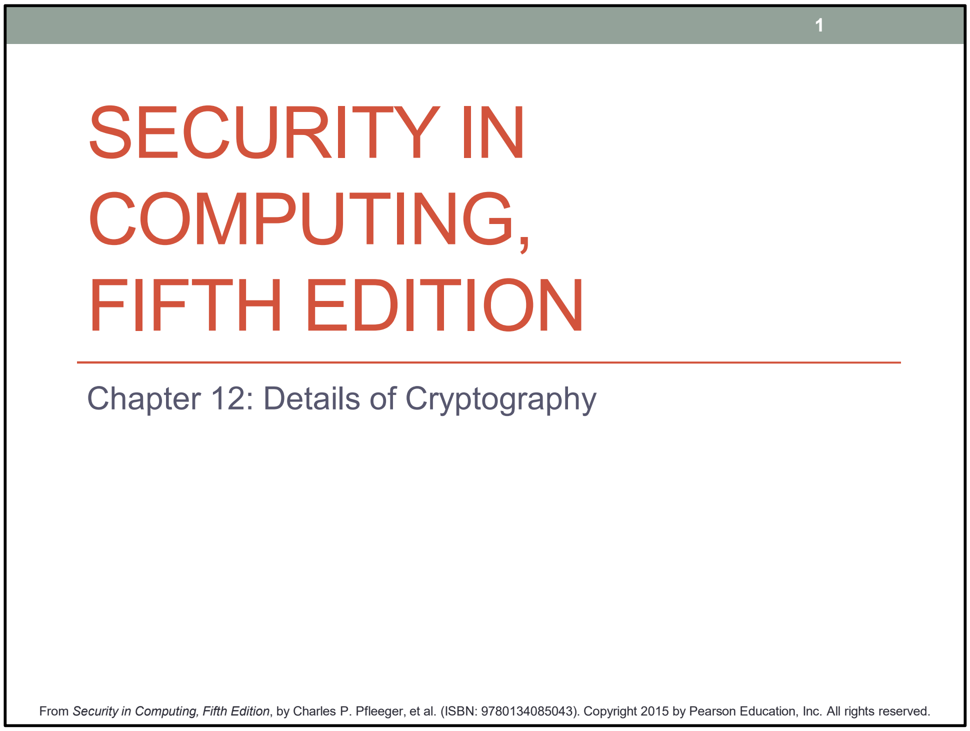
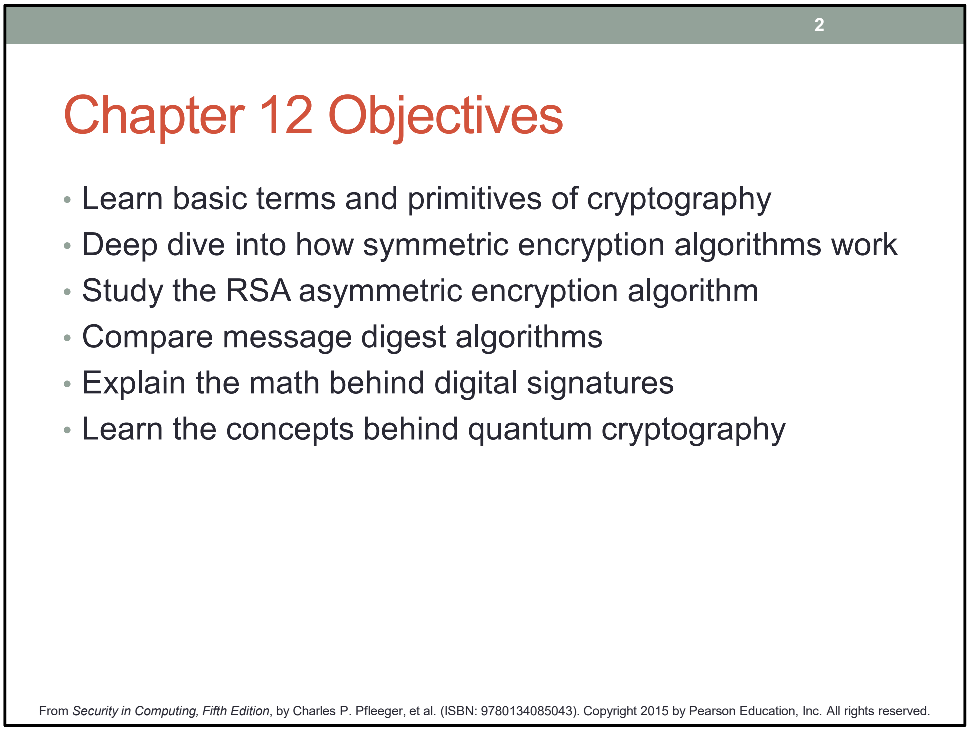


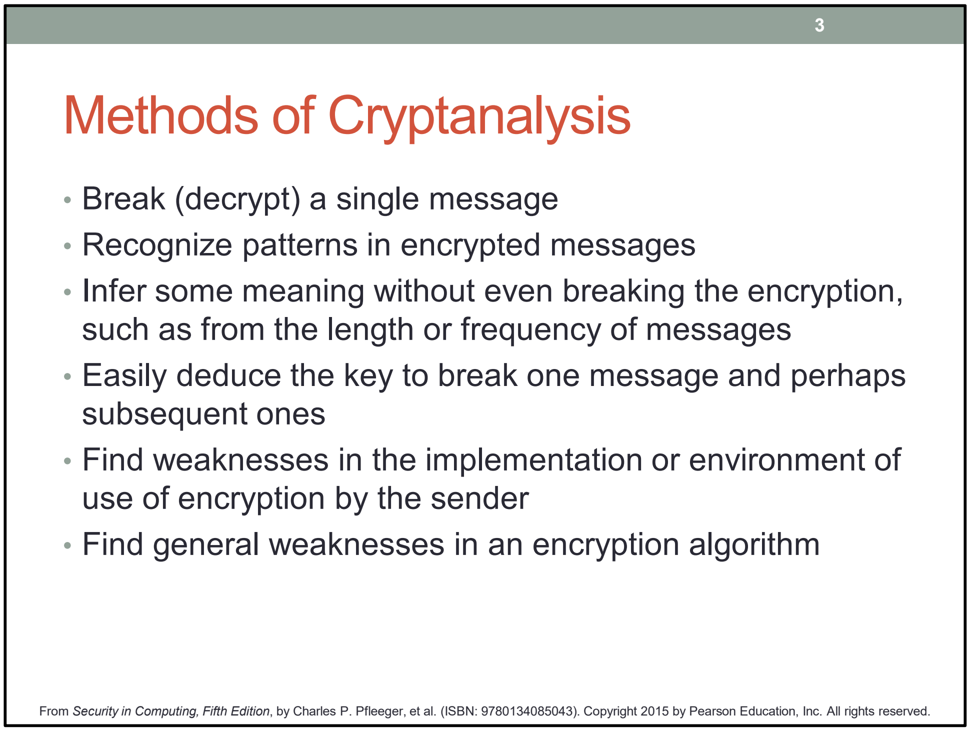
SECURITY IN COMPUTING, FIFTH EDITION

Chapter 12: Details of Cryptography



Chapter 12 Objectives

- Learn basic terms and primitives of cryptography
- Deep dive into how symmetric encryption algorithms work
- Study the RSA asymmetric encryption algorithm
- Compare message digest algorithms
- Explain the math behind digital signatures
- Learn the concepts behind quantum cryptography

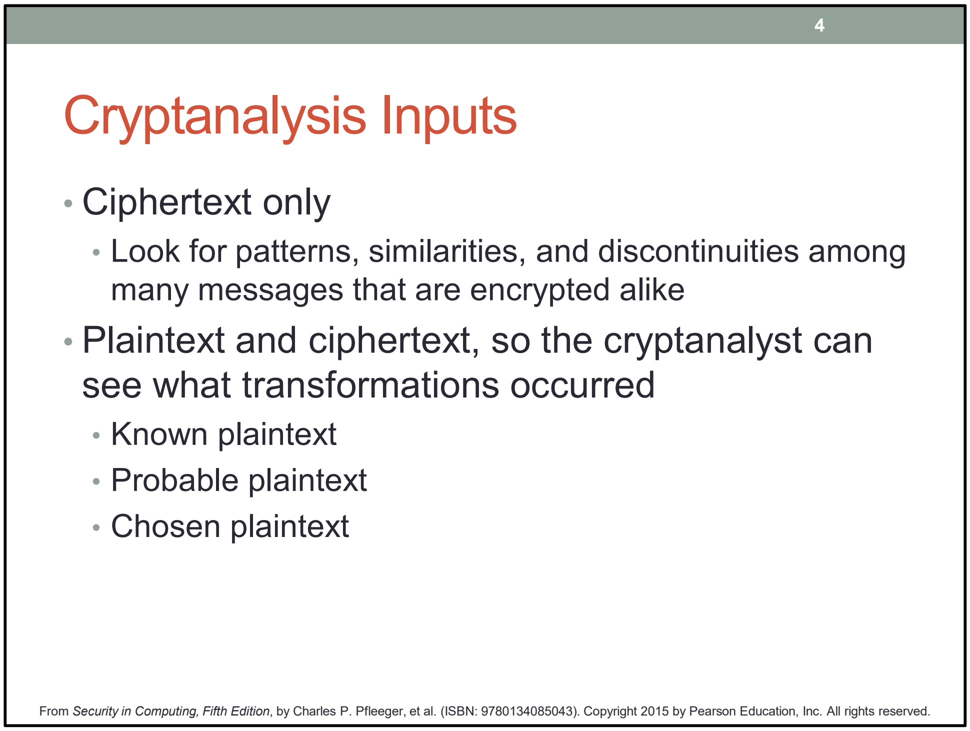


Methods of Cryptanalysis

- Break (decrypt) a single message
- Recognize patterns in encrypted messages
- Infer some meaning without even breaking the encryption, such as from the length or frequency of messages
- Easily deduce the key to break one message and perhaps subsequent ones
- Find weaknesses in the implementation or environment of use of encryption by the sender
- Find general weaknesses in an encryption algorithm

We start with a brief discussion of cryptanalysis because an understanding of what attackers are trying to accomplish (and how they are trying to accomplish it) informs the study of how to protect data from them. The methods listed here are not mutually exclusive, and which ones are applied will depend on a number of factors:

- Expertise of the attacker
- What information is available to the attacker
- What access is available to the attacker
- Other constraints, such as time

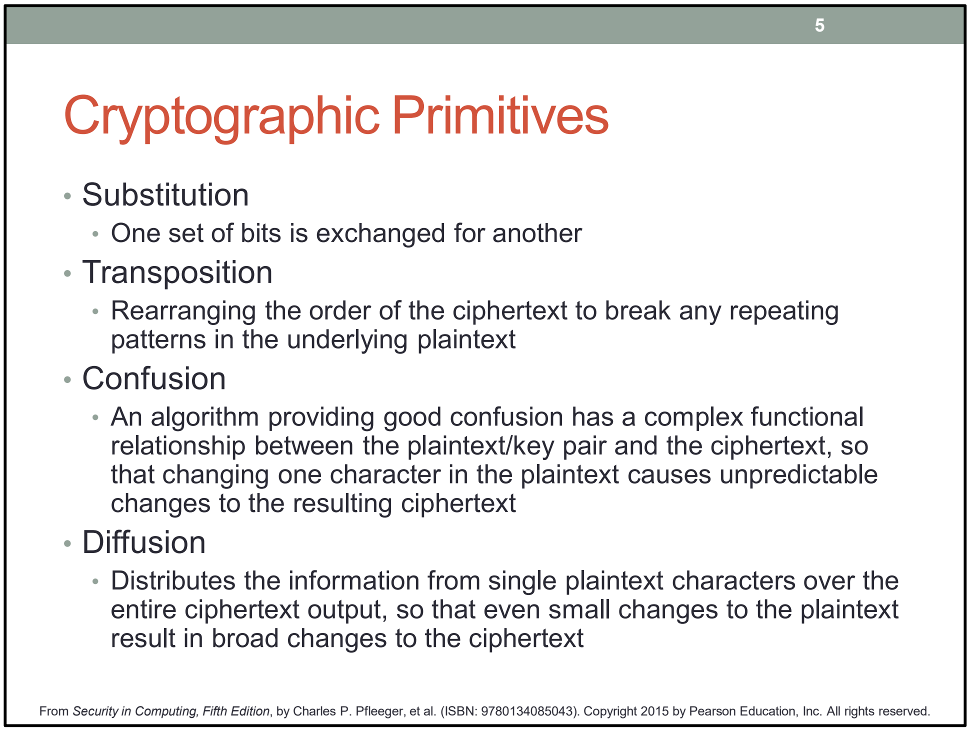


Cryptanalysis Inputs

- Ciphertext only
  - Look for patterns, similarities, and discontinuities among many messages that are encrypted alike
- Plaintext and ciphertext, so the cryptanalyst can see what transformations occurred
  - Known plaintext
  - Probable plaintext
  - Chosen plaintext

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

- Known plaintext—the analyst has an exact copy of the plaintext and ciphertext
- Probable plaintext—message is very likely to have certain content, such as a date header
- Chosen plaintext—the attacker gains sufficient access to the system to generate ciphertext from arbitrary plaintext inputs

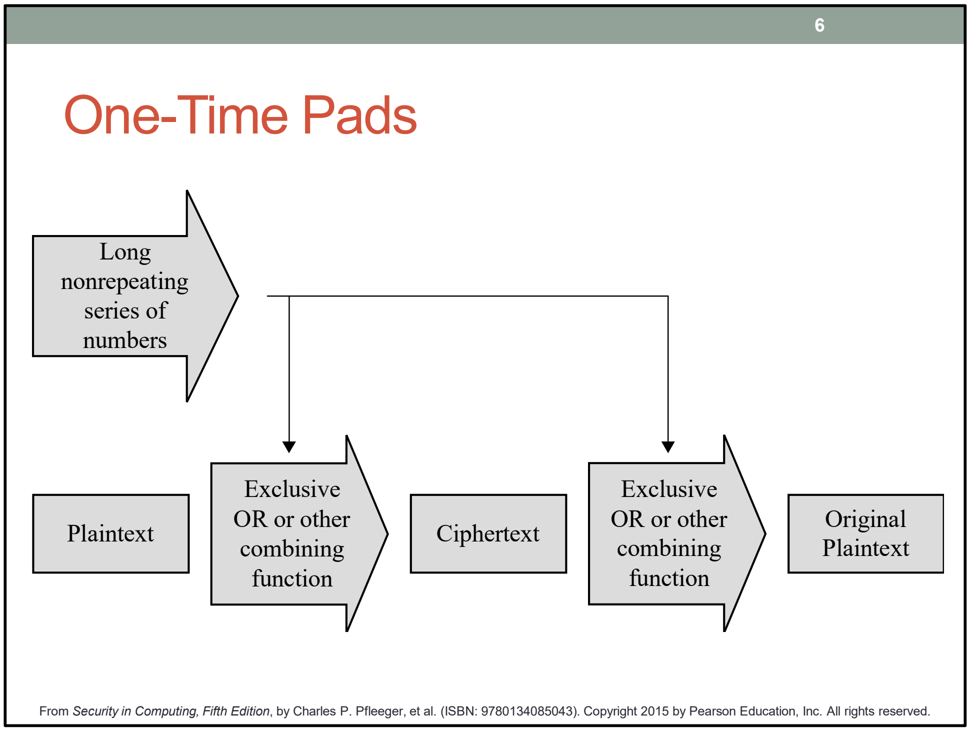


5

Cryptographic Primitives

- Substitution
  - One set of bits is exchanged for another
- Transposition
  - Rearranging the order of the ciphertext to break any repeating patterns in the underlying plaintext
- Confusion
  - An algorithm providing good confusion has a complex functional relationship between the plaintext/key pair and the ciphertext, so that changing one character in the plaintext causes unpredictable changes to the resulting ciphertext
- Diffusion
  - Distributes the information from single plaintext characters over the entire ciphertext output, so that even small changes to the plaintext result in broad changes to the ciphertext

These are the basic techniques that make up cryptographic algorithms. As we study some algorithms in depth later in the chapter, we'll see these again and again. The first two—substitution and transposition—are simple mathematical operations used within complex cryptosystems. The latter two—confusion and diffusion—are more conceptual and may be accomplished in a number of different ways depending on the cryptographic algorithm.



6

One-Time Pads

Long nonrepeating series of numbers

Plaintext

Exclusive OR or other combining function

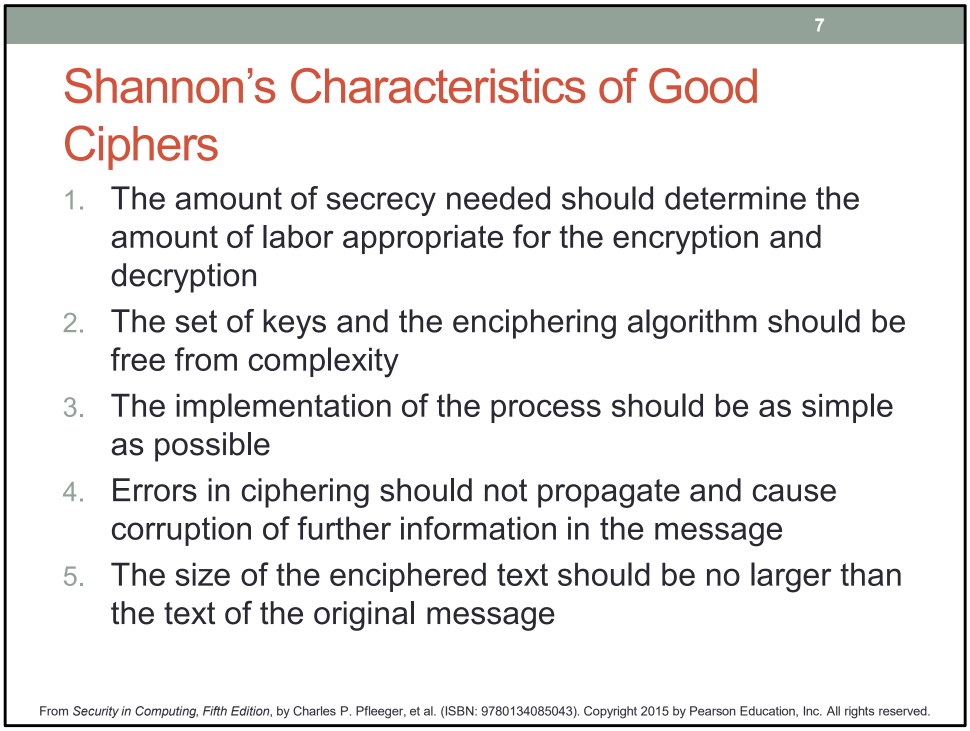
Ciphertext

Exclusive OR or other combining function

Original Plaintext

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

This is a diagram of the Vernam cipher, a type of one-time pad. A one-time pad is often used as an example of the perfect cipher, but it is only useful as a concept, as it is completely impractical. A one-time pad is a substitution cipher that uses an arbitrarily large, nonrepeating set of keys for substitution (in the diagram of the Vernam cipher, XOR is used instead of pure substitution), and requires both an unlimited set of completely random keys and absolute synchronization between sender and receiver, both of which are impractical. In terms of resistance to cryptanalysis, the one-time pad is the gold standard against which other encryption algorithms are measured, as it offers no patterns for attackers to analyze.

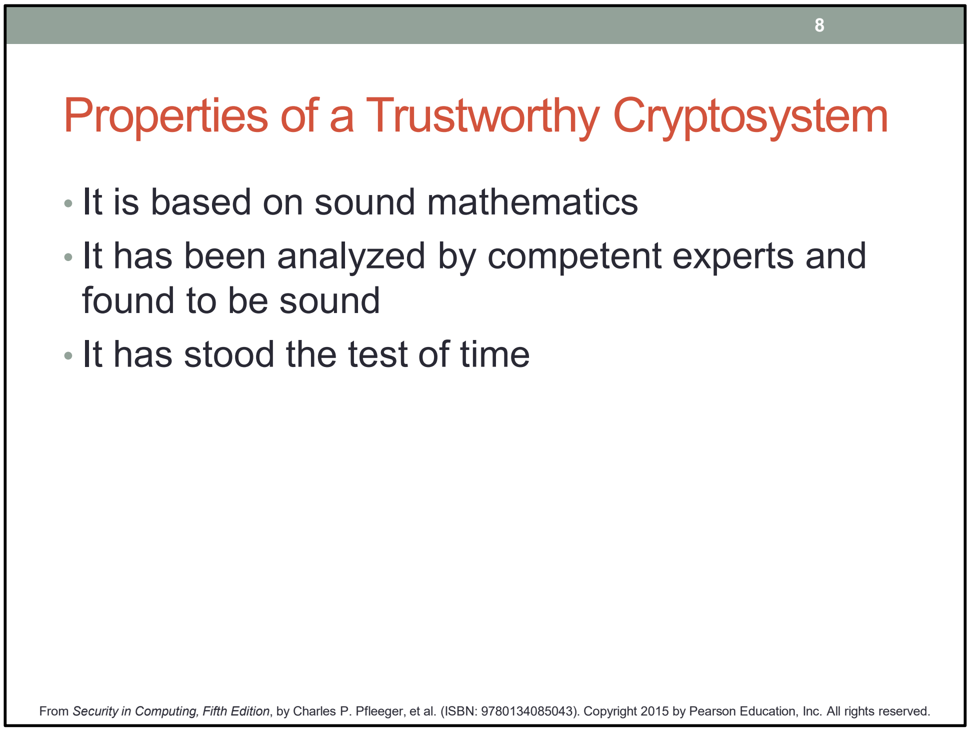


Shannon's Characteristics of Good Ciphers

1. The amount of secrecy needed should determine the amount of labor appropriate for the encryption and decryption
2. The set of keys and the enciphering algorithm should be free from complexity
3. The implementation of the process should be as simple as possible
4. Errors in ciphering should not propagate and cause corruption of further information in the message
5. The size of the enciphered text should be no larger than the text of the original message

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

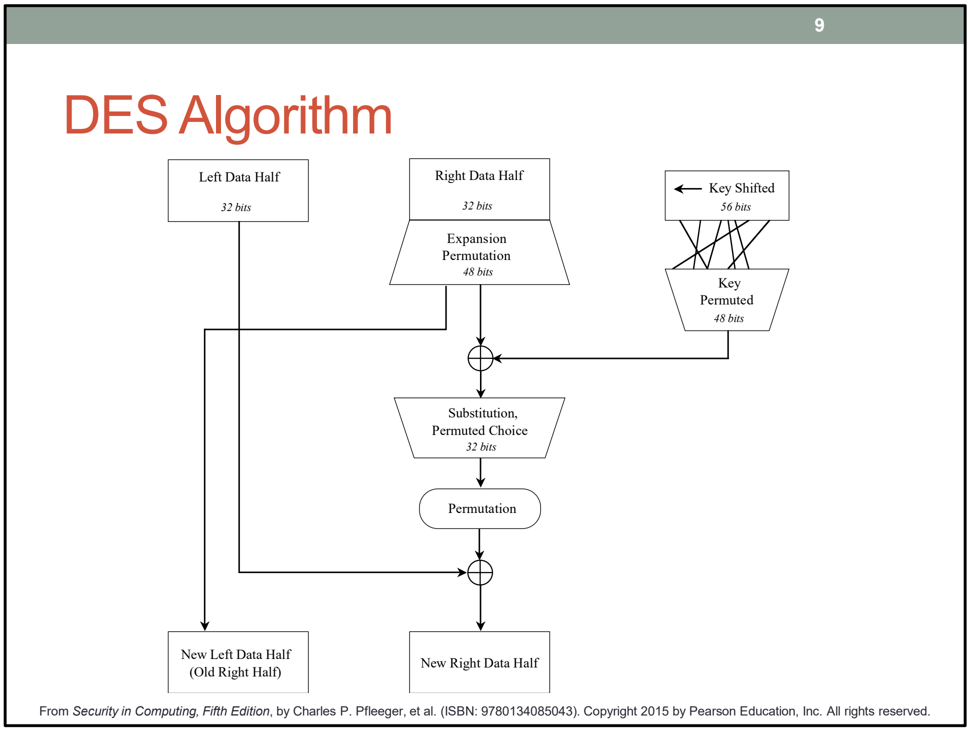
1. The degree of secrecy required factors into questions such as key length and number of rounds and should be based on implementation of the algorithm, current and predicted speeds of computers, and resources of likely attackers.
2. The process has to work on any kind of plaintext input, and keys should be easy for users to generate, transmit, and store.
3. As we saw earlier in the book, complexity is the enemy of good security analysis. It is easier to identify flaws in, and to correctly implement, a simpler algorithm, and a simpler algorithm is therefore more likely to be free of flaws.
4. Communication errors do happen, and when they do, the need for retransmission should be as limited as possible.
5. A ciphertext that expands dramatically in size cannot possibly carry more information than the source plaintext, yet it gives the cryptanalyst more data from which to infer a pattern. Larger messages also require more transmission time and storage and are therefore less practical for users.



Properties of a Trustworthy Cryptosystem

- It is based on sound mathematics
- It has been analyzed by competent experts and found to be sound
- It has stood the test of time

Good cryptographic algorithms are derived from sound principles and have security properties that are proven by expert mathematicians. Historically, algorithms that have not met this standard have been easily broken. Because cryptographic algorithms are complex, it can take years of analysis before serious flaws are identified.



DES Algorithm

Left Data Half
32 bits

Right Data Half
32 bits

Key Shifted
56 bits

Expansion Permutation
48 bits

Key Permuted
48 bits

Substitution, Permuted Choice
32 bits

Permutation

New Left Data Half (Old Right Half)

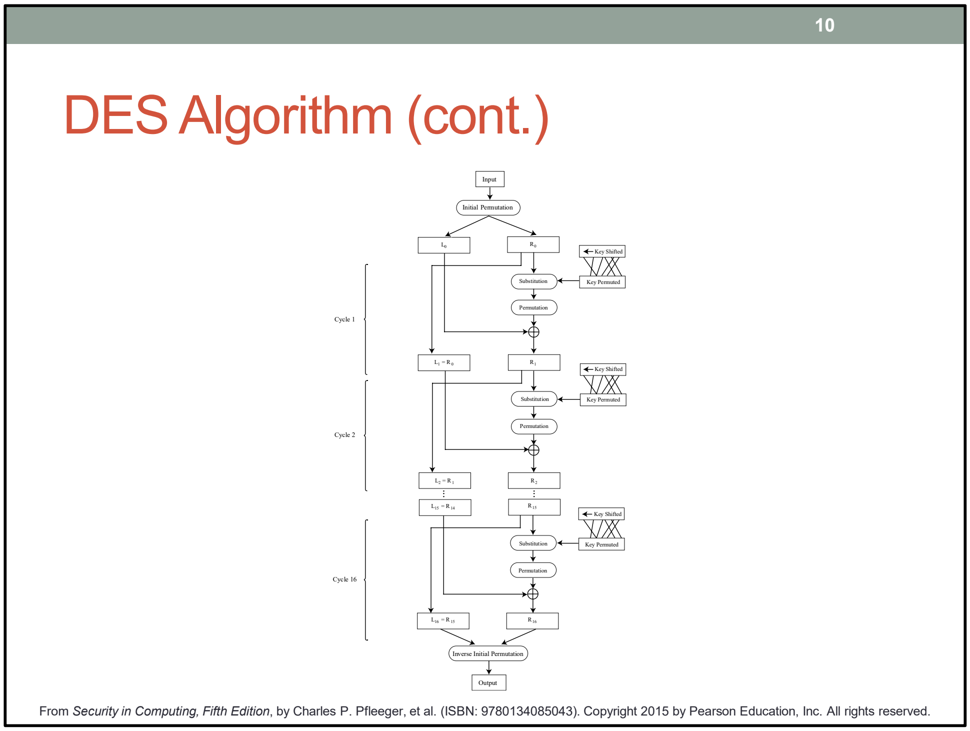
New Right Data Half

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

We first explained symmetric cryptography and introduced DES in Chapter 2. Here we look in detail at how DES works. While DES is no longer practical for use against modern technology, the algorithm has a combination of strong fundamentals and relative simplicity that makes it useful for teaching how symmetric encryption actually works. This diagram shows a single DES cycle. Once this cycle is explained, we'll look at a more complete representation of the DES process.

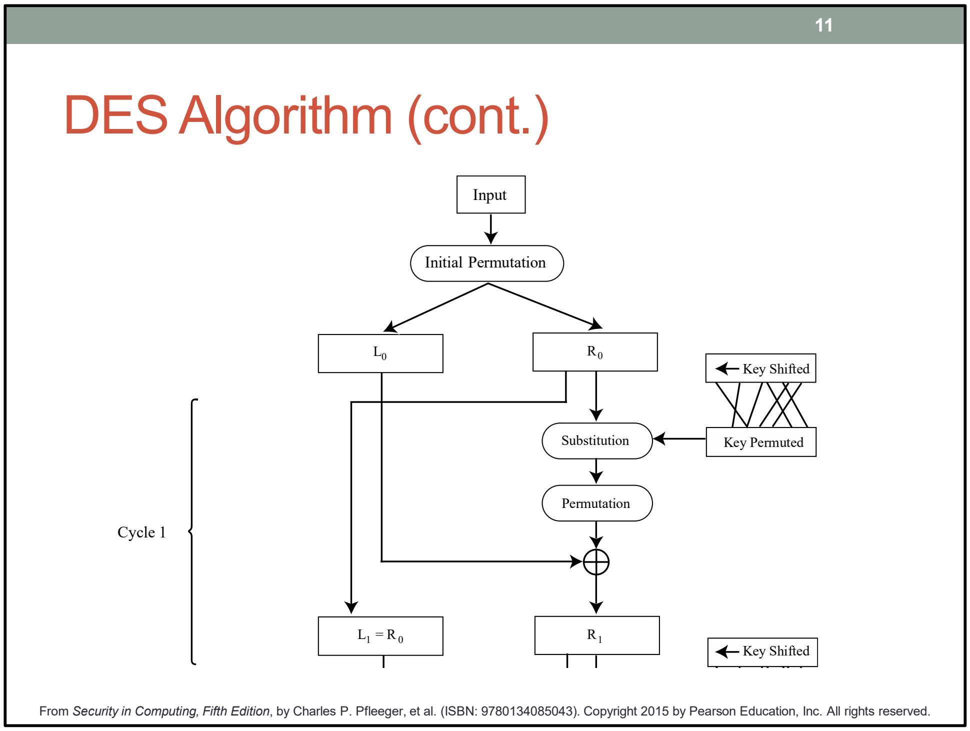
- Input to DES is divided into blocks of 64 bits (not shown)
- The data bits are permuted by an "initial permutation" (not shown)
- The key is reduced from 64 bits to 56 bits (parity bits are removed) (not shown)
- The 64 permuted data bits are broken into a left half and right half
- The 32-bit right half is expanded to 58 bits by repeating certain bits
- The key is reduced to 48 bits by choosing only certain bits according to tables called S-boxes (S-boxes are not shown for simplicity)
- The key is shifted left by a number of bits and also permuted
- The key is combined with the right half, which is then combined with the left half
- The result of these combinations becomes the new right half, while the old right half becomes the new left half

DES Algorithm (cont.)



From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

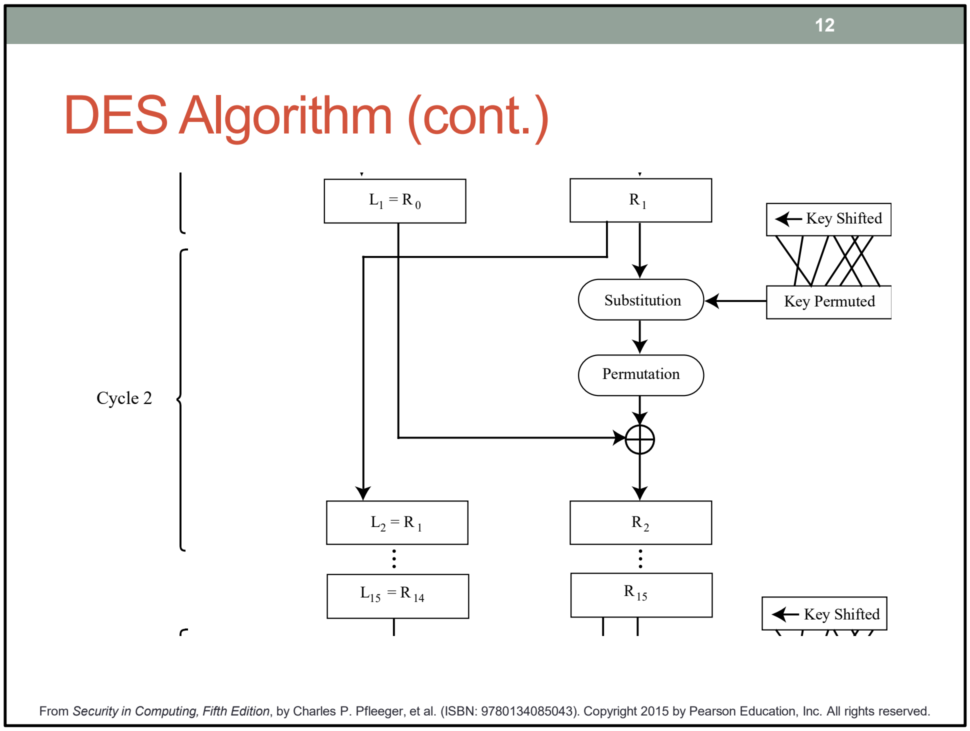
This is a much more complete view of the DES algorithm, showing three cycles along with the initial permutation and the final permutation. We'll zoom in on the following slides.



DES Algorithm (cont.)

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

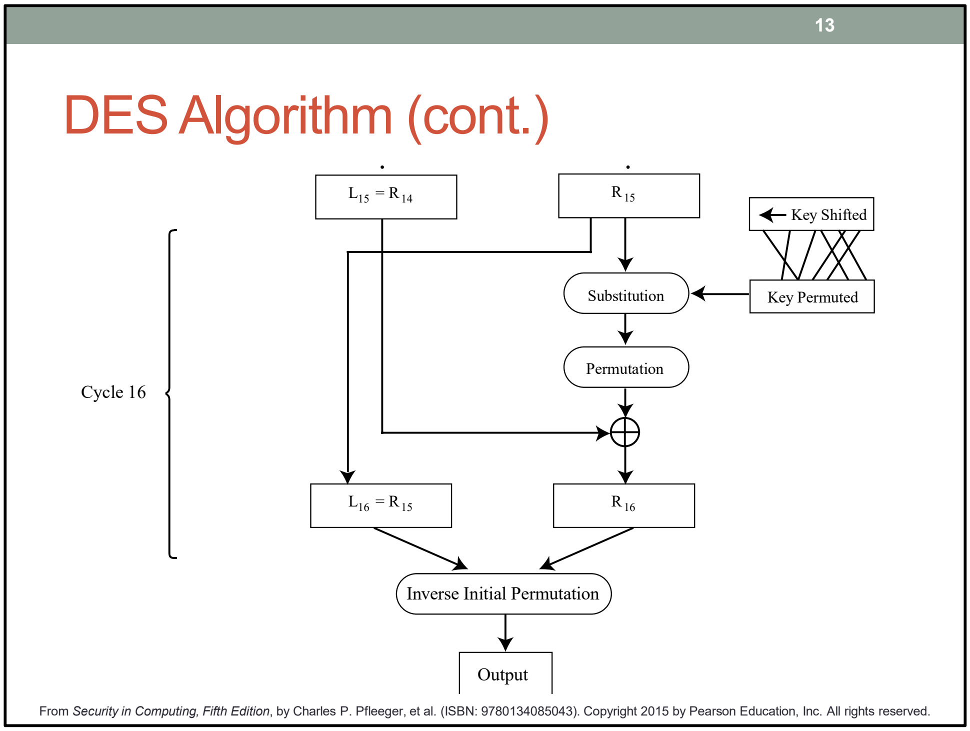
Here we see where the initial permutation fits in before the first cycle.



DES Algorithm (cont.)

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Here we see how each cycle connects to the last.



DES Algorithm (cont.)

$L_{15} = R_{14}$

R_{15}

Key Shifted

Substitution

Key Permuted

Permutation

Cycle 16

$L_{16} = R_{15}$

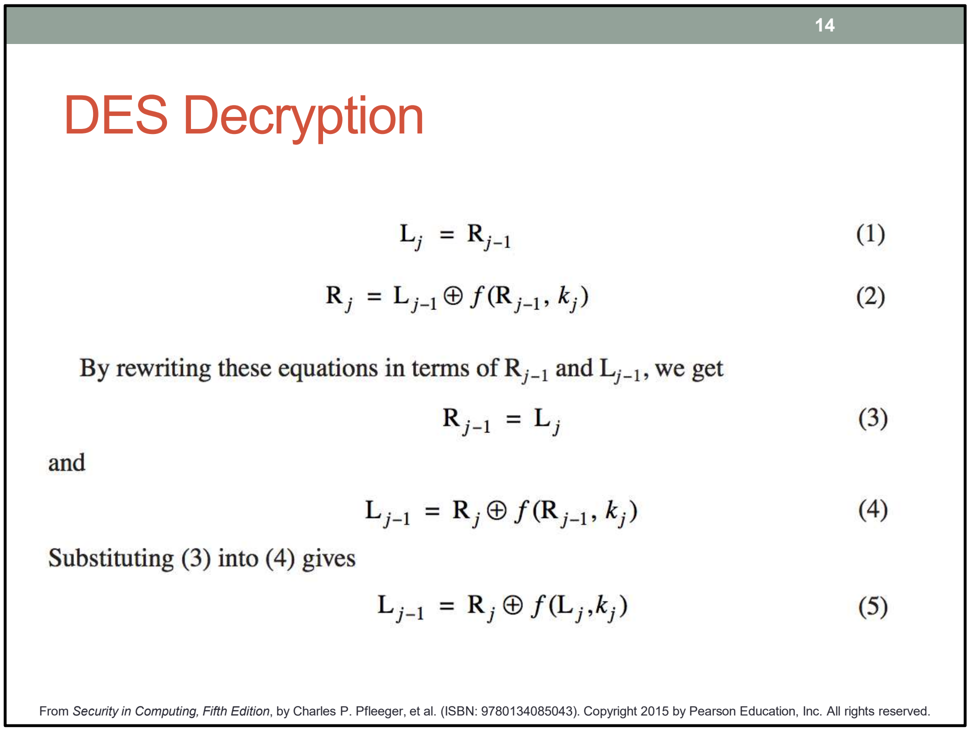
R_{16}

Inverse Initial Permutation

Output

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

Here we see that the final permutation is an inverse of the initial permutation, performed against the outputs of the final cycle.



DES Decryption

$$L_j = R_{j-1} \qquad (1)$$

$$R_j = L_{j-1} \oplus f(R_{j-1}, k_j) \qquad (2)$$

By rewriting these equations in terms of R_{j-1} and L_{j-1}, we get

$$R_{j-1} = L_j \qquad (3)$$

and

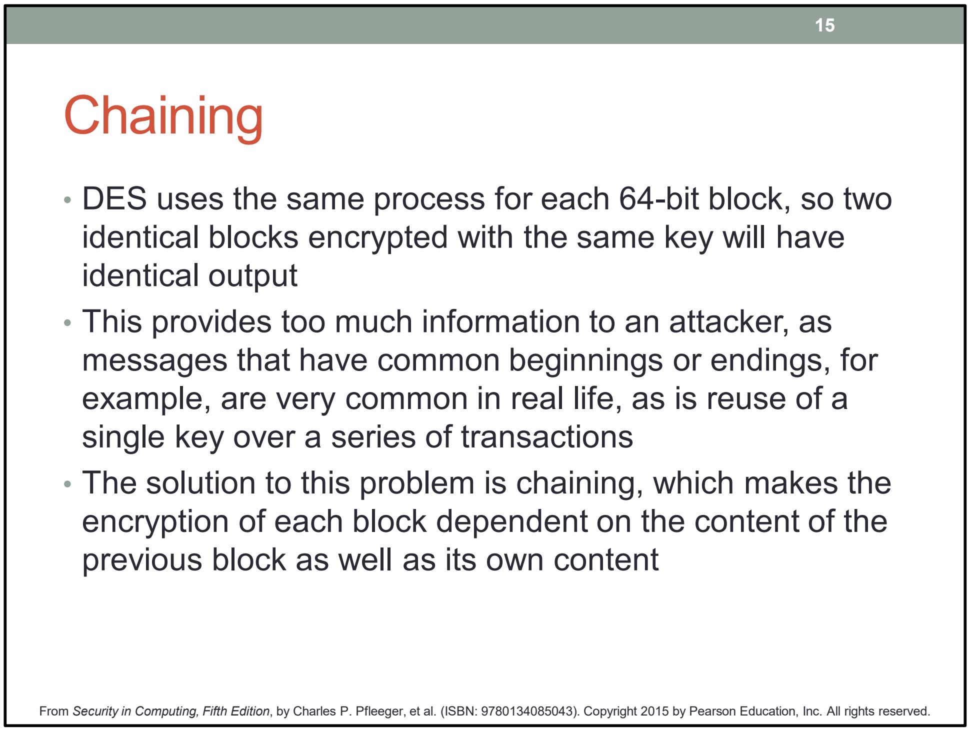
$$L_{j-1} = R_j \oplus f(R_{j-1}, k_j) \qquad (4)$$

Substituting (3) into (4) gives

$$L_{j-1} = R_j \oplus f(L_j, k_j) \qquad (5)$$

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

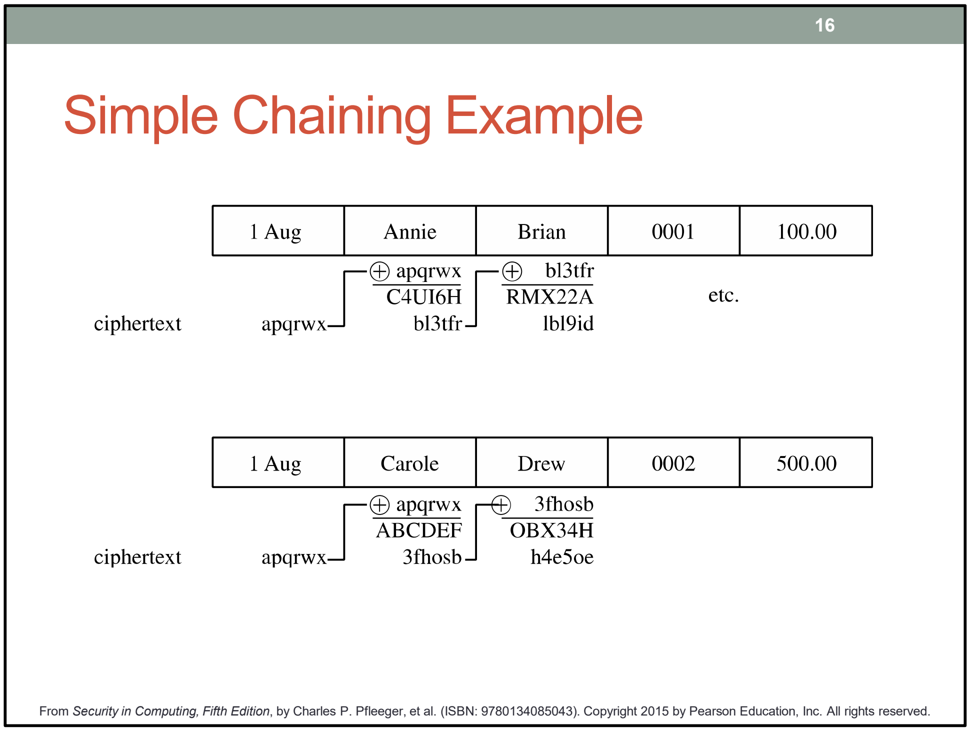
In DES, a single algorithm is used for both encryption and decryption. In these equations describing a single DES cycle, L is the left-half input, R is the right-half input, *j* is the current cycle, *k* is the key for the current cycle, and *f* is the function computed in an expand-shift-substitute-permute cycle. Equations (3) and (5) show that R and L for the previous cycle can be derived entirely from R and L of the current cycle, demonstrating that the DES algorithm can work in reverse.



Chaining

- DES uses the same process for each 64-bit block, so two identical blocks encrypted with the same key will have identical output
- This provides too much information to an attacker, as messages that have common beginnings or endings, for example, are very common in real life, as is reuse of a single key over a series of transactions
- The solution to this problem is chaining, which makes the encryption of each block dependent on the content of the previous block as well as its own content

While we write about chaining in the context of DES, this is a general problem for which chaining is a general solution.



Simple Chaining Example

1 Aug	Annie	Brian	0001	100.00

⊕ apqrwx
C4UI6H
ciphertext apqrwx bl3tfr

⊕ bl3tfr
RMX22A
lbl9id

etc.

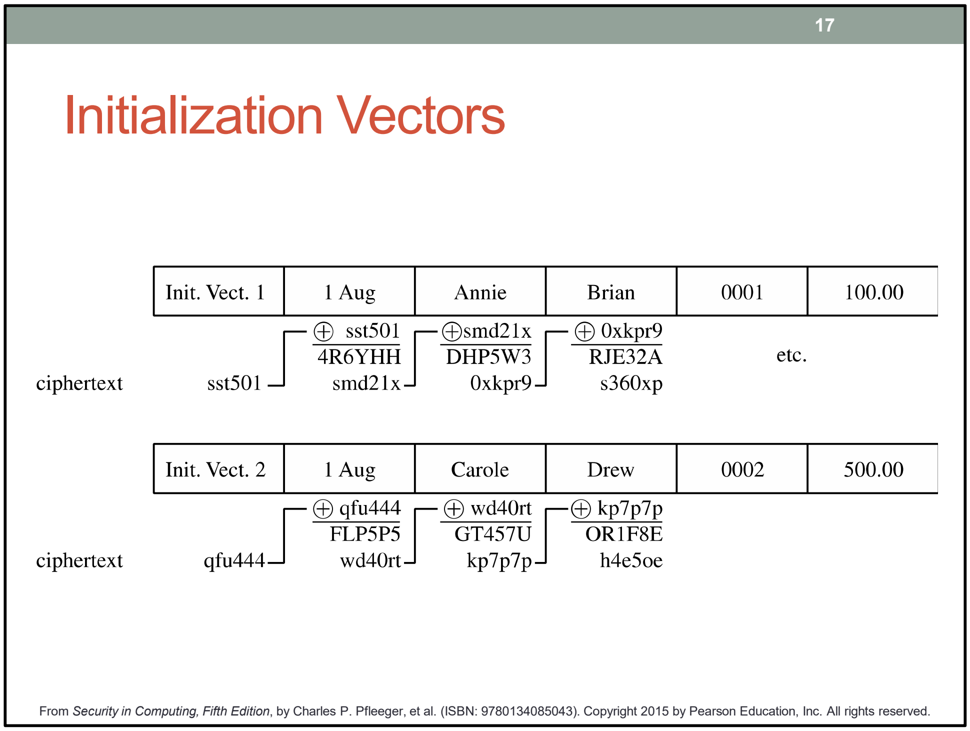
1 Aug	Carole	Drew	0002	500.00

⊕ apqrwx
ABCDEF
ciphertext apqrwx 3fhosb

⊕ 3fhosb
OBX34H
h4e5oe

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

In this simple chaining example, the input of the second block is an XOR of the output of the first block and the plaintext of the second block. This has the effect of making identical plaintext in two different messages produce completely different ciphertext. But what about the first block? We’ll look at that on the next slide.



17

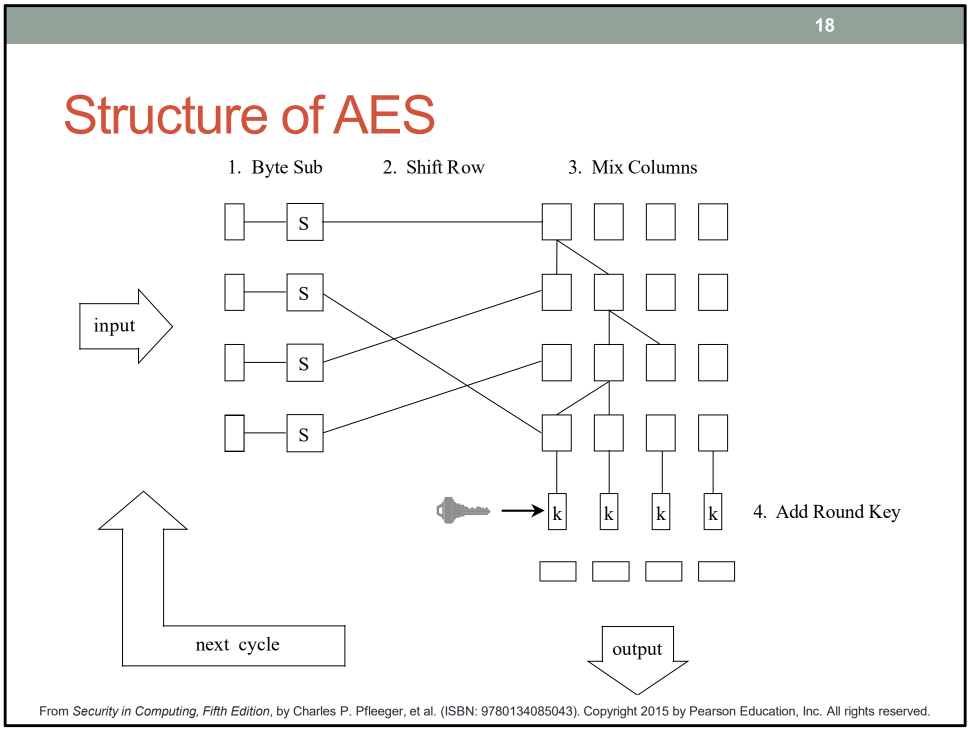
Initialization Vectors

	Init. Vect. 1	1 Aug	Annie	Brian	0001	100.00
		⊕ sst501	⊕smd21x	⊕ 0xkpr9		
		4R6YHH	DHP5W3	RJE32A	etc.	
ciphertext	sst501	smd21x	0xkpr9	s360xp		

	Init. Vect. 2	1 Aug	Carole	Drew	0002	500.00
		⊕ qfu444	⊕ wd40rt	⊕ kp7p7p		
		FLP5P5	GT457U	OR1F8E		
ciphertext	qfu444	wd40rt	kp7p7p	h4e5oe		

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

To protect against the problem of identical first blocks, we start with an initialization vector—an unpredictable (usually random) value that changes for each message, so that the positive effect of chaining can be useful even for the first block of data.



18

Structure of AES

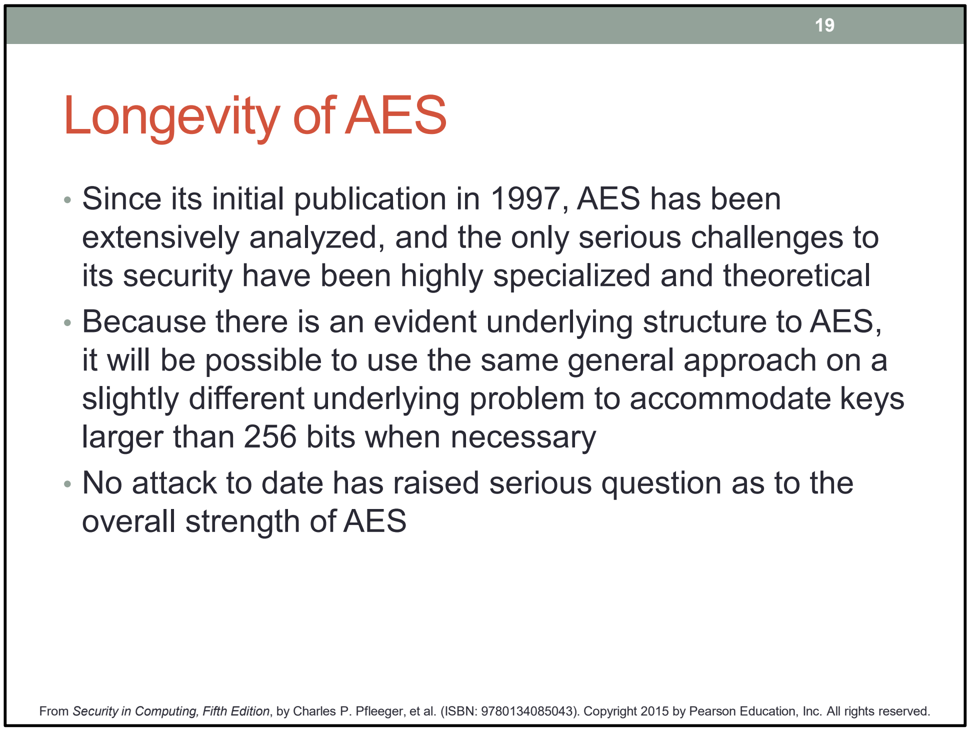
From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

AES is much more complex than DES, and we will not explain it in detail here. The goal of this chapter is to provide a high-level understanding of how and why these algorithms work. For a detailed understanding, a full course dedicated to cryptography is appropriate.

The algorithm consists of 10, 12, or 14 cycles, for a 128-, 192-, or 256-bit key, respectively. Each cycle consists of four steps:

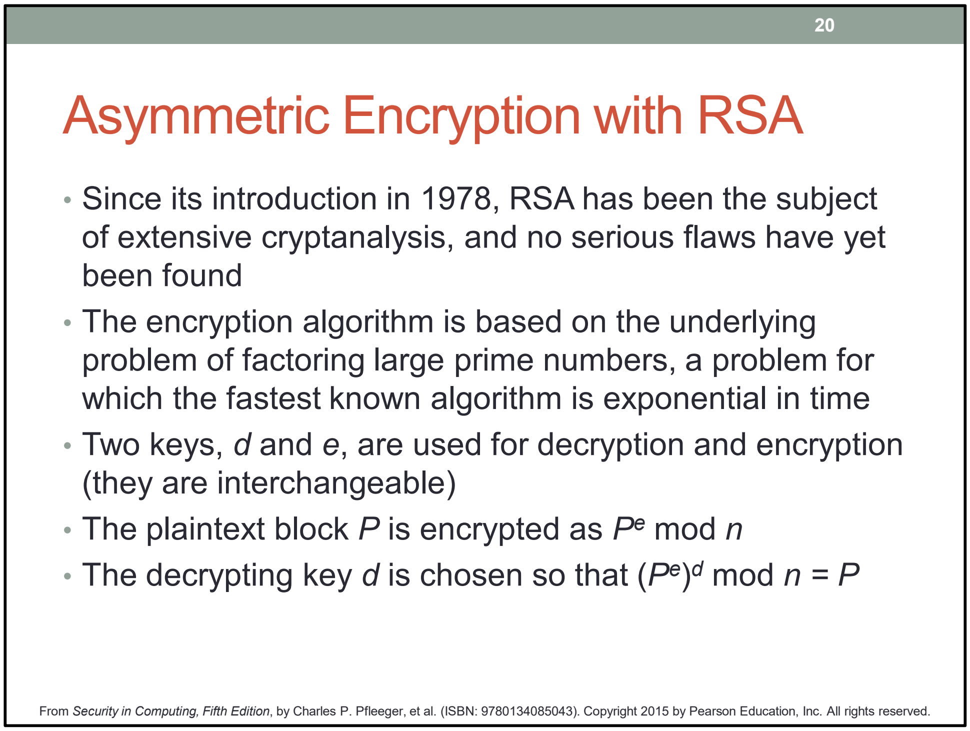
1. *Byte substitution.* This step substitutes each byte of a 128-bit block according to a substitution table. This is a straight diffusion operation.
2. *Shift row.* Certain bits are shifted to other positions. This is a straight confusion operation.
3. *Mix column.* This step involves shifting left and XORing bits with themselves. These operations deliver both confusion and diffusion.
4. *Add subkey.* Here, a portion of the key unique to this cycle is XORed with the cycle result. This operation delivers confusion and incorporates the key.

Each cycle performs both confusion and diffusion as well as blends the key into the result.



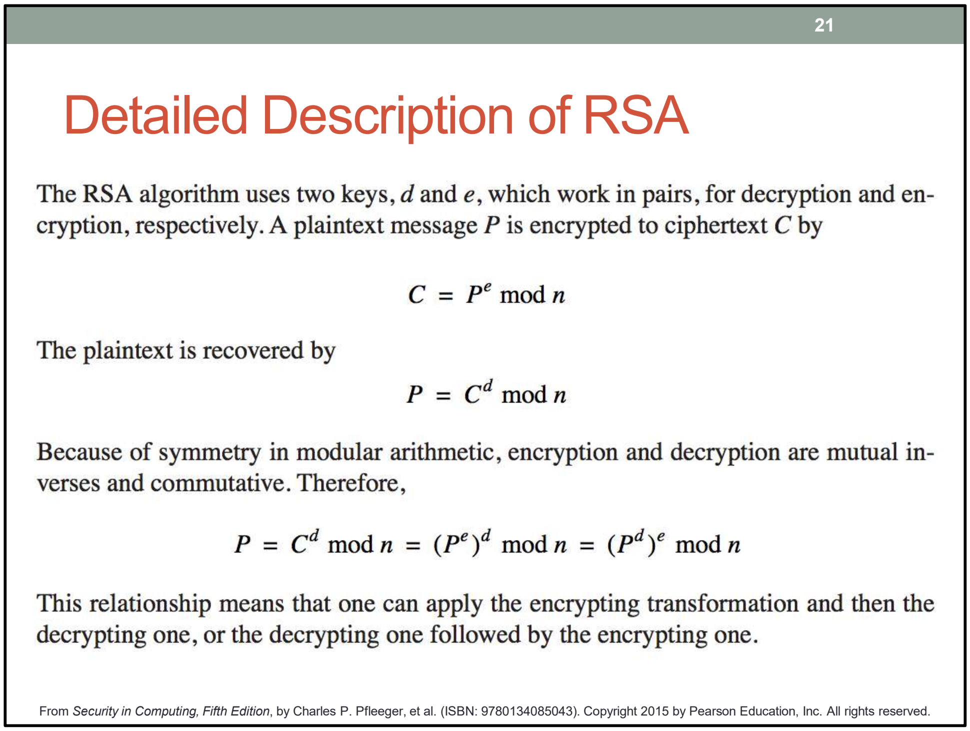
Longevity of AES

- Since its initial publication in 1997, AES has been extensively analyzed, and the only serious challenges to its security have been highly specialized and theoretical
- Because there is an evident underlying structure to AES, it will be possible to use the same general approach on a slightly different underlying problem to accommodate keys larger than 256 bits when necessary
- No attack to date has raised serious question as to the overall strength of AES



Asymmetric Encryption with RSA

- Since its introduction in 1978, RSA has been the subject of extensive cryptanalysis, and no serious flaws have yet been found
- The encryption algorithm is based on the underlying problem of factoring large prime numbers, a problem for which the fastest known algorithm is exponential in time
- Two keys, *d* and *e*, are used for decryption and encryption (they are interchangeable)
- The plaintext block *P* is encrypted as P^e mod *n*
- The decrypting key *d* is chosen so that $(P^e)^d$ mod $n = P$



Detailed Description of RSA

The RSA algorithm uses two keys, d and e, which work in pairs, for decryption and encryption, respectively. A plaintext message P is encrypted to ciphertext C by

$$C = P^e \bmod n$$

The plaintext is recovered by

$$P = C^d \bmod n$$

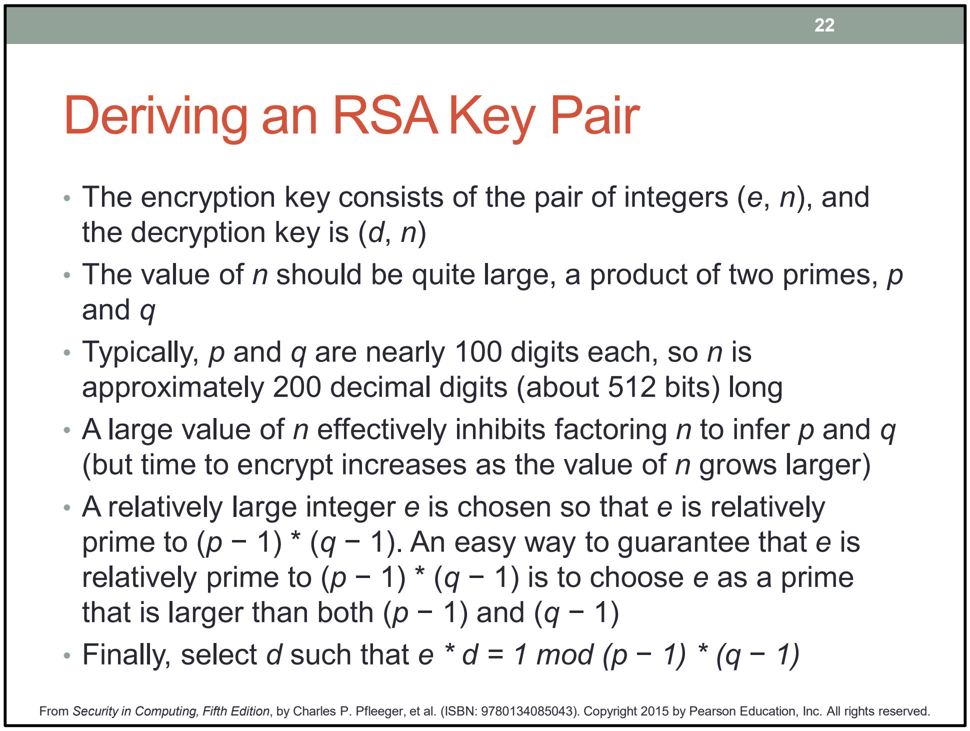
Because of symmetry in modular arithmetic, encryption and decryption are mutual inverses and commutative. Therefore,

$$P = C^d \bmod n = (P^e)^d \bmod n = (P^d)^e \bmod n$$

This relationship means that one can apply the encrypting transformation and then the decrypting one, or the decrypting one followed by the encrypting one.

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

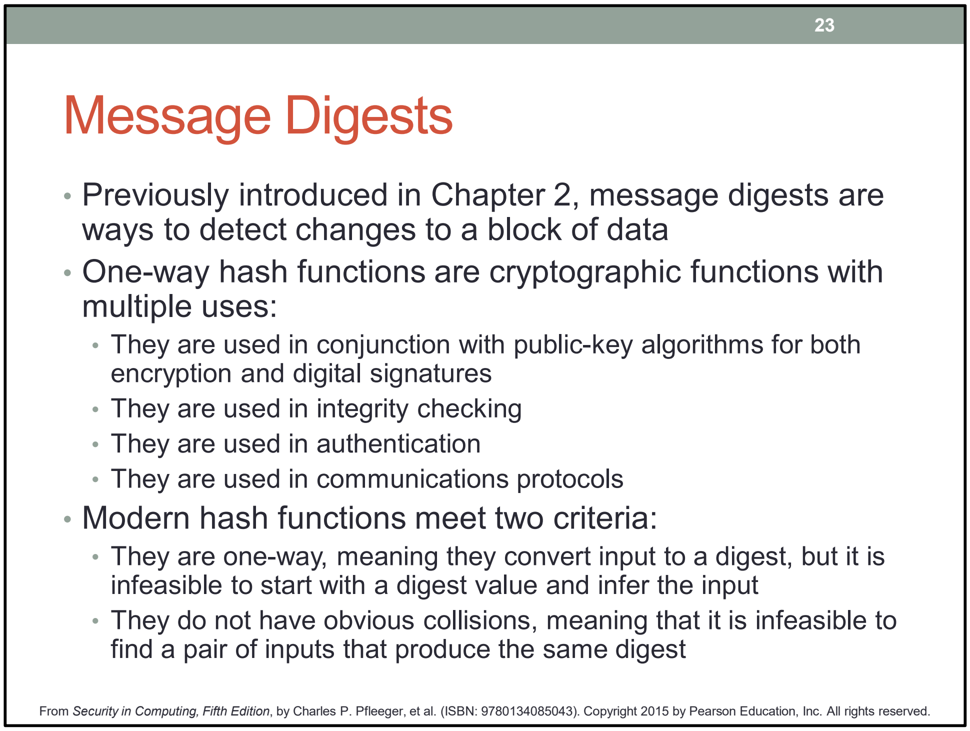
Because of the hard problem on which it is based, the RSA algorithm is elegant in its simplicity and analyzability.



Deriving an RSA Key Pair

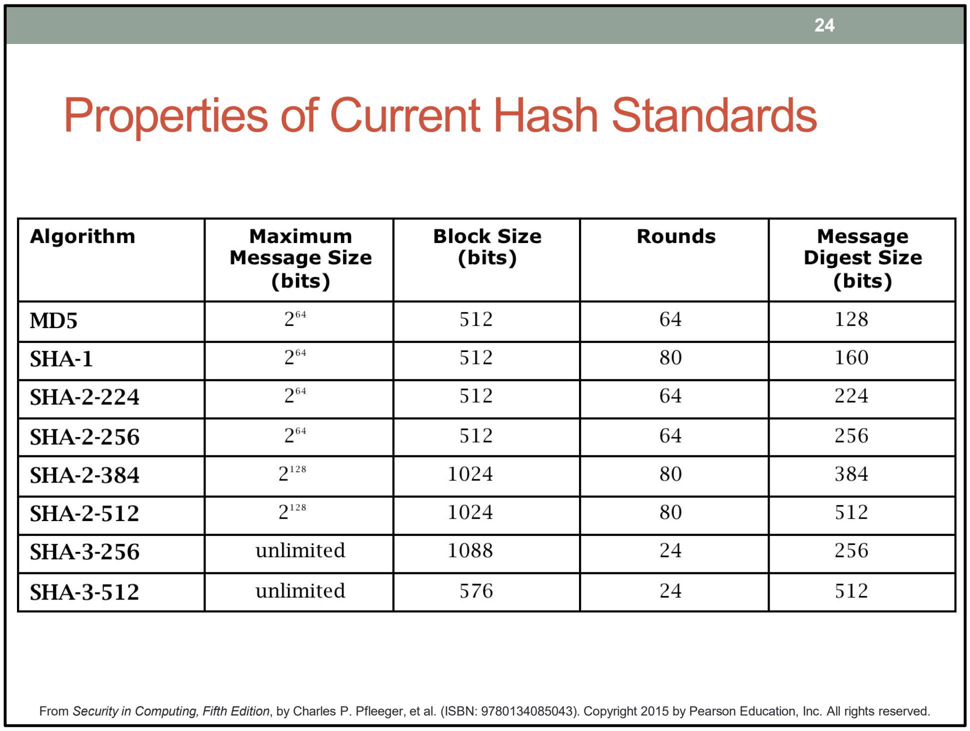
- The encryption key consists of the pair of integers (*e*, *n*), and the decryption key is (*d*, *n*)
- The value of *n* should be quite large, a product of two primes, *p* and *q*
- Typically, *p* and *q* are nearly 100 digits each, so *n* is approximately 200 decimal digits (about 512 bits) long
- A large value of *n* effectively inhibits factoring *n* to infer *p* and *q* (but time to encrypt increases as the value of *n* grows larger)
- A relatively large integer *e* is chosen so that *e* is relatively prime to ($p - 1$) * ($q - 1$). An easy way to guarantee that *e* is relatively prime to ($p - 1$) * ($q - 1$) is to choose *e* as a prime that is larger than both ($p - 1$) and ($q - 1$)
- Finally, select *d* such that $e * d = 1 \mod (p - 1) * (q - 1)$

These days, 2048-bit keys are increasingly becoming a standard requirement (thanks to increased computing power). The user of RSA distributes the value of *e* and *n* and keeps *d* secret.



Message Digests

- Previously introduced in Chapter 2, message digests are ways to detect changes to a block of data
- One-way hash functions are cryptographic functions with multiple uses:
  - They are used in conjunction with public-key algorithms for both encryption and digital signatures
  - They are used in integrity checking
  - They are used in authentication
  - They are used in communications protocols
- Modern hash functions meet two criteria:
  - They are one-way, meaning they convert input to a digest, but it is infeasible to start with a digest value and infer the input
  - They do not have obvious collisions, meaning that it is infeasible to find a pair of inputs that produce the same digest

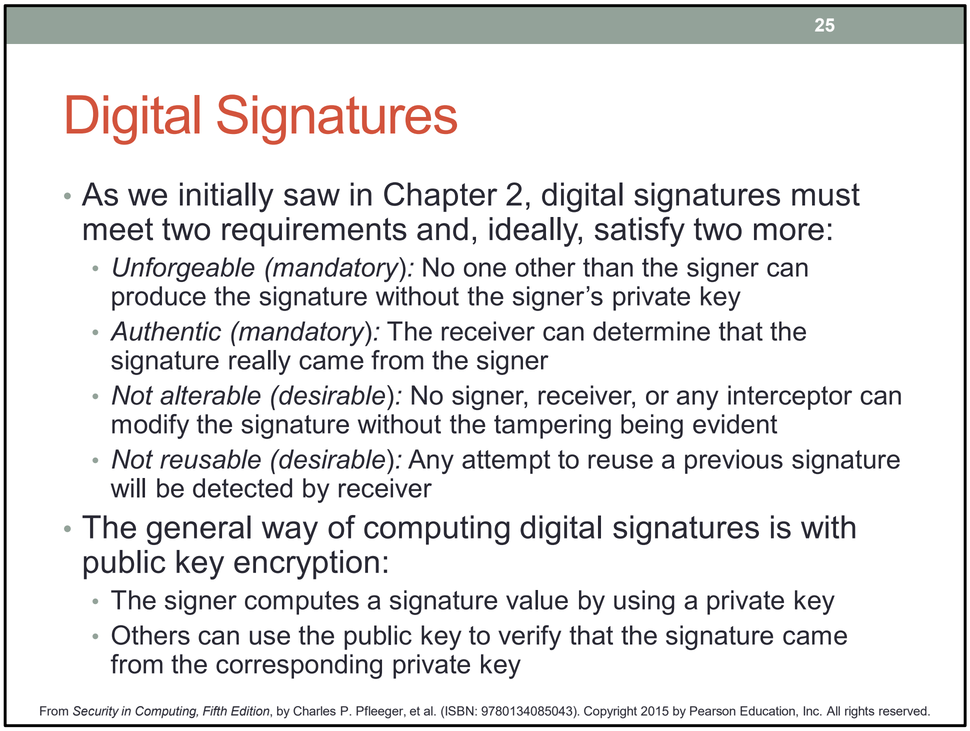


Properties of Current Hash Standards

Algorithm	Maximum Message Size (bits)	Block Size (bits)	Rounds	Message Digest Size (bits)
MD5	2^{64}	512	64	128
SHA-1	2^{64}	512	80	160
SHA-2-224	2^{64}	512	64	224
SHA-2-256	2^{64}	512	64	256
SHA-2-384	2^{128}	1024	80	384
SHA-2-512	2^{128}	1024	80	512
SHA-3-256	unlimited	1088	24	256
SHA-3-512	unlimited	576	24	512

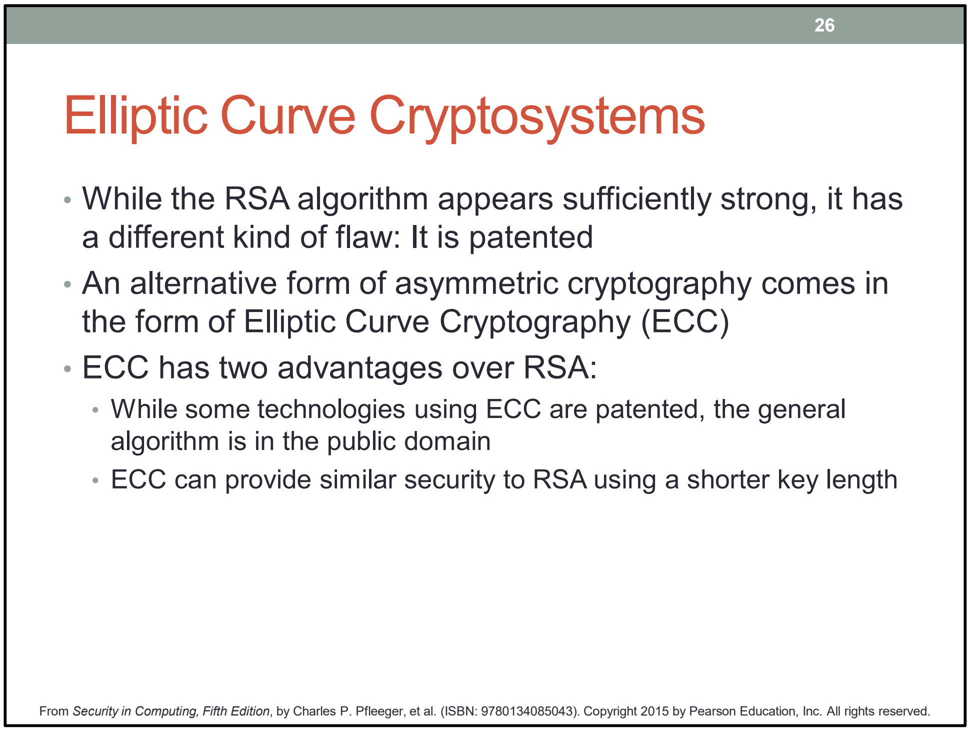
From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

In practice, for security purposes, MD5 and SHA-1 are too weak for modern use. SHA-3 has much better processing performance than its predecessors, but it is relatively new and has not yet stood the test of time (nor have its implementations).



Digital Signatures

- As we initially saw in Chapter 2, digital signatures must meet two requirements and, ideally, satisfy two more:
  - *Unforgeable (mandatory):* No one other than the signer can produce the signature without the signer’s private key
  - *Authentic (mandatory):* The receiver can determine that the signature really came from the signer
  - *Not alterable (desirable):* No signer, receiver, or any interceptor can modify the signature without the tampering being evident
  - *Not reusable (desirable):* Any attempt to reuse a previous signature will be detected by receiver
- The general way of computing digital signatures is with public key encryption:
  - The signer computes a signature value by using a private key
  - Others can use the public key to verify that the signature came from the corresponding private key

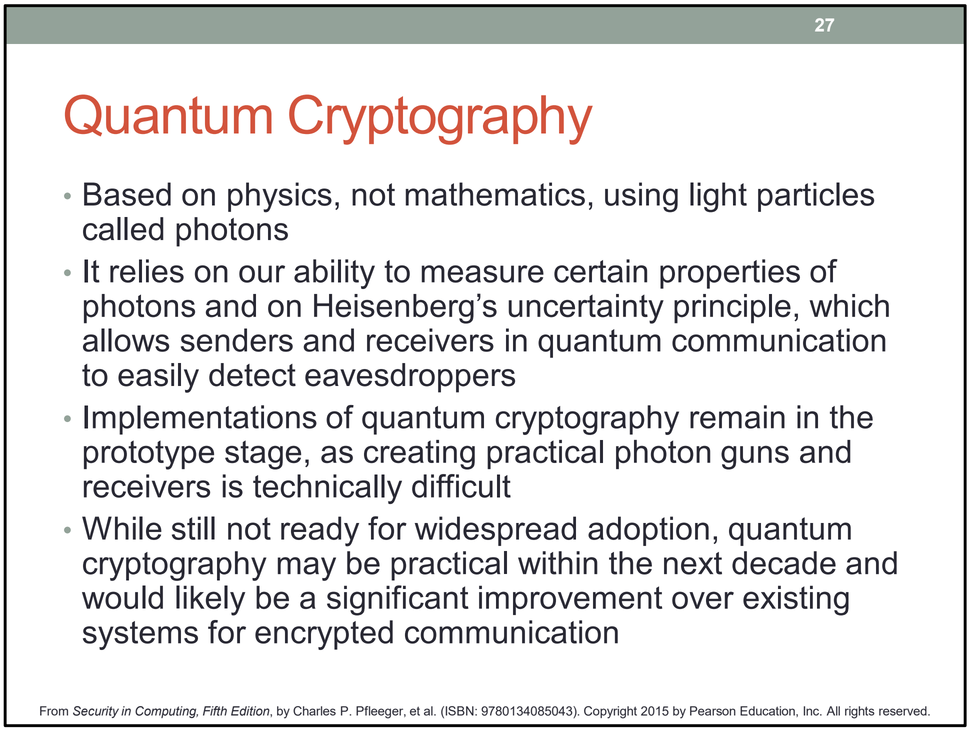


Elliptic Curve Cryptosystems

- While the RSA algorithm appears sufficiently strong, it has a different kind of flaw: It is patented
- An alternative form of asymmetric cryptography comes in the form of Elliptic Curve Cryptography (ECC)
- ECC has two advantages over RSA:
  - While some technologies using ECC are patented, the general algorithm is in the public domain
  - ECC can provide similar security to RSA using a shorter key length

From *Security in Computing, Fifth Edition*, by Charles P. Pfleeger, et al. (ISBN: 9780134085043). Copyright 2015 by Pearson Education, Inc. All rights reserved.

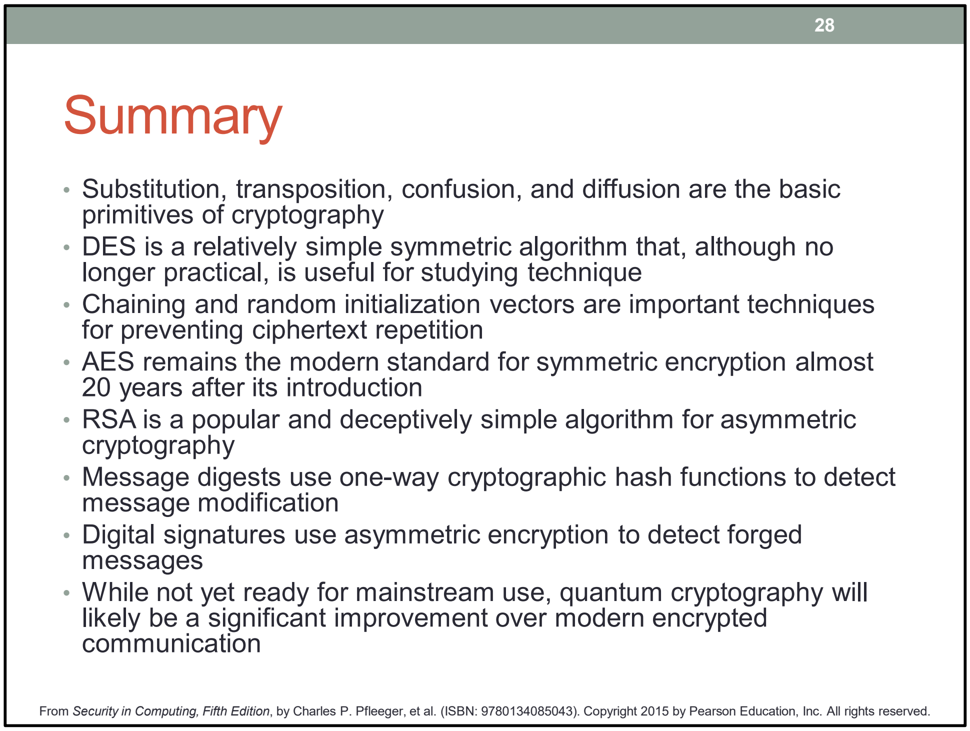
ECC use very complex math that is unnecessarily deep for this course, so we describe them here only at the highest level.



Quantum Cryptography

- Based on physics, not mathematics, using light particles called photons
- It relies on our ability to measure certain properties of photons and on Heisenberg’s uncertainty principle, which allows senders and receivers in quantum communication to easily detect eavesdroppers
- Implementations of quantum cryptography remain in the prototype stage, as creating practical photon guns and receivers is technically difficult
- While still not ready for widespread adoption, quantum cryptography may be practical within the next decade and would likely be a significant improvement over existing systems for encrypted communication

We deliberately do not go into great detail in these slides, as this is not a physics course. The goal here is for students to understand how quantum cryptography might impact security in the relatively near future. For more detail on the physics of quantum computing, see the textbook.



Summary

- Substitution, transposition, confusion, and diffusion are the basic primitives of cryptography
- DES is a relatively simple symmetric algorithm that, although no longer practical, is useful for studying technique
- Chaining and random initialization vectors are important techniques for preventing ciphertext repetition
- AES remains the modern standard for symmetric encryption almost 20 years after its introduction
- RSA is a popular and deceptively simple algorithm for asymmetric cryptography
- Message digests use one-way cryptographic hash functions to detect message modification
- Digital signatures use asymmetric encryption to detect forged messages
- While not yet ready for mainstream use, quantum cryptography will likely be a significant improvement over modern encrypted communication