Solving Binary Linear Equation Systems over the Rationals and Binaries

Benedikt Driessen, Christof Paar

Horst Görtz Institute for IT-Security
Ruhr-University Bochum, Germany

WAIFI 2012, Bochum, Germany
18.7.2012

Solve many linear equation systems over $\mathbb{F}_2$ quickly

- Given several equation systems of the form $A\vec{x} = \vec{b}$, find a way to solve them very fast
  - Binary equation systems, i.e., $A \in \mathbb{F}_2^{n \times n}, \vec{b}, \vec{x} \in \mathbb{F}_2^n$

Solve many linear equation systems over $\mathbb{F}_2$ quickly

- Given several equation systems of the form $A\vec{x} = \vec{b}$, find a way to solve them very fast
  - Binary equation systems, i.e., $A \in \mathbb{F}_2^{n \times n}, \vec{b}, \vec{x} \in \mathbb{F}_2^n$
- Practical application in cryptanalysis
  - A5/1: Solve many ($\approx 2^{40}$) equation systems with $n = 64$

Approach using a special "solving device"

- In spirit with Shamir's uncommon TWIRL/TWINKLE device, use analog hardware to do computation
  - Find $\vec{u} \in \mathbb{Q}^n$ over the rationals such that $A\vec{u} \approx \vec{b}$ with a dedicated device
  - Interpret $\vec{u}$ in order to obtain $\vec{x} \in \mathbb{F}_2^n$ which solves $A\vec{x} = \vec{b}$ over $\mathbb{F}_2$

Op-amps as building blocks

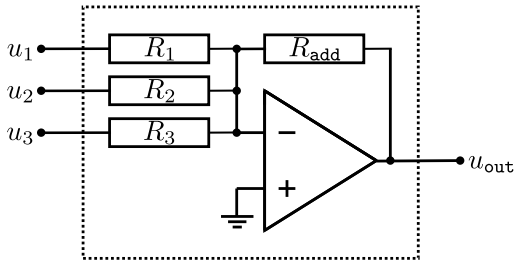


Figure: An op-amp configured as inverting adder

- The operational amplifier is configured as **inverting adder**

Op-amps as building blocks

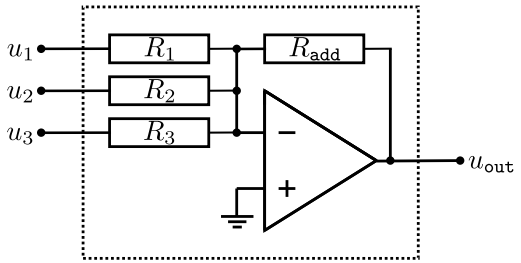


Figure: An op-amp configured as inverting adder

- The operational amplifier is configured as **inverting adder**
- All input voltages are added up:

$$u_{\text{out}} = -R_{\text{add}} \left(\frac{u_1}{R_1} + \frac{u_2}{R_2} + \frac{u_3}{R_3} \right)$$

Op-amps as building blocks

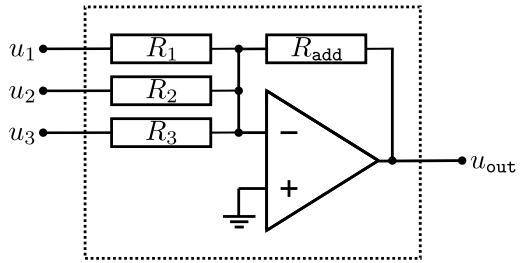


Figure: An op-amp configured as inverting adder

- The operational amplifier is configured as **inverting adder**
- All input voltages are added up:

$$u_{\text{out}} = -R_{\text{add}} \left(\frac{u_1}{R_1} + \frac{u_2}{R_2} + \frac{u_3}{R_3} \right)$$

- Simplify by setting $R_{\text{add}} = R_1 = R_2 = R_3$, thus

$$-u_{\text{out}} = u_1 + u_2 + u_3$$

A toy example

- Example circuit solves equation systems in 3 variables over $\mathbb{Q}$:

$$\begin{aligned} a_{11}u_1 + a_{12}u_2 + a_{13}u_3 &= b_1 \\ a_{21}u_1 + a_{22}u_2 + a_{23}u_3 &= b_2 \\ a_{31}u_1 + a_{32}u_2 + a_{33}u_3 &= b_3 \end{aligned}$$

A toy example

- Example circuit solves equation systems in 3 variables over $\mathbb{Q}$:

$$\begin{aligned} a_{11}u_1 + a_{12}u_2 + a_{13}u_3 &= b_1 \\ a_{21}u_1 + a_{22}u_2 + a_{23}u_3 &= b_2 \\ a_{31}u_1 + a_{32}u_2 + a_{33}u_3 &= b_3 \end{aligned}$$

- Binary coefficients of $\vec{b}$ and A are modeled as switches
  - 0: switch open
  - 1: switch closed

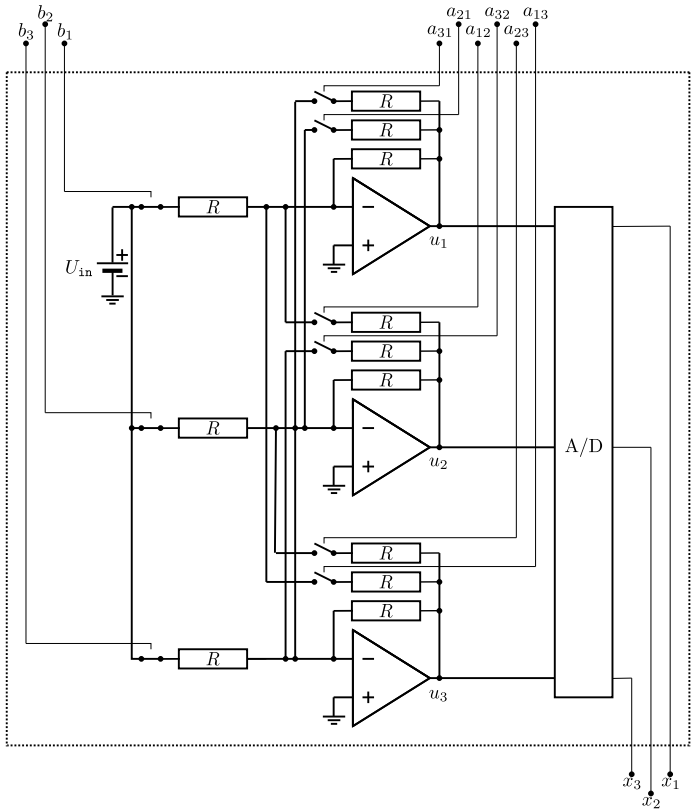
A toy example

- Example circuit solves equation systems in 3 variables over $\mathbb{Q}$:

$$\begin{aligned} a_{11}u_1 + a_{12}u_2 + a_{13}u_3 &= b_1 \\ a_{21}u_1 + a_{22}u_2 + a_{23}u_3 &= b_2 \\ a_{31}u_1 + a_{32}u_2 + a_{33}u_3 &= b_3 \end{aligned}$$

- Binary coefficients of $\vec{b}$ and A are modeled as switches
  - 0: switch open
  - 1: switch closed
- Values proportional to $\vec{u}$ measured with some accuracy at output of op-amps

A toy example: the basic circuit



$$\begin{aligned} \mathbf{u_1} &= -\left(b_1 U_{\text{in}} + a_{12}\mathbf{u_2} + a_{13}\mathbf{u_3}\right) \\ -b_1 U_{\text{in}} &= \mathbf{u_1} + a_{12}\mathbf{u_2} + a_{13}\mathbf{u_3} \end{aligned}$$

$$\begin{aligned} \mathbf{u_2} &= -\left(a_{21}\mathbf{u_1} + b_2 U_{\text{in}} + a_{23}\mathbf{u_3}\right) \\ -b_2 U_{\text{in}} &= a_{21}\mathbf{u_1} + \mathbf{u_2} + a_{23}\mathbf{u_3} \end{aligned}$$

$$\begin{aligned} \mathbf{u_3} &= -\left(a_{31}\mathbf{u_1} + a_{32}\mathbf{u_2} + b_3 U_{\text{in}}\right) \\ -b_3 U_{\text{in}} &= a_{31}\mathbf{u_1} + a_{32}\mathbf{u_2} + \mathbf{u_3} \end{aligned}$$

Figure: An analog solver for $I\vec{u} = -U_{\text{in}}(1,1,1)^T$ where I is the 3×3 identity matrix

However, there are some difficulties..

- Problems of the basic design
  - Computation over $\mathbb{Q}$ requires conversion
  - Special form of matrix A:
    - Non-zero diagonal
    - Loop-free (if understood as adjacency matrix)

However, there are some difficulties..

- Problems of the basic design
  - Computation over $\mathbb{Q}$ requires conversion
  - Special form of matrix A:
    - Non-zero diagonal
    - Loop-free (if understood as adjacency matrix)
- Implementation challenges
  - Precision of computation and A/D conversion
  - Input/output range of op-amps

Solve equation systems in two steps

- Assume we have obtained $\vec{u} \in \mathbb{Q}^n$ with "sufficient" precision, i.e., $\vec{u} \approx A^{-1}\vec{b}$
  - Small or special (e.g., low determinant as in the A5/1 attack) linear equation systems
  - Precision can be improved with *iterative refinement* methods
  - Other means..

Solve equation systems in two steps

- Assume we have obtained $\vec{u} \in \mathbb{Q}^n$ with "sufficient" precision, i.e., $\vec{u} \approx A^{-1}\vec{b}$
  - Small or special (e.g., low determinant as in the A5/1 attack) linear equation systems
  - Precision can be improved with *iterative refinement* methods
  - Other means..
- Interpret rational solution vector $\vec{u} \in \mathbb{Q}^n$ in order to obtain representation $\vec{x} \in \mathbb{F}_2^n$ which solves $A\vec{x} = \vec{b}$ over $\mathbb{F}_2$

Recall Cramer's rule

- A rational solution to $A\vec{u} = \vec{b}$ can be computed with the help of Cramer's rule, i.e.,

$$\vec{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = \frac{1}{|A|} \begin{pmatrix} |A_1| \\ \vdots \\ |A_n| \end{pmatrix}, \quad |A|, |A_i| \in \mathbb{Z}$$

Recall Cramer's rule

- A rational solution to $A\vec{u} = \vec{b}$ can be computed with the help of Cramer's rule, i.e.,

$$\vec{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = \frac{1}{|A|} \begin{pmatrix} |A_1| \\ \vdots \\ |A_n| \end{pmatrix}, \quad |A|, |A_i| \in \mathbb{Z}$$

- Due to $|A| \equiv 1 \bmod 2$, a solution over $\mathbb{F}_2$ is given by

$$\vec{x} = \begin{pmatrix} |A_1| \bmod 2 \\ \vdots \\ |A_n| \bmod 2 \end{pmatrix}, \quad |A|, |A_i| \in \mathbb{Z}$$

Looking at the binary expansion

- Unfortunately we don't know the $|A_i|$
  - But we have a vector $\vec{u}$ of "sufficiently precise" rational numbers

Looking at the binary expansion

- Unfortunately we don't know the $|A_i|$
  - But we have a vector $\vec{u}$ of "sufficiently precise" rational numbers
- The binary expansion of any u_i is always purely periodic (refer to paper for proofs)
  - Consider the binary expansion of the i-th element of $\vec{u} \approx A^{-1}\vec{b}$ with

$$u_i = c_{l-1} \cdots c_0.\overline{d_0 \cdots d_{k-1}}, \quad c_i, d_i \in \{0, 1\}$$

Another simple trick

- Rewrite the binary expansion of u_i, i.e.,

$$\begin{aligned} u_i &= c_{l-1} \cdots c_0 . \overline{d_0 \cdots d_{k-1}} \\ \Leftrightarrow 2^k u_i &= c_{l-1} \cdots c_0 d_0 \cdots d_{k-1} . \overline{d_0 \cdots d_{k-1}} \\ \Leftrightarrow 2^k u_i - u_i &= c_{l-1} \cdots c_0 d_0 \cdots d_{k-1} - c_{l-1} \cdots c_0 \\ \Leftrightarrow u_i &= \frac{c_{l-1} \cdots c_0 d_0 \cdots d_{k-1} - c_{l-1} \cdots c_0}{2^k - 1} \end{aligned}$$

Another simple trick

- Rewrite the binary expansion of u_i, i.e.,

$$\begin{aligned} u_i &= c_{l-1} \cdots c_0 . \overline{d_0 \cdots d_{k-1}} \\ \Leftrightarrow 2^k u_i &= c_{l-1} \cdots c_0 d_0 \cdots d_{k-1} . \overline{d_0 \cdots d_{k-1}} \\ \Leftrightarrow 2^k u_i - u_i &= c_{l-1} \cdots c_0 d_0 \cdots d_{k-1} - c_{l-1} \cdots c_0 \\ \Leftrightarrow u_i &= \frac{c_{l-1} \cdots c_0 d_0 \cdots d_{k-1} - c_{l-1} \cdots c_0}{2^k - 1} \end{aligned}$$

- We know two ways to express u_i, i.e.,

$$\frac{|A_i|}{|A|} \stackrel{!}{=} \frac{c_{l-1} \cdots c_0 d_0 \cdots d_{k-1} - c_{l-1} \cdots c_0}{2^k - 1} \tag{1}$$

Another simple trick

- Rewrite the binary expansion of u_i, i.e.,

$$\begin{aligned} u_i &= c_{l-1} \cdots c_0 . \overline{d_0 \cdots d_{k-1}} \\ \Leftrightarrow 2^k u_i &= c_{l-1} \cdots c_0 d_0 \cdots d_{k-1} . \overline{d_0 \cdots d_{k-1}} \\ \Leftrightarrow 2^k u_i - u_i &= c_{l-1} \cdots c_0 d_0 \cdots d_{k-1} - c_{l-1} \cdots c_0 \\ \Leftrightarrow u_i &= \frac{c_{l-1} \cdots c_0 d_0 \cdots d_{k-1} - c_{l-1} \cdots c_0}{2^k - 1} \end{aligned}$$

- We know two ways to express u_i, i.e.,

$$\frac{|A_i|}{|A|} \stackrel{!}{=} \frac{c_{l-1} \cdots c_0 d_0 \cdots d_{k-1} - c_{l-1} \cdots c_0}{2^k - 1} \tag{1}$$

- Since both denominators in Eq.1 are odd, we have

$$x_i = |A_i| \bmod 2 = c_{l-1} \cdots d_{k-1} - c_{l-1} \cdots c_0 \bmod 2 = c_0 \oplus d_{k-1}$$

Conclusion

- Initial approach for computing rational solutions has its limitations
  - Most limitations can be overcome
  - Matrix still has to be loop-free to guarantee convergence

Conclusion

- Initial approach for computing rational solutions has its limitations
  - Most limitations can be overcome
  - Matrix still has to be loop-free to guarantee convergence
- Conversion from rational to binary solution is simple
  - *Recognizing and obtaining* the period requires sufficient precision
  - Length of period is determined by denominator $|A|$

Conclusion

- Initial approach for computing rational solutions has its limitations
  - Most limitations can be overcome
  - Matrix still has to be loop-free to guarantee convergence
- Conversion from rational to binary solution is simple
  - *Recognizing and obtaining* the period requires sufficient precision
  - Length of period is determined by denominator $|A|$
- Practical use (apart from analog solver) unclear..

Thanks.

- Any questions?