# Public Key Cryptography, Number Theory and Applications

Preda Mihăilescu (University of Göttingen, Germany) and Michael Th. Rassias (ETH Zürich, Switzerland)

*In this article we review the advent and development of public key cryptography. The exposition is driven by the application aspect, while providing more details for certain issues in which beautiful number theory is involved.*

## 1 Introduction

In the early 1970s, the US army entertained the ARPA-net, an ancestor of the internet, and between 1972 and 1974 a number of universities on the East and West Coast were connected to this net for research and experimental purposes. The notion of remote computer-communication became tangible for the users of the net. It is conceivable that, under these conditions, the question of how to communicate in a secure way in a wide area net – like the ARPA – became an actual reality for those using the net. In fact, the concept of public-key cryptography, which gives the simple conceptual frame for algorithms successfully solving this problem, was born in Stanford from the joint work of W. Diffie and R. Hellman who studied public key infrastructures and R. Merkle who studied secret key distribution. Here is the way Diffie and Hellman presented the problem in [DH], a paper which mentions the joint work with Merkle: "In turn, such applications (fast computers) create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature."

### Public Key Cryptography arises

The idea was remarkably simple and efficient. Traditionally, a protected communication was established by using secret key cryptography. Two major characteristics of this craft, which still played an important role during the Second World War, are that the sender and receiver must agree upon a common secret key prior to communication and that it was a widespread conception that keeping the encryption method secret was a sensible method for increasing security. In a wide area communication network, in which numerous peers can communicate over large distances, the chance of establishing a common secret key prior to communication are low. This was more than just an abstract realisation of the inventors of public key cryptography; it was a fact of which all users of the ARPA-net were aware.

The new concepts are widely known and are currently taught in school. We will briefly review them in order to introduce notation which will be useful when discussing some interesting instances, allowing us to bring in more mathematics. If $X$ is any peer who wants to engage in secure network communication, he should start by generating a set of data, which is bundled into his own *secret key* $S_X$. A subset of this data, bundled in the *public key* $P_X$ will be made public to all peers with whom he might wish to communicate. The two keys should have the following two properties:

1. Both keys can be used for encrypting texts according to some algorithm yet to be defined; messages encrypted by $S_X$ can be decrypted by $P_X$ and vice versa. Moreover, the keys should be sufficiently random: the chance of two peers accidentally generating the same secret key should be close to zero.
2. It should be computationally unfeasible to derive $S_X$ from $P_X$.

On the basis of these premises, if two peers A and B – cryptographers like using names so these peers are often called Alice and Bob – want to communicate, then Alice sends to Bob messages encrypted by $P_B$, which she may retrieve from the public key repository. However, only Bob can decrypt the message so the communication is secure. On the basis of this idea, a further application emerged: it is often useful to be able to certify the ownership of some message, to *sign* the message in a unique and non-repudiable way. In this case, secrecy is less of a concern than ownership. The solution consists of associating a short cryptographic *hash-value* $H$ to the message, which is encrypted by the secret key $S_A$. Any receiver will then be able to regenerate the hash value on their own, decrypt the encrypted hash with $P_A$ and then compare the two results. If they match, Bob has a proof that it was Alice who has sent the message. It was in the same context that the paradigm was set that security of encryption algorithms is increased by their becoming public and not the contrary, as had been previously perceived. The reasoning behind this is that algorithms known to the scientific community would be well analysed and the process of academic scrutiny would help select the most secure and efficient algorithms. *Security lays solely in the secrecy of the keys* is the slogan of this paradigm. In fact this sensible point of view quickly spread within the scientific community. Public key cryptography changed not only the concept of secret communication but also the way of perceiving security: an algorithm is more reliable when it has long resisted public scrutiny by the cryptographic community and not when it is based on sophisticated "secret tricks". Within the next 20 years this point of view probably reached most of the banks. In the late '90s, manufacturers of cryptographic hardware had only a precious few customers insisting on the "privilege" of purchasing machines which ran according to some unique and "secret" algorithm.

### The invention of the classical systems: DH and RSA

In the next two years after the abstract definition of public key cryptography, two major algorithms that implement this idea and are still in use today were invented. The first is based on the difficulty in solving the *discrete logarithm* problem in the multiplicative group of finite fields and it was proposed

by Diffie, Hellman and Merkle, the inventors of public key cryptography. The algorithm was initially meant to serve for a variant of the public key cryptography idea, in which Alice and Bob only wish to establish a shared secret key – they do not wish to encrypt with the same algorithm; rather, they will proceed by using the secret key for some classical, faster secret key encryption scheme. Incidentally this two-step approach to encryption is the core idea in the TSL protocol, developed between 1992 and 2002 and currently used in all confidential https communications on the internet – for instance when you book a flight or buy a book from Amazon.

If $\mathbb{F}_q$ is some large finite field and $g \in \mathbb{F}_q^\times$ is a generator of the multiplicative group of the field – both being public data – then Alice and Bob start by choosing some random one time keys $A_R, B_R$ which are elements of $\mathbb{Z}/((q-1)\cdot\mathbb{Z})$. Then Alice sends to Bob $M_A = g^{A_R}$ and receives from Bob $M_B = g^{B_R}$. The reader can verify that by using the private data and the data received, both Alice and Bob may retrieve $S = g^{A_R \cdot B_R}$, which is the data from which the common secret key is extracted. However, an eavesdropper, who is always called Eve[1] in cryptography, would only know $g^{A_R}$ and $g^{B_R}$ but not $A_R$ or $B_R$. The problem of recovering these secret data from the ones communicated on the net is the discrete logarithm problem in finite fields, which is known to be a hard problem.[2] It is however not hard in the strong sense of complexity theory, since it is not known to reduce to any NP complete problem; the same holds in fact for the problem of factoring integers. The procedure is widely known as the *Diffie-Hellman key exchange algorithm* and it does not provide a direct solution to the problem of public key encryption/decryption and of signatures.

This was provided one year later, in 1977, by R. Rivest, A. Shamir and L. Adleman at MIT. Their algorithm, widely known as RSA after the initials of their names, uses the problem of factoring integers as the problem intended to prevent recovery of the secret key from the public ones. A secret key consists of $S_A = \{p, q, d\}$, where $p, q$ are two large primes satisfying some additional randomness conditions and

$$0 < d < (p-1)(q-1), \quad \text{with} \quad (d, pq(p-1)(q-1)) = 1$$

is a random number; if

$$e \in \mathbb{N} \text{ such that } ed \equiv 1 \bmod (p-1)(q-1),$$

the public key consists only of $P_A = (n, e)$, with $n = p \cdot q$. In some instances $e$ is a fixed number for the whole system, so $d$ will be determined by the holder of the secret key using the same defining congruence. With these prerequisites, if $M$ is a short message, it will be identified with a number in $\mathbb{Z}/(n\cdot\mathbb{Z})$ and its public key encryption $M_e \equiv M^e \bmod n$ can be computed in the open but can only be decrypted by Alice, the holder of $d$, since $M \equiv M_e^d = M^{ed} \bmod n$. Conversely, if Alice encrypts $M$ with $d$, then anyone can recover $M$ and upon doing so will have proof of Alice having produced the encryption; indeed, only the owner of the secret key could produce this encryption, which can thus act as a private signature of Alice.

In 1978, Hellman and Merkle invented a public key cryptosystem that did not rely on number theory but rather on the NP-complete *knapsack* problem. In 1983, J. L. Massey and J. K. Omura developed a variant of the Diffie-Hellman key exchange, which works as a public key encryption scheme too. It has the advantage that, like for the key exchange, a single public key can be used by a whole *domain* – the public key consists of a large prime $p$ and a generator $g \in \mathbb{F}_p^\times$.

Despite initial attempts of the NSA to inhibit the publicising of the ideas of public key encryption and RSA, these had already been brought to public perception in 1977 by Martin Gardner in his widely read column Mathematical Games in the Scientific American magazine and were eventually published in the communications of the ACM [RSA]: the way to public key cryptography was open! While the NSA tried to stop the diffusion of the public key idea, it turned out that the idea had already been invented in 1970 by a researcher for MI6's General Communication Head-Quarters GCHQ, who had also discovered the DH and RSA algorithms in 1971 and 1972 … but in the reverse order, with RSA discovered first. The information was declassified and made public in 2000 and the director of RSA's research team Dr Burt Kaliski confirmed the truth of the information. This news was accepted with some reticence by the community, which speculated upon an a posteriori wish for academic acknowledgement from a person who had accepted working in secrecy. However, both the idea and the most widely spread instances of public key cryptography have the making of good mathematical work - they were discovered independently by people working on the same question.

## 2 Cryptanalysis

The first major success of public key cryptography was that the expectation came true and the domain of cryptanalysis – concerned with the analysis of possible attacks against cryptographic schemes – became a flourishing academic domain of investigation. One of the most spectacular successes was due to the development of the *lattice reduction* algorithm by A. Lenstra, H. Lenstra Jr. and L. Lòvasz, the LLL-algorithm. Given a lattice $\mathcal{L} \subset \mathbb{Z}^n$, there exists a base consisting of shortest vectors. Classical algorithms for finding such a base are known from the work of Charles Hermite. Only, in the case when the base is presented by an initial generating system of very large vectors, the process is exponential. The algorithm was developed from techniques used by Lòvasz in integer programming; the idea was to use an approximate Gram-Schmidt-reduction which provides some *close* to minimal vectors in $\mathcal{L}$. The advantage is that the algorithm runs in polynomial time and has therefore a wide variety of applications both in cryptography and in number theory itself. One of the first applications of LLL was in showing that the keys of the knapsack cryptosystem could be cracked in polynomial time: in order to do that, one had only to solve a particularly simple *subfamily* of problems belonging to the knapsack family. This result showed the advantage of public academic scrutiny of cryptographic schemes, since it had only taken five years to reveal the weaknesses of one of them, but it also blocked the way for applications of the knapsack. Some improved versions have been presented that could never be attacked – but they never made it to public applications.

The most important effect of cryptanalysis was less visible. The community quickly developed its own language and defined a variety of subtle *attack scenarios*, in which the eavesdropper *Eve* was offered increasing levels of advantages:

Eve can simply tap a wire communication or she might also collect large amounts of data signed by Alice or even induce her into signing a chosen suite of messages. Later, the encryption hardware began being regarded as a point of attack, as it was observed that physical measurements on a chip while it is computing an RSA encryption may reveal some bits of the secret key. In this way, well defined attack scenarios are used for checking the security of various cryptosystems and protocols.

This is a good place to mention the fact that cryptography advances with a tension between the needs for security and for efficiency. It has happened repeatedly that the latter has led to the usage of particular constellations of keys that allow for faster computations. But eventually, when simplification has reached too far, an attack has been discovered; certain keys, or even whole cryptographic schemes, are then discarded. It is for instance useful to have a universal, short public exponent $e$ for the RSA scheme and this had also been used in practice in the late 1980s. But D. Coppersmith showed that if $e$ is too small, it is easy to gather sufficiently many messages signed by the same key $S_A$ and then use simple arithmetic in order to crack that key. Therefore, the smallest fixed key currently allowed by standards is $e = 2^{16} + 1$ and this may change with the growth of computing and storage capacities.

It is easy to verify that an efficient method for factoring integers breaks the RSA scheme and a solution to the discrete logarithm problem breaks the DH key exchange and the Massey-Omura cryptosystem. On the other hand, there is no proof either for the expectation that efficient attacks on RSA are equivalent (in a complexity theory understanding) to factoring integers, or that breaking the DH scheme is equivalent to solving the discrete logarithm problem. In fact various scenarios of cryptanalysis investigate conditions under which successful attacks can be completed *without* solving the underlying hard problems.

The closest mathematical problem is the *DH*-problem: given a prime $p$ and a generator $g \in \mathbb{F}_p^\times$, let

$$A = g^a, B = g^b, C = g^{ab} \in \mathbb{F}_p$$

be given for unknown $a, b \in \mathbb{Z}/((p-1)\cdot\mathbb{Z})$; find the values of $a$ and $b$. It is used for theoretic analysis of provable security of cryptographic schemes, the investigation of which was initiated by Maurer [Ma] and established by Shoup and various coauthors, e.g., in [CS].

### Dickman's Theorem and its impact

In the 1930s, J. Dickson considered the question of estimating the largest prime factors of some random integer $n$. Using heuristic estimates on the repartition of primes he found for instance that if $p|n$ is the largest prime dividing $n$ then $p = O(n^{\ln 2})$. More generally, an integer $n > 1$ is $y$-smooth if none of its prime factors exceeds $y$. The function

$$\psi(x, y) = \sharp\{\ 1 \le n \le x\ :\ n \text{ is } y\text{ - smooth }\}$$

counts the smooth numbers less than $x$. Dickman also proved that for all $u > 0$ there is a real number $\rho(u)$ such that

$$\psi(x, x^{1/u}) \sim \rho(u)x.$$

The function $\rho(u)$ was described in terms of a differential equation, in which $u$ was fixed for $x \to \infty$. Half a century later, the gap was filled by Canfield, Erdős and Pomerance [CEP], who proved that

**Theorem 1** (Canfield, Erdős and Pomerance)**.**

$$\begin{aligned} \psi(x, x^{1/u}) &= xu^{-u+o(u)}, \\ &\quad \textit{uniformly, for} \\ u &< (1-\epsilon)\ln x/\ln\ln x \textit{ when } u \to \infty. \end{aligned} \tag{1}$$

This theorem is at the heart of all the state-of-the-art algorithms for factoring integers and discrete logarithm in multiplicative groups. The classical application to factoring integers is the *quadratic sieve method* and it has its origin in the following simple observation of Fermat: if $m$ is a composite integer then the congruence $x^2 \equiv c \bmod m$ will have at least four solutions and there are $x, y$ such that $x \not\equiv \pm y \bmod m$ but $x^2 \equiv y^2 \bmod m$. Then $(x + y, m)$ is a nontrivial factor of $m$. Theorem 1 helps find such pairs $x, y$, as follows: for numbers $x(i)$ in some interval $\lceil \sqrt{n} \rceil + i, 0 \le i \le B$, one computes the remainder[3]

$$r(i) = x(i)^2 \ \text{rem}\ m$$

and retains only those values of $x$ for which $r$ is a $B$-smooth number. By choosing $B$ accordingly, it is possible to factorise $r(i)$ efficiently. After gathering sufficiently many such relations, one may hope that the product of some $r(i)$ is a square: namely, that there is an index subset $J \subset [0, B]$ such that

$$\prod_{i \in J} r(i) = R^2,\ R \in \mathbb{Z}.$$

Then letting $X = \prod_{i \in J} x(i)$, we obtain the congruence $X^2 \equiv R^2 \bmod m$. If, in addition, $X \not\equiv \pm R \bmod m$, which should happen with probability $\ge 1/2$, then $(X \pm R, m)$ is a nontrivial factor of $m$. The method relies on some empirical assumptions on the repartition of factors of $r(i)$: namely, that the distribution of these residues is such that one may apply the relation (1) for estimating the probability that one of these numbers is $B$-smooth. These allow one to establish an *optimal* bound

$$B \sim \exp(\sqrt{\ln(m)\ln\ln(m)}).$$

This and similar functions occur often in algorithms using smoothness, so it received a name:

$$L(n; a) := \exp\left(\ln(n)^a \ln\ln(n)^{1-a}\right).$$

In our case $B = L(n; 1/2)$ and the quadratic sieve runs in time polynomial in $B$ – experience having so far confirmed the underlying heuristical assumptions. The following nice example is taken from the book of R. Crandall and C. Pomerance [CP]: let $m = 1649$, with $41 = \lceil \sqrt{m} \rceil$. We find

$$41^2 \equiv 32 \bmod m; \quad 42^2 \equiv 115 \bmod m; \quad 43^2 \equiv 200 \bmod m.$$

Since $32{\cdot}200 = 2^{5+3}{\cdot}5^2 = 80^2$, we let $R = 80$ and $X = 41{\cdot}43 = 42^2 - 1 \equiv 114 \bmod m$, finding that $114^2 \equiv 80^2 \bmod m$ and eventually $17 = (114 - 80, 1649)$, which is a nontrivial factor.

For the discrete logarithm problem in $\mathbb{F}_p^\times$, which consists of determining $x$ such that $g^x \equiv b \bmod p$, one uses smooth numbers as follows. Fix a smoothness bound $y$ and let $q_1, \ldots, q_r < y$ be all the primes up to $y$. For random values of $m$, one computes $u = g^m \ \text{rem}\ p$ and keeps only those values of $u$ which are $y$-smooth. After collecting sufficiently many relations, one will then be able to compute the discrete logarithms $l_i$ such that $q_i \equiv g^{l_i} \bmod p$. Next, one tries random values of $k$ searching such ones that make $v = bg^{-k} \ \text{rem}\ p$ be a $y$-smooth number. The precomputed values $l_i$ will then help determine $x = k + \log_g(v)$ from the prime decomposition of

the $v$. This algorithm also relies on heuristic assumptions, on the basis of which the running time is $L(p, 1/2)^{\sqrt{2}}$.

At the end of the 1980s, John Pollard found a way of applying the idea of the quadratic sieve to integers in number fields rather than $\mathbb{Q}$. The method was first applied to the factorisation of the Fermat number $F_9 = 2^{2^9} + 1$. In the following years, it was generalised and improved by a series of mathematicians, starting with A. Lenstra and M. Manasse. The resulting number field sieve is currently the asymptotically fastest factoring method and it runs in time $O(L[n; 1/3])$. Similar methods are known for the discrete logarithm method: they use number fields in case of larger characteristics and function fields for small characteristics. Like in the case of factoring, their running time is also $O(L[n; 1/3])$. Current records reach as high as 7–800 binary digits for factoring composites of general form and ~ 5–600 for the discrete logarithm in prime fields.

## 3 Elliptic curves

In 1984, René Schoof opened the way for discovery of a polynomial time algorithm for counting the number of points on an elliptic curve over a finite field. This brought the groups of algebraic geometry to the realm of applications and algorithms. Within one year, H. W. Lenstra Jr. proposed an important variant of Pollard's rho-method for factoring, based on elliptic curves: the *ellptic curve method* or ECM. Also, V. Miller and N. Koblitz proposed independently the use of elliptic curves for cryptography. The ECM method has a runtime comparable to the quadratic sieve but it behaves particularly well for numbers $m$ which have some small prime factors (i.e. sensibly smaller than $\sqrt{m}$: the runtime is estimated to be $L(p; 1/2)^{\sqrt{2}}$, where $p$ is the smallest prime dividing $m$).

### Counting points

The idea of Schoof is both elegant and important, beyond even the immediate algorithmic and cryptographic applications: it led to an area of research for practical algorithms for counting points on finite varieties. This research area is still growing, while the main domain of application has ceased to be cryptography for at least a decade. The algorithms are more and more used for larger computations related to mathematical questions such as the Birch Swinnerton-Dyer conjecture and other properties of $L$-series. See also [Ra] for an interesting elementary theoretical application of point counting.

Initially, Schoof [Sc1] started from the following simple remark: if $E_p(a, b) : Y^2 = X^3 + aX + b$ is an elliptic curve defined over the finite field $\mathbb{F}_p$, of which one assumes that it is ordinary, then Riemann's conjecture for elliptic curves implies that, in $\mathrm{End}(E_p, \overline{\mathbb{F}}_p)$, the Frobenius verifies the quadratic equation

$$\Phi^2 - t\Phi + p = 0. \tag{2}$$

Since $E_p$ is fixed by $\Phi$, we have $|E_p(a, b)| = p - t + 1$. Counting the points is thus equivalent to determining the value of the *trace of the Frobenius* $t$; since the Hasse inequality states that $t < 2\sqrt{p}$, it suffices to determine the remainder $t \mod \ell$ for some small primes with

$$L = \prod \ell > 2\sqrt{p}.$$

Therefore, the core step of the algorithm consists of modelling the $\ell$-torsion $E_p[\ell]$ into an algebra

$$\mathbb{B} = \mathbb{F}_p[X, Y]/\left(\psi_\ell(X),\ Y^2 - (X^3 + aX + b)\right),$$
$$P = \left(X + (\psi_\ell(X)),\ Y + (Y^2 - (X^3 + aX + b))\right) \in \mathbb{B},$$

in which $\psi_\ell(X)$ is the $\ell$-division polynomial which has as roots all the $x$-coordinates of $\ell$-division points. Therefore, any such point enjoys the properties which define the *generic* $\ell$-torsion point $P \in \mathbb{B}$. It is then a straightforward computation to determine $t \mod \ell$ from the identity

$$\Phi^2 P + pP = t\Phi P.$$

The seminal idea of Schoof to determine the parameters of the Riemann $\zeta$-function from projections in torsion spaces, and thus counting points on varieties over finite fields, was both improved for simple varieties, such as elliptic curves, and extended to more general abelian varieties. In the first case, the primary thing to do was to reduce the size of the algebra $\mathbb{B}$ – which can be done by finding smaller factors of $\psi_\ell(X) \bmod p$.

The breakthrough in this direction was indicated by Noam Elkies (see [Sc2]), who brought in modular forms, thus showing how to find in half of the cases some factors $f(X)|\psi_\ell(X)$ of linear degree, compared to the quadratic degree in $\ell$ of the division polynomial. The $\ell$-torsion $E_p[\ell] \cong \mathbb{F}_\ell^2$ as a vector space, fixing two linear independent points $P, Q \in E_p[\ell]$, we see that $G := \mathrm{Gal}\,(\mathbb{B}/\mathbb{F}_p)$ acts on the vector space $E_p[\ell]$ by acting on the base $P, Q$. We obtain a representation $\rho : G \to \mathrm{GL}_2(\mathbb{F}_\ell)$, with respect to which $\rho(\Phi)$ verifies the same quadratic equation. If $\delta$ is the discriminant of this equation, according to the value of the Legendre symbol $\left(\frac{\delta}{\ell}\right) \in \{1, 0, -1\}$, the matrix $\rho(\Phi)$ is diagonalisable, has normal upper triangular form or has eigenvalues in $\mathbb{F}_{\ell^2}$. In the first case, there are two *eigenpoints* $P, Q$ of the Frobenius and the orbit of their $x$ coordinates under multiplication on the curve is galois invariant. We obtain the *eigenpolynomials*

$$f_P(X) = \prod_{k=1}^{(\ell-1)/2} (X - ([k]P)_x) \mid \psi_\ell(X),$$

where

$$\deg(f_P) = (\ell - 1)/2 \quad \text{and} \quad \deg(\psi_\ell) = (\ell^2 - 1)/2,$$

together with a new algebra $\mathbb{B}'$, obtained by replacing $\psi_\ell$ with $f_P$. For the computation of $F_P$, Elkies considered the function field $\mathbb{C}[[j(q)]]$. Some classical arguments on Eisenstein series and $\Gamma_0(\ell)$-modular forms imply that for each $j$-invariant $j_m$ of an $\ell$-isogenous curve to $E_p$ – or, also, for each zero of the modular equation $\Phi_\ell(X, j(q))$ – there is a polynomial $f_j(X) \in C[[j(q)]][X]$ which has the $x$-coordinates of the kernel of the respective isogeny as zeroes. The polynomials can be constructed in the function field by manipulations of $q$-expansions and they have the useful property that all the coefficients are algebraic integers. The insight of Elkies was to show that one can substitute for $j_m$ the value of some zero $\Phi_\ell(X, j(E_p)) \bmod p$ and reduce the coefficients of $f_j(X)$ modulo $p$, thus obtaining some eigenpolynomial corresponding to the value of $j_m$. Indeed, if $\mathbf{E}$ is any curve over $\overline{\mathbb{Q}}$ which reduces to $E_p$ at some prime ideal above $p$ then its $j$-invariant reduces to the one of $E_p$ and so do the invariants

of its $\ell$-isogenies. Therefore, if the modular equation has linear factors $j_m$ over $\mathbb{F}_p$, by inserting these in the expression for $f_j(X)$, upon reduction at the same prime, the coefficients of the polynomial $f_j$ map to the ones of some eigenpolynomial. Using improved algorithms for manipulation of series [BMSS], one can compute the eigenpolynomials in time $O(\log^3(p))$, the running time being dominated by the computation of zeroes of $\Phi_\ell(X, j(E_p)) \bmod p$. Further improvements can be achieved by using the galois structure of the resulting algebras [MV].

For curves defined over finite fields of small characteristic $p$, it is possible to project to $p^N$-torsion. Using different flavours of cohomology combined with Newton iterations, various authors starting with T. Satoh, K. Kedlaya and A. Lauder have developed the most efficient point counting algorithms for elliptic curves. Some of them are generalised to super elliptic curves, elliptic surfaces, etc.

### Cryptography

The elliptic curve based cryptographic schemes which have survived scrutiny and become part of current standards on public key cryptography are essentially variants of the Diffie-Hellman key exchange scheme and are based on the difficulty of solving the discrete logarithm problem: find $x$ such that

$$[x]P = Q, \quad \text{for} \quad P, Q \in E_p(a, b)$$

being points on an elliptic curve, such that $Q$ is known to generate a cyclic group of high order. Unlike in finite fields, the DL problem on elliptic curves is not known to allow any sub-exponential time solutions. The best known methods have runtime $O(\sqrt{p})$, where $p$ is the characteristic of the (prime) field over which $E_p$ is defined. As a consequence, one can work in much smaller groups than in the case of the multiplicative groups of finite fields, still achieving the same estimated security of a scheme, with respect to state-of-the-art attacks. This advantage led to a new wave of interest for elliptic curve cryptography in connection with security of mobile phones. Over the last two decades, the evolution in this direction oscillated between the search of special curves with most efficient group operations[4] and the care required to maintain security, when using special cases. The core problem in this respect is the reduction to the discrete logarithm in finite fields, using Weil pairings and descent methods. The use of the Weil pairing for the discrete logarithm on some special elliptic curves was pointed out for the first time by Gerhard Frey. The problem came to light when Frey was asked to estimate software using special curves with fast group operation for its security. He was able to show that for the specific curves, the Weil pairing reduced the elliptic curve logarithm problem to one in finite fields of critically small size. The idea was taken over by A. J. Menezes, P. C. van Oorschot and S. A. Vanstone and is currently known in the literature under the name of *MOV attack*. Much experience has been gathered in this field yet, even nowadays, the attraction of efficient arithmetic for mobile phones leads to the use of special curves – e.g. the so-called Koblitz curves, defined over fields $\mathbb{F}_p$ of small characteristic and having $a, b \in \mathbb{F}_p$ – which are close to the critical area where one may expect that reduction to some feasible discrete logarithm in finite fields is just one idea away. Despite standardisation, which made cryptographic developments obsolete on the internet, there are thus reasons why research in this particular area is still very fertile. We recommend the detailed and lively survey of Heß et al. [HeSSL].

## 4 Quantum cryptography and other interesting applications

The main intensively used public key cryptography methods rely on the number theoretic problems described above. There have been numerous interesting attempts to use the large list of NP complete problems in order to derive some trapdoor function (the one allowing the concealment of the secret key behind the public ones) – the knapsack problem is only one of the most famous ones. We can hardly go into the detail necessary in order to pay justice both to the interest of the attempts and the reasons for their failure or restricted use.

One cannot complete even a brief survey of public key cryptography without mentioning several examples which have withstood the test of time and do have either a theoretical interest or even a practical niche of application. Since the early 1980s the Canadian mathematicians G. Brassard and C. Crépeau have suggested the use of quantum effects for security applications: the simple idea was that Eve could not tap a quantum communication wire without destroying the information content transmitted, so security would be provided by a *self-destruction mechanism* introduced by quantum mechanics in the confidential information transmitted. The physical and cryptographical aspects of the idea have been in active research ever since and, in the first decade of this century, several practical implementations of quantum[5] cryptography have been announced, reaching over distances of up to 100 km.

Public key cryptography is much slower than secret key encryption – by at least a factor of 1000, as a rule of thumb. This has led to the wish to design some fast asymmetric schemes, even at the cost of, say, very large keys. A successful solution in this respect was invented by three number theorists: J. Hoffstein, J. Pipher and J. H. Silverman, who called their method "Number Theorists aRe Us": NTRU [HPS]. The method found a niche of applications and has the intriguing property that lattice reductions are used both for legitimate decryptions as well as for attacks. The asymmetry is thus not one between polynomial and exponential algorithms but is rather due to the high dimensions of lattices in which reduction must happen for the attacker.

A further family of interesting public key schemes uses non commutative groups – such as, for instance, *braid groups* (e.g., see [MSU]). Their developers make a point out of the fact that, if one day quantum computing is feasible, all number theoretic schemes can be broken in short polynomial time. However, using the current models of quantum computing, no attack to braid group cryptography is known.

## Notes

1. This is one of the few instances of which we are aware in which a politically incorrect denomination introduced in the '70s, was neither revindicated nor changed up to the present. The reason

may possibly lay in the fact that political correctness lacks fantasy, so people can hardly imagine Adam offering the apple to Eve.

2. It has been neither proved nor disproved that solving the discrete logarithm problem is the only way for Eve to recover the secret key $S$. Since this is an independent problem of interest on its own, it has received more recently a name: the DH - Problem, after Diffie-Hellman.
3. In computational algebra, the notation $x$ rem $y$ stands for the unique representative of the equivalence class of $x$ mod $y$ which lays in the interval $[0, y)$.
4. On the one hand, the base fields are much smaller – 200 bits for the size of the base fields is still a very safe bound, which compares well, in terms of security, to RSA moduli of over 1000 bits – but on the other hand additions on a curve require more operations than a mere multiplication in $\mathbb{Z}/(n \cdot \mathbb{Z})$. This explains the reason for the efficiency struggle.
5. The reader should not confuse *quantum cryptography* with *quantum computing*, where quantum effects are used to help computations and not only secure information transmission: the physical challenges are even larger in the latter case.

## Bibliography


[BMSS] A. Bostan, F. Morain, B. Salvy and É Schost: *Fast algorithms for computing isogenies between elliptic curves*, Math. Comp. **77** (2008), 1755–1778.

[CEP] E. R. Canfield, P. Erdős, C. Pomerance, *On a problem of Oppenheim concerning Factorisatio Numerorum*, J. Number Theory 17 (1983) 1–28.

[CS] R. Cramer and V. Shoup, *Signature Schemes based on strong RSA assumptions*, Extended abstract in Proc. ACM CCS 1999.

[CP] R. Crandall and C. Pomerance, *Prime Numbers – A Computational Perspective*, Springer, 2004.

[DH] Whitfield Diffie and Martin Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory; Nov. 1976.

[EL] N. D. Elkies, *Elliptic and modular curves over finite fields and related computational issues*, Computational Perspectives on Number Theory: Proc. Conf. in honour of A. O. L. Atkin (D. A. Buell and J. T. Teitelbaum, eds.), AMS/International Press, 1998, 21–76.

[HeSSL] F. Heß, A. Stein, S. Stein and M. Lochter, *The Magic of Elliptic Curves and Public Key Cryptography*, Jahresbericht Deutsch Math.-Ver. **1**14 (2012), 59–88.

[HPS] J. Hoffstein, J. Pipher and J.H. Silverman: *An Introduction to Mathematical Cryptography*, Springer (2008)

[Kb1] N. Koblitz, *Elliptic curve cryptosystems*, Math. Comp., **48**(1987), 203–209.

[Kb2] N. Koblitz, *Course in Number Theory and Cryptography*, Springer-Verlag, New York, 1994.

[Len] H. W. Lenstra, *Factoring integers with elliptic curves*, Annals Math., **126**(3)(1987), 649–673.

[Ma] U. Maurer: *Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms*. Advances in Cryptology – Crypto '94, Springer-Verlag, (1994), 271–281.

[MV] P. Mihăilescu and V. Vuletescu, *Elliptic Gauss sums and applications to point counting.* J. Symb. Comput. 45, **8**(2010), 825–836.

[ML] V. Miller, *Uses of elliptic curves in cryptography*, Advances in Cryptology: Proc. of Crypto '85, Lecture Notes in Computer Science, **218**(1986), Springer-Verlag, New York, pp. 417–426.

[MSU] A. Myasnikov, V. Shpilrain and A. Ushakov, *Group-based Cryptography*, Advanced Courses in Math. CRM Barcelona, Birkhäuser Verlag (2008).

[Ra] M. Th. Rassias, *On the representation of the number of integral points of an elliptic curve modulo a prime number*, http://arxiv.org/abs/1210.1439.

[RSA] R. Rivest, A. Shamir and L. Adleman, *A method for obtaining signatures and public key cryptography*, Communications of the ACM, **21** (1978), 121–126.

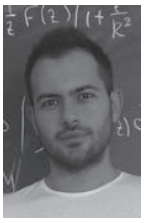
[Sc1] R. Schoof, *Elliptic Curves over Finite Fields and Computation of Square Roots mod p*, Math. Comp. **4**3(1985), 483–494.

[Sc2] R. Schoof, *Counting Point on Elliptic Curves over Finite Fields*, Journal de Th. des Nombres Bordeaux,

[Sil] J. Silverman, *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics **106**, Springer-Verlag, New York, 1986.

[Was] L. C. Washington, *Elliptic Curves-Number Theory and Cryptography*, CRC Press, London, New York, 2008.


*Preda Mihăilescu [preda@uni-math.gwdg.de] was born in Bucharest, 1955. He studied mathematics and computer science in Zürich receiving his PhD from ETH-Zürich. He was active during 15 years in the industry, as a numerical analyst and cryptography specialist. In 2002, Mihăilescu proved Catalan's conjecture. This number theoretical conjecture, formulated by the French mathematician E. C. Catalan in 1844, had stood unresolved for over a century. The result is known as Mihăilescu's Theorem. He is currently a professor at the Institute of Mathematics of the University of Göttingen.*



*Michael Th. Rassias [michail.rassias@math.ethz.ch] was born in Athens, 1987. He received his school education in Athens, during which he participated at various mathematical contests; he has received two gold medals at the Pan-Hellenic Mathematical Olympiads of 2002 and 2003, a silver medal at the Balkan Mathematical Olympiad of 2002 and a silver medal at the 44th International Mathematical Olympiad of 2003. He holds a Diploma from the School of Electrical and Computer Engineering of the National Technical University of Athens and a Master of Advanced Study in Mathematics from the University of Cambridge. He is currently a PhD student in Mathematics at ETH-Zürich, under the supervision of Professor E. Kowalski.*