comput complexity 5 (1995), 237–247 1016-3328/95/040237-11 $1.50+0.20

ON THE ORACLE COMPLEXITY OF FACTORING INTEGERS

Ueli M. Maurer

Abstract. The problem of factoring integers in polynomial time with the help of an infinitely powerful oracle who answers arbitrary questions with yes or no is considered. The goal is to minimize the number of oracle questions. Let N be a given composite n-bit integer to be factored, where $n = \lceil \log_2 N \rceil$. The trivial method of asking for the bits of the smallest prime factor of N requires $n/2$ questions in the worst case. A non-trivial algorithm of Rivest and Shamir requires only $n/3$ questions for the special case where N is the product of two $n/2$-bit primes. In this paper, a polynomial-time oracle factoring algorithm for general integers is presented which, for any $\epsilon > 0$, asks at most ϵn oracle questions for sufficiently large N, thus solving an open problem posed by Rivest and Shamir. Based on a plausible conjecture related to Lenstra's conjecture on the running time of the elliptic curve factoring algorithm, it is shown that the algorithm fails with probability at most $N^{-\epsilon/2}$ for all sufficiently large N.

Key words. Oracle complexity; number theory; factoring; elliptic curves; cryptography.

Subject classifications. 68Q25, 94C60.

1. Factoring integers

One of the most prominent problems in computational number theory is integer factorization, which is exceedingly simple to describe to anyone without much background in mathematics. While no efficient algorithm for factoring is known, every child knows how to multiply two integers efficiently, which is the inverse operation of factoring. Another reason for the importance of the factoring problem is its crucial importance in cryptography. The security of several cryptographic systems and protocols, including the seminal and widely-used Rivest-Shamir-Adleman public-key cryptosystem (Rivest *et al.* 1978), is based on the assumed difficulty of factoring large integers, typically the product of two large prime numbers. The discovery of an efficient factoring algorithm would have dramatic impact on cryptography and in particular on the security of millions of information security systems implemented world-wide.

This paper is concerned with the difficulty of factoring integers. While no progress in efficient factoring algorithms is reported, we show that the apparent difficulty of factoring is concentrated in a small number of difficult yes/no questions. The number of such questions to be answered is an arbitrarily small fraction of the problem size. While this result has no direct impact on cryptography, it answers to the affirmative a question raised by Rivest and Shamir (Rivest & Shamir 1986), motivated by cryptographic security considerations.

For simplicity, and without loss of generality, we consider as the integer factoring problem the task of finding a non-trivial factor of a given integer. In other words, for a given problem instance to be solved, we do not require the complete factorization into primes. This distinction is not significant because repeated application of such a factoring algorithm would finally result in the complete factorization. Hence, according to our definition, many numbers are easy instances of the factoring problem, for example all numbers containing a small prime factor. The most difficult-to-factor integers appear to be those consisting of two primes of roughly equal size, which corresponds to the type of modulus normally used in implementations of the RSA system.

We briefly review some known results on factoring integers. Let N be an n-bit integer to be factored. The fastest known general-purpose factoring algorithm is the number-field sieve (Lenstra *et al.* 1990) whose asymptotic running time is $e^{cn^{1/3}(\log n)^{2/3}}$ for some small constant c. However, recent factoring records were achieved with a variant of the asymptotically slower quadratic sieve (Lenstra & Manasse 1991). The largest general integer that has been factored by a massively parallel computation of several months (Atkins *et al.* 1994) has 129 decimal digits.

There exist various special-purpose algorithms for factoring integers of a special form. Lenstra's elliptic curve factoring algorithm (Lenstra 1987) finds small factors efficiently. The largest prime factor found by an implementation of this algorithm has 40 decimal digits (Dixon & Lenstra 1993), and its asymptotic running time for finding a k-bit prime factor is $e^{ck^{1/2}(\log k)^{1/2}}$ for some small constant c. Other special-purpose algorithms exist for finding prime factors p of a special form, for instance when $p-1$ contains no large prime factor (Pollard 1974) or, more generally, when some cyclotomic polynomial evaluated at p contains no large prime factor (Bach & Shallit 1994).

A result with potentially dramatic consequences was recently obtained by Shor (Shor 1994) who showed that there exists a polynomial-time factoring algorithm for a quantum computer. Quantum computers are far from realizable but their existence appears to be consistent with quantum theory.

2. Oracle complexity

An interesting direction of research in complexity theory is to determine to what extent the difficulty of a conjectured difficult problem can be concentrated in a few difficult bits. More precisely, we consider the number of binary-valued (yes/no) questions that need to be answered (say by an oracle) for the problem to become easy. By easy, we mean probabilistic polynomial time in accordance with the common practice in complexity theory to distinguish, as a coarse classification of feasibility of an algorithm, between polynomial and superpolynomial time. We allow the questions to be asked adaptively, and we require the problem to be solved only with overwhelming probability rather than always. We call the minimal number of questions needed to solve a problem with overwhelming probability in probabilistic polynomial time the *oracle complexity* of the problem. Clearly, the oracle complexity is of interest only for problems not known to be in P because it is 0 for every problem in P.

One motivation for considering the oracle complexity of a problem is that when the number of questions can be reduced to $O(\log n)$, where n is the input size, then all possible oracle answers can be checked in polynomial time. This would correspond to a polynomial-time algorithm without access to an oracle.

Motivated by a paper by Rivest and Shamir (Rivest & Shamir 1986), this paper is concerned with the problem of factoring integers, which is widely believed to have no polynomial-time algorithm. A non-trivial factor of every n-bit integer N can easily be determined by asking $n/2$ questions, namely, "What is the i-th bit of the smallest prime factor of N?", for $1 \leq i \leq n/2$. For the special case of integers that are the product of two primes of roughly equal size, Rivest and Shamir described a polynomial-time algorithm based on integer programming techniques which asks at most $n/3$ questions. In this paper, a polynomial-time algorithm is presented which, for any given $\epsilon > 0$, asks at most ϵn questions. The claim that the algorithm fails only with exponentially small probability is based on a plausible number-theoretic conjecture about the distribution of smooth numbers in certain intervals and is closely related to a conjecture used by Lenstra in the running time analysis of his elliptic curve factoring algorithm (Lenstra 1987).

The major motivation of Rivest and Shamir for investigating this problem was that an adversary often has some side-information about the secret parameters of a cryptographic system, and that leakage of small amounts of such side-information should not strongly weaken the system. Our analysis shows that in a worst-case scenario in which the leaked side-information can be selected by an adversary who is restricted only in the size of the side-information string, an arbitrarily small fraction of bits of the solution are sufficient to break a system based on factoring. However, because oracles do not exist in real systems, the results of this paper have no direct implication on the security of existing cryptographic systems.

3. Preliminaries

The following lemma shows that in a sequence of k pairwise independent events, each having probability p, the probability that none of these events occur is at most $(1-p)/(kp)$. The events are pairwise independent if for any two events A and B, $P(A \cap B) = P(A) \cdot P(B)$.

LEMMA 3.1. *Let $X_1, \ldots, X_k$ be pairwise independent binary random variables where $P(X_i = 1) = p$ for $1 \leq i \leq k$. Then,*

$$P(X_1 = X_2 = \cdots = X_k = 0) \;\leq\; \frac{1-p}{kp}.$$

PROOF. Note that the expected value and the variance of X_i are given by $E[X_i] = p$ and $\operatorname{Var}[X_i] = p(1-p)$, respectively. Let S be the integer sum of $X_1, \ldots, X_k$, i.e., $S = X_1 + \cdots + X_k$. Hence we have $S = 0$ if and only if $X_1 = \cdots = X_k = 0$, and $E[S] = kp$. It is not difficult to prove (Chor & Goldreich 1989) that the variance of the sum of several pairwise independent random variables is equal to the sum of the individual variances. Thus, $\operatorname{Var}[S] = kp(1-p)$. For every real-valued random variable Y, we have

$$\operatorname{Var}[Y] \geq P[Y = 0] \cdot E[Y]^2$$

since the right-hand side is only one of several positive terms summing to the variance of Y. We conclude that $P[S = 0] \leq \operatorname{Var}[S]/E[S]^2 = (1-p)/(kp)$. □

For fixed p and $k \to \infty$ the proved bound on the probability that $X_1 = \cdots = X_k = 0$ is $O(1/k)$, which is optimal for pairwise independent random variables.

It is well-known that a polynomial of degree at most d over a field can be interpolated from any set of $d+1$ distinct arguments and their corresponding polynomial values. For the case of a finite field $GF(q)$ with q elements (where q is a prime power) this observation leads to a construction of a sequence of length q of $(d+1)$-wise independent random variables: When the $d+1$ coefficients of the polynomial are selected independently and at random with uniform distribution over $GF(q)$, then the polynomial's values for any set of $d+1$ arguments are also statistically independent and uniformly distributed. We will make use only of the special case $d = 1$ (pairwise independence), i.e., if the coefficents a and b of a linear polynomial $ax+b$ are chosen independently at random from the field, then any pair of polynomial values $ax_1 + b$ and $ax_2 + b$ are statistically independent and uniformly distributed over the field, which leads to a sequence of length q of pairwise independent random variables.

For a prime $p > 3$ the elliptic curve over $GF(p)$ with parameters a and b satisfying $4a^3 + 27b^2 \neq 0$ is defined as the set of points (x, y) with $x, y \in GF(p)$ satisfying the congruence equation

$$y^2 \equiv x^3 + ax + b \mod p, \tag{3.1}$$

together with a special element denoted $\mathcal{O}$ and called the point at infinity. This curve is denoted as $E_{a,b}(p)$. It is well-known that a group operation, which is called addition, can be defined on the set of points of the elliptic curve $E_{a,b}(p)$. Let P and Q be two points on $E_{a,b}(p)$. The point $P+Q$ is defined according to the following rules. $P + \mathcal{O} = \mathcal{O} + P = P$ for all P on E (i.e., $\mathcal{O}$ is the identity element of $E_{a,b}(p)$). Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$. If $x_1 = x_2$ and $y_1 = -y_2$, then $P + Q = \mathcal{O}$ (i.e., the inverse of the point (x, y) is the point $(x, -y)$). In all other cases the coordinates of $P + Q = (x_3, y_3)$ are computed as follows. Let λ be defined by

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } x_1 \neq x_2, \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if } x_1 = x_2, \end{cases}$$

where all operations are to be computed modulo p. (When $P + Q \neq \mathcal{O}$, then the denominator is not zero and thus the quotient is defined.) The resulting point $P + Q = (x_3, y_3)$ is defined by

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= \lambda(x_1 - x_3) - y_1. \end{aligned}$$

The prime p can be replaced by a composite N in the above definition and equations. However, $E_{a,b}(N)$ defined in this manner is not a group, but it can be extended to form a group by adjoining a small number of additional elements. (In the case where $N = p_1 \cdots p_r$ is the product of distinct primes

$p_i > 3$, $E_{a,b}(N)$ is the direct product of the corresponding elliptic curves over $GF(p_1), \ldots, GF(p_r)$.) Nevertheless, the addition operation, which is in this case called pseudo-addition, can be performed as long as it is defined, i.e., when the denominator is relatively prime to N, and it corresponds in fact to the addition operation on the extended curve. We refer to (Lenstra 1987) for further information on elliptic curves. Note that in (Lenstra 1987), points (x, y) are represented in projective coordinates as triples $(x : y : 1)$, and $\mathcal{O}$ is represented as $(0 : 1 : 0)$.

Unless stated otherwise, logarithms in this paper are to the natural base e. The cardinality of a set S is denoted by $\#S$.

4. The oracle factoring algorithm

Let N be a given composite n-bit integer and let $\epsilon < 0.5$ be an arbitrary given positive constant. If N is not known to be composite, a simple probabilistic compositeness test such as the Miller-Rabin test (Rabin 1980) can be used to prove the compositeness of N. In the sequel, a polynomial-time (in n) algorithm is described for finding a non-trivial divisor d of N $(1 < d < N)$ which, for all sufficiently large N, succeeds with probability at least $1 - N^{-\epsilon/2}$ and asks at most ϵn oracle questions.

The algorithm consists of four steps.

(i) (Special cases.) If 2 or 3 divides N or if N is a prime power $N = q^t$, output 2, 3, or q, respectively, and stop.

(ii) (Setup.) Choose δ with $0 < \delta < \epsilon$ as an arbitrary positive constant and let

$$\begin{aligned} c &= \frac{1}{\epsilon - \delta}, \\ w &= (\log N)^c, \\ h &= \prod_{r \le w,\, r \text{ prime}} r^{e(r)}, \end{aligned} \tag{4.2}$$

where $e(r)$ is the largest integer m with $r^m \le N^{1/2} + 2N^{1/4} + 1$. Choose s and t at random with uniform distribution from $GF(2^{3n})$. Fix a natural enumeration of the elements of $GF(2^{3n})$: $\alpha_1, \alpha_2, \ldots, \alpha_{2^{3n}}$. For a given natural representation of the elements of $GF(2^{3n})$ as triples of n-bit integers, let $(a_k, x_k, y_k) \in Z_{2^n} \times Z_{2^n} \times Z_{2^n}$ be the triple corresponding to

$s\alpha_k + t$ where α_k is the k-th element of $GF(2^{3n})$, and let $b_k \in Z_N$ be defined by

$$b_k \equiv y_k^2 - x_k^3 - a_k x_k \mod N.$$

REMARK 4.1. *A theorem due to Hasse states that every elliptic curve modulo a prime p has between $p - 2\sqrt{p} + 1$ and $p + 2\sqrt{p} + 1$ elements. Therefore, $N^{1/2} + 2N^{1/4} + 1$ is an upper bound on the order of an elliptic curve over $GF(p)$, where p is the smallest prime factor of N. As mentioned in Section 2, the above construction guarantees that the triples (a_k, x_k, y_k) are pairwise statistically independent. Instead of the field $GF(2^{3n})$, any other finite field with cardinality greater than N^3 could be used to create an appropriate list of pairwise independent triples (a_k, x_k, y_k). Only triples for which all three components are smaller than N will actually be of possible use.*

(iii) (Oracle questions.) Ask the oracle the following question. If there exists a positive integer $k < 2^{\lfloor \epsilon n \rfloor}$ such that the following two conditions are satisfied:

(1) for the smallest prime factor p of N,

$$4a_k^3 + 27b_k^2 \not\equiv 0 \mod p,$$

and each prime factor r dividing $\#E_{a_k,b_k}(p)$ satisfies $r \leq w$,

(2) for some prime factor $q \neq p$ of N,

$$4a_k^3 + 27b_k^2 \not\equiv 0 \mod q$$

and $\#E_{a_k,b_k}(q)$ is not divisible by the largest prime number dividing the order of the point (x_k, y_k) on the elliptic curve $E_{a_k,b_k}(p)$,

where a_k, x_k, y_k and b_k are defined as in step (ii), then output the binary representation of the smallest such k, else output 0.

REMARK 4.2. *Of course, this question can easily be transformed into $\lfloor \epsilon n \rfloor$ questions with a yes/no answer.*

(iv) (Factorization.) If the oracle's answer is 0, stop. In this case, the algorithm fails. If the oracle's answer is some $k > 0$, proceed as follows. Compute (a_k, x_k, y_k) and b_k as described in step (ii). Let $P = (x_k, y_k)$ be a point on $E_{a_k,b_k}(N)$ (which is not a group). Try to compute $h \cdot P$ using the pseudo-addition method described in (2.4) of (Lenstra 1987), pretending that N is prime. At some point during this computation, the addition of two points (x', y') and (x'', y'') will fail because $\gcd(x' - x'', N) > 1$. Output this divisor of N.

5. Analysis of the algorithm

We need to prove two things: that the algorithm runs in polynomial time and that the failure probability is at most $N^{-\epsilon/2}$.

THEOREM 5.1. *If the oracle's answer is $k > 0$, then the algorithm runs in polynomial time and always finds a non-trivial divisor of N.*

PROOF. That the algorithm runs in polynomial time follows from the facts that the pseudo-addition can be performed in time $O(n^2)$ and that the number of pseudo-additions required for computing $h \cdot P$ is at most $2\lceil \log_2 h \rceil - 1$, which is polynomial in n since according to (4.2),

$$\log_2 h \;=\; \sum_{r \leq w,\, r \text{ prime}} e(r) \log_2 r \;\leq w \log_2 w$$

and $w = O(n^c)$. N is guaranteed to have a prime factor smaller than $\sqrt{N}$ and hence Proposition (2.6) in (Lenstra 1987) for $\nu = \sqrt{N}$ implies that the algorithm always succeeds. □

It follows from the Corollary to Theorem 3.1 of Canfield, Erdős and Pomerance (Canfield *et al.* 1993) that the probability that a random positive integer $s \leq x$ has all its prime factors $\leq L(x)^\alpha$, where

$$L(x) = e^{\sqrt{\log x \log\log x}},$$

is $L(x)^{-1/(2\alpha)+o(1)}$, for $x \to \infty$. In the analysis of his elliptic curve factoring algorithm (Lenstra 1987), Lenstra stated the plausible conjecture that the same result is valid if s is a random integer in the interval $(x + 1 - \sqrt{x}, x + 1 + \sqrt{x})$. We will need a similar conjecture with a smaller smoothness bound.

One can prove that for every $\beta > 0$, the mentioned result of Canfield *et al.* implies that the probability that a random positive integer $s \leq x$ has all its prime factors $\leq (\log x)^c$, for $c > 1$, is greater than $x^{-1/c-\beta}$ for all sufficiently large x. The conjecture we will need is that the same result is valid if $1/c + \beta < 1/2$ and s is a random integer in the interval $(x + 1 - \sqrt{x}, x + 1 + \sqrt{x})$.

CONJECTURE 5.2. *For every $\beta > 0$ and $c > 1/(0.5-\beta)$, and for all sufficiently large x, the fraction of integers in the interval $(x+1-\sqrt{x}, x+1+\sqrt{x})$ with no prime factor greater than $(\log x)^c$ is at least $x^{-1/c-\beta}$.*

We believe that our conjecture is as plausible as Lenstra's conjecture. Note that in our algorithm we have $c > 2$ but that for $1/c + \beta > 1/2$, the conjecture cannot be true since the expected number of smooth integers in the given interval would be less than 1.

THEOREM 5.3. *If Conjecture 5.2 is true, then the oracle outputs* 0 *(and hence the oracle factoring algorithm fails) with probability at most $N^{-\epsilon/2}$.*

PROOF. Let p be the smallest prime divisor of N, and let U be the number of integers in the interval $(p+1-\sqrt{p}, p+1+\sqrt{p})$ for which no prime factor is greater than $w = (\log N)^c$. According to our conjecture with $\beta = \delta/2$, U is bounded from below by

$$U > (2\lfloor\sqrt{p}\rfloor + 1)p^{-1/c-\delta/2},$$

for all sufficiently large p. Note that $-1/c - \delta/2 = -\epsilon + \delta/2$. It follows from proposition (2.7) of (Lenstra 1987) that the number T of triples $(a, x, y) \in Z_N \times Z_N \times Z_N$ that are successful in step (iii) of our algorithm is, for sufficiently large p, lower bounded by

$$\begin{aligned} T &> N^3 \frac{C_1}{\log p} \cdot \frac{U-2}{2\lfloor\sqrt{p}\rfloor + 1} \\ &> N^3 \frac{C_2}{\log p} \cdot p^{-\epsilon+\delta/2}, \end{aligned}$$

where C_1 and C_2 are positive constants. Hence, the probability that a triple selected randomly from $Z_{2^n} \times Z_{2^n} \times Z_{2^n}$ is successful is equal to $T/2^{3n}$. Because the triples (a_k, x_k, y_k), for $1 \leq k \leq 2^{3n}$, are pairwise independent, it follows from Lemma 3.1 that the probability Q that none of the triples (a_k, x_k, y_k), for $1 \leq k < 2^{\lfloor \epsilon n \rfloor} - 1$, is successful (and therefore the oracle answers 0) is upper bounded by

$$\begin{aligned} Q &< \frac{1}{(T/2^{3n}) \cdot (2^{\lfloor \epsilon n \rfloor} - 1)} \\ &< \frac{8 \log p}{C_2 p^{-\epsilon+\delta/2}(\frac{1}{2}N^\epsilon - 1)}, \end{aligned}$$

where we have made use of $N^3/2^{3n} > 1/8$ and $2^{\lfloor \epsilon n \rfloor} > 2^{\epsilon n - 1} > \frac{1}{2}N^\epsilon$. Since $p \leq N^{1/2}$, the last expression is smaller than $N^{-\epsilon/2}$ for all sufficiently large N. □

6. Concluding remarks

Our conjecture appears plausible, and could also be replaced by weaker conjectures, but it remains an open problem to prove that our result also holds without any number-theoretic conjecture. Further, we suggest to investigate the oracle complexity of other number-theoretic problems like the discrete logarithm problem. It is also an open problem to determine whether the oracle complexities of cryptographically relevant problems are related to the security of the corresponding systems.

The oracle complexity can always be reduced by an additive term of size $O(\log n)$ where n is the size of the input because the answers to this many oracle questions can all be guessed and checked in polynomial time. It is an interesting open problem to find computational problems for which the oracle complexity is as close to $O(\log n)$ as possible, for instance $O((\log n)^c)$ for some constant $c > 1$.

Acknowledgements

It is a pleasure to thank Uri Feige, Kevin McCurley, and Andrew Odlyzko for helpful discussions. Claus Schnorr has pointed out that a result similar to ours could be obtained by using the class group factoring algorithm (Schnorr & Lenstra 1984) instead of the elliptic curve factoring algorithm.

References

D. Atkins, M. Graff, A.K. Lenstra, and P.C. Leyland, The magic words are squeamish ossifrage. In *Advances in Cryptology—Asiacrypt '94*, Lecture Notes in Computer Science **917**, Springer-Verlag (Berlin), 1994, 263–277.

E. Bach and J. Shallit, Factoring with cyclotomic polynomials. *Mathematics of Computation* **52** (1989), 201–219.

E.R. Canfield, P. Erdős, and C. Pomerance, On a problem of Oppenheim concerning "Factorisatio Numerorum". *Journal of Number Theory* **17** (1983), 1–28.

B. Chor and O. Goldreich, On the power of two-point based sampling. *Journal of Complexity* **5**(1) (1989), 96–106.

B. Dixon and A.K. Lenstra, Massively parallel elliptic curve factoring. In *Advances in Cryptology—EUROCRYPT '92*, Lecture Notes in Computer Science **658**, Springer-Verlag (Berlin), 1993, 183–193.

A.K. Lenstra, H.W. Lenstra, M.S. Manasse, and J.M. Pollard, The number field sieve. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, 1990, 564–572.

A.K. Lenstra and M.S. Manasse, Factoring with two large primes. In *Advances in Cryptology—EUROCRYPT '90*, Lecture Notes in Computer Science **473**, Springer-Verlag (Berlin), 1991, 69–80.

H.W. Lenstra, Jr., Factoring integers with elliptic curves. *Annals of Mathematics* **126** (1987), 649–673.

A. Menezes, *Elliptic curve public key cryptosystems*. Kluwer Academic Publishers, 1993.

J.M. Pollard, Theorems on factorization and primality testing. *Proceedings of the Cambridge Philosophical Society* **76** (1974), 521–528.

C. Pomerance, Factoring. In *Cryptology and computational number theory*, ed. C. Pomerance, *Proceedings of the Symposium in Applied Mathematics* **42**, American Mathematical Society, 1990, 27–47.

M.O. Rabin, Probabilistic algorithm for testing primality. *Journal of Number Theory* **12** (1980), 128–138.

R.L. Rivest and A. Shamir, Efficient factoring based on partial information. In *Advances in Cryptology—EUROCRYPT '85*, Lecture Notes in Computer Science **219**, Springer-Verlag (Berlin), 1986, 31–34.

R.L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* **21**(2) (1978), 120–126.

P. Shor, Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, 1994, 124–134.

C.P. Schnorr and H.W. Lenstra, A Monte Carlo factoring algorithm with linear storage. *Mathematics of Computation* **43**(167) (July 1984), 289–311.

Manuscript received 3 February 1995

Ueli M. Maurer
Department of Computer Science
Swiss Federal Institute of Technology (ETH)
CH-8092 Zurich
Switzerland
maurer@inf.ethz.ch