


Encryption
Introduction

CSCI E 45a: The Cyber World – part A

1 Copyright © Scott Bradner & Ben Gaucherin 2015

Learning goals



- Understand the various encryption technologies and a little bit of the history of encryption
- Understand the two basic types of encryption and what a cryptographic hash is and how it is used
- Understand digital signatures, certificates and certificate authorities




2 Copyright © Scott Bradner & Ben Gaucherin 2015

Introduction: this module

- This module deals with a mixture of technology, policy and operations practice






3 Copyright © Scott Bradner & Ben Gaucherin 2015

Topics (all required)

-  • **Background**
Basic concepts of encryption, History of encryption, encryption types
-  • **Symmetric encryption**
The history and technology of shared secret encryption
-  • **Asymmetric encryption**
The history and technology of public key encryption
- **Encryption keys**
Handling encryption keys

4 Copyright © Scott Bradner & Ben Gaucherin 2015

Topics, contd.

-  • **Hashes**
The technology and use of cryptographic hashes
-  • **Digital signatures**
How digital signatures work to protect information
- 
Scott's Public key • **Certificates**
Authenticating public keys
-  • **Public Key Infrastructure**
Certificate server structures
-  • **The Law**
Encryption and the law

5 Copyright © Scott Bradner & Ben Gaucherin 2015

Image credits

All drawings and photos by Scott Bradner unless noted

Slide#	credit
4	Jefferson - https://franceshunter.wordpress.com/2011/09/07/thomas-jefferson-the-cryptographer/ Enigma - https://en.wikipedia.org/wiki/File:Enigma-machine-bob-lord.jpg
5	sha - http://www.ulb.ac.be/di/scsi/sha3/program/index.html signature - http://recombo.com/ usa today - http://perspecsys.com/article/analysis-why-venture-capital-is-gushing-into-security/usa-today-logo-2/ key signing - http://resende.blogspot.com/2010/11/ive-been-to-a-keysigning-party-now-what.html mack truck - http://top-img.com/my/mack-truck-grill

6 Copyright © Scott Bradner & Ben Gaucherin 2015

Encryption Background

CSCI E 45a: The Cyber World – part A

1 Copyright © Scott Bradner & Ben Gaucherin 2015

Some terminology

The quick brown fox jumped over the lazy dog

↓

EA

↓

r[_0SG;{hnp6fM# +^!_fU->&TCbVbc WGeIMZbr!

↓

DA

↓


The quick brown fox jumped over the lazy dog

- **Clear-text**
The information un-encrypted
- **Cipher-text**
The information encrypted
- **Encryption algorithm**
A program to turn clear-text into cipher-text
- **Decryption algorithm**
A program to turn cipher-text into clear-text


2 Copyright © Scott Bradner & Ben Gaucherin 2015

Encryption: background

- **Purpose:**
Reversibly render material unreadable by an adversary
- **Mechanism:**
Mechanical device whose settings determine character string transformation
Mathematical algorithm that uses keys to determine character string transformation
Knowledge of both algorithm and keys is required



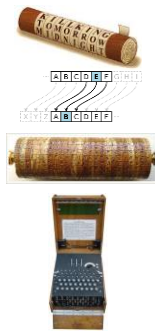
Jefferson wheel



RSA algorithm

3 Copyright © Scott Bradner & Ben Gaucherin 2015

Encryption: history



- **Hardware**
 - 400 BC: Spartan's scytales
 - 50 BC: Caesar cipher
 - 1795: Jefferson wheel
 - 1920: Enigma machine
- **Theory**
 - 1883: Kerchoffs' Principle
- **Software**
 - 1976: Public Key cryptography
 - 1977: DES
 - 1991: PGP
 - 1998: 3DES
 - 2001: AES

4

Copyright © Scott Bradner & Ben Gaucher in 2015

Encryption: vulnerabilities



- **Brute force guessing of keys**
 - harder, in general, with longer keys
- **Key exposed**
 - OpSec – operational security
- **Software bugs**
- **Faulty algorithms**
 - Flaw in design or back door
- **Faulty random number generators**

5

Copyright © Scott Bradner & Ben Gaucher in 2015

Encryption: vulnerabilities, contd.

CILLY this is message 1
CILLY another msg
CILLY yet a different one



Alan Turing

- **Knowing something about the text that was encrypted**
 - e.g., constant set of bits in separate messages
 - e.g., character probabilities in the message language
- Then analyzing multiple messages encrypted using same key

6

Copyright © Scott Bradner & Ben Gaucher in 2015

Encryption: types

- Symmetric encryption
Same key used to encrypt as to decrypt
- Asymmetric encryption
Different keys used to encrypt & decrypt

7 Copyright © Scott Bradner & Ben Gaucherin 2015

Image credits

All drawings and photos by Scott Bradner unless noted

Slide# credit
3 Jefferson - <https://franceshunter.wordpress.com/2011/09/07/thomas-jefferson-the-cryptographer/>
rsa T shirt - http://www.geocaching.com/geocache/GCW5YB_rsa

4 scytale
<http://www.westfieldnj.com/nis/Cryptography%20Project/History.htm>
caesar - https://en.wikipedia.org/wiki/Caesar_cipher
Jefferson - <https://franceshunter.wordpress.com/2011/09/07/thomas-jefferson-the-cryptographer/>
enigma <https://en.wikipedia.org/wiki/File:Enigma-machine-bob-lord.jpg>

5 <http://www.ikmb.uni-kiel.de/research/genetics-bioinformatics/bioinformatics/fpga-implementation>

8 turning - <http://www.stuartschool.org/turning/alan-turing>

Encryption

Symmetric encryption

CSCI E 45a: The Cyber World – part A

1 Copyright © Scott Bradner & Ben Gaucherin 2015

Symmetric Encryption

- Same key used for encryption and decryption
- Original encryption concept used by Julius Caesar and Captain Midnight ring also Enigma Machine and DES and AES

2 Copyright © Scott Bradner & Ben Gaucherin 2015

One Time Pad

- Theoretically perfect encryption
- Because key only used once
No across-message patterns to find

3 Copyright © Scott Bradner & Ben Gaucherin 2015

German Enigma Machine



- Used by Germany in WWII
- Broken by Polish Cipher Bureau
“break was biggest secret of war”
- Operation
 - Set “code of the day” in rotors using dials
 - Press key to encrypt character
 - Resulting character is illuminated
 - Every keypress advances dials
 - So repeated key gets different results

4

Copyright © Scott Bradner & Ben Gaucherin 2015

Data Encryption Standard (DES)



Horst Feistel

- US National Bureau of Standards issued RFP for a “Data Encryption Standard” in 1974
- One proposal from Horst Feistel at IBM (Lucifer cipher)
 - Developed to protect cash dispensing machines
 - 128 bit key

5

Copyright © Scott Bradner & Ben Gaucherin 2015

DES, contd.

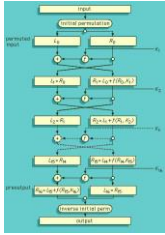


- NSA reviewed and changed Lucifer to use 56 bit key
 - Some worried about NSA changes adding vulnerabilities
 - Some worries were later dispelled, but not key length issue
 - The revised Lucifer adopted as the DES standard in January 1977

6

Copyright © Scott Bradner & Ben Gaucherin 2015

DES, contd.



- Data Encryption Standard
 - Old US federal standard
 - Symmetric system
 - 56 bit key
 - Data encrypted in chunks of 64 bits
 - Generally used in chaining or feedback mode
- Results of encrypting one chunk used in encrypting next chunk
Special operation on 1st chunk helps mask data repetitions

7

Copyright © Scott Bradner & Ben Gaucher in 2015

DES, breakability



- 1977 - Hellman described \$20 M machine to find DES keys by brute force
Trying 256 possible keys, could break one key per day
- 1997 - Goldberg & Wagner proposed \$45,000 cracking machine using FPGAs
- 1998 - EFF built "Deep Crack" DES key finder
Cost: \$250,000
Found a DES key in 56 hours
Deep Crack showed that spy agencies had been able to break DES for quite a while

8

Copyright © Scott Bradner & Ben Gaucher in 2015

DES, breakability, contd.

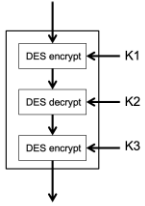


- 1999: NIST recommended triple-DES
DES only permitted for legacy systems
- 2008: RIVYERA
Brute force engine can find a DES key in a day average
128 FPGA-based machine
292 B keys/sec
up to \$1.3 M

9

Copyright © Scott Bradner & Ben Gaucher in 2015


Triple-DES (3DES)




- Encrypt with DES key 1
- “Decrypt” result with DES key 2
- Encrypt result with DES key 3
- Very strong, $3 \times 56 = 168$ independent key bits
- Other keying options
 - With $k_3 = k_1$, $2 \times 56 = 112$ independent key bits
 - Backwards compatible with DES if all three keys are identical (but no better than DES, with 56 bits)

10 Copyright © Scott Bradner & Ben Gaucherin 2015

Advanced Encryption Standard (AES)



Vincent Rijmen




Joan Daemen

- a.k.a. Rijndael
- Selected by a public competition to replace DES
 - Proposers tried to break competitors’ proposals
- Can be implemented in hardware or software
- Federal Standard
 - Effective in 2002 - FIPS 197
- No patents (in theory)
- Multiple key lengths
 - 128, 192 & 256
- Not known to be breakable

11 Copyright © Scott Bradner & Ben Gaucherin 2015

AES, contd.



- AES 128 Brute force: with 1M CPUs: $1M \times$ age of universe
- AES & US government rules
 - AES 128 – good enough for US SECRET information
 - AES 256 – good enough for US TOP SECRET information (both assuming the implementation is correct)
- So AES should be good enough for your data
- Note, weak point likely to be key management

12 Copyright © Scott Bradner & Ben Gaucherin 2015

Image credits

All drawings and photos by Scott Bradner unless noted

Slide# credit

3 <http://www.cryptomuseum.com/crypto/otp.htm>

4 <https://en.wikipedia.org/wiki/File:Enigma-machine-bob-lord.jpg>

6 NSA - https://commons.wikimedia.org/wiki/File:National_Security_Agency.svg

7 des <http://www.britannica.com/topic/Data-Encryption-Standard/images-videos/Flow-diagram-for-the-16-step-Data-Encryption-Standard-operation/59837>

8 https://en.wikipedia.org/wiki/EFF_DES_cracker

9 <http://www.ikmb.uni-kiel.de/research/genetics-bioinformatics/bioinformatics/fpga-implementation>

13

Copyright © Scott Bradner & Ben Gaucherin 2015

Encryption
Asymmetric encryption

CSCI E 45a: The Cyber World – part A

1 Copyright © Scott Bradner & Ben Gaucherin 2015

Asymmetric Encryption

- Uses two mathematically related keys
Anything encrypted by one key can only be decrypted by the other
Works both ways
But can not decrypt with the key used to encrypt
- Keys are large (≥ 512 bits)

2 Copyright © Scott Bradner & Ben Gaucherin 2015

Asymmetric Encryption: history

Whitfield Diffie

Martin Hellman

- Early 1970s - secret work in England - Government Communications Headquarters (GCHQ)
British intelligence agency focusing on signals intelligence (SIGINT) and information assurance
- 1976 - Whitfield Diffie & Martin Hellman - proposed idea of public key cryptography and a method for secure key exchange

3 Copyright © Scott Bradner & Ben Gaucherin 2015

Asymmetric Encryption example: RSA



Ron Rivest, Adi Shamir, Len Adleman

- 1978 - Rivest, Shamir and Adleman (all at MIT) published RSA algorithm
- MIT obtained patent - rights assigned to RSA, Inc.



4

Copyright © Scott Bradner & Ben Gaucherin 2015

RSA Algorithm

```
#!/bin/perl -  
sp0777i<-X+d*IMLa**IN%  
0]dsXx++!MIN/dsMO<]dsj  
$/=unpack("H*",$_);  
  
$_="echo  
16dio(U$k"SK$/SM$nEs  
N0p|IN*1  
lK[d2%Sa2/d0$^ixp"dc';s  
^W//g;$_=pack("H*",(L.*)"")  
$/)
```

- Product of two large (~100 digit) prime numbers used as base
- Mathematical manipulation of this product produces the keys
 - Keys should be 1024 bits long or longer
- U.S. patents expired 09/20/2000

5

Copyright © Scott Bradner & Ben Gaucherin 2015

Public key encryption



- A use of asymmetric encryption
- Each person has a asymmetric encryption key-pair
- Designate one key as a "public key"
- Designate the other key as a "private key"

6

Copyright © Scott Bradner & Ben Gaucherin 2015

Public key encryption, contd.



Scott's Public key

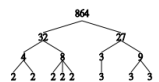


- Anyone can know the public key
The only use is to encrypt something that can only be decrypted by the corresponding private key
- Anything encrypted with a private key can only be decrypted by the corresponding public key
- The private key must be kept secret

7

Copyright © Scott Bradner & Ben Gaucher in 2015

RSA Breakability



- Depends on difficulty in factoring large numbers
- Can lengthen numbers used in future
Current recommended length is 2048-bits
- Can be mathematically shown not to have a trap door

8

Copyright © Scott Bradner & Ben Gaucher in 2015

RSA Breakability, contd.



Peter Shor

- Cannot be "proven" to be unbreakable
But many have tried and failed
Except brute force: 250 digit RSA key using 2700 years of processor time using a cluster of thousands of computers around the world
Processing required doubles for each added 10 bits of key length
- Quantum research may provide tool
Shor's Algorithm: non-linear method of factoring composite numbers - from 1994 by Peter Shor

9

Copyright © Scott Bradner & Ben Gaucher in 2000

Image credits

All drawings and photos by Scott Bradner unless noted

Slide# credit

3 GCHQ -

https://en.wikipedia.org/wiki/File:Government_Communications_Headquarters_logo.svg

diffy

<http://www.computerhistory.org/fellowawards/hall/bios/Whitfield,Diffie/>

hellman https://en.wikipedia.org/wiki/Martin_Hellman

4 [http://www.usc.edu/dept/molecular-science/RSA-](http://www.usc.edu/dept/molecular-science/RSA-2003.htm)

2003.htm

5 <http://www.cypherspace.org/adam/rsa/>

7 safe -

<http://locksmith.armstronglock.com/blog/2013/april/important-documents-you-should-store-in-your-hom.aspx>

8 https://en.wikipedia.org/wiki/Integer_factorization

9 shor https://en.wikipedia.org/wiki/Peter_Shor

10

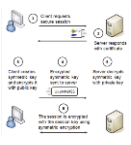
Copyright © Scott Bradner & Ben Gaucherin 2015

Encryption
Encryption keys

CSCI E 45a: The Cyber World – part A

1 Copyright © Scott Bradner & Ben Gaucherin 2015

Keys for encryption




HTTPS key exchange


- Public key algorithms are more expensive (require more computing) than symmetric key algorithms
- So most public key systems combine both public and symmetric key algorithms

2 Copyright © Scott Bradner & Ben Gaucherin 2015

Creating a session key



Whitfield Diffie



Martin Hellman

- Diffie-Hellman can be used to securely develop a symmetric session key using public key technology
- Session key is secure even if message exchange is monitored
- New key for each session - user cannot find out old session keys
- Use session key with "agreed to" algorithm to encrypt session traffic

3 Copyright © Scott Bradner & Ben Gaucherin 2021

Diffie-Hellman Key Agreement Protocol

Alice and Bob need to share a secret key (S) without meeting

$ga = g^a \text{ mod } p$
 $gb = g^b \text{ mod } p$

S for Alice: $gb^a \text{ (mod } p)$
 S for Bob: $ga^b \text{ (mod } p)$

to be secure, p should be at least 2048 bits

- p and g are public, p is a prime number and g is a base of p
- Alice chooses a secret number a which is less than p , and sends Bob $g^a \text{ (mod } p)$
- Bob chooses a secret number b which is less than p , and sends Bob $g^b \text{ (mod } p)$
- Secure because of difficulty of computing discreet logs of large numbers

4 Copyright © Scott Bradner & Ben Gaucher in 2021

Keys required: secret keys

Total # keys = $\frac{(n)(n-1)}{2}$

Secret keys per participant = $n-1$

- Issues with secret keys
 - 1/a key needed per pair of correspondents
 - To keep a particular conversation secret from others
 - 2/must exchange keys in a secure way
 - To keep others from learning key

5 Copyright © Scott Bradner & Ben Gaucher in 2021

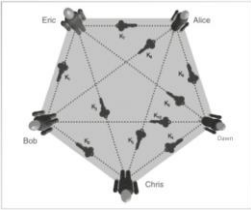
Keys required: secret keys

- 3 people need 3 keys to privately communicate
- # secret keys per participant = 2

6 Copyright © Scott Bradner & Ben Gaucher in 2015

Keys required: secret keys

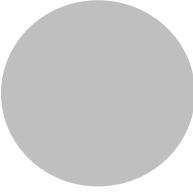
- Five people need 10 keys to privately communicate
- # secret keys per participant = 4



7 Copyright © Scott Bradner & Ben Gaucherin 2015

Keys required: secret keys

- 1000 people need 499,500 keys to privately communicate
- # secret keys per participant = 999
one for each of the other people



8 Copyright © Scott Bradner & Ben Gaucherin 2015

Keys required: public keys

- Each person only needs a single key-pair

Total # keys = $n \times 2$

Secret keys per participant = 1

- 1 "private key"
- 1 "public key"

So 1,000 people only need 1,000 key pairs

- # secret keys per participant = 1
i.e., only their own private key

9 Copyright © Scott Bradner & Ben Gaucherin 2015

Key Exchange: secret keys



- Must get key to other party in secret
 - Can be quite a problem if there is a risk of eavesdropping
- Many schemes used over the years (phonebooks, etc.)

10

Copyright © Scott Bradner & Ben Gaucherin 2015

Key Exchange: public keys



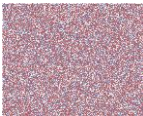
Scott's Public key

- Public key algorithms
 - the "public key" is a non-secret
 - Since it can only be used to encrypt things for the "private key" to decrypt
 - So it's easy to distribute the key needed to encrypt
- But how do I know that it's *your* key?
 - We will come back to this question - certificates

11

Copyright © Scott Bradner & Ben Gaucherin 2015

Random Numbers



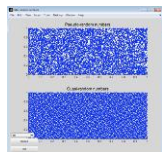
30,000 random numbers

- Good random numbers required in cryptography in generating keys and digital signatures
 - Random numbers also used in TCP for initial sequence numbers
- Hard to get good random numbers
 - RFC 4086 discusses randomness requirements

12

Copyright © Scott Bradner & Ben Gaucherin 2015

Random Numbers, contd.



- Small biases in a sequence of “random” numbers can provide a hook to discover the key
- Some worry that the NSA put a backdoor in a NIST random number generator standard

13

Copyright © Scott Bradner & Ben Gaucherin 2015

Image credits

Slide#	credit
2	http://blogs.getcertifiedgetahead.com/understanding-https-process-security/
3	diffie http://www.computerhistory.org/fellowawards/hall/bios/Whitfield, Diffie/
4	hellman https://en.wikipedia.org/wiki/Martin_Hellman Based on an example from <i>Unlocking Information Security</i> , Tel Aviv University. Image: Randal L. Carr
6 & 7	Simson Garfinkel
10	https://www.pinterest.com/pin/289637819762716695/
12	http://blog.codinghorror.com/computers-are-lousy-random-number-generators/
13	http://www.nag.co.uk/IndustryArticles/usingtoolboxmatlabpart3.asp

14

Copyright © Scott Bradner & Ben Gaucherin 2015

Encryption

Hashes

CSCI E 45a: The Cyber World – part A

1 Copyright © Scott Bradner & Ben Gaucherin 2015

Hash Functions

- **Hash** = short mathematical summary of data
Fixed length hash values created from variable length inputs

A	→ hash →	bf072e9119077b4e76437a93986787ef
B	→ hash →	30cf3d7d133b08543cb6c8933c29dfd7
ABCDEF	→ hash →	f6674e62795f798fe2b01b08580c3fdc
abcfed	→ hash →	5ab557c937e38f15291c04b7e99544ad
111111	→ hash →	77a319564621b96fa0656e24c67960ef

2 Copyright © Scott Bradner & Ben Gaucherin 2015

Hash features

- “computationally infeasible” to:
1/ Recreate original message knowing hash
2/ Make up new message that produces same hash
Takes up to $2^{\text{hash-length}}$ tries
- Half or more of the output bits can change with a 1-bit change on the input

2^{hash-length}

A	→ hash →	bf072e9119077b4e76437a93986787ef
B	→ hash →	30cf3d7d133b08543cb6c8933c29dfd7

3 Copyright © Scott Bradner & Ben Gaucherin 2015

Hash Use in Communications

- Hash of message is calculated by sender and appended to message
- Receiver removes hash from message then computes hash of the remainder of the message
- If the hash values are equal the message was not modified in transit
i.e., a high quality checksum

4 Copyright © Scott Bradner & Ben Gaucher in 2015

Hash Use in Communications, contd.

- Also used by file integrity checkers
Make and save hashes of program files
Periodically take hashes of program files and compare to stored values
E.g. application: Tripwire
Commercial & open source versions

5 Copyright © Scott Bradner & Ben Gaucher in 2015

Keyed Hash Function

- Add a secret key to input data before computing hash
- Receiver must add same key to message before computing hash of message body
- Verifies sender because only sender would know key
- Used in message authentication codes (MAC)

6 Copyright © Scott Bradner & Ben Gaucher in 2015

Keyed Hash Function, contd.



- No encryption done to make or verify keyed hash
- Thus does not violate any encryption controls including export controls

7

Copyright © Scott Bradner & Ben Gaucherin 2015

Hash Algorithms



Ron Rivest



- MD5 - Message Digest 5 (Rivest)
128 bit hash
Not recommended for new applications
- SHA-1 - Secure Hash Algorithm
Developed by NSA
Published by NIST
160 bit hash - "souped-up version of MD5"
NIST said to stop using SHA-1 before 2010

8

Copyright © Scott Bradner & Ben Gaucherin 2015

Hash Algorithms, contd.



- SHA-2 (SHA-256, SHA-384 & SHA-512)
Designed by NSA
Published by NIST
Federal standards - different hash lengths
- SHA-3 competition selected Oct 2012
"Keccak" selected to be SHA-3 after public competition
NIST: SHA-3 is a standard (FIPS 202)
Published August 5, 2015

9

Copyright © Scott Bradner & Ben Gaucherin 2015

Goals of attacks on hash functions

```
graph TD; A[?????] --> B[hash]; B --> C["Bf072e9119077b4e764  
37a93986787ef"]; C --> D[unhash]; D --> E[A];
```

- Determine how to craft an input to produce a given hash value
E.g., creating an input (e.g., program + virus) that would be verified by an existing digital signature
- Determine what was hashed
E.g., finding out what password was stored as a hash value

10 Copyright © Scott Bradner & Ben Gaucherin 2015

Produce a given hash value

```
graph TD; A[z13F] --> B[hash]; B --> C["Bf072e9119077b4e764  
37a93986787ef"];
```

- Major hash algorithms have been “broken” in such a way that you can create a given hash value
SHA-0, SHA-1, MD4, MD5, HAVAL-128, and RIPEMD
- “Broken” means that we have found a way to get around the computational complexity
- E.g., NIST: must not use SHA-1 after 12/31/2014

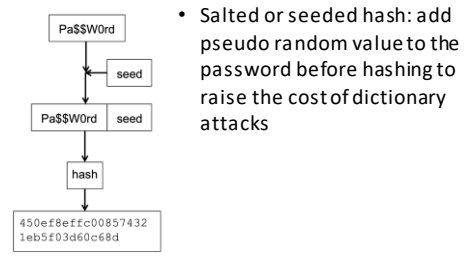
11 Copyright © Scott Bradner & Ben Gaucherin 2015

Determine what was hashed

- E.g., passwords stored as hash values
Common in UNIX, Active Directory, databases for web applications, etc.
- Dictionary attack – attack that uses large collections of known possible/probable options to try them out
In this context compare the hashed password you are trying to “break” to pre-computed hash values (e.g., rainbow tables) for list of passwords people are known to use

12 Copyright © Scott Bradner & Ben Gaucherin 2015

Determine what was hashed, contd.



- Salted or seeded hash: add pseudo random value to the password before hashing to raise the cost of dictionary attacks

Image credits


- All drawings and photos by Scott Bradner unless noted
- | Slide# | credit |
|--------|---|
| 7 | http://www.americanconference.com/encryption |
| 8 | http://people.csail.mit.edu/rivest/ |
| 9 | http://www.ulb.ac.be/di/scs/sha3/program/index.html |
| 12 | http://www.recycledplasticbuildingmaterials.co.uk/picnic-tables/junior-multicoloured-picnic-table-ribble-rainbow-range-recycled-plastic.html |

Encryption
Digital signatures

CSCI E 45a: The Cyber World – part A

1 Copyright © Scott Bradner & Ben Gaucherin 2015

Digital Signatures



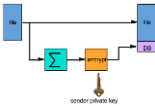
- Need method to be sure that a message came from **A** and has not been modified

Use a *Digital Signature*

Append to message before sending

2 Copyright © Scott Bradner & Ben Gaucherin 2015

Digital Signature, sending process



- **A** computes a hash of the contents of the message
- **A** encrypts the hash code with **A**'s **private** key the result is appended to the message

Encrypting a hash with a private key = "*signing the message*"

3 Copyright © Scott Bradner & Ben Gaucherin 2015

Digital Signature, checking process

- When it gets the message, **B** computes the same hash function on the body of the message
- **B** then decrypts the received hash code using **A's public** key
- If the hash codes match, the message came from **A** and the contents were not altered in transmission

4 Copyright © Scott Bradner & Ben Gaucherin 2015

Digital Signature contd.

5 Copyright © Scott Bradner & Ben Gaucherin 2015

Digital Signatures, Features

If it's not accurate, it might as well not exist.

- **Data integrity**
Ensure that the data did not change since DS created
- **Data origin authentication**
Only person with knowledge of your private key could create DS
So I can be sure you created it
- **Non-repudiation of origin**
Different way to say data origin authentication
I can show that it must have been you who created DS
Unless you can show that your private key was compromised

It could only have come from Bill.

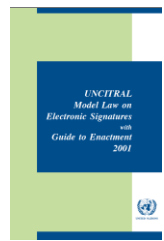
6 Copyright © Scott Bradner & Ben Gaucherin 2015

Digital Signatures, Legality



- US Public Law 106-229
A Federal law that says a contract is not void just because a digital signature was used on it
Does not require that everyone accept digital signatures
Signed electronically (and with a pen) by President Clinton
- Some states also have enabling laws

Digital Signatures, international



- The United Nations Commission on International Trade Law (UNCITRAL) adopted an updated model law on electronic signatures in 2001

Image credits

All drawings and photos by Scott Bradner unless noted

Slide#	credit
2	http://recombo.com/
6	http://www.ibswblog.com/category/data-integrity/
7	https://www.govtrack.us/congress/bills/106/s/761
8	http://www.uncitral.org/uncitral/en/uncitral_texts/electronic_commerce/2001Model_signatures.html

Encryption
Certificates

CSCI E 45a: The Cyber World – part A

1 Copyright © Scott Bradner & Ben Gaucherin 2015

Certificates



Scott's Public key

- I need to find out your public key to send you a secure message
- I need to get your public key in a secure, non-forgable way
- You need to find out my public key to authenticate that the message is from me
- You need to get my public key in a secure, non-forgable way

2 Copyright © Scott Bradner & Ben Gaucherin 2015

Certificates, contd.

public key
signature

- A **certificate** is a public key with one or more digital signatures
Digital signature: hash of public key encrypted with a particular private key
- “signed” by someone or some organization you trust
Personal knowledge or Certificate Authority (CA)

3 Copyright © Scott Bradner & Ben Gaucherin 2015

X.509 certificates

- X.509 is ISO standard for digital certificates
- Types of X.509 certificates
 - Certificate Authority Certificates
 - User Certificates
 - Server Certificates
- Data in base certificate
 - Version - X.509 version
 - Serial Number - assigned by CA
 - Algorithm - identifies hash and signature algorithms used
 - Issuer - name of CA
 - Validity - start and end of valid time
 - Subject - name of entity certificate identifies

4 Copyright © Scott Bradner & Ben Gaucherin 2015

X.509 Certificates

- X.509 v3 adds extensibility
- Extensions support use in the Internet
 - AlternativeNames (DNS or other names)
 - KeyUsage (limit functions certificate can be used for)
- RFC 3779 adds IP information
 - IP addresses
 - ASNs

5 Copyright © Scott Bradner & Ben Gaucherin 2015

Certificates, contd.

- e.g., one of Ben's PGP certificate signatures

Search results for '0xf41b2de887e52625'

```

Type bits/keyID cr. time exp time key expir
-----
pub 4096R/04420840 2013-09-05
uid Ben.Gaucherin@cs.harvard.edu [email]
sig 0xf41b2de887e52625 2013-09-05 2017-09-05 [selfsig]
sig 0xf41b2de887e52625 2013-09-04 2017-09-04 [Amelia R. McDonald - amem@alumia.com]
sig 0xf41b2de887e52625 2013-09-04 2017-09-04 [Benjamin M. Smith - bsmith@alumia.com]
sig 0xf41b2de887e52625 2013-09-04 2017-09-04 [Leri C. Meala - leri@alumia.com]
sig 0xf41b2de887e52625 2013-09-04 2017-09-04 [Christopher Michael - cmichael@alumia.com]
sig 0xf41b2de887e52625 2013-09-04 2017-09-04 [Anna Vesio - avasio@alumia.com]
pub 4096R/04420840 2013-09-05 2017-09-05 []
sig 0xf41b2de887e52625 2013-09-05
    
```

- Note: Ben signed his own certificate
 - To ensure integrity of public key during signing process

6 Copyright © Scott Bradner & Ben Gaucherin 2015

Certificate Authority

- A trusted third party - issues certificates for others to use
- CA process:
 - User creates public/private keypair
 - User sends Certificate Signing Request (CSR) to the CA
 - Includes public key (but not private key)
 - CA verifies the sender's identity & signs certificate
 - CA sends the certificate back to the user

7 Copyright © Scott Bradner & Ben Gaucher in 2015

Certificate Authority, contd.

- CA's public key must be known within user community
- 100s of commercial CAs plus industry, enterprise and personal CAs
 - Main difference is scope of the trust of the CA
 - Multiple CAs have been hacked or subverted
- Browsers include a list of "trusted" CAs
 - Firefox includes more than 200

8 Copyright © Scott Bradner & Ben Gaucher in 2015

What does a Certificate Mean?

- It means that a CA believes some information about you
- What information depends on what is verified during certification process
 - Could be just a name, real or not
 - Could include that you work for Harvard University
 - Could say that you are a female over 18 years of age

Betsy Parker
Harvard University
bparker6723@harvard.edu

9 Copyright © Scott Bradner & Ben Gaucher in 2015

Trusting Certificates

- How much trust should you put into a certificate?
- Depends on the process CA used to verify your information



Could be just that requester had a credit card #

Name on credit card used in certificate

Could mean you presented a government picture ID

...
Current commercial processes very mixed in quality

What Can You Do With a Certificate?



- With a Server Certificate
You can run a HTTPS web server
- With a User Certificate
You can identify yourself
You can send your public key to someone so they can send you encrypted messages or verify signed messages you send them
You can prove you are part of some group if the CA only gives certificates to members of the group
e.g., Harvard employees

I am Betsy Parker

signature

Certificates Are Not Good Forever

- What can go wrong?
 - Users forget their private key passphrase
 - Private keys get compromised
 - User gets fired from company and a certificate in the name of user@company.com is no longer valid
 - CA can sign the wrong certificate
 - CA can get hacked
- Thus, you need to have a way to check if a certificate is still valid

HR accidentally signed malware, will revoke certificate

From: "HR" <hr@company.com>, get signed by malware

Unknown CAs



- How do you check the validity of an unknown CA?

Either, the CA public key is built into browsers

Need to assume browser developers used good process to get CA public keys
Does not happen with personally created certificates

Or, a CA could have their certificate signed by a 'higher-level' CA

Or, trust it for some other reason
e.g., knowing the individual running the server

16

Copyright © Scott Bradner & Ben Gaucherin 2015

Image credits

All drawings and photos by Scott Bradner unless noted

Slide# credit

4

<http://www.x500standard.com/index.php?n=X509.X509P>

KC

7

http://dx.courts.wa.gov/index.cfm?fa=dx.displayServicePage&serviceName=VehicleRelatedViolations&item=technical/how_to_obtain_certificates.html

14

<http://www.webune.com/forums/https-error-sec-error-revoked-certificate.html>

15

<http://securityaffairs.co/wordpress/24240/digital-id/certificate-revocation-checks-heartbleed.html>

16

https://developer.mozilla.org/en-US/docs/Archive/Security/Introduction_to_Public_Key_Cryptography

17

Copyright © Scott Bradner & Ben Gaucherin 2015

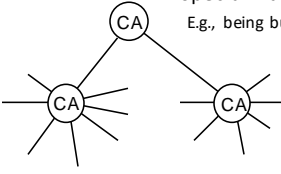
Encryption
Public key infrastructure

CSCI E 45a: The Cyber World – part A

1 Copyright © Scott Bradner & Ben Gaucherin 2015


Public key Infrastructure (PKI)

- Hierarchical infrastructure of certificate authorities
Provides a chain of trust
- Find out the public key of the highest level CA in some special way
E.g., being built into the browser



2 Copyright © Scott Bradner & Ben Gaucherin 2015

PKI Issues



- A PKI would be good except
 - Need system that covers all relevant users
 - Corporate-wide for corporate applications
 - World-wide for general Internet commerce
 - Liability issues: what could CA be liable for?
 - Privacy issues: identity assurance - how about anonymity?
 - Jurisdictional relationships: what laws to follow?
 - Local CA procedures: how to know what identity assurance was used to create certificate?

3 Copyright © Scott Bradner & Ben Gaucherin 2015

PKI future

- Global PKI will not happen soon
but many application-specific PKIs already exist

E.g., Resource PKI

4 Copyright © Scott Bradner & Ben Gaucher in 2015

Web-of-Trust

- web-of-trust - see www.gnupg.org
You send me certificate signed by someone I know
or who I know would only sign your certificate if he or she knew that you were you

5 Copyright © Scott Bradner & Ben Gaucher in 2015

Web-of-Trust Issues

- Can be hard to find public key of random person
Some key repositories exist but do you know any of the signors?
e.g., pgp.mit.edu
But not easy to find the right key
- No easy CRL function
e.g., Scott's sob@harvard.edu certificate in the MIT repository is no longer valid
It is in an old, no longer supported, format

6 Copyright © Scott Bradner & Ben Gaucher in 2015

Web-of-Trust, key signing

- Key signing process can be time consuming

See:
http://www.cryptnet.net/fdp/cryptnet/keysigning_party/en/keysigning_party.html#overview



7

Copyright © Scott Bradner & Ben Gaucherin 2015

Image credits

All drawings and photos by Scott Bradner unless noted

Slide# credit

4 <http://www.potaroo.net/ispcol/2008-12/resourcecertificates.html>

7 <http://lresende.blogspot.com/2010/11/we-been-to-a-keysigning-party-now-what.html>

8

Copyright © Scott Bradner & Ben Gaucherin 2015

Encryption
The law

CSCI E 45a: The Cyber World – part A

1 Copyright © Scott Bradner & Ben Gaucherin 2015


Law and Encryption



- Encryption software is officially illegal in some countries
e.g., Russia, France, South Africa, China & Ukraine
- Some countries will require you to give your keys if asked by law enforcement
- A secret key gets in the way of governments to find out what is going on
Assertions that criminals can hide behind encryption
Saying something different than the government says is criminal in some places

2 Copyright © Scott Bradner & Ben Gaucherin 2015

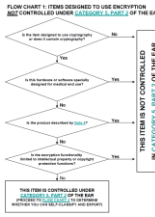
Law and Encryption, contd.



- Illegal to export encryption from the U.S. to some countries
Iran, North Korea, Sudan, and Syria
Must get a license or remove encryption technology before visiting
- Otherwise export for your own use is OK
e.g., in your own laptop when you are traveling

3 Copyright © Scott Bradner & Ben Gaucherin 2015

Law and Encryption, contd.



- Other export (e.g., products) is controlled
But much less than it was
Used to be that only systems with a max 40-bit key could be exported
Somehow assuming that non-US folk did not understand cryptographic technology

4

Copyright © Scott Bradner & Ben Gaucherin 2015

Key Escrow



- Save copy of private key (or symmetric key) with "trusted" third party
- For business backup/continuity
Recover from accountant getting "truck fade"
- For law enforcement
Government able to decrypt everything
In theory, only with a court order
e.g., Clipper?

5

Copyright © Scott Bradner & Ben Gaucherin 2015

Key Escrow, contd.



- Clipper
Proposed hardware & process to escrow encryption keys for all communications in U.S
Underlying assumption: no way for bad guys to get non-clipper encryption technology
Unwarranted assumption
- Constant tension between governments & citizens
- Note: if private key is escrowed
Court can not trust signed messages

6

Copyright © Scott Bradner & Ben Gaucherin 2015

Commerce & Network Security



% of people using e-commerce



- Today, too much of the world economy depends on good encryption
All web commerce inter- and intra-business communication
- Businesses need to know their communications are private from hostile powers
- Any back door can, in theory, be exploited by others

7

Copyright © Scott Bradner & Ben Gaucherin 2015

Image credits

All drawings and photos by Scott Bradner unless noted

Slide# credit

2 export cntrl conf

<http://www.americanconference.com/encryption>

England <http://www.clker.com/clipart-24410.html>

3 <https://www.bis.doc.gov/>

<http://info.chpowell.com/blog/bid/83523/Are-You-Subject-to-Export-Administration-Regulations-EAR>

4 https://www.bis.doc.gov/index.php/forms-documents/doc_download/327-flowchart-1

5 <http://top-img.com/m/mack-truck-grill>

6 <https://w2.eff.org/Misc/Graphics/sinkclipper.gif>

7 <http://mashable.com/2011/03/25/e-commerce-infographic/>

<http://arstechnica.com/security/2014/05/root-backdoor-found-in-surveillance-gear-used-by-law-enforcement/>

8

Copyright © Scott Bradner & Ben Gaucherin 2015


Encryption
Conclusion

CSCI E 45a: The Cyber World – part A

1 Copyright © Scott Bradner & Ben Gaucherin 2015

Summary

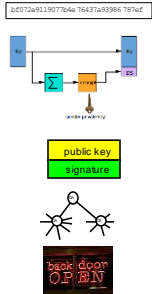
- Encryption is used to hide information in plain sight
- Symmetric encryption is strong but the keys are hard to manage
- Asymmetric encryption is also strong and takes more power but the keys are much easier to manage
- Creating sessions keys with Diffie-Helman is useful math magic



2 Copyright © Scott Bradner & Ben Gaucherin 2015

Summary, contd.

- Cryptographic hashes play a critical role and are not export controlled
- Digital signatures can prove who sent a message and tell if it has been modified
- I can use a certificate to tell you my public key
- I can get your certificate from a PKI
- There is a tension between the use of encryption and the law



3 Copyright © Scott Bradner & Ben Gaucherin 2015

Image credits

All drawings and photos by Scott Bradner unless noted

Slide# credit

2 Jefferson -
<https://franceshunter.wordpress.com/2011/09/07/thomas-jefferson-the-cryptographer/>

chip - https://en.wikipedia.org/wiki/EFF_DES_cracker

GCHQ -
https://en.wikipedia.org/wiki/File:Government_Communications_Headquarters_Logo.svg

NSA -
https://commons.wikimedia.org/wiki/File:National_Security_Agency.svg

3 backdoor -
<http://arstechnica.com/security/2014/05/root-backdoor-found-in-surveillance-gear-used-by-law-enforcement/>

4

Copyright © Scott Bradner & Ben Gaucherin 2015
