

Introduction to Cryptography as an Upper-Level Elective

Seth D. Bergmann
Rowan University
Glassboro, NJ

Slides are available at:
<http://cs.rowan.edu/~bergmann>

Presented at CCSCE 2009
Villanova University

Seth D. Bergmann

- Teaching CS for 29 years
- Primary interest in Compilers
- Recent interest in Cryptography
 - Self-taught
 - Offered a one semester undergraduate course as advanced elective in the CS major, 2004, 2006, 2008

Objectives of this Tutorial

- Basic terminology of cryptography
- Basic concepts of cryptography
- How much math do you need?
- Provide some simple examples
- This tutorial is designed for cryptographic novices!

Prerequisites

- Math you'll need for this tutorial
 - What is a prime number?
 - Exclusive OR operation \oplus
 - Modulo arithmetic and exponents
- Prerequisites for my course
 - Foundations of CS (basic theory)
 - Data Structures
 - Recommended, at least two of the following
 - Linear Algebra
 - Probability and Stats
 - Digital Design (exclusive OR operation \oplus)

Basic Terminology

- Cryptography: Encryption and decryption for confidentiality.
- Cryptanalysis: Breaking the enemy's codes
- Cryptology: Cryptography and Cryptanalysis
- Plain text: Message to be encrypted
- Cipher text: Encrypted message, to be decrypted.
- Message integrity: Has anyone tampered with the message?
- Authenticity: Is the message being sent by an impostor?
- Key: Information used for encryption, decryption, etc.

Symmetric Key Cryptography

- Sender and receiver share a secret key.
- Same key is used for encryption and decryption.

Symmetric Key Cryptography (Classical) Example

- Encode the letters of the alphabet as 0..25

a = 0	f = 5	k = 10	p = 15	u = 20	z = 25
b = 1	g = 6	l = 11	q = 16	v = 21	
c = 2	h = 7	m = 12	r = 17	w = 22	
d = 3	i = 8	n = 13	s = 18	x = 23	
e = 4	j = 9	o = 14	t = 19	y = 24	

- Choose a key of any length, e.g. “bad” = 1 0 3
- Choose a plain text to be encrypted, e.g. “the red fox”

Symmetric Key Cryptography (Classical) Example: Encryption

- Encode the letters of the alphabet as 0..25

a = 0	f = 5	k = 10	p = 15	u = 20	z = 25
b = 1	g = 6	l = 11	q = 16	v = 21	
c = 2	h = 7	m = 12	r = 17	w = 22	
d = 3	i = 8	n = 13	s = 18	x = 23	
e = 4	j = 9	o = 14	t = 19	y = 24	

Encryption: add the letters of the key to the letters of the plain text, mod 26

Plain text =		t	h	e	r	e	d	f	o	x
=		19	7	4	17	4	3	5	14	23
+Key =		<u>1</u>	<u>0</u>	<u>3</u>	<u>1</u>	<u>0</u>	<u>3</u>	<u>1</u>	<u>0</u>	<u>3</u>
Cipher text =		20	7	7	18	4	6	6	14	0
=		u	h	h	s	e	d	d	o	a

Symmetric Key Cryptography (Classical) Example: Decryption

- Encode the letters of the alphabet as 0..25

a = 0	f = 5	k = 10	p = 15	u = 20	z = 25
b = 1	g = 6	l = 11	q = 16	v = 21	
c = 2	h = 7	m = 12	r = 17	w = 22	
d = 3	i = 8	n = 13	s = 18	x = 23	
e = 4	j = 9	o = 14	t = 19	y = 24	

Decryption: Subtract the letters of the key from the letters of the cipher text, mod 26

Cipher text =	u	h	h	s	e	d	d	o	a
=	20	7	7	18	4	6	6	14	0
Key =	- 1	0	3	1	0	3	1	0	3
Plain text =	19	7	4	17	4	3	5	14	23
=	t	h	e	r	e	d	f	o	x

Symmetric Key Cryptography

Modern

- Use the bit patterns of the plain text, e.g. ASCII
- Plain text can be *any* digital info, not just text
- Use Exclusive OR to encrypt and decrypt

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Symmetric Key Cryptography

Modern

- Properties of Exclusive OR

$$x \oplus 0 = x$$

$$x \oplus x = 0$$

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z$$

$x \oplus y \oplus z$ is not ambiguous

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Symmetric Key Cryptography

Modern

- Encryption

$$\text{plain} \oplus \text{key} = \text{cipher}$$

- Decryption

$$\text{cipher} \oplus \text{key}$$

$$= (\text{plain} \oplus \text{key}) \oplus \text{key}$$

$$= \text{plain} \oplus (\text{key} \oplus \text{key})$$

$$= \text{plain} \oplus 0$$

$$= \text{plain}$$

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Symmetric Key Cryptography (Modern) Example - Decryption

cipher \oplus key = plain

Use XOR to decrypt

cipher =	0010	1100
\oplus key =	<u>1011</u>	<u>1011</u>
plain =	1001	0111

Symmetric Key Cryptography

Modern

- Encryption

$$\text{plain} \oplus \text{key} = \text{cipher}$$

- Decryption

$$\text{cipher} \oplus \text{key} = \text{plain}$$

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Most modern symmetric key algorithms use XOR
(plus mish-mosh)

Symmetric Key Cryptography

Modern

- *Mish mosh* is used to hold off statistical attacks:
 - Substitution
 - Permutation
 - Repetition
 - Blocking modes
- Standard algorithms
 - DES – Data Encryption Standard (1976)
 - AES – Advanced Encryption Standard (2001)

Symmetric Key Cryptography

Modern

- Keys will be compromised!
- Key distribution is a problem!
- Keys need to be distributed through a secure (expensive) channel.

Asymmetric Key Cryptography (i.e. Public Key Cryptography)

- Each person has a pair of (related) keys
 - public key: Used for encryption
 - private key: Used for decryption
- Public keys are accessible to everyone.
- To encrypt a message to Bob, encrypt the message using Bob's public key.
- Bob uses his private key to decrypt the message.

Math Needed for Public Key Cryptography

- All arithmetic is done with whole numbers (modulo)

- Examples:

$$-3 \equiv 2 \equiv 7 \equiv 12 \equiv 17 \pmod{5}$$

$$-7 \equiv 0 \equiv 7 \equiv 14 \equiv 21 \pmod{7}$$

$$3^4 \equiv 1 \pmod{10}$$

- Multiplicative inverses are used extensively

$$3^{-1} \equiv 7 \pmod{10}$$

$$\text{because } 3 \times 7 = 21 \equiv 1 \pmod{10}$$

$$7^{-1} \equiv 3 \pmod{10}$$

Math Needed for Public Key Cryptography

- Prime number:

Whole number, greater than 1, which is divisible only by 1 and itself:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, ...

- From Fermat (1640) and Euler (1736):

If p and q are primes, $a^x \equiv a^{x \pmod{(p-1)(q-1)}} \pmod{p \cdot q}$

- Example: $p=5$, $q=7$

$3^{25} \equiv 3^{25 \pmod{(4 \cdot 6)}} \equiv 3^1 \equiv 3 \pmod{35}$

Public Key Cryptography - RSA

- Rivest, Shamir, Adleman (1977)
- Alice wishes to send an encrypted message to Bob using RSA
- Bob
 - chooses two large primes: p, q
 - $m = p * q$ is Bob's public modulus
 - $n = (p-1) * (q-1)$ is private
 - chooses a public encryption exponent, e , which shares no factors with n
 - calculates a private decryption exponent, $d \equiv e^{-1} \pmod{n}$
 - The values (m,e) form Bob's public key.
- Alice encrypts the plain text, msg , to produce ciphertext, c :
$$c \equiv msg^e \pmod{m}$$
- Alice sends ciphertext, c , to Bob
- Bob decrypts the ciphertext
 - $$c^d \equiv (msg^e)^d \pmod{m}$$
 - $$\equiv msg^{ed} \pmod{m}$$
 - $$\equiv msg^1$$

Public Key Cryptography - RSA

- No problem with key distribution!
- Each person has their own pair of related public/private keys
- To encrypt a message to someone, use their public key.
- The private keys do not need to be shared!
- Can RSA be broken?
 - The enemy, Eve, knows Bob's public modulus, m , and encryption exponent, e .
 - If she can factor m , she can calculate his private decryption key:
$$\text{factor}(m) \Rightarrow p, q$$
$$n = (p-1) * (q-1)$$
$$d = e^{-1} \pmod{n}$$
- **But factoring of large numbers is computationally hard !**

Public Key Cryptography

Benefit – Digital Signatures

- Public Key Cryptography also allows for digital signatures
- Proof that the sender is who they claim to be
- Bob wishes to send a message to Alice, with proof that the message is really from him.
- Bob applies his *private decryption* key, d , to the message.

- This is the signature:

$$\text{sig} = \text{msg}^d \pmod{m}$$

- Bob sends both the message and the signature to Alice.
- Alice applies Bob's *public encryption* key, e , to the signature.

$$\begin{aligned} \text{sig}^e \pmod{m} &= (\text{msg}^d)^e \pmod{m} \\ &= (\text{msg}^{de}) \pmod{m} \\ &\equiv \text{msg} \end{aligned}$$

- If the result is not the same as the message, she knows that either it is not from Bob, or someone has tampered with the message and/or signature.

Public Key Cryptography Problems

- For security, the public modulus needs to be huge, and encryption and decryption will be slow.
- Solution:
 - Alice creates a 'session key', encrypts it with Bob's public key.
 - Alice sends the encrypted session key to Bob.
 - Bob and Alice use the session key to communicate securely, using a standard symmetric-key algorithm, such as DES.

Public Key Cryptography Problems

- Authenticity can be a problem
- Eve sends a message to Alice:
From: Bob@gmail
Message: I am Bob, my public key is ...
- Alice will respond with a confidential message, encrypted with that public key, which Eve will be able to decrypt.

Public Key Cryptography

Authenticity

- Digital Certificates can be used to establish identity
- Certificate Authority is a trusted organization which can issue certificates (example: Verisign)
- The certificate is like a driver's license or passport; it certifies an individual's identity.
- The certificate will typically contain the individual's public key.
- Now Bob can send a message to Alice:

“Hi, I'm Bob, and my certificate is attached; it contains my public key, so that we can communicate securely”

Books

- Barr, Thomas H, *Invitation to Cryptology*, Prentice-Hall, 2002
 - Good textbook for undergrads
- Menezes, A.J., et. al, *Handbook of Applied Cryptography*, CRC, 1997
 - This is the ‘bible’
 - Use as a reference; not appropriate as a textbook
- Schneier, Bruce, *Applied Cryptography*, Wiley, 1996
 - Considered by some to be the ‘bible’
 - Use as a reference; not appropriate as a textbook
 - Has some good stories
- Trappe and Washington, *Introduction to Cryptography with Coding Theory*, Prentice-Hall, 2006
 - Good textbook for undergrads
 - More mathematically sophisticated than Barr

Questions?

- Thank you for attending

Seth D. Bergmann
Computer Science Department
Rowan University
Glassboro, NJ 08028

cs.rowan.edu/~bergmann

Lunch!