The preceding was a brief overview

Now on to the real lecture

Cryptographic Tools

- Cryptographic algorithms important element in security services
- Review various types of elements
  - symmetric encryption
  - secure hash functions
  - public-key (asymmetric) encryption
  - digital signatures and key management
  - random numbers (and just how random are they?)
- Example use to encrypt stored data or networked data
  - “data at rest”
  - “data in motion”

Symmetric and Asymmetric

ENCRYPTION

Symmetric Encryption

- the universal technique for providing confidentiality for transmitted or stored data
- also referred to as conventional encryption or single-key encryption
- two requirements for secure use:
  - need a strong encryption algorithm
  - sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure

SYMMETRIC ENCRYPTION

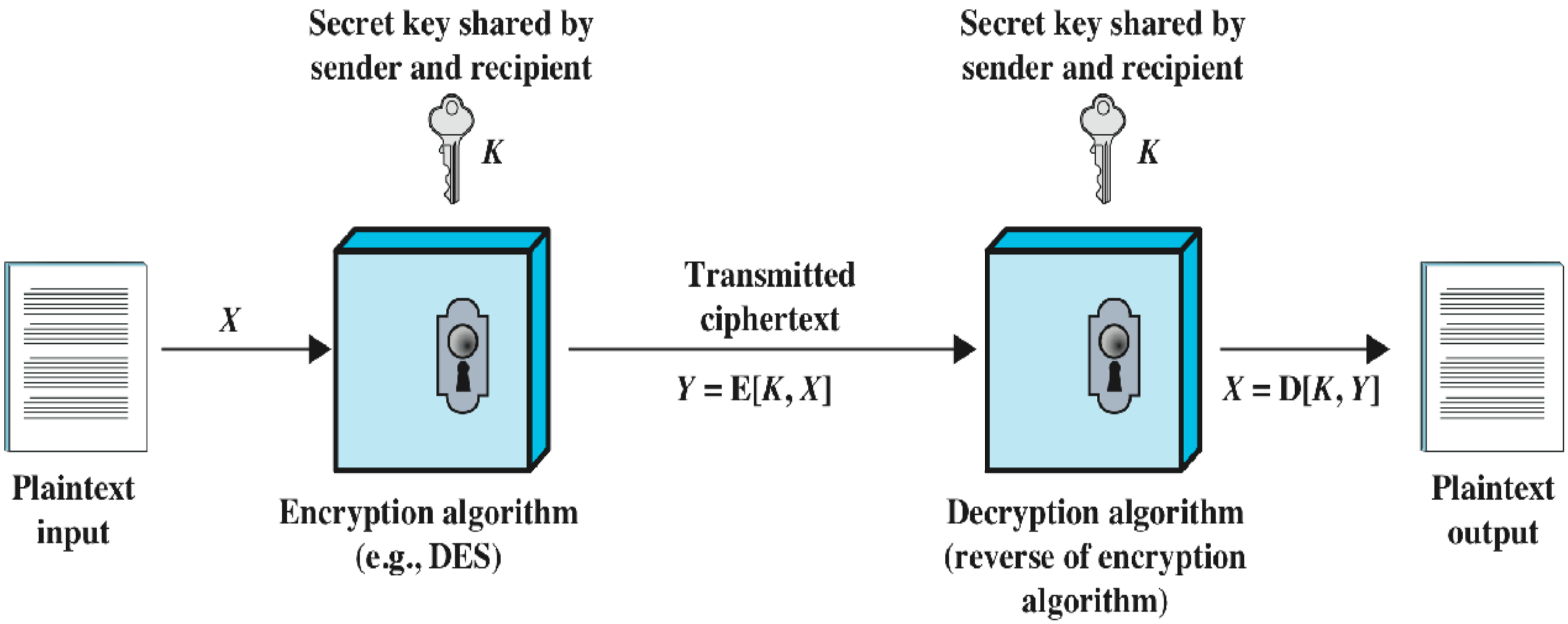

Secret key shared by sender and recipient
K
Secret key shared by sender and recipient
K
Plaintext input
X
Encryption algorithm (e.g., DES)
Transmitted ciphertext
$Y = \mathrm{E}[K, X]$
Decryption algorithm (reverse of encryption algorithm)
$X = \mathrm{D}[K, Y]$
Plaintext output

Figure 2.1 Simplified Model of Symmetric Encryption

Block & Stream Ciphers

Block Cipher

- processes the input one block of elements at a time
- produces an output block for each input block
- can reuse keys
- more common

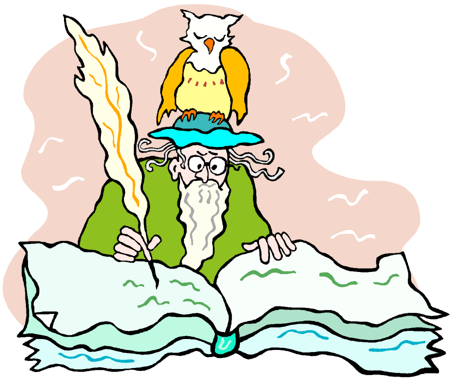
Stream Cipher

- processes the input elements continuously
- produces output one element at a time
- primary advantage is that they are almost always faster and use far less code
- encrypts plaintext one byte at a time
- pseudorandom stream is one that is unpredictable without knowledge of the input key

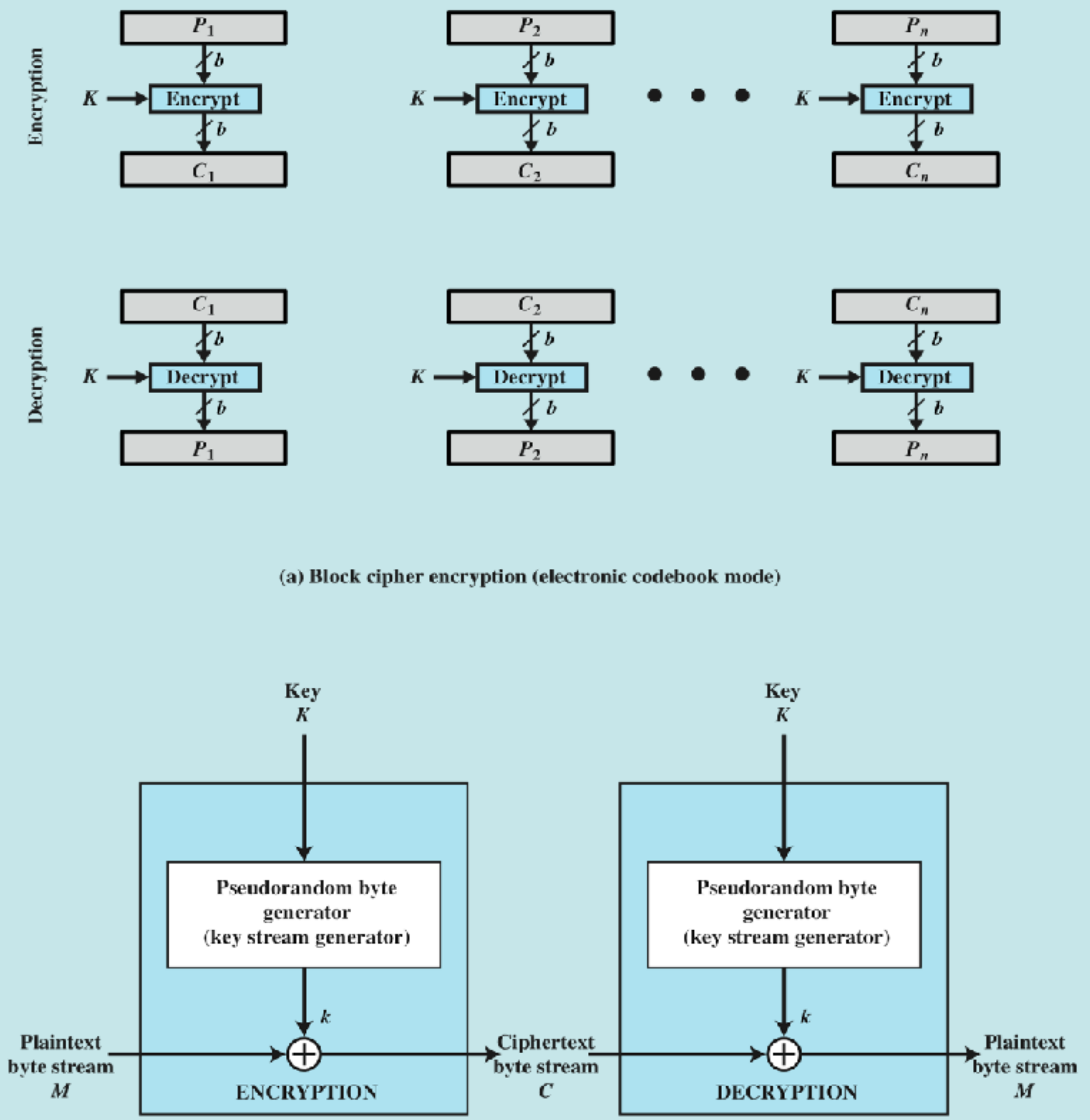
Block Cipher Concepts

1. Divide (plaintext) Data Into Fixed Blocks
   - DES divides message into 64 bit blocks
2. Apply The Algorithm to Each Block
   - Input is block and symmetric key
   - Output is a block of encrypted data
3. Transmit the Encrypted Block
4. Decrypt the Block
   - Input is block and symmetric key
   - Output is a block of decrypted data

Block Cipher Encryption



Stream Encryption


Encryption
P_1
b
K
Encrypt
C_1
P_2
Encrypt
C_2
P_n
Encrypt
C_n
Decryption
C_1
Decrypt
P_1
C_2
Decrypt
P_2
C_n
Decrypt
P_n
Key
K
Pseudorandom byte generator (key stream generator)
k
Plaintext byte stream M
ENCRYPTION
Ciphertext byte stream C
Key
K
Pseudorandom byte generator (key stream generator)
k
DECRYPTION
Plaintext byte stream M

(a) Block cipher encryption (electronic codebook mode)

(b) Stream encryption

Figure 2.3 Types of Symmetric Encryption

Data Encryption Standard (DES)

- **the most widely used encryption scheme**
  - **FIPS PUB 46**
  - **referred to as the Data Encryption Algorithm (DEA)**
  - **uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block**

- **strength concerns:**
  - **concerns about algorithm**
    - **DES is the most studied encryption algorithm in existence**
  - **use of 56-bit key**
    - **Electronic Frontier Foundation (EFF) announced in July 1998 that it had broken a DES encryption**

Attacking Symmetric Encryption

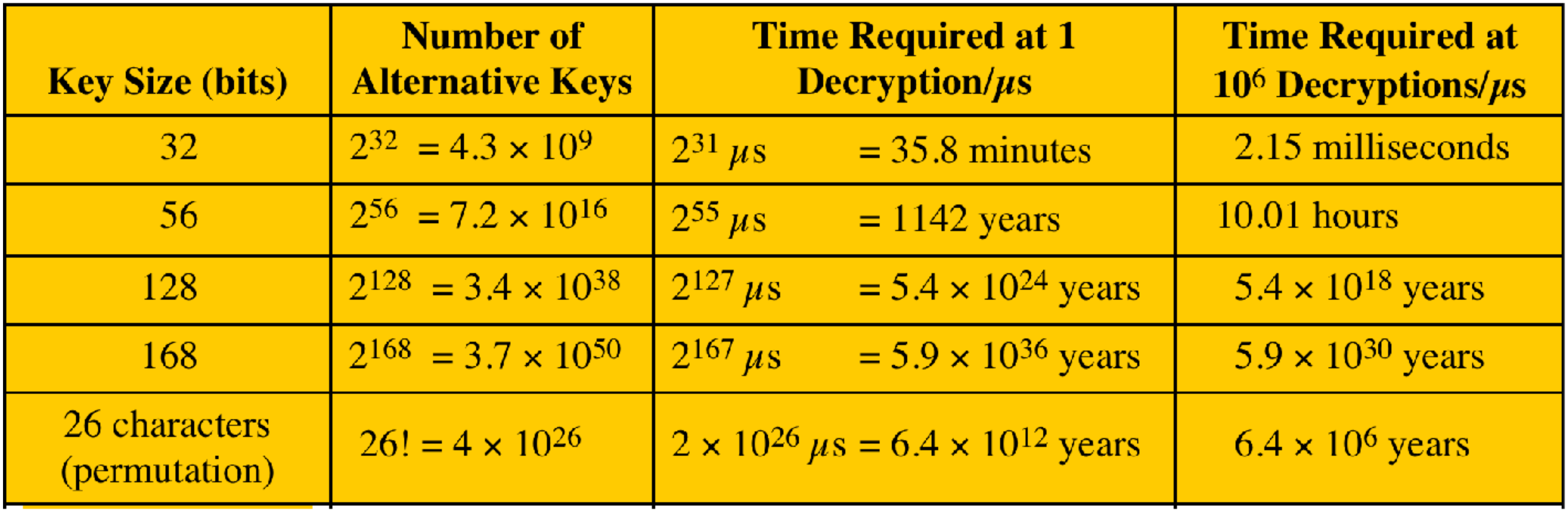
Cryptanalytic Attacks

- rely on:
  - nature of the algorithm
  - some knowledge of the general characteristics of the plaintext
  - some sample plaintext-ciphertext pairs
- exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or the key being used
  - if successful all future and past messages encrypted with that key are compromised

Brute-Force Attack

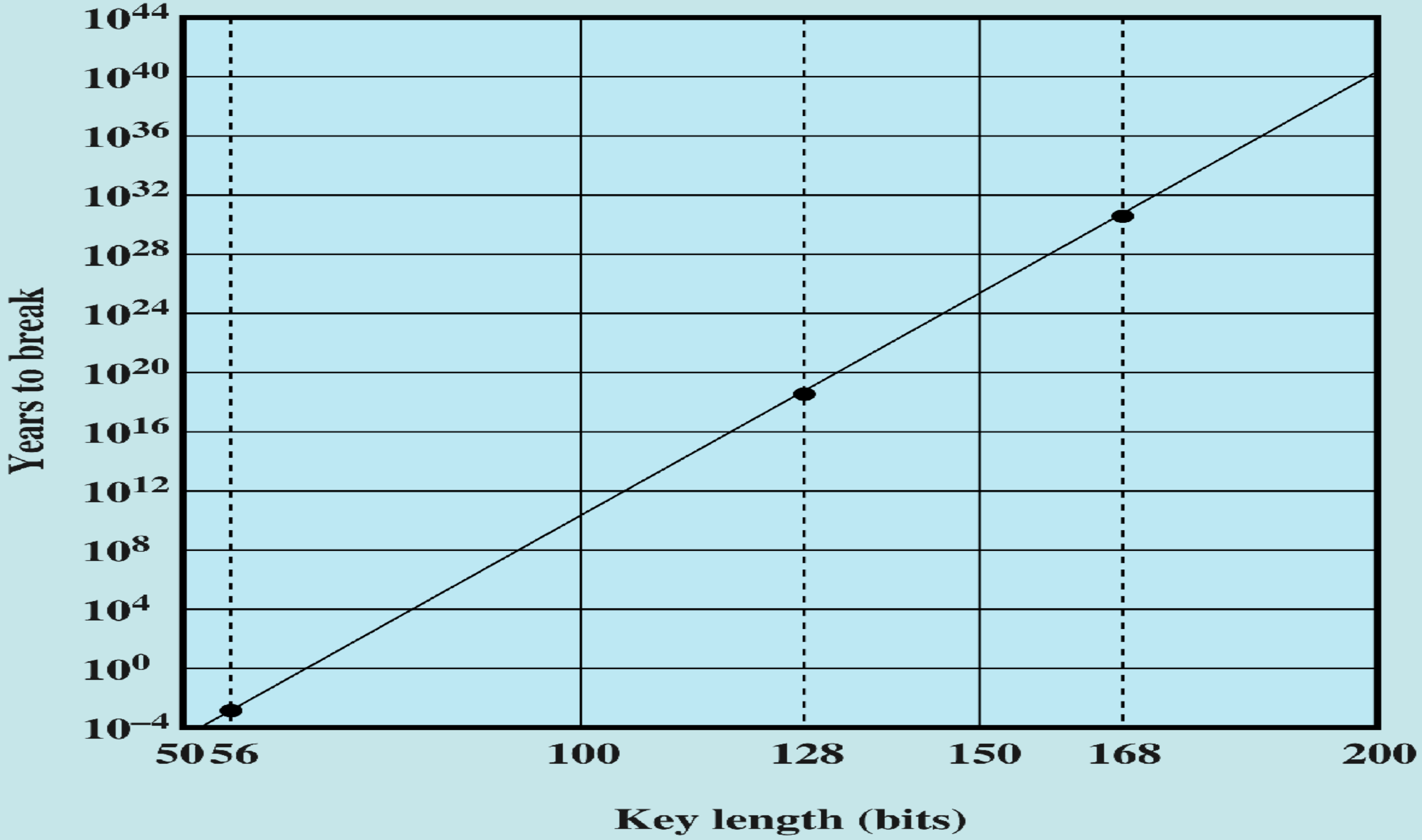
- try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained
  - on average half of all possible keys must be tried to achieve success

Exhaustive Key Search



Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/μs	Time Required at 10^6 Decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} years	6.4×10^6 years

- **Average Time Required for Exhaustive Key Search**



- **Figure 2.2 Time to Break a Code (assuming 106 decryptions/ms)** The graph assumes that a symmetric encryption algorithm is attacked using
- a brute-force approach of trying all possible keys

Triple DES (3DES)

- repeats basic DES algorithm three times using either two or three unique keys
- first standardized for use in financial applications in ANSI standard X9.17 in 1985
- attractions:
  - 168-bit key length overcomes the vulnerability to brute-force attack of DES
  - underlying encryption algorithm is the same as in DES
- drawbacks:
  - algorithm is sluggish in software
  - uses a 64-bit block size

Advanced Encryption Standard (AES)

needed a replacement for 3DES

- 3DES was not reasonable for long term use

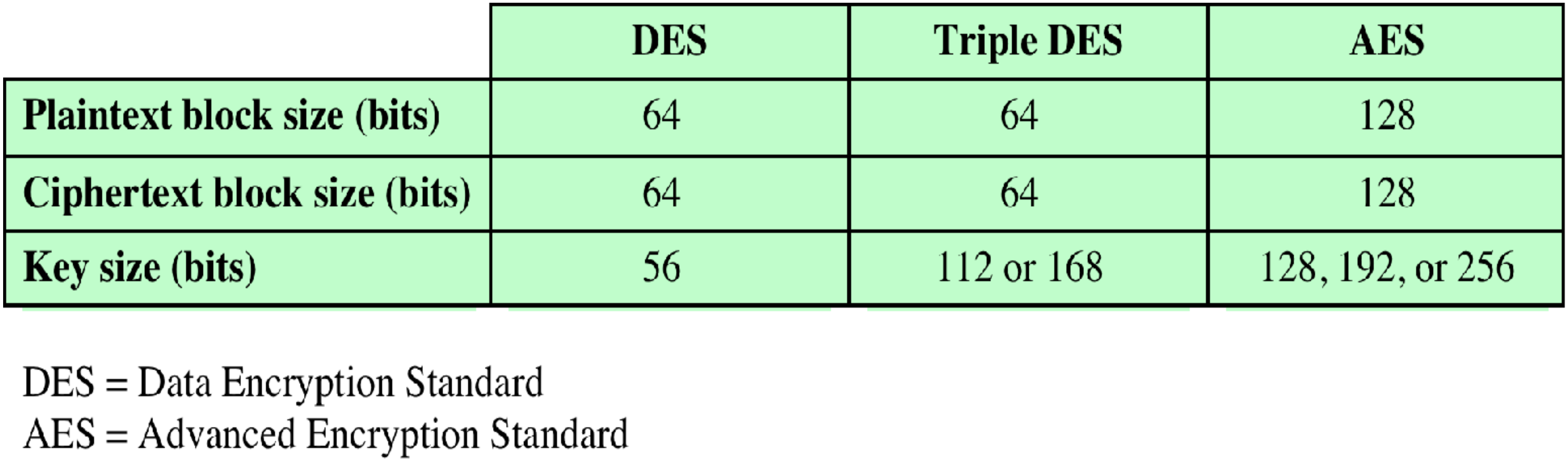
NIST called for proposals for a new AES in 1997

- should have a security strength equal to or better than 3DES
- significantly improved efficiency
- symmetric block cipher
- 128 bit data and 128/192/256 bit

selected Rijndael in November 2001

- published as FIPS 197

DES, 3DES, and AES



	DES	Triple DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192, or 256

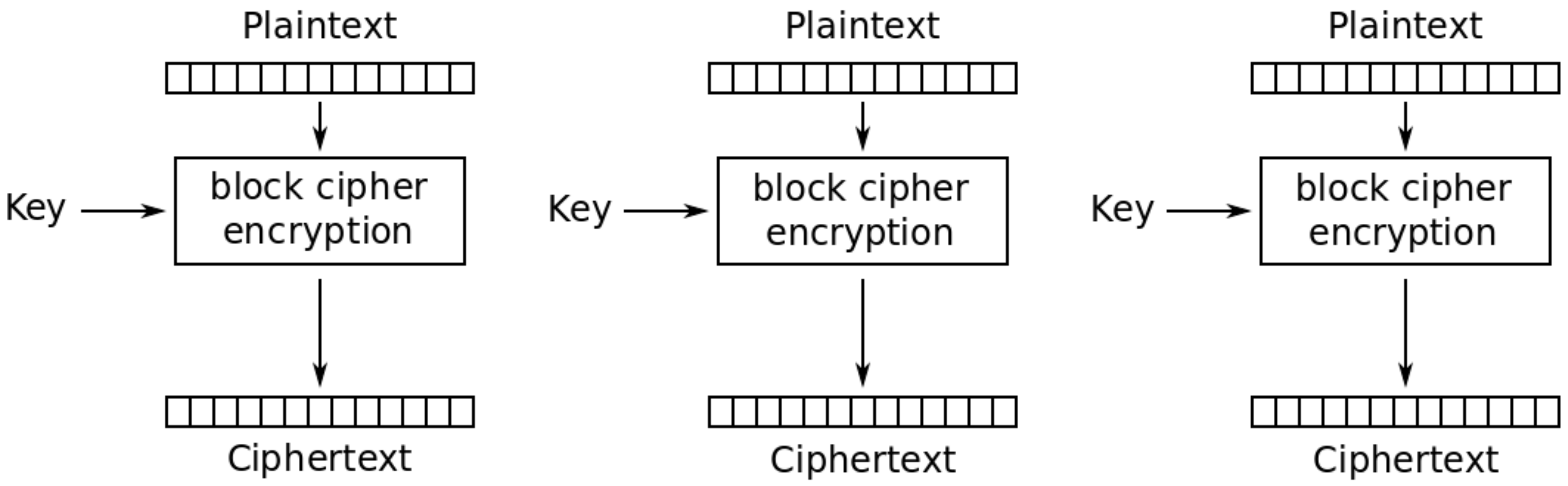
DES = Data Encryption Standard
AES = Advanced Encryption Standard

- **Comparison of Three Popular Symmetric Encryption Algorithms**

Practical Security Issues

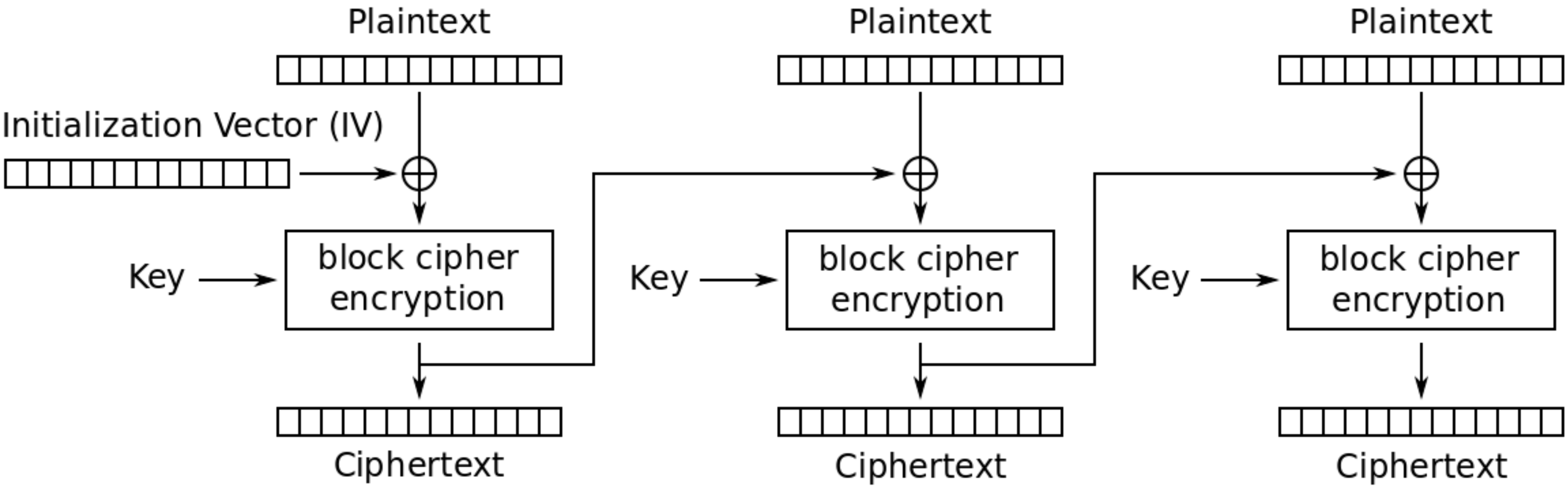
- typically symmetric encryption is applied to a unit of data larger than a single 64-bit or 128-bit block
- electronic codebook (ECB) mode is the simplest approach to multiple-block encryption
  - each block of plaintext is encrypted using the same key
  - cryptanalysts may be able to exploit regularities in the plaintext
- modes of operation
  - alternative techniques developed to increase the security of symmetric block encryption for large sequences
  - overcomes the weaknesses of ECB

Block Encryption — ECB mode



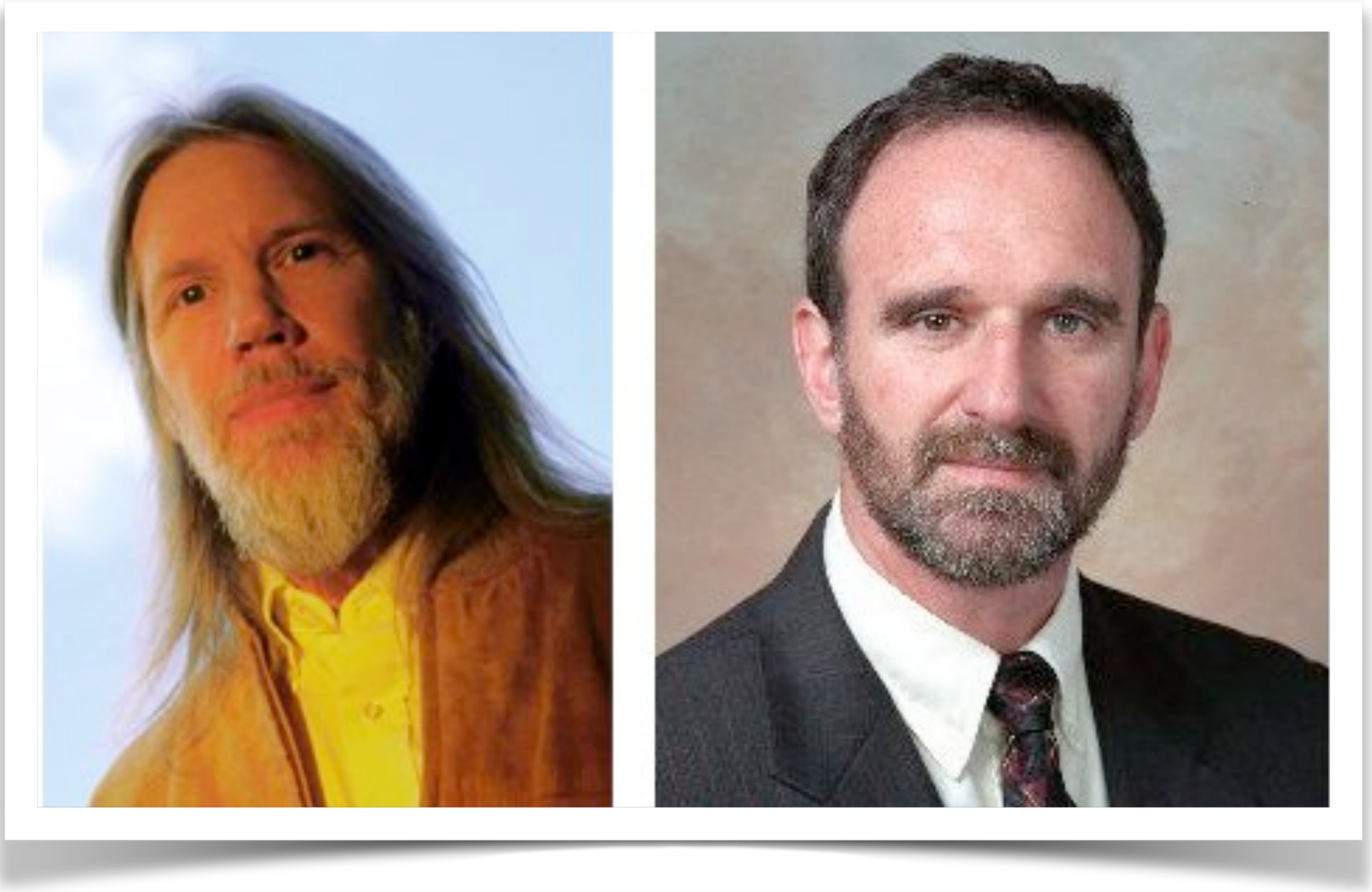
Electronic Codebook (ECB) mode encryption

Block Encryption — mode



Cipher Block Chaining (CBC) mode encryption

Public Key Encryption Pioneers Diffie and Hellman



Diffie-Hellman Key Exchange

using a “coloring” analogy

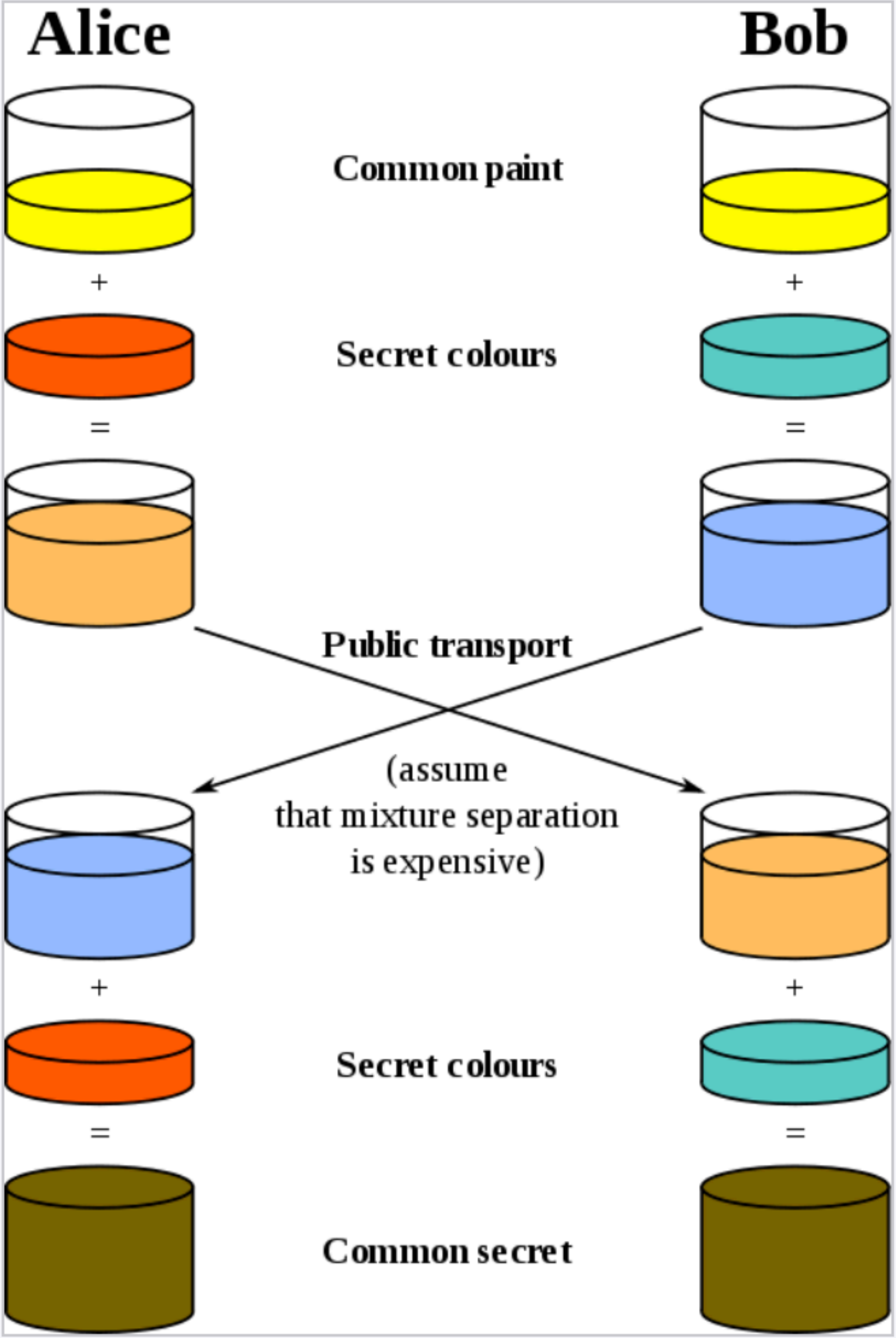


Illustration of the idea behind Diffie–Hellman key exchange

source: wikipedia

Public-Key Encryption Structure

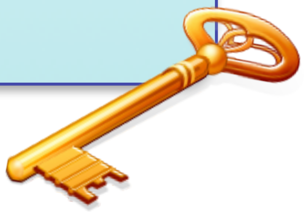
publicly proposed by Diffie and Hellman in 1976

based on mathematical functions

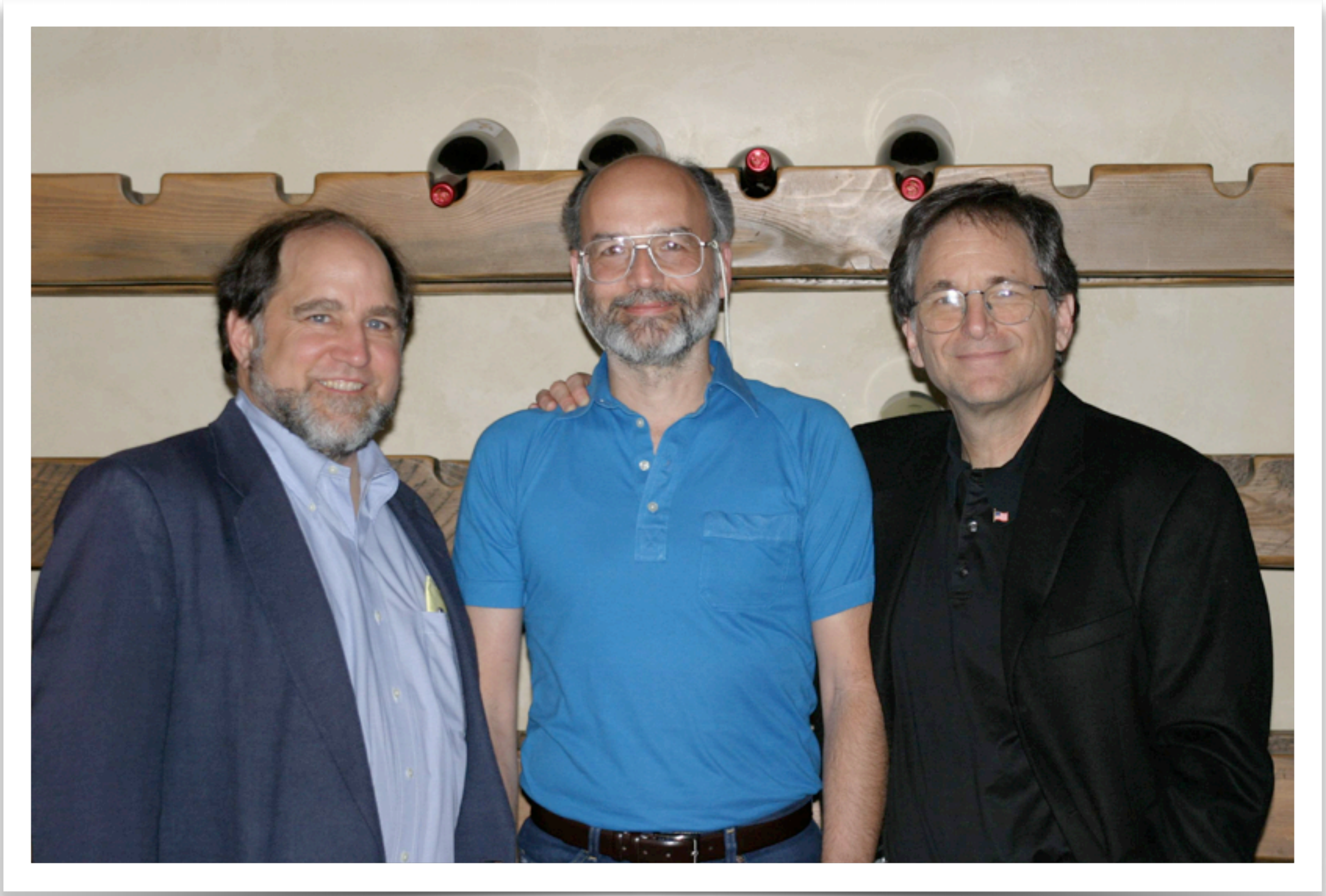
asymmetric

- uses two separate keys
- public key and private key
- public key is made public for others to use

some form of protocol is needed for distribution



RSA algorithm: Ron Rivest, Adi Shamir & Len Adleman



Very Simple Key-Pair Example

- Key-pairs are normally derived from very large prime numbers (more later)
- Algorithms
  - `Encrypt: ciphertext = plaintext^e mod n`
  - `Decrypt: plaintext = ciphertext^d mod n`

- But…. Let's use three small integers for simplicity:
  - Set keys to <e=3, n=55> and <d=27, n=55>
- Example Encrypt the number 43 into 32 and back again to 43 (use the linux "bc" command to do math)

Another example

- Using prime numbers 3 and 11
  - 3 x 11 = 33

- Public Key based on 2 numbers, 7 and 33: encrypt as n^7 mod 33

- Private key based on 3 and 33: decrypt as n^3 mod 33

- Not very secure, not very practical (can’t encrypt large numbers). So use really BIG prime numbers.

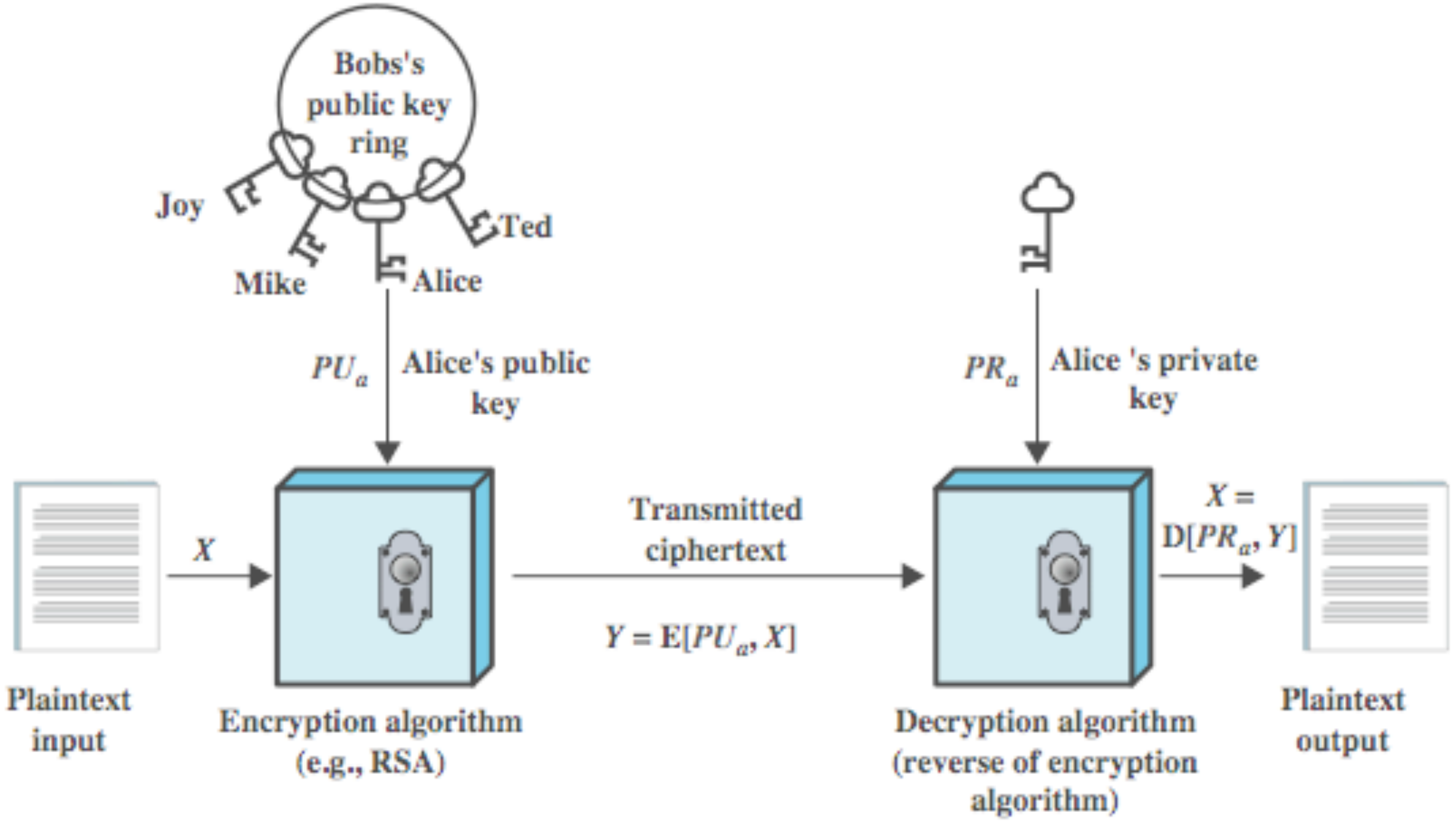
Fun with Prime Numbers

- Can you find the prime factors of 33?
  - (you should have learned in the fifth grade....)

- Can you find the prime factors of 678457899457285950934857985694487593569304817698746743959308567093457384957049739485703947503947563983857489670493248758945748978486788211283438038000979283764384685678576498769866453 1?

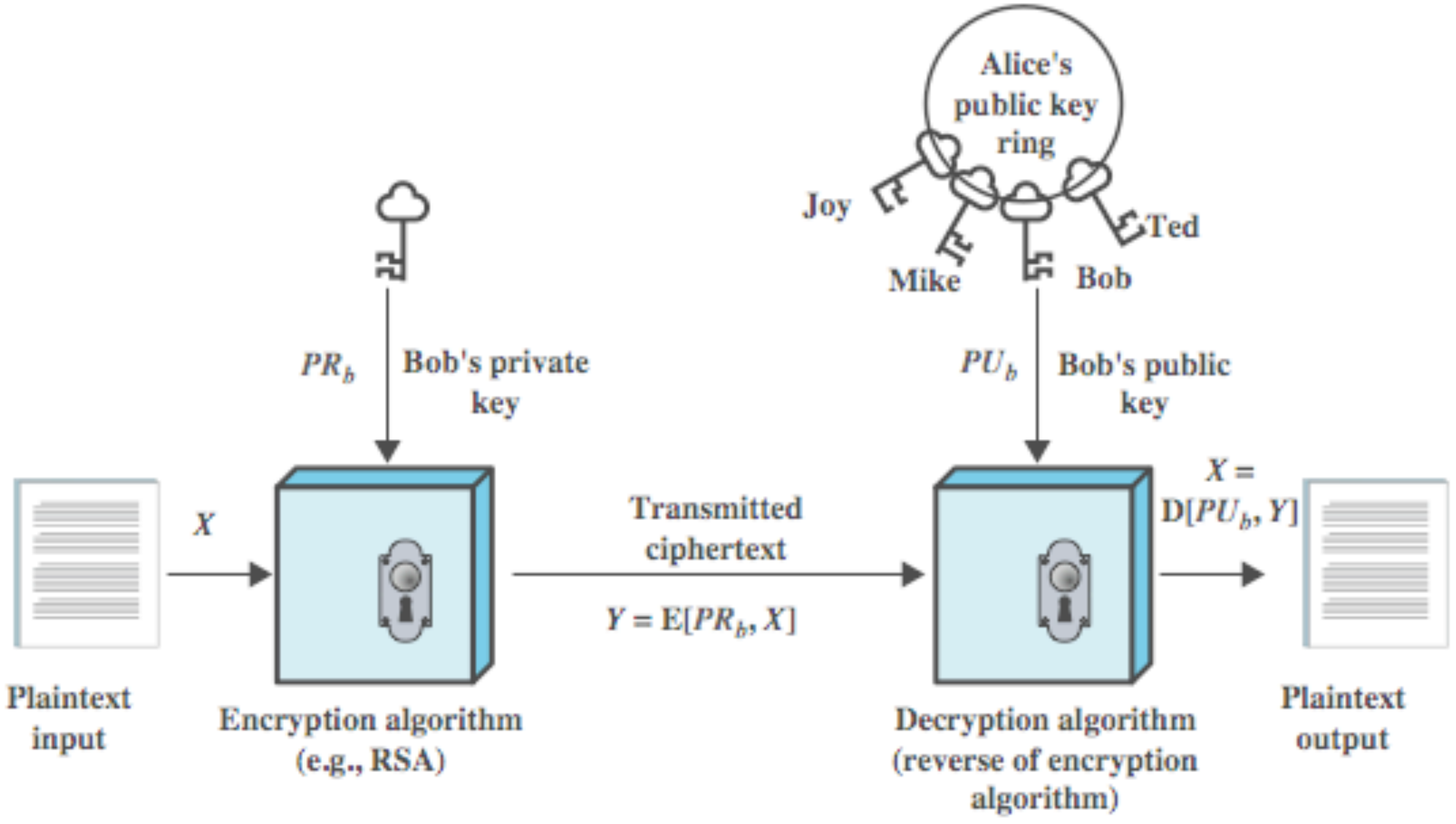
- A BIG PRIME NUMBER (e.g. 100 digits) times another BIG PRIME is *really really* BIG. And practically impossible to factor to get the prime numbers back again.

Public Key Encryption



(a) Confidentiality

Public Key Authentication



(b) Authentication

Requirements for Public-Key Cryptosystems

computationally easy to create key pairs

computationally easy for sender knowing public key to encrypt messages

computationally easy for receiver knowing private key to decrypt ciphertext

computationally infeasible for opponent to determine private key from public key

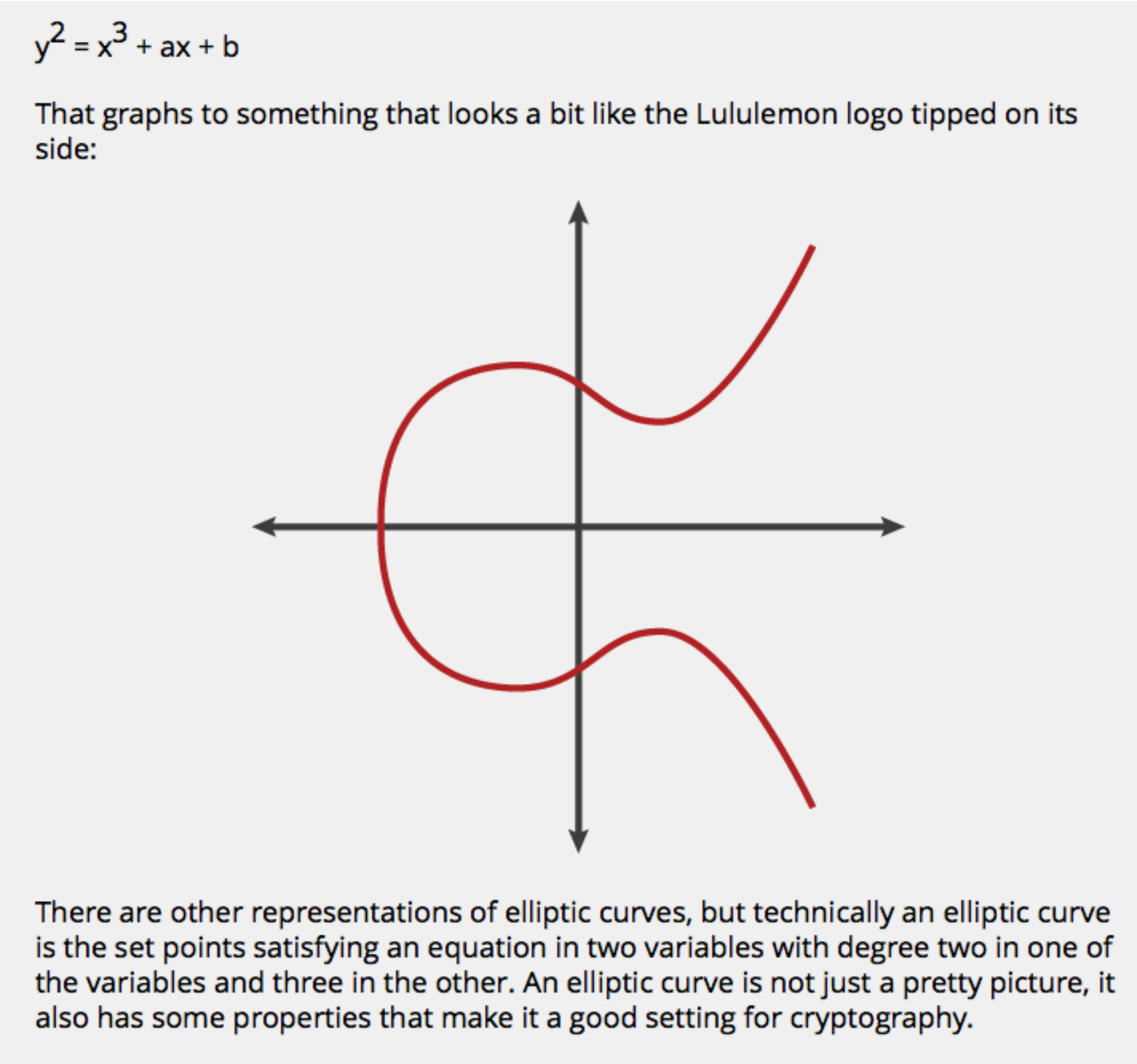
computationally infeasible for opponent to otherwise recover original message

useful if either key can be used for each role

Public Key Algorithms

- RSA (Rivest, Shamir, Adleman)
  - developed in 1977
  - only widely accepted public-key encryption algorithm
  - given tech advances need 1024+ bit keys. 2048 bits now common.
- Diffie-Hellman key exchange algorithm
  - only allows exchange of a secret key
- Digital Signature Standard (DSS)
  - provides only a digital signature function with SHA-1
- Elliptic curve cryptography (ECC)
  - new, security like RSA, but with much smaller keys

Elliptic Curve Cryptography



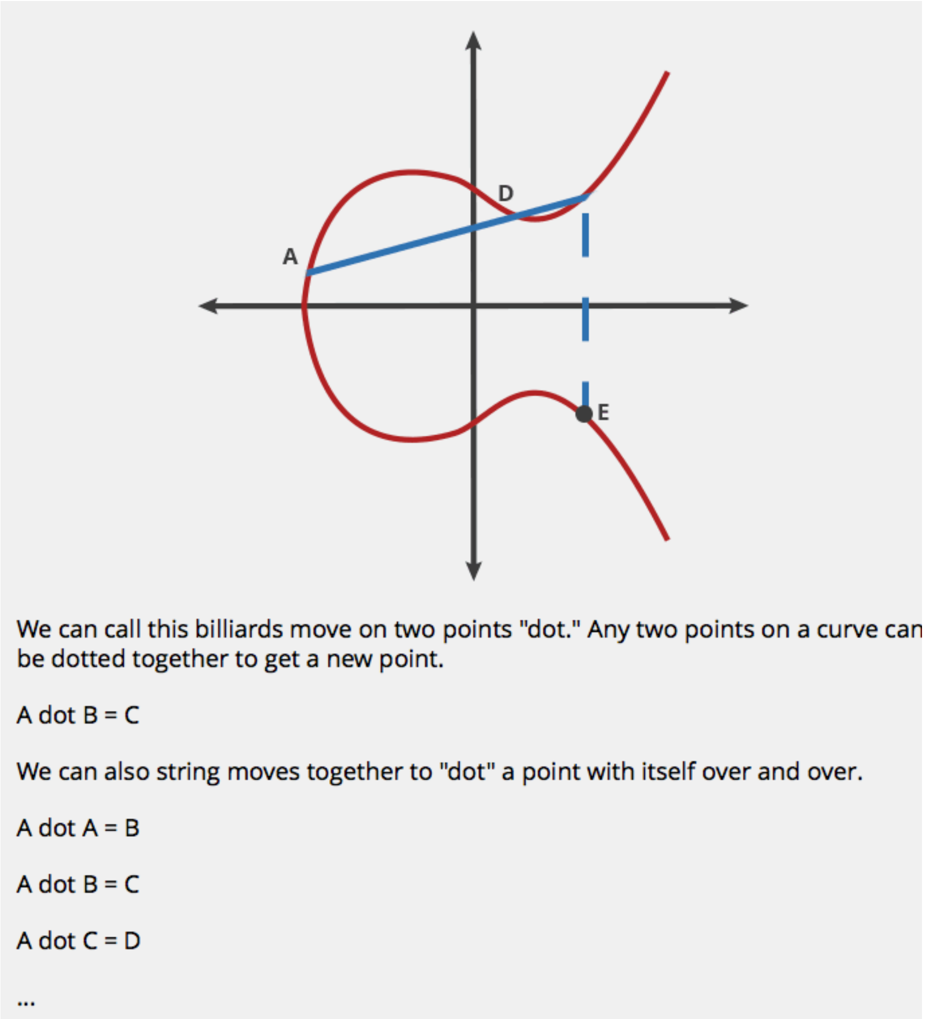
$y^2 = x^3 + ax + b$

That graphs to something that looks a bit like the Lululemon logo tipped on its side:

There are other representations of elliptic curves, but technically an elliptic curve is the set points satisfying an equation in two variables with degree two in one of the variables and three in the other. An elliptic curve is not just a pretty picture, it also has some properties that make it a good setting for cryptography.

Elliptic Curve Cryptography

- https://arstechnica.com/information-technology/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/2/



We can call this billiards move on two points "dot." Any two points on a curve can be dotted together to get a new point.

A dot B = C

We can also string moves together to "dot" a point with itself over and over.

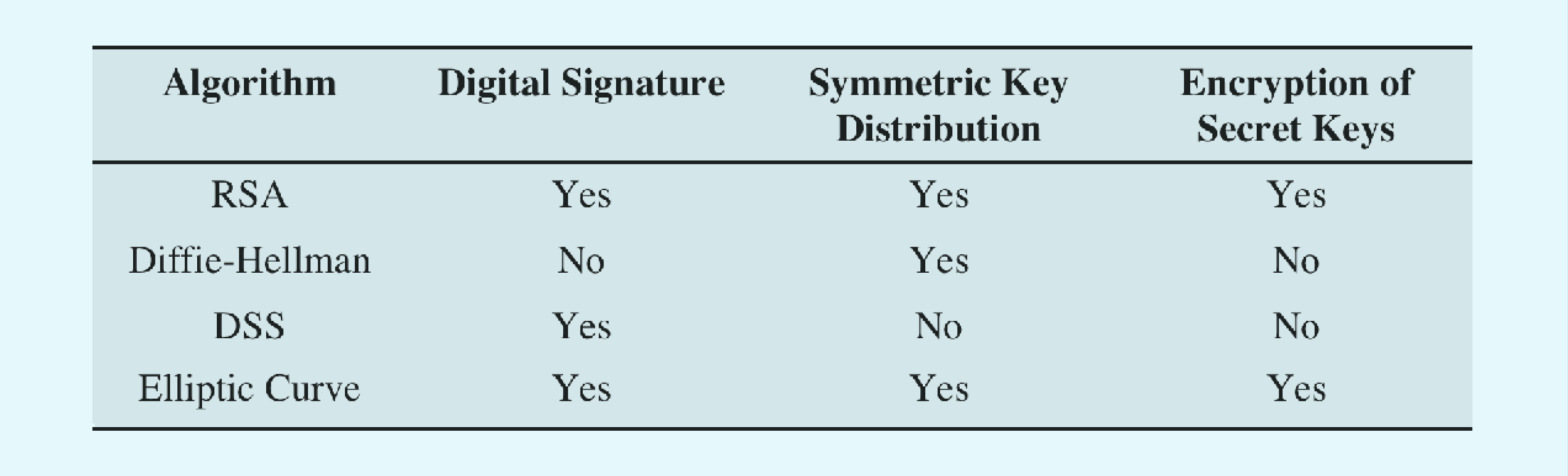
A dot A = B

A dot B = C

A dot C = D

...

Table 2.3



Algorithm	Digital Signature	Symmetric Key Distribution	Encryption of Secret Keys
RSA	Yes	Yes	Yes
Diffie-Hellman	No	Yes	No
DSS	Yes	No	No
Elliptic Curve	Yes	Yes	Yes

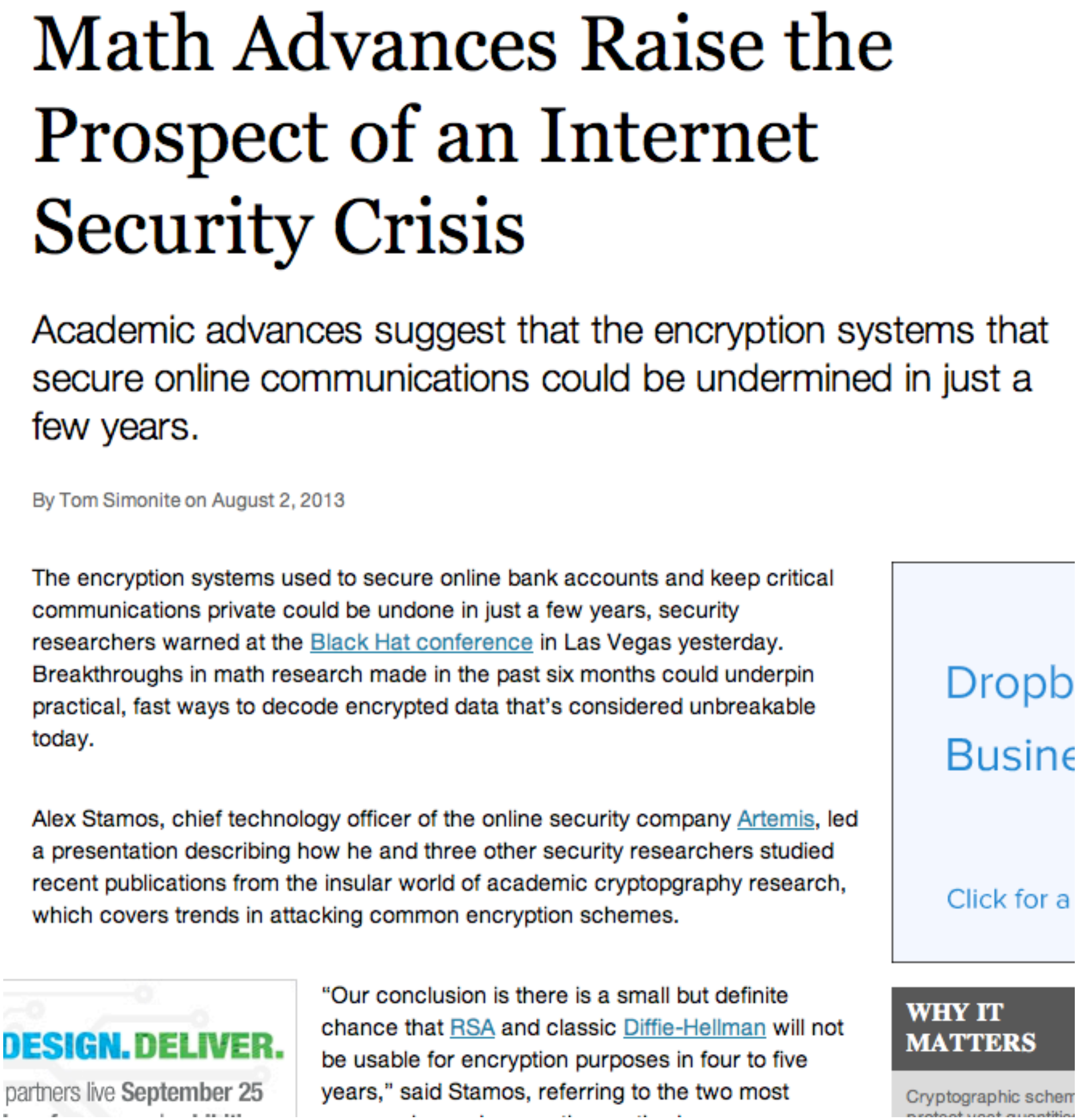
- **Applications for Public-Key Cryptosystems**

Crypto Applications

- common to encrypt transmitted data
  - “data in motion”
  - example TLS (HTTPS)
- Stored data
  - “data at rest”
    - which can be copied, backed up, recovered
  - approaches to encrypt stored data:
    - back-end appliance
    - library based tape encryption
    - background laptop/PC data encryption, disk

Cryptography is like magic fairy dust,
we just sprinkle it on our protocols
and its makes everything secure

Recent Headline in MIT Technology Review



Math Advances Raise the Prospect of an Internet Security Crisis

Academic advances suggest that the encryption systems that secure online communications could be undermined in just a few years.

By Tom Simonite on August 2, 2013

The encryption systems used to secure online bank accounts and keep critical communications private could be undone in just a few years, security researchers warned at the Black Hat conference in Las Vegas yesterday. Breakthroughs in math research made in the past six months could underpin practical, fast ways to decode encrypted data that's considered unbreakable today.

Alex Stamos, chief technology officer of the online security company Artemis, led a presentation describing how he and three other security researchers studied recent publications from the insular world of academic cryptopgraphy research, which covers trends in attacking common encryption schemes.

"Our conclusion is there is a small but definite chance that RSA and classic Diffie-Hellman will not be usable for encryption purposes in four to five years," said Stamos, referring to the two most

WHY IT MATTERS

Cryptographic schem

There is no magic fairy dust