Lectures on Cryptography
Heraklion 2003

Gerhard Frey
IEM, University of Essen

Part I

1 General cryptographical background

1.1 Cryptographical methods

We discussed

- The tasks and aims of cryptography
- Communication scheme (two parties)
- Symmetric systems
- Some examples (permutations)
- Randomization
- Secure channels replaced by Public Key
- Principle of Public Key
- Advantages and disadvantages
- Hybrid systems
- Packages with encrypted symmetric keys and signatures

1.2 Key exchange (abstract)

Assume that $A \subset \mathbb{N}$ is finite and that $B \subset End_{set}(A)$.
Assume that the elements of B commute: For all a and $b_1, b_2 \in B$ we have

$$b_1(b_2(a)) = b_2(b_1(a)).$$

Then we can use

$$A, B$$

for a key exchange system in an obvious way - using (publicly known) base points in B-orbits of A.

Note:
The private keys are elements in B, the common secret is an element in A, the parameters are B and a chosen base point in a $B-$orbit of A.

The security depends (not only) on the complexity to find from the knowledge of randomly chosen $a \in A$ and given a_1, a_2 in $B \circ \{a\}$ **all** elements $b \in B$ with $b(a) = a_1$ modulo

$$Fix_B(a_2) = \{b \in B; b(a_2) = a_2\}.$$

The efficiency depends on the "size" of elements in A, B and on the complexity of evaluating $b \in B$.

1.3 Signature of ElGamal type (abstract)

Again we assume that $B \subset End_{set}(A)$.
In addition we assume that there are three more structures:

1. $$h : \mathbb{N} \rightarrow B,$$

   a hash function

2. $$\mu : A \times A \rightarrow C$$

   a map into a set C in which equality of elements can be checked fast

3. $$\nu : B \times B \rightarrow D \subset Hom_{set}(A, C)$$

with

$$\nu(b_1, b_2)(a) = \mu(b_1(a), b_2(a)).$$

Signature:
$a \in A$ is given (or introduced as part as the public key).

P chooses b and publishes $b(a)$.
Let m be a message.
P chooses a random element $k \in B$.
P computes

$$\phi := \nu(h(m) \circ b, h(k(a)) \circ k)$$

in D.
P publishes

$$(\phi, m, k(a)).$$

Verification:
V computes

$$\mu(h(m)(b(a)), h(k(a))(k(a)))$$

and compares it with $\phi(a)$.

2 Factorization and the system RSA

2.1 The arithmetical domain $\mathbb{Z}/N$

Let $N \in \mathbb{N}$ be a large number.
$\mathbb{Z}/N$ is the ring of residues of $\mathbb{Z}$ modulo N.
We identify it with the subset $\{1, ..., N\} \subset \mathbb{N}$ whenever this is convenient, and induce the corresponding rules for addition and multiplication in this set.

By $\mathbb{Z}/N^*$ we denote the set of numbers between 1 and $N-1$ which are prime to N. They correspond to the group of units in $\mathbb{Z}/N$.

The fundamental fact for the arithmetic in $\mathbb{Z}/N$ is the Euclidean algorithm which associates (efficiently) to any integer its residue in $\mathbb{Z}/N$.

It is used to make the addition, the substraction and the multiplication to a fast procedure.
With the usual "square-and multiply"-technique we make exponentiation to an operation with complexity $O(log(N))$ (hence **polynomial** in the number of bits of N.
For the inversion of (and hence for the division by) elements of $\mathbb{Z}/N^*$ we use the extended Euclidean algorithm which computes

$$d := gcd(a, b) = \lambda_1 a + \lambda_2 b.$$

So: For $x \in \mathbb{Z}/N^*$ we get

$$1 = y \cdot x + k \cdot N$$

and $y = x^{-1}$.

Remark:

This algorithm solves the additive version of the discrete logarithm: For given $x, y \in \mathbb{N}$ find k with $ky = x$.

Exercise: Generalize this to the additive group of finite fields!

An alternative method to compute the inverse in $(\mathbb{Z}/N)^*$ would be:
Let $\varphi(N)$ be the order of $\mathbb{Z}/N^*$. Then $x^{\varphi(N)-1} = x^{-1}$ for $x \in \mathbb{Z}/N^*$.

But to compute

$$\varphi(N) = \prod_{p^{\alpha_p} || N \text{ with } \alpha_p > 0} p^{\alpha_p - 1}(p-1)$$

one needs the factorization of N , and this is regarded as hard problem.
In fact it is the background for the RSA crypto system discussed in the next section.

2.2 Encryption with RSA

We shall discuss only a very simple and not secure protocol. But it demonstrates all essential features.

Let R be the person which wants to get messages (from everybody) in a secure way.

R chooses two (large) primes p, q and computes $N := pq$.
He publishes N together with a (usually small) number $e \in (\mathbb{Z}/N)^*$. He computes the inverse d with

$$de \equiv 1 \bmod \varphi(N)$$

.

This is easy for him since he knows that $\varphi(N) = (p-1)(q-1)$. **This is his most important secret.**

A person S wants to send a message $m < N$. He computes $m_c := m^e \in \mathbb{Z}/N$ and sends it.
R receives the residue m_c and computes $m_c^d \in \mathbb{Z}/N$ and gets as result the original message m.

It is obvious that this protocol has flaws. For instance we have no randomization. But it is easy to remedy this.

In fact there is a practicable protocol which is as secure as the so called RSA-challenge:
For randomly $m \in \mathbb{Z}/N$ and known (e, m^e) determine m!

It is obvious that this challenge is easy if we can factorize N. The converse is not known and may be not true. Due to Coppersmith we know that for too small d ($< N^{0.3}$) lattice reduction methods can answer the challenge.
In the next section we shall concentrate to methods breaking RSA via factorization.

2.3 Factorization algorithms

Let N be a natural number. We want to determine its prime divisors.
There are easy tests to decide (at least probabilistically) whether N is a power of a prime. So we shall assume that at least two different primes p and q divide N.

2.3.1 Smooth numbers

Let B be a natural number and let $\mathcal{P}$ be all prime numbers $\leq B$.
A number a is $B-$**smooth** iff all of its prime divisors lie in $\mathcal{P}$.
To describe how many numbers up to a bound x are $B-$smooth we use a so-called subexponential function which will play an important role in all complexity estimates: Define

$$L_x[\alpha, c] := exp(clog(x)^{\alpha} \cdot loglog(x)^{(1-\alpha)}$$

with $0 \leq \alpha \leq 1$.
In the extreme cases for α we get:

$L_x[0,c]$ is **polynomial** in the number of bits of x, and $L_x[1,c]$ is **exponential** in the number of bits of x.

We have the following important result of Canfield, Erdös an Pomerance:

Theorem 1 *Let $\alpha, \beta, r, s \in \mathbb{R}_{>0}$ with $\beta < \alpha \leq 1$. Then a random positive integer $\leq L_x[\alpha, r]$ is $L_x[\beta, s]-$smooth with probability $L_x[\alpha - \beta, -r(\alpha - \beta)/s]$.*

For cryptographical analysis we shall assume as **heuristic** that this probability does not change if we replace the interval from 1 to $L_x[\alpha, r]$ by a (large) interval beginning with some x_0. This is not proved but in all applications and experiments it worked very well.

2.3.2 The classical Fermat approach

We state the well known fact:
Let a be a number prime to N (if not we have found a non trivial divisor) then for all $l \in \mathbb{N}$ we have

$$p \mid (a^{l(p-1)} - 1)$$

and, since $p \neq N$ with high probability the greatest common divisor of $(a^{l(p-1)} - 1)$ with N is non trivial.

How to find a multiple of $(p-1)$?
Assume that $p-1$ is $B-$smooth.
Then try products of powers of elements in $\mathcal{P}$ (this are sparse products (few factors, small exponents) because their size is bounded by N. For each such product choose an (or the same) a and compute the resulting *gcd*.
If B is not too large this is done very quickly.
The disadvantage is that is is very unlikely that $p-1$ is $B-$smooth with a reasonable small B (see Theorem1), and such primes p can be easily avoided in designing RSA.

2.3.3 A Twist

Suppose that a number d is not a square modulo p.
Of course we cannot decide this since we do not know p but the probability

is $1/2$, and if the algorithm fails we try another d.
So $\sqrt{d}$ generates the field $\mathbb{F}_{p^2} = \{z := x + y\sqrt{d}; x, y \in \mathbb{Z}/p\}$.
Define $Re(z) := x$.
The norm homomorphism from $\mathbb{F}^*_{p^2}$ to $(\mathbb{Z}/p)^*$ is given by

$$x + y\sqrt{d} \mapsto x^2 - dy^2.$$

Its kernel has order $p + 1$.
In this kernel are the elements $z(x) = (x^2 + d)/(x^2 - d) + 2x/(x^2 - d) \cdot \sqrt{d}$ for all $x \in (\mathbb{Z}/p)^*$.
Hence $z(x)^{l \cdot (p+1)} = 1_{\mathbb{Z}/p}$.
Hence for randomly chosen $a \in \mathbb{Z}/N^*$ and for all multiples l of $p + 1$ we get: There is a non trivial common divisor of $z(x)^{l(p+1)}$ and N.
Now we are in the same situation as in the last section and so, if $p + 1$ is $B-$smooth we find a non trivial factor of N in time depending polynomially on the bit size of B.
But again the chance of success (i.e B not too big) is very small. We have to find a more systematical way to find factors.

2.4 Geometrical interpretation

To do this we have to interpret the two methods from above as special cases of a general principle.
Look at the first case: We used the properties of the multiplicative group of $\mathbb{Z}/p$ to get a congruence modulo p.
This multiplicative group can be seen as the $\mathbb{Z}/p-$valued points of the affine plane curve $G_m : XY = 1$ by identifying x with (x, x^{-1}). This curve has the additional property that two points can be multiplied by multiplying the coordinates: It is an algebraic group called the **multiplicative group**.
Its defining equation is universal, and hence it makes sense to speak about the points on the curve over $\mathbb{Z}/N$ which are in a natural way isomorphic to $(\mathbb{Z}/N)^*$ and which can be reduced modulo each divisor of N. This reduction and the knowledge of the order of $G_m(\mathbb{Z}/p)$ imply the algorithm in 2.3.2.
The method in 2.3.3 is again closely related to the multiplicative group. But it is now the subgroup of elements of norm 1 in $\mathbb{F}_{p^2}$. These elements are given by $z = x + y\sqrt{d}$ with $x^2 - dy^2 = 1$.
Hence (x, y) are $\mathbb{Z}/p-$points on the curve

$$T : X^2 - dY^2 = 1$$

which is again an affine curve with a multiplication defined on the points:

$$(x, y) \circ (x', y') := (xx' + dyy', xy' + yx').$$

The corresponding algebraic group is a **quadratic twist** of G_m, i.e. a one-dimensional torus T_d.
Again it is defined universally, and we constructed in 2.3.3 $\mathbb{Z}/N-$rational points on it and then used their reduction modulo p and the information over group orders to get divisors of N.

Hence a generalization will be: Look for other algebraic groups A defined over $\mathbb{Z}/N$ with reduction morphisms to $\mathbb{Z}/p$ and known orders of $A(\mathbb{Z}/p)$, the group of $\mathbb{Z}/p-$rational points of A.
One could go on to use tori but this leads to high dimensional varieties, and so they are not practical.

But there are other plane curves with simple equations such that the points form an abelian group: **Elliptic curves!**
One difference to G_m is that they cannot be defined in the affine plane, one has to go to the projective plane.

The advantage is that there are many (about p) elliptic curves over $\mathbb{Z}/p$ and the number of rational points are concentrated in an interval of length $2\sqrt{p}$.
Using this and Theorem 1 we shall be able to show that **every number** N can be factorized in subexponential time $L_N(1/2, 1)$! This is the worst running time, and it only occurs if the smallest prime in n has size $\approx \sqrt{N}$.
So the elliptic curve method is nowadays mostly used to find "small" divisors.
Before explaining this method we shall have to explain a little bit about elliptic curves, and since this will play an important role for other crypto systems, too, I shall postpone it till we have discussed an algorithm which is of general purpose and which is used to detect large factors.

2.5 Sieving

We use the following fact:
The polynomial $X^2 - 1$ has over **fields** at most two zeroes but over $\mathbb{Z}/N$ it has 2^k zeroes if k is the number of primes dividing N. So we have a good chance ($> 1/2$) that if $a, b \in \mathbb{N}$ with $a^2 \equiv b^2 \bmod N$ then $\gcd((a - b, N)$ or

$\gcd(a+b, N)$ are non trivial.
So we look for strategies to find such pairs a, b.
The general method used is called "**Index-Calculus**". It will become important later on again.
First step:
We choose a bound B and $\mathcal{P}$ as in 2.3.1. Recall that $\mathcal{P}$ has about $B/log(B)$ elements.
Second step:
We find $s >| \mathcal{P} |$ integers v with

$$v^2 \bmod N \ B - \text{smooth}.$$

I.e:

$$v^2 = \prod_{p \in \mathcal{P}} p^{e_{p,v}} + \lambda_v N.$$

This step is called: looking for relations.
The set of elements v yielding relations is denoted by V.
Third step:
The vectors $\{(..., e_{p,v} \bmod 2, ...)\} \subset (\mathbb{Z}/2)^{|\mathcal{P}|}$ are linearly dependent and so we find (for instance by Gauß elimination) a non trivial subset $S \subset V$ such that for **all** $p \in \mathcal{P}$ we have integers s_p with

$$\sum_{v \in S} e_{v,p} = 2s_p.$$

This step is called the Linear Algebra part.
Fourth step:
Now take $a := \prod_{v \in S} v \bmod N$, $b := \prod_{p \in \mathcal{P}} p^{s_p} \bmod N$ and compute the gcd's.
To make this efficient one has to be very careful in step 2 and step 3. This last one is solved by using techniques known in the cases of sparse matrices (Wiedemann, Lanczos). The parallelisation is still not completely solved.
The second step is to find relations. Of course this becomes easier if B is large. But then the third step becomes difficult. So one has to find an optimal trade-off.
This is found nowadays by sieving methods (and many tricks).

2.5.1 Sieves

The idea relies on the simple observation that for polynomials $f(X) \in \mathbb{Z}[X]$ and $x \in \mathbb{Z}$ we have

$$f(x + ip) \equiv f(x) \bmod p.$$

We want to find smooth numbers as values of f taken on an interval $\mathcal{L} := [0, ..., l-1]$.
We begin with $p_1 \in \mathcal{P}$ and assume that x_1 is the first element in $\mathcal{L}$ such that $f(x_1)$ is divisible by p.
Then we sieve and find $\mathcal{L}_1 := \{x_1 + \lambda \cdot p_1\} \subset \mathcal{L}$ with the same property. Now go to $p_2 \in \mathcal{P}$ and do the same procedure applying $f_2 := f(X)/p_1$ to $\mathcal{L}_1$ with respect to p_2 and so on till the value of f_i on the resulting subinterval is equal to 1.
The surviving elements in this sieving are $B-$smooth.
Now go to the list $\mathcal{L} \setminus \mathcal{L}_1$ and proceed till the list $\mathcal{L}$ is exhausted.
In the procedure there are a lot of divisions necessary. So one replaces this by going to logarithms and sieving with respect to them. A suspect for a smooth number then is an element for which the logarithm during sieving becomes nearly 0.

The question is: Which polynomials should one use, and which intervals are best in order to get relations?

2.5.2 The quadratic sieve

Pomerance had the idea to use polynomials and he proposed

$$f(X) := (X + \lfloor N \rfloor)^2 - N = X^2 + 2\lfloor\sqrt{N}\rfloor X - (N - \lfloor\sqrt{N}\rfloor).$$

The values for small x are near to $2x\sqrt{N}$ and so they have a better chance to be smooth than numbers near N.
Take x with $f(x)$ smooth. Then $v = x + \lfloor N \rfloor$ gives a relation.

This **quadratic sieve** has numerous variants like using more quadratic polynomials simultaneously. In any case the running time for optimal choice of B (w.r.t. the steps 2 and 3) is

$$L_N(1/2, 1) = exp(log(N)^{1/2} loglog(x)^{1/2}).$$

Theorem 2 *The complexity of the factorization of numbers is bounded by a subexponential function.*

Though this is the basic and best result we have we shall introduce two more methods (one of them announced already) with the same type of complexity but with better performance for medium sized resp. very large factors.

2.5.3 The number field sieve

The idea is again to produce non trivial relations modulo N of the form $x^2 - y^2 \equiv 0$ modulo N.

But now one uses elements x, y in **different** number fields K_i (i.e. finite algebraic extensions of $\mathbb{Q}$) given by polynomials $f_1(x)$ and $f_2(x)$.

In order to relate the elements of the different fields one has to assume that f_1 and f_2 have a common zero z modulo N. Hence one gets morphisms $\varphi_1 : \mathbb{Q}[X]/f_1(X) \rightarrow \mathbb{Z}/N$ by mapping X to z.

One introduces smoothness in number fields by using prime ideals of bounded norm instead of prime numbers and one has to overcome class number problems and problems with units (resulting in a higher number of needed relations) in order to find by sieving elements w_i in K_i which are squares in K_i with $\varphi_1(w_1) = \varphi_2(w_2)$.

By known technics one finds a square root z_i of w_i and then one gets:

$$\varphi_1(z_1)^2 = \varphi_2(z_2)^2$$

and is done.

2.6 The elliptic curve method

The theory of elliptic curves began over the complex numbers (mostly in the 19th century), in the first half of the 20th century it was seen how to use them over any field, and in the second half of last century the notion of elliptic schemes (i.e. elliptic curves over any ground ring) became a key notion of arithmetical geometry.

We shall need elliptic curves over finite fields and over $\mathbb{Z}/N$. They are the simplest examples for abelian varieties and at the same time, the most important ones in theory and applications today.

Let R be a commutative ring with unity. For the sake of simplicity we shall assume here that 6 is invertible in R

2.6.1 The projective plane

We take $\mathbb{P}^2_R$ as projective plane over R. Note that this space is, for arbitrary ring R, defined as projective scheme.
Example 1: Let $R = \mathbb{Z}$. Then the $\mathbb{Z}-$valued points of $\mathbb{P}^2_{\mathbb{Z}}$ are the triples $(x, y, z) \in \mathbb{Z}^3$ which are relatively prime and which are determined up to a common sign.
So every rational point extends in a unique way to a $\mathbb{Z}-$point and conversely.
Example 2:$R = \mathbb{Z}/N$ with $N = pq$.
The $\mathbb{Z}/N-$rational points are given by triples (x, y, z) which are classes modulo N of relatively prime integers, and the equivalence relation is scalar multiplication with elements in $(\mathbb{Z}/N)^*$.
There are the usual "lines at infinity" given by $l_X : X = 0$, $l_Y : Y = 0$, $l_Z : Z = 0$, and their complements U_X, U_Y, U_Z given by the condition that the variable in the index is not equal to zero.
But note that in general we cannot describe e.g. U_Z as affine plane V_Z by replacing Z by 1: If R is not a field then there are non-zero elements which are not invertible.
Take the second example: Triples (x, y, z) with x or y prime to q and $z = \lambda q$, λ prime to p, represent points in $\mathbb{P}^2_{\mathbb{Z}/N}$, lie in U_Z but not in V_Z.

2.6.2 Elliptic curves

Definition:
An elliptic curve over R is a curve in the projective plane given by one cubic equation

$$E : \; Y^2Z \;-\; X^3 - AXZ^2 - BZ^3 = 0$$

with $A, B \in R$ such that $4A^3 + 27B^2 \in R^*$.
An important property is that E has a zero section $0 := (0_R, 1_R, 0_R) \in l_Z$.
In fact E is a group scheme, i.e. for any $R-$algebra S the set $E(S)$ of S-points of E is abelian group with neutral element O, and the addition law $\oplus$ is given by homogenous polynomials with coefficients in R.

This implies that if we restrict to affine parts of the plane we get a **partial addition law** on the points in this part which works only for additions which do not leave this part.

So go to V_Z.
Taking $Z = 1$ and (ab-)using X, Y as affine coordinates we get the affine equation

$$Y^2 \ = \ X^3 + AX + B$$

for $E_{| \, V_Z}$.
The birational version of the addition $\oplus$ is given by the well known addition formulas for elliptic curves: For

$$P_1 = (x_1, y_1) \ , P_2 = (x_2, y_2).$$

we get

$$P_3 \ = (x_3, y_3) \ := \ P_1 \ \oplus \ P_2$$

with (in general):

$$x_3 \ = \ -(x_1 + x_2) + \ ((y_1 - y_2)/(x_1 - x_2))^2$$

and y_3 such that $(P_1, P_2, (x_3, -y_3))$ are collinear.
There is a special cases:
If $P_1 = P_2$ we get
$P_3 = 2P_1 = -2x_1 + ((3x_1^2 + A)/2y_1)^2$.

One sees immediately that addition formula is valid only if $x_1 - x_2$ resp. y_1 are units in R.
If R is a field this means: The partial addition is defined iff $P_1 \neq -P_2$, and in this case the sum is equal to O.

2.6.3 Factor finding with one elliptic curves

Now we are ready to apply these observations to factorization. Let be $N = p \cdot N'$ and take E as elliptic curve over $\mathbb{Z}/N$. The reduction map modulo p is the transition from residue classes modulo N to residue class modulo p. This extends in a natural way to reduction of points in the projective plane and in the affine part V_Z, to the equation of E resulting to

a cubic E^p and to points on E on V_Z. By the very definition the partial addition reduces, too.
After reduction we are dealing with elliptic curves over the field $\mathbb{Z}/p$ and with the usual properties of addition.

Let k_p be the order of the group of $\mathbb{Z}/p-$rational points on E^p. Hence for any l and any $\mathbb{Z}/N-$rational point $P \in V_Z$ of E we get:
The reduction of $lk_p \circ P$ modulo p is equal to 0.
As usually we do the scalar multiplication by doubling and adding using the 2-adic expansion of lk_p.
In the end we have left V_Z since we got 0 modulo p. So during the scalar multiplication we have to leave V_Z at some stage t after scalar multiplication by k_t.
There are two possibilities:
At the step t we have: $k_tP = 0$. Then k_t is a multiple of the order of the reduction of P modulo **all** divisors of N and this is highly improbable (cf. trivial congruences in 2.3.2).
Or we have left V_Z but stayed in U_Z. But this means that we have found two points P_1, P_2 on E with $\gcd((x_1 - x_2), N) = p$ or one point P_1 with $\gcd(y_1, N)$ $= p$.
Hence we can proceed as in 2.3.2 and 2.3.3 if k_p is $B-$smooth.

2.6.4 Varying the elliptic curve

We needed in the last section that the order of the group of $\mathbb{Z}/p-$rational points of E^p was $B-$smooth. Now we vary E and get different orders. So the chance of success for given N depends on the probability that elliptic curves over finite fields have $B-$smooth orders.
Using results from arithmetical geometry (like Hasse bound for orders and Deuring‘s lifting theorem) one gets

Theorem 3 *There exist effectively computable constants c_1, c_2 such that for all p and for all subsets $S \subset [p+1-\sqrt{p}, p+1+\sqrt{p}]$ the probability r_S that a random pair $(A, B) \in \mathbb{Z}/p \times \mathbb{Z}/p$ determines an elliptic curve*

$$E : Y^2Z - X^3 - AXZ^2 - BZ^3 = 0$$

with $\mid E(\mathbb{Z}/p) \mid \in S$ *is bounded as follows:*

$$c_1 \frac{|S|-2}{2\sqrt{p}+1}\, log(p)^{-1} \leq r_S \leq c_2 \frac{|S|}{2\sqrt{p}+1}\, log(p)loglog(p)^2.$$

Combining this with Theorem1 and estimating the cost (there are refinements!) of the check for one curve we get:

Result:
Using elliptic curves we can factorize any number N with complexity bounded by $L_N[1/2, 1]$.
This forces us to take for N numbers with at least 1024 bits (now), 2048 bits (for the next five years) and 4048 bits (for mid term purposes), and so RSA becomes quite clumsy. Hence there is a strong need for more efficient systems.

3 Discrete Logarithm Systems

3.1 Key exchange and signature in DL-systems

In section 1 we have discussed the tasks of cryptography and given an abstract setting for key exchange and signature. We have avoided to give a similar abstract setting for encryption (which could be done easily) but given an encryption scheme by using RSA.
Now we come back to our abstract setting and give (the only known) realization. (It is not difficult to find an encryption protocol with the method discussed below, too.)
We take the notions of subsections 1.2 and 1.3.
For A we take a cyclic group of prime order p embedded into $\mathbb{N}$, i.e. a group G **with a numeration.**
Take $B = Aut_{\mathbb{Z}}(A) \cong (\mathbb{Z}/p)^*$ identified with $\{1, ..., p-1\}$ by $b(a) := a^b$.
Take $C = A$ and $\mu =$ as addition in A. $\nu =$ is addition of automorphisms.

One fixes a publicly known generator $g_0 \in A$.

3.1.1 Key exchange

Each partner P_i chooses a (random) number $s_i \in \{1, ..., p-1\}$ as secret (and not the group order as in RSA schemes) and publishes $p_i := g_0^{s_i}$.
It is obvious but has to be emphasized that there is know leakage of security if one knows everything about the group A.
If P_1 wants to share a secret with P_i he powers p_i by s_1. The security considerations boil down to the complexity of the computation of the
Discrete Logarithm: How difficult is it to compute for randomly chosen $a_1, a_2 \in A$ a number $n \in \mathbb{N}$ with

$$a_2 = a_1^n?$$

3.1.2 Signature:

The person S who wants to sign a message chooses a secret $x \in \{1, ..., p-1\}$ and publishes $y := g_0^x$.
In addition it is publicly known that he uses a *hash function* h which maps $\mathbb{N}$ to A. Recall that it has to be impossible in practice to construct a number z such that $h(z)$ is a given value.

S chooses a second random number k and does the (for him since he knows k and x) easy computation

$$s := h(m)x + h(g_0^k) \cdot k \quad \mathbf{modulo}\ p.$$

The signed message consists of

$$(m, g_0^k, s).$$

To check the authenticity of m V computes

$$S = g_0^s, P = y^{h(m)}, H = (g_0^k)^{h(g_0^k)}.$$

Now the properties of exponentiation imply:

$$S = P \oplus H$$

if the signature is authentic. Otherwise it is rejected.
Again the security depends crucially on the difficulty to compute the discrete logarithm in A.
In fact one can change the two protocols such that under strong attack models the security is equivalent with the hardness of the discrete logarithm.

3.2 Generic attacks

We use the algebraic structure “group”.
This allows “generic” attacks.

3.2.1 Shanks’ Baby-Step-Giant-Step Method

Task: Take $P, Q \in G$ and find n with $Q = k \cdot P$.
The key fact is that looking up an element in an ordered set is inexpensive.
Do the baby step: For $i = 0, ..., S \leq \sqrt{p}$ compute

$$(i \cdot P, i).$$

Combine it with the giant step: Compute

$$Q - i \cdot S \cdot P$$

.
Now compare the two lists. If

$$i_0 \cdot P = Q - i_1 \cdot S \cdot P$$

then

$$k = i_0 + i_1 \cdot S.$$

The complexity is $O(\sqrt{p})$.
The disadvantage is that the algorithm needs $O(\sqrt{p})$ as space.

3.2.2 Pollard’s ρ-Algorithm

It is probabilistic. Its principle is the birthday paradoxon applied to random walks in G. They have loops with high probability after

$$\approx 1.03\sqrt{p}$$

steps.
One cannot control a random walk but in practice the following works very good:
Define the walk by induction. The result x_i of the $i-$th step should depend

only on x_{i-1}.
So partite G "randomly" into three sets T_j of size $\approx p/3$ and take

$$x_i = P + x_{i-1} \,\text{if}\, x_{i-1} \in T_1,$$

$$x_i = Q + x_{i-1} \,\text{if}\, x_{i-1} \in T_2,$$

$$x_i = 2x_{i-1} \,\text{if}\, x_{i-1} \in T_3.$$

There are efficient methods to detect collisions and having one it is easy to compute discrete logarithms.
This algorithm needs only very small space and has expected running time

$$\approx 1.03\sqrt{p}.$$

3.2.3 Mathematical task and outlook

It can be shown that in black box groups attacks to the discrete logarithm problem cannot be better as these generic attacks.
So we have to find numerated groups of order $p \approx 10^{180}$ for which no (known) attacks of smaller complexity than $p^{1/2}$ exist.
So we want to find groups of large prime order p such that the size of elements and the time needed to evaluate a group operation are **linear** in $log(p)$ but the time (or the space) needed to solve the discrete logarithm (for random elements) is **exponential** in $log(p)$.
The methods for solving the task are found in arithmetical and algebraic geometry and will be discussed during the next part of the lecture.