Horst-Görtz Institut
für IT Sicherheit

SMITH

A parallel Hardware Architecture for fast Gaussian Elimination over GF(2)

SHARCS '06, 03.-04. April 2006, Köln

A. Bogdanov, M.C. Mertens, C. Paar, J. Pelzl, A. Rupp

http://www.crypto.rub.de

Outline

1. **Motivation**
2. Gaussian Elimination
3. SMITH
4. Conclusions and Outlook

LSEs appearing in Cryptanalysis

- Very large but sparse
  - Resulting from factorization problem, discrete logarithm problem (RSA, DSA, ElGamal, …)
  - Size: 10^{10}x10^{10} and more but only 10^{12} nonzero entries when factoring 1024 bit integers
- Very large, not sparse
  - Size: 10^{15}x10^{15} or more necessary for secure 128 bit symmetric cipher
- „Medium“ sized, not sparse
  - Description of internal state of finite state machine
  - Resulting from linearization attempts of nonlinear LSEs
  - Stream ciphers: A5 → GSM encryption scheme, E0 → Bluetooth, …
  - Size: So that uncompressed matrix completely fits into available memory while still allowing useful operations
  - Especially interesting for embedded ciphers

Motivation

Horst-Görtz Institut für IT Sicherheit

Recent research contributions for solving very large but sparse systems in hardware

- Carry out Wiedemann‘s algorithm, Lanczos algorithm
- Leads to the problem of efficiently computing matrix-by-vector products
- Meshrouting, Pipelining
- Bernstein, Lenstra, Shamir, Tomlinson, Tromer, Geiselmann, Steinwandt, …

Motivation

Horst-Görtz Institut für IT Sicherheit

Solving „medium“ sized systems is *possible* even on standard homecomputers

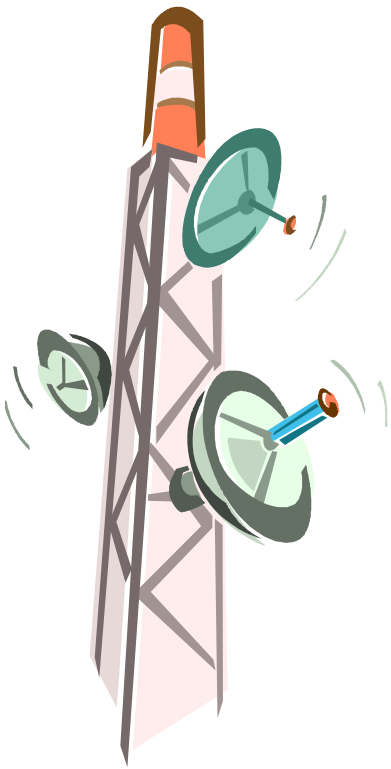
- Certain problems require solving large numbers of LSEs
- Resulting from certain classes of correlation attacks, guess and determine attacks, …
- E.g., known-plaintext attack on A5/2 by Barkan, Biham, Keller:
  - Needs few ms of known plaintext
  - Requires solving up to 2^{16} LSEs of size 656x656
    → Takes ~40 minutes on 800 MHz P3

This motivates us to investigate ways to accelerate the process of solving „medium“ sized problems

Decrypting GSM Communication

Horst-Görtz Institut für IT Sicherheit

Nice application: Solve LSEs fast enough to mount passive attack in real time



This is Eve

Eve has an antenna…

Outline

1. Motivation
2. **Gaussian Elimination**
3. SMITH
4. Conclusions and Outlook

Gaussian Elimination

- Named after Carl Friedrich Gauss
- Algorithm for solving linear systems of equations (LSEs)
- Cubic computational complexity

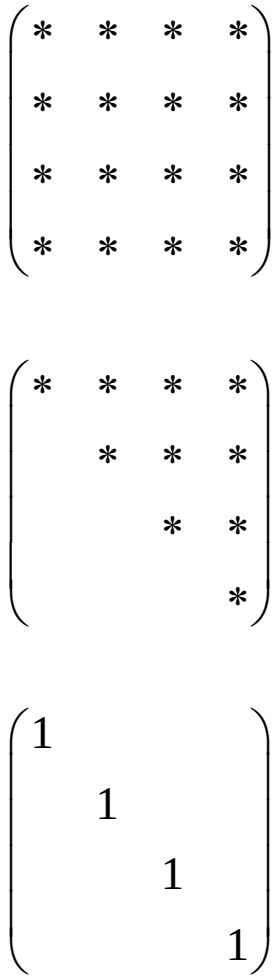
- Simplification over GF(2): All row operations are XORs

Gaussian Elimination

Horst-Görtz Institut für IT Sicherheit

Conventional Gaussian Elimination

- Eliminate all entries below the diagonal
  - Invoke suitable row operations
  - Sometimes two rows must be switched
- Yields upper triangular matrix
- Solve LSE using backward substitution
  - Substitute already obtained values for unknowns in previous equations
- Yields unit matrix



Gaussian Elimination

Modifications for efficient implementation in hardware

- Execute complete Gauss-Jordan elimination
  → Combine elimination and backward substitution
- Instead of changing the row/column used for elimination, always examine the first row/column and shift the whole matrix
- When pivoting is necessary, do not switch rows, but execute cyclic shiftup
- Last but not least: Execute all operations required for one column elimination in parallel → operate on the whole matrix!

Outline

1. Motivation
2. Gaussian Elimination
3. **SMITH**
4. Conclusions and Outlook

SMITH

Horst-Görtz Institut für IT Sicherheit

Modified Gaussian Elimination implemented in hardware yields…

- **S**calable
- **M**atrix
- **I**nversion on
- **T**ime-Area optimized
- **H**ardware

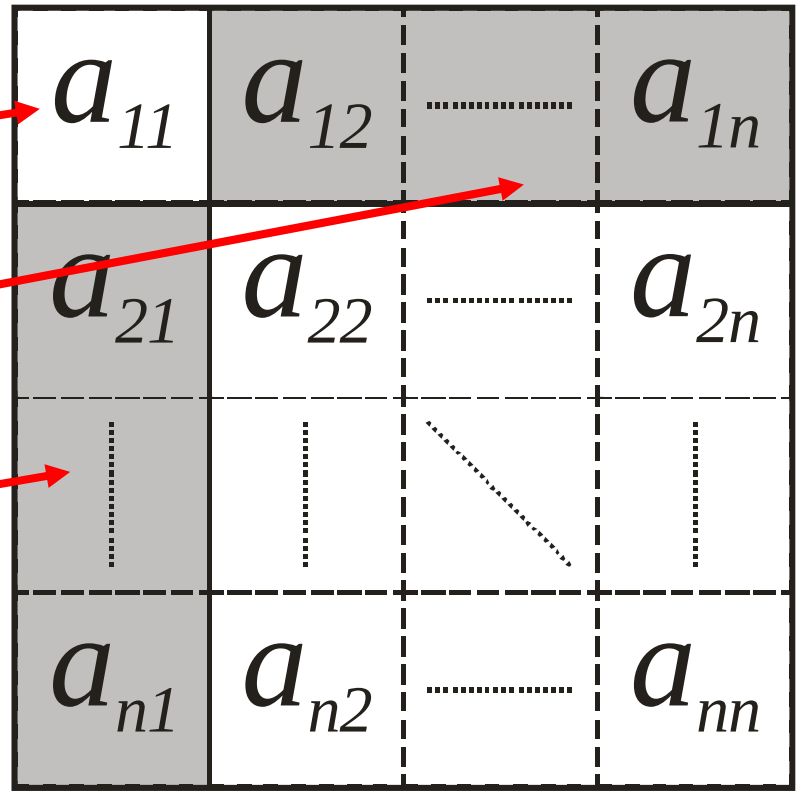
→ SMITH

SMITH

Horst-Görtz Institut für IT Sicherheit

Architecture:

- Pivot Element
  - a_{11} determines operation to be executed
- Pivot Row
  - a_{1i} controls cell inversion of *i*-th column
- Pivot Column
  - a_{i1} controls cell inversion of *i*-th row

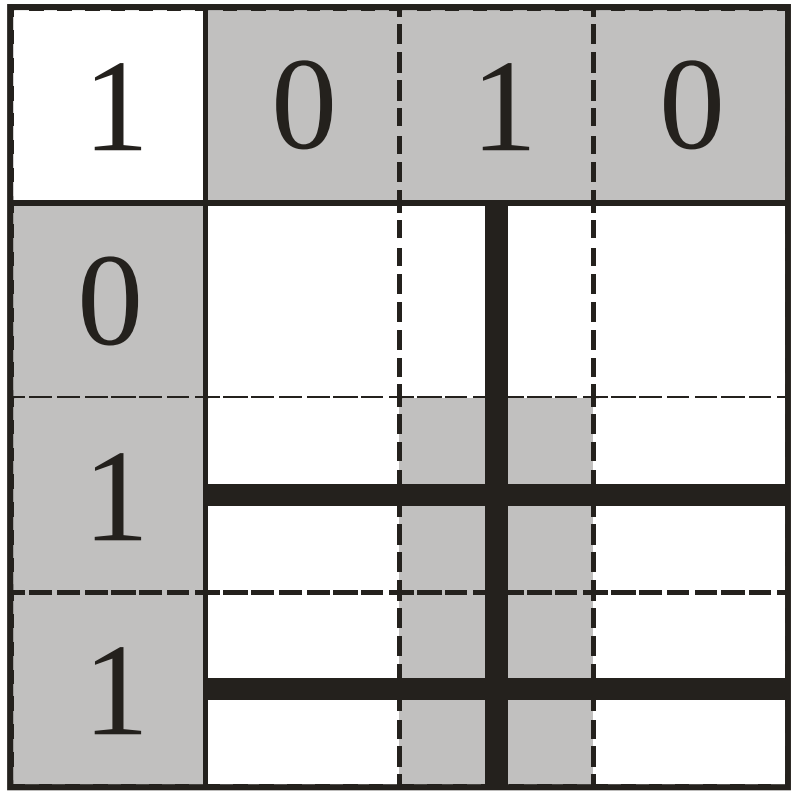


SMITH

Horst-Görtz Institut für IT Sicherheit

Instruction Set:

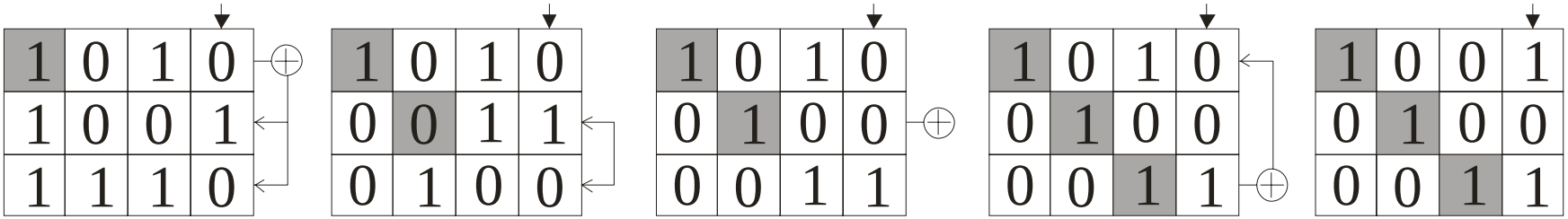
- Eliminate
  - Executed when Pivot Element is nonzero
  - Pivot Row and Pivot Column control cell inversion
  - Cyclic shiftup and shiftleft
  - Mark last row as used
- Shiftup
  - Executed when Pivot Element is zero
  - Cyclic shiftup of all rows not yet used for elimination



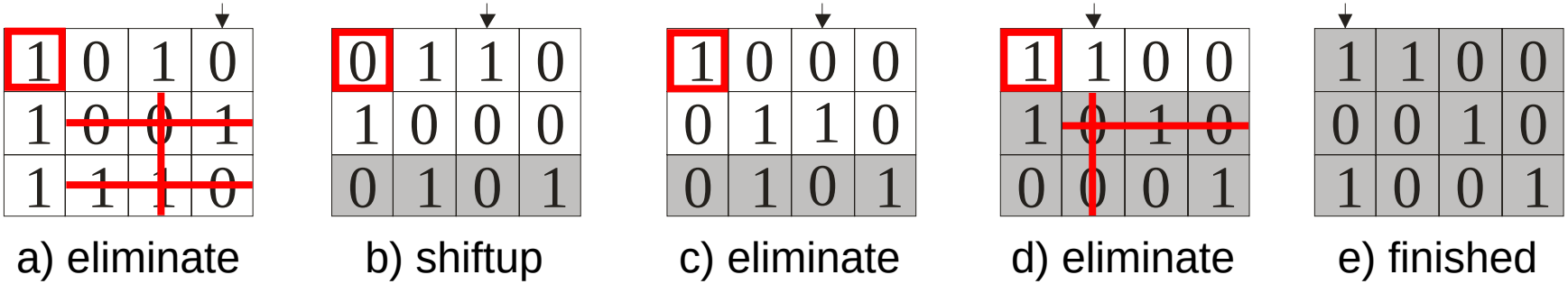
SMITH: Example

Horst-Görtz Institut für IT Sicherheit

Parallelized Gauss-Jordan elimination



Parallelized Gauss-Jordan elimination with shifting approach



SMITH: Benefits

Runtime analysis:

- Worst case
  → $O(n^2)$
- Average case
  → $2 \cdot n$

Area complexity:

→ $O(n^2)$

Require: Regular matrix $\mathbf{A} \in \{0,1\}^{n \times n}$

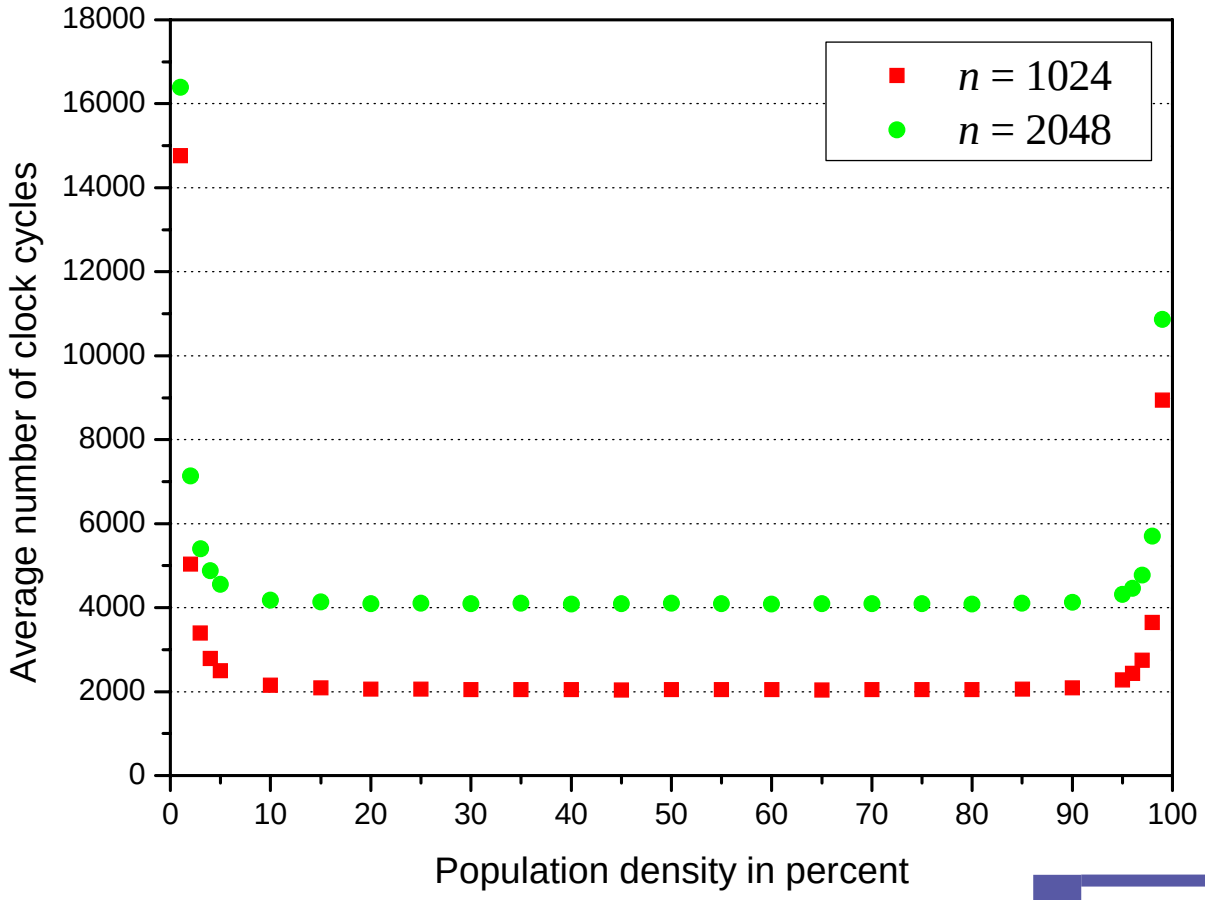
1: **for** each column $k = 1 : n$ **do**
2: **while** $a_{11} = 0$ **do**
3: $\mathbf{A} := shiftup(n - k + 1, \mathbf{A})$;
4: **end while**
5: $\mathbf{A} := eliminate(\mathbf{A})$;
6: **end for**

→ Improved Area-Time product compared to Gaussian elimination on a PC

SMITH: Benchmark

Horst-Görtz Institut für IT Sicherheit

Dependency: Runtime – Population Density



Outline

1. Motivation
2. Gaussian Elimination
3. SMITH
4. **Conclusions and Outlook**

Conclusions and Outlook

Horst-Görtz Institut für IT Sicherheit

Proof-of-concept implementation on an FPGA

- Realized on contemporary low-cost FPGA (Xilinx XC3S 1000)
- FPGA area sufficient for matrices up to 70x70
- Clock speed up to 300 MHz possible
- Solves 50x50 LSE in 333ns

Conclusions and Outlook

Horst-Görtz Institut für IT Sicherheit

Exception handling

- Not uniquely solvable $n \times n$ LSE
  - Non-regular square coefficient matrix
  - Yields zero column → Deadlock during pivoting
  - Solution: Limit number of shiftups per elimination

- Overdetermined LSE
  - Uniquely solvable → as for regular matrices
  - Solution manifold → as for square matrices
  - Unsolvable → might yield wrong result → consistency check

Conclusions and Outlook

Horst-Görtz Institut für IT Sicherheit

Further matrix operations

- Matrix inversion $\mathbf{A} \cdot \mathbf{X} = \mathbf{I}_n$
- Matrix multiplication

$$\begin{pmatrix} \mathbf{I}_n & \mathbf{A} & 0 \\ 0 & \mathbf{I}_n & \mathbf{B} \\ 0 & 0 & \mathbf{I}_n \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{I}_n & -\mathbf{A} & \mathbf{A} \cdot \mathbf{B} \\ 0 & \mathbf{I}_n & -\mathbf{B} \\ 0 & 0 & \mathbf{I}_n \end{pmatrix}$$

  - Optimized version: linear time, quadratic space

Using Strassen‘s algorithm to break down large matrix problems

- Allows solving systems larger than „SMITH size“
- Use SMITH devices to compute basic inversions and multiplications of Strassen‘s algorithm
- We are currently working on this

Conclusions and Outlook

Horst-Görtz Institut für IT Sicherheit

ASIC implementation

- 1000x1000 matrices need < $6 \cdot 10^7$ equivalent gates
  → Die size less than a conventional Pentium 4
  But: Bear power consumption and heat dissipation in mind
- Can solve up to 250000 LSEs of dimension 1000x1000 per second when clocked at 1 GHz
  But: Fast I/O is crucial

Optical implementation

- Replace long wiring by optical waveguides
- Optical coupling of large number of SMITH devices
- But: Optical technology still needs lots of research
  (Remember talk by Yusuf Leblebici)

Summary

- First (?) dedicated hardware for solving medium sized LSEs over GF(2)
- Quadratic area complexity and linear time complexity
- Direct solution of 70x70 matrices on lowcost FPGAs
  → 10 to 10^2 times faster than contemporary CPU
- ASIC implementation allows dimensions up to 1000x1000
  → 10^3 to 10^4 times faster than contemporary CPU
  (maybe use a Playstation instead?)
- Acceleration of attacks on some symmetric ciphers by several orders of magnitudes
- Can complete example attack on A5/2 in less than 100 ms
- Size restrictions could be eased by algorithmic means (Strassen) and technological improvements (Optics)

Thank you for your attention!

Horst-Görtz Institut für IT Sicherheit

Questions