Pratt Certification and the Primality of $2^{255} - 19$

David M. Russinoff

December 21, 2015

The theoretical import of Pratt's method of prime certification [3] is that for every prime p there is a procedure for establishing its primality with execution time that is polynomial in $\log p$. As a practical consequence, this enables straightforward certification of primes that are too large to be checked by exhaustive computation. In particular, we shall construct a Pratt certificate for $2^{255} - 19$, the basis of the Diffie-Hellman function known as Curve25519 [2], that we have used in the formal verification of its primality. All results presented below are formalized in the mechanically checked proof script `books/projects/-quadratic-reciprocity/pratt` of the ACL2 repository.

Definition 1 *Let $n \in \mathbb{Z}$, $n > 1$,and $r \in \mathbb{Z}$. The order of r modulo n, if it exists, is the least positive integer m such that $r^m \bmod n = 1$.*

Lemma 1 *Let $p \in \mathbb{Z}$, $p > 1$. and let $r \in \mathbb{Z}$ with order m modulo p. Then for all $k \in \mathbb{Z}$, $r^k \bmod n = 1$ if and only iff $m|k$.*

PROOF: If $k = ma$, then $r^k = (r^m)^a \equiv 1^a = 1 \pmod{n}$. On the other hand, suppose $k = ma + b$, where $0 < b < m$. Then

$$r^k = (r^m)^a r^b \equiv r^b \pmod{n},$$

and by minimality of m, $r^b \bmod n \neq 1$. □

It may be shown that for every prime p there exists an integer of order $p-1$ mod p, called a *primitive root* of p. Here we are interested in the converse of this statement:

Lemma 2 *Let $p \in \mathbb{Z}$, $p > 1$. and $r \in \mathbb{Z}$. If the order of r mod p is $p-1$, then p is prime.*

PROOF: First note that if $1 \leq j \leq i < p$ and $r^j \equiv r^i \pmod{p}$, then

$$r^{i-j} \equiv r^{i-j}\left(r^{p-1}\right)^j = r^i r^{(p-2)j} \equiv r^j r^{(p-2)j} = \left(r^{p-1}\right)^j \equiv 1 \pmod{p},$$

and since $0 \leq i - j < p - 1$, where $p-1$ is the order of r mod p, we must have $i = j$. It follows that

$$\{r^i \bmod p | 1 \leq i < p\} = \{1, 2, \ldots p-1\}.$$

Now suppose $1 \leq q < p$ and $q|p$. Then $q|(q^{p-1} \bmod p)$. But for some i, $q = r^i \bmod p$, and hence $q^{p-1} \bmod p \equiv \left(r^i\right)^{p-1} = r^{(p-1)i} \equiv 1 \bmod p$. It follows that $q|1$, and hence $q = 1$. □

Thus, to establish primality of p, it suffices to exhibit a primitive root r. This requires computing $r^m \bmod p$ for large values of m, which can be done efficiently by binary exponentiation:

Definition 2 *If $b \in \mathbb{Z}$, $e \in \mathbb{Z}$, $n \in \mathbb{Z}$, and $e \geq 0$, then*

$$BE(b, e, n) = BE'(b, e, n, 1),$$

where BE' is defined recursively by

$$BE'(b, e, n, r) = \begin{cases} BE'(b^2 \bmod n, \frac{e}{2}, n, r) & \textit{if } e \textit{ is even} \\ BE'(b^2 \bmod n, \frac{e-1}{2}, n, rb \bmod n) & \textit{if } e \textit{ is odd.} \end{cases}$$

Lemma 3 *If $b \in \mathbb{Z}$, $e \in \mathbb{Z}$, $n \in \mathbb{Z}$, and $e \geq 0$, then $BE(b, e, n) = b^e \bmod n$.*

Proof: By induction, if e is even, then

$$BE'(b, e, n, r) = \left((b^2 \bmod n)^{\frac{e}{2}} r\right) \bmod n = b^e r \bmod n,$$

and if e is odd, then

$$BE'(b, e, n, r) = \left((b^2 \bmod n)^{\frac{e-1}{2}} (rb \bmod n)\right) \bmod n = b^e r \bmod n.$$

Thus, $BE(b, e, n) = BE'(b, e, n, 1) = b^e \bmod n$. □

The following result limits the number of exponentiations required to establish a primitive root.

Lemma 4 *Let $p \in \mathbb{Z}$, $p > 1$, and $r \in \mathbb{Z}$. If $r^{p-1} \bmod p = 1$ and for every prime factor q of $p - 1$, $r^{\frac{p-1}{q}} \bmod p \neq 1$, then p is prime.*

Proof: Let m be the order of r mod p. By Lemma 1, $m|p - 1$, and by Lemma 2, we need only show that $m = p - 1$. But if not, then $\frac{p-1}{m}$ has a prime factor q, which must also be a factor of $p - 1$, But this implies $m|\frac{p-1}{q}$, and therefore $r^{\frac{p-1}{q}} \bmod p = 1$. □

Thus, given the prime factorization of $p - 1$ and a primitive root of p, if k is the number of distinct prime in the factorization, then p may be certified as a prime by computing $k + 1$ exponentials and certifying the primality of each prime factor recursively. This suggests a prime certificate structured as a tree.

Definition 3 *Let $p \in \mathbb{Z}$, $p > 1$. A prime certificate for p is a list*

$$(r\ \ (q_1 \ldots q_k)\ \ (e_1 \ldots e_k)\ \ (c_1 \ldots c_k)),$$

where

(1) r is a primitive root of p;

(2) $q_1, \ldots, q_k$ are distinct primes and $e_1, \ldots, e_k$ are positive integers such that $p = \prod_{i=1}^{k} q_i^{e_i}$;

(3) For $1 \leq i \leq k$, c_i is either `NIL` *or a prime certificate for q_i.*

The intention is that the leaves of the tree, i.e., the primes for which no certificate is supplied, are small enough to be certified by direct computation. Thus, according to Lemma 4, the primality of p may be established by verifying a prime certificate for p as follows:

(1) Check that $BE(r, p-1, p) = 1$ and for $i = 1, \ldots, k$, $BE(r, \frac{p-1}{q_i}, p) \neq 1$.

(2) Check that $p = \prod_{i=1}^{k} q_i^{e_i}$;.

(3) For $i = 1, \ldots, k$, if $c_i =$ `NIL`, then verify the primality of q_i by exhaustively checking for divisors; otherwise recursively verify that c_i is a certificate for q_i.

To establish the complexity bound mentioned at the beginning of this note, we argue informally as follows. First, note that the number k of prime factors of $p-1$ is bounded by $\log_2 p$. Thus, the number of exponentiations (which dominate the computation) required at the root node is $O(\log p)$. Each of these involves $O(\log p)$ multiplications, and each multiplication has execution time $O(\log^2 p)$ (although a better estimate is possible). Thus, the computation at each internal node is $O(\log^4 p)$.

It is clear that any prime p admits a certificate in which the prime at each leaf node is 2. We shall show by induction that the number of internal nodes of this tree is bounded by $4\log_2 p - 4$. In the base case, $p = 2$, this holds trivially: there are no internal nodes and $4\log_2 p - 4 = 0$. Suppose $p > 2$. By inductive hypothesis, the number of internal nodes is at most

$$1 + \sum_{i=1}^{k} (4log_2 q_i - 4) = 1 - 4k + 4\log_2 \prod_{i=1}^{k} q_i.$$

If $k \geq 2$, then the claim follows from $\prod_{i=1}^{k} q_i \leq p$. But if $k = 1$, then $p \geq q_1^2$, the above bound reduces to

$$-3 + 4\log_2 q_1 \leq -3 + 2\log_2 p < -4 + 4\log_2 p.$$

Consequently, the total certification time is $O(\log^5 p)$.

Of course, the existence of a prime certificate for p does not guarantee that it can be easily found. This requires producing a prime factorization of $p-1$ and identifying a primitive root of p. Note, however, that our methods for generating these values need not be trusted, since their correctness will be verified through the certification process. In order for the method to be effective, it is sufficient that the values used happen to be correct.

Our particular interest in this method is motivated by our investigation of the Diffie-Hellman key agreement algorithm Curve25519 [4], which requires establishing the primality of $\wp = 2^{255} - 19$. For this purpose, we use the following certificate:

```
(2
 (2 3 65147 74058212732561358302231226437062788676166966415465897661863160754340907)
 (2 1 1 1)
 (() () ()
  (2
   (2 3 353 57467 132049 1923133 31757755568855353 75445702479781427272750846543864801)
   (1 1 1 1 1 1 1 1)
   (() () () () () ()
    (10
     (2 3 31 107 223 4153 430751)
     (3 1 1 1 1 1 1)
     (() () () () () () ()))
    (7
     (2 3 5 75707 72106336199 1919519569386763)
     (5 2 2 1 1 1)
     (() () () () ()
      (2
       (2 3 7 19 47 127 8574133)
       (1 1 1 1 2 1 1)
       (() () () () () () ())))))))))
```

According to this certificate, 2 is a primitive root of $\wp$ and $\wp - 1$ has four distinct prime factors, with corresponding exponents 2, 1, 1, and 1. The first three of these factors are small enough to be verified by exhaustive computation. The fourth factor requires its own certificate, which involves eight prime facors, two of which are large enough to require certificates.

The prime factorizations that appear in this certificate may be computed by any of the many factorization programs that are accessible on the Web. Each of the primitive roots was found by a linear search, checking the integers $2, 3, 4, \ldots$ to find the least value that satisfies the requirements of Lemma 4. There is no guarantee that this process terminates quickly (little is known about the relative size of the smallest primitive root of a prime), but experience suggests that it generally does.

This certificate has been verified through the above process by ACL2, with a reported execution time less than 0.01 second. Thus, we have a formal proof of the desired result:

Lemma 5 *$\wp = 2^{255} - 19$ is prime.*

The Curve25519 algorithm is based on a group operation on the elliptic curve defined by the equation $y^2 = x^3 + Ax^2 + x$, where $A = 486662$, over the Galois field $\mathbb{F}_\wp$ of order $\wp$. In addition to the primality of $\wp$, our analysis requires identifying quadratic residues modulo $\wp$, i.e., integers $m \in \mathbb{F}_\wp$ for which there exists $x \in \mathbb{Z}$ satisfying $x^2 \bmod \wp = m$. According to a well known result of Euler, if m is an integer not divisible by a prime p, then

$$m^{\frac{p-1}{2}} \bmod p = \begin{cases} 1 & \text{if } m \text{ is a quadratic residue mod } p \\ -1 & \text{otherwise.} \end{cases}$$

A formalization of Euler's criterion may be found in the script `books/quadratic-reciprocity/euler` in the ACL2 repository. Combining this result with Lemma 3, we may efficiently compute the quadratic character of an integer modulo $\wp$ by binary exponentiation. The following two results have been verified by direct computation. According to the first of these, the value 9 occurs as the x-coordinate of a point on the curve:

Proposition 1 *$9^3 + A \cdot 9^2 + 9$ is a quadratic residue modulo $\wp$.*

The second implies that the origin is the only point of intersection of the curve with the x-axis:

Proposition 2 *$A^2 - 4$ is not a quadratic residue modulo $\wp$.*

Corollary 1 *If $(x, 0) \in EC$, then $x = 0$.*

PROOF: (The following operations are in the field $\mathbb{F}_\wp$.) Suppose $x^3 + Ax^2 + x = 0$ and $x \neq 0$. Then $x^2 + Ax + 1 = 0$, and hence

$$\begin{aligned}
A &= -\frac{x^2+1}{x}, \\
A - 2 &= -\frac{x^2+2x+1}{x} = -\frac{(x+1)^2}{x}, \\
A + 2 &= -\frac{x^2-2x+1}{x} = -\frac{(x-1)^2}{x}, \\
A^2 - 4 &= \frac{(x+1)^2(x-1)^2}{x^2} = \left(\frac{x^2-1}{x}\right)^2,
\end{aligned}$$

contradicting Proposition 2. □

References

[1] ACL2 home page, `www.cs.utexas.edu/users/moore/acl2/`.

[2] Bernstein, Daniel J.: Curve25519: New Diffie-Hellman Speed Records. In: 9th International Conference on Theory and Practice of Public Key Cryptography. Springer (2006)

[3] Pratt, Vaughn: Every Prime Has a Succinct Certificate. In: SIAM Journal on Computing, vol. 4 (1975)

[4] Russinoff, David M.: A Computationally Surveyable Proof of the Curve25519 Group Axioms. Unpublished manuscript, `www.russinoff.com/papers/group.pdf`.