Kryptoanalyse und Formale Verifikation

Gert–Martin Greuel (Boardmitglied)

Gerhard Pfister

Beteiligte Arbeitsgruppen:

- AG Greuel
- AG Pfister
- Abt. Adaptive Systeme (P. Lang, ITWM)
- AG Kunz (FB Elektro– und Informationstechnik)
- AG Informatik

Bisher beteiligte Mitarbeiter:

FB Mathematik:

- S. Bulygin (Doktorand)
- D. Ruano (Postdoc)

nicht aus Mitteln des Clusters:

- O. Wienand
- M. Brickenstein

ITWM

- A. Dreyer

Elektro– und Informationstechnik

- M. Wedler
- K. Sulimma

Projekt

Formale Verifikation mikroelektronischer Systeme

Ziel des Projekts:

Entwicklung neuartiger Algorithmen und Tools zur formalen Hardwareverifikation bei arithmetischen Schaltungs– und Prozessorblöcken.

Fortgeschrittene Gröbnerbasistechniken und ihre effiziente Implementierung unter Verwendung von SINGULAR bilden eine wesentliche Grundlage.

Projekt

Kryptoanalyse und Kodierungstheorie

Ziel des Projekts:

Durch Anwendung von Gröbnerbasis–Methoden sollen neue Ansätze in der Kryptographie und Kodierungstheorie entwickelt werden.

Zusammenarbeit mit

TU Eindhoven

Universität Kiew

Bundesamt für Sicherheit in der Informationstechnik (Bonn)

Möglichkeiten zur Modellierung des Verifikationsproblems durch Polynome

- Das Input–Output–Verhalten kombinatorischer Schaltkreise kann durch Boolesche Funktionen modelliert werden.

- Eine Boolesche Funktion $F(b_1, \ldots, b_n)$ in n Booleschen Veränderlichen:

  $$F : (\mathbb{Z}/2)^n \to \mathbb{Z}/2$$

  F lässt sich auf verschiedene Weise als Polynom in $(\mathbb{Z}/2)[x_1, \ldots, x_n]$ repräsentieren.

  Die Darstellung wird eindeutig im Booleschen Ring

  $$(\mathbb{Z}/2)[x_1, \ldots, x_n]/\langle x_1^2 + x_1, \ldots, x_n^2 + x_n \rangle.$$

- Der Äquivalenzvergleich und die Eigenschaftsprüfung lassen sich formulieren als Aufgabe, festzustellen, ob eine gegebene Boolesche Funktion f unter gegebenen Booleschen Restriktionen $f_1 = \ldots = f_k = 0$ erfüllbar ist.

- Man muss entscheiden, ob f auf der durch die Gleichungen $f_1 = \ldots f_n = 0$ in $(\mathbb{Z}/2)^n$ definierten Varietät eine Nullstelle hat.

  Das ist genau dann nicht der Fall, wenn eine minimale Gröbnerbasis 1 ist.

Was ist eine Gröbnerbasis?

- Wir fixieren auf den Monomen $\{x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n}\}$ eine Wohlordnung und schreiben alle Polynome sortiert auf. Das erste Monom eines Polynoms f nennen wir $L(f)$, Leitmonom. Seinen Koeffzienten $C(f)$, Leitkoeffizient.

  Beispiel:

  $X_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n} > X_1^{\beta_1} \cdot \ldots \cdot x_n^{\beta_n}$ wenn $\sum \alpha_i > \sum \beta_i$

  o d e r

  $\sum \alpha_i = \sum \beta_i$ und $\alpha_j = \beta_j$ für $j \leq k-1$ und $\alpha_k > \beta_k$

- Normalform von f bezüglich $G = \{f_1, \ldots, f_k\}$:

  $NF(f \mid G)$

  $h := f$

  while ($\exists$ Monom $m, L(h) = mL(f_i)$ für ein i)

  $h := h - \frac{C(h)}{C(f_i)} m f_i$

  return $(C(h)L(h) + NF(h - C(h)L(h) \mid G)$

- Ideal erzeugt von $f_1, \ldots, f_k$:

$$I := \langle f_1, \ldots, f_k \rangle = \{ \sum_{i=1}^{k} h_i f_i, h_i \text{ Polynom } \}$$

- $G = \{g_1, \ldots, g_s\}$ ist Gröbnerbasis von I, wenn

$$f \in I \Leftrightarrow NF(f|G) = 0$$

Beispiele für Gröbnerbasen:

$$\begin{array}{rcrcrcrcl} x_1 & + & x_2 & + & x_3 & - & 1 & = & f_1 \\ x_1 & + & 2x_2 & - & x_3 & + & 2 & = & f_2 \end{array}$$

$$\begin{array}{rcrcrcrcl} x_1 & + & x_2 & + & x_3 & - & 1 & = & g_1 \\ & & x_2 & - & 2x_3 & + & 3 & = & g_2 \end{array}$$

oder

$$\begin{array}{rcrcrcrcl} x_1 & & & + & 3x_3 & - & 4 & = & g_1 \\ & & x_2 & - & 2x_3 & + & 3 & = & g_2 \end{array}$$

Der Normalform–Algorithmus ist der Gauß'sche Algorithmus.

- Gröbnerbasen können sehr kompliziert sein und ihre Berechnung kann lange dauern.

  $abds^{19} + acs^2 + bcs^8 + 1$

  $abcs^8 + bds^{19} + a$

  $ads^{11} + cds^{10} + b$

  $acds^{19} + bcds^{24} + abs^4 + c$

  $abcs^{31} + 1$

- Der Algorithmus hat im Allgemeinen doppelte exponentielle Komplexität in der Anzahl der Variablen.

- Bei 0–dimensionalen Systemen hat er einfach exponentielle Komplexität (worst case).

- In unseren Anwendungen (viel mehr Gleichungen als Variable) zeigt er subexponentiales Verhalten.

Modellierung auf der Wortebene (m–Bit) führt zu Polynomen in

$$(\mathbb{Z}/2^m)[x_1, \ldots, x_r]$$

Eine Modellierung eines speziellen Multiplizierers liefert auf der Bit–Ebene 23 Elemente in der Gröbnerbasis, auf der Wortebene (4–Bit) 10 Elemente.

- Grundlage der modernen Kryptographie sind Einwegfunktionen
- **Beispiel RSA**: öffentlich: (n, e), geheim: (p, q, d)

  $n = p \cdot q \quad ggt(e, (p-1)(q-1)) = 1$,
  $d \cdot e = 1 \bmod (p-1)(q-1)$

  verschlüsseln: $C_{RSA}(m) = m^e \bmod n$

  entschlüsseln: $d_{RSA}(x) = x^d \bmod n$
- HFE (Hidden Field Equations)

  $F = \{f_1, \ldots, f_m\}, G = \{g_1, \ldots, g_s\}$ Gröbner Basis von $\langle F \rangle$
  $M \subseteq$ Monome: $m \in M \Rightarrow NF(m|G) = 0$

  öffentlich: F, M

  geheim: G

  Klartextraum: $P = \sum_{m \in M} Km$

  Verschlüsseln: $f \in P$ wählen zufällig $h_1, \ldots, h_m$

  $$c_{HFE}(f) = f + \sum h_i f_i$$

  Entschlüsseln: $d_{HFE}(f) = NF(f|G)$
- Derartige Codes sollen untersucht werden. Es soll versucht werden, bekannte Codes zu brechen,

Ziel

Spezielle hocheffiziente Gröbnerbasis Algorithmen für

- $(\mathbb{Z}/2)[x_1, \ldots, x_n]/\langle x_1^2 + x_1, \ldots, x_n^2 + x_n \rangle$ (Boolescher Ring)
- $(\mathbb{Z}/2^n)[x_1, \ldots, x_n]$

entwickeln und implementieren.

- spezielle Datenstrukturen
- spezielle Implementierungen der wichtigsten Operationen (Addition, Multiplikation, Monomvergleich)
- spezielle Strategien im Gröbnerbasis Algorithmus
- Für beide Projekte relevant