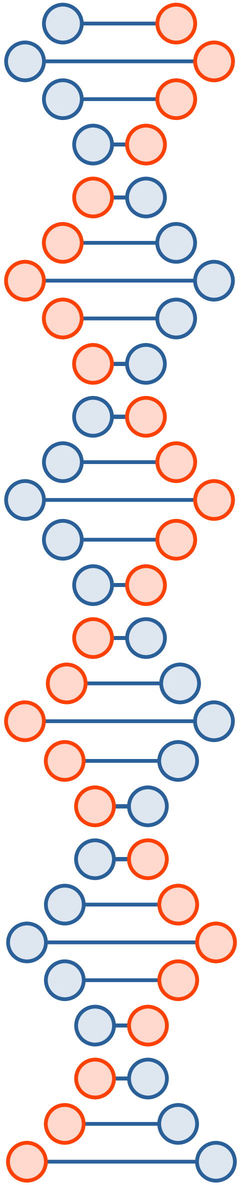
Finite fields, Diffie-Hellman, and fast modular exponentiation

CSE 468 Fall 2025

jedimaestro@asu.edu



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
  - Math!
- 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 (mod 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 (mod 2)
0 * 1 = 0 (mod 2)
1 * 0 = 0 (mod 2)
1 * 1 = 1 (mod 2)

Where have you seen this before?

GF(2)

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

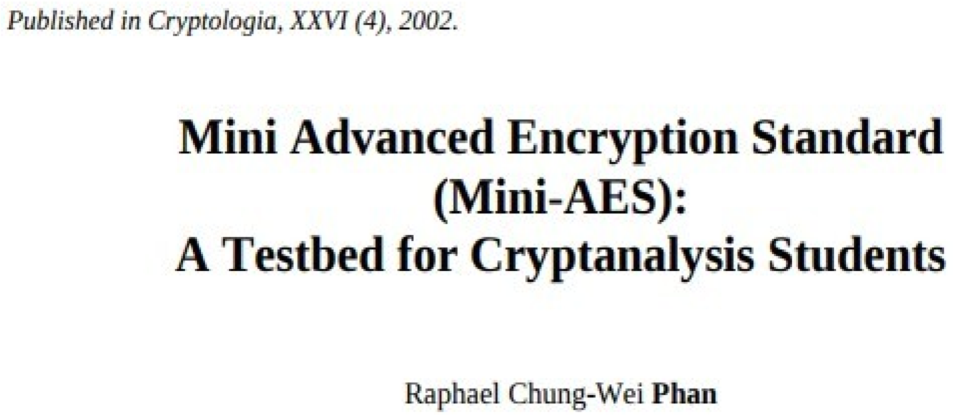
XOR

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

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:

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

This is because:

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

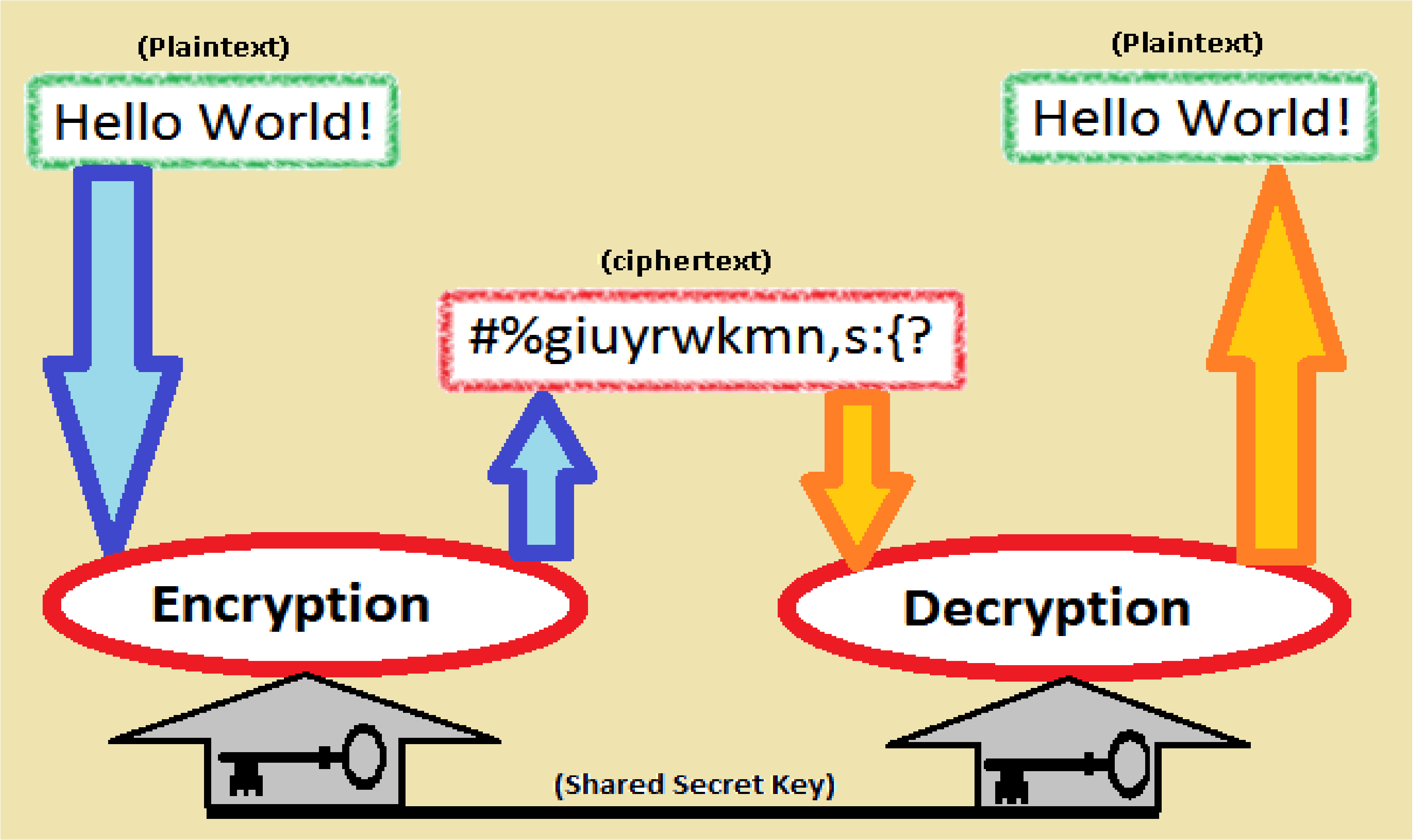
Note that since the coefficients of the polynomials are in {0,1}, then addition is simply exclusive-OR and hence subtraction is also exclusive-OR since exclusive-OR is its own inverse.

Basics of crypto...

- Symmetric encryption
  - Assumes two parties wishing to communicate already have a shared secret
- Asymmetric encryption
  - Makes different assumptions (*e.g.*, that everybody knows the public key or that the eavesdropper is passive)
  - Quantum computers break current algorithms that are used in practice
- Secure hash functions and message authentication

Symmetric Crypto

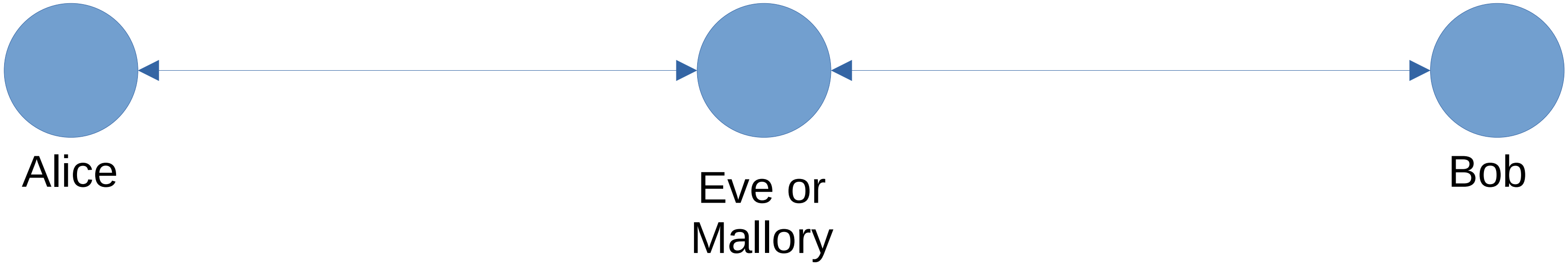
- Confidentiality
- Integrity
- ~~Availability~~
- Authentication
- ~~Non-repudiation~~
- ~~A way to distribute the shared secret keys~~



Source: Wikipedia

Terminology

- Plaintext – before encryption, easy to read
- Ciphertext – after encryption, hopefully indecipherable without the key
- Key – the shared secret, typically just bits that were generated with a high entropy process



WiFi, electric path, or optical… Eve or Mallory get their own copy!
So how to Alice and Bob exchange a key?

A nice video about Diffie-Hellman

- https://www.youtube.com/watch?v=YEBfamv-_do

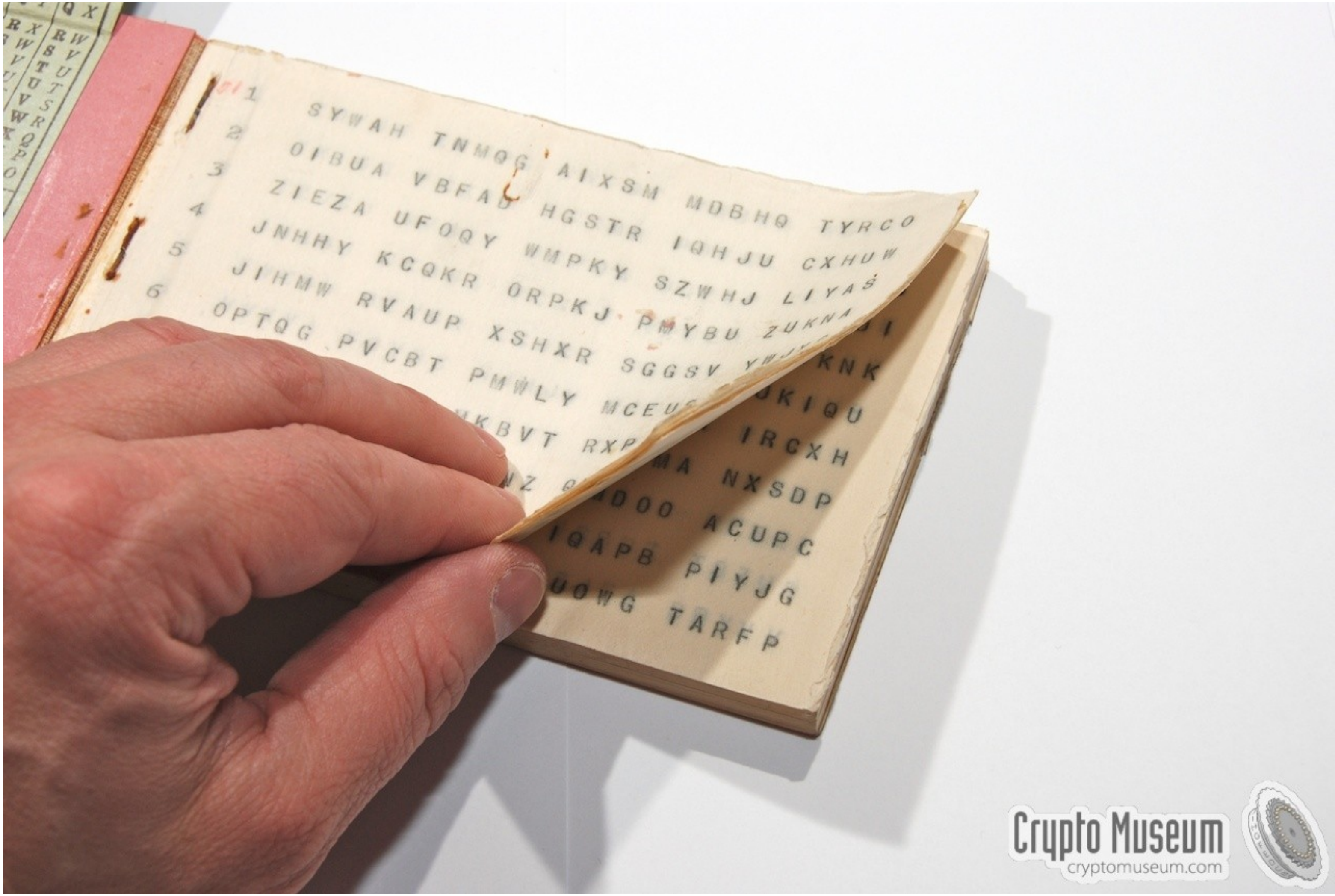
Diffie-Hellman is *asymmetric* crypto

Darknet Diaries, Episode 83

https://darknetdiaries.com/transcript/83/

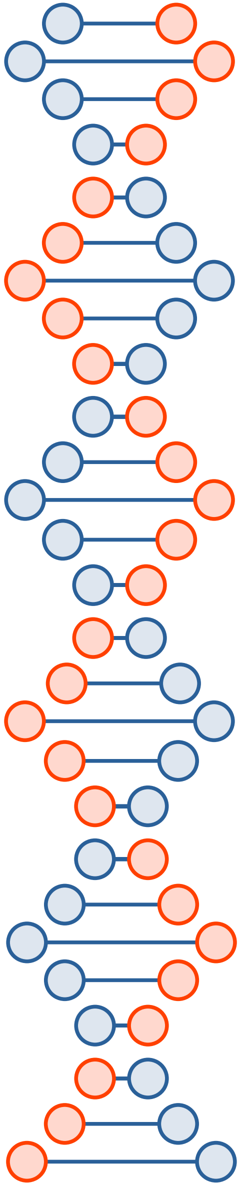
- “There was no concept of doing anything cryptographic in terms of software back in the late 80s. I say this, I’m in contact with a fellow alumni from the InfoSec organization and people that were there years before I was, and I’ve asked. To the best that I have been able to figure out, what we ended up producing which was half paper pad, half key on a floppy, and a computer program that would do the encryption and decryption. That was the first foray into software-based cryptography that NSA produced.”

  --Jeff Man



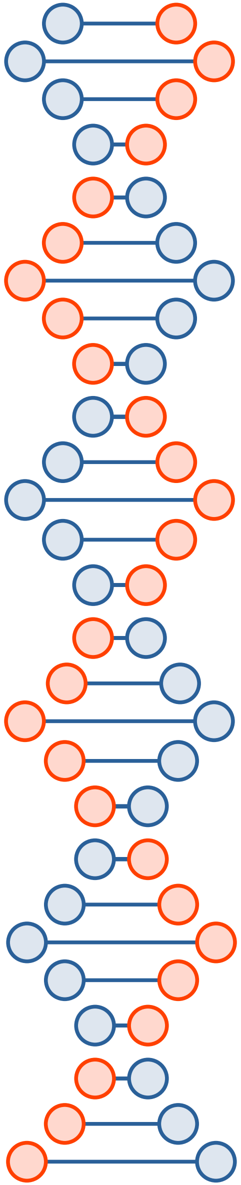
```
1 SYWAH TNMOG AIXSM MOBHQ TYRCO
2 OIBUA VBFAD HGSTR IQHJU CXHUW
3 ZIEZA UFOQY WMPKY SZWHJ LIYAS
4 JNHHY KCQKR ORPKJ PMYBU ZUKNA
5 JIHMW RVAUP XSHXR SGGSV
6 OPTQG PVCBT PMWLY MCEU
```

PX 1000Cr
Crypto Museum
cryptomuseum.com
LIST PRINT
RCVE
LOW MGN MARGIN
SET CLR TAB
CAPS LOCK
SHIFT
INS TXT INSERT
DEL TXT DELETE
SEARCH
TEXT
KEY TXT CODE
CLR LINE CLEAR ALL
LOAD DUMP
ON STOP



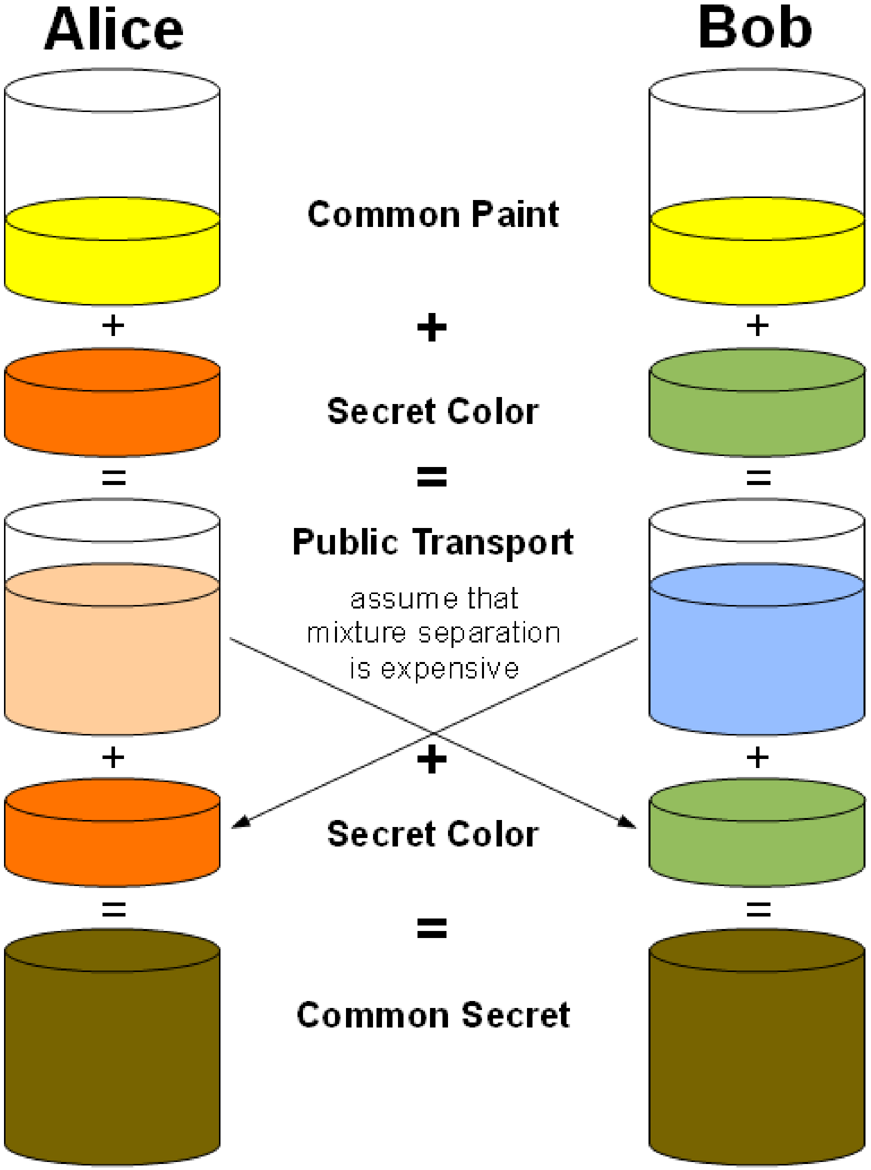
Couple of footnotes

- Diffie-Hellman-Merkle?
- Who was first?
  - Diffie-Hellman conceived and then published 1976
  - GCHQ version conceived 1969, published 1997



Basics...

- https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange



Alice

Bob

Common Paint

+

Secret Color

=

Public Transport

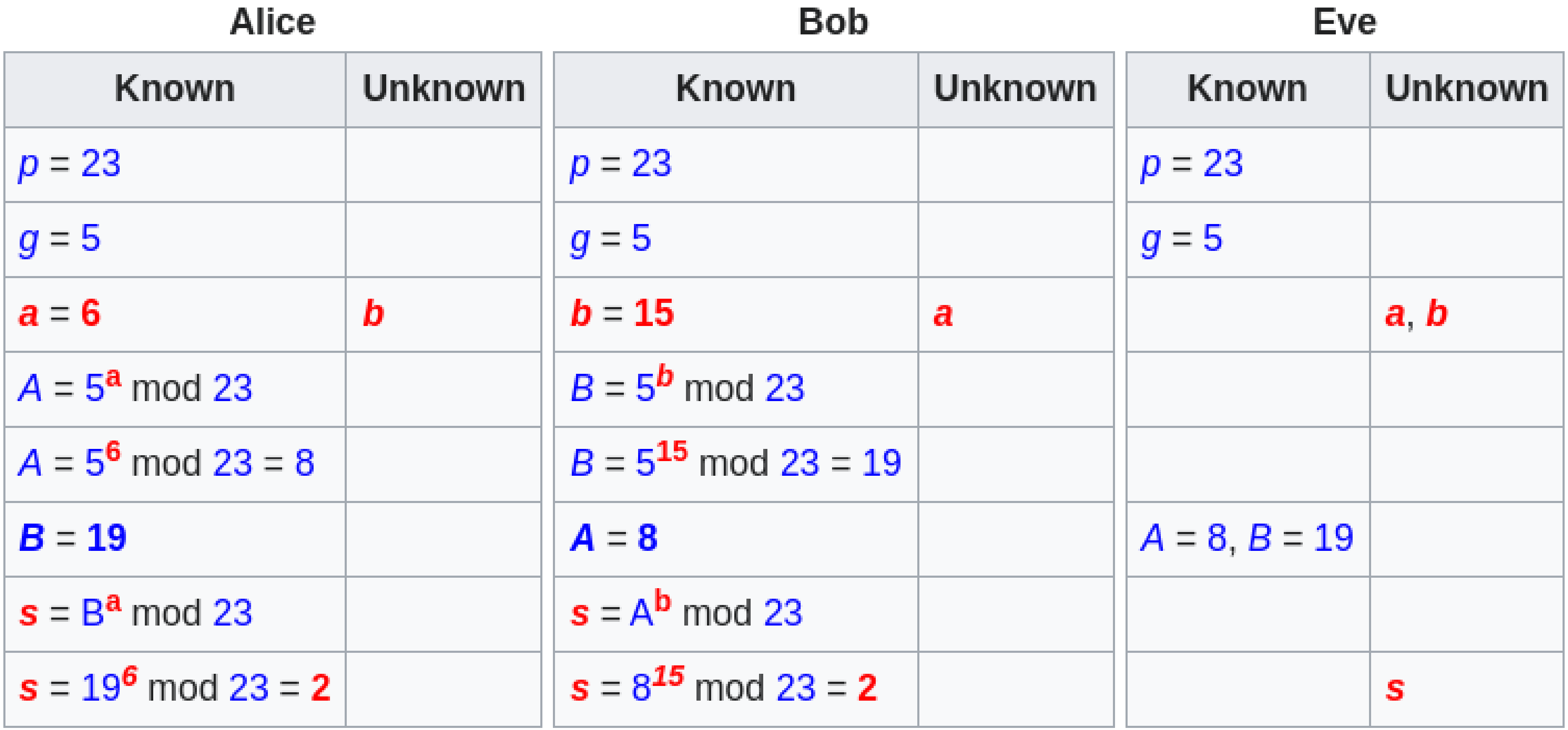
assume that
mixture separation
is expensive

+

Secret Color

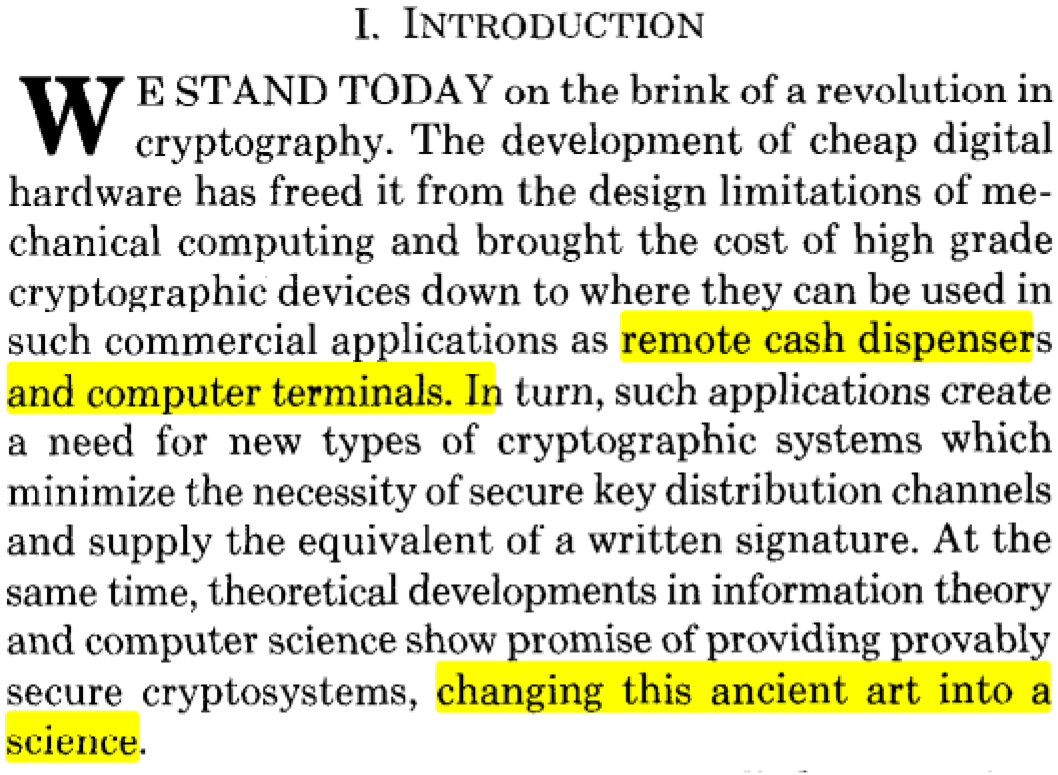
=

Common Secret



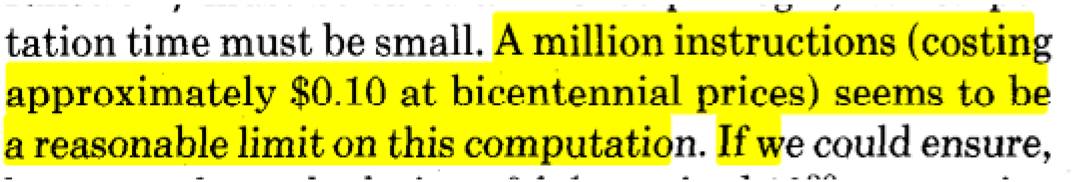
Alice		Bob		Eve	
Known	**Unknown**	**Known**	**Unknown**	**Known**	**Unknown**
$p = 23$		$p = 23$		$p = 23$	
$g = 5$		$g = 5$		$g = 5$	
$a = 6$	***b***	**$b = 15$**	**a**		**a**, ***b***
$A = 5^{a} \bmod 23$		$B = 5^{b} \bmod 23$			
$A = 5^{6} \bmod 23 = 8$		$B = 5^{15} \bmod 23 = 19$			
$B = 19$		**$A = 8$**		$A = 8$, $B = 19$	
$s = B^{a} \bmod 23$		$s = A^{b} \bmod 23$			
$s = 19^{6} \bmod 23 = 2$		$s = 8^{15} \bmod 23 = 2$			**s**

The paper...

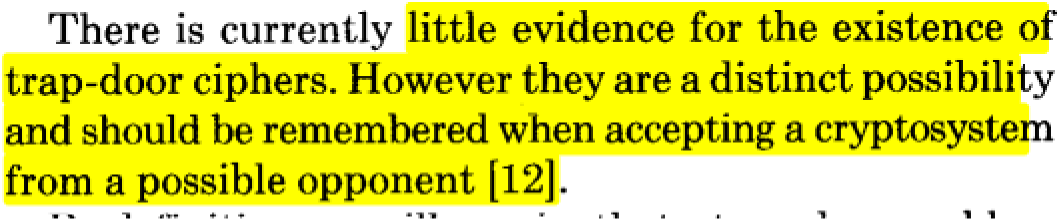


I. Introduction

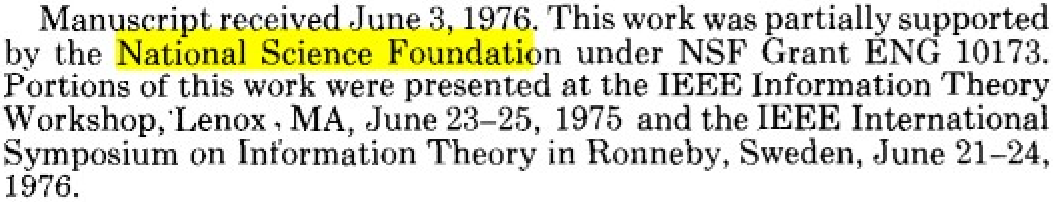
WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.



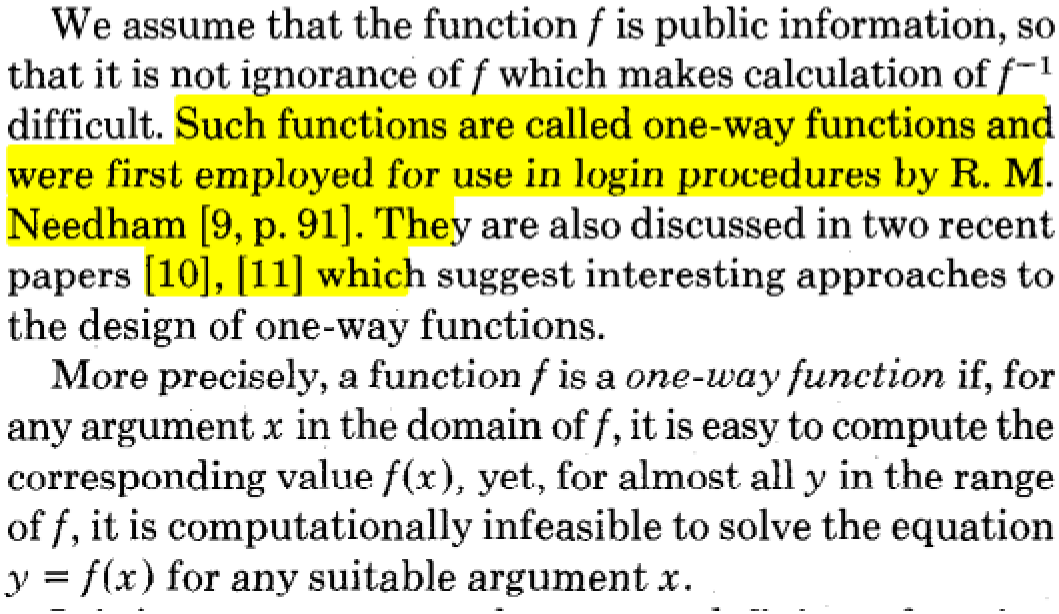
tation time must be small. A million instructions (costing approximately $0.10 at bicentennial prices) seems to be a reasonable limit on this computation. If we could ensure,



There is currently little evidence for the existence of trap-door ciphers. However they are a distinct possibility and should be remembered when accepting a cryptosystem from a possible opponent [12].

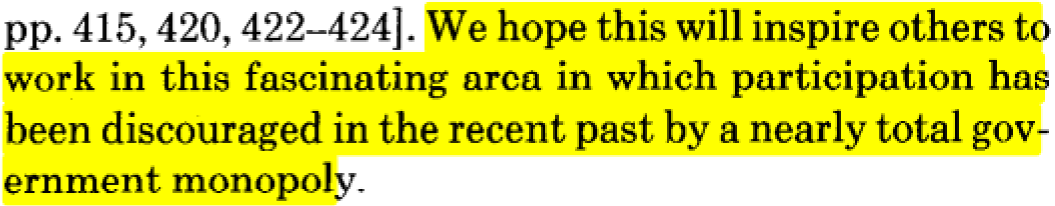


Manuscript received June 3, 1976. This work was partially supported by the National Science Foundation under NSF Grant ENG 10173. Portions of this work were presented at the IEEE Information Theory Workshop, Lenox, MA, June 23–25, 1975 and the IEEE International Symposium on Information Theory in Ronneby, Sweden, June 21–24, 1976.



We assume that the function f is public information, so that it is not ignorance of f which makes calculation of f^{-1} difficult. Such functions are called one-way functions and were first employed for use in login procedures by R. M. Needham [9, p. 91]. They are also discussed in two recent papers [10], [11] which suggest interesting approaches to the design of one-way functions.

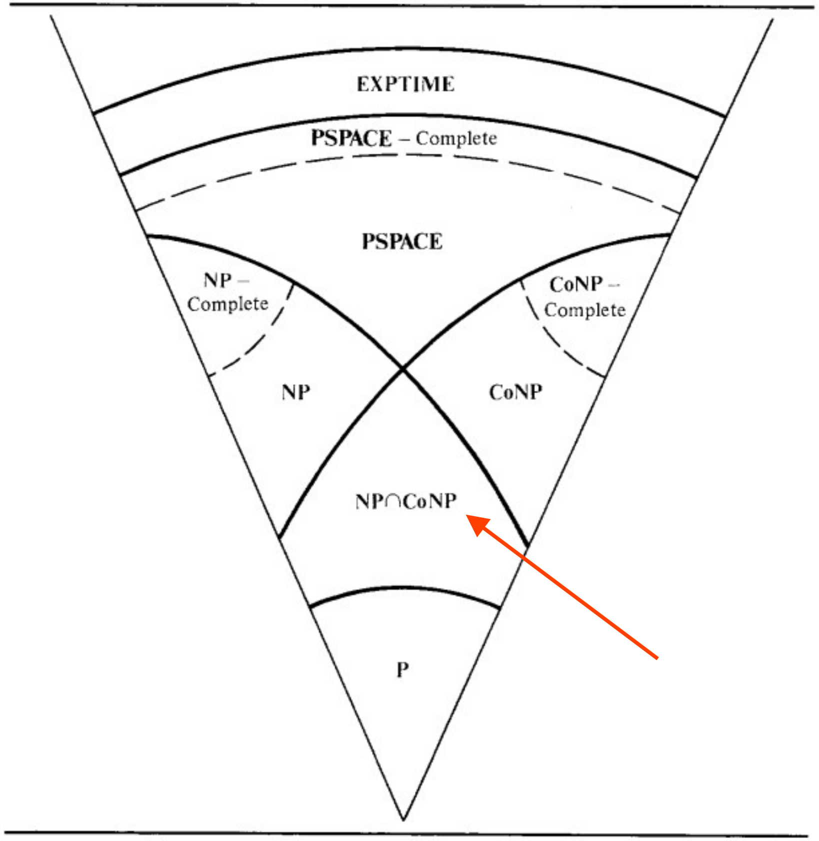
More precisely, a function f is a *one-way function* if, for any argument x in the domain of f, it is easy to compute the corresponding value $f(x)$, yet, for almost all y in the range of f, it is computationally infeasible to solve the equation $y = f(x)$ for any suitable argument x.

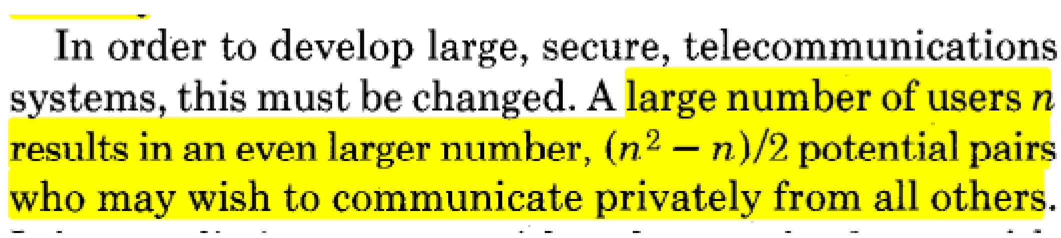


pp. 415, 420, 422–424]. We hope this will inspire others to work in this fascinating area in which participation has been discouraged in the recent past by a nearly total government monopoly.

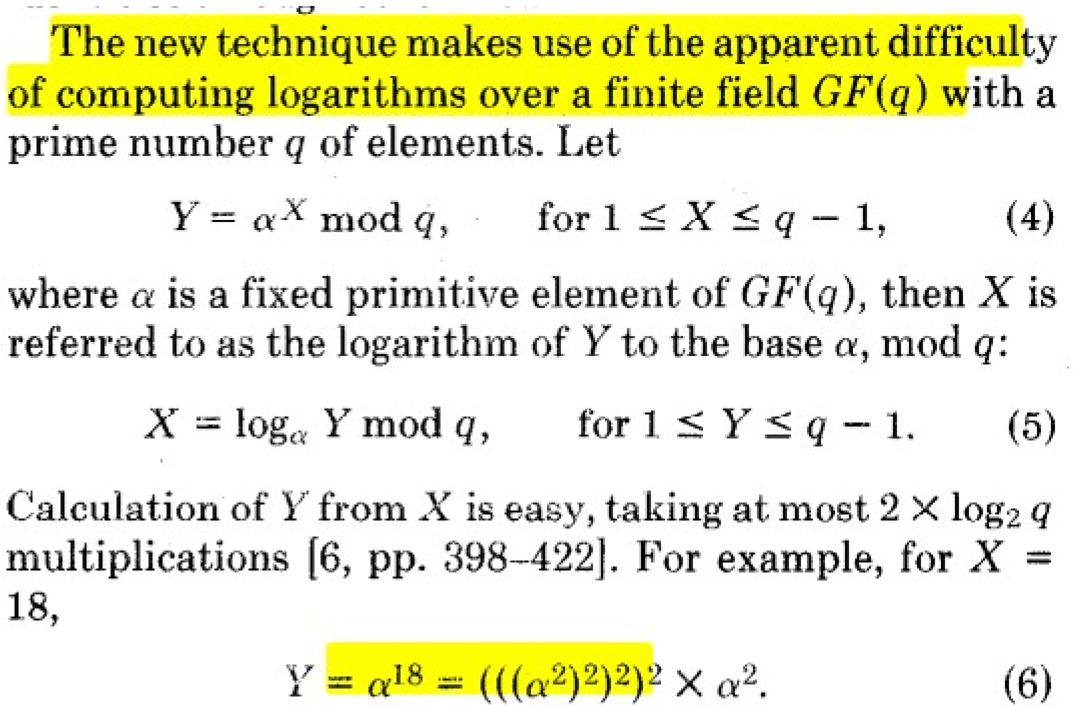
https://faculty.nps.edu/dedennin/publications/Denning-CryptographyDataSecurity.pdf

FIGURE 1.18 Complexity classes.





In order to develop large, secure, telecommunications systems, this must be changed. A large number of users n results in an even larger number, $(n^2 - n)/2$ potential pairs who may wish to communicate privately from all others.



The new technique makes use of the apparent difficulty of computing logarithms over a finite field $GF(q)$ with a prime number q of elements. Let

$$Y = \alpha^X \bmod q, \qquad \text{for } 1 \leq X \leq q - 1, \qquad (4)$$

where α is a fixed primitive element of $GF(q)$, then X is referred to as the logarithm of Y to the base α, mod q:

$$X = \log_\alpha Y \bmod q, \qquad \text{for } 1 \leq Y \leq q - 1. \qquad (5)$$

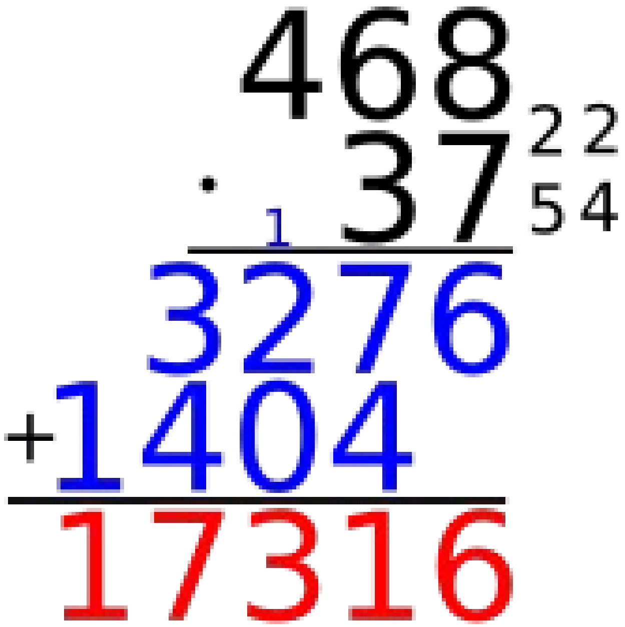
Calculation of Y from X is easy, taking at most $2 \times \log_2 q$ multiplications [6, pp. 398–422]. For example, for $X = 18$,

$$Y = \alpha^{18} = (((\alpha^2)^2)^2)^2 \times \alpha^2. \qquad (6)$$

RSA *vs.* DH

- Diffie-Hellman (1976)
  - Key exchange
  - *Both sides get to choose something random*
- RSA (1977)
  - Encryption
  - Signatures

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

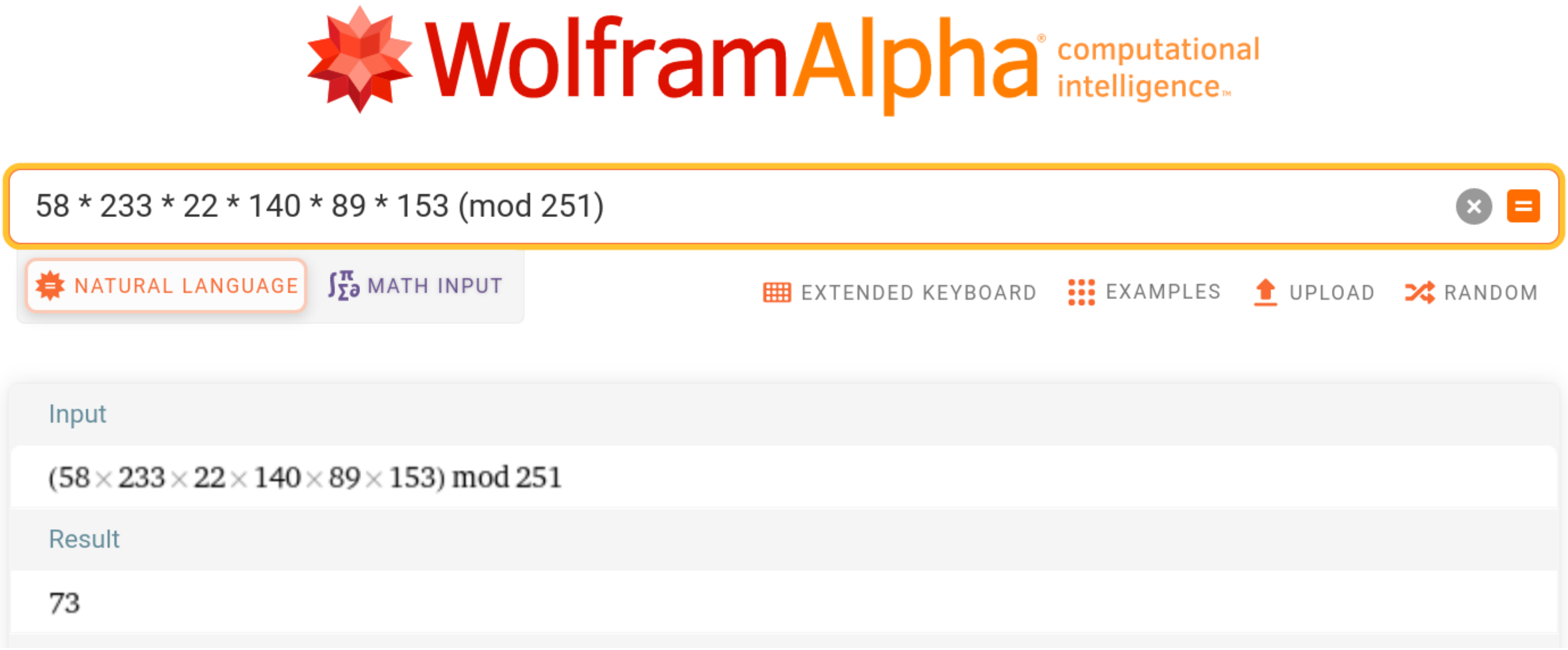
Better way

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

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

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

  $= 73$



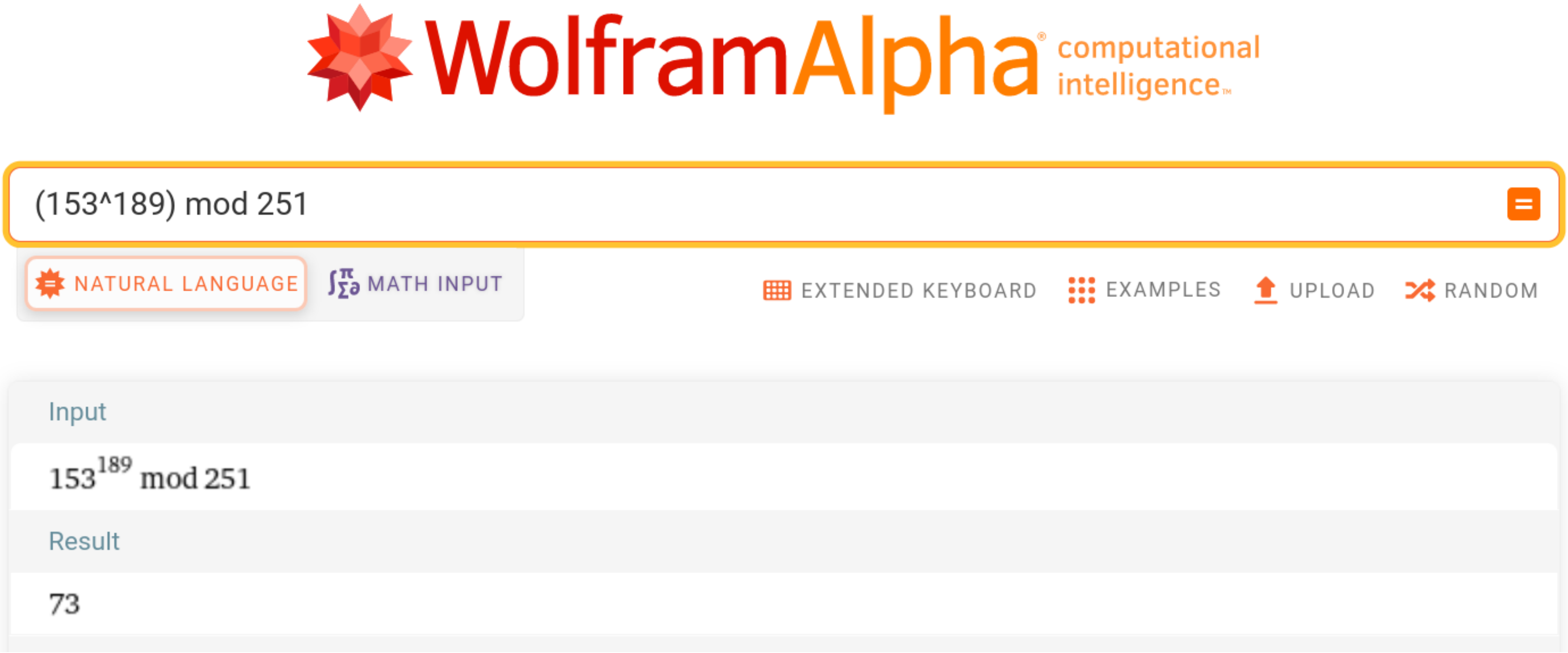
58 * 233 * 22 * 140 * 89 * 153 (mod 251)

Input

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

Result

73



(153^189) mod 251

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.

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

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

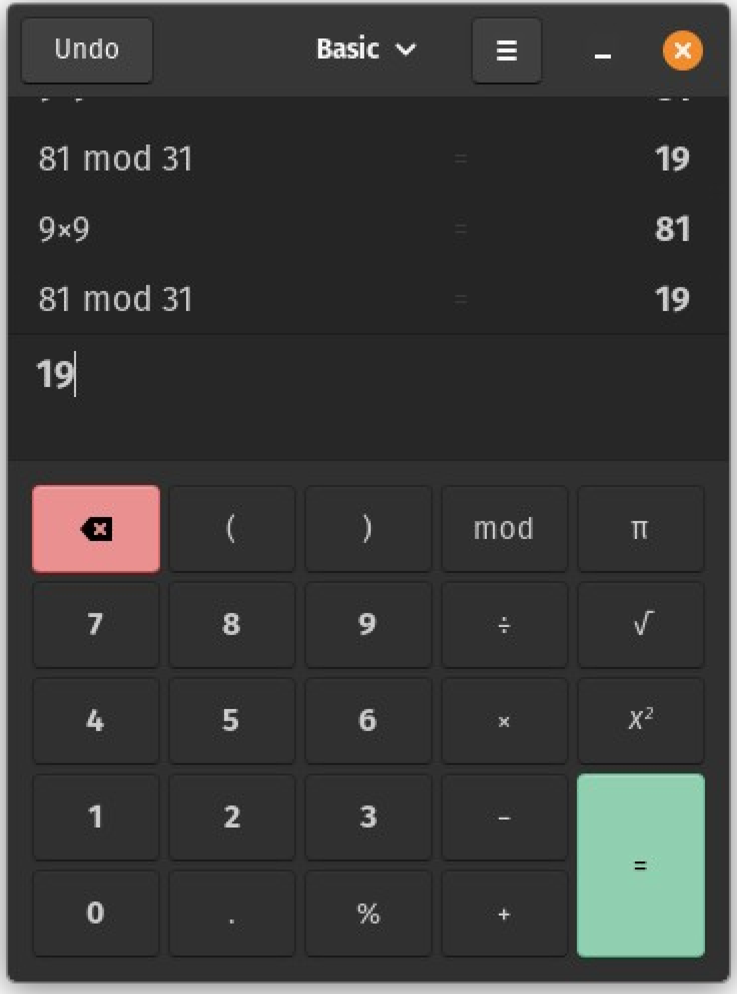
$$153^{189} \equiv 73 \pmod{251}$$

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

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

An example…

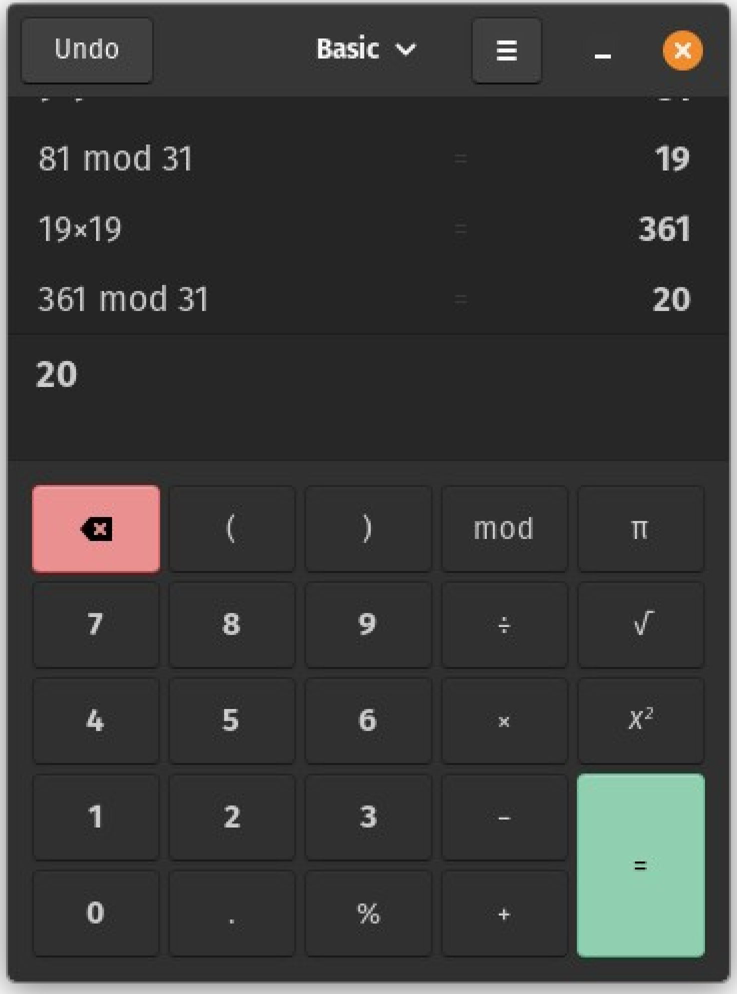
- 3^{17} mod 31
- $17 = 16 + 1$
- $16 = 2^4$, $(((3^2)^2)^2)^2=3^{16}$
- All mod 31…
  - $3^1=3$, $3^2=9$, …



```
81 mod 31        =        19
9×9              =        81
81 mod 31        =        19
19
```

An example…

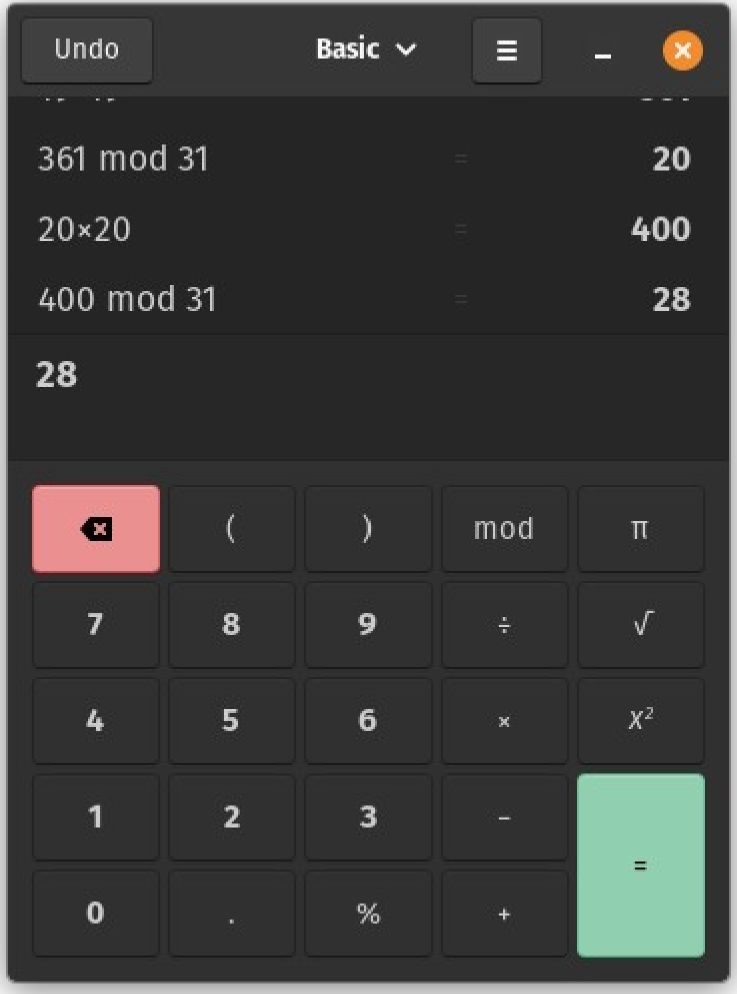
- 3^{17} mod 31
- 17 = 16 + 1
- 16 = 2^4 , $(((3^2)^2)^2)^2=3^{16}$
- All mod 31…
  - $3^1=3$, $3^2=9$, $3^4=19$, …



```
81 mod 31     =     19
19×19         =    361
361 mod 31    =     20
20
```

An example…

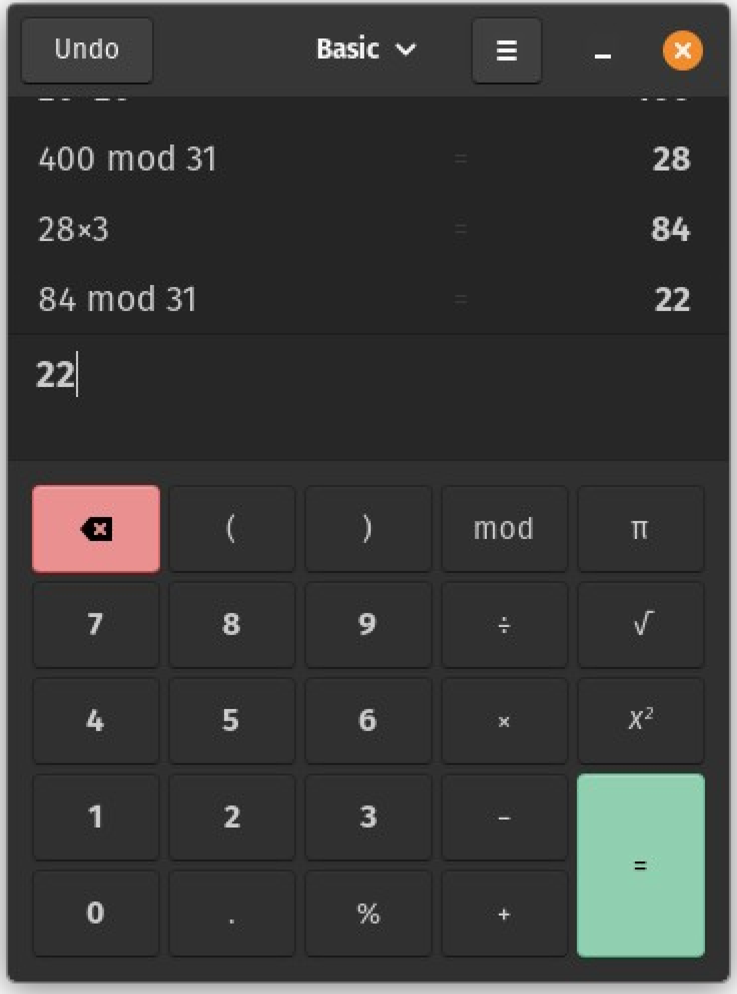
- 3^{17} mod 31
- 17 = 16 + 1
- $16 = 2^4$, $(((3^2)^2)^2)^2=3^{16}$
- All mod 31…
  - $3^1=3$, $3^2=9$, $3^4=19$, $3^8=20$, …



```
361 mod 31          =          20
20×20               =         400
400 mod 31          =          28
28
```

An example…

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



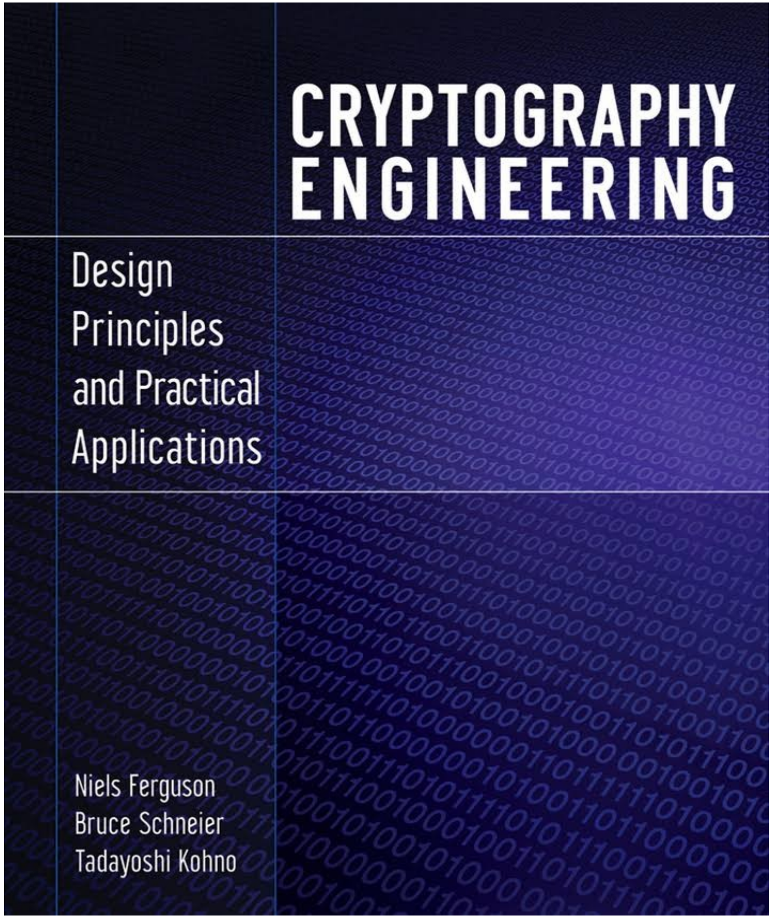
```
400 mod 31     =     28
28×3           =     84
84 mod 31      =     22
22
```

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

Cryptography Engineering by Ferguson *et al.*



CRYPTOGRAPHY ENGINEERING

Design Principles and Practical Applications

Niels Ferguson
Bruce Schneier
Tadayoshi Kohno