SAT Encodings for Testing Prime and Quadratic Residue

Jingchao Chen
School of Informatics, Donghua University
2999 North Renmin Road, Songjiang District, Shanghai 201620, P. R. China
chen-jc@dhu.edu.cn

***Abstract*—Here we present SAT encodings of prime and quadratic residue testing. The quadratic residue testing problem is to ask whether there exists an integer x such that $x^2 = a \bmod p$, where a and p are integers given. If an integer p is prime, there exist no integers x and y such that $x \times y = p, x > 1, y > 1$. Although several algorithms in cryptography can test prime and quadratic residue efficiently, so far no SAT solver can solve it efficiently.**

I. Introduction

The quadratic residue testing problem is formalized as follows:

Given three positive integer a and p, find an integer x such that $x^2 = a \bmod p$.

The Jacobi symbol is a generalization of the Legendre symbol, which can be used to compute quadratic residues by the law of quadratic reciprocity and many of the properties of the Legendre symbol.

The prime testing problem is formalized as follows:

Given a positive integer p, find two integers x and y such that $x \times y = p, x > 1, y > 1$.

Prime testing can implement efficiently by sieve of Eratosthenes, sieve of Euler, Solovay-Strassen primality testing algorithm [1] and Rabin-Miller testing algorithm etc.

Here we encode two problems mentioned above into SAT problems directly. By our observation, no known SAT solver can solve efficiently the resulting SAT problems. That is, they are more difficult than the original problems.

II. Encoding Primality Testing

We translate the primality testing problem into a SAT problem by encoding directly $x \times y = p, x > 1, y > 1$. The pseudocode of this encoding algorithm is shown in Algorithm 1. In this algorithm, we assume that x and y are denoted by binary variable strings $x_m, \ldots x_2, x_1$ and $y_n, \ldots, y_2, y_1$, respectively. Every binary y_k is processed, middle result f is updated. Every update need generate new two binary variable strings $z_m, \ldots z_2, z_1$ and $f_{m+n}, \ldots f_2, f_1$. Therefore, we require at least 2mn middle binary variables.

III. Encoding Quadratic Residue Testing

Encoding quadratic residue testing is the same as encoding primality testing. It can be done by replacing y with x, since $x^2 = a$ can be viewed as a special instance of $x \times y = p$. Therefore, we can get a SAT encoding algorithm for testing quadratic residue by rewriting y and p into x and a, deleting the encoding of $x \neq 1$, $y \neq 1$ in Algorithm 1, and adding the encoding of $(f_{m+n} \ldots f_2 f_1) = (a_{m+n} \ldots a_2 a_1) \bmod p$, where a and p are constants. If the sizes are the same, the SAT problem generated by quadratic residue testing is more difficult than one generated by primality testing.

Algorithm 1 Encode $x \times y = p$

x is denoted by binary variable string $x_m, \ldots x_2, x_1$
y is denoted by binary variable string $y_n, \ldots, y_2, y_1$
p has binary expansion $(p_{m+n} \ldots p_2 p_1)_2$
middle result f is denoted by binary variable string $f_{m+n}, \ldots f_2, f_1$
encode $x \neq 1$, $y \neq 1$
for $k = 1$ to n **do**
  encode $z_m \ldots z_2 z_1 = x_m \wedge y_k \ldots x_2 \wedge y_k x_1 \wedge y_k$
  encode $f_{k+m} \ldots f_{k+2} f_{k+1} := z_m \oplus f_{k+m} \ldots z_2 \oplus f_{k+2} z_1 \oplus f_{k+1}$
end for
encode $(f_{m+n} \ldots f_2 f_1) = (p_{m+n} \ldots p_2 p_1)$

References

[1] Solovay, Robert M.; Strassen, Volker: A fast Monte-Carlo test for primality, SIAM Journal on Computing,6 (1), 84C85, 1977.