Sieving Hardware for Factoring and ECM support

Christine Priplata

EDIZONE GmbH, Bonn, Germany

priplata@edizone.de

Outline

- Why Special Purpose Hardware for Factoring?
- Factoring Algorithms
- General Number Field Sieve and Lattice Sieving
- Hardware Sieving Devices
- Architecture of the SHARK Sieving Device and ECM
- Conclusions

Special Purpose Hardware and Estimates

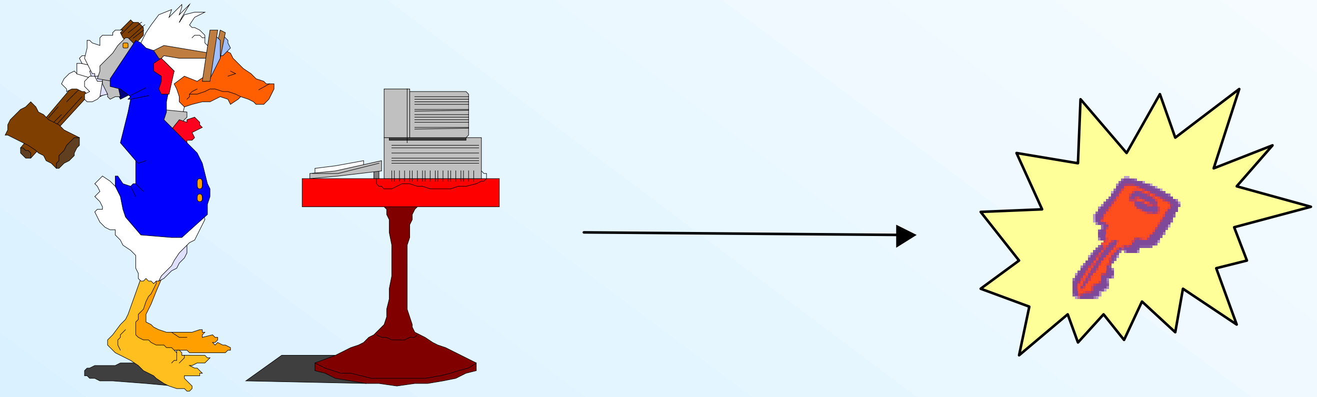
Features:

- really get an impression how far we can go by estimating the costs of an attack and receiving conclusions about necessary key lengths resp. the security of algorithms

- challenges for hardware design and development and perhaps trigger new algorithmic ideas
- helps to evaluate the „miniaturized" crypto primitives or smaller key sizes in tiny devices, which are a main topic of this workshop
- have fun

Most wanted!

To break RSA it suffices to factor the modulus N.



The **G**eneral **N**umber **F**ield **S**ieve (GNFS) is currently the best method available to attack RSA by trying to factor N.

Factoring Algorithms: Classical Choices

Factoring Problem: Find non-trivial factor p of composite integer N.

- **Trial Division**
  - nice and still relevant method to find the smallest prime factors of N (e.g. within GNFS for the cofactors)
  - just divide N by the primes 2, 3, 5, 7 ...

- **Pollard p-1 Method**
  - good to find small factors of N, faster than trial division
  - for N with prime factor p such that p-1 has many small prime factors
  - N is said to be **B-smooth** iff all prime divisors of N are smaller than B

  Method: Choose an integer B and try if $\gcd(2^{\mathrm{lcm}(1,2,\ldots,B)} - 1, N)$ is a proper factor of N.

  - depends on the size of the smallest prime factor of N

Factoring Algorithms: Elliptic Curve Method (ECM)

- **Elliptic Curve Method (ECM)**

  - also depends on the size of the smallest prime factor of N
  - good for medium size factors
  - instead of using $\mathbf{F}_p^*$ as for Pollard p-1 one uses groups $E(\mathbf{F}_p)$ of elliptic curves giving many more candidates for the same p

  Method: Fix some $B \in \mathbb{N}$ and some elliptic curve E over $\mathbf{F}_p$ with a point $P \neq O$. Try to calculate $\mathrm{lcm}(1,2,3,\dots,B) \cdot P$ modulo N. If this does not detect a factor of N, repeat with some other elliptic curve E.

  - acceleration through second stage
  - ECM factoring record: 222 bit (Dodson, August 2006)
  - besides MPQS, ECM is the method of choice for the cofactorization within GNFS (e.g. for the SHARK sieving device for 1024 bit integers: to find factors of 40 bit size within cofactors of 200 bit size)

Factoring Algorithms: Sieving Methods

Find enough of certain congruences mod N within a sieving step and finally solve a huge sparse system of linear equations over $\mathbf{F}_2$.

- Quadratic Sieve (**QS**): start with one quadratic polynomial
- Multiple Polynomial Quadratic Sieve (**MPQS**): use many polynomials yielding smaller sieving areas per polynomial
- General Number Field Sieve (**GNFS**): polynomials of higher degree involved, sieving in algebraic number fields, method of choice for general N of more than 100 digits (350 bit)

Note: GNFS currently also best to attack DL problem in $\mathbf{F}_p{}^*$
for comparable bit sizes -> same designs of sieving devices

Factoring Records 1 (all in Software)

- First big success of the number field sieve (SNFS) in 1993:
  - factorization of $F_9=2^{512}+1$ (Lenstra, Lenstra, Manasse, Pollard), also called the ninth Fermat number, factors of 7, 49 and 99 digits, could use a very special number ring, entire factorization took four months

- 512 bit in 1999, CWI Report by Cavallar, Lionen, Te Riele, Dodson, A.K. Lenstra, Montgomery, Murphy, ftp://ftp.cwi.nl/pub/CWIreports/MAS/MAS-R0007.ps.Z

- RSA-160 (532 bit) in April 2003 by Franke, Kleinjung et al.

- RSA-576 (576 bit, 173 digits) in December 2003 by Franke, Kleinjung et al.

- 545 bit number in December 2003 by NTT et al. (Aoki team)

Factoring Records 2 (all in Software)

Most recent contributions for GNFS (works for general integers):

- 582 bit number in May 2005 by NTT et al. (Aoki team)
- **RSA-200 (663 bit, 200 digits) in May 2005** by Franke, Kleinjung et al., approx. one year relation collection step and four months matrix step
  http://www.crypto-world.com/announcements/rsa200.txt
- RSA-640 (640 bit) in November 2005 by Franke, Kleinjung et al.

SNFS Record (works for very special integers):

- 911 bit number in January 2006 by NTT et al. (Aoki team)

ECM Record:

- 222 bit (67 digits) in August 2006 by Dodson
  http://www.loria.fr/~zimmerma/records (ECMNET)

GNFS Steps

1. Polynomial selection: find suitable $f_0,f_1 \in \mathbb{Z}[x]$

2. **Relation collection** (often also called **sieving step**):
   find enough pairs $(a,b) \in \mathbb{Z}^2$

3. Matrix generation: put the relations in the columns and
   reduce the size of the matrix by eliminating
   some rows and columns

4. **Matrix step**: solve $M \cdot v = 0 \bmod 2$

5. Square root step: postprocess the solutions and find a factor

The expensive steps are 2. and 4.

GNFS Polynomials

Choose two irreducible, coprime polynomials $f_0,f_1 \in \mathbb{Z}[x]$ such that

$$\exists\, m \in \mathbb{Z}: \quad f_0(m) \equiv f_1(m) \equiv 0 \quad \mod N.$$

Let F_0 and F_1 be the homogenized polynomials of f_0 and f_1.

Now we look for numbers $a,b \in \mathbb{Z}$ such that $F_i(a,b)$ has a smooth prime ideal decomposition in the number field $\mathbb{Q}[x]/(f_i(x))$ for i=0,1. These will be our relations that give the columns of a matrix to be solved modulo 2.

We will find many such pairs (a,b) by sieving.

GNFS Matrix 1

Objective: Let N be a number without small factors.
Find a factor of N.

Idea: Find c,d $\in \mathbb{N}$ such that $c^2 \equiv d^2 \pmod{N}$.
If $c \neq \pm d \pmod{N}$, then gcd(c-d,N) is a factor of N.

Every solution of the established matrix corresponds to such a pair (c,d).

Every column of the matrix corresponds to a pair (a,b) from the relation collection step. More relations (a,b) give more columns in the matrix and therefore more solutions (c,d).

GNFS Matrix 2

Every row in the matrix corresponds to a prime ideal -
the upper part to the prime ideals in the number field $\mathbb{Q}[x]/(f_0(x))$,
the lower part to the prime ideals in the number field $\mathbb{Q}[x]/(f_1(x))$.

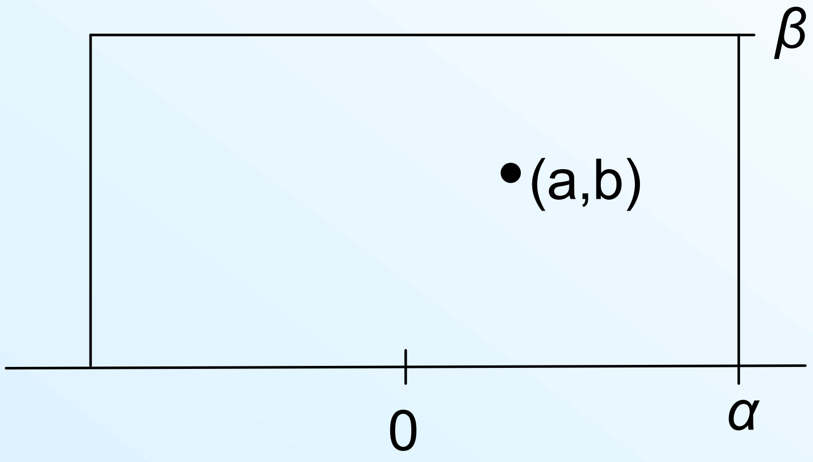
A column more precisely corresponds to a prime(ideal) decomposition of $F_0(a,b)$ in $\mathbb{Q}[x]/(f_0(x))$ and of $F_1(a,b)$ in $\mathbb{Q}[x]/(f_1(x))$.

To receive more columns than rows, these decompositions shouldn‘t have too many different prime ideals, i.e. the $F_i(a,b)$ should be smooth.

Finally square numbers c^2 and d^2 are constructed from the $F_i(a,b)$, therefore it is enough to look at the exponents mod 2 and the matrix consists of zeros and ones.

Sieving

Choose a smoothness bound $L \in \mathbb{N}$ and $\alpha, \beta \in \mathbb{N}$.
Look for suitable coprime $a,b \in \mathbb{Z}$ in a rectangle:

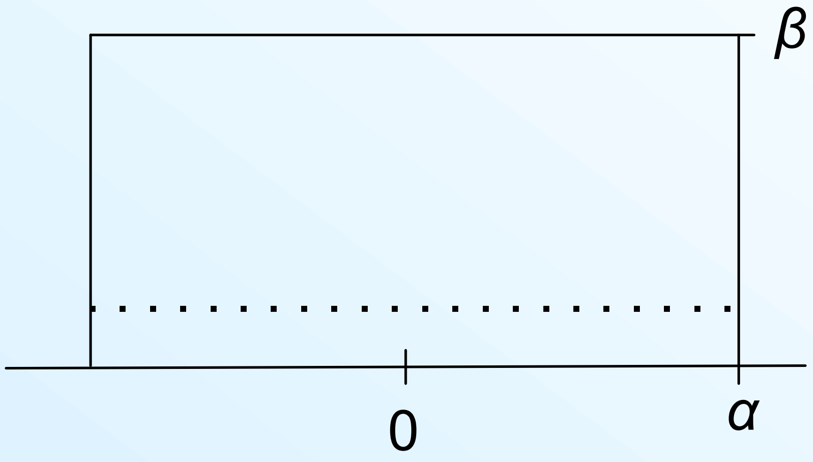


Naïve: Check for all coprime (a,b) in the rectangle, whether $F_0(a,b)$ and $F_1(a,b)$ are both L-smooth. The two factorizations will give one column of the matrix.

Line Sieving

Choose a factor base

$FB_0 := \{ (p,r) \in \mathbb{Z}^2 \mid p<L$ and $(p,x-r)$ is a prime ideal in $\mathbb{Q}[x]/(f_0(x)) \}$

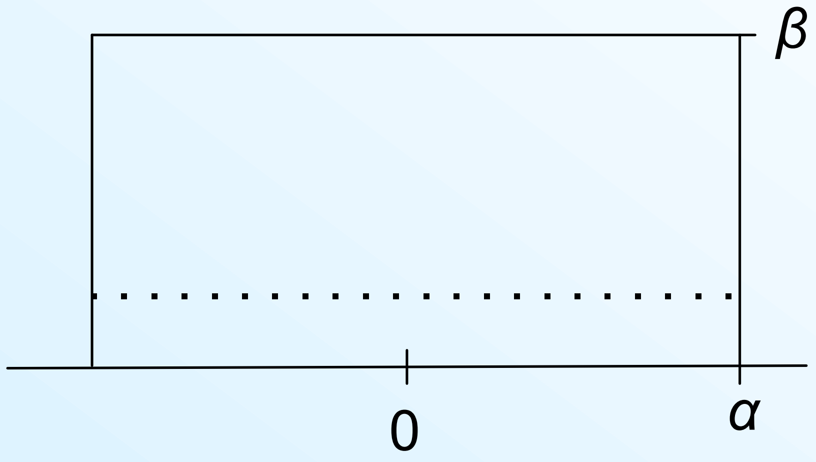


For every horizontal line in the rectangle and every element of FB_0 add log(p) to the appropriate position.

There are 2 sieves with 2 factor bases. A pair (a,b) is a potential relation (also called sieving report), if it survives both sieves.

Better Line Sieving

Choose L' smaller than L and get smaller factor bases. This means less line sieving but more factoring of the cofactors.



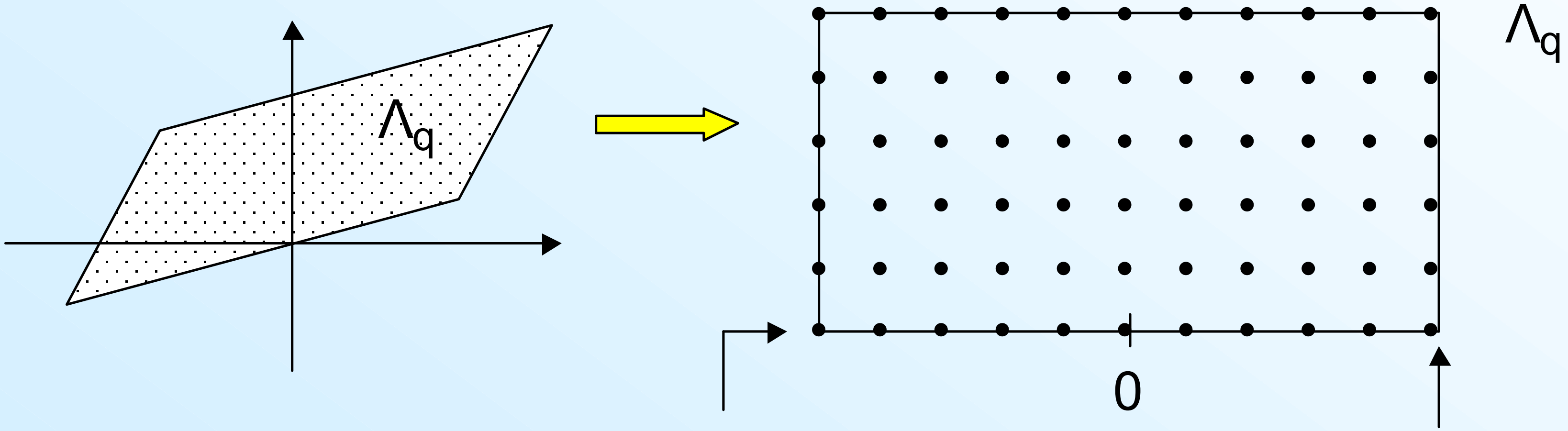
A survivor (a,b) produces a sieving report, if both $F_0(a,b)$ and $F_1(a,b)$ are L-smooth.

Lattice Sieving 1

Choose an element $(q,s) \in FB_0 \backslash FB_0'$ (“special q”), this means $L' < q < L$. We can associate the following lattice to this element (q,s):

$$\Lambda_q := \{ (a,b) \in \mathbb{Z}^2 \;\; | \;\; q \,|\, a+bs \}$$

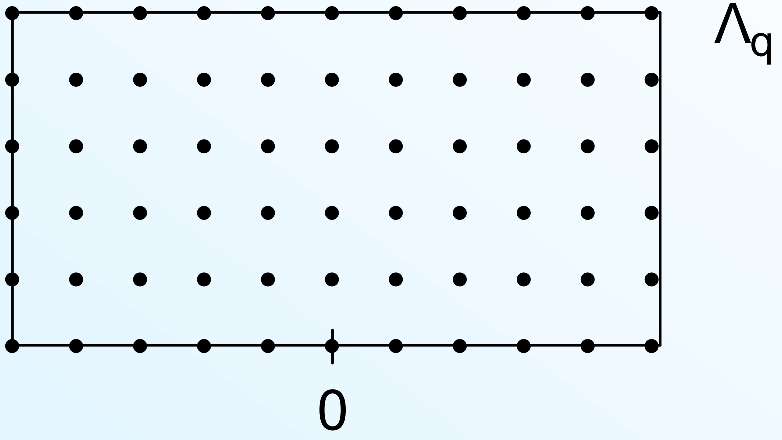
For every point $(a,b) \in \Lambda_q$ the number $F_0(a,b)$ is divisible by q.
Choose a rectangle consisting of lattice points of Λ_q.



e.g. width 2^{15} points of Λ_q

e.g. height 2^{14} points of Λ_q

Lattice Sieving 2



For every (p,r)$\in$FB$_0$' intersect Λ_q with Λ_p and add log(p) to every point of the resulting lattice.

A point (a,b) is a survivor, if its sieving sum is large enough. This bound can depend on the pair (a,b).

For every survivor test if F_0(a,b)/q is L-smooth. Then sieve Λ_q with the factor base FB$_1$' and test if F_1(a,b) is L-smooth.

Sieving Step of GNFS in Hardware

A short history of hardware sieving devices:

- 1999: TWINKLE
- 2003: TWIRL
- 2004: YASD
- 2005: SHARK

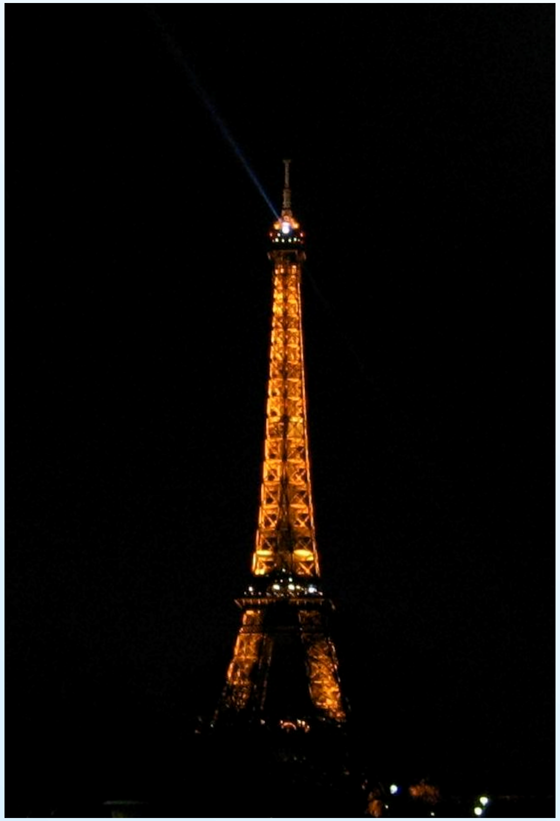
Supporting GNFS sieving with cofactorization by ECM:

- first ECM implementation on FPGA for proof-of-concept including first estimates for ASICs at SHARCS 2005

© Christine Priplata, EDIZONE

TWINKLE

TWINKLE by A. Shamir, optical device for 512 to 768 bit (CHES 1999)



TWIRL

TWIRL by A. Shamir and E. Tromer, pipelined architecture for 1024 bit RSA (Crypto 2003)

YOU CAN'T TOP THE TASTE OF A
Cadbury Twirl
42g NET
2 BARS-SWIRLS AND CURLS OF Cadbury DAIRY MILK MILK CHOCOLATE
CHERRY RIPE
THE BIG CHERRY TASTE
RIPE JUICY CHERRIES AND COCONUT IN OLD GOLD RICH DARK CHOCOLATE
YOU CAN'T TOP THE TASTE OF A
Cadbury Twirl
42g NET
2 BARS-TWIRLS AND CURLS OF Cadbury DAIRY MILK MILK CHOCOLATE

Yet **A**nother **S**ieving **D**evice

YASD by W. Geiselmann and R. Steinwandt, mesh sorting device for 512 to 768 bit RSA, adapts ideas of D. Bernstein (CT-RSA 2004)

SHARK

SHARK by J. Franke, T. Kleinjung, C. Paar, J. Pelzl, C. Priplata, C. Stahlke, modular lattice siever with small ASICs for 1024 bit RSA (SHARCS 2005, CHES 2005)

References for ECM and Other Links

- ECM designs, FPGA implementations and estimates for ASICs by team Pelzl, Simka, Franke, Kleinjung, Paar, Priplata, Stahlke, Drutarovsky, Fischer

- SHARCS	Feb 2005
- FCCM	Apr 2005
- Proc. IEE	Oct 2005

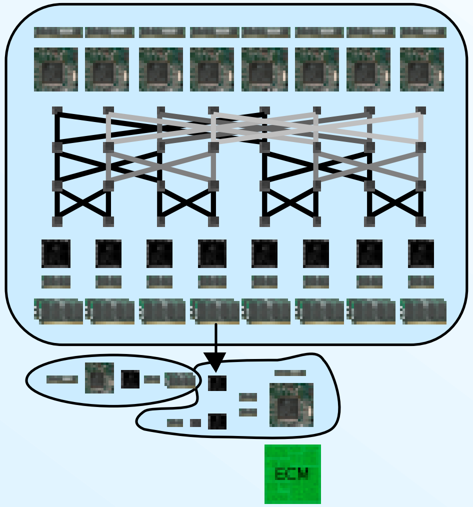
- ECM implementation of ECM in reconfigurable hardware by team Gaj, Kwon, Baier, Kohlbrenner, Le, Khaleeluddin, Bachimanchi (improvement by factor 3.4 for phase 1 and factor 5.6 for phase 2)

- SHARCS	Feb 2006
- CHES	Oct 2006

- www.sharcs.org
- www.chesworkshop.org
- www.ecrypt.eu.org (VAMPIRE Lab and AZTEC Lab)
- http://www.wisdom.weizmann.ac.il/~tromer/cryptodev/

Why SHARK?

Cracking RSA-1024



Can we do it with today´s conventional technology

for less than 1 000 000 000 US dollars?

Some Background

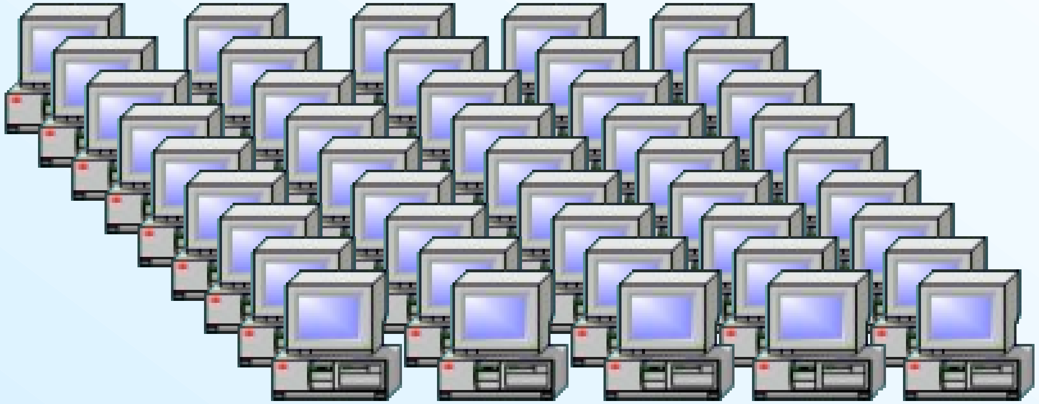
- **SHARK** comes from **S**pezial-**Ha**rdware für **R**SA-Fa**k**torisierung

- joint research started in 2004 (one year before first SHARCS)
  - University of Bonn (Jens Franke, Thorsten Kleinjung),
  - University of Bochum (Christof Paar, Jan Pelzl)
  - EDIZONE (Christine Priplata, Colin Stahlke)

- analysed different approaches/architectures in more detail (PC-based, „small“ special-purpose hardware, TWIRL-like)

- identified SHARK-like sieving as most promising w.r.t. our question and the need to study the balance between sieving and ECM, therefore in addition first ECM implementation in FPGA and for that joint work with Martin Simka (Technical University of Kosice)

Options for Relation Collection: PC-based

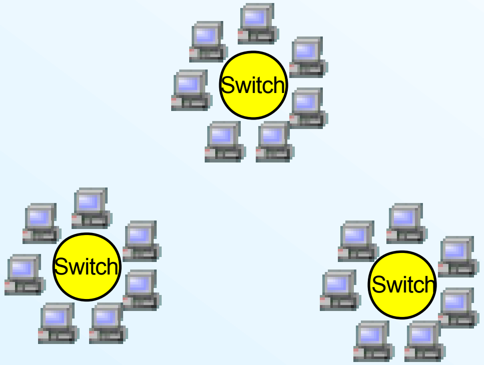


many PCs, small RAM, lattice sieving with small factor basis

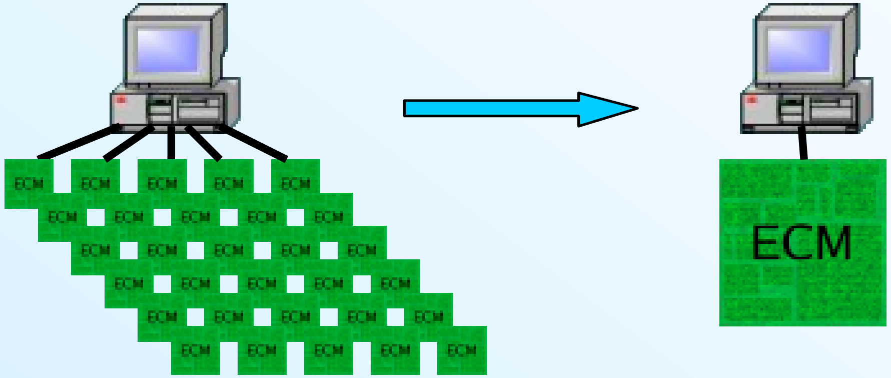
OR

cluster of connected PCs and parallelized lattice sieving to have one „unit“ with more RAM

OR



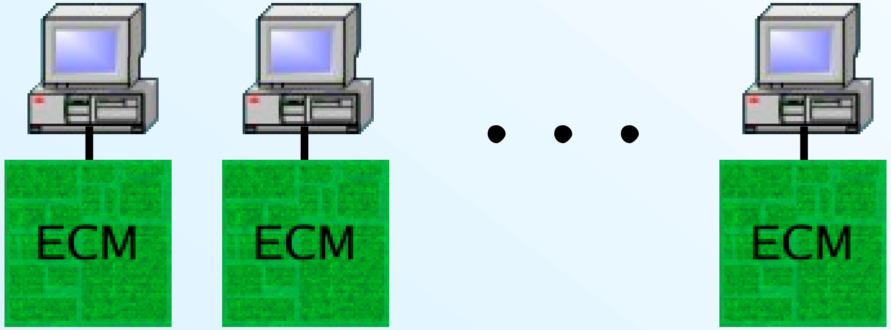
Options for Relation Collection: PC-based plus ECM



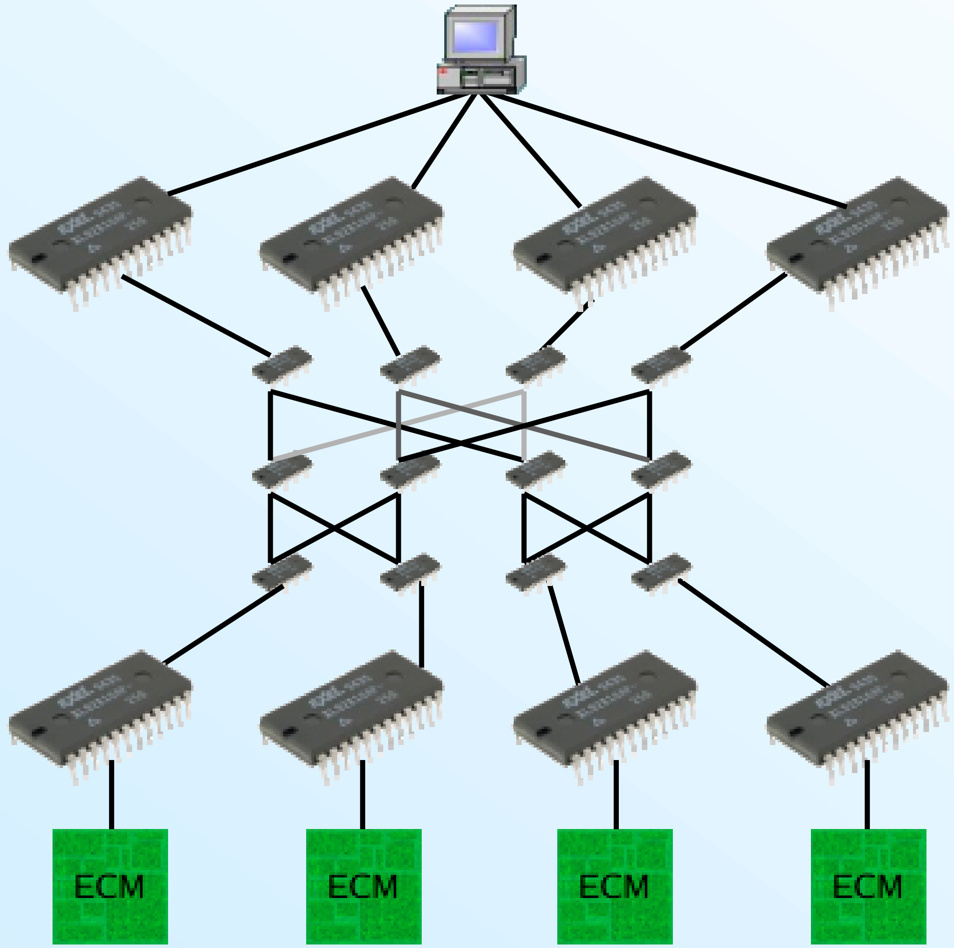
stands for special hardware of many parallel ECM units

OR

many PCs with lattice sieving and special ECM support to factor midsize numbers



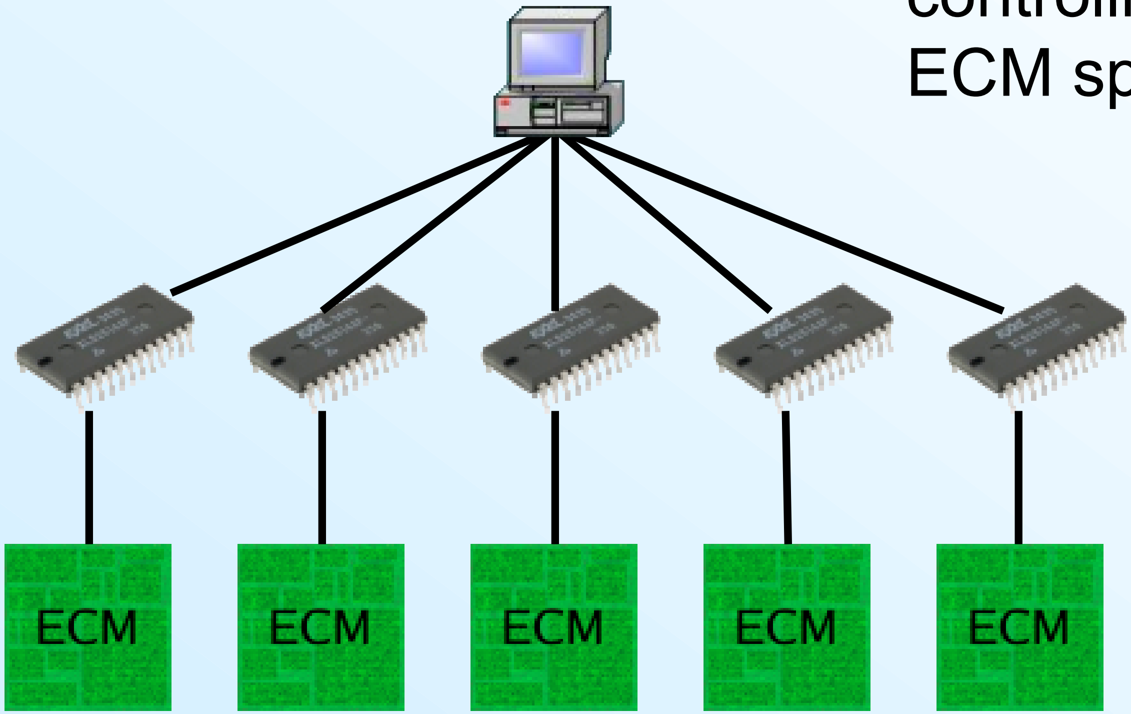
Options for Relation Collection: Small ASICs



- modular lattice siever with many small ASICs
- ECM support
- clever communication

OR

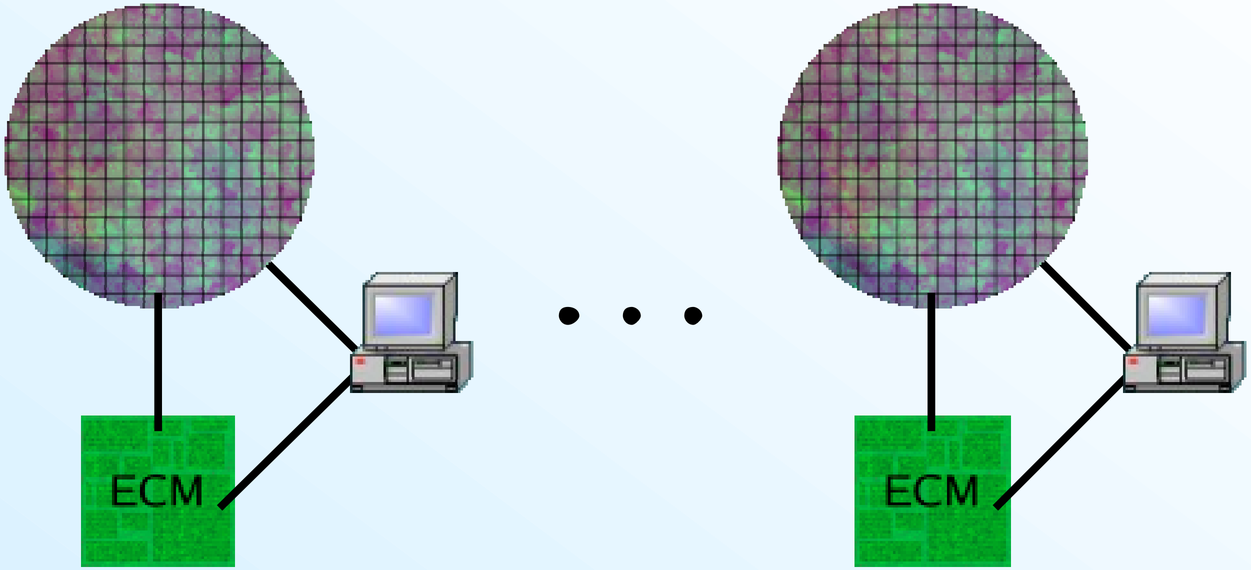
Options for Relation Collection: Small ASICs



many small processors
controlling extrmely fast
ECM special hardware

OR

Options for Relation Collection: TWIRL-like devices

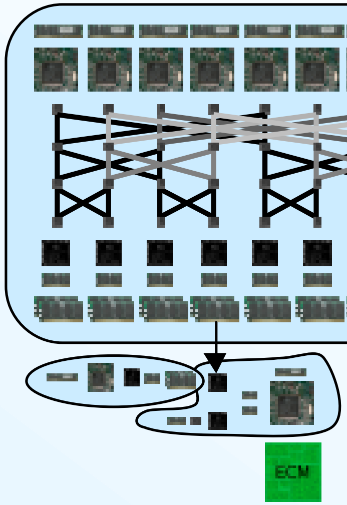


- e.g. lattice siever with „TWIRL“-technology (huge ASICs)
- ECM support to make it cheaper

SHARK Design (for 1024-bit integers, numbers from 2005)

SHARK uses lattice sieving to perform the sieving step of GNFS for a 1024-bit integer within a year for around 200 million US dollars.

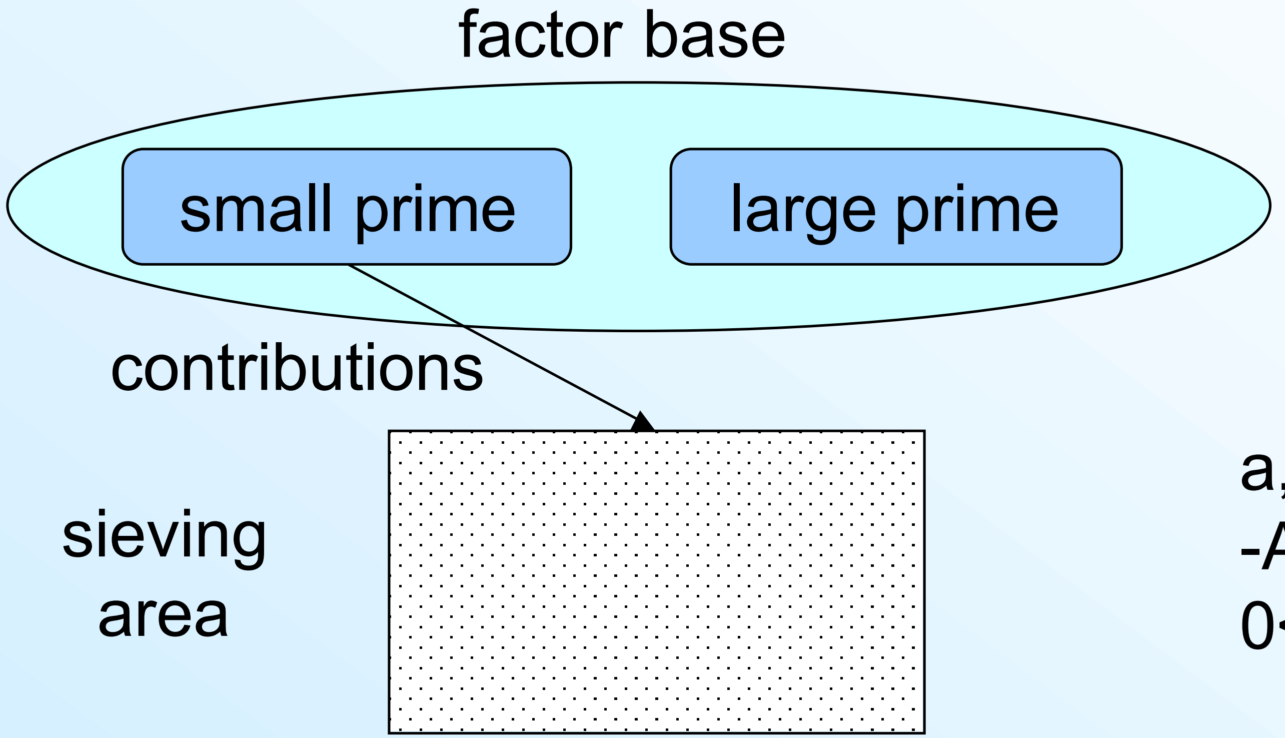
- 2300 identical machines
- small specialized ASICs
- of-the-shelf RAM
- modular architecture
- conventional data buses



The price (without development costs) is an upper bound and can be lowered considerably by changing the parameters.

GNFS Sieving

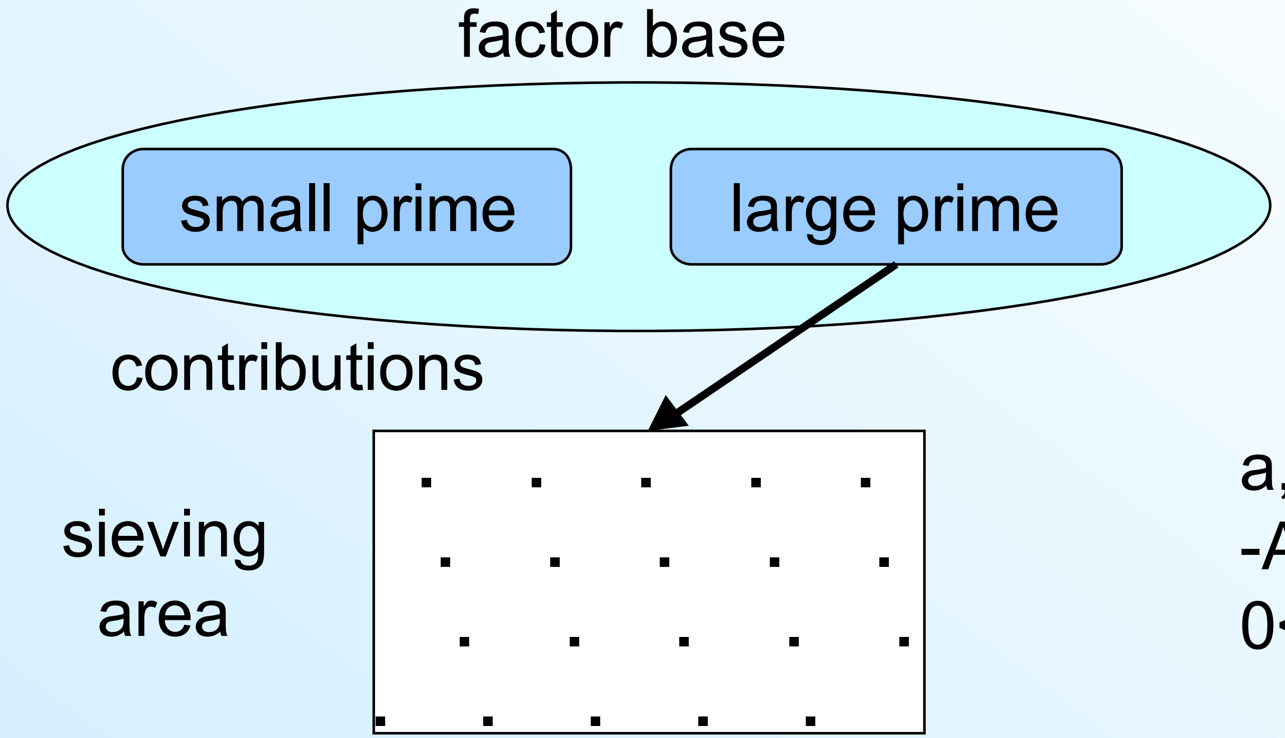
For each prime p of the factor base add contribution log(p) to certain locations in the sieving area.



$a,b \in \mathbb{Z}^2$
$-A<a<A$
$0<b<B$

GNFS Sieving

For each prime p of the factor base add contribution log(p) to certain locations in the sieving area.



$a,b \in \mathbb{Z}^2$

$-A<a<A$

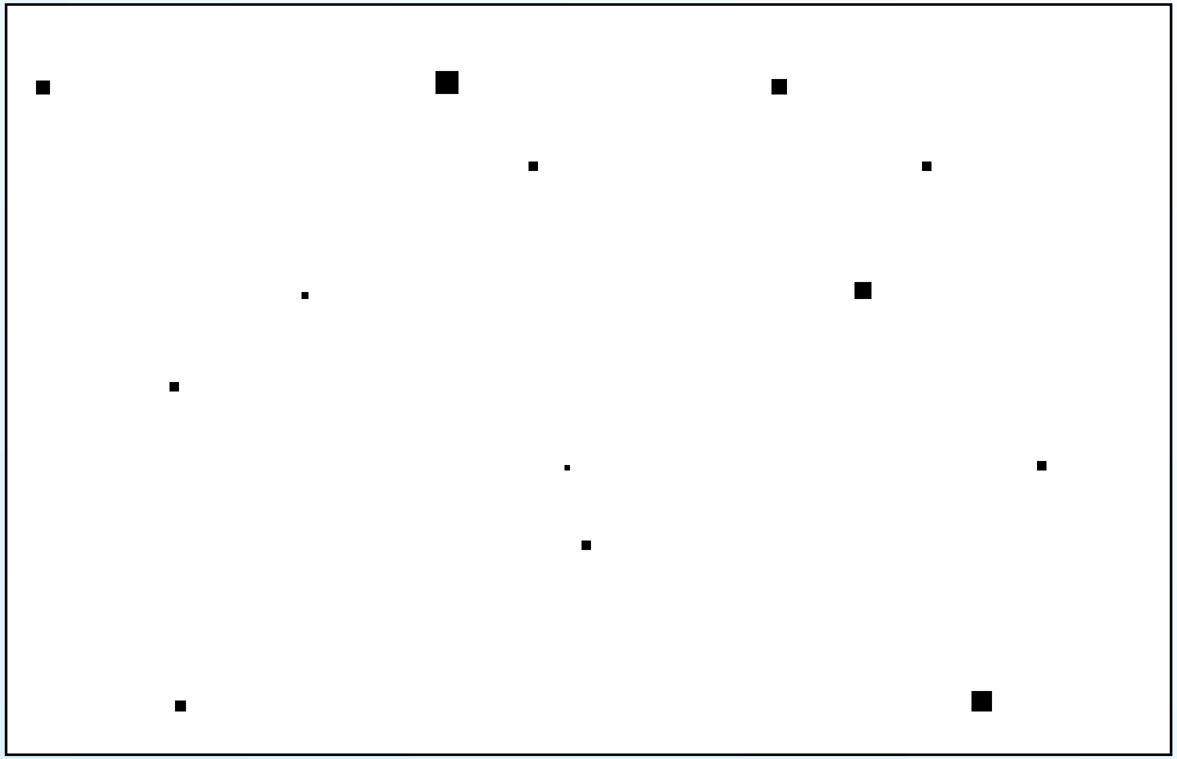
$0<b<B$

GNFS Sieving

Summing up contributions yields:

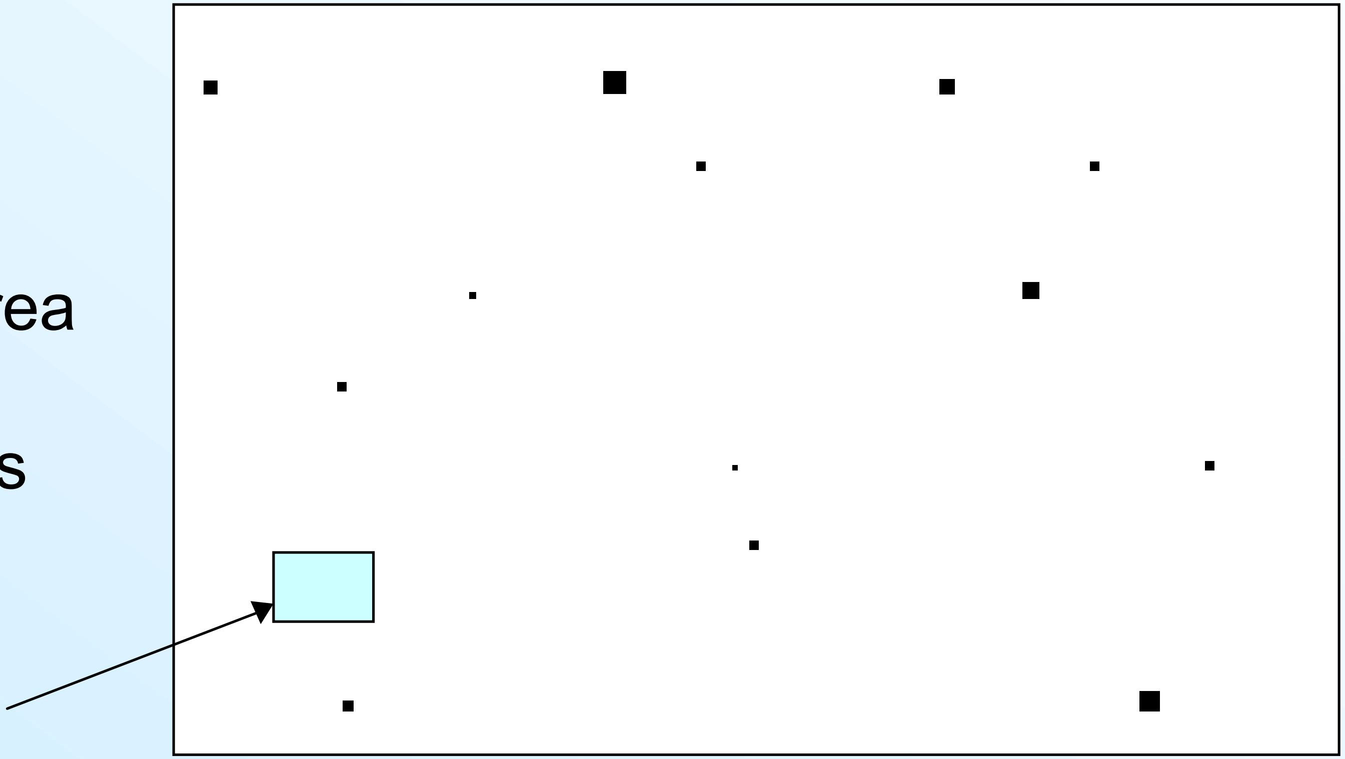
sieving area

survivors



GNFS Sieving

Summing up contributions yields:



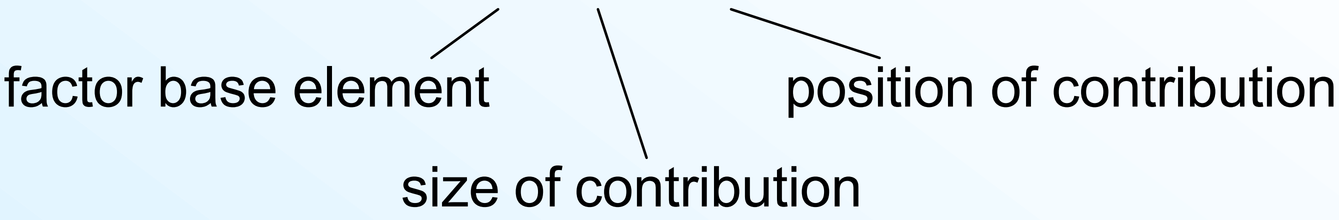
sieving area

survivors

sieving memory

Sieving Procedure

- Create contribution data (p, log p, e)



factor base element

size of contribution

position of contribution

- “Sort“ contribution data w.r.t. position e
- For each position e in the sieving area check if

$$\sum_{(p,\log p,e)} \log p \quad > \quad \text{bound depending on e}$$

Lattice Sieving

Using „Continued fractions and lattice sieving“ by Franke/Kleinjung, Sharcs´05.

Only consider most promising candidates (a,b)

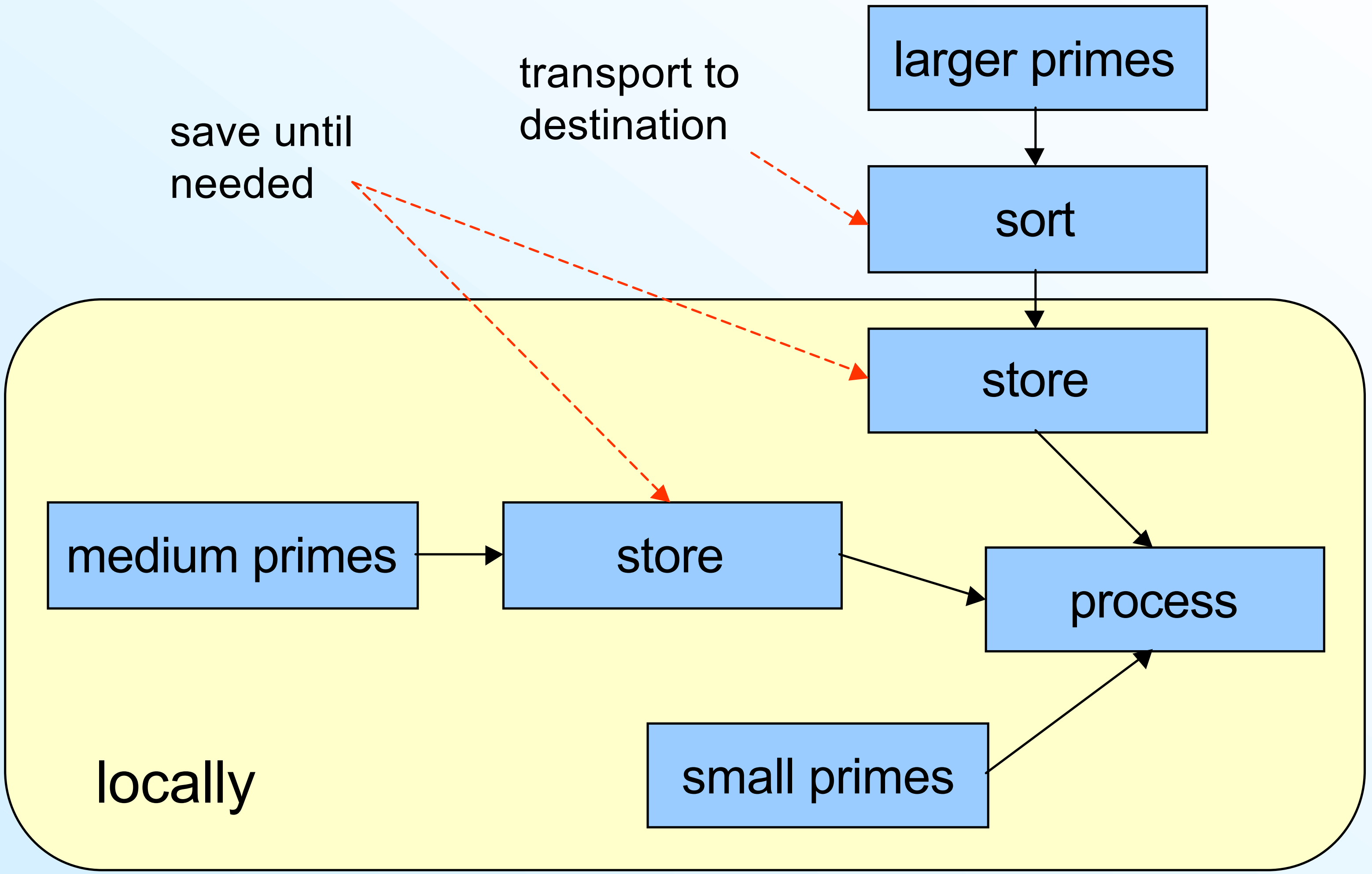
(i.e. choose large primes q, for each q consider those (a,b) where q is contributing to)

Advantage:
- more survivors (i.e. needs less sieving)

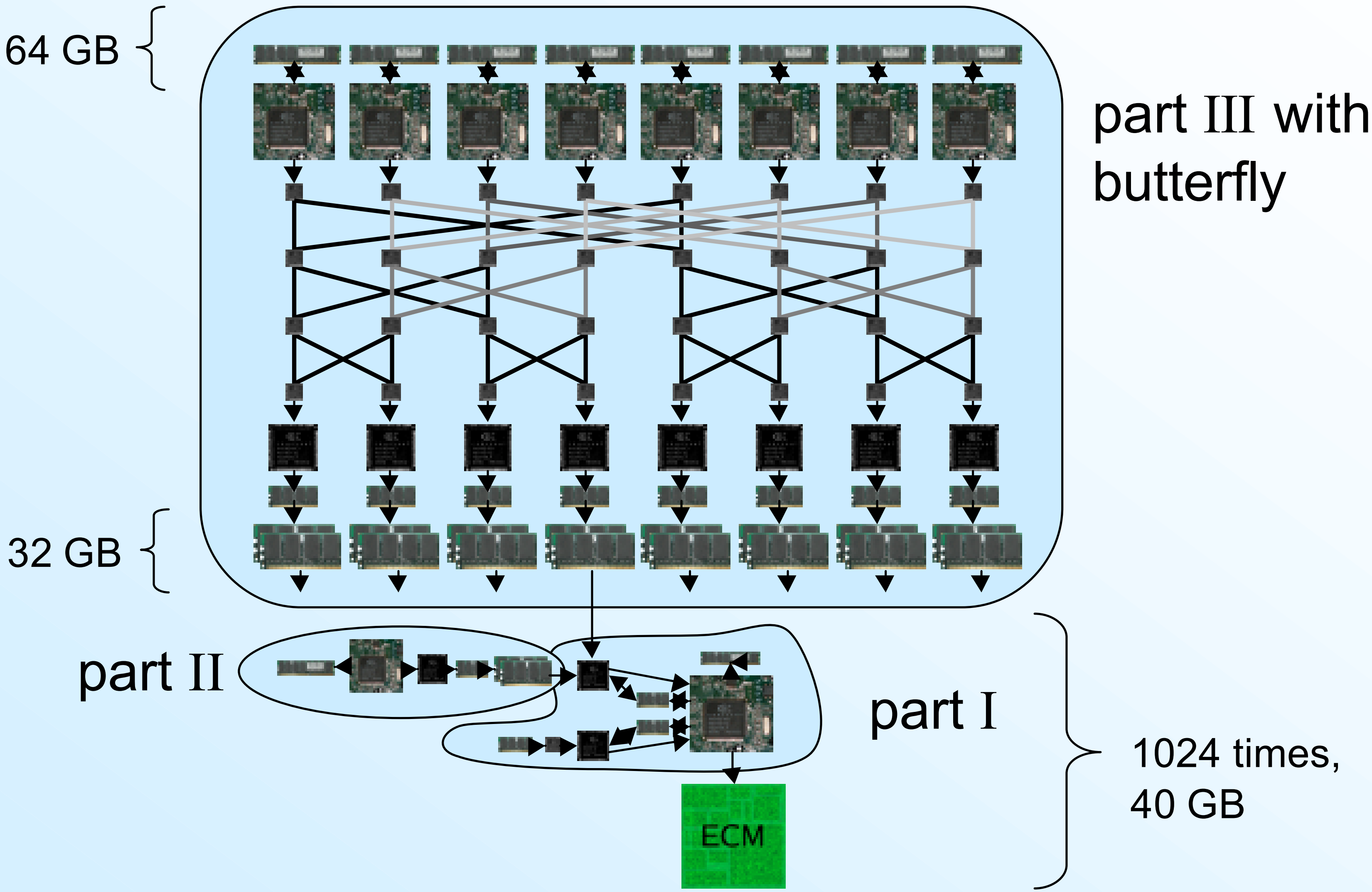
Drawbacks:
- complexer computations
- higher initialization costs
- duplicates

Lattice sieving is the most efficient sieving technique.

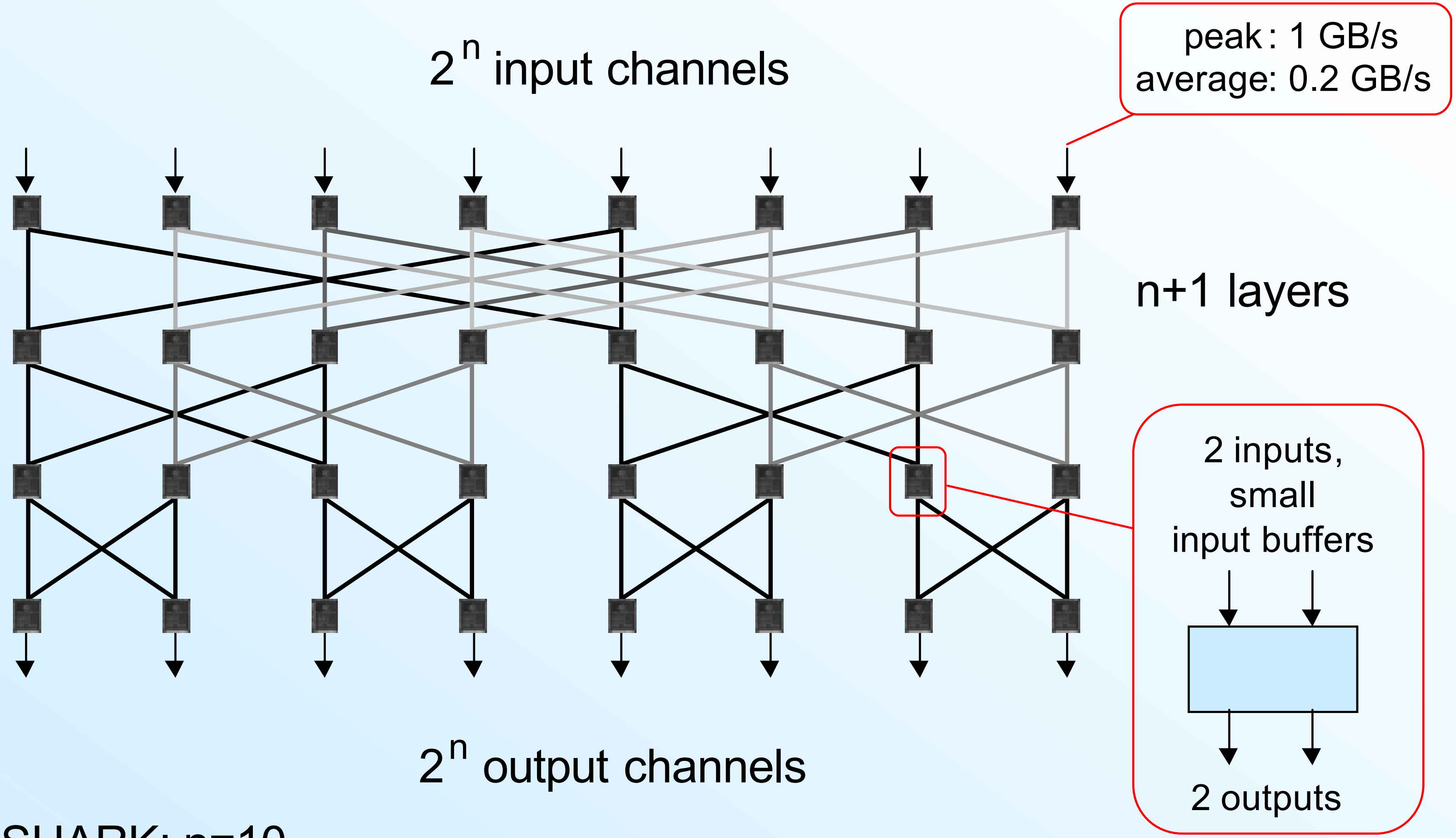
SHARK’s Main Structure



SHARK Architecture



Butterfly Transport System



2^n input channels

peak : 1 GB/s
average: 0.2 GB/s

n+1 layers

2 inputs,
small
input buffers

2 outputs

2^n output channels

SHARK: n=10

Rough Cost Estimate (as from SHARCS/CHES 2005)

1 machine:

memory:	136 GB RAM + 192 MB cache	21 000 $
processors:	1/4 wafer + transport system	9 000 $
power supply + additional electronic + cooling:		30 000 $
PCs (control) + ECM (negligible):		10 000 $
		70 000 $
power consumption: 30 kW	per year	25 000 $

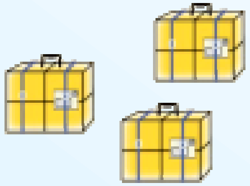
2300 machines complete the sieving step in one year and cost

160 million US $ + 60 million US $ electricity.

Summary SHARK

SHARK can perform the sieving step for a 1024-bit integer factorization in 1 year and costs around 200 million US $ (pessimistic estimate).

- modular design, small ASICs, conventional memory chips
- possible improvements: better choice of parameters, more ECM, resize transport system
- realizable with today´s technology



Concluding Remarks

- estimates of costs of special hardware for attacks to evaluate security of algorithms and secure key lengths

- challenges for factoring
  - better understanding of balance between sieving and ECM and of balance between sieving and matrix step
  - deeper understanding of good parameters for various bit lengths

- SHARK-like designs help to understand how far one can go with conventional technology