

ISO/IEC 18032:2020 (E)

Information security — Prime number generation

Contents

| | |
|---------|---|
| | Foreword |
| 1 | Scope |
| 2 | Normative references |
| 3 | Terms and definitions |
| 4 | Symbols and abbreviated terms |
| 5 | Trial division |
| 6 | Probabilistic primality test |
| 6.1 | General |
| 6.2 | Requirements |
| 6.3 | Miller-Rabin primality test |
| 7 | Deterministic primality verification methods |
| 7.1 | General |
| 7.2 | Elliptic curve primality proving algorithm |
| 7.2.1 | General |
| 7.2.2 | Elliptic curve primality certificate generation |
| 7.2.3 | Elliptic curve primality certificate verification |
| 7.3 | Primality certificate based on The Shawe-Taylor algorithm |
| 8 | Prime number generation |
| 8.1 | General |
| 8.2 | Requirements |
| 8.3 | Using the Miller-Rabin primality test |
| 8.3.1 | General |
| 8.3.2 | Random search |
| 8.3.3 | Incremental search |
| 8.3.4 | Primes with an elliptic curve primality certificate |
| 8.4 | Using deterministic methods |
| 8.4.1 | General |
| 8.4.2 | The Shawe-Taylor algorithm |
| Annex A | (normative) Error probabilities |
| A.1 | General |
| A.2 | Worst-case error estimate for t Miller-Rabin primality tests |
| A.3 | Average-case error estimates for t Miller-Rabin primality tests |
| Annex B | (normative) Generating primes with side conditions |
| B.1 | General |
| B.2 | Congruence restrictions on primes |
| B.2.1 | General |
| B.2.2 | Congruence restrictions and incremental or random search |
| B.2.3 | Congruence restrictions and The Shawe-Taylor algorithm |
| B.2.4 | Generating primes in an interval |
| Annex C | (normative) Additional random number generation methods |
| C.1 | General |
| C.2 | Simple conversion method |
| C.3 | Simple discard method (reverse order) |

C.4 Simple modular method (reverse order)

Annex D (normative) Auxiliary methods

- D.1 Sieving procedure**
- D.2 Primality tests based on Pocklington's theorem**
 - D.2.1 General**
 - D.2.2 Pocklington's primality test**
 - D.2.3 Partial Pocklington's primality test**
- D.3 Probabilistic Lucas primality test**
- D.4 Lucas' deterministic primality tests**
 - D.4.1 General**
 - D.4.2 Deterministic Lucas primality test**
 - D.4.3 The partial deterministic Lucas primality test**
- D.5 Brillhart-Lehmer-Selfridge primality test**
- 8.5 If test returns false, i.e. there does not exist a proper divisor d of N with $d \bmod \text{lcm}(F_1, F_2) = 1$, accept the certificate, return "N prime" and stop.**
- D.6 Elliptic curve primality test**
- D.7 Cornacchia's algorithm**
- D.8 Complex multiplication (CM) method**
- D.9 Testing for perfect squares**
 - D.9.1 General**
 - D.9.2 Probabilistic method**
 - D.9.3 Deterministic method**
- D.10 Jacobi symbol**
- D.11 Lucas symbol**
- D.12 Finding divisors in residue classes (Lenstra).**

Annex E (informative) Prime generation examples

- E.1 General**
- E.2 Miller-Rabin incremental search example**
- E.3 Miller-Rabin random search example**

Page count: 33