5.3 Square Roots in Finite Prime Fields

In many cases taking square roots is a trivial task as the following simple consideration shows:

Lemma 9 *Let G be a finite group of odd order m. Then for each $a \in G$ there is exactly one $x \in G$ with $x^2 = a$, and it is given by $x = a^{\frac{m+1}{2}}$.*

Proof. Since $a^m = 1$ we have $x^2 = a^{m+1} = a$. We conclude that the squaring map $x \mapsto x^2$ is surjective, hence a bijection $G \longrightarrow G$. $\diamond$

We search methods for taking square roots in a finite prime field $\mathbb{F}_p$ as efficiently as possible. The case $p \equiv 3 \pmod 4$ is extremely simple by the foregoing consideration: If $p = 4k+3$, then the group $\mathbb{M}_p^2$ of quadratic residues has odd order $\frac{p-1}{2} = 2k+1$. Hence for a quadratic residue $z \in \mathbb{M}_p^2$ the unique square root is $x = z^{k+1} \bmod p$ [Lagrange 1769]. The cost of taking this square root is at most $2 \cdot \log_2(p)$ congruence multiplications.

Examples

1. For $p = 7 = 4 \cdot 1 + 3$ we have $k+1 = 2$. By A.8 2 is a quadratic residue. A square root is $2^2 = 4$. Check: $4^2 = 16 \equiv 2$.

2. For $p = 23 = 4 \cdot 5 + 3$ we have $k+1 = 6$. By A.8 again 2 is a quadratic residue. A square root is $2^6 = 64 \equiv 18$. Check: $18^2 \equiv (-5)^2 = 25 \equiv 2$.

Unfortunately for $p \equiv 1 \pmod 4$ we cannot hope for such a simple procedure. For example -1 is a quadratic residue, but no power of -1 can be a square root of -1 since always $[(-1)^m]^2 = (-1)^{2m} = 1 \neq -1$.

Fortunately there are general procedures, for example one that is baptized AMM after Adleman, Manders, and Miller, but was described already by Cipolla in 1903. It starts by decomposing $p-1$ into $p-1 = 2^e \cdot u$ with odd u. Furthermore we choose (once and for all) an arbitrary quadratic nonresidue $b \in \mathbb{F}_p^\times - \mathbb{M}_p^2$—this is the only nondeterministic step in the algorithm, see Section A.8. (Assuming ERH the procedure is even deterministic, as it is in the many cases where a quadratic nonresidue is known anyway.)

Now we consider a quadratic residue $z \in \mathbb{M}_p^2$ and want to find a square root of it. Since $z \in \mathbb{M}_p^2$, we have $\operatorname{ord}(z) \mid \frac{p-1}{2}$, hence the 2-order $r = \nu_2(\operatorname{ord}(z))$ of $\operatorname{ord}(z)$ is bounded by $\leq e-1$, and r is minimal with $z^{u2^r} \equiv 1$.

We recursively define a sequence $z_1, z_2, \ldots$ beginning with

$$z_1 = z \quad \text{with } r_1 = \nu_2(\operatorname{ord}(z_1)).$$

If $z_i \in \mathbb{M}_p^2$ is chosen, and r_i is the 2-order of $\mathrm{ord}(z_i)$, then the sequence terminates if $r_i = 0$. Otherwise we set

$$z_{i+1} = z_i \cdot b^{2^{e-r_i}}.$$

Then $z_{i+1} \in \mathbb{M}_p^2$. Furthermore

$$z_{i+1}^{u\cdot 2^{r_i-1}} \equiv z_i^{u\cdot 2^{r_i-1}} \cdot b^{u\cdot 2^{e-1}} \equiv 1,$$

since the first factor is $\equiv -1$ due to the minimality of r_i, and the second factor is $\equiv (\frac{b}{p}) = -1$, for $u \cdot 2^{e-1} = \frac{p-1}{2}$. Hence $r_{i+1} < r_i$. The terminating condition $r_n = 0$ is reached after at most e steps with $n \leq e \leq \log_2(p)$.

Then we compute reversely:

$$x_n = z_n^{\frac{u+1}{2}} \bmod p$$

with $x_n^2 \equiv z_n^{u+1} \equiv z_n$ (since $\mathrm{ord}(z_n) \mid u$ by its odd parity). Recursively

$$x_i = x_{i+1}/b^{2^{e-r_i-1}} \bmod p$$

that by induction satisfies

$$x_i^2 \equiv x_{i+1}^2/b^{2^{e-r_i}} \equiv z_{i+1}/b^{2^{e-r_i}} \equiv z_i.$$

Hence $x = x_1$ is a square root of z.

In addition to the cost of finding b we count the following steps:

- Computing the powers $b^2, \ldots, b^{2^{e-1}}$, costing $(e-1)$ modular squares.
- Computing the powers $b^u, b^{2u}, \ldots, b^{2^{e-1}u}$, taking at most $2 \cdot \log_2(u) + e - 1$ congruence multiplications.
- Computing z^u, taking at most $2 \cdot \log_2(u)$ congruence multiplications.
- Furthermore we compute for each $i = 1, \ldots, n \leq e$:
  - z_i by one congruence multiplication,
  - z_i^u from z_{i-1}^u by one congruence multiplication,
  - $z_i^{u2^r}$ from $z_{i-1}^{u2^r}$ by one congruence multiplication,
  - and then r_i.

  This makes a total of at most $3 \cdot (e-1)$ congruence multiplications.
- x_n as a power by at most $2 \cdot \log_2(u)$ congruence multiplications.
- x_i from x_{i+1} each by one congruence division with cost $\mathrm{O}(\log(p)^2)$.

Summing up we get costs of size about $\mathrm{O}(\log(p)^3)$ with a rather small constant coefficient.

Example Let $p = 29$ and $z = 5$. Then $p-1 = 4\cdot 7$, hence $e = 2$ and $u = 7$. By the remarks above $b = 2$ is a quadratic nonresidue. We compute the powers

$$b^2 = 4, \quad b^u \equiv 128 \equiv 12, \quad b^{2u} \equiv 144 \equiv -1,$$

$$z^2 \equiv 25 \equiv -4, \quad z^4 \equiv 16, \quad z^6 \equiv -64 \equiv -6, \quad z^7 \equiv -30 \equiv -1.$$

Now

$$z_1 = 5, \quad z_1^u \equiv -1, \quad z_1^{2u} \equiv 1, \quad r_1 = 1,$$

$$z_2 \equiv z_1 b^2 \equiv 5 \cdot 4 = 20, \quad z_2^u \equiv z_1^u b^{2u} \equiv (-1)(-1) = 1, \quad r_2 = 0.$$

Now we go backwards:

$$x_2 \equiv z_2^{\frac{u+1}{2}} = z_2^4 = (z_2^2)^2 \equiv 400^2 \equiv (-6)^2 = 36 \equiv 7,$$

$$x_1 = x_2/b \bmod p = 7/2 \bmod 29 = 18.$$

Hence $x = 18$ is the wanted root. Check: $18^2 = 324 \equiv 34 \equiv 5$.

Exercises Find deterministic algorithms (= simple formulas) for taking square roots in the fields

- $\mathbb{F}_p$ with $p \equiv 5 \pmod 8$
- $\mathbb{F}_{2^m}$ with $m \geq 2$ [Hints: 1. Consider the order of the radicand in the multiplicative group. 2. Invert the linear map $x \mapsto x^2$.]
- $\mathbb{F}_q$ for $q = p^m$

Alternative algorithms: Almost all known efficient algorithms that completely cover the case $p \equiv 1 \pmod 4$ are probabilistic and have a deterministic variant whose cost is polynomial assuming ERH. The book by Forster (*Algorithmische Zahlentheorie*) has a variant of the Cipolla/AMM algorithm that uses the quadratic extension $\mathbb{F}_{p^2} \supseteq \mathbb{F}_p$ and is conceptionally quite simple. The *Handbook of Applied Cryptography* (Menezes/van Oorschot/Vanstone) contains an algorithm by Tonelli 1891 that admits a concise formulation, but cost $\mathrm{O}(\log(p)^4)$. Another method is a special case of the Cantor/Zassenhaus algorithm for factoring polynomials over finite fields, see von zur Gathen/Gerhard: *Modern Computer Algebra.* Yet another procedure by Lehmer uses the Lucas sequence (a_n) with $a_1 = b$, $a_2 = b^2 - 2z$, where $b^2 - 4z$ is a quadratic nonresidue. The only known deterministic algorithm with proven polynomial cost was given by Schoof. It uses elliptic curves, and costs $\mathrm{O}(\log(p)^9)$, so it is of theoretical interest only.

For overviews see:

- E. Bach/ J. Shallit: *Algorithmic Number Theory*. MIT Press, Cambridge Mass. 1996.
- D. J. Bernstein: Faster square roots in annoying finite fields. Preprint (siehe die Homepage des Autors http://cr.yp.to/).