∗ 23 ∗ 99, 839 Further restriction: If you look at arithmetic prime sequences, which fulfill the additional requirement, that all primes are consecutive, then its getting even more complicated. At the website of Chris Caldwell64 you can find further hints: the longest known arithmetic prime sequence, consisting only of directly consecutive primes, has a length of 10 and the distance is 210 = 2 ∗ 3 ∗ 5 ∗ 7.

3.8.2

Unproven statements / conjectures about primes

- Christian Goldbach65 conjectured: Every even natural number greater than 2 can be represented as the sum of two prime numbers. Computers have verified66 the Goldbach conjecture

64 65

http://primes.utm.edu/glossary/page.php?sort=ArithmeticSequence Christian Goldbach, German mathematician, Mar 18, 1690–Nov 20, 1764. 66 It is generally accepted today, that the Goldbach Conjecture is true, i. e. valid for all even natural numbers greater than 2. In 1999, mathematician J¨rg Richstein from the computer sciences institute at the o University of Giessen, studied even numbers up to 400 billion and found no contradictory example (see http://www.mscs.dal.ca/~joerg/res/g-en.html, http://en.wikipedia.org/wiki/Goldbach’s conjecture, http://primes.utm.edu/glossary/page.php/GoldbachConjecture.html).

for all even numbers up to 4 ∗ 1014 but no general proof has yet been found67 . • Bernhard Riemann68 put forward a formula for the distribution of primes that would further improve the estimate. However, this has neither been proved nor disproved so far.

3.8.3

Open questions

Twin primes are prime numbers whose difference is 2. Examples include 5 and 7 or 101 and 103. Triplet primes, however, only occur once: 3, 5, 7. For all other sets of three consecutive uneven numbers, one of them is always divisible by 3 and thus not a prime. • The number of twin primes is an open question: infinite or limited number? The largest twin primes known today are 1, 693, 965 · 266,443 ± 1. • Does a formula exist for calculating the number of twin primes per interval? • The above proof of the function f : N → Z with n → a3 only guarantees the existence of such a number a. How can we determine this number a and will it have a value, making the function also of some practical interest? • Is there an infinite number of Mersenne prime numbers? • Is there an infinite number of Fermat prime numbers? • Does a polynomial time algorithm exist for calculating the prime factors of a number (see [Klee1997, p. 167])? This question can be divided into the three following questions: – Does a polynomial time algorithm exist that decides whether a number is prime? This question has been answered by the AKS algorithm (see chapter 4.11.5, “Primes in P”: Primality testing is polynomial).
n

Nevertheless, this does not provide us with general proof. The fact is that despite all efforts, Goldbach’s conjecture has to date not been proven. This leads one to believe that since the pioneer work of the Austrian mathematician Kurt G¨del is well-known, not every true mathematical o theorem is provable (see http://www.mathematik.ch/mathematiker/goedel.html). Perhaps Goldbach’s conjecture was correct, but in any case the proof will never be found. Conversely, that will presumably also remain unproven. 67 The English publisher Faber and the American publisher Bloomsbury issued in 2000 the 1992 published book “Uncle Petros and Goldbach’s Conjecture” by Apostolos Doxiadis. It’s the story of an old maths professor who fails to prove a more than 250 year old puzzle. To boost the sales figures the English and American publishers have offered a prize of 1 million USD, if someone can prove the conjecture – which should be published by 2004 in a well-known mathematical journal. Surprisingly only British and American citizens are allowed to participate. The theorem which has come closest so far to Goldbach’s conjecture was proved by Chen Jing-Run in 1966 in a way which is somewhat hard to understand: Each even integer greater than 2 is the sum of one prime and of the product of two primes. E.g.: 20 = 5 + 3 ∗ 5. Most of the research about the Goldbach conjecture is collected in the book: “Goldbach Conjecture”, ed. Wang Yuan, 1984, World scientific Series in Pure Maths, Vol. 4. Especially this conjecture makes it clear, that even today we do not have a complete understanding of the deeper connections between addition and multiplication of natural numbers. 68 Bernhard Riemann, German mathematician, Sep 17, 1826–Jul 20, 1866.

– Does a polynomial time algorithm exist that calculates for a composite number from how many prime factors it is made up (without calculating these factors)? – Does a polynomial time algorithm exist that calculates for a composite number n a non-trivial (i.e. other than 1 and n) factor of n?[69] At the end of chapter 4.11.4, section RSA-200 you can see the dimensions of the numbers where the current algorithms testing for primality and calculating the factorization deliver results.

3.8.4

Quaint and interesting things around primes[70]

Primes are not only a very active and serious research area in mathematics. Also a lot of people think about them in their freetime and outside the scientific research.

Recruitment at Google in 2004 In summer 2004 the company Google used the number e[71] to attract potential employees[72] . On a promiment billboard in California’s Silicon Valley on July 12 there appeared the following mysterious puzzle: (first 10 digit prime in consecutve digits of e).com Finding the first 10 digit prime in the decimal expansion of e is not easy, but with various software tools, one can determine that the answer is 7, 427, 466, 391 Then if you visited the website www.7427466391.com, you were presented with an even more difficult puzzle. Figuring this second puzzle out took you to a web page that asks you, to submit your CV to Google. The ad campaign got high attention. Presumably Google’s conceit was that if you’re smart enough to solve the puzzles, you’re smart enough to work for them. Of course some days after the lauch, anyone who really wanted to

Please compare chapters 4.11.5 and 4.11.4. Further curious things about primes may be found at: - http://primes.utm.edu/curios/home.php - http://www.primzahlen.de/files/theorie/index.htm. 71 The base of the natural logarithm e is approximately 2.718 281 828 459. This is one of the most important numbers in all of mathematics like complex analysis, finance, physics and geometry. Now it was used the first time – as far as I know – for marketing or recruitment. 72 Most of this information is taken from the article “e-number crunching” by John Allen Paulos in TheGuardian, Sept. 30, 2004, and from the web: - http://www.mkaz.com/math/google/ - http://epramono.blogspot.com/2004/10/7427466391.html - http://mathworld.wolfram.com/news/2004-10-13/google/ - http://www.math.temple.edu/~paulos/.
70

69

discover the answers without incurring a headache could merely do a Google search for them, since many solvers immediately posted their solutions online.[73]

Contact [Directed by Robert Zemeckis, 1997] – Primes helping to contact aliens The movie originated from Carl Sagan’s book with the same title. After years of unavailing search the radio astronomer Dr. Ellie Arroway (Jodie Foster) discovers signals from the solar system Vega, 26 light years away. These signals contain the primes in the right order and without a gap. This makes the hero confident, that this message is different from the radio signals which permanently hit earth and which are random and of cosmic origin (radio galaxies, pulsars). In an unmasking scene a politician asks her after that, why these intelligent aliens didn’t just speak English ... Doing communication with absolute strange and unknown beings from deep space is very hard especially because of 2 reasons: First the big distance and therefore the long transfer time makes it impossible to exchange within an average lifetime more than one message in each direction one after the other. Secondly the first contact must give the receiver of the radio signals a good chance to notice the message and to categorize it as something from intelligent beings. Therefore the aliens send numbers at the beginning of their message, which can be considered as the easiest part of any higher language, and which are not too trivial: so they chose the sequence of primes. These special numbers play such a fundamental role in mathematics that one can assume that they are well known to each species who has the technical know-how to receive radio waves. The aliens then send a plan to build a mysterious machine ...

---

[73] The second level of the puzzle, which involved finding the 5th term of a given number sequence had nothing to do with primes any more.

3.8.5

Number of prime numbers in various intervals

Ten-sized Interval 1-10 11-20 21-30 31-40 41-50 51-60 61-70 71-80 81-90 91-100

intervals Number 4 4 2 2 3 2 2 3 2 1

Hundred-sized intervals Interval Number 1-100 25 101-200 21 201-300 16 301-400 16 401-500 17 501-600 14 601-700 16 701-800 14 801-900 15 901-1000 14

Thousand-sized intervals Interval Number 1-1000 168 1001-2000 135 2001-3000 127 3001-4000 120 4001-5000 119 5001-6000 114 6001-7000 117 7001-8000 107 8001-9000 110 9001-10000 112

Table 23: How many primes exist within the first intervals of tens?

Interval 1 - 10,000 1 - 100,000 1 - 1,000,000 1 - 10,000,000 1 - 100,000,000 1 - 1,000,000,000 1 - 10,000,000,000

Number 1,229 9,592 78,498 664,579 5,761,455 50,847,534 455,052,512

Average number per 1000 122.900 95.920 78.498 66.458 57.615 50.848 45.505

Table 24: How many primes exist within the first intervals of dimensions?

3.8.6

Indexing prime numbers (n-th prime number)

Index	Precise value	Rounded value
1	2	2
2	3	3
3	5	5
4	7	7
5	11	11
6	13	13
7	17	17
8	19	19
9	23	23
10	29	29
100	541	541
1,000	7,917	7,917
664,559	9,999,991	9.99999E+06
1E+06	15,485,863	1.54859E+07
6E+06	104,395,301	1.04395E+08
1E+07	179,424,673	1.79425E+08
1E+09	22,801,763,489	2.28018E+10
1E+12	29,996,224,275,833	2.99962E+13

Comment

All prime numbers up to 1E+07 were known at the beginning of the 20th century. This prime was discovered in 1959.

Table 25: List of particular n-th prime numbers

Comment: With gaps, extremely large prime numbers were discovered at an early stage.

Web links: http://www.math.Princeton.EDU/˜arbooker/nthprime.html. http://www.utm.edu/research/primes/notes/by year.html.

3.8.7

Orders of magnitude / dimensions in reality

In the description of cryptographic protocols and algorithms, numbers occur that are so large or so small that they are inaccessible to our intuitive understanding. It may therefore be useful to provide comparative numbers from the real world around us so that we can develop a feeling for the security of cryptographic algorithms. Some of the numbers listed below originate from [Schwenk1996] and [Schneier1996, p.18].

Probability that you will be hijacked on your next flight Annual probability of being hit by lightning Probability of 6 correct numbers in the lottery Risk of being hit by a meteorite Time until the next ice age (in years) Time until the sun dies (in years) Age of the earth (in years) Age of the universe (in years) Number of the earth’s atoms Number of the sun’s atoms Number of atoms in the universe (without dark material) Volume of the universe (in cm3)

5.5 · 10−6 10−7 7.1 · 10−8 1.6 · 10−12 14, 000 = (214) 109 = (230) 109 = (230) 1010 = (234) 1051 = (2170) 1057 = (2190) 1077 = (2265) 1084 = (2280)

Table 26: Likelihoods from physics and everyday life (dimensions)

3.8.8

Special values of the binary and decimal system Dual system 210 240 256 264 280 290 2112 2128 2150 2160 2250 2256 2320 2512 2768 21024 22048 Decimal system 1024 1.09951 · 1012 7.20576 · 1016 1.84467 · 1019 1.20893 · 1024 1.23794 · 1027 5.19230 · 1033 3.40282 · 1038 1.42725 · 1045 1.46150 · 1048 1.80925 · 1075 1.15792 · 1077 2.13599 · 1096 1.34078 · 10154 1.55252 · 10231 1.79769 · 10308 3.23170 · 10616

Table 27: Special values of the binary and decimal system

Calculation using GMP, for example: http://www.gnu.ai.mit.edu.

References

[Aaronson2003] Scott Aaronson, The Prime Facts: From Euclid to AKS, http://www.cs.berkeley.edu/~aaronson/prime.ps. Only after I had completed this article, did I come across the extremely well-written paper by Scott Aaronson, which also offers a didactically well done intoduction to this topic. It is humorous and easy to read but at the same time precise and erudite. [Bartholome1996] A. Bartholom¥, J. Rung, H. Kern, e Zahlentheorie f¨r Einsteiger, Vieweg 1995, 2nd edition 1996. u [Blum1999] W. Blum, Die Grammatik der Logik, dtv, 1999. [Bundschuh1998] Peter Bundschuh, Einf¨hrung in die Zahlentheorie, Springer 1988, 4th edition 1998. u [Doxiadis2000] Apostolos Doxiadis, Uncle Petros and the Goldbach’s Conjecture, Faber/Bloomsbury, 2000. [Graham1989] R.E. Graham, D.E. Knuth, O. Patashnik, Concrete Mathematics, Addison-Wesley, 1989. [Klee1997] V. Klee, S. Wagon, Ungel¨ste Probleme in der Zahlentheorie und der Geometrie der Ebene, o Birkh¨user Verlag, 1997. a [Knuth1981] Donald E. Knuth, The Art of Computer Programming, vol 2: Seminumerical Algorithms, Addison-Wesley, 1969, 2nd edition 1981. [Lorenz1993] F. Lorenz, Algebraische Zahlentheorie, BI Wissenschaftsverlag, 1993. [Padberg1996] F. Padberg, Elementare Zahlentheorie, Spektrum Akademischer Verlag 1988, 2nd edition 1996. [Pieper1983] H. Pieper, Zahlen aus Primzahlen, Verlag Harri Deutsch 1974, 3rd edition 1983. [Richstein1999] J. Richstein, Verifying the Goldbach Conjecture up to 4 ∗ 1014 , Mathematics of Computation 70, 2001, p. 1745-1749). [Scheid1994] Harald Scheid, Zahlentheorie, BI Wissenschaftsverlag, 2nd edition, 1994.

[Schneier1996] Bruce Schneier, Applied Cryptography, Protocols, Algorithms, and Source Code in C, Wiley and Sons, 2nd edition 1996. [Schroeder1999] M.R. Schroeder, Number Theory in Science and Communication, Springer 1984, 3rd edition 1997, Corrected Printing 1999. [Schwenk1996] J. Schwenk Conditional Access, in taschenbuch der telekom praxis 1996, Hrgb. B. Seiler, Verlag Schiele und Sch¨n, Berlin. o [Tietze1973] H. Tietze, Gel¨ste und ungel¨ste mathematische Probleme, o o Verlag C. H. Beck 1959, 6th edition 1973.

Web links

1. GIMPS (Great Internet Mersenne-Prime Search) www.mersenne.org is the home page of the GIMPS project, http://www.mersenne.org/prime.htm 2. The Proth Search Page with the Windows program by Yves Gallot http://www.utm.edu/research/primes/programs/gallot/index.html 3. Generalized Fermat Prime Search http://primes.utm.edu/top20/page.php?id=12 4. Distributed Search for Fermat Number Divisors http://www.fermatsearch.org/ 5. At the University of Tennessee you will find extensive research results about prime numbers. http://www.utm.edu/ 6. The best overview about prime numbers is offered from my point of view by “The Prime Pages” from professor Chris Caldwell. http://www.utm.edu/research/primes 7. Descriptions e.g. about prime number tests http://www.utm.edu/research/primes/mersenne.shtml http://www.utm.edu/research/primes/prove/index.html 8. Showing the n-th prime number http://www.utm.edu/research/primes/notes/by year.html 9. The supercomputer manufacturer SGI Cray Research not only employed brilliant mathematicians but also used the prime number tests as benchmarks for its machines. http://www.isthe.com/chongo/tech/math/prime/prime press.html 10. The Cunningham Project, http://www.cerias.purdue.edu/homes/ssw/cun/ 11. http://www.eff.org/coop-awards/prime-release1.html 12. http://www.math.Princeton.EDU/~arbooker/nthprime.html 13. http://www.cerias.purdue.edu/homes/ssw/cun 14. http://www.informatik.uni-giessen.de/staff/richstein/de/Goldbach.html 15. http://www.mathematik.ch/mathematiker/goedel.html 16. http://www.mscs.dal.ca/~dilcher/goldbach/index.html

Acknowledgments

I would like to take this opportunity to thank Mr. Henrik Koy and Mr. Roger Oyono for their very constructive proof-reading of this article.

4

Introduction to Elementary Number Theory with Examples

(Bernhard Esslinger, July 2001, Updates: Dec. 2001, June 2002, May 2003, May 2005, March 2006) This “introduction” is for people with a mathematical interest. There is no more pre-knowledge necessary than what you learn in the secondary school. We intentionally had “beginners” in mind; we did not take the approach of mathematical textbooks, called “introduction”, which cannot be understood at the first reading further than page 5 and which have the real purpose to deliver all information that special monographs can be read.

4.1

Mathematics and cryptography

A large proportion of modern, asymmetric cryptography is based on mathematical knowledge – on the properties (“laws”) of whole numbers, which are investigated in elementary number theory. Here, the word “elementary” means that questions raised in number theory are essentially rooted in the set of natural and whole numbers. Further mathematical disciplines currently used in cryptography include (see [Bauer1995, p. 2], [Bauer2000, p. 3]) : • Group theory • Combination theory • Complexity theory • Ergodic theory • Information theory. Number theory or arithmetic (the emphasis here is more on the aspect of performing calculations with numbers) was established by Carl Friedrich Gauss74 as a special mathematical discipline. Its elementary features include the greatest common divisor75 (gcd), congruence (remainder classes), factorisation, the Euler-Fermat theorem and primitive roots. However, the most important aspect is prime numbers and their multiplicative operation. For a long time, number theory was considered to be the epitome of pure research, the ideal example of research in the ivory tower. It delved into “the mysterious laws of the realm of numbers”, giving rise to philosophical considerations as to whether it described elements that exist everywhere in nature or whether it artificially constructed elements (numbers, operators and properties).

74 75

Carl Friedrich Gauss, German mathematician and astronomer, Apr 30, 1777–Feb 23, 1855. This article deals with the gcd (greatest common divisor) in Appendix A of this chapter.

We now know that patterns from number theory can be found everywhere in nature. For example, the ratio of rotating counterclockwise and rotating clockwise spirals in a sunflower is equal to two consecutive Fibonacci numbers[76], for example 21 : 34. Also, at the latest when number theory was applied in modern cryptography, it became clear that a discipline that had been regarded as purely theoretical for centuries actually had a practical use. Today, experts in this field are in great demand on the job market. Applications in (computer) security now use cryptography because this mathematical discipline is simply better and easier to prove than all other ”creative” substitution procedures that have been developed over the course of time and better than all sophisticated physical methods such as those used to print bank notes [Beutelspacher1996, p. 4]. This article explains the basics of elementary number theory in a way that you can easily understand. It provides numerous examples and very rarely goes into any proofs (these can be found in mathematical textbooks). The goal is not to exhaustively explain the number theory findings, but to show the essential procedures. The volume of the content is so oriented that the reader can understand and apply the RSA method. For this purpose we will use both theory and examples to explain how to perform calculations in finite sets and describe how these techniques are applied in cryptography. Particular attention will be paid to the traditional Diffie-Hellman (DH) and RSA public key procedures. Additionally I added some qualified statements about the security of the RSA algorithm.

Carl Friedrich Gauss: Mathematics is the queen of sciences and number theory is the queen of mathematics.

4.2

Introduction to number theory

Number theory arose from interest in positive whole numbers 1, 2, 3, 4, · · · , also referred to as the set of natural numbers natural numbers $\mathbb{N}$. These are the first mathematical constructs used by human civilisation. According to Kronecker[77], they are a creation of God. In Dedekind's[78] opinion, they are a creation of the human intellect. Dependent upon one's ideology, this is an unsolvable contradiction or one and the same thing. In ancient times, no distinction was made between number theory and numerology, which attributed a mystical significance to specific numbers. In the same way as astronomy and chem-

---

[76] The sequence of Fibonacci numbers $(a_i)_{i \in \mathbb{N}}$ is defined by the “recursive” rule $a_1 := a_2 := 1$ and for all numbers n = 1, 2, 3, · · · we define $a_{n+2} := a_{n+1} + a_n$. This historical sequence can be found in many interesting forms in nature (for example, see [Graham1994, p. 290 ff] or the website of Ron Knott, which is devoted to Fibonacci numbers). A lot is known about the Fibonacci sequence and it is used today as an important tool in mathematics.

[77] Leopold Kronecker, German mathematician, Dec 7, 1823 – Dec 29, 1891

[78] Julius Wilhelm Richard Dedekind, German mathematician, Oct 6, 1831 – Feb 12, 1916.

istry gradually detached themselves from astrology and alchemy during the Renaissance (from the 14th century), number theory also separated itself from numerology. Number theory has always been a source of fascination – for both amateurs and professional mathematicians. In contrast to other areas of mathematics, many of the problems and theorems in number theory can be understood by non-experts. On the other hand, mathematicians often take a long time to find solutions to the problems or prove the theorems. It is therefore one thing to pose good questions but quite another matter to find the answer. One example of this is what is known as Fermat's Last (or large) theorem[79] . Up until the mid 20th century, number theory was considered to be the purest area of mathematics, an area that had no practical use in the real world. This changed with the development of computers and digital communication, as number theory was able to provide several unexpected solutions to real-life tasks. At the same time, advances in information technology allowed specialists in number theory to make huge progress in factorising large numbers, finding new prime numbers, testing (old) conjectures and solving numerical problems that were previously impossible to solve. Modern number theory is made up of areas such as: • Elementary number theory • Algebraic number theory • Analytic number theory • Geometric number theory • Combinatorial number theory • Numeric number theory • Probability theory. All of the different areas are concerned with questions regarding whole numbers (both positive and negative whole numbers plus zero). However, they each have different methods of dealing with them. This article only deals with the area of elementary number theory.

---

[79] One of the things you learn in mathematics at school is Pythagoras theorem, which states the following for a right-angle triangle: $a^2 + b^2 = c^2$, where a and b are the lengths of the sides containing the right angle and c is the length of the hypotenuse. Fermat famously proposed that $a^n + b^n = c^n$ for whole-number exponents $n > 2$. Unfortunately, the letter in which Fermat made the claim did not have enough space for him to prove it. The theorem was not proved until over 300 years later [Wiles1994, p. 433-551].

4.2.1

Convention

Unless stated otherwise: • The letters $a, b, c, d, e, k, n, m, p, q$ are used to present whole numbers. • The letters i and j represent natural numbers. • The letters p always represents a prime number. • Te sets $N = \{1, 2, 3, \cdots\}$ and $Z = \{\cdots, -3, -2, -1, 0, 1, 2, 3, \cdots\}$ are the natural and whole numbers respectively.

Joanne K. Rowling80 This isn’t magic – it’s logic – a puzzle. A lot of the greatest wizards haven’t got an ounce of logic.

4.3

Prime numbers and the first fundamental theorem of elementary number theory

Many of the problems in elementary number theory are concerned with prime numbers. Every whole number has divisors or factors. The number 1 has just one – itself, whereas the number 12 has the six factors 1, 2, 3, 4, 6 and 1281 . Many numbers are only divisible by themselves and by 1. When it comes to multiplication, these can be regarded as the “atoms” in the realm of numbers. Definition 4.1. Prime numbers are natural numbers greater than 1 that can only be divided by 1 and themselves. By definition, 1 is not a prime number. If we write down the prime numbers in ascending order (prime number sequence), then we get: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, · · · . The first 100 numbers include precisely 25 prime numbers. After this, the percentage of primes decreases, but never reaches zero. We come across whole numbers that are prime fairly often. In the last decade only, three years were prime: 1993, 1997 and 1999. If they were rare, cryptography would not be able to work with them to the extent it does. Prime numbers can be factorised in a unique (“trivial”) way: 5 = 1∗5 17 = 1 ∗ 17 1, 013 = 1 ∗ 1, 013 1, 296, 409 = 1 ∗ 1, 296, 409. Definition 4.2. Natural numbers greater than 1 that are not prime are called composite numbers. These have at least two factors other than 1.

Joanne K. Rowling, “Harry Potter and the Philosopher’s Stone”, Bloomsbury, (c) 1997, chapter “Through the trapdoor”, p. 307, by Hermine. 81 Due to the fact that 12 has so many factors, this number – and multiples of this number – is often found in everyday life: the 12-hour scale on clocks, the 60 minutes in an hour, the 360-degree scale for measuring angles, etc. If we divide these scales into segments, the segments often turn out to be whole numbers. These are easier to use in mental arithmetic than fractions.

Examples of the decomposition of such numbers into prime factors: 4 = 2∗2 6 = 2∗3 91 = 7 ∗ 13 161 = 7 ∗ 23 767 = 13 ∗ 59 1, 029 = 3 ∗ 73 5, 324 = 22 ∗ 113 . Theorem 4.1. Each composite number a has a lowest factor greater than 1. This factor is a prime number p and is less than or equal to the square root of a. All whole numbers greater than 1 can be expressed as a product of prime numbers – in a unique way. This is the claim of the 1st fundamental theorem of number theory (= fundamental theorem of arithmetic = fundamental building block of all positive integers). This was formulated precisely for the first time by Carl Friedrich Gauss in his Disquisitiones Arithmeticae (1801). Theorem 4.2. Gauss 1801 Every even natural number greater than 1 can be written as the product of prime numbers. Given two such decompositions a = p1 ∗ p2 ∗ · · · ∗ pn = q1 ∗ q2 ∗ · · · ∗ qm , these can be resorted such that n = m and, for all i, pi = qi . In other words: Each natural number other than 1 can be written as a product of prime numbers in precisely one way, if we ignore the order of the factors. The factors are therefore unique (the "expression as a product of factors" is unique)! For example, 60 = 2 ∗ 2 ∗ 3 ∗ 5 = 22 ∗ 3 ∗ 5. And this – other than changing the order of the factors – is the only way in which the number 60 can be factorised. If you allow numbers other than primes as factors, there are several ways of factorising integers and the uniqueness is lost: 60 = 1 ∗ 60 = 2 ∗ 30 = 4 ∗ 15 = 5 ∗ 12 = 6 ∗ 10 = 2 ∗ 3 ∗ 10 = 2 ∗ 5 ∗ 6 = 3 ∗ 4 ∗ 5 = · · · The 1st fundamental theorem only appears to be obvious. We can construct numerous other sets of numbers[82] for which numbers in the set cannot be expressed uniquely as a product of the prime numbers of the set. In order to make a mathematical statement, therefore, it is important to state not only the operation for which it is defined but also the basic set on which the operation is defined. For more details on prime numbers (e.g. how "Fermat's Little Theorem" can be used to test extremely large numbers to determine whether they are prime), please refer to the article on prime numbers, chapter 3 in this script.

---

[82] These sets are formed especially from the set of natural numbers. An example of this can be found in this script on page 44 at the end of chapter 3.2.

4.4

Divisibility, modulus and remainder classes

If whole numbers are added, subtracted or multiplied, the result is always another whole number. The division of two whole numbers does not always result in a whole number. For example, if we divide 158 by 10 the result is the decimal number 15.8, which is not a whole number! If, however, we divide 158 by 2 the result 79 is a whole number. In number theory we express this by saying that 158 is divisible by 2 but not by 10. In general, we say: Definition 4.3. A whole number n is divisible by a whole number d if the quotient n/d is a whole number c such that n = c ∗ d. n is called a multiple of d, whereas d is called a divisor or factor of n. The mathematical notation for this is d|n (read “d divides n”). The notation d |n means that d does not divide the number n. In our example therefore: 10 |158 but 2|158. 4.4.1 The modulo operation – working with congruence

When we investigate divisibility, it is only the remainder of the division that is important. When dividing a number n by m, we often use the following notation: r n =c+ , m m where c is a whole number and r is a number with the values 0, 1, · · · , m − 1. This notation is called division with remainder, whereby c is called the whole-number “quotient” and r is the “remainder” of the division. Example: 19 5 =2+ 7 7 (m = 7, c = 2, r = 5)

What do the numbers 5, 12, 19, 26, · · · have in common for division by 7? The remainder is always r = 5. For division by 7, only the following remainders are possible: r = 0, 1, 2, · · · , 6. The numbers that result in the same remainder r when divided by 7 are combined to form the “remainder class r modulo 7”. Two numbers a and b belonging to the same remainder class modulo 7 are said to be “congruent modulo 7”. Or in general: Definition 4.4. The remainder class r modulo m is the set of all whole numbers a that have the same remainder r when divided by m.

Examples: Remainder class 0 modulo 4 = $\{x|x = 4 * n; n \in N\} = \{. . . , -16, -12, -8, -4, 0, 4, 8, 12, 16, . . . \}$ Remainder class 3 modulo 4 = $\{x|x = 4 * n + 3; n \in N\} = \{. . . , -13, -9, -5, -1, 3, 7, 11, 15, . . . \}$ As only the remainders $0, 1, 2, \cdots, m - 1$ are possible for division modulo m, modular arithmetic works with finite sets. For each modulo m there are precisely m remainder classes. Definition 4.5. Two numbers $a, b \in N$ are said to be congruent modulo $m \in N$ if and only if they have the same remainder when divided by m. We write: $a \equiv b \pmod{m}$ (read a is congruent b modulo m), which means that a and b belong to the same remainder class. The modulo is therefore the divisor. This notation was introduced by Gauss. Although the divisor is usually positive, a and b can also be any whole numbers. Examples: $19 \equiv 12 \pmod{7}$, because the remainders are equal: 19/7 = 2 remainder 5 and 12/7 = 1 remainder 5. $23103 \equiv 0 \pmod{453}$, because 23103/453 = 51 remainder 0 and 0/453 = 0 remainder 0. Theorem 4.3. $a \equiv b \pmod{m}$ if and only if, the difference $(a - b)$ is divisible by m, i.e. if $q \in Z$ exists with $(a - b) = q \star m$. These two statements are therefore equivalent. Therefore: If m divides the difference, there exists a whole number q such that: $a = b + q \star m$. As an alternative to the congruence notation, we can also use the divisibility notation: $m|(a - b)$. Example of equivalent statements: $35 \equiv 11 \pmod{3} \Longleftrightarrow 35 - 11 \equiv 0 \pmod{3}$, where $35 - 11 = 24$ is divisible by 3 without remainder while 35 : 3 and 11 : 3 leave the remainder 2. Comment: The above equivalence does not apply to the sum $(a + b)$! Example: $11 \equiv 2 \pmod{3}$, therefore $11 - 2 \equiv 9 \equiv 0 \pmod{3}$; but $11 + 2 = 13$ is not divisible by 3. The statement in theorem 4.3 does not even apply to sums in one direction. It is correct for sums only if the remainder is 0 and only in the following direction: if a divisor divides both summands with no remainder, it also divides the sum with no remainder. We can apply the above equivalence in theorem 4.3 if we need a quick and easy method of determining whether large numbers are divisible by a certain number. Example: Is 69, 993 divisible by 7?

The number can be written in the form of a difference in which it is clear that each operand is divisible by 7: $69,993 = 70,000 - 7$. Therefore, the difference is also divisible by 7. Although these considerations and definitions may seem to be rather theoretical, we are so familiar with them in everyday life that we no longer think about the formal procedure. For example, the 24 hours on a clock are represented by the numbers $1, 2, \cdots, 12$. We obtain the hours after 12 noon as the remainder of a division by 12 and know immediately that 2 o'clock in the afternoon is the same as 14.00. This “modular” arithmetic (based on division remainders) forms the basis of asymmetric encryption procedures. Cryptographic calculations are therefore not based on real numbers, as the calculations you performed at school, but rather on character strings with a limited length, in other words on positive whole numbers that cannot exceed a certain value. This is one of the reasons why we choose a large number m and “calculate modulo m”. That is, we ignore whole-number multiples of m and, rather than working with a number, we only work with the remainder when this number is divided by m. The result is that all results are in the range 0 to $m - 1$.

4.5
4.5.1

Calculations with finite sets
Laws of modular calculations

From algebra theorems it follows that essential parts of the conventional calculation rules are kept when we proceed to modular calculations over a basic set Z. For example, addition remains commutative. The same goes for multiplication modulo m. The result of a division[83] is not a fraction but rather a whole number between 0 and $m - 1$. The known laws apply: 1. Associative law: $((a + b) + c) \pmod{m} \equiv (a + (b + c)) \pmod{m}$. $((a * b) * c) \pmod{m} \equiv (a * (b * c)) \pmod{m}$. 2. Commutative law: $(a + b) \pmod{m} \equiv (b + a) \pmod{m}$. $(a * b) \pmod{m} \equiv (b * a) \pmod{m}$. The associative law and commutative law apply to both addition and multiplication. 3. Distributive law: $(a * (b + c)) \pmod{m} \equiv (a * b + a * c) \pmod{m}$. 4. Reducibility: $(a + b) \pmod{m} \equiv (a \pmod{m} + b \pmod{m}) \pmod{m}$. $(a * b) \pmod{m} \equiv (a \pmod{m} * b \pmod{m}) \pmod{m}$.

---

[83] When dividing modulo m we cannot use every number because some numbers have the same property as zero. See footnote 87 in chapter 4.6.1.

The order in which order the modulo operation is performed is irrelevant. 5. Existence of an identity (neutral element): $(a + 0) \pmod m \equiv (0 + a) \pmod m \equiv a \pmod m$. $(a * 1) \pmod m \equiv (1 * a) \pmod m \equiv a \pmod m$. 6. Existence of an inverse element: For all whole numbers a and m there exists a whole number $-a$ such that: $(a + (-a)) \pmod m \equiv 0 \pmod m$ (additive inverse). For each a ($a \equiv 0 \pmod p$) where p is prime there exists a whole number a^{-1}, such that: $(a * a^{-1}) \pmod p \equiv 1 \pmod p$ (multiplicative inverse). 7. Closeness[84] : $a, b \in G \Longrightarrow (a + b) \in G$. $a, b \in G \Longrightarrow (a * b) \in G$. 8. Transitivity: $[a \equiv b \bmod m,\ b \equiv c \bmod m] \Longrightarrow [a \equiv c \bmod m]$.

4.5.2 Patterns and structures

In general mathematicians investigate “Structures”. They ask e.g. at $a * x \equiv b \bmod m$, which values x can take for given values of a, b, m. Especially the case is investigated, where the result b of this operation is the neutral element. Then x is the inverse of a regarding this operation.

---

[84] The property of closeness is always defined in relation to an operation in a set. See Appendix B of this chapter.

Seneca85 : The way of theory is long, it is short and effective by examples.

4.6

Examples of modular calculations

As we have already seen: For two natural numbers a and m, a mod m denotes the remainder obtained when we divide a by m. This means that a (mod m) is always a number between 0 and m − 1. For example, 1 ≡ 6 ≡ 41 ≡ 1 (mod 5) because the remainder is always 1. Another example is: 2000 ≡ 0 (mod 4) because 4 divides 2000 with no remainder. Modular arithmetic only contains a limited quantity of non-negative numbers. The number of these is specified by a modulus m. If the modulo is m = 5, then only the 5 numbers in the set {0, 1, 2, 3, 4} are used. A calculation result larger than 4 is then reduced “modulo 5”. In other words, it is the remainder when the result is divided by 5. For example, 2 ∗ 4 ≡ 8 ≡ 3 (mod 5) because 3 is the remainder when we divide 8 by 5. 4.6.1 Addition and multiplication

The following shows • the addition table86 (mod 5) and • the multiplication tables87 for mod 5 and mod 6. Example of an addition table The result when we add 3 and 4 (mod 5) is calculated as follows: Calculate 3 + 4 = 7 and keep subtracting 5 from the result until the result is less than the modulo: 7 − 5 = 2. Therefore: 3 + 4 ≡ 2 (mod 5).

85

Lucius Annaeus Seneca, philosophical writer and poet, 4 B. C. − 65 A. D. Comment on subtraction modulo 5: 2 − 4 ≡ −2 ≡ 3 mod 5. It is therefore not true modulo 5 that −2 = 2 (see also Appendix C of this chapter). 87 Comment on division modulo 6: Due to the special role of zero as the identity for addition, division by zero is not permitted: for all a it is a ∗ 0 = 0, because a ∗ 0 = a ∗ (0 + 0) = a ∗ 0 + a ∗ 0. Obviously 0 has no inverse regarding the multiplication, because if there would be one, it must be 0 = 0 ∗ 0−1 = 1. Also see footnote 83.
86

Addition table modulo 5:

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Example of an multiplication table: The result of the multiplication $4 * 4 \pmod 5$ is calculated as follows: $4 * 4 = 16$ and subtract 5 until the result is less than the modulus. $16 - 5 = 11$; $11 - 5 = 6$; $6 - 5 = 1$. The table directly shows that $4 * 4 \equiv 1 \pmod 5$ because $16 : 5 = 3$ remainder 1. Multiplication is defined on the set Z excluding 0. Multiplication table modulo 5: * 1 2 3 4 1 1 2 3 4 2 2 4 1 3 3 3 1 4 2 4 4 3 2 1

4.6.2

Additive and multiplicative inverses

You can use the tables to read the inverses for each number in relation to addition and multiplication. The inverse of a number is the number that gives the result 0 when the two numbers are added and 1 when they are multiplied. Thus, the inverse of 4 for addition mod 5 is 1 and the inverse of 4 for multiplication mod 5 is 4 itself, because $4 + 1 = 5 \equiv 0 \pmod 5$; $4 * 4 = 16 \equiv 1 \pmod 5$. The inverse of 1 for multiplication mod 5 is 1, while the inverse modulo 5 of 2 is 3 and, since multiplication is commutative, the inverse of 3 is again 2. If we take a random number and add or multiply another number (here 4) and then add[88] or multiply the corresponding inverse (1 or 4) to the interim result (1 or 3), then the end result is the same as the initial value. Example: $2 + 4 \equiv 6 \equiv 1 \pmod 5$; $2 * 4 \equiv 8 \equiv 3 \pmod 5$; $1 + 1 \equiv 2 \equiv 2 \pmod 5$, $3 * 4 \equiv 12 \equiv 2 \pmod 5$.

[88] In general $x + y + (-y) \equiv x \pmod m$ [$(-y)$ = additive inverse of $y \pmod m$].

In the set Z5 = {0, 1, 2, 3, 4} for the addition and in the set Z∗ for the multiplication, all numbers 5 have a unique inverse modulo 5. In the case of modular addition, this is true for every modulo (not just for 5). This is not the case, however, for modular multiplication. Theorem 4.4. A natural number a from the set {1, · · · , m − 1} has one inverse if and only if it and the modulo m are co-prime[89] , in other words if a and m have no common prime factors. Since m = 5 is prime, the numbers 1 to 4 are relatively prime to 5 and each of these numbers has a multiplicative inverse in mod 5. A counterexample shows the multiplication table for mod 6 (since the modulus m = 6 is not prime, not all elements from Z6 \ {0} are relatively prime to 6): Multiplication table modulo 6: * 1 2 3 4 5 1 1 2 3 4 5 2 2 4 0 2 4 3 3 0 3 0 3 4 4 2 0 4 2 5 5 4 3 2 1

In addition to 0, the numbers 2, 3 and 4 also have no unique inverse (we can also say they have no inverse, because the elementary property of an inverse is uniqueness). The numbers 2, 3 and 4 have the factor 2 or 3 in common with the modulus 6. Only the numbers 1 and 5, which are relatively prime to 6, have multiplicative inverses, namely themselves. The number of numbers that are relatively prime to the modulus m is the same as the number of numbers that have a multiplicative inverse (see the Euler function J(m) below). For the two moduli 5 and 6 used in the multiplication tables, this means: the modulus 5 is a prime number itself. In mod 5, therefore, there are exactly J(5) = 5 − 1 = 4 numbers that are relatively prime to the modulus, that is all numbers from 1 to 4. Since 6 is not a prime number, we write it as a product of its factors: 6 = 2 ∗ 3. In mod 6, therefore, there are exactly J(6) = (2 − 1) ∗ (3 − 1) = 1 ∗ 2 = 2 numbers that have a multiplicative inverse, that is 1 and 5. Although it may seem difficult to calculate the table of multiplicative inverses for large moduli (this only applies to the areas of the table shaded dark grey), we can use Fermat’s Little Theorem

---

[89] Two whole numbers a and b are co-prime if and only if gcd(a, b) = 1. If p is prime and a is a random whole number that is not a multiple of p, then p and a are co-prime. Further name to the topic co-prime (with ai ∈ Z, i = 1, · · · , n): 1. a1 , a2 , · · · , an are relatively prime , if gcd(a1 , · · · , an) = 1. 2. An even stronger request for more than two numbers is : a1 , · · · , an are in pairs relatively prime, if for all i = 1, · · · , n and j = 1, · · · , n with i = j: gcd(ai , aj) = 1. Example: 2, 3, 6 are relatively prime, because gcd(2, 3, 6) = 1. They are not in pairs relatively prime, because gcd(2, 6) = 2 > 1.

to create a simple algorithm for this [Pfleeger1997, p. 80]. Quicker algorithms are described, for instance, in [Knuth1998]90 . Cryptographically not only the unique nature of the inverse is important, but also that the set of possible values has been exhausted. Theorem 4.5. For a, i ∈ {1, · · · , m − 1} with gcd(a, m) = 1), then the product a ∗ i mod m takes for a certain number a all values from {1, · · · , m − 1} (exhaustive permutation of the length m − 1)91 . The following three examples92 illustrate the properties of multiplicative inverses. In the multiplication table mod 17, the following was calculated for i = 1, 2, · · · , 18: (5 ∗ i)/17 = a remainder r and high-lighted 5 ∗ i ≡ 1 (mod 17), (6 ∗ i)/17 = a remainder r and high-lighted 6 ∗ i ≡ 1 (mod 17). We need to find the i for which the product remainder a ∗ i modulo 17 with a = 5 or a = 6 has the value 1. Table 1: Multiplication table modulo 17 (for a = 5 and a = 6) i 5∗i remainder 6∗i remainder 1 5 5 6 6 2 10 10 12 12 3 15 15 18 1 4 20 3 24 7 5 25 8 30 13 6 30 13 36 2 7 35 1 42 8 8 40 6 48 14 9 45 11 54 3 10 50 16 60 9 11 55 4 66 15 12 60 9 72 4 13 65 14 78 10 14 70 2 84 16 15 75 7 90 5 16 17 18 80 85 90 12 0 5 96 102 108 11 0 6

Between i = 1, · · · , m, all values between 0, · · · , m − 1 occur for the remainders, because both 5 and 6 are also relatively prime to the modulus m = 17. The multiplicative inverse of 5 (mod 17) is 7, while the inverse of 6 (mod 17) is 3.

Table 2: Multiplication table modulo 13 (for a = 5 and a = 6) i 5∗i remainder 6∗i remainder 1 5 5 6 6 2 10 10 12 12 3 15 2 18 5 4 20 7 24 11 5 25 12 30 4 6 30 4 36 10 7 35 9 42 3 8 40 1 48 9 9 45 6 54 2 10 50 11 60 8 11 55 3 66 1 12 60 8 72 7 13 65 0 78 0 14 70 5 84 6 15 75 10 90 12 16 80 2 96 5 17 85 7 102 11 18 90 12 108 4

Using Euclid’s extended theorem (extended gcd), we can calculate the multiplicative inverse and determine whether numbers have an inverse (see appendix A of this chapter). Alternatively, we can also use the primitive roots. 91 See also theorem 4.14 in chapter 4.9, Multiplicative order and primitive roots. 92 See Appendix E of this chapter for the source code to compute the tables using Mathematica and Pari-GP.

90

Between i = 1, · · · , m, all values between 0, · · · , m − 1 occur for the remainders, because both 5 and 6 are relatively prime to the modulus m = 13. The multiplicative inverse of 5 (mod 13) is 8, while the inverse of 6 (mod 13) is 11.

The following table contains an example, where the modulus m and the number a = 6 are not relatively prime. Table 3: Multiplication table modulo 12 (for a = 5 and a = 6) i 5∗i remainder 6∗i remainder 1 5 5 6 6 2 10 10 12 0 3 15 3 18 6 4 20 8 24 0 5 25 1 30 6 6 30 6 36 0 7 35 11 42 6 8 40 4 48 0 9 45 9 54 6 10 50 2 60 0 11 55 7 66 6 12 60 0 72 0 13 65 5 78 6 14 70 10 84 0 15 75 3 90 6 16 80 8 96 0 17 85 1 102 6 18 90 6 108 0

We have calculated (5 ∗ i) (mod 12) and (6 ∗ i) (mod 12). Between i = 1, · · · , m, not all values between 0, · · · , m − 1 occur and 6 does not have an inverse mod 12, because 6 and the modulus m = 12 are not co-prime. The multiplicative inverse of 5 (mod 12) is 5. The number 6 has no inverse (mod 12). 4.6.3 Raising to the power

In modular arithmetic, raising to the power is defined as repeated multiplication – as usual except that multiplication is now slightly different. We can even apply the usual rules, such as: ab+c = ab ∗ ac , (ab)c = ab∗c = ac∗b = (ac)b . Modular powers work in the same way as modular addition and modular multiplication: 32 ≡ 9 ≡ 4 (mod 5). Even consecutive powers work in the same way: Example 1: (43)2 ≡ 642 ≡ 4096 ≡ 1 (mod 5). (1) We can speed up93 the calculation by reducing the interim results modulo 5 but we need to take care because not everything will then work in the same way as

93

The time required to calculate the multiplication of two numbers normally depends on the length of the numbers. We can observe this if we use the school method to calculate, for instance, 474 ∗ 228. The time required increases in a quadratic square manner , because we need to multiply 3 ∗ 3 numbers. The numbers become considerably smaller if we reduce the interim result.

in standard arithmetic. (43)2 ≡ (43 (mod 5))2 (mod 5) ≡ (64 (mod 5))2 (mod 5) ≡ 42 (mod 5) ≡ 16 ≡ 1 (mod 5). (2) In standard arithmetic, consecutive powers can be reduced to a single power by multiplying the exponents: (43)2 = 43∗2 = 46 = 4096. This is not quite as simple in modular arithmetic because this would give: (43)2 ≡ 43∗2
(mod 5)

≡ 46

(mod 5)

≡ 41 ≡ 4 (mod 5).

But as we saw above, the correct result is 1!! (3) Therefore, the rule is slightly different for consecutive powers in modular arithmetic: we do not multiply the exponents in (mod m) but rather in (mod J(m)). Using J(5) = 4 gives: (43)2 ≡ 43 ∗ 2
(mod J(5))

≡ 46

mod 4

≡ 42 ≡ 16 ≡ 1 (mod 5).

This delivers the correct result. Theorem 4.6. (ab)c ≡ ab∗c Example 2: 328 ≡ 34 ∗ 7 ≡ 34 ∗ 7
(mod 10) (mod J(m))

(mod m). ≡ 38 ≡ 6561 ≡ 5 (mod 11).

4.6.4

Fast calculation of high powers

RSA encryption and decryption[94] entails calculating high powers modulo m. For example, the calculation (1005) (mod 3) exceeds the 32-bit long integer number range provided we calculate an by actually multiplying a with itself n times in line with the definition. In the case of extremely large numbers, even a fast computer chip would take longer than the age of the universe to calculate a single exponential. Luckily, there is an extremely effective shortcut for calculating exponentials (but not for calculating logarithms). If the expression is divided differently using the rules of modular arithmetic, then the calculation does not even exceed the 16-bit short integer number range: (a5) ≡ (((a2 (mod m))2 (mod m)) ∗ a) (mod m).

---

[94] See chapter 4.10 (Proof of the RSA procedure with Euler-Fermat) and chapter 4.13 (The RSA procedure with actual numbers).

We can generalise this by representing the exponent as a binary number. For example, the naive method would require 36 multiplications in order to calculate an for n = 37. However, if we write n in the binary representation as 100101 = 1 ∗ 25 + 1 ∗ 22 + 1 ∗ 20 , then we can rewrite the 5 2 0 5 2 expression as: a37 = a2 +2 +2 = a2 ∗ a2 ∗ a1 Example 3: 8743 (mod 103). Since 43 = 32 + 8 + 2 + 1 , 103 is prime, 43 < J(103) and the squares (mod 103) can be calculated beforehand 872 ≡ 50 (mod 103), 874 ≡ 502 ≡ 28 (mod 103), 878 ≡ 282 ≡ 63 (mod 103), 8716 ≡ 632 ≡ 55 (mod 103), 8732 ≡ 552 ≡ 38 (mod 103). we have95 : 8743 ≡ 8732+8+2+1 (mod 103) ≡ 8732 ∗ 878 ∗ 872 ∗ 87 (mod 103) ≡ 38 ∗ 63 ∗ 50 ∗ 87 ≡ 85 (mod 103). The powers (a2)k can be determined easily by means of repeated squaring. As long as a does not change, a computer can calculate them beforehand and – if enough memory is available – save them. In order to then find an in each individual case, it now only needs to multiply those (a2)k for which there is a one in the k-th position of the binary representation of n. The typical effort is then reduced from 2600 to 2 ∗ 600 multiplications! This frequently used algorithm is called “Square and Multiply”. 4.6.5 Roots and logarithms

The inverses of the powers are also defined. The roots and logarithms are again whole numbers. Yet in contrast to the usual situation, they are not only difficult to calculate but, in the case of large numbers, cannot be calculated at all within a reasonable amount of time. Let us take the equation a ≡ bc (mod m). a) Taking the logarithm (determining c) – Discrete logarithm problem : If we know a and b of the three numbers a, b and c that meet this equation, then every known method of finding c is approximately just as time-consuming as trying out all m possible values for c one after the other. For a typical m of the order of magnitude of 10180 for 600-digit binary numbers, this is a hopeless task. More precisely, for suitably

95

See Appendix E of this chapter for source code implementing the square and multiply method in Mathematica and Pari-GP, which can be used to reproduce the calculations above.

large numbers m, the time required according to current knowledge is proportional to exp C ∗ (log m[log log m]2)1/3 with a constant C > 1. b) Calculating the root (determining b): The situation is similar if b is the unknown variable and we know the values of a and c: If we know the Euler function of m, J(m), then we can easily96 calculate d with c ∗ d ≡ 1 (mod J(m)) and use theorem 4.6 to obtain: ad ≡ (bc)d ≡ bc∗d ≡ bc∗d the c-th root b of a. If J(m) cannot be determined97 , it is difficult to calculate the c-th root. This forms the basis for the security assumption used by the RSA encryption system (see chapter 4.10 or chapter 5.3.1). The time required for inverting addition and multiplication, on the other hand, is simply proportional to log m or (log m)2 . Powers (for a number x calculate xa with a fixed) and exponents (for a number x calculate ax with a fixed) are therefore typical one way functions (See Overview of the one way functions in this Script and article).
(mod J(m))

≡ b1 ≡ b (mod m)

4.7

Groups and modular arithmetic in Zn and Z∗ n

Mathematical “groups” play a decisive role in number theory and cryptography. We only talk of groups if, for a defined set and a defined relation (an operation such as addition or multiplication), the following properties are fulfilled: • The set is closed • A neutral element exists • An inverse element exists for each element • The associative law applies. The abbreviated mathematical notation is (G, +) or (G, ∗). Definition 4.6. Zn : Zn comprises all numbers from 0 to n − 1 : Zn = {0, 1, 2, · · · , n − 2, n − 1}. Zn is an often used finite group of the natural numbers. It is sometimes also called the remainder set R modulo n. For example, 32-bit computers (standard PCs) only directly work with whole numbers in a finite set, that is the value range 0, 1, 2, · · · , 232 − 1. This value range is equivalent to the set Z232 .

96 97

See Appendix A of this chapter: the greatest common divisor (gcd) of whole numbers. According to the first fundamental theorem of number theory and theorem 4.11, we can determine J(m) by reducing m to prime factors.

4.7.1

Addition in a group

If we define the operation mod+ on such a set where a mod + b := (a + b) (mod n), then the set Zn together with the relation mod+ is a group because the following properties of a group are valid for all elements in Zn : • a mod + b is an element of Zn (the set is closed), • (a mod + b) mod + c ≡ a mod + (b mod + c) • the neutral element is 0. • each element a ∈ Zn has an inverse for this operation, namely n − a (because a mod + (n − a) ≡ a + (n − a) (mod n) ≡ n ≡ 0 (mod n)). Since the operation is commutative, i.e. (a mod + b) = (b mod + a), this structure is actually a “commutative group”. 4.7.2 Multiplication in a group (mod+ is associative),

If we define the operation mod* on the set Zn where a mod ∗ b := (a ∗ b) (mod n), then Zn together with this operation is usually not a group because not all properties are fulfilled for each n. Examples: a) In Z15 , for example, the element 5 does not have an inverse. That is to say, there is no a with 5 ∗ a ≡ 1 (mod 15). Each modulo product with 5 on this set gives 5, 10 or 0. b) In Z55 \ {0}, for example, the elements 5 and 11 do not have multiplicative inverses. That is to say, there is no a ∈ Z55 such that 5 ∗ a ≡ 1 (mod 55) and no a such that 11 ∗ a ≡ 1 (mod 55). This is because 5 and 11 are not relatively prime to 55. Each modulo product with 5 on this set gives 5, 10, 15, . . . , 50 or 0. Each modulo product with 11 on this set gives 11, 22, 33, 44 or 0. On the other hand, there are subsets of Zn that form a group with the operation mod*. If we choose all elements in Zn that are relatively prime to n, then this set forms a group with the operation mod*. We call this set Z∗ . n Definition 4.7. Z∗ : n Z∗ = {a ∈ Zn | gcd(a, n) = 1}. n

Z∗ is sometimes also called the reduced remainder set R modulo n. n Example: For n = 10 = 2 ∗ 5 the following applies: full remainder set R = Zn = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} reduced remainder set R = Z∗ = {1, 3, 7, 9} −→ J(n) = 4. n Comment: R or Z∗ is always a genuine subset of R or Zn because 0 is always an element of R n but never an element of R . Since 1 and n − 1 are always relatively prime to n, they are always elements of both sets. If we select a random element in Z∗ and multiply it by every other element in Z∗ , then the n n products98 are all in Z∗ , and the results are also a unique permutation of the elements in Z∗ . n n Since 1 is always an element of Z∗ , there is a unique “partner” in this set such that the product n is 1. In other words: Theorem 4.7. Each element in Z∗ has a multiplicative inverse. n Example for a = 3 modulo 10 with Z∗ = {1, 3, 7, 9} : n 3 ≡ 3 ∗ 1 (mod 10), 9 ≡ 3 ∗ 3 (mod 10), 1 ≡ 3 ∗ 7 (mod 10), 7 ≡ 3 ∗ 9 (mod 10). The unique invertibility is an essential condition for cryptography (see section 4.10).

98

This is due to the fact that Z∗ is closed with respect to the multiplication and due to the gcd property: n [a, b ∈ Z∗] ⇒ [((a ∗ b) (mod n)) ∈ Z∗], exactly: n n [a, b ∈ Z∗] ⇒ [gcd(a, n) = 1, gcd(b, n) = 1] ⇒ [gcd(a ∗ b, n) = 1] ⇒ [((a ∗ b) (mod n)) ∈ Z∗]. n n

Eric Berne[99] : Mathematical game theory postulates players who respond rationally. Transactional game theory, on the other hand, deals with games that are not rational, perhaps even irrational and thereby closer to reality.

4.8
4.8.1

Euler function, Fermat's little theorem and Euler-Fermat
Patterns and structures

As mathematicians investigate the structure $a * x \equiv b \mod m$ (see chapter 4.5.2), so they are interested in the structure $xa \equiv b \mod m$. Again here they are interested in the case, if $b = 1$ (value of the multiplicative inverse) and if $b = x$ (the function has a fixpoint). 4.8.2 The Euler function

Given n, the number of numbers from the set $\{1, \cdot \cdot \cdot , n - 1\}$ that are relatively prime to n is equal to the value of the Euler[100] function J(n). Definition 4.8. The Euler function[101] J(n) specifies the number of elements in Z∗ . n J(n) also specifies how many whole numbers have multiplicative inverses in mod n. J(n) can be calculated if we know the prime factors of n. Theorem 4.8. For a prime number, the following is true: $J(p) = p - 1$. Theorem 4.9. If m is the product of two distinct primes, then: $J(p * q) = (p - 1) * (q - 1)$ This case is important for the RSA procedure. Theorem 4.10. If $n = p1 * p2 * \cdot \cdot \cdot * pk$ where p1 to pk are distinct prime numbers (i.e. no factor occurs more than once), then the following is true (as a generalisation of theorem 4.9): $J(n) = (p1 - 1) * (p2 - 1) * \cdot \cdot \cdot * (pk - 1)$. Theorem 4.11. In general, the following is true for every prime number p and every n in N: 1. $J(pn) = pn-1 * (p - 1)$. or $J(p * q) = J(p) * J(q)$.

Eric Berne, "Games People Play", rororo, (c) 1964, page 235. Leonhard Euler, Swiss mathematician, Apr 15, 1707 – Sep 18, 1783 101 Often written as the Euler phi function Φ(n).
100

99

2. If n = pe1 ∗ pe2 ∗ · · · ∗ pek , where p1 to pk are distinct prime numbers, then: 1 2 k J(n) = [(pe1 −1) ∗ (p1 − 1)] ∗ · · · ∗ [(pek −1) ∗ (pk − 1)] = n ∗ ([(p1 − 1)/p1] ∗ · · · ∗ [(pk − 1)/pk]). 1 k Examples: • n = 70 = 2 ∗ 5 ∗ 7 =⇒ using theorem 4.10: J(n) = 1 · 4 · 6 = 24. • n = 9 = 32 =⇒ using theorem 4.11: J(n) = 31 · 2 = 6, because Z∗ = {1, 2, 4, 5, 7, 8}. 9 • n = 2, 701, 125 = 32 ∗ 53 ∗ 74 =⇒ using theorem 4.11: J(n) = [31 ∗ 2] ∗ [52 ∗ 4] ∗ [73 ∗ 6] = 1, 234, 800.

4.8.3 The theorem of Euler-Fermat

In order to prove the RSA procedure, we need Fermat’s theorem and its generalisation (EulerFermat theorem) – please see chapter 3.5. Theorem 4.12. Fermat’s Little Theorem102 Let p be a prime number and a be a random whole number, then: ap ≡ a (mod p). An alternative formulation of Fermat’s Little Theorem is as follows: Let p be a prime number and a be a random whole number that is relatively prime to p, then: ap−1 ≡ 1 (mod p). Theorem 4.13. Euler-Fermat theorem (generalisation of Fermat’s Little Theorem) For all elements a in the group Z∗ (i.e. a and n are natural numbers that are co-prime): n aJ(n) ≡ 1 (mod n). This theorem states that if we raise a group element (here a) to the power of the order of the group (here J(n)), we always obtain the neutral element for multiplication (the number 1). The 2nd formulation of Fermat’s Little Theorem is derived directly from Euler’s theorem if n is a prime number. If n is the product of two prime numbers, we can - in certain cases - use Euler’s theorem to calculate the result of a modular power very quickly. We have: a(p−1)∗(q−1) ≡ 1 (mod pq). Examples for calculating a modular power: • With 2 = 1 ∗ 2 and 6 = 2 ∗ 3 where 2 and 3 are both prime; J(6) = 2 because only 1 and 5 are relatively prime to 6, we obtain the equation 52 ≡ 5J(6) ≡ 1 (mod 6), without having to calculate the power. • With 792 = 22 ∗ 36 and 23 ∗ 37 = 851 where 23 and 37 are both prime, it follows that 31792 ≡ 31J(23∗37) ≡ 31J(851) ≡ 1 (mod 851).

102

Pierre de Fermat, French mathematician, Aug 17, 1601 – Jan 12, 1665.

4.8.4

Calculation of the multiplicative inverse

Another interesting application is a special case of determining the multiplicative inverses using the Euler-Fermat theorem (multiplicative inverses are otherwise determined using the extended Euclidean algorithm). Example: Find the multiplicative inverse of 1579 modulo 7351. According to Euler-Fermat: aJ(n) = 1 (mod n) for all a in Z∗ . If we divide both sides by a, we n get: aJ(n)−1 ≡ a−1 (mod n). For the special case that the modulo is prime, we have J(n) = p − 1. Therefore, the modular inverse is a−1 = aJ(n)−1 ≡ a(p−1)−1 ≡ ap−2 (mod p). For our example, this means: Since the modulus 7351 is prime, p − 2 = 7349. 1579−1 ≡ 15797349 (mod p). By cleverly breaking down the exponent, we can calculate this power relatively easily (see Section 4.6.4 Fast calculation of high powers): 7349 = 4096 + 2048 + 1024 + 128 + 32 + 16 + 4 + 1 1579−1 ≡ 4716 (mod 7351). 4.8.5 Fixpoints modulo 26

According to theorem 4.6, the arithmetic operations of modular expressions are performed in the exponents modulo J(n) rather than modulo n103 . In ae∗d ≡ a1 (mod n), if we wish to determine the inverses for the factor e in the exponent, we need to calculate modulo J(n). Example (with reference to the RSA algorithm): If we calculate modulo 26, which set can e and d come from? Solution: we have e ∗ d ≡ 1 (mod J(26)). The reduced remainder set R = Z∗ = {1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25} are the elements 26 in Z26 , which have a multiplicative inverse, that is which are relatively prime to 26. The reduced remainder set R contains only the elements of R that are relatively prime to J(n) = 12 : R = {1, 5, 7, 11}. For every e in R there exists a d in R such that a ≡ (ae)d (mod n).

[103] For the following example, we will adopt the usual practice for the RSA procedure of using “n” rather than “m” to denote the modulus.

For every e in R , there exists therefore precisely one element (not necessarily different from e) such that e ⋆ d ≡ 1 (mod J(26)). For all e that are relatively prime to J(n) we could calculate d as follows using the Euler-Fermat theorem: For aJ(n) ≡ 1 (mod n) is the same as saying aJ(n)−1 ≡ a−1 (mod n). Therefore d ≡ e−1 (mod J(n)) ≡ eJ(J(n))−1 (mod J(n)). The problems of factorising n = pq with q = p and finding J(n) have a similar degree of difficulty and if we find a solution for one of the two problems, we also have a solution for the other[104] (please compare requisition 3 in section 4.10.1).

---

[104] If we know the factors of n = p ∗ q with p = q, then J(n) = (p − 1) ∗ (q − 1) = n − (p + q) + 1. Additionally the factors p and q are solutions of the quadratic equation x2 − (p + q)x + pq = 0. If only n and J(n) are known, then it is: pq = n and p + q = n − J(n) + 1. So you get p and q by solving the equation x2 + (J(n) − n − 1)x + n = 0.

4.9

Multiplicative order and primitive roots

Mathematicians often ask, in which conditions the repeated application of an operation results in the neutral element (compare patterns and structures before). For the i-times successive modular multiplication of a number a with i = 1, · · · , m−1 the product is the neutral element of the multiplication (1) if and only if a and m are relatively prime. The value of i, for which the product ai = 1, is called the multiplicative order of a. The multiplicative order and the primitive root are two useful constructs (concepts) in elementary number theory. Definition 4.9. The multiplicative order ordm (a) of a whole number a (mod m) (where a and m are co-prime) is the smallest whole number e for which ae ≡ 1 (mod m). The following table shows that in a multiplicative group (here Z∗) not all numbers necessarily 11 have the same order. The orders in this case are 1, 2, 5 and 10 and we notice that: 1. The orders are all factors of 10. 2. The numbers a = 2, 6, 7 and 8 have the order 10 - we say that these numbers have the maximum order in Z∗ . 11 Example 1: The following table105 shows the values ai mod 11 for the exponents i = 1, 2, · · · , 10 and for the bases a = 1, 2, · · · , 10 as well as the resulting value ord11 (a) for each a: Table 4: Values of ai mod 11, 1 ≤ a, i < 11 and according order of a mod m: i=1 1 2 3 4 5 6 7 8 9 10 i=2 1 4 9 5 3 3 5 9 4 1 i=3 1 8 5 9 4 7 2 6 3 10 i=4 1 5 4 3 9 9 3 4 5 1 i=5 1 10 1 1 1 10 10 10 1 10 i=6 1 9 3 4 5 5 4 3 9 1 i=7 1 7 9 5 3 8 6 2 4 10 i=8 1 3 5 9 4 4 9 5 3 1 i=9 1 6 4 3 9 2 8 7 5 10 i=10 1 1 1 1 1 1 1 1 1 1 ord11 (a) 1 10 5 5 5 10 10 10 5 2

a=1 a=2 a=3 a=4 a=5 a=6 a=7 a=8 a=9 a = 10

The table shows, for example, that the order of 3 modulo 11 has the value 5. Definition 4.10. If a and m are co-prime and if ordm (a) = J(m) (i.e. a has maximum order), then we say that a is a primitive root of m.

105

See Appendix E of this chapter for the source code to generate the table using Mathematica and Pari-GP.

A number a is not a primitive root for every modulo m. In the above table, only a = 2, 6, 7 and 8 is a primitive root with respect to mod 11 (J(11) = 10). Using the primitive roots, we can clearly establish the conditions for which powers modulo m have a unique inverse and the calculation in the exponents is manageable. The following two tables show the multiplicative orders and primitive roots modulo 45 and modulo 46. Example 2: The following table106 shows the values ai mod 45 for the exponents i = 1, 2, · · · , 12 and for the bases a = 1, 2, · · · , 12 as well as the resulting value ord45 (a) for each a: Table 5: Values of ai mod 45, 1 ≤ a, i < 13: a\i 1 2 3 4 5 6 7 8 9 10 11 12 1 1 2 3 4 5 6 7 8 9 10 11 12 2 1 4 9 16 25 36 4 19 36 10 31 9 3 1 8 27 19 35 36 28 17 9 10 26 18 4 1 16 36 31 40 36 16 1 36 10 16 36 5 1 32 18 34 20 36 22 8 9 10 41 27 6 1 19 9 1 10 36 19 19 36 10 1 9 7 1 38 27 4 5 36 43 17 9 10 11 18 8 1 31 36 16 25 36 31 1 36 10 31 36 9 1 17 18 19 35 36 37 8 9 10 26 27 10 1 34 9 31 40 36 34 19 36 10 16 9 11 1 23 27 34 20 36 13 17 9 10 41 18 12 1 1 36 1 10 36 1 1 36 10 1 36 ord45 (a) 1 12 – 6 – – 12 4 – – 6 – J(45) 24 24 24 24 24 24 24 24 24 24 24 24

J(45) is calculated using theorem 4.11: J(45) = J(32 ∗ 5) = 31 ∗ 2 ∗ 4 = 24. Since 45 is not a prime, there is no “multiplicative order” for all values of a (e. g. for the numbers that are not relatively prime to 45 : 3, 5, 6, 9, 10, 12, · · · , because 45 = 32 ∗ 5). Example 3: Is 7 a primitive root modulo 45? The requirement/condition gcd(7, 45) = 1 is fulfilled. The table ’values of ai mod 45’ shows that the number 7 is not a primitive root of 45, because ord45 (7) = 12 = 24 = J(45). Example 4: The following table107 answers the question as to whether the number 7 is a primitive root of 46. The requirement/condition gcd(7, 46) = 1 is fulfilled.

106 107

See Appendix E of this chapter for the source code to generate the table using Mathematica and Pari-GP. See Appendix E of this chapter for the source code to generate the table using Mathematica and Pari-GP.

Table 6: Values of ai mod 46, 1 ≤ a, i < 23: a\i 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 2 1 4 9 16 25 36 3 18 35 8 29 6 31 12 41 26 13 2 39 32 27 24 23 3 1 8 27 18 33 32 21 6 39 34 43 26 35 30 17 2 37 36 5 42 15 22 23 4 1 16 35 26 27 8 9 2 29 18 13 36 41 6 25 32 31 4 3 12 39 24 23 5 1 32 13 12 43 2 17 16 31 42 5 18 27 38 7 6 21 26 11 10 37 22 23 6 1 18 39 2 31 12 27 36 3 6 9 32 29 26 13 4 35 8 25 16 41 24 23 7 1 36 25 8 17 26 5 12 27 14 7 16 9 42 11 18 43 6 15 44 33 22 23 8 1 26 29 32 39 18 35 4 13 2 31 8 25 36 27 12 41 16 9 6 3 24 23 9 1 6 41 36 11 16 15 32 25 20 19 4 3 44 37 8 7 12 33 28 17 22 23 10 1 12 31 6 9 4 13 26 41 16 25 2 39 18 3 36 27 32 29 8 35 24 23 11 1 24 1 24 45 24 45 24 1 22 45 24 1 22 45 24 45 24 45 22 45 22 23 12 1 2 3 4 41 6 39 8 9 36 35 12 13 32 31 16 29 18 27 26 25 24 23 13 1 4 9 16 21 36 43 18 35 38 17 6 31 34 5 26 33 2 7 14 19 22 23 14 1 8 27 18 13 32 25 6 39 12 3 26 35 16 29 2 9 36 41 4 31 24 23 15 1 16 35 26 19 8 37 2 29 28 33 36 41 40 21 32 15 4 43 34 7 22 23 16 1 32 13 12 3 2 29 16 31 4 41 18 27 8 39 6 25 26 35 36 9 24 23 17 1 18 39 2 15 12 19 36 3 40 37 32 29 20 33 4 11 8 21 30 5 22 23 18 1 36 25 8 29 26 41 12 27 32 39 16 9 4 35 18 3 6 31 2 13 24 23 19 1 26 29 32 7 18 11 4 13 44 15 8 25 10 19 12 5 16 37 40 43 22 23 20 1 6 41 36 35 16 31 32 25 26 27 4 3 2 9 8 39 12 13 18 29 24 23 21 1 12 31 6 37 4 33 26 41 30 21 2 39 28 43 36 19 32 17 38 11 22 23 22 1 24 1 24 1 24 1 24 1 24 1 24 1 24 1 24 1 24 1 24 1 24 23 23 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 ord 1 – 11 – 22 – 22 – 11 – 22 – 11 – 22 – 22 – 22 – 22 – –

J(46) is calculated using theorem 4.9: J(46) = J(2 ∗ 23) = 1 ∗ 22 = 22. The number 7 is a primitive root of 46, because ord46 (7) = 2 = J(46). Theorem 4.14. 108,109 For a modulus n and a relative prime to n the following holds: ∗ {ai (mod n)|i = 1, . . . , J(n)} equals the multiplicative group Zn if and only if ordn (a) = J(n).

108

For prime moduli p all a with 0 < a < p are of order J(p) = p − 1. Compare table 5 for an example. In this case ai (mod n) goes through all the values 1, . . . , p − 1. Exhausting all possible values of the set is an important cryptographic proposition (compare theorem 4.5). This determines a permutation π(p − 1). 109 Table 6 demonstrates that for composite moduli n not all a are of maximal order J(n). In this example only 5, 7, 11, 15, 17, 19 and 21 are of order 22.

4.10

Proof of the RSA procedure with Euler-Fermat

Using the Euler-Fermat theorem, we can “prove” the RSA110,111 procedure in the group Z∗ . n 4.10.1 Basic idea of public key cryptography

The basic idea behind public key cryptography is that all participants possess a different pair of keys (P and S) and the public keys for all recipients are published. You can retrieve the public key P for a recipient from a directory just as you would look up someone’s phone number in the phone book. Furthermore, each recipient has a secret key S that is needed in order to decrypt the message and that is not known to anyone else. If the sender wishes to send a message M , he encrypts it using the public key P of the recipient before sending it: The cipher text C is determined as C = E(P ; M), where E (encryption) is the encryption rule. The recipient uses his private key S to decrypt the message with the decryption rule D : M = D(S; C). In order to ensure that this system works for every message M , the following four requirements must be met: 1. D(S; E(P ; M)) = M for every M (invertibility) and M takes “very many” of its possible values. 2. All (S, P) pairs are different for all participants (i.e. lots of them are needed). 3. The time required to derive S from P is at least as high as the time required to decrypt M with no knowledge of S. 4. Both C and M can be calculated relatively easily. The 1st requirement is a general condition for all cryptographic encryption algorithms. The 2nd requirement can easily be met because there is a “very” large number of prime numbers112 and because this can be ensured by a central office that issues certificates. It is this last requirement that makes the procedure actually usable. This is because it is possible to calculate the powers in a linear amount of time (because there is a restriction on the length of the numbers).

110

The RSA procedure is the most common asymmetric cryptography procedure. Developed in 1978 by Ronald Rivest, Adi Shamir and Leonard Adleman, it can be used both for signatures and for encryption. Cryptographers always associate this procedure with the abbreviation “RSA” – the following remark is meant with humor to show that each letter combination can be used with several meanings: In Britain the “Royal Society for the encouragement of Arts, Manufactures & Commerce” is commonly known as the “RSA”. 111 In literature and in movies not only classic but also modern cryptographic methods have been used (see appendix A.3). 112 According to the prime number theorem of Legendre and Gauss there are approximately n/ ln(n) prime numbers up to the number n. This means, for example, that there are 6.5 ∗ 1074 prime numbers under n = 2256 (= 1.1 ∗ 1077) and 3.2 ∗ 1074 prime numbers under n = 2255 . Between 2255 and 2256 there are therefore 3.3 ∗ 1074 prime numbers with precisely 256 bits. This large number is also the reason why we cannot simply save them all.

Although Whitfield Diffie and Martin Hellman formulated the general method as early as 1976, the actual procedure that met all four requirements was only discovered later by Rivest, Shamir and Adleman. 4.10.2 How the RSA procedure works

The individual steps for implementing the RSA procedure can be described as follows (see [Eckert2003, p. 213 ff] and [Sedgewick1990, p. 338 ff]). Steps 1 to 3 constitute key generation, steps 4 and 5 are the encryption, and steps 6 and 7 are the decryption: 1. Select two distinct random prime numbers113,114 p and q and calculate n = p ∗ q 115 . The value n is called the RSA modulus116 . 2. Select an arbitrary e ∈ {2, · · · , n − 1} such that117 : e is relatively prime to J(n) = (p − 1) ∗ (q − 1). We can then “throw away” p and q.118 3. Select d ∈ {1, · · · , n − 1} with e ∗ d ≡ 1 (mod J(n)), i.e. d is the multiplicative inverse of e modulo J(n)119,120 . We can then “throw away” J(n). → (n, e) is the public key P . → (n, d) is the secret key S (only d must be kept secret).

Compaq introduced the so-called multi-prime method with high marketing effort in 2000. n was the product of two big and one relative small prime: n = o ∗ p ∗ q. With theorem 4.10 we get: J(n) = (o − 1) ∗ (p − 1) ∗ (q − 1). This method did not assert itself yet. One reason probably is, that Compaq claimed a patent on it. Generally there is less understanding in Europe and with the Open Source Initiative , that one can claim patents on algorithms. But there is really no understanding outside the U.S., that one can get a patent for a special case (3 factors) of an algorithm (RSA), although the patent for the general case is almost expired. 114 If the two primes p and q are equal then (me)d ≡ m mod n is not true for all m < n (although e ∗ d ≡ 1 mod J(n) is fulfilled). Example: If n = 52 then according to theorem 4.11 it is J(n) = 5 ∗ 4 = 20, e = 3, d = 7, e ∗ d = 21 ≡ 1 mod J(n). But it is (53)7 ≡ 0 mod 25. 115 The GISA (German Information Security Agency) recommends, to choose the prime factors p and q almost the same, but not too close: 0.5 < | log2 (p) − log2 (q)| < 30. They recommend to generate the primes independently and check that the restriction is fulfilled (see [GISA2002]). 116 In CrypTool the RSA modulo is denoted with a capital “N ” . 117 It is recommended by cryptanalytic reasons, but not necessary to make RSA work, to select e such that: max(p, q) < e < J(n) − 1. 118 The procedure also allows us to select d freely and then calculate e. However, this has practical disadvantages. We usually want to be able to encrypt messages “quickly”, which is why we choose a public exponent e such that it has a short bit length compared to the modulus n and as few binary ones as possible (e.g. 216 + 1). So a fast exponentiation is possible when encrypting. We want to select the publicly known e to be an advantageous value that allows the exponential calculation to be performed quickly during encryption. The prime numbers 3, 17 and 65537 have proved to be particularly practical for this purpose. The most often used number is 65537 = 216 + 1, or in binary: 10 · · · 0 · · · 01 (this number is prime and therefor relatively prime to many other numbers). 119 For reasons of security, d should not be too small. 120 We start by determining either d or e depending on the implementation.

113

4. For encryption, the message represented as a (binary) number is divided into parts such that each part of the number is less than n. 5. Encryption of the plaintext (or the parts of it) $M \in \{1, \cdots, n-1\}$: $C = E((n, e); M) := M^e \pmod{n}$. 6. For decryption, the cipher text represented as a binary number is divided into parts such that each part of the number is less than n. 7. Decryption of the cipher text (or the parts of it) $C \in \{1, \cdots, n-1\}$: $M = D((n, d); C) := C^d \pmod{n}$. The numbers d, e and n are usually extremely large (e. g. d and e 300 bits, n 600 bits). Comment: The security of the RSA algorithm depends as with all public key methods on the difficulty to calculate the private key d from the public key (n, e). Concrete for the RSA method does this mean: 1. it is hard to calculate J(n) for big compounds n and 2. it is hard to calculate the prime factors of big compounds n (Factorisation!factorisation problem).

4.10.3

Proof of requirement 1 (invertibility)

For pairs of keys (n, e) and (n, d) that possess fixed properties in steps 1 to 3 of the RSA procedure, the following must be true for all $M < n$: $M \equiv (M^e)^d \pmod{n}$ with $(M^e)^d = M^{e*d}$.

This means that the deciphering algorithm above works correctly. We therefore need to show that: $M^{e*d} \equiv M \pmod{n}$. We will show this in 3 steps (see [Beutelspacher1996, p. 131ff]). Step 1: In the first step we show that: $M^{e\star d} \equiv M \pmod{p}$. This results from the requirements and from Euler-Fermat (theorem 4.13). Since $n = p * q$ and $J(p * q) = (p-1) * (q-1)$ and since e and d are selected in such a way that e ∗ d ≡ 1 (mod J(n)), there is a whole number k such that: e ∗ d = 1 + k ∗ (p − 1) ∗ (q − 1). M e∗d ≡ M 1+k∗J(n) ≡ M ∗ M k∗J(n) ≡ M ∗ M k∗(p−1)∗(q−1) (mod p) ≡ M ∗ (M p−1)k∗(q−1) (mod p) ≡ M ∗ (1)k∗(q−1) (mod p) ≡ M (mod p) The requirement for using the simplified Euler-Fermat theorem (theorem 4.12) was that M and p are relatively prime. Since this is not true in general, we need to consider the case when M and p are not relatively prime. Since p is a prime number, this implies that p is a factor of M . But this means: M ≡ 0 (mod p). If p is a factor of M , then p is also a factor of M e∗d . Therefore: M e∗d ≡ 0 (mod p). Since p is a factor of both M and M e ∗ d, it is also a factor of their difference: (M e⋆d − M) ≡ 0 (mod p). And therefore our conjecture is also true in this special case. Step 2: In exactly the same way we prove that: M e∗d ≡ M (mod q). Step 3: We now combine the conjectures from (a) and (b) for n = p ∗ q to show that: M e∗d ≡ M (mod n) for all M < n. From (a) and (b) we have (M e∗d − M) ≡ 0 (mod p) and (M e∗d − M) ≡ 0 (mod q). Therefore, p and q are both factors of the same number z = (M e∗d − M). Since p and q are distinct prime numbers, their product must also be a factor of this number z. Thus: (M e∗d − M) ≡ 0 (mod p ∗ q) or M e∗d ≡ M (mod p ∗ q) or M e∗d ≡ M (mod n). based on little Fermat : M p−1 ≡ 1 (mod p)

1st comment: We can also condense the three steps if we use the theorem 4.13 (Euler-Fermat) - i.e. not the simplified theorem where n = p and which corresponds to Fermat’s Little Theorem: (M e)d ≡ M e∗d ≡ M (p−1)(q−1)∗k+1 ≡ (M (p−1)(q−1) ≡M J(n) ≡1 (mod n)

)k ∗ M ≡ 1k ∗ M ≡ M (mod n).

2nd comment: When it comes to signing messages, we perform the same operations but first use the secret key d, followed by the public key e. The RSA procedure can also be used to create digital signatures, because: $M \equiv (M^d)^e \pmod{n}$.

4.11

Considerations regarding the security of the RSA algorithm[121]

There have always been discussions about the suitability of the RSA algorithm for digital signatures and encryption, e. g. after publications of breakthroughs in factorisation. Nevertheless the RSA algorithm has become a de-facto standard since it was published more than 20 years ago (compare 7.1). The security of the RSA algorithm rests – as with all cryptographic methods – on the following 4 central pillars: • the complexity of the number theoretical problem on which the algorithm is based (here factorisation of big numbers), • the election of fitting parameters (here the length of the module N), • the adequate usage of the algorithm and key generation and • the correct implementation of the algorithm. Usage and key generation are well understood today. Implementation based on long integer arithmetic is very easy. The following sections examine the RSA algorithm with respect to the first two points. 4.11.1 Complexity

Successful decryption or forgery of a signature – without knowing the private key – requires calculating the e-th root mod n. The private key, this is the multiplicative inverse of e mod J(n), can be easily determined if J(n) is known. J(n) again can be calculated from the prime factors of n. Breaking of RSA therefore cannot be more difficult than factorisation of the module n. The best factorisation method known today is a further development of the General Number Field Sieve (GNFS) , which was originally devised to factor only numbers of a special form (like Fermat numbers). The complexity of solving the factorisation problem with the GNFS is asymptotically $O(l) = e^{c \cdot (l \cdot \ln 2)^{1/3} \cdot (\ln(l \cdot \ln(2)))^{2/3} + o(l)}$

Please refer to: • A. Lenstra, H. Lenstra: The development of the Number Field Sieve [Lenstra1993].

---

[121] Major parts of chapters 4.11.1 and 4.11.2 follow the article “Vorzüge und Grenzen des RSA-Verfahrens” written by F. Bourseau, D. Fox and C. Thiel [Bourseau2002]

- Robert D. Silverman: A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths [Silverman2000]. This formula shows, that the factorisation problem belongs to the class of problems with subexponential time complexity (i. e. time complexity grows asymptotically not as fast as exponential √ functions like el or 2l , but strictly slower, e. g. like e l). This classification is all that is currently known; it does not preclude the possibility that the factorisation problem can be solved in polynomial time (see 4.11.5). O(l) is the average number of processor steps depending on the bit length l of the number n to be factorised. For the best currently known factorisation algorithm the constant c = (64/9)1/173 = 1, 923. The inverse proposition, that the RSA algorithm can be broken only by factorisation of n, is still not proven. Most number theorists consider the "RSA problem" and the factorisation problem equivalent in terms of time complexity. Please refer to: Handbook of Applied Cryptography [Menezes2001].

4.11.2

Security parameters because of new algorithms

Factorisation algorithms The complexity is basically determined by the length l of the module n. Higher values for this major parameter are oriented at the possibilities of the current algorithms for factorization: • In 1994 a 129-digit RSA module (428 bit), published in 1977, was factorised by a distributed implementation of the Quadratic Sieve algorithm (QS), developed 1982 by Pomerance. This effort took 8 months. Please refer to: C. Pomerance: The quadratic sieve factoring algorithm [Pomerance1984]. • In 1999 a 155-digit module (512 bit) was factored with an implementation of the General Number Field Sieve algorithm (GNFS) , developed by Buhler, Lenstra and Pomerance. The GNFS is more efficient than QS if n is longer than about 116 decimal digits. This effort took 5 months. Please refer to: J.P. Buhler, H.W. Lenstra, C. Pomerance: Factoring integers with the number field sieve [Buhler1993]. This made evident that a module length of 512 bit no longer prevents from attackers. And also past 1999 further factorization progress was made up to now (see RSA-200 in chapter 4.11.4).

Lattice base reduction algorithms The module length is not the only parameter relevant for security. Beneath requirements from implementation and engineering the sizes and the proportions of the parameters e, d and N are relevant. According attacks based on lattice reductions are a real threat for (too) simple implementations of RSA. Theses attacks can be structured into the following four categories: • Attacks against very small public keys e (e.g. e = 3). • Attacks against relatively small private exponents d(e.g. $d < N^{0.5}$). • Factorization of the modul N, if one of the factors p or q is partly known. • Attacks requireing, that a part of the private key d is known. A good overview about the current status of publications concerning these attacks can be found in the diploma thesis of Matthias Schneider [SchneiderM2004].

4.11.3

Forecasts about factorization of large integers

Within the last 20 years a lot of progress has been made. Estimations about the future development of the ability to factor RSA modules vary and depend on some assumptions: • progression in computing performance (Moore’s law: every 18 month the computing power will double) and in grid computing. • development of new algorithms. Within the last years the module bit length feasible for factorisation increased – even without new algorithms – by 10 bit per year. Larger numbers require not only more time to be factored, but also huge RAM storage for the solutions matrix being used by the best algorithms known today. This need for storage grows like the square root of the computation time, i. e. also subexponentially. Because RAM availability increased exponentially in the recent decades, it seems that this should not be the limiting factor. An estimation of the evolution of secure key lengths was done by Lenstra/Verheul[Lenstra1999] (compare figure 2 in chapter 7.1). Within the article [Bourseau2002] Dirk Fox[122] published his prognosis of an almost linear factorization progression, if all influencing factors are included: Each year the module length feasible

---

[122] His company Secorvo Ltd also delivered a statement on the recommendation for key length selection published by the GISA (German Information Security Agency). Chapter 2.3.1 of this statement contains a competent and understandable discussion of RSA security (this document exists – to my knowledge – only in German): http://www.secorvo.de/publikat/stellungnahme-algorithmenempfehlung-020307.pdf

for factorisation increases by 20 bit on average. So his forecast was below the more optimistic estimations of GISA and NIST. This forcast by Dirk Fox from the year 2001 seems to prove true by the factorisation record of RSA-200 (see chapter 4.11.4). His estimation for the year 2005, to achieve a bit length of 660 bit, was almost a precision landing (compare figure 1). If the forecast withstand in the future then the factorization of an RSA modul of 1024 bit can be expected in 15 years.

1200 1000

Bits

800 600 400 200 0 1970

1975

1980

1985

1990

1995

2000

2005

2010

2015

2020

Figure 1: Forecast about future factorisation records compared with current results (from Secorvo)

Hermann Hesse[123] : To let the possible happen, you again and again have to try the impossible.

4.11.4

Status regarding factorisation of concrete large numbers

The web pages http://www.crypto-world.com and http://www.tutorgig.com/ed/RSA number[124] contain exhaustive overviews about the factoring records of composed integers using different methods. The current record (as of May 2005) obtained using the GNFS method (General Number Field Sieve) factorised a 200 decimal digit into its both prime factors. The last records[125] with factorisation algorithms for composed numbers are listed in the following table: RSA-155 C158 RSA-160 RSA576 C176 RSA600 RSA-200 Decimal digits 155 158 160 174 176 193 200 Binary digits 512 523 530 576 583 600 663 Factored on August, 1999 January, 2002 April, 2003 December, 2003 May, 2005 November, 2005 May, 2005 Factored by Herman te Riele et al. Jens Franke et al. Jens Franke et al. Jens Franke et al. Kazumaro Aoki et al. Jens Franke et al. Jens Franke et al.

Table 28: The current general purpose factoring records (as of November 2005)

Below the last records are explained in more detail.

---

[123] Hermann Hesse, German/Swiss writer and Nobel Prize winner, July 2, 1877 – August 9, 1962. This site was not quite up-to-date end of May 2005: RSA-200 was missing.
[125] The ’RSA numbers’ are certain large semiprime numbers (i.e., numbers with exactly two prime factors). They were generated and published by the company RSA Security and they form the basis of the RSA Factoring Challenge, in which factorisations for these numbers are sought. See http://www.rsasecurity.com/rsalabs/challenges/factoring/numbers.html. The first RSA Factoring Challenge labelled the numbers, from RSA-100 to RSA-500, according to their number of decimal digits; the second RSA Factoring Challenge labelled the numbers according to their number of binary digits. Within the second challenge cash prizes have been offered for successful factorisations of RSA576 to RSA2048 (RSA576, RSA640 etc. using 64 bit steps upwards). An exception to this is RSA617, which was created prior to the change in the numbering scheme. The researchers around Professor Jens Franke (from the University of Bonn, the GISA and the CWI) do not aim on getting cash prizes but in extending the research limits. So statements about the necessary length of a secure RSA modulus are more well-founded. The ’C-numbers’ originate from the Cunningham project: http://www.cerias.purdue.edu/homes/ssw/cun/
124

RSA-155 On August 22, 1999 researchers from the Netherlands found the solution of this RSA challenge. They factorised a 155-digit number into its both 78-digit primes (see chapter 4.11.2). This 512 bit RSA-155 meant to reach a kind of magic border.

C158 On January 18, 2002 researchers at the German University of Bonn126 factorised a 158-digit decimal number into its both prime factors (these are build with 73 and 86 decimal digits) using the GNFS method (General Number Field Sieve). This record got much less attention within the press than the solution of RSA-155. The task of the researchers from Bonn was not initiated by a challenge, but they wanted to find the last prime factors of the integer 2953 − 1 (see “Wanted List” of the Cunningham Project127). The 6 smaller prime factors, already found before have been: 3, 1907, 425796183929, 1624700279478894385598779655842584377, 3802306738549441324432139091271828121 and 128064886830166671444802576129115872060027. The first 3 factors can be easily computed128 . The next three prime factors were found by P. Zimmerman129 , T. Grandlund130 and R. Harley during the years 1999 and 2000 using the elliptic curve factorisation method. The last remaining factor, called “C158”, was known to be composite by then, but its factors where not known (the following 3 lines contain one number): 395058745832651445264197678006144819960207764603049364
454139376051579355626529450683609727842468219535093544
3058704902519956553357102097992264849779494429556033 The factorisation of C158 resulted in the following two 73- and 86-digit prime factors: 3388495837466721394368393204672181522815830368604993048084925840555281177 and 1165882340667125990314837655838327081813101
2258146392600439520994131344334162924536139.

126 127

http://www.ercim.org/publication/Ercim News/enw49/franke.html, 2002-01 Cunningham project: http://www.cerias.purdue.edu/homes/ssw/cun/ 128 E.g. using CrypTool via menu Indiv. Procedures \ RSA Cryptosystem \ Factorisation of a Number. CrypTool can factorize in a reasonable time numbers no longer than 250 bit. Numbers bigger than 1024 bits are currently not accepted by CrypTool. 129 http://www.loria.fr/~zimmerma/ecmnet 130 http://www.swox.se/gmp/

So now all 8 prime factors of 2953 − 1 have been found. Links: http://www.loria.fr/ zimmerma/records/gnfs158 http://www.crypto-world.com/FactorRecords.html http://www.crypto-world.com/announcements/c158.txt

RSA-160 On January 18, 2002 researchers at the German University of Bonn131 factorised a 160-digit number into its both prime factors (these are build with each 80 decimal digits) using the GNFS method (General Number Field Sieve). The computations for the factorization of RSA-160 also took place at the German Information Security Agency (GISA) in Bonn132 . The 160-digit decimal number origins from the old challenge list of RSADSI. This number was retracted after RSA-155 (RSA512) had been factorized successfully. The prime factors of RSA160 were still unknown. So this record of the team of Prof. Franke provides the solution of the old challenge, for which RSADSI didn’t award a price anymore. The composite number called “RSA-160” is (the following 3 lines contain one number):
215274110271888970189601520131282542925777358884567598017049
767677813314521885913567301105977349105960249790711158521430
2079314665202840140619946994927570407753 The factorisation of RSA-160 resulted in the following two prime factors: p = 45427892858481394071686190649738831
656137145778469793250959984709250004157335359 and q =
47388090603832016196633832303788951 973268922921040957944741354648812028493909367
The calculations took place between December 2002 and April 2003.

http://www.loria.fr/~zimmerma/records/rsa160 http://www.loria.fr/~zimmerma/records/factor.html http://www.crypto-world.com/FactorWorld.html 132 Every year the GISA creates a paper to describe which crypto algorithms are feasible to generate digital signatures according to the German signature law – under participation of experts from economy and science. To review signature methods based on the factorisation problem the GISA also co-operates with researchers from the University of Bonn. Further information about crypto algorithms can be found on the web page of GISA: http://www.bsi.bund.de/esig/basics/techbas/krypto/index.htm.

131

RSA-200 On May 9, 2005 the research group of Prof. Jens Franke at the German University of Bonn133 announced, that they achieved a new world record in number factorization together with their colleagues of the Amsterdam Centrum voor Wiskunde en Informatica. They factorised a 200-digit number into its both prime factors (these are build with each 100 decimal digits) using the GNFS method (General Number Field Sieve). The composite number called “RSA-200” is (the following 3 lines contain one number):
2799783391122132787082946763872260162107044678695542853756000992932
6128400107609345671052955360856061822351910951365788637105954482006
576775098580557613579098734950144178863178946295187237869221823983 The factorisation of RSA-200 resulted in the following two prime factors: p =
35324619344027701212726049781984643686711974001976
25023649303468776121253679423200058547956528088349 and q =
79258699544783330333470858414800596877379758573642
19960734330341455767872818152135381409304740185467 The calculations took place between December 2003 and May 2005. The factorization done by the group around Bahr, B¨hm, Franke, Kleinjung, Montgomery and te Riele lasted almost 17 o months. The operating expense of the calculations was about 120,000 MIPS-years134 .

Size of factorized numbers compared to primality proven numbers As you notice the factorised compound numbers built of 2 prime factors are much smaller than the especially structured numbers, for which primality tests are able to decide whether these numbers are prime or not (see chapters 3.4, 3.5 and 3.6). Length in bits of the current world records: (RSA-200 ←→ 43rd known Mersenne prime) 663 ←→ 30, 402, 457.

133 134

http://www.loria.fr/~zimmerma/records/rsa200 A MIPS-year (MY) is the quantity of operations a machine can perform in one year, if the machine constantly achieves one million integer operations per second (MIPS). For illustration: a INTEL Pentium 100 processor achieves about 50 MIPS. To factorize a 2048 bit module it is estimated to need about 8.5 · 1040 MY.

4.11.5

Further current research about primes and factorisation

Prime numbers are part of very many topical research areas in number theory and computer science. Progress made with factorisation is bigger than was estimated 5 years ago – this is not only due to faster computers but also new knowledge. The security of the RSA algorithm is based on the empirical observation that factoring large numbers is a hard problem. A module n (typically, 1024 bit) can be easily constructed as the product of two large primes p, q (typically, 500–600 bit each), by calculating n = pq. However, it is a hard problem to extract p, q from n. Without knowing p or q, the private key cannot be calculated. Thus, any progress in efficiency of factorising large integers will effect the security of the RSA. As a consequence, the underlying primes p, q and, thus, the module n (1024 bit as of today) have to be increased. In case of a quantum leap in factorisation, the RSA algorithm might be compromised.

Bernstein’s paper and its implication on the security of the RSA algorithm In his paper “Circuits for integer factorisation: a proposal” (http://cr.yp.to/djb.html), published November 2001, D. J. Bernstein [Bernstein2001] addresses the problem of factorising large integers. Therefore, his results are of relevance from a RSA point of view. As a main result Bernstein claims that the implementation of the General Number Field Sieve algorithm (GNFS) can be improved to factor, with the same effort as before, integers with three times more digits. We note that the definition of effort is a crucial point: Bernstein claims that effort is the product of time and costs of the machine (including the memory used). The gist of the paper lies in the fact that he can reduce a big part of factorising to sorting. Using Schimmler’s scheme, sorting can be optimized by massive parallel computing. At the end of section 3 Bernstein explains this effect: The costs of m2 parallel computers with a constant amount of memory is a constant times m2 . The costs of a computer with a single processor and memory of size m2 is also of the order of m2 , but with a different constant factor. With m2 processors in parallel, sorting of m2 numbers (with Schimmler’s scheme) can be achieved in time m, while a m2 -memory computer needs time of the order of m2 . Decreasing memory and increasing the number of processors, the computing time can be reduced by a factor 1/m without additional effort in terms of total costs. In section 5 it is said that massive parallel computing can also increase efficiency of factorising using Lenstra’s elliptic-curve-method (a search algorithm has costs that increase in a quadratic square manner instead of cubically). We note that all results achieved so far are asymptotic results. This means that they only hold in the limit n to infinity. Unfortunately, there is no upper limit for the residual error (i.e. the difference between the real and the asymptotic value) for finite n – a problem which has already been addressed by the author. As a consequence, one cannot conclude whether the costs (in the sense of Bernstein) for factorising 1024–2048-bit RSA modules can be significantly reduced. There is no doubt that Bernstein’s approach is innovative. However, the reduction of computing time under constant costs comes along with a massive use of parallel computing – a scenario which seems not to be realistic yet. For example, formally 1 sec computing time on one machine and 1/1,000,000 sec time parallel computing time on 1,000,000 machines might have same costs. In reality, it is much harder to realize the second situation, and Bernstein does not take into account the fixed costs, in particular for building a network between all these computers. Although distributed computing over a large network might help to overcome this problem, realistic costs for data transfer have to be taken into account – a point which was not addressed in Bernstein's proposal. As long as there is neither (low cost) hardware nor a distributed computing approach (based on Bernstein's ideas), there should not be a problem for RSA. It has to be clarified from which magnitude of n on Bernstein's method could lead to a significant improvement (in the sense of the asymptotic result). Arjen Lenstra, Adi Shamir et. al. analyzed the paper of Bernstein [Lenstra2002]. In summary they expect a factorisation improvement on how much longer the bit length of the keys could be with a factor of 1.17 (instead of factor 3 as proposed by Bernstein). The abstract of their paper "Analysis of Bernstein's Factorization Circuit" says: "... Bernstein proposed a circuit-based implementation of the matrix step of the number field sieve factorisation algorithm. We show that under the non-standard cost function used in [1], these circuits indeed offer an asymptotic improvement over other methods but to a lesser degree than previously claimed: for a given cost, the new method can factor integers that are 1.17 times larger (rather than 3.01). We also propose an improved circuit design based on a new mesh routing algorithm, and show that for factorisation of 1024-bit integers the matrix step can, under an optimistic assumption about the matrix size, be completed within a day by a device that costs a few thousand dollars. We conclude that from a practical standpoint, the security of RSA relies exclusively on the hardness of the relation collection step of the number field sieve." RSA Security's analysis of the Bernstein paper [RSA Security 2002] from April, 8 2002 also – as expected – concludes, that RSA is still not compromised. This is still an ongoing discussion. When this section was written (June 2002) nothing was publicly known about, how far there exist implementations of his theoretical onsets and how much financing there was for his research project. Links: http://cr.yp.to/djb.html http://www.counterpane.com/crypto-gram-0203.html#6 http://www.math.uic.edu The TWIRL device In January 2003 Adi Shamir and Eran Tromer from the Weizmann Institute of Science published a preliminary draft called "Factoring Large Numbers with the TWIRL Device" raising concerns about the security of key sizes till 1024 bits [Shamir2003].

Their abstract summarizes their results very well: “The security of the RSA cryptosystem depends on the difficulty in factoring large integers. The best current factoring algorithm is the Number Field Sieve (NFS), and its most difficult part is the sieving step. In 1999 a large distributed computation involving thousands of workstations working for many months managed to factor a 512-bit RSA key, but 1024-bit keys were believed to be safe for the next 15-20 years. In this paper we describe a new hardware implementation of the NFS sieving step ... which is 3-4 orders of magnitude more cost effective than the best previously published designs ... . Based on a detailed analysis of all the critical components (but without an actual implementation), we believe that the NFS sieving step for 1024-bit RSA keys can be completed in less than a year with a $10M device, and that the NFS sieving step for 512-bit RSA keys can be completed in less than ten minutes with a $10K device. Coupled with recent results about the difficulty of the NFS matrix step ... this raises some concerns about the security of these key sizes.” A detailed explanation from these two authors also can be found in the RSA Laboratories CryptoBytes [Shamir2003a]. The 3-page article in the DuD issue of June 2003 [Weis2003] contains a very good explanation, how the attack using the Generalized Number Field Sieve (GNFS) works and which progress is made, to factorize numbers. At GNFS we can distinguish 2 general steps: The sieve step (relation collecting) and the matrix reduction. Besides the sieve step is highly parallelizable, it dominates the overall calculation burdon. Shamir and Tromer haven’t built a TWIRL device yet, but the estimated costs of 10 till 50 million Euro (in order to factorise a 1024-bit number) is not prohibitive for secret agencies or big criminal organizations, because the “costs for a single espionage satellite is estimated e.g. to be several billion USD”. The authors therefore recommend, to get as soon as possible rid of today used sensible RSA, Diffie-Hellman or ElGamal keys up to 1024 bit and to use then keys of at least 2048 bit length. The planned TCPA/Palladium hardware will use 2048-bit RSA keys! So recommendations like the ones from the GISA (German Information Security Agency) to use higher key lengths are very valid.

“Primes in P”: Primality testing is polynomial In August 2002 the three Indian researchers M. Agrawal, N. Kayal and N. Saxena published the paper “PRIMES in P” about a new primality testing algorithm called AKS [Agrawal2002]. They discovered a polynomial time deterministic algorithm for determining if a number is prime or not. The importance of this discovery is that it provides number theorists with new insights and opportunities for further research. Lots of people over centuries have been looking for a polynomial time test for primality, and this result is a major theoretic breakthrough. It shows that new results can be generated from already known facts. But even its authors note that other known algorithms may be faster (for example ECPP). The new algorithm works on any integer. For example the GIMPS project uses the Lucas-Lehmer primality test which takes advantage of the special properties of Mersenne numbers. This makes the Lucas-Lehmer test much faster, allowing to test numbers with millions of digits while general purpose algorithms are limited to numbers with a few thousand digits. Current research results on this topic can be found at: http://www.mersenne.org/ http://fatphil.org/maths/AKS/ Original paper in English http://ls2-www.cs.uni-dortmund.de/lehre/winter200203/kt/material/primes.ps Good explanation in German by Thomas Hofmeister.

Joanne K. Rowling[135] : It is our choices, that show what we truly are, far more than our abilities.

4.12

Applications of asymmetric cryptography using numerical examples

The results of modular arithmetic are used extensively in modern cryptography. Here we will provide a few examples from cryptography using small[136] numbers. Enciphering a text entails applying a function (mathematical operation) to a character string (number) to generate a different number. Deciphering entails reversing this function, in other words using the distorted image that the function has created from the plaintext in order to restore the original image. For example, the sender could take the plaintext M of a confidential message and add a secret number, the key S, to obtain the cipher text C: $C = M + S$. The recipient can reconstruct the plaintext by reversing this operation, in other words by subtracting S: $M = C - S$. Adding S reliably makes the plaintext impossible to read. However, this encryption is rather weak, because all an interceptor needs to do to calculate the key is obtain a plaintext and the associated cipher text $S = C - M$, and can then read any subsequent messages encrypted using S. The essential reason for this is that subtraction is just as simple an operation as addition.

4.12.1 One way functions

If the key is to be impossible to determine even with knowledge of both the plaintext and the cipher text, we need a function that is, on the one hand, relatively easy to calculate – we don’t want to have problems encrypting messages. On the other hand, the inverse function should exist (otherwise information would be lost during encryption), but should be de facto incalculable. What are possible candidates for such a one way function? We could take multiplication rather than addition, but even primary school children know that the inverse function, division, is only slightly more difficult than multiplication itself. We need to go one step higher in the hierarchy of calculation methods. It is still relatively simple to calculate the power of a number, but the corresponding two reverse functions – taking roots (find b in the equation $a = bc$ when a and c

---

Joanne K. Rowling, “Harry Potter and the Chamber of Secrets”, Bloomsbury, 1998, last chapter “Dobby’s reward”, p. 245, by Dumbledore.

[136] In the RSA procedure, we call numbers “small” if the bit lengths are much less than 1024 bits (i.e. 308 decimal points). In practice, 1024 bits is currently the minimum length for a secure Certification Authority RSA modulus.

135

are known) and calculating logarithms (find c in the above equation when a and b are known) are so complicated that pupils normally do not learn them at school. Although a certain structure can still be recognised for addition and multiplication, raising numbers to the power of another or calculating exponentials totally mixes up all the numbers. Knowing a few values of the function doesn't tell us much about the function as a whole (in contrast to addition and multiplication).

4.12.2 The Diffie-Hellman key exchange protocol

Whitfield Diffie, Martin E. Hellman and Ralph Merkle developed this DH key exchange protocol in Stanford in 1976[137]. Alice and Bob[138] use a one way function to obtain a key S, the session key, for subsequent correspondence. This is then a secret that is only known to the two of them. Alice selects a random number a and keeps it secret. She applies a one way function to a to calculate the number $A = g^a$ and sends it to Bob. He does the same, by selecting a secret random number b, calculating $B = g^b$ and sending it to Alice. The number g is random and can be publicly known. Alice applies the one way function together with her secret number a to B, while Bob does the same with his secret number b and the received number A. The result S is the same in each case because the one way function is commutative: $(g^a)^b = (g^b)^a$. But even Bob cannot reconstruct Alice's secret number a from the data available to him, while Alice cannot determine Bob's secret number b. And a perpetrator who knows g and has intercepted both A and B cannot use this knowledge to determine a, b or S.

---

[137] With CrypTool this exchange protocol has been visualized: you can execute the single steps with concrete numbers using menu Indiv. Procedures \ Protocols \ Diffie-Hellman Demonstration.

[138] Bob and Alice are the default names used for the two authorised participants in a protocol (see [Schneier1996, p. 23]).

```
#

¨


#

¨


Alice
Alice generates number randomly g

Bob
Public: g

#

-g
B = gb

Bob generates number randomly

?
a

-

A = ga

?

?

? #
b

A secret key

? PP

?
S :=

Ba

#

B

# ) #

PP ## PP ## #P ## PP ## P

PP

? #B ##
secret key S := Ab

?
```

PA q P

-

?
S

?
S

#

©

#

©

Procedure: Alice and Bob want to negotiate a secret session key S via a channel that may be intercepted. 1. They select a prime number p and a random number g and exchange this information openly. 2. Alice now selects a, a random number less than p and keeps it secret. Similarly, Bob selects b, a random number less than p and keeps it secret. 3. Alice now calculates A ≡ g a (mod p). Bob calculates B ≡ g b (mod p). 4. Alice sends the result A to Bob. Bob sends the result B to Alice. 5. In order to now determine the session key to be used by both, they both separately raise the respective results they have received to the power of their secret random number modulo p. This means: - Alice calculates S ≡ B a (mod p) and - Bob calculates S ≡ Ab (mod p). Even if a spy intercepts g, p, and the interim results A and B, he cannot use these to determine the session key used due to the difficulty of calculating the discrete logarithm. We will now use an example with (unrealistically) small numbers to illustrate this.

Example using numbers: 1. Alice and Bob select g = 11, p = 347. 2. Alice selects a = 240, Bob selects b = 39 and they keep a and b secret. 3. Alice calculates A ≡ g a ≡ 11240 ≡ 49 (mod 347). Bob calculates B ≡ g b ≡ 1139 ≡ 285 (mod 347). 4. Alice sends Bob: A ≡ 49, Bob sends Alice: B ≡ 285. 5. Alice calculates B a ≡ 285240 ≡ 268 (mod 347), Bob calculates Ab ≡ 4939 ≡ 268 (mod 347). Alice and Bob can now communicate securely using their shared session key. Even if spies were to intercept everything transferred via the connection: g = 11, p = 347, A = 49 and B = 285, they would not be able to calculate the secret key. Comment: In this example using such small numbers, it would be possible, but with large numbers the discrete logarithm problem139,140 is extremely difficult to solve. Here, we need to calculate: For Alice: 11x ≡ 49 (mod 347), that means log11 (49) (mod 347). For Bob: 11y ≡ 285 (mod 347), that means log11 (285) (mod 347).

4.13

The RSA procedure with actual numbers

Having described above how the RSA procedure works, we will now work through the steps using actual, but small, numbers.

139

If you try to determine the discrete logarithm x that solves the equation 11x ≡ 49 (mod 347) with Mathematica by means of Solve, you obtain the em tdep message “The equations appear to involve the variables to be solved for in an essentially non-algebraic way”. Mathematica therefore claims not to know a direct algebraic procedure for solving the equation. Yet Mathematica is able to calculate this with the general function for the multiplicative order (here for Alice): MultiplicativeOrder[11, 347, 49] delivers the value 67. The syntax with Pari-GP is: znlog(Mod(49,347),Mod(11,347)). Such number-theory tasks can also be solved using other tools such as the LiDIA or BC package (see web links in appendix). The dl function in the LC user interface for LiDIA also delivers the value 67 for dl(11,49,347). 140 Why have the functions delivered the value 67 rather than 240 for the dl problem for Alice? The discrete logarithm is the smallest natural exponent that solves the equation 11x ≡ 49 (mod 347). Both x = 67 and x = 240 (the number selected in the example) satisfy the equation and can therefore be used to calculate the session key: 285240 ≡ 28567 ≡ 268 (mod 347). If Alice and Bob had selected a primitive root modulo p as base g, then for every remainder from the set {1, 2, · · · , p − 1} there is exactly one exponent from the set {0, 1, · · · , p − 2}. For info: there are 172 different primitive roots for modulo 347, 32 of which are prime (not necessary). Since the number 11 selected for g in the example is not a primitive root of 347, the remainders do not take all values from the set {1, 2, · · · , 346}. Thus, for a particular remainder there may be more than one exponent or even no exponent at all in the set {0, 1, · · · , 345} that satisfies the equation. PrimeQ[347] = True; EulerPhi[347] = 346; GCD[11, 347] = 1; MultiplicativeOrder[11, 347] = 173 The syntax with Pari-GP is: isprime(347); eulerphi(347); gcd(11,347); znorder(Mod(11,347)).

4.13.1

RSA with small prime numbers and with a number as message

Before applying the RSA procedure to a text, we will first demonstrate it directly using a single number as message141 . 1. Let the selected prime numbers be p = 5 and q = 11. Thus, n = 55 and J(n) = (p − 1) ∗ (q − 1) = 40. 2. e = 7 (should lie between 11 and 40 and must be relatively prime to 40). 3. d = 23 (since 23 ∗ 7 ≡ 161 ≡ 1 (mod 40)), → Public key of the recipient: (55, 7), → Private key of the recipient: (55, 23). 4. Let the message be the number M = 2 (so no division into blocks is required). 5. Encryption: C ≡ 27 ≡ 18 (mod 55). 6. The cipher text is simply the number C = 18 (we therefore do not need to divide it into blocks). 7. Decryption: M ≡ 1823 ≡ 18(1+2+4+16) ≡ 18 ∗ 49 ∗ 36 ∗ 26 ≡ 2 (mod 55). We will now apply the RSA procedure to a text, first using the upper case alphabet (26 characters), then using the entire ASCII character set as the basis for the messages.

i	0	1	2	3	67	172	173	174	175	176	240
11i mod 347	1	11	121	290	49	284	1	11	121	290	49

141

searched exponent = multiplicative order of 11 (mod 347)

searched exponent

Using CrypTool you can solve this with the menu Indiv.Procedures \ RSA Cryptosystem \ RSA Demonstration.

4.13.2

RSA with slightly larger primes and a text of upper case letters

We have the text “ATTACK AT DAWN” and the characters are coded in the following simple manner142 : Table 7: capital letters alphabet Character Numerical value Character Blank 0 M A 1 N 2 O B C 3 P D 4 Q E 5 R F 6 S G 7 T H 8 U I 9 V J 10 W K 11 X L 12 Y Z Numerical value 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Key generation (steps 1 to 3): 1. p = 47, q = 79 (n = 3, 713; J(n) = (p − 1) ∗ (q − 1) = 3, 588). 2. e = 37 (should lie between 79 and 3, 588 and must be relatively prime to 3, 588). 3. d = 97 (since e ∗ d = 1 mod J(n); 37 ∗ 97 ≡ 3, 589 ≡ 1 (mod 3, 588))143 . 4. Encryption: Text: A T T A C K A T D A W N Number: 01 20 20 01 03 11 00 01 20 00 04 01 23 14 This 28-digit number is divided into 4-digit parts (because 2, 626 is still smaller than n = 3, 713): 0120 2001 0311 0001 2000 0401 2314 All 7 parts are encrypted using: C ≡ M 37 (mod 3, 713)144 : 1404 2932 3536 0001 3284 2280 2235 5. Decryption: Cipher text: 1404 2932 3536 0001 3284 2280 2235 This 28-digit number is divided into 4-digit parts.

142

Using CrypTool you can solve this with the menu Indiv.Procedures \ RSA Cryptosystem \ RSA Demonstration. This is also described in the tutorial/scenario in CrypTool’s online help [Options, specify alphabet, number system, block length 2 and decimal representation]. 143 How to compute d = 97 using the extended gcd algorithm is shown in appendix A of this chapter 144 See Appendix E of this chapter for source code to do RSA encryption using Mathematica and Pari-GP. You can also encrypt the message with CrypTool via the menu path Indiv. Procedures \ RSA Cryptosystem \ RSA Demonstration.

All 7 parts are decrypted using: M ≡ C 97 (mod 3, 713): 0120 2001 0311 0001 2000 0401 2314 The 2-digit numbers are transformed into capital letters and blanks. Using the selected values it is easy for a cryptanalyst to derive the secret values from the public parameters n = 3, 713 and e = 37 by revealing that 3, 713 = 47 ∗ 79. If n is a 768-bit number, there is, according to present knowledge, little chance of this. 4.13.3 RSA with even larger primes and a text made up of ASCII characters

In real life, the ASCII alphabet is used to code the individual characters of the message as 8-bit numbers. The idea for this task145 is taken from the example in [Eckert2003, p. 271]. Coded in decimal notation, the text “RSA works!” is as follows: Text: R S A w o r k s ! Number: 82 83 65 32 119 111 114 107 115 33 We will work through the example in 2 variants. The steps 1 to 3 are common for both. Key generation (steps 1 to 3): 1. p = 503, q = 509 (n = 256, 027; J(n) = (p − 1)(q − 1) = 255, 016 = 23 ∗ 127 ∗ 251)146 . 2. e = 256, 027 (should lie between 509 and 255, 016 and must be relatively prime to 255, 016)147 . 3. d = 231, 953 (since e ≡ d−1 mod J(n) : 65, 537 ∗ 231, 953 ≡ 15, 201, 503, 761 ≡ 1 (mod 67, 000))148 . Variant 1: All ASCII characters are en-/decrypted separately (no blocks are formed). 4. Encryption: Text: R S Number: 82 83

A 65

32

w 119

o 111

r 114

k 107

s 115

! 33

The letters are not combined149 !

Using CrypTool you can solve this via the menu path Indiv.Procedures \ RSA Cryptosystem \ RSA Demonstration. 146 See Appendix E of this chapter for the source code to factorise the number J(n) using Mathematica and Pari-GP. Using CrypTool you can solve this with the Indiv.Procedures \ RSA Cryptosystem \ Factorisation of a Number. 147 e cannot, therefore, be 2, 127 or 251 (65537 = 216 + 1). In real life, J(n) is not factorised but rather the Euclidean algorithm is used for the selected e to guarantee that gcd(d, J(n)) = 1. 148 Other possible combinations of (e, d) include: (3, 170, 011), (5, 204, 013), (7, 36, 431). 149 For secure procedures we need large numbers that assume – as far as possible – all values up to n−1. If the possible value set for the numbers in the message is too small, even large prime numbers cannot make the procedure secure. An ASCII character is represented by 8 bits. If we want larger values we must combine several numbers. Two characters need 16 bits, whereby the maximum value that can be represented is 65536. The modulus n must then be greater than 216 = 65536. This is applied in variant 2. When the numbers are combined, the leading zeros are kept in binary notation (just as if we were to write all numbers with 3 digits in decimal notation above and were then to obtain the sequence 082 083, 065 032, 119 111, 114 107, 115 033).

Each character is encrypted using: C = M 65,537 (mod 256, 027)150 : 212984 025546 104529 031692 248407 100412 054196 100184 058179 227433 5. Decryption: Cipher text: 212984 100412 025546 054196 104529 100184 031692 058179 248407 227433

Each character is decrypted using: M ≡ C 231,953 mod 256, 027: 82 83 65 32 119 111 114 107 115 33 Variant 2: The ASCII characters are en-/decrypted two at a time as blocks. In variant 2 the block formation is done in two different sub-variants: (4./5. and 4’./5’.). Text: Number: R 82 S 83 A 65 32 w 119 o 111 r 114 k 107 s 115 ! 33

4. Encryption: Blocks are formed151 (each ASCII character is encoded into a 8 digit binary number below): 21075 16672 30575 29291 29473152 Each block is encrypted using: C ≡ M 65,537 (mod 256, 027)153 : 158721 137346 37358 240130 112898 5. Decryption: Cipher text: 158721 137346 37358 240130 112898 Each block is decrypted using: M ≡ C 231,953 (mod 256, 027): 21075 16672 30575 29291 29473 4’. Encryption: Blocks are formed: (each ASCII character is encoded into a 3 digit decimal number below):

150 151

See Appendix E of this chapter for the source code for RSA exponentiation using Mathematica and Pari-GP.

binary representation decimal representation 01010010, 82 01010010 01010011 =21075 01010011, 83 01000001, 65 01000001 00100000 =16672 00100000, 32 01110111, 119 01110111 01101111 =30575 01101111, 111 01110010, 114 01110010 01101011 =29291 01101011, 107 01110011, 115 01110011 00100001 =29473 00100001, 33: 152 Using CrypTool you can solve this with the menu Indiv.Procedures \ RSA Cryptosystem \ RSA Demonstration with the following options: all 256 ASCII characters, b-adic, block length 2 and decimal representation. 153 See Appendix E of this chapter for the source code for RSA exponentiation using Mathematica and Pari-GP.

82083 65032 119111 114107 115033154 Each block is encrypted using: C ≡ M 65,537 (mod 256, 027)155 : 198967 051405 254571 115318 014251 5’. Decryption: Cipher text: 198967 051405 254571 115318 014251 Each block is decrypted using: M ≡ C 2,473 (mod 67, 519): 82083 65032 119111 114107 115033 4.13.4 A small RSA cipher challenge (1)

The task is taken from [Stinson1995, Exercise 4.6]: The pure solution has been published by Prof. Stinson at http://www.cacr.math.uwaterloo.ca/~dstinson/solns.html.156 However, it is not the result that is important here but rather the individual steps of the solution, that is, the explanation of the cryptanalysis157 : Two samples of RSA cipher text are presented in Tables 4.1 and 4.2. Your task is to decrypt them. The public parameters of the system are n = 18, 923 and e = 1, 261 (for Table 4.1) and n = 31, 313 and e = 4, 913 (for Table 4.2). This can be accomplished as follows. First, factor n (which is easy because it is so small). Then compute the exponent d from J(n), and, finally, decrypt the cipher text. Use the square-andmultiply algorithm to exponentiate modulo n. In order to translate the plaintext back into ordinary English text, you need to know how alphabetic characters are “encoded” as elements in Zn . Each element of Zn represents three alphabetic characters as in the following examples: DOG CAT ZZZ → → → 3 ∗ 262 + 14 ∗ 26 + 6 = 2, 398 2 ∗ 262 + 0 ∗ 26 + 19 = 1, 371 25 ∗ 262 + 25 ∗ 26 + 25 = 17, 575.

You will have to invert this process as the final step in your program. The first plaintext was taken from “The Diary of Samuel Marchbanks”, by Robertson Davies, 1947, and the second was taken from “Lake Wobegon Days”, by Garrison Keillor, 1985.

The RSA encryption works correctly with the modulus n = 256.027 because each ASCII block of two characters will be encoded into a number that is smaller or equal than the number 255, 255. 155 See Appendix E of this chapter for the source code for RSA exponentiation using Mathematica and Pari-GP. 156 or http://bibd.unl/~stinson/solns.html. 157 The method of solving the problem is outlined in the scenario of the online help to CrypTool and in the presentation on the website. If anyone sends us a well prepared exact method of solving the problem, we would be pleased to include it in the documentation.

154

TABLE 4.1158 : RSA cipher text 12423 9792 5300 2264 12693 12161 13236 15061 2620
3533 3460 12867 12192 2430 7913 796 9792 56 4277 2364 16979 1367 2186 18676 2364
11748 9522 18628 2951 11524 13629 13951 961 9553 13071 5300 12347 6276 13842 9886
13203 56 9741 6246 195 14251 4118 10617 15570 15404 2512 9433 4782 6789 14616
14838 14326 722 7243 14407 81 17459 18194 16900 13951 7817 8500 7537 8687 5102
2471 11675 14301 9872 1498 11302 874 3460 6127 14407 13293 11374 11634 11453 7437
9175 15334 7459 18817 8986 4101 3830 7233 8850 7946 201 12259 4481 4742 15334 424
1144 16979 11296 5988 13211 9886 9872 5053 7555 446 4493 17666 3880 9061 841 14303
18830 8007 2999 2664 8270 12129 11675 8850 18110 11231 5053 841 6686 9056 15404
1105 3363 11821 9988 3652 1521 13618 4165 4063 925 11476 650 15610 6127 13556
13167 14569 13998 17086 6091 13924 11178 44 7547 15407 13995 738 15967 14130 4502
15827 3090 3798 14838 297 13000 11634 4576 56 8305 18110 2443 10964 3159 10022
17183 12501 9792 18110 13892 16477 2364 11383 2976 17592 13874 7328 9105 16979
6928 18110 1158 7437 10935 6490 3846 17955 4118 5102 8720 11056 16399 16647 17213
15827 18873 14266 3332 18031 10161 15570 17910 9330 13297 8168 13203 2001 1105
4191 44 9872 2540 17137 5310 14611 7965 18031 2999 15404 2186

158

The numbers of this table can be worked with via Copy and Paste.

TABLE 4.2159 : RSA cipher text 6340 23614 27584 25774 7908 4082 15698 1417 12437 23005 15930 27486 18154 2149 19554 3183 6000 25973 4.13.5 8309 7135 14999 7647 8635 11803 30317 26905 1108 8267 29748 9741 22319 16975 23614 17347 31280 4477 14010 24996 4517 23901 2149 5314 4685 25809 27106 9917 8635 2149 27705 16087 7553 25234 29413 30989 8936 30590 12146 7372 1908 107 14696 28347 18743 7994 23645 29329 20321 14600 4734 4595 2066 27358 27570 29421 25774 22076 7359 30388 26277 24144 9694 11738 2149 23254 27705 8091 21498 369 25023 26486 26439 18436 7372 22470 8671 7897 10685 2149 24591 5501 13624 19386 23973 6360 23204 16481 30388 1606 12056 8686 7372 29956 20240 25234 10042 20240 14015 3249 7325 14015 19837 8425 25809 9395 17881 13547 1304 22827 15705 21519 30155 27705 27212 30155 5443 26277 107 8463 7792

A small RSA cipher challenge (2)

The following task is a corrected version from the excellent book written by Prof. Yan [Yan2000, Example 3.3.7, p. 318]. However, it is not the result that is important here but rather the individual steps of the solution, that is, the explanation of the cryptanalysis160 . There are three tasks with completely different degrees of difficulty here. In each case we know the cipher text and the public key (e, n): (a) Known plaintext: find the secret key d using the additionally known original message. (b) Cipher text only: find d and the plaintext. (c) Calculate the RSA modulus, in other words factorisation (with no knowledge of the message).

159 160

The numbers of this table are in the online-help “Example illustrating the RSA demonstration” of CrypTool. The method of solving the problem is outlined in the scenario of the online help to CrypTool and in the CrypTool presentation. If anyone sends us a well prepared exact method of solving the problem, we would be pleased to include it in the documentation.

n = 63978486879527143858831415041, e = 17579 Message161 : 1401202118011200, 1421130205181900, 0118050013010405, 0002250007150400 Cipher: 45411667895024938209259253423, 16597091621432020076311552201, 46468979279750354732637631044, 32870167545903741339819671379 Comments: The original message consisted of a sentence containing 31 characters (coded with the capital letters alphabet from section 4.13.2). Each group of 16 decimal numbers is then combined to form one number (the last number is filled with zeros). These numbers are raised to the power of e. When you decrypt the message you must fill the calculated numbers with leading zeros in order to obtain plaintext. This needs to be stressed because the type of padding is extremely important during implementation and standardisation for interoperable algorithms.

161

The numbers of this table are in the online help “Example illustrating the RSA demonstration” of CrypTool.

References

[Agrawal2002] M. Agrawal, N. Kayal, N. Saxena, PRIMES in P, August 2002 http://www.cse.iitk.ac.in/news/primality.html [Bartholome1996] A. Bartholome, J. Rung, H. Kern, Zahlentheorie f¨r Einsteiger, Vieweg 1995, 2nd edition 1996. u [Bauer1995] Friedrich L. Bauer, Entziﬀerte Geheimnisse, Springer, 1995. [Bauer2000] Friedrich L. Bauer, Decrypted Secrets, Springer 1997, 2nd edition 2000. [Bernstein2001] D. J. Bernstein, Circuits for integer factorization: a proposal, http://cr.yp.to/papers/nfscircuit.ps http://cr.yp.to/djb.html [Beutelspacher1996] Albrecht Beutelspacher, Kryptologie, Vieweg 1987, 5th edition 1996. [Bourseau2002] F. Bourseau, D. Fox, C. Thiel, Vorz¨ge und Grenzen des RSA-Verfahrens, u In: Datenschutz und Datensicherheit (DuD) 26/2002, pp 84-89 (see www.dud.de), http://www.secorvo.de/publikationen/rsa-grenzen-fox-2002.pdf [Brands2002] Gilbert Brands, Verschl¨sselungsalgorithmen – Angewandte Zahlentheorie rund um Sicherheitsprotokolle, u Vieweg, 2002. [Buchmann1999] Johannes Buchmann, Einf¨hrung in die Kryptographie, Springer, 1999. u [Buhler1993] J.P. Buhler, H.W. Lenstra, C. Pomerance, Factoring integers with the number field sieve, In: A.K. Lenstra, H.W. Lenstra (Hrsg.): The Development of the Number Field Sieve, Lecture Notes in Mathematics, Vol. 1554, Springer, Heidelberg 1993, pp 50–94. [Eckert2003] Claudia Eckert, IT-Sicherheit: Konzepte-Verfahren-Protokolle, Oldenbourg 2001, 2nd edition 2003. [Ertel2001] Wolfgang Ertel, Angewandte Kryptographie, Fachbuchverlag Leipzig FV 2001.

[GISA2002] GISA (German Information Security Agency), Recommendation for key length selection, Bonn, Sep. 2002, http://www.bsi.bund.de/esig/basics/techbas/krypto/bund02v7.pdf A statement on these recommendations: http://www.secorvo.de/publikat/ stellungnahme-algorithmenempfehlung-020307.pdf

[Graham1994] Graham, Knuth, Patashnik, Concrete Mathemathics, a Foundation of Computer Science, Addison Wesley 1989, 6th printing 1990. [Kippenhahn1997] Rudolph Kippenhahn, Verschlüsselte Botschaften – Geheimschrift, Enigma und Chipkarte, Rowohlt, 1997. [Kippenhahn1999] Rudolph Kippenhahn, Code Breaking – A History and Exploration, Constable, 1999. [Knuth1998] Donald E. Knuth, The Art of Computer Programming, vol 2: Seminumerical Algorithms, Addison-Wesley, 2nd edition 1998. [Lenstra1993] A. Lenstra, H. Lenstra: The development of the Number Field Sieve, Lecture Notes in Mathematics 1554, Springer, New York 1993 [Lenstra2002] Arjen K. Lenstra, Adi Shamir, Jim Tomlinson, Eran Tromer, Analysis of Bernstein's Factorization Circuit, http://www.cryptosavvy.com/mesh.pdf [Menezes2001] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone Handbook of Applied Cryptography, CRC Press 1997, 5th printing 2001. [Pfleeger1997] Charles P. Pfleeger, Security in Computing, Prentice-Hall, 2nd edition 1997. [Pomerance1984] C. Pomerance, The quadratic sieve factoring algorithm, In: G.R. Blakley, D. Chaum (Hrsg.): Proceedings of Crypto '84, LNCS 196, Springer Berlin 1995, pp 169-182. [RSA Security 2002] RSA Security, Has the RSA algorithm been compromised as a result of Bernstein's Paper?, April 8th, 2002, http://www.rsasecurity.com/

[SchneiderM2004] Matthias Schneider, Analyse der Sicherheit des RSA-Algorithmus. M¨gliche Angriffe, deren Einfluss auf sichere Implementierungen und ¨konomische Kono o sequenzen, Diploma thesis at the University of Siegen, Germany, 2004. [Schneier1996] Bruce Schneier, Applied Cryptography, Protocols, Algorithms, and Source Code in C, Wiley and Sons, 2nd edition 1996. [Schwenk2002] J¨rg Schwenk, o Sicherheit und Kryptographie im Internet, Vieweg 2002. [Sedgewick1990] Robert Sedgewick, Algorithms in C, Addison-Wesley, 1990. [Shamir2003] Adi Shamir, Eran Tromer, Factoring Large Numbers with the TWIRL Device, Januar 2003, http://www.wisdom.weizmann.ac.il/~tromer/. [Shamir2003a] Adi Shamir, Eran Tromer, On the Cost of Factoring RSA-1024, RSA Laboratories CryptoBytes Volume 6, No. 2, Summer 2003, p. 11-20 http://www.rsasecurity.com/rsalabs/cryptobytes/CryptoBytes August 2003.pdf [Silverman2000] Robert D. Silverman: A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths In: RSA Laboratories Bulletin, No. 13, April 2000, p. 1-22 [Stinson1995] Douglas R. Stinson, Cryptography - Theory and Practice, CRC Press, 1995. [Weis2003] R¨diger Weis, Stefan Lucks, Andreas Bogk, u Sicherheit von 1024 bit RSA-Schl¨sseln gef¨hrdet, u a In: Datenschutz und Datensicherheit (DuD) 6/2003, pp 360-362 (see www.dud.de) The article explains details about the TWIRL device. [Welschenbach2001] Welschenbach, Michael, Kryptographie in C und C++, Springer 2001. [Wiles1994] Wiles, Andrew, Modular elliptic curves and Fermat's Last Theorem, In: Annals of Mathematics 141 (1995). [Wolfenstetter1998] Albrecht Beutelspacher, J¨rg Schwenk, Klaus-Dieter Wolfenstetter, o Moderne Verfahren in der Kryptographie, Vieweg 1995, 2nd edition 1998. [Yan2000] Song Y. Yan, Number Theory for Computing, Springer, 2000.

Web links

1. Ron Knott’s Fibonacci page, Here, everything revolves around Fibonacci numbers. http://www.mcs.surrey.ac.uk/personal/R.Knott/Fibonacci/fib.html 2. CrypTool, E-Learning freeware to illustrate cryptography and cryptanalysis http://www.cryptool.de, http://www.cryptool.org, http://www.cryptool.com 3. Mathematica, Commercial mathematics package http://www.wolfram.com 4. LiDIA, Extensive library containing number-theory functions and the LC interpreter http://www.informatik.tu-darmstadt.de/TI/LiDIA 5. BC, Interpreter with number-theory functions http://www.maths.uq.edu.au/~krm/gnubc.html 6. Pari-GP, Excellent, fast, free interpreter with number theoretical functions http://www.parigp-home.de and http://www.parigp-home.com 7. Only after I had completed this article, did I come across the website of Mr. M¨nchenbach, u which interactively and didactically uses elementary number theory to provide a sophisticated description of the fundamental mathematical thought processes. It was created for a teaching project in the 11th grade of the technical grammar school (unfortunately only available in German): http://www.hydrargyrum.de/kryptographie 8. Once again only after finishing this I happened upon the web site of Mr. Wagner, who is responsible for the development of the curriculum of computer science in one of the German federal states (L¨nder). Here you can get hold of a collection of texts and (Java-) programs a (available only in German): http://www.hom.saar.de/~awa/kryptolo.htm 9. GISA, German Information Security Agency http://www.bsi.bund.de 10. Factorisation records and challenges, http://www.crypto-world.com/

http://www.crypto-world.com/FactorWorld.html, page by Scott Contini
http://www.loria.fr/~zimmerma/records/factor.html http://www.tutorgig.com/ed/RSA
number http://www.uni-bonn.de/Aktuelles/Pressemitteilungen/pm02/pm035-02.html
http://www.ercim.org/publication/Ercim News/enw49/franke.html, 2002-01
http://www.loria.fr/~zimmerma/records/rsa160
http://www.rsasecurity.com/rsalabs/challenges/factoring/numbers.html 11. The Cunningham Project, http://www.cerias.purdue.edu/homes/ssw/cun/

Acknowledgments

I would like to take this opportunity to thank • Mr. Henrik Koy for making many very useful suggestions, for the very constructive proofreading this article and for helping with TeX. Mr. Koy designed and developed in his leisure time the functions and the complex dialog box of the RSA cryptosystem, which enables you to execute the RSA samples of this article. • J¨rg Cornelius Schneider for his TeX support und for the many cases where he helped when o facing programming or design problems. • Dr. Georg Illies for pointing me to Pari-GP .

Appendix A: the greatest common divisor (gcd) of whole numbers and the two Algorithms of Euclid

1. The greatest common divisor of two natural numbers a and b is an important value that can be calculated very quickly. Here we make use of the fact that if a number c divides the numbers a and b (i.e. there exists an a and a b such that a = a ∗ c and b = b ∗ c), then c also divides the remainder r of a/b. In short notion we can write: If c divides a and b it follows that c divides r = a − a/b ∗ b162 . As the latter statement is valid for each common divisor c of a and b it follows that: gcd(a, b) = gcd(a − a/b ∗ b, b). Using this information, the algorithm for calculating the gcd of two numbers can be written as follows (in pseudo code): INPUT: a,b != 0 1. if (a < b) then x = a; a = b; b = x; // Swap a and b (a > b) 2. a = a - int(a/b) * b // a is smaller than b, the // gcd(a, b) is unchanged 3. if (a != 0) then goto 1. // a falls after each step and // the algorithm ends when a==0. OUTPUT "gcd(a,b) = " b // b is the gcd of the original a and b 2. However, to other relationships can be derived from the gcd: For this, we need the set of equations for a and b: a = 1∗a+0∗b b = 0 ∗ a + 1 ∗ b, or, in matrix notation: a b = 10 01 ∗ a b .

We summarise this information in the extended matrix: a|10 b|01 If we apply the above gcd algorithm to this matrix, we obtain the extended gcd algorithm:

162

The Gauss bracket x of a real number x is defined via: x is the next integer less or equal x.

INPUT: a, b = 0 0. x1,1 := 1, x1,2 := 0, x2,1 := 0, x2,2 := 1 1. a | x1,1 x1,2 b | x2,1 x2,2 := 0 1 1 − a/b ∗ b ∗ a | x1,1 x1,2 b | x2,1 x2,2 .

2. if (b != 0) then goto 1. OUTPUT: “gcd(a, b) = a ∗ x + b ∗ y: ”, “gcd(a, b) = ” b, “x =” x2,1 , “y =” x2,2 Since this algorithm only performs linear transformations, the same equations always apply a = x1,1 ∗ a + x1,2 ∗ b b = x2,1 ∗ a + x2,2 ∗ b, and we have the extended gcd equation at the end of the algorithm163 : gcd(a, b) = a ∗ x2,1 + b ∗ x2,2 . Example: Using the extended gcd we can determine for e = 37 the multiplicative inverse number d to modulo 3588 (i.e. 37 ∗ d ≡ 1 (mod 3588)): 0. 1. 2. 3. 3588 | 1 0 37 | 0 1 37 | 1 0 36 | 0 −96 36 | 1 −96 1 | −1 97 = = 0 1 1 −(3588/36 = 96) ∗ 37 0 1 1 −(37/36 = 1) ∗ 36 ∗ ∗ 3588 | 1 0 37 | 0 1 37 | 1 0 36 | 0 −96 . . .

1 | −1 97 0 1 36 | 1 −96 = ∗ 0 | 37 −3588 1 −(36/1 = 36) ∗ 1 1 | −1 97 OUTPUT: gcd(37, 3588) = a ∗ x + b ∗ y: gcd(37, 3588) = 1, x = −1, y = 97. Thus (a) 37 and 3588 are relatively prime (37 has an inverse modulo 3588).

(b) 37 ∗ 97 = (1 ∗ 3588) + 1 in other words 37 ∗ 97 ≡ 1 (mod 3588). and therefore the number 97 is the multiplicative inverse to 37 modulo 3588.

---

[163] By termination of the gcd algorithm, the program variables a and b contain the values a = 0 and b = gcd(a, b). Please keep in mind, that the program variables are different to the numbers a and b and that they are only relevant for the scope of the algorithm.

Appendix B: Forming closed sets

The property of closeness is always defined in relation to an operation in a set. The following shows how to construct the “closed set” G with respect to the operation + (mod 8) for a given initial set G0 : G0 = {2, 3} addition of the numbers in G0 determines further numbers : 2 + 3 ≡ 5 (mod 8) = 5 2 + 2 ≡ 4 (mod 8) = 4 3 + 3 ≡ 6 (mod 8) = 6 G1 = {2, 3, 4, 5, 6} addition of the numbers in G1 determines : 3 + 4 ≡ 7 (mod 8) = 7 3 + 5 ≡ 8 (mod 8) = 0 3 + 6 ≡ 9 (mod 8) = 1 G2 = {0, 1, 2, 3, 4, 5, 6, 7} addition of the numbers in G2 does not extend the set! G3 = G2 we say : G2 is closed for addition (mod 8). End of forming a closed set.

Appendix C: Comments on modulo subtraction

Comment on subtraction modulo 5: 2 − 4 ≡ −2 ≡ 3 mod 2. It is therefore not true modulo 5 that −2 = 2 ! People often make the mistake of equating this. You can show this clearly if you place the permutation (0, 1, 2, 3, 4) in Z5 , for example from −11 to +11, over the range of numbers in Z.

range of numbers modulo 5 40123401234012340123401 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11 range of numbers in Z

-11-10 -9 -8 -7 -6

Appendix D: Base representation of numbers, estimation of length of digits

For a given number z one may ask how to represent such a number. In general we use represen√ tations like z = 2374 or z = 2. The second number consists of an infinite number of digits and therefore it can never be described precisely by the first representation. In this case the number must be rounded. We represent numbers usually in the decimal system (base 10). Computers are working with the binary representation of numbers – only for the display numbers are represented in decimal or sometimes hexadecimal (base 16) form. This appendix describes how to generate arbitrary base representations of anay positive integer and how to determine the number of required digits via the logarithm function. b-adic sum representation of positive integers Given base b, each positive integer z can be represented as a b-adic sum z = an bn + an−1 bn−1 + · · · + a1 b + a0 , where ai ∈ {0, 1, . . . , b − 1}, i = 0, 1, . . . , n are called digits. For this sum, it follows that for arbitrary digits a0 , a1 , . . . , an we have bn+1 > an bn + an−1 bn−1 + · · · + a1 b + a0 . On the other hand there exist a0 , a1 , . . . , an (namely ai = b − 1 for i = 0, . . . , n) following that bn+1 − 1 ≤ an bn + an−1 bn−1 + · · · + a1 b + a0 (using these inequalities it can be shown that each positve integer can be represented by a b-adic sum). By writing the digits an an−1 · · · a1 a0 in a row directly after each other (without the bi) the usual writing for numbers comes to hand. Examples: Base b = 10: 10278 = 1 · 104 + 0 · 103 + 2 · 102 + 7 · 101 + 8 and Base b = 16: F E70A = 15 · 164 + 14 · 163 + 7 · 162 + 0 · 161 + 10. Number of digits to represent a positive integer For a positive integer z the length of the b-adic representation can be determined via the following steps. Starting from the inequality bn+1 > z ≥ bn we have – after applying the logarithm function on basis b164 : n + 1 > logb z ≥ n. Therefore we have n = logb z 165 . We call lb (z) the number of required digits to represent the number z on the base b. We have lb (z) := logb z + 1.

164

Applying the logarithm formula on base b and b we have logb z = logb z/ logb (b). It is therefore easy using e.g. logarithm tables for the base b = 10 to compute the logarithm of base b = 2. 165 The function x determines the next integer smaller than x (in case x ≥ 0 the digits after the decimal point are truncated).

Example 1: We compute for the decimal number z = 234 (EA in hex) the hexadecimal representation (number base b = 16) l16 (z) = log16 (z) + 1 = ln(z)/ ln(16) + 1 = 1.96... + 1 = 1 + 1 = 2. Example 2: We compute for the decimal number z = 234 (11101010 in binary) the binary representation (number base b = 2) l2 (z) = log2 (z) + 1 = ln(z)/ ln(2) + 1 = 7.87... + 1 = 7 + 1 = 8.

Algorithm to compute the base representation Given the number z one can compute the base b representation of z using the following algorithm input: z, b n := 0, z := z while z > 0 do an := z mod b, z := z /b n := n + 1 end do output: an an−1 · · · a1 a0 in base b representation. Example 1: The integer z = 234 on the number base 10 will be transformed into the hex representation via a0 = 234 mod 16 = 10 = A, 234/16 = 15 = E, a1 = 14 mod 16 = E and therefore we have EA. Example 2: The binary number z = 1000100101110101 is transformed into the decimal representation via the following steps: 1000100101110101 = 1001 (mod 1010) =⇒ a0 = 9, 1000100101110101/1010 = 110110111110 110110111110 = 1000 (mod 1010) =⇒ a1 = 8, 110110111110/1010 = 101011111 101011111 = 1 (mod 1010) =⇒ a2 = 1, 10101111/1010 = 100011 100011 = 101 (mod 1010) =⇒ a3 = 5, 100011/1010 = 1 11 = 11 (mod 1010) =⇒ a4 = 3 therefore z = 35189.

Appendix E: Examples using Mathematica and Pari-GP

This appendix gives you the source code to compute the tables and examples using Mathematica or the free software Pari-GP. Multiplication table modulus m The multiplication tables modulo m = 17 for a = 5 and a = 6 on page 89 can be computed in Mathematica with the following commands: m = 17; iWidth = 18; iFactor1 = 5; iFactor2 = 6; Print[’’i ’’, Table[i, {i, 1, iWidth}]]; Print[iFactor1, ’’*i ’’, Table[iFactor1*i, {i, 1, iWidth }]]; Print[’’Remainder ’’, Table[Mod[iFaktor1*i, m], {i, 1, iWidth }]]; Print[iFactor2, ’’*i ’’, Table[iFactor2*i, {i, 1, iWidth }]]; Print[’’Remainder ’’, Table[Mod[iFactor2*i, m], {i, 1, iWidth }]]; Pari-GP computes the tables via: m=17; iWidth=18; iFactor1=5; iFactor2=6; matrix(1,iWidth, x,y, iFactor1*y) yields [5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90] matrix(1,iWidth, x,y, (iFactor1*y)%m) yields [5 10 15 3 8 13 1 6 11 16 4 9 14 2 7 12 0 5] Note: Pari-GP generates when using the Mod function compound Mod objects, which are displayed as shown below: matrix(1,iWidth, x,y, Mod(iFactor1*y, m)) [Mod(5, 17) Mod(10, 17) Mod(15, 17) Mod(3, 17) Mod(8, 17) Mod(13, 17) Mod(1, 17) Mod(6, 17) Mod(11, 17) Mod(16, 17) Mod(4, 17) Mod(9, 17) Mod(14, 17) Mod(2, 17) Mod(7, 17) Mod(12, 17) Mod(0, 17) Mod(5, 17)] From a Mod object you can get back the components with the component or lift function: component(Mod(5, 17),1) component(Mod(5, 17),2) component(Mod(17,5), 1) component(Mod(17,5), 2) lift(Mod(17,5)) → 2 → → → → 17 5 5 2

The other multiplication table examples modulo 13 and modulo 12 on page 89 can computed by replacing m=17 with m=13 and m=12 respectively.

Fast exponentiation The fast exponentiation modulo m belongs to the built in functions of Mathematica and Pari-GP. Using those programs you can comprehend the idea of the square and multiply method. With Mathematica you can compute the exponentiations of the example on page 92 as follows: Mod[{87^43, 87^2, 87^4, 87^8, 87^16, 87^32}, 103] = {85, 50, 28, 63, 55, 38}. and in Pari-GP the syntax is: Mod([87^43,87^2,87^4,87^8,87^16,87^32],103) Multiplicative order and primitive roots

∗ The order ordm (a) of a number a in the multiplicative group Zm is the smallest number i ≥ 1, i ≡ 1 mod m holds. For the example on page 100 you can make Mathematica print all for with a exponentiations ai mod 11 using the following syntax:

```
m=11;
```

```
Table[ Mod[a^i, m], {a, 1, m-1}, {i, 1, m-1} ]
```

Equivalent Pari-GP syntax: m=11; matrix(10,10, x,y, (x^y)%m) The table on page 101 gives examples for the order modulo 45 ord45 (a) and the Euler number J(45). Mathematica can be used to create this table with the following program (please note that Print cannot be used inside of Do-loops and each Print outputs a newline). m= Do[’’, ’’, {a, 45; Print[Table[Mod[a^i, m], {i, 1, 12}], ’’, MultiplicativeOrder[a, m, 1], ’’, EulerPhi[m]], 1, 12}];

Here is the corresponding Pari-GP syntax: m=45; matrix(12,14, x,y, if(y<=12, (x^y)%m, if(y==13, if(gcd(x,m)==1, znorder(Mod(x,m)), "--"), eulerphi(m)))) znorder(Mod(x,m)) can only be calculated if x is relatively prime to m, which can be checked with gcd(x,m).

Performance can be improved by using Mod(x,m)∧ y instead of (x∧ y)%m. Loops are also supported by Pari-GP. When you remove the table formatting the result looks like this: for(x=1,12, for(y=1,12, print(Mod(x^y,m))); if(gcd(x,m)==1, print(znorder(Mod(x,m))), print("--")); print(eulerphi(m))) The third example on page 102 displays exponentiations ai mod 46 as well as the order ord46 (a). Mathematica can create this table with the following loop: m= Do[’’, {a, 46; Print[Table[Mod[a^i, m], {i, 1, 23}], ’’, MultiplicativeOrder[a, m, 1] 1, 23}]];

In Pari-GP the syntax looks like this: m=46; matrix(23,24, x,y, if(y<=23, (x^y)%m, if(y==24, if(gcd(x,m)==1, znorder(Mod(x,m)), "--")))) RSA examples This section list the source code of the RSA examples in section 4.13 (“The RSA procedure with actual numbers”) using Mathematica and Pari-GP syntax. Example on page 124. The RSA exponentiation M 37 mod 3713 on message M = 120 can be calculated in Mathematica like this: PowerMod[120, 37, 3713]. Here is the corresponding Pari-GP syntax: Mod(120,3713)^37 or Mod(120^37,3713). Example on page 125. The factorisation of J(256, 027) = 255, 016 = 23 ∗ 127 ∗ 251 can be calculated with Mathematica like this: FactorInteger[255016]= {{2,3}, {127,1}, {251,1}}. Pari-GP does the same with: factor(255016). Example on page 125. Mathematica can do RSA encryption with the command:

PowerMod[{82, 83, 65, 32, 119, 111, 114, 107, 115, 33}, 65537, 256027]} Pari-GP needs the following syntax: vecextract([Mod(82,256027)^65537, Mod(83,256027)^65537, Mod(65,256027)^65537, Mod(32,256027)^65537, Mod(119,256027)^65537, ...]) Remarks on using Mod in Pari-GP: Mod(82,256027)^65537 is much faster than – Mod(82^65537, 256027) and – (82^65537) % 256027. Example on page 126. Mathematica can do RSA encryption with the following command: PowerMod[{21075, 16672, 30575, 29291, 29473}, 65537, 256027] The same calculation with Pari-GP: vecextract([Mod(21075,256027)^65537, Mod(16672,256027)^65537, Mod(30575,256027)^65537, Mod(29291,256027)^65537, Mod(29473,256027)^65537], 31) Example on page 127. RSA encryption using Mathematica: PowerMod[{82083, 65032, 119111, 114107, 115033}, 65537, 256027] RSA encryption with Pari-GP: vecextract([Mod(82083,256027)^65537, Mod(65032,256027)^65537, Mod(119111,256027)^65537, Mod(114107,256027)^65537, Mod(115033,256027)^65537], 31)

Appendix F: List of the formulated definitions and theorems

Short description prime numbers composite numbers factors of composite numbers 1. fundamental theorem of number theory divisibility remainder class r modulo m congruent congruence with difference multiplicative inverse exhaustive permutation power mod m Zn Z∗ n multiplicative inverse in Z∗ n Euler function J(n) J(p) J(p ∗ q) J(p1 ∗ · · · ∗ pk) J(pe1 ∗ · · · ∗ pek) 1 k little Fermat Euler-Fermat theorem multiplicative order ordm (a) primitive root of m exhausting of all possible values Page 80 80 81 81 82 82 83 83 88 89 91 93 94 95 96 96 96 96 96 97 97 100 100 102

Definition 4.1 Definition 4.2 Theorem 4.1 Theorem 4.2 Definition 4.3 Definition 4.4 Definition 4.5 Theorem 4.3 Theorem 4.4 Theorem 4.5 Theorem 4.6 Definition 4.6 Definition 4.7 Theorem 4.7 Definition 4.8 Theorem 4.8 Theorem 4.9 Theorem 4.10 Theorem 4.11 Theorem 4.12 Theorem 4.13 Definition 4.9 Definition 4.10 Theorem 4.14

5

The Mathematical Ideas behind Modern Cryptography

(Oyono R. / Esslinger B., Sep. 2000, Updates Nov. 2000, Feb. 2003)

5.1

One way functions with trapdoor and complexity classes

A one way function is a function that can be calculated efficiently, but whose inverse is extremely complicated and practically impossible to calculate. To put it more precisely: A one way function is a mapping f from a set X to a set Y, such that f (x) can be calculated easily for each element x of X, whereas for (almost) every y from Y it is practically impossible to find an inverse image x (i.e. an x where f (x) = y). An everyday example of a one way function is a telephone book: the function to be performed is to assign a name to the corresponding telephone number. This can be done easily due to the fact that the names are sorted alphabetically. However, the inverse function - assigning a name to a given number - is obviously difficult if you only have a telephone book available. One way functions play a decisive role in cryptography. Almost all cryptographic terms can be rephrased using the term one way function. Let's take for example public key encryption (asymmetric cryptography): Each subscriber T to the system is assigned a private key dT and what is known as a public key eT . These keys must have the following property (public key property): For an opponent who knows the public key eT , it is practically impossible to determine the private key dT . In order to construct useful public key procedures, therefore, we look for a one way function that is "easy" to calculate in one direction , but is "difficult" (practically impossible) to calculate in the other direction, provided that a particular piece of additional information (trapdoor) is not available. This additional piece of information allows the inverse to be found efficiently. Such functions are called trapdoor one way functions. In the above case, dT is the trapdoor information. In this process, we describe a problem as "easy" if it can be solved in polynomial time as a function of the length of the input. If the length of the input is n bits, then the time for calculating the function is proportional to na , where a is a constant. We say that the complexity of such problems is O(na) [Landau- or Big-O notation]. If you compare two functions 2n and na , where a is a constant, then there always exists a value for n, from which for all further n applies: na < 2n . The function na has a lower complexity. Sample: for a = 5 the following applies: from the length n = 23, 2n is greater than n5 ; for further n 2n clearly increases more quickly [(222 = 4, 194, 304, 225 = 5, 153, 632), (223 = 8, 388, 608, 235 = 6, 436, 343), (224 = 16, 777, 216, 245 = 7, 962, 624)]. The term "practically impossible" is slightly less precise. In general, we can say that a problem cannot be solved efficiently, if the time required to solve it increases more quickly than the polynomial time as a function of the size of the input. If, for example, the length of the input is n bits and the time required for calculating the function is proportional to 2n , then the following currently applies: the function practically cannot be calculated for n > 80. In order to develop a public key procedure that can be implemented in practice, it is therefore necessary to discover a suitable trapdoor one way function. In order to tidy things up among this confusing multitude of possible problems and their complexities, we group problems with similar complexities into classes. The most important complexity classes are the classes P and NP: • The class P: This class contains those problems that can be solved in a polynomial amount of time. • The class NP: The definition of this class doesn’t look at the time required to solve a problem, but rather at the time required to verify a given solution. The class NP consists of those problems for which a given solution can be verified in a polynomial amount of time. Hereby, the term NP “non-deterministic” means polynomial and is based on a calculation model, i.e. on a computer that only exists in theory and can “guess” correct solutions non-deterministically then verify them in polynomial time. The class P is contained in the class NP. A well-known unsolved problem is the question whether or not P = NP is true, i.e. whether or not P is a true subset. An important property of the class NP is that it also contains what are known as “NP-complete” problems. These are problems that represent the class NP as follows: If a “good” algorithm for such a problem exists, then “good” algorithms exist for all problems from NP. In particular: if P only contained one complete problem, i.e. if a polynomial solution algorithm existed for this problem, then Pwould be equal to NP. In this sense, the NP-complete problems are the most difficult problems in NP. Many cryptographic protocols are formed in such a way that the “good” subscribers only have to solve problems from P, whereas a perpetrator is faced with problems from NP. Unfortunately, we do not yet know whether one way functions actually exist. However, we can prove that one way functions exist if and only if P = NP [Balcazar1988, S.63]. Mathematicians have again and again claimed to have proven the equivalence, e.g. http://www.geocities.com/st busygin/clipat.html), but so far the claims have always turned out to be false. A number of algorithms have been suggested for public key procedures. In many cases - although they at first appeared promising - it was discovered that they could be solved polynomially. The most famous failed applicant is the knapsack with trapdoor, suggested by Ralph Merkle [Merkle1978].

5.2 Knapsack problem as a basis for public key procedures

5.2.1 Knapsack problem

You are given n objects G1 , . . . , Gn with the weights g1 , . . . gn and the values w1 , · · · , wn . The aim is to carry away as much as possible in terms of value while restricted to an upper weight limit g. You therefore need to find a subset of {G1 , · · · , Gn }, i.e. {Gi1 , . . . , Gik }, so that wi1 + · · · + wik is maximised under the condition gi1 + · · · + gik ≤ g. Such questions are called NP-complete problems (not deterministically polynomial) that are difficult to calculate. A special case of the knapsack problem is: Given the natural numbers a1 , . . . , an and g., find x1 , . . . , xn ∈ {0, 1} where g = n xi ai (i.e. i=1 where gi = ai = wi is selected). This problem is also called a 0-1 knapsack problem and is identified with K(a1 , . . . , an ; g). Two 0-1 knapsack problems K(a1 , . . . , an ; g) and K(a1 , . . . , an ; g) are called congruent if two co-prime numbers w and m exist in such a way that 1. m > max{
n i=1 ai , n i=1 ai },

2. g ≡ wg mod m, 3. ai ≡ wai mod m for all i = 1, . . . , n. Comment: Congruent 0-1 knapsack problems have the same solutions. No quick algorithm is known for clarifying the question as to whether two 0-1 knapsack problems are congruent. A 0-1 knapsack problem can be solved by testing the 2n possibilities for x1 , . . . , xn . The best method requires O(2n/2) operations, which for n = 100 with 2100 ≈ 1.27·1030 and 2n/2 ≈ 1.13·1015 represents an insurmountable hurdle for computers. However, for special a1 , . . . , an the solution is quite easy to find, e.g. for ai = 2i−1 . The binary representation of g immediately delivers x1 , . . . , xn . In general, the a 0-1 knapsack problem can be solved easily if a permutation[166] j−1 π of 1, . . . , n exists with aπ(j) > i=1 aπ(i) . If, in addition, π is the identity, i.e. π(i) = i for i = 1, 2, . . . , n, then the sequence a1 , . . . , an is said to be super-increasing. The following algorithm solves the knapsack problem with a super-increasing sequence in the timeframe of O(n).

---

[166] A permutation π of the numbers 1, . . . , n is a change in the order in which these numbers are listed. For example, a permutation π of (1, 2, 3) is (3, 1, 2), i.e. π(1) = 3, π(2) = 1 and π(3) = 2.

```
for i = n to 1 do
    if $T \geq a_i$ then
        $T := T - s_i$
        $x_i := 1$
    else
        $x_i := 0$
if $T = 0$ then
    $X := (x_1, \ldots, x_n)$ is the solution.
else
    No solution exists.
```

Algorithm 1. Solving knapsack problems with super-increasing weights

5.2.2

Merkle-Hellman knapsack encryption

In 1978, Merkle and Hellman [Merkle1978] specified a public key encryption procedure that is based on “defamiliarising” the easy 0-1 knapsack problem with a super-increasing sequence into a congruent one with a super-increasing sequence. It is a block ciphering that ciphers an n-bit plaintext each time it runs. More precisely:

Let (a1 , . . . , an) be super-increasing. Let m and w be two co-prime numbers with m > n ai and 1 ≤ w ≤ m − 1. Select w with ww ≡ 1 Ø Ø i=1 mod m the modular inverse of w and set bi := wai mod m, 0 ≤ bi < m for i = 1, . . . , n, and verify whether the sequence b1 , . . . bn is not superincreasing. A permutation bπ(1) , . . . , bπ(n) of b1 , . . . , bn is then published and the inverse permutation μ to π is defined secretly. A sender writes (j) (j) his/her message in blocks (x1 , . . . , xn) of binary numbers n in length, calculates
n

g (j) :=
i=1

xi bπ(i)

(j)

and sends g (j) , (j = 1, 2, . . .). The owner of the key calculates G(j) := wg
(j) Ø
(j)

mod m,

0 ≤ G(j) < m
(j)

and obtains the xμ(i) ∈ {0, 1} (and thus also the xi) from
n n

G(j) ≡ wg (j) = Ø
i=1 n

xi bπ(i) w ≡ Ø
i=1 n

(j)

xi aπ(i) mod m,

(j)

mod m

=
i=1

xμ(i) aπ(μ(i)) =
i=1

(j)

xμ(i) ai

(j)

by solving the easier 0-1 knapsack problems K(a1 , . . . , an ; G(j)) with super-increasing sequence a1 , . . . , an .

Merkle-Hellman procedure (based on knapsack problems).

In 1982, Shamir [Shamir1982] specified an algorithm for breaking the system in polynomial time without solving the general knapsack problem. Len Adleman [Adleman1982] and Jeff Lagarias [Lagarias1983] specified an algorithm for breaking the twice iterated Merkle-Hellman knapsack encryption procedure in polynomial time. Ernst Brickell [Brickell1985] then specified an algorithm for breaking multiply iterated Merkle-Hellman knapsack encryption procedures in polynomial time. This made this procedure unsuitable as an encryption procedure. It therefore delivers a one way function whose trapdoor information (defamiliarisation of the 0-1 knapsack problem) could be discovered by an evesdropper.

5.3
5.3.1

Decomposition into prime factors as a basis for public key procedures

The RSA procedure167

As early as 1978, R. Rivest, A. Shamir, L. Adleman [RSA1978] introduced the most important asymmetric cryptography procedure to date. Key generation: Let p and q be two different prime numbers and N = pq. Let e be any prime number relative to φ(N) , i.e. gcd(e, φ(N)) = 1. Using the Euclidean algorithm, we calculate the natural number d < φ(N), such that ed ≡ 1 mod φ(N). whereby φ is the Euler phi Function. The output text is divided into blocks and encrypted, whereby each block has a binary value x(j) ≤ N . Public key: N, e. Private key: d. Encryption: y = eT (x) = xe Decryption: dT (y) = y d mod N. mod N.

RSA procedure (based on the factorisation problem).

Comment: The Euler phi function is defined as: φ(N) is the number of natural numbers that do not have a common factor with N x ≤ N. Two natural numbers a and b are co-prime if gcd(a, b) = 1. For the Euler phi function: φ(1) = 1, φ(2) = 1, φ(3) = 2, φ(4) = 2, φ(6) = 2, φ(10) = 4, φ(15) = 8. For example, φ(24) = 8, because |{x < 24 : gcd(x, 24) = 1}| = |{1, 5, 7, 11, 13, 17, 19, 23}|. If p is a prime number, then φ(p) = p − 1. If we know the various prime factors p1 , . . . , pk of N, then φ(N) = N · (1 −
1 p1)

· · · (1 −

1 168 . pk)

167

Using CrypTool you can gain practical experience with the RSA procedure via the menu Indiv.Procedures \ RSA Cryptosystem \ RSA Demonstration. 168 Further formulas for the Euler phi function are in the article “Introduction to Elementary Number Theory with Examples”, chapter 4.8.1.

In the case of $N = pq$, $\varphi(N) = pq(1 - 1/p)(1 - 1/q) = p(1 - 1/p)q(1 - 1/q) = (p - 1)(q - 1)$.

n 1 2 3 4 5 6 7 8 9 10 15

φ(n) 1 1 2 2 4 2 6 4 6 4 8

The natural 1 1 1, 2 1, 3 1, 2, 3, 4 1, 5 1, 2, 3, 4, 5, 1, 3, 5, 7 1, 2, 4, 5, 7, 1, 3, 7, 9 1, 2, 4, 7, 8,

numbers that are co-prime to n and less than n.

6 8 11, 13, 14

The function eT is a one way function whose trapdoor information is the decomposition into primes of N . At the moment, no algorithm is known that can factorise two prime numbers sufficiently quickly for extremely large values (e.g. for several hundred decimal places). The quickest algorithms known today [Stinson1995] factorise a compound whole number N in a time period proportional √ to L(N) = e ln(N) ln(ln(N)) . N L(N) 1050 1.42 · 1010 10100 2.34 · 1015 10150 3.26 · 1019 10200 1.20 · 1023 10250 1.86 · 1026 10300 1.53 · 1029

To this date, it has not been proved that the problem of breaking RSA is equivalent to the factorisation problem. Nevertheless, it is clear that the RSA procedure will no longer be safe if the factorisation problem is “solved”.169
5.3.2 Rabin public key procedure (1979)

In this case it has been shown that the procedure is equivalent to breaking the factorisation problem. Unfortunately, this procedure is susceptible to chosen-cipher text attacks.

169

In 2000 the authors assumed that values of the order magnitude 100 to 200 decimal places are currently safe. They estimates that the current computer technology indicates that a number with 100 decimal places could be factorised in approximately two weeks at justifiable costs, and using an expensive configuration (e.g. of around 10 million US dollars), a number with 150 decimal places could be factorised in about a year, and a 200–digit number should remain impossible to factorise for a long time to come, unless there is a mathematical breakthrough. However, you can never be sure that there won’t be a mathematical breakthrough tomorrow. How easy it is to guess the future wrong is shown by the Factorization of RSA-200 (see chapter 4.11.4) – completely without a “mathematical breakthrough”.

Let p and q be two different prime numbers with p, q ≡ 3 mod 4 and n = pq. Let 0 ≤ B ≤ n − 1. Public key: e = (n, B). Private key: d = (p, q). Encryption: y = eT (x) = x(x + B) Decryption: dT (y) = y + B 2 /4 − B/2 mod n. mod n.

Rabin procedure (based on the factorisation problem). Caution: Because p, q ≡ 3 mod 4 the encryption is easy to calculate (if the key is known). This is not the case for p ≡ 1 mod 4. In addition, the encryption function is not injective: There are precisely four different source codes that have eT (x) as inverse image: x, −x − B, ω(x + B/2) − B/2, −ω(x + B/2) − B/2, where ω is one of the four roots of unity. The source codes therefore must be redundant for the encryption to remain unique! Backdoor information is the decomposition into prime numbers of n = pq.

5.4

The discrete logarithm as a basis for public key procedures

Discrete logarithms form the basis for a large number of algorithms for public-key procedures. 5.4.1 The discrete logarithm in Zp

Let p be a prime number and let g ∈ Z∗ = {0, 1, . . . , p − 1}. Then the discrete exponential p function base g is defined as eg : k −→ y := g k mod p, 1 ≤ k ≤ p − 1.

The inverse function is called a discrete logarithm function logg ; the following holds: logg (g k) = k. The problem of the discrete logarithm (in Z∗) is understood to be as follows: p Given p, g and y, determine k such that y = g k mod p.

It is much more difficult to calculate the discrete logarithm than to evaluate the discrete exponential function (see chapter 4.9). There are several procedures for calculating the discrete logarithm [Stinson1995]:

Name Baby-Step-Giant-Step Silver-Pohlig-Hellman Index-Calculus

Complexity √ O(p) polynomial in q, the greatest prime factor of p − 1. √
O(e(1+o(1))
ln(p) ln(ln(p))

)

5.4.2

Diffie-Hellman key agreement170

The mechanisms and algorithms of classical cryptography only take effect when the subscribers have already exchanged the secret key. In classical cryptography you cannot avoid exchanging secrets without encrypting them. Transmission safety here must be achieved using noncryptographic methods. We say that we need a secret channel for exchanging secrets. This channel can be realised either physically or organisationally. What is revolutionary about modern cryptography is, amongst other things, that you no longer need secret channels: You can agree secret keys using non-secret, i.e. public channels. One protocol that solves this problem is that of Diffie and Hellman. Two subscribers A and B want to agree on a joint secret key. Let p be a prime number and g a natural number. These two numbers do not need to be secret. The two subscribers then select a secret number a and b from which they calculate the values $\alpha = g\ a \bmod p$ and $\beta = g\ b \bmod p$. They then exchange the numbers α and β. To end with, the two subscribers calculate the received value to the power of their secret value to get $\beta\ a \bmod p$ and $\alpha b \bmod p$. Thus $\beta\ a \equiv (g\ b\ )a \equiv g\ ba \equiv g\ ab \equiv (g\ a\ )b \equiv \alpha b \bmod p$

Diffie-Hellman key agreement.

The safety of the Diffie-Hellman protocol is closely connected to calculating the discrete logarithm mod p. It is even thought that these problems are equivalent.

5.4.3 ElGamal public key encryption procedure in $Z*\ p$

By varying the Diffie-Hellman key agreement protocol slightly, you can obtain an asymmetric encryption algorithm. This observation was made by Taher ElGamal.

170

With CrypTool this exchange protocol has been visualized: you can execute the single steps with concrete numbers using menu Indiv. Procedures \ Protocols \ Diffie-Hellman Demonstration.

Let p be a prime number such that the discrete logarithm in Zp is difficult to compute. Let $\alpha \in Z*$ be a primitive element. Let $a \in IN$ and $\beta = \alpha a$ p mod p. Public key: p, α, β. Private key: a. Let $k \in Zp-1$ be a random number and $x \in Z*$ the plaintext. p Encryption: $eT (x, k) = (y1 , y2)$, where $y1 = \alpha$ k and $y2 = x\beta$ k
Decryption:
a $dT (y1 , y2) = y2 (y1)-1$

mod p mod p. mod p

ElGamal procedure (based on the factorisation problem).

5.4.4

Generalised ElGamal public key encryption procedure

The discrete logarithm can be generalised in any number of finite groups $(G, \circ)$. The following provides several properties of G, that make the discrete logarithm problem difficult.

Calculating the discrete exponential function Let G be a group with the operation $\circ$ and $g \in G$. The (discrete) exponential function base g is defined as $eg : k \longrightarrow g$ k, where g $k := g \circ . . . \circ g$.
k times

for all $k \in N$.

The exponential function is easy to calculate:

Lemma. The power g k can be calculated in at most 2 $\log2$ k group operations.

Proof Let k = 2n +kn−1 2n−1 +· · ·+k1 2+k0 be the binary representation of k. Then n ≤ log2 (k), because 2n ≤ k < 2n+1 . k can be written in the form k = 2k + k0 with k = 2n−1 + kn−1 2n−2 + · · · + k1 . Thus g k = g 2k +k0 = (g k)2 g k0 . We therefore obtain g k from g k by squaring and then multiplying by g. The claim is thus proved by induction to n.

Problem of the discrete logarithm Let G by a finite group with the operation ◦. Let α ∈ G and β ∈ H = {αi : i ≥ 0}. We need to find a unique a ∈ N with 0 ≤ a ≤ |H| − 1 and β = αa . We define a as logα (β). Calculating the discrete logarithm A simple procedure for calculating the discrete logarithm of a group element, that is considerably more efficient than simply trying all possible values for k, is the Baby-Step-Giant-Step algorithm. Theorem 5.1. [Baby-Step-Giant-Step algorithm] Let G be a group and g ∈ G. Let n be the smallest natural number with |G| ≤ n2 . Then the discrete logarithm of an element h ∈ G can be calculated base g by generating two lists each containing n elements and comparing these lists. In order to calculate these lists, we need 2n group operations. Proof First create the two lists Giant-Step list: {1, g n , g 2n , . . . , g n·n }, Baby-Step list: {hg −1 , hg −2 , . . . , hg −n }. If g jn = hg −i , i.e. h = g i+jn , then the problem is solved. If the lists are disjoint, then h cannot be represented as g i+jn , i, j ≤ n,. As all powers of g are thus recorded, the logarithm problem does not have a solution. You can use the Baby-Step-Giant-Step algorithm to demonstrate that it is much more difficult to calculate the discrete logarithm than to calculate the discrete exponential function. If the numbers that occur have approximately 1000 bits in length, then you only need around 2000 multiplications to calculate g k but around 2500 ≈ 10150 operations to calculate the discrete logarithm using the Baby-Step-Giant-Step algorithm. In addition to the Baby-Step-Giant-Step algorithm, there are also numerous other procedures for calculating the discrete logarithm [Stinson1995]. The theorem from Silver-Pohlig-Hellman In finite Abelian groups, the discrete logarithm problem can be reduced to groups of a lower order.

Theorem 5.2. [Silver-Pohlig-Hellman] Let G be a finite Abelian group with |G| = pa1 pa2 ·. . .·pas . s 12 The discrete logarithm in G can then be reduced to solving logarithm problems in groups of the order p1 , . . . , ps . If |G| contains a “dominant” prime factor p, then the complexity of the logarithm problem is approximately √ O(p). Therefore, if the logarithm problem is to be made difficult, the order of the group used G should have a large prime factor. In particular, if the discrete exponential function in the group Z∗ is to p be a one way function, then p − 1 must be a large prime factor. Let G be a finite group with operation ◦, and let α ∈ G, so that the discrete logarithm in H = {αi : i ≥ 0} is difficult, Let a with 0 ≤ a ≤ |H| − 1 and let β = αa . Public key: α, β. Private key: a. Let k ∈ Z|H| be a random number and x ∈ G be a plaintext. Encryption: eT (x, k) = (y1 , y2), where y1 = α k and y2 = x ◦ β k . Decryption:
a dT (y1 , y2) = y2 ◦ (y1)−1

Generalised ElGamal procedure (based on the factorisation problem).

Elliptic curves provide useful groups for public key encryption procedures.

References

[Adleman1982] Adleman L.: On breaking the iterated Merkle-Hellman public key Cryptosystem. Advances in Cryptology, Proceedings of Crypto 82, Plenum Press 1983, 303-308. [Balcazar1988] Balcazar J.L., Daaz J., Gabarr¥ J.: Structural Complexity I. Springer Verlag, pp 63. [Brickell1985] Brickell E.F.: Breaking Iterated Knapsacks. Advances in Cryptology: Proc. CRYPTO¥ Lecture Notes in Computer Science, vol. 196, 84, Springer-Verlag, New York, 1985, pp. 342-358. [Lagarias1983] Lagarias J.C.: Knapsack public key Cryptosystems and diophantine Approximation. Advances in Cryptology, Proseedings of Crypto 83, Plenum Press. [Merkle1978] Merkle R. and Hellman M.: Hiding information and signatures in trapdoor knapsacks. IEEE Trans. Information Theory, IT-24, 1978. [RSA1978] Rivest R.L., Shamir A. and Adleman L.: A Method for Obtaining Digital Signatures and Public Key Cryptosystems. Commun. ACM, vol 21, April 1978, pp. 120-126. [Shamir1982] Shamir A.: A polynomial time algorithm for breaking the basic Merkle-Hellman Cryptosystem. Symposium on Foundations of Computer Science (1982), 145-152. [Stinson1995] Stinson D.R.: Cryptography. CRC Press, Boca Raton, London, Tokyo, 1995.

Web links

1. http://www.geocities.com/st busygin/clipat.html

6

Hash Functions and Digital Signatures

(Schneider J. / Esslinger B. / Koy H., June 2002, Update: Feb. 2003, June 2005)
The aim of digital signatures is to guarantee the following two points: • User authenticity: It can be checked whether a message really does come from a particular person. • Message integrity: It can be checked whether the message has been changed (on route). An asymmetric technique is used again (see encryption procedures). Participants who wish to generate a digital signature for a document must possess a pair of keys. They use their secret key to generate signatures and the recipient uses the sender's public key to verify whether the signature is correct. As before, it must be impossible to use the public key to derive the secret key[171] . In detail, a Signature procedure looks like this: Senders use their message and secret key to calculate the digital signature for the message. Compared to hand-written signatures, digital signatures therefore have the advantage that they also depend on the document to be signed. Signatures from one and the same participant are different unless the signed documents are completely identical. Even inserting a blank in the text would lead to a different signature. The recipient of the message would therefore detect any injury to the message integrity as this would mean that the signature no longer matches the document and is shown to be incorrect when verified. The document is sent to the recipient together with the signature. The recipient can then use the sender's public key, the document and the signature to establish whether or not the signature is correct. Because a signature is about as long as the straight datastream to be signed, the procedure we just described has in practice, however, a decisive disadvantage. The signature is approximately as long as the document itself. To prevent an unnecessary increase in data traffic, and also for reasons of performance, we apply a cryptographic hash function[172] to the document

---

[171] With CrypTool you can also generate and check digital signatures: using the submenus of the main menu Digital Signatures / PKI or using menu Indiv. Procedures \ RSA Cryptosystem \ Signature Demonstration (Signature Generation).

[172] Hash functions are implemented within CrypTool at several places. Using menus Individual Procedures \ Hash and Analysis \ Hash you have the possibilities • to apply one of 6 hash functions to the content of the current window, • to calculate the hash value of a file, • to test, how changes to a text change the according hash value, • to calculate a key from a password according to the PKCS#5 standard, • to calculate HMACs from a text and a secret key, and • to perform a simulation, how digital signatures could be attacked by a targeted search for hash value collisions.

– before signing. The output of the hash function will then be signed.

Stanislaw Lem[173] : We can make everything out of this world, but we cannot create a world, where humans in some ten thousand years can think: ’Ok, now it is enough. Everything should stay like it is. Let’s do no changes any more, don’t do inventions any more, because it cannot become better, and if, then we don’t want this.’

6.1

Hash functions

A hash function[174] maps a message of any length to a string of characters with a constant size, the hash value. 6.1.1 Requirements for hash functions

Cryptographically secure hash functions fulfill the following requirements (the order is in a way that the requirements increase): • Resistance against 1st Pre-Image attacks: It should be practically impossible, for a given number, to find a message that has precisely this number as hash value. Given (fix): hash value H’, Searched: message m, so that: H(m) = H’. • Resistance against 2nd Pre-Image attacks: It should be practically impossible, for a given message, to find another message, which has precisely the same hash value. Given (fix): message m1 [and so the hash value H1 = H(m1)], Searched: message m2, so that: H(m2) = H1. • Collision resistance: It should be practically impossible to find any two messages with the same hash value (it doesn’t matter what hash value). Searched: 2 messages m1 and m2, so that: H(m1) = H(m2).

173

This was the answer of Stanislaw Lem to heavy critics at his philosophical main book “Summa Technologiae”, 1964, where he thought about the posibilty of an evolution creating artificial intelligence. Hash algorithms compute a condensed representation of electronic data (message). When a message is input to a hash algorithm, the result is an output called a message digest. The message digests typically range in length from 128 to 512 bits, depending on the algorithm. Secure hash algorithms are typically used with other cryptographic algorithms, such as digital signature algorithms and keyed-hash message authentication codes, or in the generation of random numbers (bits).

174

6.1.2

Current attacks against hash functions like SHA-1

So far, no formal proof has been found that perfectly secure cryptographic hash functions exist. During the past several years no new attacks against hash algorithms came up, and so the candidates that had not yet shown any weaknesses in their structure in practice (e.g. SHA–1175 or RIPEMD-160176) were trusted. At Crypto 2004 (August 2004)177 this safety-feeling was disputed: Chinese researchers published collision attacks against MD4, SHA-0 and parts of SHA-1. This globally caused new motivation to engage in new hash attack methods. Right now the details of the results of the Chinese cryptographers are only announced, but not revealed completely: They stated that collision attacks on SHA-1 can be found with a work load of 269 . This would mean that SHA-1 has cryptographic weaknesses, because the design of SHA-1 should ensure a work load for collision search of about 280 . The value of 269 is currently a prognosis based on theoretic forecasts. The security of already created digital signatures is not affected by the described attacks. According to our current knowledge there is no need to run scared. But in the future digital signatures should use longer hash values and/or other hash algorithms. Already before Crypto 2004 NIST announced, to discontinue SHA-1 in the next few years. So it is recommended not to use SHA-1 for new products generating digital signatures. Further information about this topic can be found in the article “Hash cracked – The consequences of the successful attacks on SHA-1” by Reinhard Wobst and J¨rgen Schmidt178 by Heise Security. u

SHA-1 is a 160 bit hash function specified in FIPS 180-1 (by NIST), ANSI X9.30 Part 2 and [FIPS186]. SHA means Secure Hash Algorithm, and is widely used, e.g. with DSA, RSA or ECDSA. The current standard [FIPS180-2] defines four secure hash algorithms – SHA-1, SHA-256, SHA-384, and SHA-512. For these hash algorithms there are also validation tests defined in the test suite FIPS 140-2. The output length of the SHA algorithms was enhanced because of the possibility of birthday attacks: these make n-bit AES and a 2n-bit hash roughly equivalent: - 128-bit AES – SHA-256 - 192-bit AES – SHA-384 - 256-bit AES – SHA-512. With CrypTool you can comprehend the birthday attack on digital signatures: using the menu Analysis \ Hash \ Attack on the Hash Value of the Digital Signature. 176 RIPEMD-160, RIPEMD-128 and the optional extension RIPEMD-256 have object identifiers defined by the ISOidentified organization TeleTrusT, both as hash algorithm and in combination with RSA. RIPEMD-160 is also part of the ISO/IEC international standard ISO/IEC 10118-3:1998 on dedicated hash functions, together with RIPEMD-128 and SHA-1. Further details: - http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html - http://www.ietf.org/rfc/rfc2857.txt (“The Use of HMAC-RIPEMD-160-96 within ESP and AH”). 177 http://www.iacr.org/conferences/crypto2004/ 178 http://www.heise.de/security/artikel/56634. Further references are e.g.: http://www.bsi.bund.de/esig/basics/techbas/krypto/index.htm http://csrc.nist.gov/CryptoToolkit/tkhash.html.

175

6.1.3

Signing with hash functions

The hash function procedure is as follows: Rather than signing the actual document, the sender now first calculates the hash value of the message and signs this. The recipient also calculates the hash value of the message (the algorithm used must be known), then verifies whether the signature sent with the message is a correct signature of the hash value. If this is the case, the signature is verified to be correct. This means that the message is authentic, because we have assumed that knowledge of the public key does not enable you to derive the secret key. However, you would need this secret key to sign messages in another name. Some digital signature schemes are based on asymmetric encryption procedures, the most prominent example being the RSA system, which can be used for signing by performing the private key operation on the hash value of the document to be signed. Other digital signature schemes where developed exclusively for this purpose, as the DSA (Digital Signature Algorithm), and are not directly connected with a corresponding encryption scheme. Both, RSA and DSA signature are discussed in more detail in the following two sections. After that we go one step further and show how digital signatures can be used to create the digital equivalent of ID cards. This is called Public Key Certification.

6.2

RSA signatures

As mentioned in the comment at the end of section 4.10.3 it is possible to perform the RSA private and public key operation in reverse order, i. e. raising M to the power of d and then to the power of e $\pmod{N}$ yields M again. Based on this simple fact, RSA can be used as a signature scheme. The RSA signature S for a message M is created by performing the private key operation: $S \equiv M^d \pmod{N}$ In order to verify, the corresponding public key operation is performed on the signature S and the result is compared with message M : $S^e \equiv (M^d)^e \equiv (M^e)^d \equiv M \pmod{N}$ If the result matches the message M , then the signature is accepted by the verifier, otherwise the message has been tampered with, or was never signed by the holder of d. As explained above, signatures are not performed on the message itself, but on a cryptographic hash value of the message. To prevent certain attacks on the signature procedure (alone or in combination with encryption) it is necessary to format the hash value before doing the exponentiation, as described in the PKCS#1 (Public Key Cryptography Standard #1 [PKCS1]). The fact that this standard had to be revised recently, after being in use for several years, can serve as an example of how difficult it is to get the details of cryptography right.

6.3

DSA signatures

In August of 1991, the U.S. National Institute of Standards and Technology (NIST) proposed a digital signature algorithm (DSA), which was subsequently adopted as a U.S. Federal Information Processing Standard (FIPS 186 [FIPS186]). The algorithm is a variant of the ElGamal scheme. Its security is based on the Discrete Logarithm Problem. The DSA public and private key and its procedures for signature and verification are summarised below. Public Key p prime q 160-bit prime factor of $p - 1$ $g = h(p−1)/q \bmod p$, where $h < p - 1$ and $h(p−1)/q > 1 \pmod p$ $y \equiv g\ x \bmod p$ Remark: Parameters p, q and g can be shared among a group of users. Private Key $x < q$ (a 160-bit number) Signing m the message to be signed k choose at random, less than q $r = (g\ k \bmod p) \bmod q$ $s = (k\ −1\ (\text{SHA-1}(m) + xr)) \bmod q$ Remark: • (s, r) is the signature. • The security of the signature depends not only on the mathematical properties, but also on using a good random source for k. • SHA-1 is a 160-bit hash function. Verifying $w = s−1 \bmod q$ $u1 = (\text{SHA-1}(m)w) \bmod q$ $u2 = (rw) \bmod q$ $v = (g\ u1\ y\ u2\ ) \bmod p) \bmod q$ Remark: If $v = r$, then the signature is verified. While DSA was specifically designed, so that it can be exported from countries regulating export of encryption soft and hardware (like the U.S. at the time when it was specified), it has been noted [Schneier1996, p. 490], that the operations involved in DSA can be used to emulate RSA and ElGamal encryption.

6.4

Public key certification

The aim of public key certification is to guarantee the connection between a public key and a user and to make it traceable for external parties. In cases in which it is impossible to ensure that a public key really belongs to a particular person, many protocols are no longer secure, even if the individual cryptographic modules cannot be broken. 6.4.1 Impersonation attacks

Assume Charlie has two pairs of keys (PK1, SK1) and (PK2, SK2), where SK denotes the secret key and PK the public key. Further assume that he manages to palm off PK1 on Alice as Bob’s public key and PK2 on Bob as Alice’s public key (by falsifying a public key directory). Then he can attack as follows: • Alice wants to send a message to Bob. She encrypts it using PK1 because she thinks that this is Bob’s public key. She then signs the message using her secret key and sends it. • Charlie intercepts the message, removes the signature and decrypts the message using SK1. If he wants to, he can then change the message in any way he likes. He then encrypts the message again, but this time using Bob’s genuine public key, which he has taken from a public key directory, signs the message using SK2 and forwards it to Bob. • Bob verifies the signature using PK2 and will reach the conclusion that the signature is correct. He then decrypts the message using his secret key. In this way Charlie can listen in on communication between Alice and Bob and change the exchanged messages without them noticing. The attack will also work if Charlie only has one pair of keys. Another name for this type of attack is “man-in-the-middle attack”. Users are promised protection against this type of attack by publickey certification, which is intended to guarantee the authenticity of public keys. The most common certification method is the X.509 standard. 6.4.2 X.509 certificate

Each participant who wants to have an X.509 certificate ([X.509]) verifying that his public key belongs to a real person consults what is known as a certification authority (CA)[179] . He proves his identity to this CA (for example by showing his ID). The CA then issues him an electronic document (certificate) which essentially contains the name of the certificate-holder and the name

---

[179] Often called trust center, if the certificates are not only offered to a closed user group.

of the CA, the certificate-holder’s public key and the validity period of the certificate. The CA then signs the certificate using its secret key. Anyone can now use the CA’s public key to verify whether a certificate is falsified. The CA therefore guarantees that a public key belongs to a particular user. This procedure is only secure as long as it can be guaranteed that the CA’s public key is correct. For this reason, each CA has its public key certified by another CA that is superior in the hierarchy. In the upper hierarchy level there is usually only one CA, which can of course then have its key certified by another CA. It must therefore transfer its key securely in another way. In the case of many software products that work with certificates (such as the Microsoft and Netscape Web browsers), the certificates of these root CAs are permanently embedded in the program right from the start and cannot be changed by users at a later stage. However, (public) CA keys, in particularly those of the root entity, can also be secured by means of making them available publicly.

References

[FIPS180-2] U.S. Department of Commerce/N.I.S.T. , Secure Hash Standard (SHS), August 1, 2002. (FIPS 180-2 will supersede FIPS 180-1 beginning February 1, 2003.) [FIPS186] U.S. Department of Commerce/N.I.S.T. , Entity authentication using public key cryptography, February 18, 1997. No more valid. [FIPS186-2] U.S. Department of Commerce/N.I.S.T. , Digital Signature Standard (DSS), January 27, 2000. Change Note: October 5, 2001. http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf [PKCS1] RSA Laboratories, PKCS #1 v2.1 Draft 3: RSA Cryptography Standard, April 19, 2002. [Schneier1996] Bruce Schneier, Applied Cryptography, Protocols, Algorithms, and Source Code in C, Wiley, 2nd edition, 1996. [X.509] ITU-T, ITU-T Recommendation X.509 (1997 E): Information Technology – Open Systems Interconnection – The Directory: Authentication Framework, June 1997. [X.509v3] ITU-T, X.509 (1993) Amendment 1: Certificate Extensions, The Directory Authentication Framework, International Telecommunication Union, Geneva, Switzerland, July 1995 (equivalent to amendment 1 to ISO/IEC 9594-8).

7

Elliptic Curves

(Filipovics B. / B¨ger M. / Esslinger B. / Oyono R., April 2000, Updates: Dec. 2001, June 2002, u Mar. 2003)

7.1

Elliptic curve cryptography – a high-performance substitute for RSA?

In many business sectors secure and efficient data transfer is essential. In particular, the RSA algorithm is used in many applications. Although the security of RSA is beyond doubt, the evolution in computing power has caused a growth in the necessary key length. Today, 1024-bit RSA keys are standard, but the GISA (German Information Security Agency) recommends the usage of 2048-bit keys from 2006 on (compare section 4.11). The fact that most chips on smart cards cannot process keys extending 1024 bit shows that there is a need for alternatives. Elliptic curve cryptography (ECC) can be such an alternative in the area of asymmetric cryptography. The efficiency of a cryptographic algorithm depends on the key length and the calculation effort that is necessary to provide a prescribed level of security. The major advantage of ECC compared to RSA is that it requires much shorter key lengths. If we assume that the computing power increases by Moore’s law (i. e. it doubles every 18 months)[180], then the evolution of the key lengths for secure communication will be as figure 2 [Lenstra1999] (source: Arjen Lenstra and Eric Verheul: http://cryptosavvy.com/table.htm).

3500 3000

Key length needed (bits)

2500 2000 1500 1000 500 0 2000

RSA

ECC
2010 2020 2030 2040

Year

Figure 2: Prognosis of the key lengths to be regarded safe for RSA and Elliptic Curves In addition, a digital signature can be processed 10-times faster with ECC than with RSA. However, verification of a given signature is still more efficient with RSA than with ECC. Refer to figure 3 (source: Dr. J. Merkle, Elliptic Curve Cryptography Workshop, 2001) for a comparison.

---

[180] empirical knowledge by Gordon Moore, co-founder of Intel, 1965

The reason is that RSA public keys can be chosen relatively small as long as the secret key is long enough.

600 Sign 500 Verify

1024–bit multiplications

400

300

200

100

0

ECC 160 Bit

RSA 1024 Bit

Figure 3: Comparison of signing and verification time for RSA and Elliptic Curves

Nevertheless, thin clients like smart cards usually have to store the (long) secret key and have to process a digital signature rather than verify one. Therefore, there is a clear advantage in using ECC in terms of efficiency. Nowadays, the major problem with ECC-implementations is the lack of standardization. There is only one way to implement RSA, but there are many ways for ECC: One can work with different sets of numbers, different (elliptic) curves – described by parameters[181] – , and a variety of representations of the elements on the curve. Each choice has its advantages and disadvantages, and one can certainly construct the most efficient for each application. However, this causes problems in interoperability. But if all ECC-tools should be able to communicate with each other, they will have to support all different algorithms, which might put the advantage of efficient computation and the need of less storage capacity to the contrary. Therefore, international standardization organizations like IEEE (P1363), ASC (ANSI X9.62, X9.63), ISO/IEC as well as major players like RSA labs or Certicom have recently started standardization initiatives. While the IEEE only describes the different implementations, the ASC has explicitly stated 10 elliptic curves and recommends their usage. The advantage of the ASC approach is that one needs only a single byte to indicate which curve is meant. However, it is

---

[181] see chapter 7.4

not yet clear whether the ASC-curves will become a de facto standard. Although we see no need to replace RSA in any application today[182] , one should take the usage of ECC-based tools into consideration whenever a new system is set up – in particular, when the tool should be available beyond 2005[183] .

7.2 Elliptic curves – history

Mathematicians have been researching elliptic curves for over 100 years. Over the course of time, many lengthy and mathematically complex results have been found and published which are connected to elliptic curves. A mathematician would say that elliptic curves (or the mathematics behind them) are widely understood. This research was originally purely mathematical. That is to say, elliptic curves were investigated, for example, in the mathematical areas of number theory and algebraic geometry, which are generally highly abstract. Even in the recent past, elliptic curves played an important role in pure mathematics. In 1993 and 1994, Andrew Wiles published mathematical works that triggered enthusiasm far beyond the specialist audience. In these works, he proved a conjecture put forward in the 1960’s. To put it short, this conjecture was concerned with the connection between elliptic curves and what are called module forms. What is particularly interesting for most people is that the works of Wiles also proved the famous second theorem of Fermat. Mathematicians had spent centuries (Fermat lived from 1601 to 1665) trying to find a strict proof of this theorem. Understandably, therefore, Wiles’ proof got a good response. Fermat formulated his theorem as follows (written in the border of a book): Cubum autem in duos cubos, aut quadratoquadratum in duos quadratoquadratos, et generaliter nullam in infinitum ultra quadratum potestatem in duos ejusdem nominis fas est dividere: cujus rei demonstrationem mirabilem sane detexi. Hanc marginis exiguitas non caperet. With a free translation, using the denotation of modern mathematics, this means: No positive whole numbers x, y and z greater than zero exist such that $x^n + y^n = z^n$ for $n > 2$. I have found an amazing proof of this fact, but there is too little space within the confines of this book to include it. This is truly amazing: A statement that is relatively simple to understand (we are referring to Fermat’s second theorem here) could only be proved after such a long period of time, although Fermat himself claimed to have found a proof. What’s more, the proof found by Wiles is extremely extensive (all of Wiles publications connected with the proof made up a book in themselves). This should therefore make it obvious that elliptic curves are generally based on highly complex mathematics. Anyway that’s enough about the role of elliptic curves in pure mathematics. In 1985 Neal Koblitz and Victor Miller independently suggested using elliptic curves in cryptography. Elliptic curves have thus also found a concrete practical application. Another interesting area of application

---

[182] Current informationen about the security of the RSA algorithm can be found in chapter 4.11.

[183] Compare the recommendation of GISA: “Fitting Crypto Algorithms” from October 24th, 2002.

for elliptic curves is for factorising whole numbers (the RSA cryptographic system is based on the difficulty/complexity of finding prime factors of an extremely large number; compare section 4.11.). In this area, procedures based on elliptic curves have been investigated and used since 1987 (compare section 7.8). There are also prime number tests based on elliptic curves. Elliptic curves are used differently in the various areas. Encryption procedures based on elliptic curves are based on the difficulty of a problem known as elliptic curve discrete logarithm. The factorisation of whole numbers uses the fact that a large number of elliptic curves can be generated for a natural composite number n with several prime factors; however, these curves are not then groups for composite n. More information about this can be found under the chapter 7.8.

7.3

Elliptic curves – mathematical basics

This section provides information about groups and fields. 7.3.1 Groups

Because the term group is used differently in everyday language than in mathematics, we will, for reasons of completeness, begin by introducing the essential statement of the formal definition of a group: • A group is a non-empty set G on which an operation "·". The set G is closed under this operation, which means that for any two elements a, b taken from G, performing the operation on them gives an element in G, i.e. $ab = a \cdot b$ lies in G. • For all elements a, b and c in G: $(ab)c = a(bc)$ (associative law). • There exists an element e in G that behaves neutrally with respect to the operation ·. That means that for all a in the set G : $ae = ea = a$. • For each element a in G there exists a so-called inverse[184] element a^{-1} in G such that: $aa^{-1} = a^{-1} a = e$. If also $ab = ba$ (commutative law) for all a, b in G, then we call the group an Abelian group. Since we may define different operations on the same set, we distinguish them by giving them different names (e.g. + addition or · multiplication). The simplest example of an (Abelian) group is the group of whole numbers under the standard operation of addition. The set of whole numbers is denoted as Z. Z has an infinite number of elements, because $Z = \{\cdots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \cdots\}$. For example, the operation of 1 + 2 lies in Z, for 1 + 2 = 3 and 3 lies in Z. The neutral element in the group Z is 0. The inverse element of 3 is −3, for 3 + (−3) = 0.

---

[184] The inverse is uniquely determined because if $x, y \in G$ are each inverse to a, i.e. $ax = xa = e$ and $ay = ya = e$, then $x = xe = x(ay) = (xa)y = ey = y$.

For our purpose, so-called finite groups play an important role. This means that these exists a set M with a fixed number of elements and an operation + such that the above conditions are fulfilled. One example of this is any set Zn where Zn = {0, 1, 2, 3, · · · , n − 1}, n is a positive whole number and the operation is addition mod n, i.e. a and b in Zn are subject to the operation a + b mod n. Cyclic groups Cyclic groups[185] are those groups G that possess an element g from which the group operation can be used to generate all other elements in the group. This means that for each element a in G there exists a positive whole number i such that if g is subject to the operation i times (i.e. “g · i”), g + g + · · · + g = a (additive group) or g i = g · g · · · g = a (multiplicative group). The element g is the generator of the cyclic group – each element in G can be generated using g and the operation. Group order Now to the order of an element of the group: Let a be in G. The smallest positive whole number r for which a subject to the operation with itself r times is the neutral element of the group G (i.e.: r · a = a + a + · · · + a = e respectively ar = e), is called the order of a. The order of the group is the number of elements in the set G. 7.3.2 Fields

In mathematics, one is often interested in sets on which at least two (group) operations are defined – frequently called addition and multiplication. Most prominent are so called fields. A field is understood to be a set K with two operations (denoted as + and ·) which fulfils the following conditions: • The set K forms an Abelian group together with the operation + (addition), where 0 is the neutral element of the operation +. • The set K \ {0} also forms an Abelian group together with the operation · (multiplication). • For all elements a, b and c in K, we have c · (a + b) = c · a + c · b and (a + b) · c = a · c + b · c (distributive law). Fields may contain an infinite number of elements (e.g. the field of real numbers). They are called infinite fields. In contrast we call a field finite, if it contains only a finite number of elements (e.g. Zp = {0, 1, 2, 3, · · · , p − 1} , where p is a prime. Zp with addition mod p and multiplication mod p).

---

[185] Cyclic groups can be in general also endless like the additive group of the integer numbers. We consider here only finite cyclic groups.

Characteristic of a field Let K be a field and 1 be the neutral element of K with respect to the multiplicative operation “·”. Then the characteristic of K is said to be the order of 1 with respect to the additive operation. This means that the characteristic of K is the smallest positive integer n such that 1 + 1 + · · · + 1 = 0. n times If there is no such n, i.e. if 1 + 1 + · · · + 1 = 0 no matter how many 1s we add, then we call K a field with characteristic 0. Thus, fields with characteristic 0 are infinite since they contain the (pairwise distinct) elements 1, 1 + 1, 1 + 1 + 1, ... On the other hand, fields with finite characteristic may by finite or infinite. If the characteristic is finite, it has to be prime. This fact can easily be proved: Assume n = pq, p, q < n, is the characteristic of a field K. By definition of n, the elements p = 1 + 1 + · · · + 1, Ø q = 1 + 1 + · · · + 1 of K are not equal to 0. Thus, there exist inverse elements Ø p−1 , q −1 ØØ p times with

q times respect to multiplication. It follows that (Øq)(Ø−1 q −1) = 1, which contradicts the fact that pØ p Ø pq = n = 1 + 1 + · · · + 1 = 0 and, hence, (Øq) (Ø−1 q −1) = 0. ØØ Ø pØ p Ø
=0 n times Comment: The field of real numbers has the characteristic 0; the field Zp has the characteristic p. If p is not prime, Zp is not a field at all.

The most simple field is Z2 = {0, 1}. It contains only two elements, the neutral elements with respect to addition and multiplication. In particular, we have 0 + 0 = 0, 0 + 1 = 1 + 0 = 1, 1 + 1 = 0, 1 · 1 = 1, 0 · 0 = 0 · 1 = 1 · 0 = 0. Finite Fields As mentioned above, each finite field has a characteristic p = 0, where p is a prime. On the other hand, given a prime p there is a field which has exactly p elements, that is Zp . However, the number of elements of a field need not be prime in general. For example, it is not hard to construct a field with 4 elements186 . One can show that the order of any field is a prime power (i.e. the power of a prime number). On the other hand, we can construct a field with pn elements for any given prime p and positive integer n. Since two fields that have the same number of elements can not be distinguished187 ,

186

The + 0 1 a b

set 0 0 1 a b

K = {0, 1, a, b} fitted with the 1ab · 1ab 0 und 0ba 1 b01 a a10 b

operation 01a 000 01a 0ab 0b1

defined in the tabular below is a field: b 0 b 1 a

187

If K, K are fields with k = pn elements, then there is a one-to-one map ϕ : K → K , that respects the arithmetic of the field. Such a map is called an isomorphy. Isomorphic fields mathematically behave in the same way so that dass it makes no sense to distinguish between them. For example, Z2 und K = {ZERO, ON E} with zero-we say that there is the field with pn elements and denote it by GF (pn). Here GF stands for Galois Field to commemorate the French Mathematician Galois. The fields GF (p) of prime order play a prominent role. They are called prime fields and often denoted by Zp 188 .

7.4

Elliptic curves in cryptography

In cryptography elliptic curve are a useful tool. Such curves are described by some equation. A detailed analysis has shown that curves of the form189 F (x1 , x2 , x3) = −x3 + x2 x3 + a1 x1 x2 x3 − a2 x2 x3 + a3 x2 x2 − a4 x1 x2 − a6 x3 = 0, 1 2 1 3 3 3 (1)

are especially useful. The variables x1 , x2 , x3 and parameters a1 , . . . , a4 , a6 are elements of a given field K, which has certain properties that are make it useful from the cryptographic point of view. The underlying field K might be the well known field of real numbers or some finite field (see last section). In order to obtain a cure that is useful for cryptography, the parameters have to be chosen in a way that the following conditions hold ∂F ∂F ∂F = 0, = 0, = 0. ∂x1 ∂x2 ∂x3 We identify points on the curve that can be derived from each over by multiplying each component with some scalar. This makes sense since (x1 , x2 , x3) solves (1) if and only if α(x1 , x2 , x3) (α = 0) does. Formally, this means that we consider classes of equivalent points instead of single points, where points are called equivalent if one is the scalar multiple of the other one. If we put x3 = 0 in the basic equation (1), then this equation collapses to −x3 = 0, leading 1 to x1 = 0. Thus, the equivalence class which includes the element (0, 1, 0) is the only one that contains a point with x3 = 0. For all points on the curve that are not equivalent to (0, 1, 0), we may apply the following transformation K ◊ K ◊ (K \ {0}) (x1 , x2 , x3) → (x, y) := x1 x2 , x3 x3 ∈K ◊K,

which reduces the number of variables to two instead of three. We note that the basic equation (1) F (x1 , x2 , x3) = 0 was chosen in a way that this transformation leads to the famous so-called Weierstrass-Equation190 holds y 2 + a1 xy + a3 y = x3 + a2 x2 + a4 x + a6 . (2)

element ZERO and one-element ON E are isomorphic. We note that mathematical objects are only defined by their mathematical properties. 188 For prime fields additive as well as multiplicative group are cyclic. Furthermore, each field GF (pn) contains a subfield that is isomorphic to the prime field Zp . 189 This curve is given by the zeros of a polynomial F of degree three in three variables. In general, expressions of P i1 in the form P = i1 ,...,in ∈I 0 ai1 ...in x1 . . . xn with coefficients ai1 ...in ∈ K are called polynomials in n variables N x1 , . . . , xn with underlying field K, if deg P := max{i1 + · · · + in : ai1 ...in = 0} is finite, i.e. the sum has only finitely many non-zero terms (monomials). The sum of the exponents of the variables of each term of the sum is at most 3, at least one term of the sum has a single variable with 3 as value of the according exponent. 190 Karl Weierstrass, 31.10.1815–19.12.1897, German mathematician, famous for his rigorous formal approach to mathematics.

Since all but one point (i.e. equivalence class) of the elliptic curve can be described using equation (2), this equation is often called the elliptic equation, and its solutions written as E = (x, y) ∈ K ◊ K | y 2 + a1 xy + a3 y = x3 + a2 x2 + a4 x + a6 ∪ {O}. Here, O represents the point (0, 1, 0) that is loosely speaking mapped to infinity by the transformation (division by x3) that reduces the three variables to two.

y

x

Figure 4: Example of an elliptic curve with the real numbers as underlying field.

In contrast to figure 4 only finite fields K = GF (pn) are used in elliptic curve cryptography. The reason is loosely speaking that in modern communication engineering data processing is always based on discrete data (simply because computers accept only discrete data). For practical reasons, it turned out to be useful to take either GF (p) with a large prime p or GF (2n) with a (large) positive integer n. Using GF (p) has the advantage of providing a relatively simple arithmetic; on the other hand GF (2n) allows a binary representation of each element that supports the way computers work. Other fields like, for example, GF (7n) do not have any of these advantages and are, thus, not considered, although there is no mathematical reason why they should not. A coordinate transformation can result in a simpler version[191] of the Weierstrass equation. Depending whether p > 3, different transformations are used, and we obtain • in case of GF (p), p > 3, the elliptic curve equation of the form y 2 = x3 + ax + b with 4a3 + 27b2 =0 (3)

---

[191] Such a coordinate transformation is combination of a rotation and a dilatation of the coordinate system without changing the elliptic curve itself.

- in case of GF (2n) the elliptic curve equation of the form y 2 + xy = x3 + ax2 + b with b = 0192 . This conditions on the parameters a, b ensure that the elliptic equation can be used in the context of cryptography193 . Let |E| denote the number of elements of an elliptic curve E given an underlying field GF (k) (for practical reasons either k = p with p prim or k = 2n). Then Hasse’s theorem[Silverman1986] √ √ √ yields | |E| − k − 1 | ≤ 2 · k. This Inequality is equivalent to k + 1 − 2 k < |E| < k + 1 + 2 k. In particular, this means that the number of elements of an elliptic curve is approximately k (for large k). (4)

7.5

Operating on the elliptic curve

In order to work with elliptic curves in practice, we define an operation (often written in an additive way +) on the set of points on the curve. If we have a curve over the field GF (p), we define the commutative operation + by 1. P + O = O + P = P for all P ∈ E, 2. for P = (x, y) and Q = (x, −y) we set P + Q = O, 3. for P1 = (x1 , x2), P2 = (x2 , y2) ∈ E with P1 , P2 = O and (x2 , y2) = (x1 , −y1) we set P3 := P1 + P2 , P3 = (x3 , y3) defined by x3 := −x1 − x2 + λ2 , with the auxiliary quotient λ :=
y1 −y2 x1 −x2 3x2 +a 1 2y1

y3 := −y1 + λ(x1 − x3)

if P1 = P2 , if P1 = P2 .

In particular, we obtain −P = (x, −y) for P = (x, y) ∈ E. If we deal with a curve over the field GF (2n), we define the operation + in an analogous way by 1. P + O = O + P = P for all P ∈ E, 2. for P = (x, y) and Q = (x, x + y) we set P + Q = O,

The form (3) is called the standard form of the Weierstrass-equation. If the characteristic of the field is 2 or 3, we obtain 4 = 0 respectively 27 = 0, which means that the condition on parameters a, b collapse. Loosely speaking, this is the reason why the transformation to the standard form does not work in these cases. 193 Formally we call such curves non singular.

192

3. for P1 = (x1 , x2), P2 = (x2 , y2) ∈ E with P1 , P2 = O and (x2 , y2) = (x1 , x1 + y1) we set P3 := P1 + P2 , P3 = (x3 , y3) defined by x3 := −x1 + x2 + λ + λ2 + a , with auxiliary quotient λ :=
y1 +y2 x1 +x2 y x1 + x1 1

y3 := y1 + x3 + λ(x1 + x3) if P1 = P2 , if P1 = P2 .

In particular, we obtain −P = (x, −y) for P = (x, y) ∈ E. (Note that −(−P) = (x, x + (x + y)) = (x, 2x + y) = (x, y), since the underlying field has characteristic 2.)[194] One can verify that + defines a group operation on the set E ∩ {O}. In particular this means that the sum of two points is again a point on the elliptic curve. How his operation works is geometrically visualized in the following section.

[194] An animation of the addition of points on elliptic curves can be found on the Certicom-Homepage http://www.certicom.com/resources/ecc tutorial/ecc tutorial.html

How to add points on an elliptic curve The following figures show how points on an elliptic curve over the field of real numbers are summed up using affine coordinates. We note that the point infinity O cannot be shown in the affine plane.

y P=Q L R

x L’

2P
Figure 5: Doubling of a point

y Q P L R

x
L’ P+Q

Figure 6: Summing up two different points over the real number field

7.6

Security of elliptic-curve-cryptography: The ECDLP

As mentioned above in section 7.4, we only consider elliptic curves over the finite[195] fields GF (2n) or GF (p) (for a large prime p). This means that all parameters that describe the curve are taken from this underlying field. If E is an elliptic curve over such a field and P is a point on the curve E, then we can derive for all positive integers m mP := P + P + · · · + P . m times Looking on this operation from the cryptographic point of view, it turns out to be very interesting by the following reason: On the one hand one needs only log m operations to calculate mP – one simply has to calculate P , 2P , 22 P , 23 P , . . . , write m in a binary form and finally add all these multilples 2k P of P with respect to the binary representation of m – on the other hand it seems to be very hard to find m given P and Q = mP on E. Of course, we may simply calculate P, 2P, 3P, 4P, 5P, . . . and compare each of them with Q. But this will take as much as m operations. Yet there is no algorithm known that efficiently derives m given P and G. The best algorithms √ known so far need about q operations where q is the (largest) prime factor of p − 1, in case the underlying field is GF (p); here m should be between 1 and q liegen so that one needs at most √ q log q operations to calculate mP . However, the quotient log q tends to +∞ very fast for large q. If we choose the parameters sufficiently large (for example, let p be prime and at least 160 bits long), an computer will easily be able to calculate mP (in less than a second). The inverse problem however, to derive m from mP and P , can (still) not be solved in reasonable time. This problem is known as the “over Elliptic Curve Discrete Logarithm Problem” (for short ECDLP). In elliptic curve cryptography we formally look at points on the elliptic curve as elements of a group with point addition + as operation. Furthermore, we use only elliptic curves that have a sufficiently large number of points. However, in special cases curves may be weak and not useful due to other reasons. For such special cases the ECDLP can be much easier to solve than in the general case. This means that one has to look carefully at the parameters when choosing an elliptic curve for cryptographic applications. Not useful for cryptography are a-normal (that are curves over Zp , for which the set E consists of exactly p elements) and supersingular curves (that are curves, for which the ECDLP can be reduced to the “normal” discrete logarithms in another, smaller finite field). This means that there are cryptographically useful and non-useful elliptic curves. Given the parameters a and b, it is possible to determine whether a curve is useful or not. In many publications one can find parameters that turned out to be useful for cryptography. The open (scientific) discussion guarantees that these results take into account latest research. Given a secure curve, the time that is needed to solve the ECDLP is strongly correlated with parameter p in case GF (p) respectively n in case of GF (2n). The larger these parameters become,

---

[195] Discrete in contrast to continuous.

the more time an attacker needs to solve the ECDLP – at least with the best algorithms known so far. Experts recommend bit-lengths of 200 for p for secure curves. A comparison with RSA modulus length shows why elliptic curves are so interesting for applications. We note that the computation effort for signing and encryption is closely related to the bit-length of the parameters. In addition the initiation process, i.e. the generation of the private-public-key-pair, becomes more complicated the larger p is. Thus, one looks for the smallest parameters that still come along with the security required. It is remarkable that a length of 200 bits for p is sufficient to construct a good elliptic curve that is as secure as RSA with a 1024 bit RSA modulus (as far as we know today). For short, the reason for this advantage of ECC lies in the fact that the best algorithms known for solving the ECDLP need exponential time while the best algorithms for factorizing are sub-exponential (number field sieve, quadratic sieve or factorizing with elliptic curves). Hence, the parameters for a cryptosystem that is based on the problem of factorizing large integers have to be larger than the parameters for a system based on ECDLP.

7.7

Encryption and signing with elliptic curves

The elliptic curve discrete logarithm problem (ECDLP) is the basis for elliptic curve cryptography. Based on this problem, there are different signature schemes. In order to apply one of these, we need: • An elliptic curve E with an underlying field GF (pn). • A prime q = p and a point G on the elliptic curve E with order q. This means that $qG = O$ and $rG = O$ for all $r \in \{1, 2, . . . , q - 1\}$. Thus q is a factor of the group order (i.e. the number of elements) #E of E. Since q is prime, G generates a cyclic sub-group of E of order q. The parameters mentioned are often called Domain parameter. They describe the elliptic curve E and the cyclic sub-group of E on which the signature scheme is based. 7.7.1 Encryption

Using elliptic curves one can construct a key exchange protocol based on the Diffie-Hellman protocol (see chapter 5.4.2). The key exchanged can be used for a subsequent symmetric encryption. We note that in contrast to RSA there is no pair of private and public key that can be used for encryption and decryption! In the notation of elliptic curves, the Diffie-Hellman protocol reads as follows: First both partners (A und B) agree on a group G and an integer q. Then they choose rA , rB ∈ {1, 2, . . . , q − 1} at random, derive the points RA = rA G, RB = rB G on the elliptic curve and exchange them (using an insecure channel). After that A easily obtains R = rA RB ; B gets the same point (R = rA rB G) by calculating rB RA = rB rA G = rA rB G = R. We note that RA , RB are easy to derive as long as rA respectively rB are known G. However, the inverse operation, to get RA respectively RB from rA respectively rB is hard.

Using the best algorithms known so far, it is impossible for any attacker to obtain R without knowing either rA or rB – otherwise he would have to solve the ECDLP. In order to prohibit a “Man-in-the-middle” attack, one may sign the values G, q, RA , RB as described in chapter 6.4.1. 7.7.2 Signing

Using the DSA signature scheme, one can proceed as follows: The signing party chooses a (nontrivial) number s ∈ Zq , which will be the private key, and publishes q, G and R = sG. We note that s cannot be obtained from G and R are not sufficient – a fact on which the security of the signature scheme is based. Given the message m, which should be signed, one first constructs a digital finger print using a hash-algorithm h such that h(m) has its values in {0, 1, 2, . . . , q − 1}. Thus, h(m) can be considered as an Element of Zq . Then the signing party chooses a random number r ∈ Zq and derives R = (r1 , r2) = rG. We note that the first component r1 of R is an element of GF (pn). This component will then be projected onto Zq , i.e. in case of n = 1 it is interpreted as the remainder of an element of {0, 1, . . . , p − 1} divided by q. This projection of r1 onto Zq is denoted by r1 . Then one determines x ∈ Zq such that Ø rx − sØ1 − h(m) = 0. r The triple (m, r1 , x) is then published as the digital signature of message m. 7.7.3 Signature verification

In order to verify a signature, one has to build u1 = h(m)/x, u2 = r1 /x (in Zq and derive Ø V = u1 G + u2 Q. Since we have Q = sG, the point V = (v1 , v2) satisfies v1 = u1 +u2 s. We note that this operations take place in the field GF (pn). The projection of GF (pn) on Zq mentioned above should be chosen in such a way that v1 = u1 + u2 s is an element of Zq . Then it follows that Ø v1 = u1 + u2 s = h(m)/x + r1 s/x = (h(m) + r1 s)/x = rx/x = r. Ø Ø Ø Since R = rG, we obtain v1 = r1 , i.e. R and V coincide modulo the projection onto Zq . Ø Ø

7.8

Factorisation using elliptic curves

There are factorisation[196] algorithms based on elliptic curves[197]. More precisely, these procedures exploit the fact that elliptic curves can be defined over Zn (n composite number). Elliptic curves over Zn do not form a group, because not every point on such an elliptic curve has an inverse point. This is connected with the fact that - if n is a composite number - there exist elements in Zn that do not have an inverse with respect to multiplication mod n. In order to add two points on an elliptic curve over Zn , we can calculate in the same way as on elliptic curves over Zp . Addition of two points (on an elliptic curve over Zn), however, fails if and only if a factor of n has been found. The reason for this is that the procedure for adding points on elliptic curves gives elements in Zn and calculates the inverse elements for these (with respect to multiplication mod n) in Zn . The extended Euclidean algorithm is used here. If the addition of two points (that lie of an elliptic curve over Zn) gives an element in Zn that does not have an inverse element in Zn , then the extended Euclidean algorithm delivers a genuine factor of n. Factorisation using elliptic curves thus principally works as follows: Random curves over Zn are selected, as well as random points (that lie on this curve) and add them; you thus obtain points that also lie on the curve or find a factor of n. Factorisation algorithms based on elliptic curves therefore work probabilistically. The opportunity of defining large number of elliptic curves over Zn allows you to increase the probability of finding two points which you can add to obtain a factor of n. These procedures are therefore highly suitable for parallelisation.

7.9

Implementing elliptic curves

CrypTool also offers elliptic curves for the digital signature function[198]. It implements the basic algorithms for group operations, for generating elliptic curves, for importing and exporting parameters for elliptic curves over finite fields with p (p prime) elements. The algorithms have been implemented in ANSI C and comply with draft no. 8 of the IEEE P1363 work group Standard Specifications for Public Key Cryptography http://grouper.ieee.org/groups/1363. The procedure implements the cryptographic primitives for generating and verifying signatures for the variations of Nyberg-Rueppel signatures and DSA signatures based on elliptic curves (in accordance with draft no. 8 of the IEEE P1363 work group). This was done in collaboration with the Secude GmbH – using the above library and the Secude SDK. In case one uses the field GF (2n) is used instead of the prime field GF (p), one has to make sub-

---

Especially John M. Pollard was involved in the development of many different factorisation algorithms; also at factorisation with ECC he was one of the leading heads. As an employee of British Telekom he never published much. At the RSA data Security Conference in 1999 he was awarded for his “outstanding contributions in mathematics”: http://www.eff.org/Privacy/Crypto misc/DESCracker/HTML/19990118 rsa awards.html.

[197] In 1987 H.W. Lenstra published a factorization algorithm, based on elliptic curves (see [Lenstra1987]). The biggest compound number currently factorised with elliptic curves is the number 62859 − 1, which has 55 decimal digits. It was found Oct. 6th, 2001 by M. Izumi (See ECMNET).

[198] The dialog box, which appears in CrypTool after clicking the menu Digital Signatures/PKI \ Sign Message, offers the EC methods ECSP-DSA and ECSP-NR.

stantial changes in the implementation. The advantage of GF (2n) lies in the fact that calculations in GF (2n) can be implemented very efficiently using the binary representation. In particular, divisions are much easier to process compared to GF (p) (this is particularly important in the signature scheme mentioned above where a division is needed for processing a signature as well as for the verification). In order to achieve maximal gain in efficiency, one may choose a field that allows special basis like polynomial basis (useful for software implementations) or normal basis (best for hardware implementations). For special n (like, for example, n = 163, 179, 181) one may even combine both advantages. However, they are still non-standard. Sometimes only the first component and one additional bit is used as representation of a point on the elliptic curve instead of the full two components. Since the first component together with the additional bit is sufficient to derive the full point, this representation minimizes the memory capacity needed. In particular, for normal basis this point compression can be implemented efficiently. In addition, the cryptographic protocols themselves become more effective. A disadvantage is, however, that point compression can be used for about half of all elliptic curves only and is protected under US patent (US Patent 6141420, Certicon), causing additional costs. In the gerenal case GF (pn) (and also in case n = 1) often so called affine or projective co-ordinates are used. Depending on the application, these co-ordinates may result in a gain in efficiency as well. A comprehensive description of all implementations and their advantages and disadvantages would go far beyond the scope of this paper. We only want to state that there is a variety of possible implementations for elliptic curve cryptography, much more than for RSA. Therefore, there are serious efforts to reduce this large to a small number of standard implementations. Some standardization committees even try to reduce the complexity by focussing on a small number of (prescribed) curves (ASC-approach). Today it is still not clear whether these standardization initiatives will be successful or not. However, without agreed standards, ECC is not likely to become a real alternative for RSA. The committees might be forced to act fast if there was a break-through in factorization.

7.10

Elliptic curves in use

Today elliptic curve cryptography is already in use. A prominent example is the information network Bonn-Berlin[199] , used for the exchange of strictly confidential documents between different German federal governmental institutions in Berlin and Bonn. With the help of ECC a high security solution could be realized. Interoperability, however, played only a minor role. Based on information from the head of the Austrian e-Government projects, Prof. Posch, a smartcard based on ECC will shortly be launched in Austria: A bank card that allows digital signing will be issued in Austria from 2004 on to all citizens. Both examples show the typical range of application for elliptic curve cryptography: For high

[199] The Informationsverbund Bonn-Berlin (IVBB) connects governmental institutions in the old and new German capital.

security solutions and for implementations on smartcards in which the key length is crucial (because of physical memory available).

References

[Cassels1991] J. W. S. Cassels, Lectures on elliptic curves, Cambridge University Press, 1991, 143 pages. [Koblitz1984] N. Koblitz, Introduction to elliptic curves and modular forms, Graduate Texts in Mathemathics, Springer-Verlag, 1984. [Koblitz1998] N. Koblitz, Algebraic aspects of Cryptography. With an appendix on Hyperelleptic curves by Alfred J. Menezes, Yi Hong Wu and Robert J. Zuccherato, Springer-Verlag, 1998, 206 pages. [Menezes1993] A. J. Menezes, Elliptic curve public key cryptosystems, Kluwer Academic Publishers, 1993. [Lenstra1987] H.W. Lenstra, Factoring integers with elliptic curves, Annals of Mathematics 126, pp. 649-673, 1987. [Lenstra1999] Arjen K. Lenstra, Eric R. Verheul Selecting Cryptographic Key Sizes (1999), Journal of Cryptology: the journal of the International Association for Cryptologic Research http://www.cryptosavvy.com/cryptosizes.pdf [Silverman1986] J. Silverman, The Arithmetic of Elliptic Curves, Springer-Verlag, 1986. [Silverman1992] J. Silverman, The arithmetic of elliptc curves, Graduate Texts in Mathemathics, Springer-Verlag, 1992. [SilvermanTate1992] J. Silverman, J. Tate, Rational points on elliptic curves, Springer-Verlag, 1992.

Web links

1. Certicom Online Tutorial, http://www.certicom.com/resources/ecc tutorial/ecc tutorial.html

2. Working group IEEE P1363 http://grouper.ieee.org/groups/1363 3. An informative web page about factorisation with elliptic curves. http://www.loria.fr/~zimmerma/records/ecmnet.html It contains literature related to the topic factorisation with elliptic curves as well as links to other web page. 4. Key length comparison by Arjen Lenstra and Eric Verheul http://cryptosavvy.com/table.htm

A

Appendix

A.1

CrypTool Menus

This appendix contains the complete menu tree of CrypTool version 1.4.00. Which menu items are active (that is not greyed), depends on the type of the currently active document window. The brute-force analysis for DES e. g. is only available, if the active window is opened in the hexadecimal view. On the other hand the menu item “Generate Random Numbers. . . ” is always available. The following types of documents exist in CrypTool: Code letter A T H P Type of document ASC Text Hexadecimal Plot

File Edit New Open... Close Save Save as... File Properties... Print... Print Setup... Recent Files Exit Parent Window Permutation... Symmetric (modern) IDEA... RC2... RC4... DES (ECB)... DES (CBC)... Triple DES (ECB)... Triple DES (CBC)... MARS... RC6... Rijndael (AES)... Serpent... Twofish... AES (self extracting)... Asymmetric RSA Encryption... RSA Decryption... RSA-Demonstration... Hybrid RSAAES Encryption... RSAAES Decryption... Protocols Diffie-Hellman Demonstration... Network Authentication... Chinese Remainder Theorem Applications Astronomy and Planetary Motion... Modular Foreward and Backward Transformation... Secret Sharing by CRT... Visualization of Algorithms using ANIMAL Caesar... VigÈnere... Nihilist... DES... Codes Base64 Encode/Decode UU-Encode/Decode Decode ASN.1 Code of a File... Compress Zip UnZip Generate Random Numbers... Hash RSA Demonstration... Factorisation of a Number... Signature Demonstration (Signature Generation)... Lattice Based Attacks on RSA Factoring with a Hint... Attack on Stereotyped Messages... Attack on Small Secret Keys... Generate Prime Numbers... RSA Cryptosystem Homophone... Generation of MACs... Show Key... Vernam... Key Generation from Password... Select All XOR... Hash Demonstration... Replace... Byte Addition... Hash Value of a File... Find Next ADFGVX... Signature Demonstration (Signature Generation)... RIPEMD-160 Find... Playfair... Extract Signature SHA-1 Delete Bar Chart Substitution... Verify Signature... SHA Paste As HexDump Hill... Sign Message... MD5 Copy Show as Text VigenËre... Key Display/Export... MD4 Cut Status Bar Caesar... Key Generation/Import... MD2 Entropy Floating Frequency Histogram N-Gram... Autocorrelation Periodicity Undo Analysis Tools for Analysis View Toolbar Digital Signatures/PKI PKI Crypt/Decrypt Symmetric (classic)

Indiv. Procedures Hash

Options Plot Options... Analysis Options... Text Options... Starting Options... Further Options...

Window Cascade

Help

Starting Page

Tile

Topics

Arrange Icons

Index

Close All

Scenarios (Tutorials)

Readme

Script

About CrypTool Mass Pattern Search... Symmetric Encryption (classic) Ciphertext only Caesar VigenËre ADFGVX... Addition XOR Known Plaintext Hill... Manual Analysis Substitution... Playfair... Symmetric Encryption (modern) IDEA... RC2... RC4... DES (ECB)... DES (CBC)... Triple DES (ECB)... Triple DES (CBC)... MARS... RC6... Rijndael (AES)... Serpent... Twofish... Asymmetric Encryption Factorisation of a Number... Lattice Based Attacks on RSA Factoring with a Hint... Attack on

Stereotyped Messages... Attack on Small Secret Keys... Side-Channel Attack on "Textbook RSA"...

Figure 7: Complete overview of the CrypTool menu tree

Attack on the Hash Value of the Digital Signature... Analyse Randomness Frequency Test Poker Test Runs Test Serial Test FIPS PUB-140-1 Test Battery Vitany 3-D Visualization...

A.2

Authors of the CrypTool Script

This appendix lists the authors of this document. Please refer to the top of each individual chapter for their contribution. Bernhard Esslinger, initiator of the CrypTool project, main author of this script, head IT security at Deutsche Bank and lecturer on IT security at the University of Siegen. E-mail: besslinger@web.de. Matthias Büger, contribution to the chapter “Elliptic Curves”, research analyst at Deutsche Bank. Bartol Filipovic, original author of the CrypTool elliptic curve implementation and the corresponding chapter in this script. Henrik Koy, main developer and co-ordinator of CrypTool development since version 1.3; script reviewer and TEX guru; cryptographer and project leader IT at Deutsche Bank. Roger Oyono, implementer of the CrypTool factorization dialogue and original author of chapter “The Mathematical Ideas behind Modern Cryptography”. Jörg Cornelius Schneider, design and support of CrypTool; crypto enthusiast and senior project leader IT at Deutsche Bank. Christine Stötzel, Master of Business and Computer Science at the University of Siegen.

A.3

Bibliography of Movies and Fictional Literature with Relation to Cryptograpy, Books for Kids with Collections of Simple Ciphers

Cryptographic applications – classical as well as modern ones – have been used in literature and movies. In some media they are only mentioned and are a pure addmixture; in others they play a primary role and are explained in detail; and sometimes the purpose of the story, which forms the framework, is primarily to transport this knowledge and achieve better motivation. Here is the beginning of an overview. [Poe1843] Edgar Allan Poe, The Gold Bug, 1843. In this short story Poe tells as first-person narrator about his acquaintanceship with the curious Mr. Legrand. They detect a fabulous treasure via a gold bug and a vellum found at the coast of New England. The cipher consists of 203 cryptic symbols and it proves to be a monoalphabetic substitution cipher. The story tells how they solve the riddle step by step using a combination of semantic and syntax analysis (frequency analysis of single letters in English texts). In this novel the code breaker Legrand says the famous statement: “Yet it may be roundly asserted that human ingenuity cannot concoct a cipher which human ingenuity cannot resolve – given the according dedication.” Yet it may be roundly asserted that human ingenuity cannot concoct a cipher which human ingenuity cannot resolve... [Verne1885] Jules Verne, Mathias Sandorf, 1885. This is one of the most famous novels of the French author Jules Verne (1828-1905), who was called “Father of Science fiction”. In “Mathias Sandorf” he tells the story of the freedom fighter Earl Sandorf, who is betrayed to the police, but finally he can escape. The whistle-blowing worked, because his enemies captured and decrypted a secret message sent to him. For decryption they needed a special grille, which they stole from him. This turning grille was a quadratic piece of jig with 6x6 squares, of which 1/4 (nine) were holes (see the turning grille in chapter 2.1.1). [Doyle1905] Arthur Conan Doyle, The Adventure of the Dancing Men, 1905. In this Sherlock-Holmes short story (first published in 1903 in the “Strand Magazine”, and then in 1905 in the collection “The Return of Sherlock Holmes” the first time in bookform) Sherlock Holmes has to solve a cipher which at first glance looks like a harmless kid’s picture. But it proves to be the monoalphabetic substitution cipher of the criminal Abe Slaney. Sherlock Holmes solves the riddle using frequency analysis. [Sayer1932] Dorothy L. Sayer, Have his carcase, Harper / Victor Gollancz Ltd., 1932.

In this novel the writer Harriet Vane finds a dead body at the beach. The police believe the death is suicide. Harriet Vane and the elegant amateur sleuth Lord Peter Wimsey together clear of the disgusting murder in this second of Sayers's famous Harriet Vane mystery series. This requires to solve a cryptogram. Surprisingly the novel not only describes the Playfair cipher in detail, but also the cryptanalysis of this cipher (see Playfair in chapter 2.2.3). [Arthur196x] Robert Arthur, The Three Invesigators: The Secret Key (German version: Der geheime Schl¨ssel nach u Alfred Hitchcock (volume 119), Kosmos-Verlag (from 1960) The three detectives Justus, Peter and Bob have to decrypt covered and encrypted messages within this story to find out what is behind the toys of the Copperfield company. [Seed1990] Directed by Paul Seed, House of Cards, 1990. In this movie Ruth tries to solve the secret, which made her daughter fall silent. Here two young people suffering from autism communicate via 5- and 6-digit primes. After more than 1 hour the movie contains the following undecrypted two series of primes: 21, 383; 176, 081; 18, 199; 113, 933; 150, 377; 304, 523; 113, 933 193, 877; 737, 683; 117, 881; 193, 877 [Robinson1992] Directed by Phil Alden Robinson, Sneakers, Universal Pictures Film, 1992. In this movie the “sneakers”, computer experts under their boss Martin Bishop, try to get back the deciphering box SETEC from the “bad guys”. SETEC, invented by a genius mathematician before he was killed, allows to decrypt all codes from any nation. The code is not described in any way. [Brown1998] Dan Brown, Digital Fortress, E-Book, 1998. Dan Brown's first novel was published in 1998 as e-book, but it was largely unsuccessful then. The National Security Agency (NSA) uses a huge computer, which enables it to decrypt all messages (needless to say only of criminals and terrorists) within minutes even if they use the most modern encryption methods. An apostate employee invents an unbreakable code and his computer program Diabolus forces the super computer to do self destructing operations. The plot, where also the beautiful computer expert Susan Fletcher has a role, is rather predictable. The idea, that the NSA or another secret service is able to decrypt any code, is currently popular on several authors: In “Digital Fortress” the super computer has 3 million processors – nevertheless from todays sight this is by no means sufficient to hack modern ciphers. [Elsner1999] Dr. C. Elsner, The Dialogue of the Sisters, c't, 1999. In this short story, which is included in the CrypTool package as PDF file, the sisters confidentially communicate using a variant of RSA. They are residents of a madhouse being under permanent surveillance. [Stephenson1999] Neal Stephenson, Cryptonomicon, Harper, 1999. This very thick novel deals with cryptography both in WW2 and today. The two heroes from the 40ies are the excellent mathematician and cryptanalyst Lawrence Waterhouse, and the overeager and morphium addicted US marine Bobby Shaftoe. They both are members of the special allied unit 2702, which tries to hack the enemy’s communication codes and at the same time to hide the own existance. This secretiveness also happens in the present plot, where the grandchildren of the war heroes – the dedicated programmer Randy Waterhouse and the beautiful Amy Shaftoe – team up. Cryptonomicon is notably heavy for non-technical readers in parts. Several pages are spent explaining in detail some of the concepts behind cryptography. Stephenson added a detailled description of the Solitaire cipher (called “Pontifex” in the book), a paper and pencil encryption algorithm developed by Bruce Schneier. The used modern algorithm is not revealed. [Elsner2001] Dr. C. Elsner, The Chinese Labyrinth, c’t, 2001. In this short story, which is included in the CrypTool package as PDF file, Marco Polo has to solve problems from number theory within a competition to become a major consultant of the Great Khan. [Colfer2001] Eoin Colfer, Artemis Fowl, Viking, 2001. In this book for young people the 12 year old Artemis, a genius thief, gets a copy of the top secret “Book of the Elfs”. After he decrypted it with his computer, he finds out things, men never should have known. The used code is not described in detail or revealed. [Howard2001] Ron Howard, A Beautiful Mind, 2001. This is the film version of Sylvia Nasar’s biography of the game theorist John Nash. After the brilliant but asocial mathematician accepts secret work in cryptography, his life takes a turn to the nightmarish. His irresistible urge to solve problems becomes a danger for himself and his family. Nash is – within his belief – a most important hacker working for the government. Details of his way analysing code are not described in any way. [Apted2001] Directed by Michael Apted, Enigma, 2001. This is the film version of Robert Harris’ “historical fiction” Enigma (Hutchinson, London, 1995) about the World War II codebreaking work at Bletchley Park in early 1943, when the actual inventer of the analysis Alan Turing (after Polish pre-work) already was in the

US. So the fictional mathematician Tom Jericho is the lead character in this spy-thriller. Details of his way analysing the code are not described. [Isau2003] Ralf Isau, The Museum of the stolen memories (original title: Das Museum der gestohlenen Erinnerungen), Thienemann-Verlag, 2003. In this exciting novel the last part of the oracle can only be solved with the joined help of the computer community. [Brown2003] Dan Brown, The Da Vinci Code, Doubleday, 2003. The director of the Louvre is found murderered in his museum in front of a picture of Leonardo da Vinci. And the symbol researcher Robert Langdon is involved in a conspiracy. The plot mentions different classic codes (substitution like Caesar or Vigenere, as well as transposition and number codes). Also there are hints about Schneier and the sunflower. The second part of the book contains a lot of theologic considerations. This book has become one of the most widely read books of all time. [McBain2004] Scott McBain, Final Solution, manuscript not pubished by Harper Collins, 2004 (German version has been published in 2005). In a near future politicians, chiefs of military and secret services of many different countries take over all the power. With a giant computer network called “Mother” and complete surveillance they want to cement their power and commercialisation of life forever. Humans are only assessed according to their credit rating and globally acting companies elude of any democratic control. Within the thriller the obvious injustice, but also the realistic likelihood of this development are considered again and again. With the help of a cryptographer a code to destroy was built into the super computer “Mother”: In a race several people try to start the deactivation (Lars Pedersen, Oswald Plevy, the female American president, the British prime minister and an unknown Finish named Pia, who wants to take revenge for the death of her brother). On the opposite side a killing group acts under the special guidance of the British foreign minister and the boss of the CIA. [Burger2006] Wolfgang Burger, Heidelberg Lies (original title: Heidelberger L¨gen), Piper, 2006. u This detective story playing in the Rhein-Neckar area in Germany has several independant strands and local stories, but mainly it is about Kriminalrat Gerlach from Heidelberg. On page 207 f. the cryptographic reference for one strand is shortly explained: The soldier H¨rrle had copied circuit diagrams of a new digital NATO decryption device and the o murdered man had tried to sell his perceptions to China.

Further samples of cryptology in fictional literature can be found on the following German web page:

http://www.staff.uni-mainz.de/pommeren/Kryptologie99/Klassisch/1 Monoalph/ Literat.html For some older authors (e.g. Jules Verne, Karl May, Arthur Conan Doyle, Edgar Allen Poe) there are links to the original and relevant text pieces.

Kid books with collections of simpler cryptographic encryption methods, prepared in a didactical and exciting manner are in the following list (please send us similar English kid books, because at the moment our list contains only German kid books): [Mosesxxxx] [no named author], Top secret – The Book for Detectives and Spies (original title: Streng geheim – Das Buch f”ur Detektive und Agenten), Edition moses, [no year named]. This is a thin book for small kids with Inspector Fox and Dr. Chicken. [Para1988] Para, Ciphers (original title: Geheimschriften), Ravensburger Taschenbuch Verlag, 1988 (1st edition 1977). On 125 pages filled with a small font this mini format book explains many methods which young children can apply directly to encrypt or hide their messages. A little glossar and a short overview about the usage of encryption methods in history complete this little book. Right at page 6 it summarizes for beginners in an old fashion style “The Important Things First” about paper&pencil encryption (compare chapter 2): - “It must be possible to encrypt your messages at any place and at any location with the easiest measures and a small effort in a short time. - Your cipher must be easy to remember and easy to read for your partners. But strangers should not be able to decrypt them. Remember: Fastness before finesse, security before carelessness. - Your message must always be as short and precise as a telegram. Shortness outranks grammer and spelling. Get rid of all needless like salutations or punctuation marks. Preferably use only small or only capital letters.” [Kippenhahn2002] Rudolf Kippenhahn, Top secret! – How to encrypt messages and to hack codes (original title: Streng geheim! – Wie man Botschaften verschl¨sselt und Zahlencodes knackt), rororo, 2002. u In this novel a grandpa, an expert for secret writings teaches his four grandchildren and their friends, how to encrypt messages which nobody should read. Because there is someone who hacks their secrets, the grandpa has to teach them more and more complicated methods.

Within this story, which forms the framework, the most important classic encryption methods and its analysis are explained in a manner exciting and appropriate for children. [Flessner2004] Bernd Flessner, The Three Invesigators: Manual for Secret Messages (original title: Die 3 ???: Handbuch Geheimbotschaften), Kosmos, 2004. On 127 pages you learn in an easy and exciting manner, structered by the method types, which secret languages (like the one of the Navajo indians or dialects) and which secret writings (real encryption or hiding via technical or liguistic steganography) existed and how simple methods can be decrypted. The author tells where in history the methods were used and in which novells authors used encryption methods [like in Edgar Allan Poe’s “The Gold Bug”, like with Jules Verne’s hero Mathias Sandorf or like with Astrid Lindgren’s master detective Blomquist who used the ROR language (similar inserting ciphers are the spoon or the B language)]. This is a didactically excellent introduction for younger teens.

If you know of futher literature and movies, where cryptography has a major role or if you know of futher books, which address cryptography in a didactical and for children adequate way, then we would be very glad if you could send us the exact book titel and a short explanation about the book’s content. Thanks a lot.

Index