Birthday paradox, finite fields, Fermat's Little Theorem, fast modular exponentiation

Or, all the math you need for exam 1 other than FFTs

CSE 548 Spring 2026
jedimaestro@asu.edu

Birthday paradox...

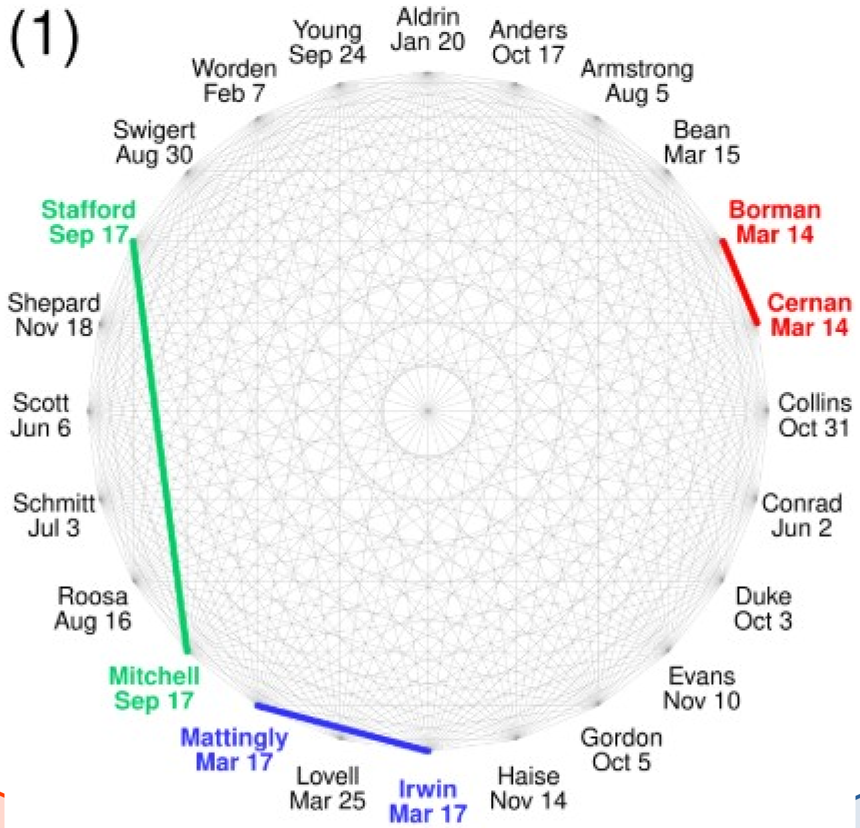
Birthday Attacks on DNS

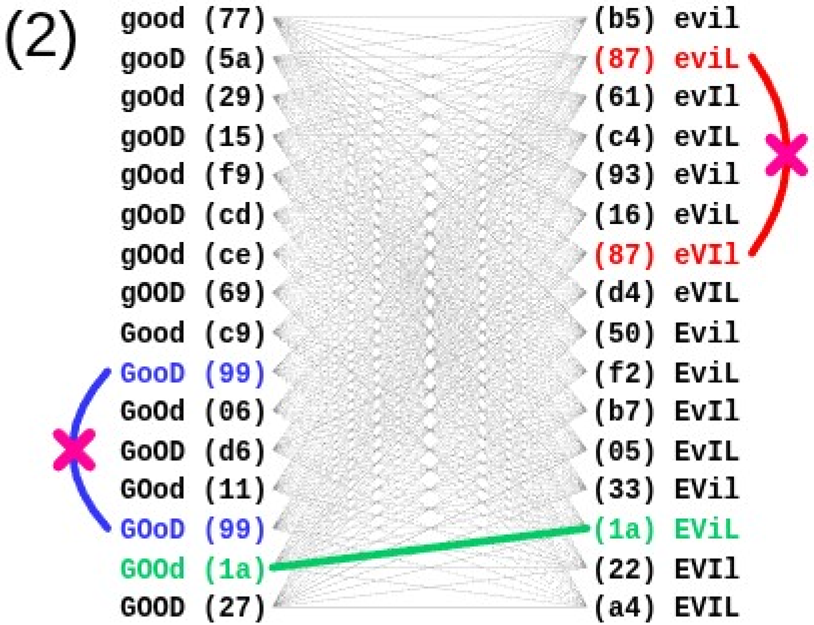
- https://www.kb.cert.org/vuls/id/457875
- 2002

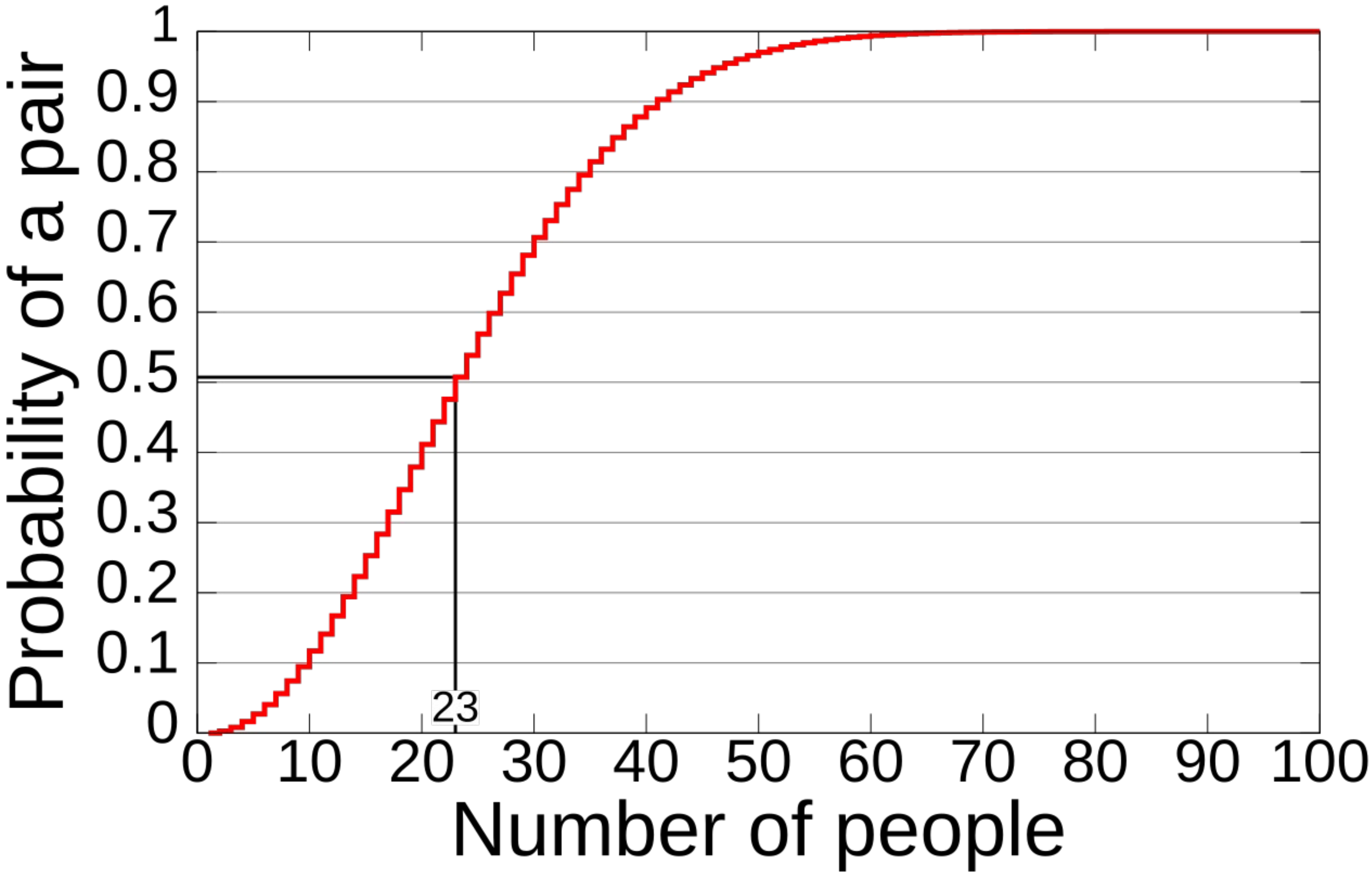
If the attacker has to guess...	...and is limited to the following number of open requests...	...it will take the following number of packets to achieve a 50% success rate (includes both requests and responses)
TID only (16bits)	1	32.7 k (2^{15})
TID only (16bits)	4	10.4 k
TID only (16bits)	200	427
TID only (16bits)	unlimited	426
TID and port (32 bits)	1	2.1 billion (2^{31})
TID and port (32 bits)	4	683 million
TID and port (32 bits)	200	15 million
TID and port (32 bits)	unlimited	109 k

Table 1: Number of packets required to reach 50% success probability for various numbers of open queries

https://en.wikipedia.org/wiki/Birthday_attack






Probability of a pair
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
23
0 10 20 30 40 50 60 70 80 90 100
Number of people

This process can be generalized to a group of n people, where $p(n)$ is the probability of at least two of the n people sharing a birthday. It is easier to first calculate the probability $\bar{p}(n)$ that all n birthdays are *different*. According to the pigeonhole principle, $\bar{p}(n)$ is zero when $n > 365$. When $n \leq 365$:

$$\bar{p}(n) = 1 \times \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \times \cdots \times \left(1 - \frac{n-1}{365}\right)$$

The Taylor series expansion of the exponential function (the constant $e \approx 2.718\,281\,828$)

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots$$

provides a first-order approximation for e^x for $|x| \ll 1$:

$$e^x \approx 1 + x.$$

To apply this approximation to the first expression derived for $\bar{p}(n)$, set $x = -\frac{a}{365}$. Thus,

$$e^{-a/365} \approx 1 - \frac{a}{365}.$$

Then, replace a with non-negative integers for each term in the formula of $\bar{p}(n)$ until $a = n - 1$, for example, when $a = 1$,

$$e^{-1/365} \approx 1 - \frac{1}{365}.$$

The first expression derived for $\bar{p}(n)$ can be approximated as

$$\bar{p}(n) \approx 1 \cdot e^{-1/365} \cdot e^{-2/365} \cdots e^{-(n-1)/365}$$

$$= e^{-\left(1+2+\cdots+(n-1)\right)/365}$$

$$= e^{-\frac{n(n-1)/2}{365}} = e^{-\frac{n(n-1)}{730}}.$$

Therefore,

$$p(n) = 1 - \bar{p}(n) \approx 1 - e^{-\frac{n(n-1)}{730}}.$$

$$p(n,d) \approx 1 - e^{-\frac{n(n-1)}{2d}}$$

An even coarser approximation is given by

$$p(n) \approx 1 - e^{-\frac{n^2}{730}},$$

A good rule of thumb which can be used for mental calculation is the relation

$$p(n,d) \approx \frac{n^2}{2d}$$

which can also be written as

$$n \approx \sqrt{2d \times p(n)}$$

which works well for probabilities less than or equal to $\frac{1}{2}$. In these equations, d is the number of days in a year.

For instance, to estimate the number of people required for a $\frac{1}{2}$ chance of a shared birthday, we get

$$n \approx \sqrt{2 \times 365 \times \tfrac{1}{2}} = \sqrt{365} \approx 19$$

Which is not too far from the correct answer of 23.

Finite fields...

https://en.wikipedia.org/wiki/%C3%89variste_Galois

https://en.wikipedia.org/wiki/Quadratic_equation

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

https://en.wikipedia.org/wiki/Cubic_equation

$$\frac{a}{q}x^2 + \frac{bq + ap}{q^2}x + \frac{cq^2 + bpq + ap^2}{q^3}$$

What is a field?

- “In mathematics, a field is a set on which addition, subtraction, multiplication, and division are defined and behave as the corresponding operations on rational and real numbers do.”

  --Wikipedia

- In cryptography, we often want to “undo things” or get the same result two different ways
  - Zmap will also use this trick
- On digital computers the math you learned in grade school is not good enough
  - Suppose we want to multiply by a plaintext, and the plaintext is 3. Great!
  - Now the decryption needs the inverse operation. Crap!
  - 1/3 is not easy to deal with (not even in floating point or fixed point)

Field

- Commutative

  $a + b = b + a$

  $a * b = b * a$

- Associative

  $(a + b) + c = a + (b + c)$

  $(a * b) * c = a * (b * c)$

- Identity

  $0 != 1, a + 0 = a, a * 1 = a$

- Inverse

  $a + -a = 0$

  $a * a^{-1} = 1$

- Distributive

  $a * (b + c) = (a * b) + (a * c)$

Arithmetic modulo a prime is a finite field

6 + 4 = 3 (mod 7)

3 – 6 = 4 (mod 7)

5 * 2 = 3 (mod 7)

5 * 3 = 1 (mod 7)

$3 * 5^{-1} = 3 * 3 = 2 \pmod 7$

This is called GF(7)

GF(2)

$0 + 0 = 0 \pmod 2$
$0 + 1 = 1 \pmod 2$
$1 + 0 = 1 \pmod 2$
$1 + 1 = 0 \pmod 2$

How to subtract?
Where have you seen this before?

GF(2)

$0 * 0 = 0 \pmod 2$
$0 * 1 = 0 \pmod 2$
$1 * 0 = 0 \pmod 2$
$1 * 1 = 1 \pmod 2$

Where have you seen this before?

GF(2)

- $K + K = 0$
- $(P + K) + K = P$
- $(A + K) + (B + K) = A + B$
- $0 + K = K$

XOR

- $K \oplus K = 0$
- $(P \oplus K) \oplus K = P$
- $(A \oplus K) \oplus (B \oplus K) = A \oplus B$
- $0 \oplus K = K$

How to use GF(2) to achieve what we want?

- Want to define a field over 2^k possibilities for a k-bit number
- 2 is prime, all other powers of 2 are not
  - Need to use irreducible polynomials

https://jedcrandall.github.io/courses/cse548spring2024/miniaesspec.pdf

Published in Cryptologia, XXVI (4), 2002.

Mini Advanced Encryption Standard (Mini-AES): A Testbed for Cryptanalysis Students

Raphael Chung-Wei **Phan**

2.1 The Finite Field $GF(2^4)$

The nibbles of Mini-AES can be thought of as elements in the finite field $GF(2^4)$. Finite fields have the special property that operations (+,–, × and ÷) on the field elements always cause the result to be also in the field. Consider a nibble $n = (n_3, n_2, n_1, n_0)$ where $n_i \in \{0,1\}$. Then, this nibble can be represented as a polynomial with binary coefficients i.e having values in the set {0,1}:

$$n = n_3 x^3 + n_2 x^2 + n_1 x + n_0$$

Example 1

Given a nibble, n = 1011, then this can be represented as

$$n = 1 x^3 + 0 x^2 + 1 x + 1 = x^3 + x + 1$$

Note that when an element of $GF(2^4)$ is represented in polynomial form, the resulting polynomial would have a degree of at most 3.

2.2 Addition in GF(2^4)

When we represent elements of GF(2^4) as polynomials with coefficients in {0,1}, then addition of two such elements is simply addition of the coefficients of the two polynomials. Since the coefficients have values in {0,1}, then the addition of the coefficients is just modulo 2 addition or exclusive-OR denoted by the symbol ⊕. Hence, for the rest of this paper, the symbols + and ⊕ are used interchangeably to denote addition of two elements in GF(2^4).

Example 2

Given two nibbles, n = 1011 and m = 0111, then the sum, n + m = 1011 + 0111 = 1100 or in polynomial notation:

$$n + m = (x^3 + x + 1) + (x^2 + x + 1) = x^3 + x^2$$

2.3 Multiplication in $GF(2^4)$

Multiplication of two elements of $GF(2^4)$ can be done by simply multiplying the two polynomials. However, the product would be a polynomial with a degree possibly higher than 3.

Example 3

Given two nibbles, n = 1011 and m = 0111, then the product is:

$$(x^3 + x + 1)(x^2 + x + 1) = x^5 + x^4 + x^3 + x^3 + x^2 + x + x^2 + x + 1$$
$$= x^5 + x^4 + 1$$

In order to ensure that the result of the multiplication is still within the field $GF(2^4)$, it must be reduced by division with an irreducible polynomial of degree 4, the remainder of which will be taken as the final result. An irreducible polynomial is analogous to a prime number in arithmetic, and as such a polynomial is irreducible if it has no divisors other than 1 and itself. There are many such irreducible polynomials, but for Mini-AES, it is chosen to be:

$$m(x) = x^4 + x + 1$$

Example 4

Given two nibbles, n = 1011 and m = 0111, then the final result after multiplication in $GF(2^4)$, called the 'product of n × m modulo m(x)' and denoted as ⊗, is:

$$
\begin{aligned}
(x^3 + x + 1) \otimes (x^2 + x + 1) &= x^5 + x^4 + 1 \text{ modulo } x^4 + x + 1 \\
&= x^2
\end{aligned}
$$

This is because:

$$
\begin{array}{rl}
 & \quad x + 1 \qquad \text{(quotient)} \\
x^4 + x + 1 \,\big)\! & x^5 + x^4 + 1 \\
 & + \; x^5 + x^2 + x \\
\hline
 & \qquad x^4 + x^2 + x + 1 \\
 & + \qquad x^4 + \qquad x + 1 \\
\hline
 & \qquad\qquad x^2 \qquad \text{(remainder)}
\end{array}
$$

Note that since the coefficients of the polynomials are in {0,1}, then addition is simply exclusive-OR and hence subtraction is also exclusive-OR since exclusive-OR is its own inverse.

Example 4

Given two nibbles, n = 1011 and m = 0111, then the final result after multiplication in $GF(2^4)$, called the 'product of n × m modulo m(x)' and denoted as ⊗, is:

$$(x^3 + x + 1) \otimes (x^2 + x + 1) \quad = x^5 + x^4 + 1 \text{ modulo } x^4 + x + 1$$
$$= x^2$$

This is because:

This is how to show your work on Exam 1

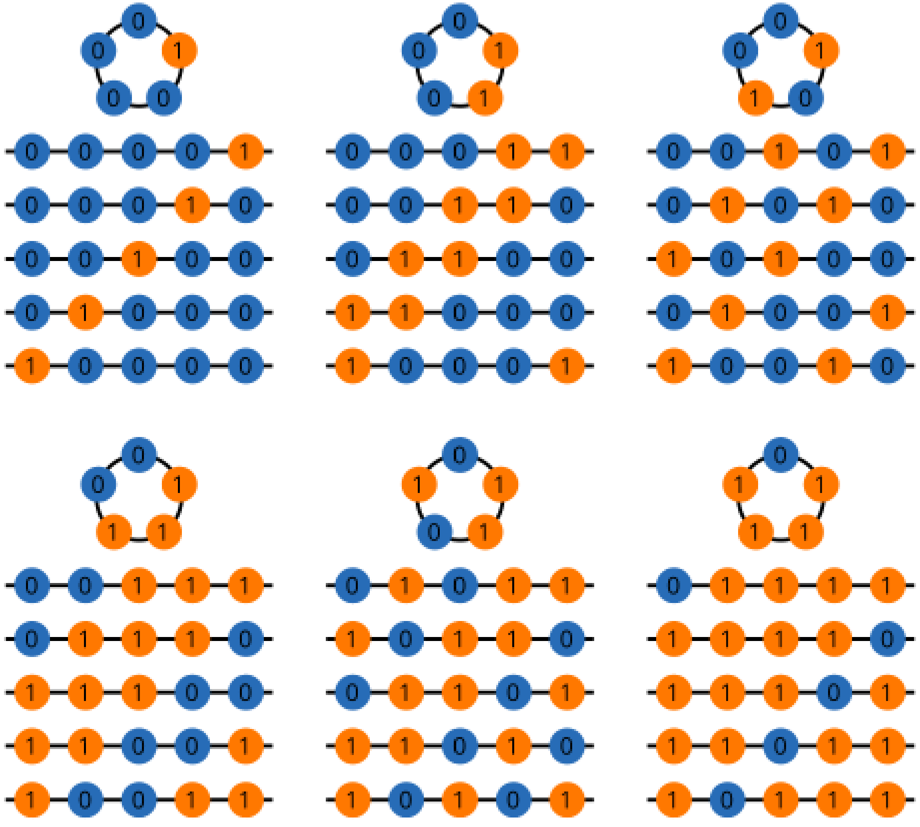
$$\begin{array}{rl}
 & x + 1 \qquad \text{(quotient)} \\
x^4 + x + 1 \,\big)\!\! & x^5 + x^4 + 1 \\
 & + \; x^5 + x^2 + x \\ \hline
 & x^4 + x^2 + x + 1 \\
 & + \; x^4 + \quad x + 1 \\ \hline
 & x^2 \qquad \text{(remainder)}
\end{array}$$

Note that since the coefficients of the polynomials are in {0,1}, then addition is simply exclusive-OR and hence subtraction is also exclusive-OR since exclusive-OR is its own inverse.

Fermat’s Little Theorem...

$$a^p \bmod p = a \pmod{p}$$
$$a^{p-1} \bmod p = 1 \pmod{p}$$
$$a^{p-2} \bmod p = a^{-1} \pmod{p}$$

https://mathlesstraveled.wordpress.com/2017/12/12/fermats-little-theorem-proof-by-necklaces/



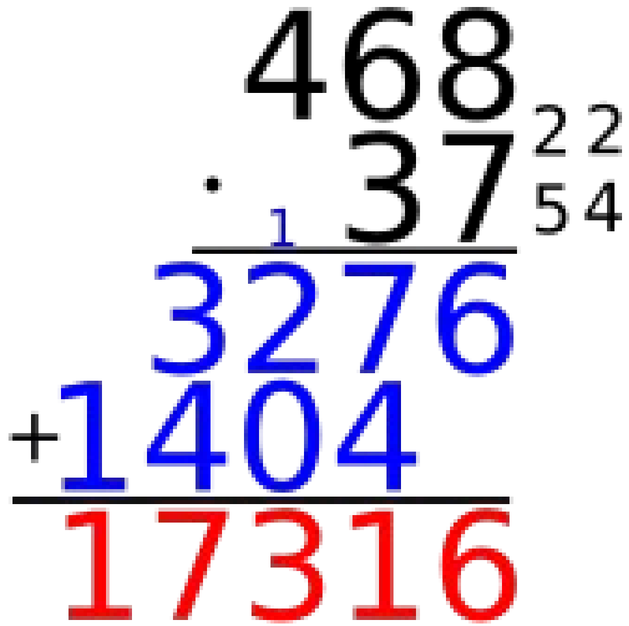
We already know there are $a^p - a$ strands with at least two colors; since we can put them in groups of p, one for each necklace of at least two colors, $a^p - a$ must be evenly divisible by p. QED!

Finite fields *mod p*

- Multiplicative inverse is just e^{p-2}
- **So why study the Extended Euclidean algorithm (later, for Exam 2)?** Because we can't do signatures with Diffie-Hellman, since Fermat's little theorem is an easy way to find multiplicative inverses.
- Preview: same is true of any finite field, so RSA uses ring theory:
  - $n = pq$ where p and q are prime, is a composite number
  - $\varphi(n) = (p - 1)(q - 1)$ is Euler's totient function, which counts the numbers less than n that are co-prime to n

Fast modular exponentiation *via* repeated squaring...

Multiplication is polynomial time in number of digits (O(n^2) or O(n log n))



Modular exponentiation

153^{189} (mod 251)

Naive way: multiply 153 times itself 189 times.
Won't work for, *e.g.*, 2048-bit numbers in the exponent

Better way (all mod 251)

$153^0 = 1$

$153^1 = 153$

$153^2 = 66$

$153^4 = 89$

$153^8 = 140$

$153^{16} = 22$

$153^{32} = 233$

$153^{64} = 73$

$153^{128} = 58$

1. Repeated squaring

2. Don’t forget the modulus

Better way

- 189 in binary is 0b10111101
- $189 = 1*2^7 + 0*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0$
- $153^{189} \pmod{251} = 153^{(128+0+32+16+8+4+0+1)} \pmod{251}$

$= 153^{128} * 153^{32} * 153^{16} * 153^{8} * 153^{4} * 153^{1} \pmod{251}$

$= 58 * 233 * 22 * 140 * 89 * 153 \pmod{251}$

$= 73$

WolframAlpha computational intelligence.

58 * 233 * 22 * 140 * 89 * 153 (mod 251)

NATURAL LANGUAGE   MATH INPUT

EXTENDED KEYBOARD   EXAMPLES   UPLOAD   RANDOM

Input

$(58 \times 233 \times 22 \times 140 \times 89 \times 153) \bmod 251$

Result

73

WolframAlpha computational intelligence.

(153^189) mod 251

NATURAL LANGUAGE | MATH INPUT | EXTENDED KEYBOARD | EXAMPLES | UPLOAD | RANDOM

Input

$153^{189} \bmod 251$

Result

73

$$153^{189} = 73 \pmod{251}$$

$$189 = \log_{153} 73 \pmod{251}$$

$$153^{???} = 73 \pmod{251}$$
$$??? = \log_{153} 73 \pmod{251}$$

This is called the discrete logarithm, and there is no known algorithm for solving it in the general case that is polynomial in the number of digits.

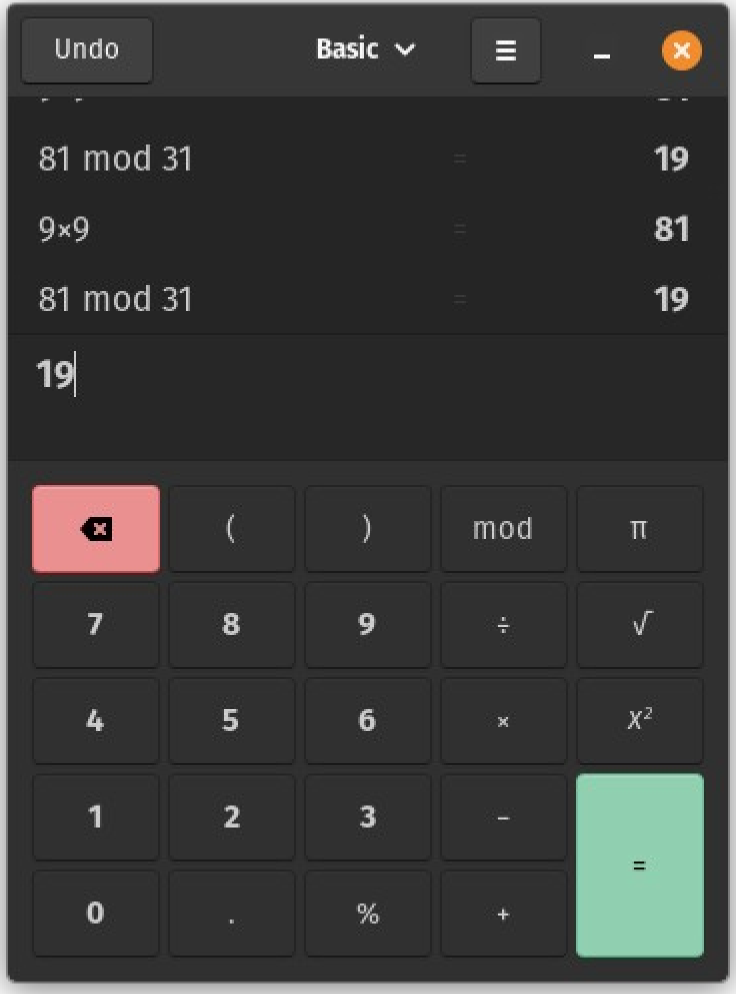
$$153^{189} = 73 \pmod{251}$$
$$153^{64} = 73 \pmod{251}$$

$$153^{189} \equiv 73 \pmod{251}$$
$$153^{64} \equiv 73 \pmod{251}$$

$$153^{189} \equiv 153^{64} \equiv 73 \pmod{251}$$

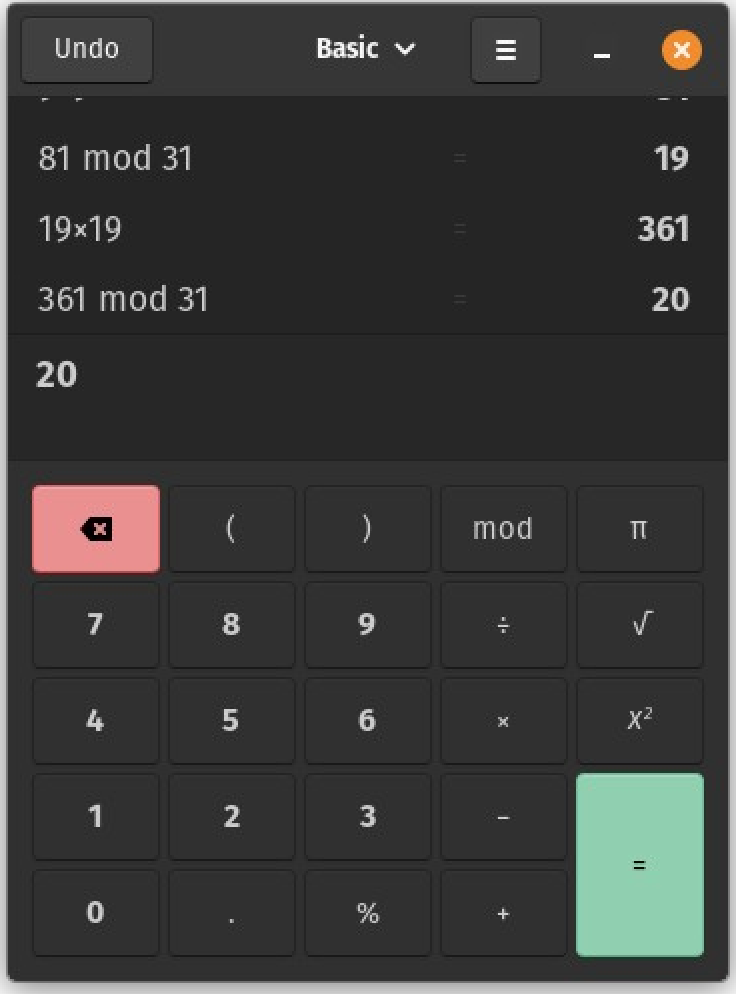
An example…

- 3^{17} mod 31
- 17 = 16 + 1
- 16 = 2^4 , $(((3^2)^2)^2)^2 = 3^{16}$
- All mod 31…
  - $3^1=3$, $3^2=9$, …


Undo
Basic
81 mod 31
19
9×9
81
81 mod 31
19
19
(
)
mod
π
7
8
9
÷
√
4
5
6
×
x²
1
2
3
−
=
0
.
%
+

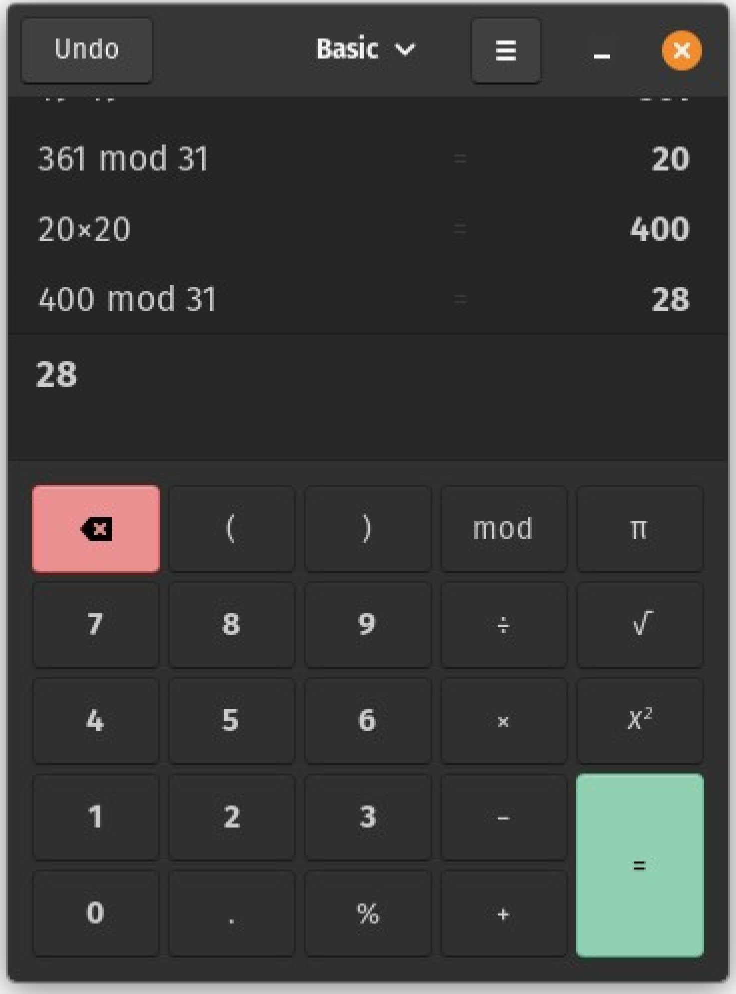
An example…

- 3^{17} mod 31
- 17 = 16 + 1
- 16 = 2^4 , $(((3^2)^2)^2)^2=3^{16}$
- All mod 31…
  - $3^1=3$, $3^2=9$, $3^4=19$, …


Undo
Basic
81 mod 31 = 19
19×19 = 361
361 mod 31 = 20
20
( ) mod π
7 8 9 ÷ √
4 5 6 × x²
1 2 3 − =
0 . % +

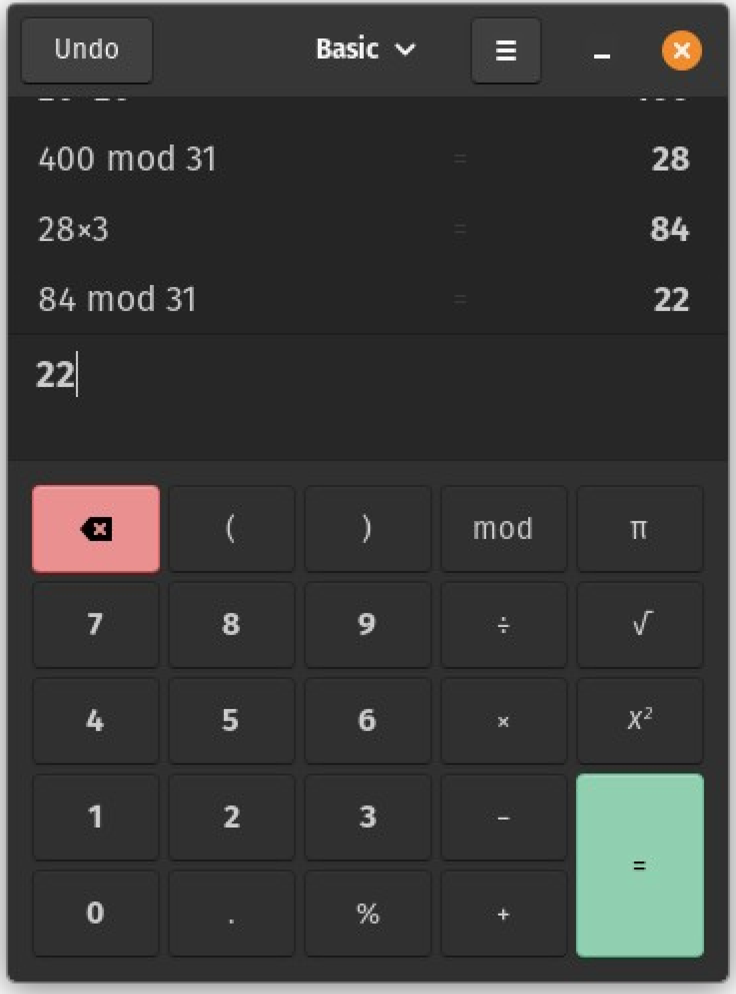
An example…

- 3^{17} mod 31
- $17 = 16 + 1$
- $16 = 2^4$, $(((3^2)^2)^2)^2=3^{16}$
- All mod 31…
  - $3^1=3$, $3^2=9$, $3^4=19$, $3^8=20$, …


Undo
Basic
361 mod 31 = 20
20×20 = 400
400 mod 31 = 28
28
( ) mod π
7 8 9 ÷ √
4 5 6 ×
1 2 3 −
0 . % +
=

An example…

- 3^{17} mod 31
- 17 = 16 + 1
- $16 = 2^4$, $(((3^2)^2)^2)^2=3^{16}$
- All mod 31…
  - $3^1=3$, $3^2=9$, $3^4=19$, $3^8=20$, $3^{16}=28$…


Undo
Basic
400 mod 31 = 28
28×3 = 84
84 mod 31 = 22
22
( ) mod π
7 8 9 ÷ √
4 5 6 × x²
1 2 3 − =
0 . % +

An example…

- 3^{17} mod 31 = $3^{16}3^{1}$ mod 31 = 22
- 17 = 16 + 1
- 16 = 2^4 , $(((3^2)^2)^2)^2=3^{16}$
- All mod 31…
  - $3^1=3$, $3^2=9$, $3^4=19$, $3^8=20$, $3^{16}=28$…

17 in binary is 0b10001