

Applied Cryptography 2

Maria Eichlseder
Christian Rechberger

Daniel Kales
Markus Schofnegger

Summer Term 2020

Cryptanalysis

...or how to scale your cipher



The best available cryptanalysis (+security margin) indicates the necessary **key size**, number of **rounds**, etc. to achieve a certain **security level**:

- **Asymmetric crypto**: Best algorithm to solve hard problem
- **Symmetric crypto**: Generic and dedicated attacks

Course topics: In more detail

Asymmetric Cryptanalysis



- Factoring & Continued Fractions
- Lattices

Multiparty Computation



- How to securely compute joint output on secret inputs
- Protocols, Applications, ...

Symmetric Cryptanalysis



- Statistical attacks on block ciphers (Linear & Differential Cryptanalysis)
- Algebraic Attacks
- Hash Function Cryptanalysis

Selected Topics



- YOU choose (because you present)

Prerequisites

If you already heard Applied Cryptography 1 – perfect! If not:

- Asymmetric Crypto:
 - Groups & Finite Fields
 - Number theory basics
 - Elliptic Curves
- Symmetric Crypto:
 - Block Ciphers, Hash Functions
 - Differential Cryptanalysis
 - (Very basic statistics)
- For the KU: Mathematical programming (Sage, C++ with library or similar)

How to get your grade

Exercises (KU)

- Programming exercises
2 submissions + interview (“Abgabegespräch”)

Lecture (VO)

You have 2 options:

- a Final exam
- or
- b Seminar presentation

Exercises (KU)

- Implement attacks and protocols from the lecture in teams of 2
- 2 Assignments à 3 tasks:
 - A** 4 points (easy)
 - B** 8 points (medium)
 - C** 12 points (pro)
- But: 100 % = 32 points (not 48), so

≥ 28 points	1
≥ 24 points	2
≥ 20 points	3
≥ 16 points	4
else	5

KU: Assignments

Assignment 1: Asymmetric Cryptanalysis and Multiparty Computation

- Release: 19 Mar 2020 (= team registration deadline!)
- Question time: 23 Apr 2020
- Submission: 30 April 2020

Assignment 2: Symmetric Cryptanalysis

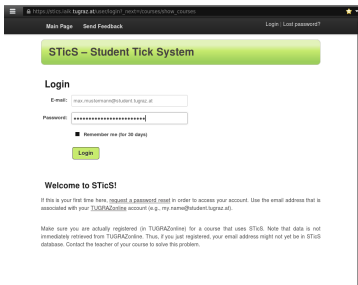
- Release: 7 May 2020
- Question time: 4 Jun 2020
- Submission: 12 Jun 2020 (Friday!)

Note: We usually won't need the 17:30–18:15 timeslots except those dates

KU: Submissions

Upload your KU submissions in the Student Tick System (STicS):

<https://stics.iaik.tugraz.at>



The screenshot shows a web browser window with the URL https://stics.iaik.tugraz.at/next/courses/show_courses. The page has a dark header with "Main Page" and "Send Feedback" links, and a "Login | Lost password?" link on the right. Below the header is a green banner with the text "STicS – Student Tick System". The main content area is titled "Login" and contains a form with the following fields: "E-mail:" with the value "max.mustermann@student.tugraz.at", "Password:" with a masked password "*****", and a checkbox labeled "Remember me (for 30 days)". A green "Login" button is below the form. Below the login form, there is a "Welcome to STicS!" section with a paragraph of text: "If this is your first time here, request a password reset in order to access your account. Use the email address that is associated with your TUGRAZonline account (e.g., my.name@student.tugraz.at).". At the bottom, there is a paragraph: "Make sure you are actually registered (in TUGRAZonline) for a course that uses STicS. Note that data is not immediately retrieved from TUGRAZonline. Thus, if you just registered, your email address might not yet be in STicS database. Contact the teacher of your course to solve this problem."

Note: Even if you've used STicS before, you might have to request a new password.

Lecture (VO) – Option **a** Final exam

Exam mode

Choose a date:

- 02 July 2020: written exam
- Later: ask for an oral exam date

Questions:

- Answer 4 out of 5 questions
- List of questions at the end of slides
- Lecture & seminar topics

$\geq 87.5 \%$	1
$\geq 75.0 \%$	2
$\geq 62.5 \%$	3
$\geq 50.0 \%$	4
else	5

Lecture (VO) – Option **b** Seminar presentation

Seminar mode

Prerequisites:

- Participate in KU
- Participate in VO
 - Regular attendance
 - Think about what you hear, and ask questions

Presentations:

- Collaboration in same team as KU
- Select your topic in STicS until [19 Mar 2020](#)
- 45 minutes (incl. time for questions)
- Send us your planned slides at least 1 week in advance
- Include possible exam questions at the end of your slides

VO: Schedule I

■ March

- Factoring + Continued Fractions
- Multiparty Computation 1+2
- Lattices

■ April

- Linear Cryptanalysis
- Differential Cryptanalysis
- Tools for Cryptanalysis

VO: Schedule II

- May
 - Algebraic Attacks
 - Advanced Differential Attacks
 - Hash Function Cryptanalysis
- May/June
 - Seminar presentations (2–3 per lesson)

VO: Seminar topics I

1. Asymmetric Cryptography: Discrete Logarithm Problem I
2. Asymmetric Cryptography: Discrete Logarithm Problem II
3. Asymmetric Cryptography: Integer Factorization Problem II
4. Block Ciphers: Statistical Attacks
5. Block Ciphers: Division Property
6. Block Ciphers: MitM Attacks, Biclique, etc.
7. Hash Functions: Multi-Collisions & Functional Graph
8. Hash Functions: Rebound Attack
9. Authenticated Encryption: Robustness & CAESAR
10. Authenticated Encryption: Permutations, Tweakable Block Ciphers

VO: Seminar topics II

11. Algebraic Attacks: Gröbner Basis, etc.
12. Elliptic-Curve Cryptography: Advanced topics
13. Lattices: The NTRU cryptosystem
14. Lattices: Learning with errors
15. Multiparty Computation: Zero-Knowledge from MPC techniques
16. Selected Topics: Tools in Symmetric Crypto (MILP, SAT, CP, etc.)
17. Selected Topics: Backdoors in Cryptography
18. Selected Topics: Password Hashing
19. Selected Topics: (Fully) Homomorphic encryption
20. Selected Topics: Error correcting codes and cryptography
21. Selected Topics: Post-Quantum Crypto (Isogenies, Hash-based, ...)

Further Information

- Website:
 - <https://www.iaik.tugraz.at/course/applied-cryptography-2-705064-sommersemester-2020/>
 - Slides, exercise sheets, deadlines, ...
- Newsgroup:
 - tu-graz.lv.angewandte-kryptografie-2
 - Questions, news, ...
- STicS:
 - <https://stics.iaik.tugraz.at>
 - Team registration, KU submissions, VO seminar submissions

Questions?

Integer Factorization and RSA

Daniel Kales

Slides by Maria Eichlseder; parts based on slides by Mario Lamberger

Applied Cryptography 2 – ST 2020

Outline

Introduction to Modern Factoring Algorithms

Factoring with Factor Bases - Dixon's random squares algorithm - Quadratic sieve algorithm

Factoring with Continued Fractions - CFRAC algorithm - Wiener's attack on RSA

Factoring with Elliptic Curves - Lenstra's ECM algorithm

Introduction to Modern Factoring Algorithms



Recall: RSA Encryption / Signatures

🔑 RSA Key Generation

- Choose 2 large, random primes p, q
- Compute public modulus $n = p \cdot q$
- Choose public exponent e co-prime to $\varphi(n)$
- Compute private exponent $d \equiv e^{-1} \pmod{\varphi(n)}$

🔑 public key = (e, n)

🔑 private key = (d, n)

Euler function:

$$\varphi(pq) = (p-1)(q-1)$$

Euler theorem:

if a, n are coprime, then
 $a^{\varphi(n)} \equiv 1 \pmod{n}$

If we can solve the IFP, we can recover p, q from n and thus break RSA:

❓ Integer Factorization Problem (IFP)

Given $n \in \mathbb{N}$, find primes $p_i \in \mathbb{P}$ and $e_i \in \mathbb{N}$ such that $n = p_1^{e_1} \cdot p_2^{e_2} \cdots p_k^{e_k}$.

→ how large should we choose n for an attack complexity of at least, say, 2^{128} ?

Factoring Methods

Fastest general factoring algorithms (take with a grain of salt):

- 1 General number field sieve
- 2 Multiple polynomial quadratic sieve
- 3 Lenstra elliptic curve factorization

You already know two conceptual forerunners of these methods:

- Fermat's difference-of-squares algorithm
- Pollard's $p - 1$ method

Two Ways to Factor n

Difference-of-Squares factorization

Finding $p, q : n = p \cdot q \iff$ finding $x, y : x^2 \equiv y^2 \pmod{n}$

- Century-old idea: Fermat's factoring algorithm (\rightarrow AK1 KU)
- Modern sieving algorithms find x, y much more efficiently
- This lecture: Dixon's random squares, Quadratic sieve, CFRAC

Algebraic Group factorization

Compute in a group \pmod{n} and try to detect identity \pmod{p}

- Example: Pollard's $p - 1$ method (\rightarrow AK1 VO)
- This lecture: Lenstra's ECM algorithm

Factoring with Factor Bases



Difference-of-Squares and Factor Bases

The base of modern factoring methods is a century-old idea:

Difference of Squares: $x^2 - y^2 = (x + y)(x - y)$

Find x, y with $x \not\equiv \pm y \pmod{n}$ such that

$$x^2 \equiv y^2 \pmod{n}.$$

Then $(x - y)(x + y) \equiv 0 \pmod{n}$, and if we are lucky,

$$\gcd(x \pm y, n) \in \{p, q\}.$$

Question: How to find such a quadratic congruence?

For random x , it is unlikely that $x^2 \bmod n$ produces a square y^2

Difference-of-Squares and Factor Bases

- **Observation:** When is a number Y a square, i.e., $Y = y^2$?

Consider the prime factorization $Y = \prod_i p_i^{e_i}$:

Y is a square y^2 iff every exponent e_i is even, and we get $y = \prod_i p_i^{e_i/2}$

- **Idea:** Try many x_i^2 and combine the outputs Y_i to make e_i even:

$$x_1^2 \mod n = Y_1 = 2^3 \cdot 3^2 \cdot 5$$

$$x_2^2 \mod n = Y_2 = 2 \cdot 5$$

\Downarrow

$$(x_1 \cdot x_2)^2 \mod n = Y_1 \cdot Y_2 = 2^4 \cdot 3^2 \cdot 5^2 = (2^2 \cdot 3 \cdot 5)^2 = y^2$$

Difference-of-Squares and Factor Bases

- Obvious problem: So now we need to factor all Y_i to factor n ?
- **Solution:** We use a **factor base** $\mathcal{B} = \{p_1, p_2, \dots, p_k\}$ containing all prime numbers $\leq B$ (and sometimes -1). We only check if the Y_i can be factored wrt. \mathcal{B} .

Definition (B -smooth numbers)

n is **B -smooth** (\mathcal{B} -smooth) if every prime factor p of n is $\leq B$ ($\in \mathcal{B}$)

Example: $n = 864 = 2^5 \cdot 3^3$ is **3-smooth**

Dixon's Random Squares Method

- 1 Select **factor base** of small prime numbers $\mathcal{B} = \{-1, p_1, p_2, \dots, p_k\}$
- 2 Collect **relations** (x_i, Y_i) with $Y_i = \underline{x_i^2} \pmod{n}$ and $Y_i = \prod_t p_t^{e_{it}}$
(select random x_i , test if Y_i is \mathcal{B} -smooth)
(typically $x_i \in [\sqrt{n} - C, \sqrt{n} + C]$, so $Y_i = x_i^2 - n$ is small)
- 3 **Solve**: select subset of Y_i such that their product is square
(= all factors p_t occur an even number of times)
 \Rightarrow solving a linear equation system (mod 2): $\mathbf{E} \cdot \mathbf{s} \equiv \mathbf{0}$
- 4 $x = \prod x_i$ and $y = \sqrt{\prod Y_i}$
- 5 Hope that $\gcd(x \pm y, n) \in \{p, q\}$

Factoring with Factor Bases: Example I

Factor $n = 2769$ using factor base $\mathcal{B} = \{2, 3, 5, 7\}$

$x_i = \lfloor \sqrt{n} \rfloor + i$	53	54	55	56	57	58	...
$Y_i = x_i^2 - n$	40	147	256	367	480	595	...
$\div 2$	2^3		2^8		2^5		
$\div 3$		3			3		
$\div 5$	5				5	5	
$\div 7$		7^2				7	
Rest	1	1	1	367	1	17	...

Factoring with Factor Bases: Example II

Factor $n = 2769$ using factor base $\mathcal{B} = \{2, 3, 5, 7\}$

$x_i = \lfloor \sqrt{n} \rfloor + i$	53	54	55	56	57	58	...
$Y_i = x_i^2 - n$	40	147	256	367	480	595	...
$\div 2$	1	0	0		1		
$\div 3$	0	1	0		1		
$\div 5$	1	0	0		1		
$\div 7$	0	0	0		0		
Rest	✓	✓	✓	✗	✓	✗	...

- Solve the linear system (mod 2) $\rightarrow \mathbf{s} = (1, 1, 0, 1)$ or $(0, 0, 1, 0)$
- $x = \prod x_i = 53 \cdot 54 \cdot 57 = 163134$
 $y = \sqrt{\prod Y_i} = 2^{(3+5)/2} \cdot 3^{(1+1)/2} \cdot 5^{(1+1)/2} \cdot 7^{2/2} = 1680$
- $\gcd(x + y, n) = \gcd(164814, 2769) = 39$ $(n = 3 \cdot 13 \cdot 71)$

Quadratic Sieve Method

Observation: If p divides Y , i.e., $x^2 - n \equiv 0 \pmod{p}$, then $(x + p)^2 - n \equiv 0 \pmod{p}$.
This is useful to speed up the trial divisions by \mathcal{B} (“**Sieving**”):

- 1 Select a factor base $\mathcal{B} = \{-1, p_1, p_2, \dots, p_k\}$.
For each prime p_j , solve $\alpha_j^2 - n \equiv 0 \pmod{p_j}$ (0 or 2 solutions)
(if there are 0 solutions, remove p_j from factor base)
- 2 Set up table of x_i, Y_i for x_i in some interval $[\sqrt{n} - C, \sqrt{n} + C]$.
For each α_j , divide only Y_i with $x_i = \alpha_j + k \cdot p_j$ for some $k \in \mathbb{N}$ by powers of p_j
- 3 ... (continue from step 3 of Dixon's Random Squares)

Factoring with Continued Fractions

%

And now for something completely different...

You may or may not celebrate π day on an upcoming Saturday (3.14)

But did you know about π approximation day in July: 22/7

$$\frac{22}{7} = 3.1428 \dots$$

is a useful approximation for

$$\pi = 3.1415 \dots$$

Which raises a number of questions:

How do we approximate irrational numbers?

Would there be any better π approximation dates to celebrate?

And what the heck does this have to do with factorization?

Continued fractions to represent real numbers

Definition: Continued fraction expansion

The **continued fraction expansion** of $\alpha \in \mathbb{R}^+$ is

$$\alpha = c_0 + \frac{1}{c_1 + \frac{1}{c_2 + \frac{1}{c_3 + \dots}}} = [c_0; c_1, c_2, c_3, \dots]$$

with $c_0 \in \mathbb{Z}$ and $c_i \in \mathbb{N}$ for $i \geq 1$.

The values c_i can be successively computed via:

$$c_0 = \lfloor \alpha \rfloor$$

$$\varepsilon_0 = \alpha - c_0$$

$$c_1 = \lfloor 1/\varepsilon_0 \rfloor$$

$$\varepsilon_1 = 1/\varepsilon_0 - c_1$$

$$c_2 = \lfloor 1/\varepsilon_1 \rfloor$$

$$\varepsilon_2 = 1/\varepsilon_1 - c_2$$

$$\vdots$$

$$\vdots$$

Continued fractions: Example I

Find the continued fraction expansion of $\alpha = \frac{45}{89}$:

Solution

$$c_0 = \lfloor \alpha \rfloor = \left\lfloor \frac{45}{89} \right\rfloor = 0$$

$$\varepsilon_0 = \alpha - c_0 = \frac{45}{89} - 0 = \frac{45}{89}$$

$$c_1 = \left\lfloor \frac{1}{\varepsilon_0} \right\rfloor = \left\lfloor \frac{89}{45} \right\rfloor = 1$$

$$\varepsilon_1 = \frac{1}{\varepsilon_0} - c_1 = \frac{89}{45} - 1 = \frac{44}{45}$$

$$c_2 = \left\lfloor \frac{1}{\varepsilon_1} \right\rfloor = \left\lfloor \frac{45}{44} \right\rfloor = 1$$

$$\varepsilon_2 = \frac{1}{\varepsilon_1} - c_2 = \frac{45}{44} - 1 = \frac{1}{44}$$

$$c_3 = \left\lfloor \frac{1}{\varepsilon_2} \right\rfloor = \left\lfloor \frac{44}{1} \right\rfloor = 44$$

$$\varepsilon_3 = \frac{1}{\varepsilon_2} - c_3 = \frac{44}{1} - 44 = 0$$

$$\Rightarrow \frac{45}{89} = [0; 1, 1, 44] = 0 + \frac{1}{1 + \frac{1}{1 + \frac{1}{44}}}$$

Continued fractions: Example II

More examples for irrational numbers:


$$\varphi = [1; 1, 1, 1, 1, 1, 1, 1, 1, 1, \dots]$$

$$\sqrt{2} = [1; 2, 2, 2, 2, 2, 2, 2, 2, 2, \dots]$$

$$\sqrt{19} = [4; 2, 1, 3, 1, 2, 8, 2, 1, 3, 1, 2, 8, \dots]$$

$$e = [2; 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, \dots]$$

$$\pi = [3; 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, \dots]$$


$$\pi = 3 + \frac{1}{7 + \frac{1}{\dots}} \approx \frac{21 + 1}{7}$$

Continued fractions to approximate real numbers

Definition: n -th convergent

The n -th convergent of $\alpha = [c_0; c_1, c_2, \dots] \in \mathbb{R}^+$ is

$$\frac{a_n}{b_n} = [c_0; c_1, c_2, \dots, c_n]$$

- Convergents can be computed by recursion:

$$\frac{a_0}{b_0} = \frac{c_0}{1}, \quad \frac{a_1}{b_1} = \frac{c_0 c_1 + 1}{c_1}, \quad \dots, \quad \frac{a_n}{b_n} = \frac{c_n a_{n-1} + a_{n-2}}{c_n b_{n-1} + b_{n-2}}$$

- Convergents are in a sense the “best” approximation of α :

$$\left| \frac{a_n}{b_n} - \alpha \right| < \left| \frac{a}{b} - \alpha \right| \quad \text{for all } \frac{a}{b} \in \mathbb{Q} \text{ with } \frac{a}{b} \neq \frac{a_n}{b_n} \text{ and } b \leq b_n.$$

Factoring with continued fractions

Remember factoring of n via factor bases:

- Use a **factor base** $\mathcal{B} = \{-1, p_1, \dots, p_L\}$
- Collect squares that are **\mathcal{B} -smooth**: $x_k^2 \bmod n = Y_k = \prod_t p_t^{e_{kt}}$
- If Y_k is small, it is more likely to factor over \mathcal{B} successfully!

Continued fraction factoring

Let $\frac{a_k}{b_k}$ be the k -th convergent of \sqrt{n} . Consider the square candidates $x_k := a_k$, so $Y_k := a_k^2 \bmod n = a_k^2 - nb_k^2$.

- This choice of x_k asserts that $Y_k = a_k^2 - \sqrt{n}^2 b_k^2 \approx a_k^2 - \frac{a_k^2}{b_k^2} b_k^2 = 0$ is fairly small
- There's an **easy algorithm** to compute the expansion of \sqrt{n} accurately

Factoring with continued fractions: Example I

Task

Factor $n = 9073$ with the continued fraction method.

- Compute convergents for $\sqrt{9073} = 95.2523 \dots$:

$$\frac{a_0}{b_0} = \frac{95}{1}, \quad \frac{a_1}{b_1} = \frac{286}{3}, \quad \frac{a_2}{b_2} = \frac{381}{4}, \quad \frac{a_3}{b_3} = \frac{10192}{107}, \quad \frac{a_4}{b_4} = \frac{20765}{218}$$

- Smallest absolute residue Y_i of $a_i^2 \bmod 9073$:

i	0	1	2	3	4	...
$x_i = a_i$	95	286	381	10192	20765	...
$Y_i = a_i^2 \bmod n$	-48	139	-7	87	-27	...

Factoring with continued fractions: Example II

- Choose factor base $\mathcal{B} = \{-1, 2, 3, 5, 7\}$
- Check smoothness of the Y_i and factorize:

$$Y_0 = (1, 4, 1, 0, 0), \quad Y_2 = (1, 0, 0, 0, 1), \quad Y_4 = (1, 0, 3, 0, 0).$$

- Combine to get squares x and y :

$$y^2 = Y_0 \cdot Y_4 = (-1 \cdot 2^2 \cdot 3^2)^2 = (-36)^2$$

$$x = x_0 \cdot x_4 = 95 \cdot 20765 \equiv 3834 \pmod{9073}$$

with $(-36)^2 \equiv 3834^2 \pmod{9073}$.

- Factor n : $\gcd(3834 + 36, 9073) = 43 \Rightarrow 9073 = 43 \cdot 211$

Wiener's attack on RSA

Wiener's attack

- **Goal:** Find private d in RSA with $N = p \cdot q$.
- **Wiener's Theorem:** d appears in convergents of $\frac{e}{N}$ if
 - primes $q < p < 2q$,
 - public exponent $e < \varphi(N)$,
 - small private exponent $d < \frac{1}{3}\sqrt[4]{N}$.

RSA private key:
primes p, q , exp. d

RSA public key:
 $\text{mod } N = pq$,
exp. $e \cdot d \equiv 1 \text{ mod } \varphi(N)$

Useful property of continued fractions

Let $\alpha \in \mathbb{R}$ and $a, b \in \mathbb{Z}$, such that $\left| \alpha - \frac{a}{b} \right| < \frac{1}{2b^2}$.

Then $\frac{a}{b}$ is a convergent of the continued fraction expansion of α .

Wiener's attack on RSA: Proof of Wiener's theorem

- **Idea:** there exists some $k \in \mathbb{Z}$ with $ed - k\varphi(N) = 1$, so $\left| \frac{e}{\varphi(N)} - \frac{k}{d} \right| = \frac{1}{d\varphi(N)}$; that means, $\frac{e}{\varphi(N)}$ **approximates** $\frac{k}{d}$.
- $\varphi(N)$ is private, but we can **use N instead of $\varphi(N)$** :
 $|N - \varphi(N)| = |N - (p-1)(q-1)| = |p+q-1| < 3\sqrt{N}$, so
$$\left| \frac{e}{N} - \frac{k}{d} \right| = \dots \leq \frac{3k}{d\sqrt{N}} < \frac{1}{2d^2}. \quad (\text{using } k < d < \frac{1}{3}\sqrt[4]{N})$$
- The property now says that $\frac{a}{b} = \frac{k}{d}$ **is a convergent of** $\alpha = \frac{e}{N}$.
- **Attack:** Compute continued fraction convergents of $\frac{e}{N}$ and test all candidates d for $(m^e)^d \equiv m \pmod{N}$ with some m .

Wiener's attack on RSA: Example

- Public: $N = 9449868410449$ and $e = 6792605526025$. Assume that d satisfies $d < \frac{1}{3}\sqrt[4]{N} \approx 584$.
- Perform Wiener's attack by computing convergents $\frac{a_i}{b_i}$ of $\frac{e}{N}$:

$$\begin{array}{cccc}\frac{a_0}{b_0} = \frac{1}{1}, & \frac{a_1}{b_1} = \frac{2}{3}, & \frac{a_2}{b_2} = \frac{3}{4}, & \frac{a_3}{b_3} = \frac{5}{7}, \\ \frac{a_4}{b_4} = \frac{18}{25}, & \frac{a_5}{b_5} = \frac{23}{32}, & \frac{a_6}{b_6} = \frac{409}{569}, & \frac{a_7}{b_7} = \frac{1659}{2308}, \dots\end{array}$$

- Testing each denominator as possible d reveals $d = 569$.

Factoring with Elliptic Curves



Pollard's $p - 1$ Method, Revisited

Recall Pollard's $p - 1$ method to factor $n = p \cdot q$:

- 1 Pick $a \in \mathbb{Z}_n^*$ and $k \in \mathbb{N}$, e.g., $k = B!$ for bound B
- 2 If k is such that $p - 1 \mid k$ and $p \nmid a$, then

$$a^k \equiv 1 \pmod{p}.$$

- 3 Consequently, p divides both n and $a^k - 1$. If

$$d = \gcd(a^k - 1, n) \neq 1, n$$

Success! Else, adapt B (larger if $d = 1$, smaller if $d = n$)

Fermat's theorem:
for $a \in \text{group } G$,

$$a^{|G|} = 1$$

Using different groups

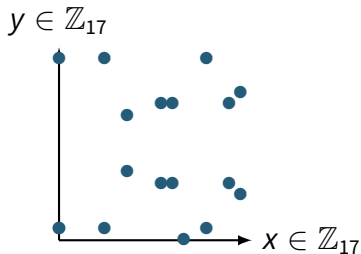
Pollard's $p - 1$ operates in subgroup $\text{mod } p$ (of structure $\text{mod } n$).

It only works if group order $|\mathbb{Z}_p^*| = p - 1$ is smooth.

Idea: \mathbb{Z}_p^* isn't the only group we know \rightarrow Elliptic Curve Group!



Modular group \mathbb{Z}_{17}^*
(order 16)



Elliptic curve group $E(\mathbb{Z}_{17})$
 $y^2 = x^3 + x + 1$ (order 18)

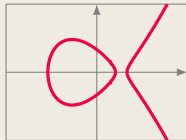
Elliptic Curve Group

Elliptic curve

= solutions (x, y) of equation in Weierstrass Form

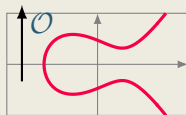
$$y^2 = x^3 + ax + b$$

where $\Delta = -16(4a^3 + 27b^2) \neq 0$.

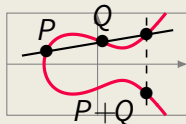


Elliptic Curve Group

Neutral element \mathcal{O} : Special point “ $(0, \infty)$ ”



Addition $P + Q$: Chord rule



How many points are in an EC group?

Order of the group E

The number of points (x, y) on E (incl. \mathcal{O}) is its **order** $|E|$.

Hasse's Theorem

The order of $E(\mathbb{Z}_p)$ is $|E| = p + 1 - t$ for some $|t| \leq 2\sqrt{p}$.

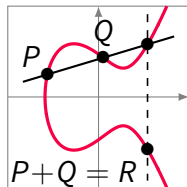
In other words: $|E(\mathbb{Z}_p)| \approx |\mathbb{Z}_p^*|$, but exact value depends on curve!

By trying different curve equations, we get different orders!

This gives us many “candidate orders” that might be smooth.

Addition in $E(\mathbb{Z}_p)$

Points $P = \begin{pmatrix} x_P \\ y_P \end{pmatrix}$, $Q = \begin{pmatrix} x_Q \\ y_Q \end{pmatrix}$, $R = \begin{pmatrix} x_R \\ y_R \end{pmatrix}$



$$P + Q = \begin{cases} Q & \text{if } P = \mathcal{O} \\ P & \text{if } Q = \mathcal{O} \\ \mathcal{O} & \text{if } P = -Q \text{ } (x_P = x_Q, y_P = -y_Q) \\ \begin{pmatrix} \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P \\ \left(\frac{3x_P^2 + a}{2y_P}\right)(x_P - x_R) - y_P \end{pmatrix} & \text{if } P = Q \text{ } (x_P = x_Q, y_P = y_Q) \\ \begin{pmatrix} \left(\frac{y_Q - y_P}{x_Q - x_P}\right)^2 - x_P - x_Q \\ \left(\frac{y_Q - y_P}{x_Q - x_P}\right)(x_P - x_R) - y_P \end{pmatrix} & \text{else} \end{cases}$$

Addition involves computing inverses $\frac{u}{v} \pmod{p}$ (=Euclid)!

Addition in $E(\mathbb{Z}_n)$, $n = p \cdot q$

Idea: Simply perform the same computations $\bmod n$ (if possible).

What can go wrong when computing $\frac{u}{v} \pmod n$?

- If $\gcd(v, n) = 1$: everything ok
- If $\gcd(v, n) = n$ (and $\gcd(u, n) = 1$): Means $P = -Q$, result \mathcal{O}
- If $\gcd(v, n) \neq n, 1$: Addition failed, but...

We've found a factor of n !

Lenstra's Elliptic Curve Method for Factorization

Repeat until successful:

- 1 Pick random curve $E(\mathbb{Z}_n) : y^2 = x^3 + ax + b$, point $P = (x_0, y_0)$

Hint: First pick $x_0, y_0, a \in \mathbb{Z}_n$, compute $b = y_0^2 - x_0^3 - ax_0 \pmod{n}$

- 2 Pick number k with many small prime factors, e.g., $k = B!$

- 3 Compute $k \cdot P = P + P + \dots + P$

Hint: Step by step: $2P$, then $3(2P)$, then $4(3!P)$, ...

- If all computations successful...bad luck, next curve
- If intermediate result \mathcal{O} ...bad luck, next curve
- If addition fails with $\gcd(v, n) = p \neq n, 1$: Success!

Runtime comparison

Using L -Notation: $L_n[\alpha, c] = \exp \left[(c + o(1)) (\ln n)^\alpha (\ln \ln n)^{1-\alpha} \right]$

$0 \leq \alpha \leq 1$: $\alpha = 0$ is polynomial; $\alpha = 1$ is exponential (wrt. $\ln n$)

- Dixon's random squares: $L_n[\frac{1}{2}, 2\sqrt{2}]$
- CFRAC: $L_n[\frac{1}{2}, \sqrt{2}]$
- Lenstra's ECM: $L_p[\frac{1}{2}, \sqrt{2}]$ (p : smallest factor of n)
- Quadratic sieve: $L_n[\frac{1}{2}, 1]$
- General number field sieve: $L_n[\frac{1}{3}, 1.923]$

Example: 1024-bit RSA n (ca. 80-bit security): $\begin{cases} L_n[\frac{1}{3}, 1.923] \approx 2^{101} \\ L_n[\frac{1}{2}, 1] \approx 2^{122} \end{cases}$

Questions you should be able to answer

1. Explain factoring with factor bases. What is the underlying idea? How are the relations collected in Dixon's Random Square algorithm? How are the relations combined to get a factorization of N ?
2. Explain the Quadratic Sieve algorithm. What is the main difference compared to Dixon's algorithm?
3. What is a continued fraction of a number? What is the n -th convergent of a number? How can continued fractions be applied to factoring?
4. Explain the idea of Wiener's attack on RSA.
5. Explain Lenstra's elliptic-curve method. What is the basic idea? What other factoring algorithm is it based on? What is the essential improvement?

Bibliography

- [1] John D. Dixon. **Asymptotically Fast Factorization of Integers**. *Mathematics of Computation* 36.153 (1981), pp. 255–260. DOI: [10.2307/2007743](https://doi.org/10.2307/2007743).
- [2] H. W. Lenstra. **Factoring Integers with Elliptic Curves**. *Annals of Mathematics* 126.3 (1987), pp. 649–673. ISSN: 0003486X. DOI: [10.2307/1971363](https://doi.org/10.2307/1971363).
- [3] Michael A. Morrison and John Brillhart. **A Method of Factoring and the Factorization of F_7** . *Mathematics of Computation* 29.129 (1975), pp. 183–205. DOI: [10.2307/2005475](https://doi.org/10.2307/2005475).
- [4] Carl Pomerance. **The Quadratic Sieve Factoring Algorithm**. EUROCRYPT. Ed. by Thomas Beth, Norbert Cot, and Ingemar Ingemarsson. Vol. 209. LNCS. Springer, 1984, pp. 169–182. DOI: [10.1007/3-540-39757-4_17](https://doi.org/10.1007/3-540-39757-4_17).
- [5] Michael J. Wiener. **Cryptanalysis of short RSA secret exponents**. *IEEE Transactions on Information Theory* 36.3 (1990), pp. 553–558. DOI: [10.1109/18.54902](https://doi.org/10.1109/18.54902).