Square root computation in finite fields

Arthur Braga de Campos Tinoco

Arthur Gonçalves de Moraes

Dados do artigo

- Autores: Ebru Adiguzel-Goktas, Enver Ozdemir
- Periódico: Designs, Codes and Cryptography
- Ano de publicação: 2024

Problema abordado no artigo

Encontrar raízes quadradas em campos (ou corpos) finitos.

Motivação

"Encontrar raízes quadradas em corpos finitos é algo interessante para muitos pesquisadores na teoria computacional dos números."

Objetivo

“Apresentar uma revisão de três algoritmos práticos de raiz quadrada amplamente utilizados.”

Algoritmo 1 - Tonelli–Shanks

Algorithm 1 Tonelli–Shanks Algorithm

Input: a quadratic residue a modulo p where p is an odd prime such that
$p - 1 = 2^e m$.
Output: $\sqrt{a} \mod p$
(1) Choose numbers n at random until $\left(\frac{n}{p}\right) = -1$
(2) Set $z = n^m \mod p$ and $b \equiv a^m \mod p$.
(4) Find the smallest integer $r \geq 0$ such that $b \equiv z^r \equiv n^{mr} \mod p$.
(5) Set $x \equiv a^{(m+1)/2} z^{-r/2} \equiv \sqrt{a} \mod p$.

Algoritmo 2 - Cipolla

Algorithm 2 : Cipolla's Algorithm

Input: an odd prime p and a quadratic residue a modulo p.

Output: $\sqrt{a} \mod p$

(1) Find an integer t with $0 \leq t \leq p-1$ such that $u = t^2 - a$ is a quadratic non-residue $\mod p$.

(2) Return $(t + \sqrt{u})^{(p+1)/2}$.

Algoritmo 3 - Peralta

Algorithm 3 : Peralta's Algorithm I

Input: a quadratic residue a modulo p

Output: $x \equiv \sqrt{a} \mod p$.

(1) Choose $r \in \mathbb{Z}_p^*$ at random such that $r^2 \not\equiv a \mod p$, otherwise output is r.

(2) Compute $(r + \sqrt{a})^{(p-1)/2} = u + v\sqrt{a}$.

(3) If $u = 0$, output $x \equiv v^{-1} \mod p$ else go to (1).

Algoritmo 3 - Peralta

Algorithm 4 : Peralta's Algorithm II

Input: a quadratic residue a modulo p.
Output: $x \equiv \sqrt{a} \mod p$.
(1) Choose r at random $\in \mathbb{Z}_p^*$.
(2) If $r^2 \equiv -a \mod p$, choose a new r.
(3) Compute $(r + \sqrt{-a})^m = u + v\sqrt{-a}$, where $p - 1 = 2^e m$ and m is an odd integer.
(4) If either u or v is 0, choose a new r.
(5) Compute $(u + v\sqrt{-a})^{2^i}$ for some $i = 1, 2, \cdots, e$ until $(u + v\sqrt{-a})^{2^i} = 0 + w\sqrt{-a}$ for some w.
(6) Let $(u + v\sqrt{-a})^{2^{i-1}} = k + l\sqrt{-a}$. Then $k^2 - l^2 a \equiv 0 \mod p$ and output k/l.

Conclusão

Table 1 The tests are conducted for primes p where $p - 1 = 2^e m$ and the time (in millisecond) is average of 1000 runs for each algorithm

Finite field size (size of p)	256-bit, e = 4	512-bit, e = 5	1024-bit, e = 8
Tonelli–Shanks	0.753	1.548	4.792
Tonelli–Shanks (Quadratic Reciprocity)	0.328	0.642	2.372
Cipolla	0.583	1.391	4.484
Peralta	0.407	0.720	2.188
Singular cubics	0.317	0.992	4.298