

Introduction

According to Merriam-Webster's Collegiate dictionary, the noun *cryptography* means 1: *secret writing*; and 2: *the enciphering and deciphering of messages in secret code or cipher*. The first definition is just the english translation of the etymology of the noun. The second definition has however become obsolete since Diffie and Hellman published their seminal paper *New directions in cryptography* [DH 1976], which opened up new possibilities. States, public and private organisations, and common people might not simply want to store and exchange data in a safe way: In a *wired* world, one might desire to *sign* a digital document in such a way that everybody is able to *verify* the resulting *signature* without however being able to counterfeit it.

In order to make this possible Diffie and Hellman introduced the notion of a *public key cryptosystem* but it was not until 1978 that the first such system, RSA, was introduced by Rivest, Shamir and Adleman [RSA 1978]. Diffie and Hellman also mentioned the idea of basing cryptosystems on the *discrete logarithm problem* in finite fields, an idea they attributed to Professor John Gill of Stanford University. One such cryptosystem was first proposed in 1985 by T. ElGamal [ELG 1985].

It is the task of mathematics to provide the so-called cryptographic *primitives* around which cryptosystems are developed. Using these primitives two parties can agree on a common key over an insecure channel, encrypt and decrypt messages, sign documents and verify signatures. *Cryptography* consists then in the design of these primitives and application of these to protocols which provide the aforementioned services. The evaluation of the strength of cryptographic methods is called *cryptanalysis*. Cryptography and cryptanalysis together form *cryptology*.

In symmetric systems each pair of parties uses one key both for encryption and for decryption, which has to be communicated in a safe way. On the contrary, in a public key system (also called an asymmetric system) each party has a key which consists of two parts: a public and a private one. Roughly speaking, a public key is used to encrypt information which can be decrypted only by means of the private key. Alternatively, the private key can be used to sign a document and the corresponding public key to verify the signature. Shared public keys can be used to establish a secret communication channel among two parties, for example, to deliver the key of a symmetric system – this is done because, in practice, symmetric systems allow higher throughput than asymmetric ones.

The security of public key systems relies on the one hand on the secrecy of the private key, and on the other hand on the impossibility to decrypt (or sign) without access to the private key. Especially it should be infeasible to derive the private key from its corresponding public key, whereas it must be easy to derive the public key from the private one, or else public key systems could not be realized at all. This can be reformulated as saying that public key systems involve functions which are *effectively* one-way: Such a function can be seen as a map F from the natural numbers \mathbb{N} to another set A such that

- F can be evaluated rapidly at every element of \mathbb{N} but
- for randomly chosen $y \in A$ it is a *hard problem* (in other words it is practically impossible) to compute $x \in \mathbb{N}$ with $F(x) = y$ without further information (i.e. the private key).

The set A is usually the set of elements of a suitable group or semigroup on which the operations are actually performed. So we have to find functions F together with algorithms to evaluate them with a proven small complexity, and we have to estimate the probability of finding the inverse images without help. The second task is done usually by proposing possible “attacks”, and in almost all cases it cannot be solved in a truly satisfactory way: What we can get are estimates from above for the complexity of solving the above mentioned *hard problem*, and estimates from below for certain attacks. This will lead us to reject some functions, and say that others are “to our best knowledge” still secure. This is a vague statement, and we would feel much better if we could use a *one way function* in the sense of information theory, or a *trap door one way function* (for the definition cf. [MOV 1996]). But since we even don’t know whether such functions exist we have to use what we have and should be extremely sensitive to new developments in mathematics and technology.

Problems assumed to be hard

Two problems have found widespread applications in public key cryptography so far: The *Integer factorization problem* i.e. find a non-trivial factor of a composite integer n and the *Discrete Logarithm Problem (DLP)* i.e. find an integer t with $g^t = h$ where g is a generator of a group (G, \cdot) and $h \in G$, see also Definition 1.1.2. Here we are mainly interested in the DLP.

In cryptosystems based on the first problem, the key size (in bits) is the base 2 logarithm of n . In the second case the key size depends on the way the elements of G are represented and numerated, but it is usually a linear function of the logarithm of the order of G . The integer factorization problem is not always hard, not even if n is large. Most randomly selected integers (say of some fixed large size) have many relatively small non-trivial factors that can quickly be found for instance by brute force methods. A hard instance of the integer factoring problem can however easily be generated just upon computing the product of two sufficiently large primes.

In DL systems we have to make sure that the order of the group G is good for cryptographical applications, because if its order is “bad”, i.e. the largest prime dividing it is small, the corresponding DLP is not hard (cf. Subsection 5.1.2). So we have to devise algorithms to find or to construct good groups. Usually one looks for a family of candidate groups and proves that (i) random members of that family are secure under known generic and specific attacks and that (ii) with a reasonable high probability we find members whose group orders are (nearly) prime. This all has to be done in a reasonable amount of time, and so computing the group order (see Chapters 5 and 8) is a crucial part of the key generation. Another possible approach consists in constructing groups of given order (for elliptic and hyperelliptic curves see Chapter 9).

One choice for the group to be used in a DL system is a suitably chosen subgroup of the multiplicative group of a finite field. It is easy to construct instances thereof in which the DLP is believed to be intractable, but the corresponding keys in some cases are considered quite large. For appropriately chosen subgroups compression methods based on traces such as LUC [SS 1995] and XTR [LV 2000] can be used to represent the subgroup elements.

Other groups where the DLP is considered difficult in general are the groups of points of elliptic curves and of jacobian varieties of hyperelliptic curves over a finite field [MIL 1986b, KOB 1987,

KOB 1989, KOB 1990]. These types of groups form the subject of this study. They have the advantage that less attacks are known for the DLP on them. For randomly chosen elliptic curves and for Jacobian varieties of hyperelliptic curves of genus 2 and 3 (in fact the curves or the fields of definition have not to be very special: cf. Chapters 5, 11 and 12) only generic attacks are known to work till today. This allows the implementor of a cryptosystem to achieve security matching or even surpassing that of other systems using at the same time smaller key sizes and, hopefully, getting better performance as well.

A proof that integer factorization is indeed a hard problem has never been published. The hardness of the DLP in black box groups has been proven ([NEC 1994, SHO 1997]; cf. Chapter 1). This is of considerable theoretical and practical interest since it shows that at least in principle DL-systems in groups with few additional properties are promising as cryptographic primitives and that there is no subexponential attack for all DLs at once. But in all special cases one has to analyze the situation with extreme care.

We are not concerned here with other types of supposedly hard problems, such as:

- The shortest vector problem and the closest vector problem in lattices [AJT 1996, COP 1997, AJT 1998, NS 2000].
- For suitable integer parameters $N > 0$ and $q > 0$, given a polynomial $h \in (\mathbb{Z}/q\mathbb{Z})[X]$ of degree $< N$, find polynomials $f, g \in (\mathbb{Z}/q\mathbb{Z})[X]$ of degrees $< N$ with $hf = g \pmod{X^N - 1}$. The encryption scheme NTRU [HPS 1998] and the signature scheme NSS are based on this problem [HPS 2001], where some strong restrictions are imposed on f, g and h . This problem can be interpreted as a lattice shortest vector problem [CS 1997].
- The conjugacy problem in infinite groups like the Braid Groups [AAG 1999, KL⁺ 2000].
- Other NP-complete problems [BRA 1979] such as Patarin's Hidden Field Equations (HFE) [PAT 1996a, PAT 1996b]. Cryptosystems based on the knapsack problem [SAH 1975, IK 1975] were very well received when they were created as an alternative to RSA. However, they have been broken since cryptologists were able to find a subset of knapsack problems that was easy to solve in polynomial time [ODL 1990]. The weakness was fixed, but the system was broken again by the same method. Since then cryptosystems based on NP-complete problems have been regarded with suspicion. However, HFE-based systems have not been broken at the time of this writing.

Structure of the report

The state of the art in the research on mathematical aspects of public key cryptology based on the discrete logarithm problem as of the beginning of 2002, forms the sole scope of the present report. Further we only review primitives based on geometric constructions, such as elliptic and hyperelliptic curves, and methods which presume the knowledge of the group order and its factorization, which are required by digital signatures.

The structure of the document is as follows:

Chapter 1 motivates the choice of groups as the basis for public key cryptosystems. Objects such as finite field elements, and thus also points on a curve defined over a finite field, are usually represented as a collection of integers satisfying some algebraic relations. Efficient finite field arithmetic is thus clearly of the utmost importance, and we deal with its algorithmic aspects

in Chapters 2 to 4, including the efficiency of integer arithmetic and general exponentiation algorithms. We then treat in Chapter 5 generic security considerations which apply to all DLP based cryptosystems. Chapters 6 and 7 are devoted to the arithmetic of elliptic and hyperelliptic curves respectively. In Chapter 8 we review the current known methods for counting points on arbitrary elliptic and hyperelliptic curves. In Chapter 9 we show how to construct curves with rational point groups of given order by the method of complex multiplication. Chapter 10 deals with transfers of the DLP from a group to another where the solution might be easier. The last two chapters are devoted to different aspects of the security of systems based on the DLP in elliptic curves and Jacobian varieties of hyperelliptic curves. Chapters 13 to 22 give a detailed overview of the various aspects of smart cards. Chapter 13 and 14 discuss the main components of a smart card and chapter 15 contains a discussion of the state of the art on attacks on smart cards. In chapter 16, we present an overview of the different types of memory embedded in smart cards and chapter 17 deals with the electrical properties such as supply voltage and current. Chapter 18 discusses the transmission protocols which are used to transmit information between the outside world, e.g. a terminal and the smart card and chapter 19 provides a description of some additional hardware such as interfaces. In chapter 20 we study fast arithmetic in hardware and highlight the differences with fast arithmetic in software. Chapter 21 contains an overview of the existing cryptographic coprocessors and chapter 22 discusses implementation aspects of elliptic and hyperelliptic curve cryptography in hardware. In chapter 23 we take a look at the most important standards in elliptic curve cryptography and chapter 24 finally gives an overview of the legal aspects of cryptography in the European Union and the United States.

Contributions of partners

In this section we give a brief overview of which partner is responsible for each chapter of the present document.

- Univeristy of Bordeaux I, Laboratoire A2X: C. Doche has written chapters 2, 3, 4, 6 (except subsection 6.3.2 on Koblitz curves) and the sections 8.2 and 8.3.
- Universität Essen, Institut für Experimentelle Mathematik: Dr. Roberto Avanzi, Prof. Dr. Dr.h.c. Gerhard Frey and Dr. Tanja Lange are responsible for the content and presentation of Chapters 1, 5, 7, sections 8.1 and 8.4 and chapters 9, 10, 11 and 12. Moreover, Chapter 9 contains material provided by Mr. Sebastian Leske and Dr. Annegret Weng.
- Oberthur Card Systems SA: B. Byramjee and B. Feix have written the chapters 13, 14 and 15.
- University of Bremen: W. Anheier, A. Weigl and R. Jährgig are responsible for chapters 16, 17, 18 and 19.
- Philips Semiconductors GmbH: K. Nguyen and S. van Rijnsouwou have written chapters 20, 21 and 22.
- Katholieke Universiteit Leuven: J. Scholten has written chapter 23 and F. Vercauteren is responsible for the editing of the document.
- Maya Software Technologies Ltd: G. Stern and J.L. Stehle delivered chapter 24.