

4-26-2012

Introduction to Cryptography

Gary C. Kessler

Embry-Riddle Aeronautical University, kessleg1@erau.edu

Follow this and additional works at: <https://commons.erau.edu/db-applied-aviation>



Part of the [Communication Technology and New Media Commons](#)

Scholarly Commons Citation

Kessler, G. C. (2012). Introduction to Cryptography. , (). Retrieved from <https://commons.erau.edu/db-applied-aviation/17>

This Presentation without Video is brought to you for free and open access by the College of Aviation at Scholarly Commons. It has been accepted for inclusion in Applied Aviation Sciences - Daytona Beach by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.




Introduction to Cryptography

Gary C. Kessler

Vermont Internet Crimes Against Children Task Force
Burlington, VT


April 26, 2012




ICAC Task Force

Webinar Information

- All attendees will be muted.
- If you desire to ask a question, please use the questions section of the GoToWebinar dialogue box, typically in the upper right corner of the screen.
- Please do not raise your hand for questions we can not unmute you.
- The questions will either be answered directly by a panelist or asked to the presenter who will answer.



Office of Juvenile Justice and Delinquency Prevention




Fox Valley Technical COLLEGE
Knowledge That Works

CRIME'S AGENDA
Training and
Assistance
Program
TASK FORCE

ICAC Task Force

Webinar Information

- Poll questions may be asked during the webinar. They will be left open only a short period of time so please respond promptly.



QUICKPOLL

Which best describes your role?

Select one of the following:

- Investigator
- Forensic Examiner
- Prosecutor
- Other

Submit

Poll in progress for attendees only.

OJJDP
Office of Juvenile Justice and
Delinquency Prevention

Fox Valley
Technical
COLLEGE
Knowledge That Works

CRIME'S AGENDA
Training and
Assistance
Program
TASK FORCE

ICAC Task Force

Webinar Information

- At the conclusion of the webinar a short survey will appear. Please complete it before signing off.
- A link to view the recorded webinar and the Powerpoint slides will be provided to you via email after the webinar.

OJJDP
Office of Juvenile Justice and
Delinquency Prevention

Fox Valley
Technical
COLLEGE
Knowledge That Works

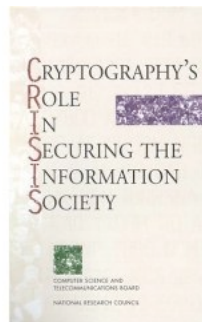
Overview

- **GOAL:** *Introduce terms, concepts, and applications*
- The role of cryptography
- Types of cryptographic algorithms
 - » Hash functions
 - » Secret key cryptography
 - » Public key cryptography
- Putting them altogether... case studies in cryptography
- Trust models
 - » Certificates
 - » Sample applications: SSL, personal certificates

© 1998-2012, Gary C. Kessler

4

The Role of Cryptography



© 1998-2012, Gary C. Kessler

5

Cryptography

- The science of writing in secret codes
 - » Dates back to 1900 B.C. in Egypt (non-standard hieroglyphics); probably appears spontaneously soon after writing is developed
- Historically, two types of cryptography:
 - » Substitution
 - » Transposition/Permutation

© 1998-2012, Gary C. Kessler

6

Substitution Ciphers

- Most famous: *Caesar's Cipher*
 - » Shift each letter to the right by 3
- Today: *Rotation 13 (ROT13)* still found on Unix systems and Usenet to hide offensive text, puzzle solutions, passwords, etc.

```
PLAIN: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
rot13: n o p q r s t u v w x y z a b c d e f g h i j k l m
```

© 1998-2012, Gary C. Kessler

7

Transposition Cipher

Columnar transposition cipher

PLAINTEXT: CRYPTO TODAY IS A LOT MORE COMPLEX THAN IT USED TO BE

CRYPTOTODA
YISALOTMOR
ECONPLEXTH
ANITUSEDTO
BENSIVEGFL

ciphertext: cyea bric neys oinp amts tlpu iool svtt
eeee mxdg dott farh olxy

© 1998-2012, Gary C. Kessler

8

Cryptography Today

- Cryptography is necessary today in telecommunications when communicating over any untrusted medium
- Digital cryptography basically comes in three varieties:
 - » Hash functions (no key)
 - » Secret key cryptography (one key)
 - » Public key cryptography (two keys)

© 1998-2012, Gary C. Kessler

9

Secure Communications

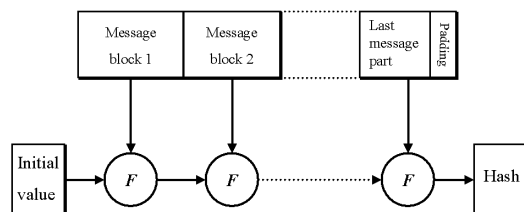
- Secure communications requires:
 - » Authentication
 - » Message integrity
 - » Non-repudiation
 - » Privacy/confidentiality

- » Key exchange

© 1998-2012, Gary C. Kessler

10

Hash Functions



© 1998-2012, Gary C. Kessler

11

Poll question #2

Hash Functions

plaintext $\xrightarrow{\text{hash function}}$ ciphertext

- No key
 - » Plaintext (and length of plaintext) is not recoverable from the ciphertext
 - » Examples: HMAC, MD2, MD4, MD5, RIPEMD-160, SHA
 - » Also called *message digests* or *one-way encryption*
- Primary use: Message integrity

Hashing: UNIX Password File

```
carol:FM5ikbQt1K052:502:100:Carol Monaghan:/home/carol:/bin/bash
alex:IqAi7Mdyg/HcQ:503:100:Alex Insley:/home/alex:/bin/bash
gary:FkJXupRyFqY4s:501:100:Gary Kessler:/home/gary:/bin/bash
todd:edGqQUAaGv7g6:506:101:Todd Pritsky:/home/todd:/bin/bash
sarah:Jbw6BwE4XoUHo:504:101:Sarah Antone:/home/schedule:/bin/bash
josh:FiH0ONcjPut1g:505:101:Joshua Kessler:/home/webroot:/bin/bash
```

Clear-text passwords is one good reason to image RAM!

SHA and MD5 Hashing

Command Prompt output:

```
C:\My Programs\forensics>copy con tyui_a.txt
a
^Z
1 file(s) copied.

C:\My Programs\forensics>copy con tyui_b.txt
b
^Z
1 file(s) copied.

C:\My Programs\forensics>sha_verify tyui_a.txt

File: tyui_a.txt
MD5      933222B19FF3E7EA5F65517EA1F7D57E
SHA      764C16AF46DD4F15EDB05ECC5595B50CBE3714EA

C:\My Programs\forensics>sha_verify tyui_b.txt

File: tyui_b.txt
MD5      C5053D4DA03789BFBC4BEE760FADE936
SHA      854D675A26DC0254A07B5725BA71242555863EB5

C:\My Programs\forensics>
```

Drive/Image Verify Results window:

General	
Name	USB_Thumb_Drive.E01
Sector count	15600

MD5 Hash	
Computed hash	80ba5fdb4805808b5c399a50d10ef167
Stored verification hash	80ba5fdb4805808b5c399a50d10ef167
Report Hash	80ba5fdb4805808b5c399a50d10ef167
Verify result	Match

SHA1 Hash	
Computed hash	6962a9ee3a515fdb11f027a95ff6f5dfcbfd0a4
Report Hash	6962a9ee3a515fdb11f027a95ff6f5dfcbfd0a4
Verify result	Match

Close

© 1998-2012, Gary C. Kessler

14

Hash Collisions

- There are 2^K possible hash values (where K = hash length) while there are an infinite number of files
 - » Since $\infty \gg 2^K$, there *will* be hash collisions
 - In fact, an infinite number of files will have the same hash!
- The problem: *Can hash collisions be forced?*
 - » What is the impact on information security?
 - » What is the impact on digital forensics?
- Solutions to collisions
 - » Use longer hashes (e.g., SHA-256)
 - » Use multiple hashes (e.g., MD5 and SHA-1)

Ref: <http://www.garykessler.net/library/crypto.htm#hash>

© 1998-2012, Gary C. Kessler

15

Sidebar: Hashing and Imaging

- There is an extraordinarily low probability that you will find two different files with the same hash
 - » ~1 in 10^{43}
- Hashes are the basis for Known File Filters when searching for images of child pornography and for P2P networks
- Hashes are only one way in which you should establish the correctness of your forensic imaging
 - » Experience, training, and valid tools (à la Fred Cohen)

What About Hashed Passwords?

The logo for md5crack.com, featuring the text 'md5crack' in a colorful, 3D-style font. The 'm' is blue, 'd' is green, '5' is yellow, 'c' is red, 'r' is blue, 'a' is yellow, and 'c' is green.

Using Google to crack passwords.

Your Results

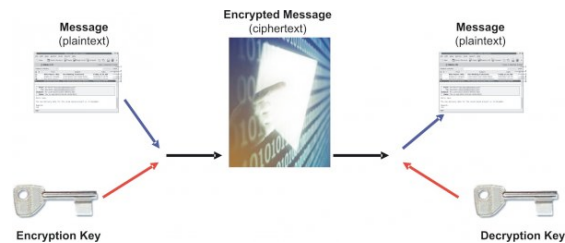
Found: c87d19bfa5f5a0c8dc75379411af75a6 = md5("kumquat")

Your Results

Sorry! Guess we couldn't find it.

[Using 2a9e402f3b2a4db8826606d527a27609, the MD5 hash of a disk drive.](#)

Secret Key Cryptography



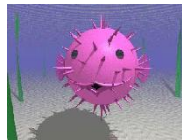
© 1998-2012, Gary C. Kessler

18

Secret Key Cryptography



- Single key (*symmetric cryptography*)
 - » Same key is used for encryption and decryption
 - » Examples: AES, DES, IDEA, 3DES, RC4, RC5, CAST, Blowfish, Twofish
- Primary use: Privacy

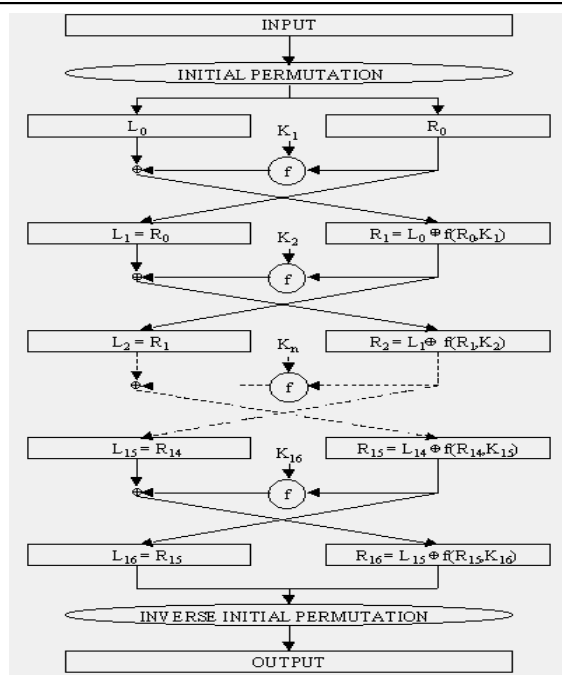


© 1998-2012, Gary C. Kessler

19

DES

- Designed to be fast in hardware, slow in software, resistant to various attacks
- Block cipher using 56-bit key and 64-bit blocks
- 56-bit key expanded to 64 bits using parity
- K_i is a 48-bit value derived from 64-bit key
- FIPS 46-2/ANS X3.92 describes entire process



© 1998-2012, Gary C. Kessler

20

A Few Words About DES...

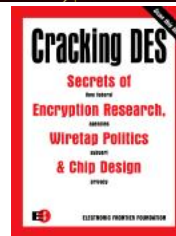
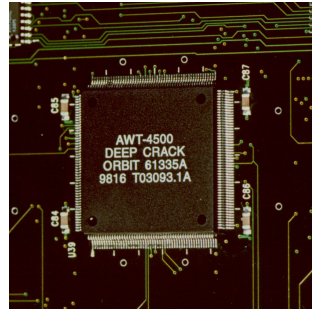
- DES introduced in 1977
 - » Proposed by IBM with 56- or 128-bit key; NSA adopted 56-bit key
- March 1998, U.S. Gov't. still claims that DES is safe from attack...
 - » July 1998, EFF introduces DES cracker designed for \$220K; can break keys in average 4.5 days
 - » For \$1M, could break DES keys in average <22 hours
- We care because DES is the most widely used crypto scheme in the financial industry!!

© 1998-2012, Gary C. Kessler

21

Breaking DES

- DES Challenge I (3/97)
 - » 84 days using thousands of computers
- DES Challenge II (1998)
 - » *distributed.net* (40 days)
 - » EFF Deep Crack (3 days)
- DES Challenge III (1/99)
 - » *distributed.net* and Deep Crack (<1 day)



© 1998-2012, Gary C. Kessler

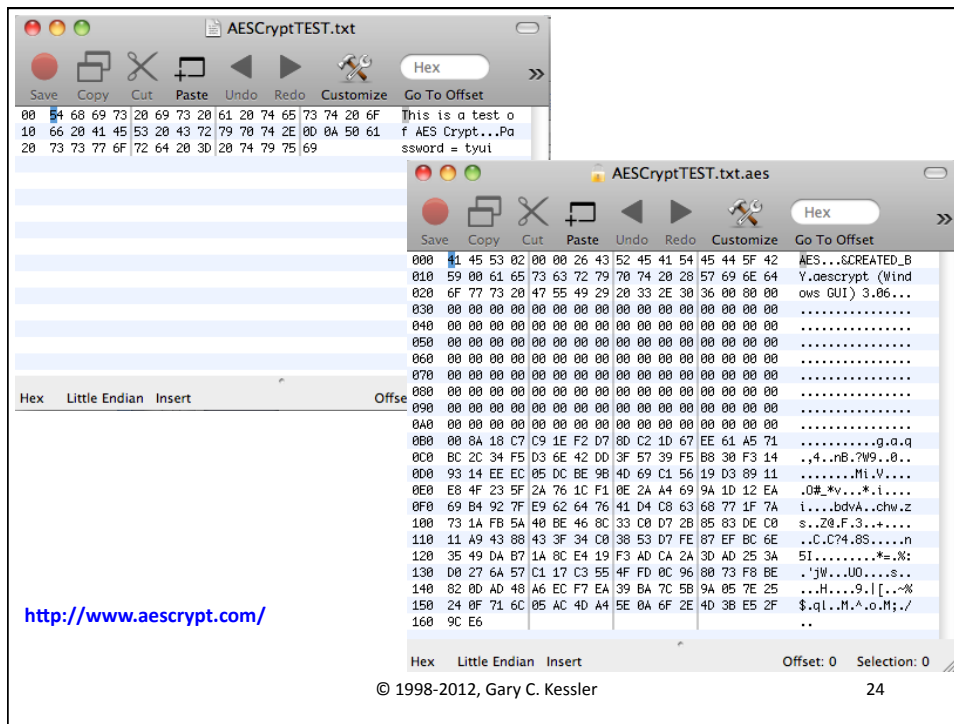
22

Advanced Encryption Standard

- NIST's next-generation SKC
 - » Open process
 - » International "competition"
 - Process started 1997, decision 2001
- Rijndael
 - » Employs 128-, 192-, or 256-bit key on a 128-, 192-, or 256-bit block
 - AES only uses a 128-bit block size
 - » Selection criteria included general security features, security implementation, software performance, smart card performance, hardware performance, and design features

© 1998-2012, Gary C. Kessler

23



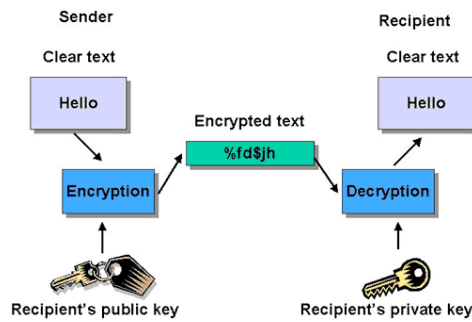
Key Length and SKC

Attacker	Budget	Tool	Time Per Recovered Key		Key Length For Protection In Late-1995
			40-bit	56-bit	
Pedestrian hacker	Tiny	PC	1 week	Never	45
			5 hours	38 years	50
Small business	\$10K	FPGA	12 min.	18 mon.	55
Corporate Dept.	\$300K	FPGA	24 sec.	19 days	60
		ASIC	0.18 sec.	3 hours	
Big Company	\$10M	FPGA	7 sec.	13 hours	70
		ASIC	5 ms	6 min.	
Government	\$300M	ASIC	0.2 ms	12 sec.	75

ASIC = Application-specific integrated circuit
 FPGA = Field programmable gate array

Source: Blaze, et al., 1996

Public Key Cryptography



© 1998-2012, Gary C. Kessler

26

Public Key Cryptography



- Two keys (*asymmetric cryptography*)
 - » One key is used for encryption, the other for decryption
 - » The two keys are related mathematically but knowledge of one key does not easily yield knowledge of the other key
 - » Examples: RSA, DSA, Diffie-Hellman, ECC, ElGamal
- Primary uses: Authentication, non-repudiation, key exchange (but invented for bulk encryption)

© 1998-2012, Gary C. Kessler

27

PKC

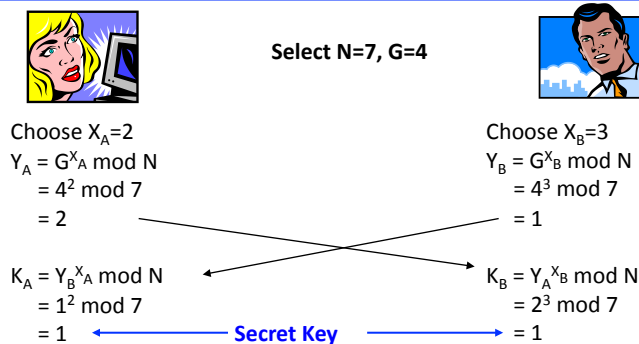
- All PKC based on some mathematical function that is easy but where the inverse is hard
 - » E.g., exponentiation vs. logarithms, multiplication vs. factorization
- Actual *invention* is unclear...
 - » NSA, 1966 (no proof)
 - » U.K. Gov't. Communication Headquarters, 1969 (classified until 1990s)
 - » Merkle's Puzzles (claim 1974, pub. 1978)
 - » Diffie & Hellman (pub. 1976)

© 1998-2012, Gary C. Kessler

28

Diffie-Hellman Key Exchange

Alice and Bob agree on the value of a large prime number, N and a generator, G . Each calculates a private key (X) and public key (Y). The secret key (K) is derived from X and the other person's Y .



© 1998-2012, Gary C. Kessler

29

RSA Mathematics

- Create private/public key pair:
 - » Choose 2 primes, p & q
 - » Modulus $n = pq$
 - » Select public exponent e , relatively prime to $(p-1)(q-1)$
 - » Calculate private exponent $d = (ed-1)/[(p-1)(q-1)]$
- To encrypt message M with public key:
 - » $C = M^e \text{ mod } n$
- To decrypt ciphertext C with private key:
 - » $M = C^d \text{ mod } n$
- Of course, either key can be used first...

© 1998-2012, Gary C. Kessler

30

RSA Example

- Select $p=3$, $q=5$
- $n = pq = 15$
- Choose $e=11$, relatively prime to $(p-1)(q-1) = 8$
- $(11d-1)/8$ must be an integer; choose $d=3$
- $M = 8384$ (0x8384)
- Encrypt
 - » Public key value is $(e,n) = (11,15)$
 - » $C_i = M_i^{11} \text{ mod } 15$
 - » $C = 0x2c24$
- Decrypt
 - » Private key value is $(d,n) = (3,15)$
 - » $M_i = C_i^3 \text{ mod } 15$
 - » $M = 0x8384$

© 1998-2012, Gary C. Kessler

31

RSA Application



$\text{ciphertext} = \text{PVT}_{\text{ALICE}}(\text{message})$

Alice can sign messages by encrypting with her own private key; this **authenticates** that she sent the message

$\text{ciphertext} = \text{PUB}_{\text{BOB}}(\text{message})$

Alice can ensure that only Bob can read a message by encrypting with his public key; this provides **privacy** and proves that Bob was the intended receiver.

Elliptic Curve Cryptography

- First described in 1985 by two independent teams
- Uses logarithms and hard-to-solve problems that fall on an elliptic curve
 - » Because problems are harder to solve than factoring, smaller keys yield better protection and faster processing than RSA
- Current uses: Smart cards, mobile devices, PDAs
 - » Primary vendor is Certicom

The Elliptic Curve

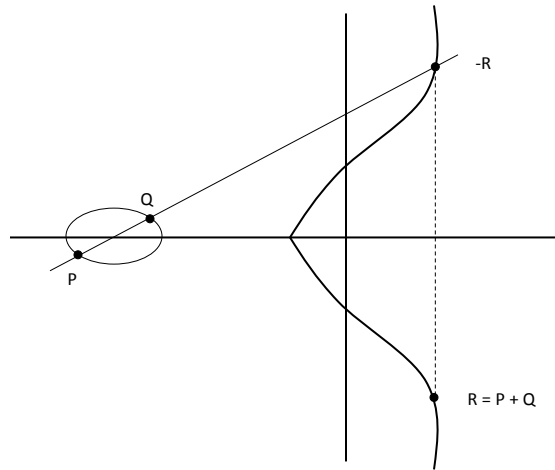
The problem: Given two points, P and Q, on an elliptic curve, find integer $i \ni P=iQ$.

- Public key = iQ
- Private key = i

Elliptic curve consists of the set of real numbers (x,y) that satisfy:

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

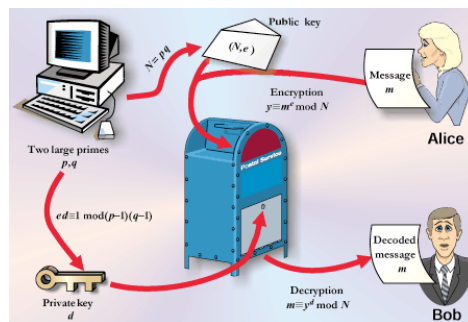
Small changes in a and b can make major changes in the shape of the curve and, therefore, the set of (x,y) points that satisfy the equation.



© 1998-2012, Gary C. Kessler

34

Sample Cryptosystems

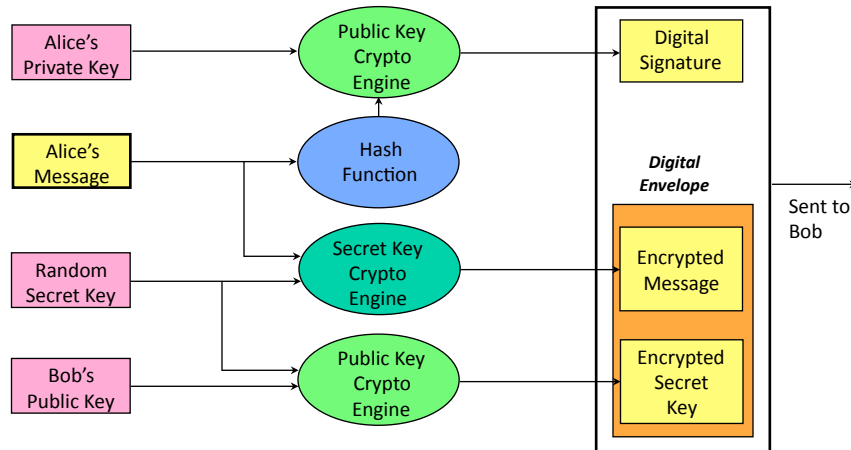


© 1998-2012, Gary C. Kessler

35

Poll question #3

Sample Hybrid Cryptosystem



© 1998-2012, Gary C. Kessler

36

Case Study: PGP Signatures

```
-----BEGIN PGP SIGNED MESSAGE-----
```

```
Hash: SHA1
```

```
Hi Carol.
```

```
What was that pithy Groucho Marx quote?
```

```
/kess
```

```
-----BEGIN PGP SIGNATURE-----
```

```
Version: PGP for Personal Privacy 5.0
```

```
Charset: noconv
```

```
iQA/AwUBNFUdO5W0cz5SFtuEEQJx/ACaAgR97+vvDU6XWELV/GANjAAgBtUANjG3
```

```
Sdfw2JgmZiOLNjFe7jP0Y8/M
```

```
=jUAU
```

```
-----END PGP SIGNATURE-----
```

© 1998-2012, Gary C. Kessler

37

Case Study: PGP Encryption

-----BEGIN PGP MESSAGE-----
Version: PGP for Personal Privacy 5.0
MessageID: DAdVB3wzpBr3YRunZwYvhK5gBKXOb/m

```
qANQR1DBwU4D/T1T68XXuiUQCADfj2o4b4aFYBcWumA7hr1Wvz9rbv2BR6WbEUsy
ZBIEFTjyqCd96qF38sp9IQiJIKlNaZfx2GLRWikPZwchUXxB+AA5+1qsG/ELBvRa
c9XefaYpbbAZ6z6LkOQ+eEOXASe7aEEPfdxvZZT37dVyyixuBBRYNLN8Bphdr2zv
z/9Ak4/OLnLiJRk05/2UNE5Z0a+31cvITMmfGajvRhkXqocavPOKiin3hv7+Vx88
uLLem2/fQHZhgCQvkqZVqXx8SmNw5gzuvwV1WHj9muDGBY0MkjiZIRI7azWnoU9
3KCnmpR60VO4rDRAS5uG19fioSvze+q8XqxubaNsgdKkoD+tB/4u4c4tznLfw1L2
YBS+dzFDw5desMFSO7JkecAS4NB9jAu9K+f7PTAsesCBNETDd49BTOFFTWwAvfE
gLYcPrcn4s3EriUgvl3OzPR4P1chNu6sa3ZJkTBbriDoA3VpngG3hxqfNy0lqAka
mJJuQ530b9ThaFH8YcE/VqUFdw+bQtrAJ6NpjIxi/x0FfOIInhC/bBw7pDLXBfNaX
HdlLQRPQdrmnWskKznOSarxq4GjprTQo4hpCRJJ5aU7tZO9HPTZXFG6iRIT0wa47
AR5nvkEkoIAjW5HaDKiJriuWldtN4OXecWvxFsJR32ebz76U8aLpAK87GZEyTzBx
dV+lH0hwYt/y1cZQ/E5USePP4oKWF4uqquPee1OpeFMB04CvuGyhZXD/18Ft/53Y
WIebvdiCqsOoabK3jEfdGEXce63zDI0=
=MpRf
-----END PGP MESSAGE-----
```

© 1998-2012, Gary C. Kessler

38

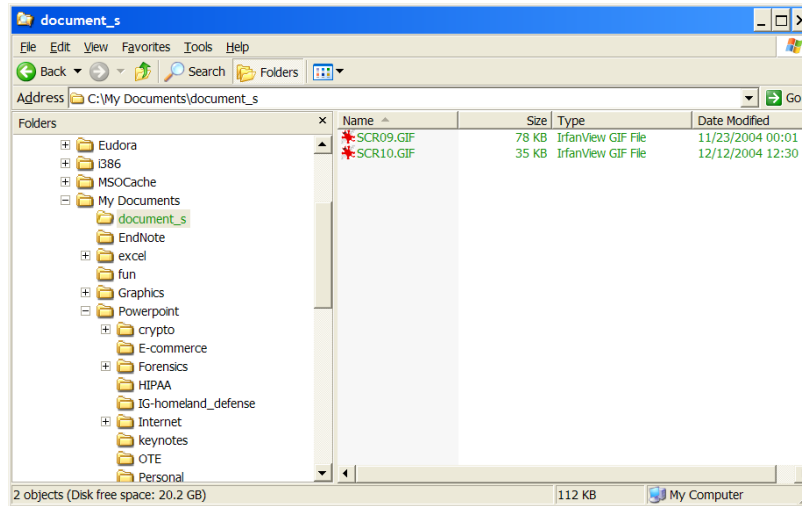
Case Study: Windows 2000/XP EFS

- Encrypting File System (EFS)
 - » When the file is saved to disk:
 - Random file encryption key (FEK) is created
 - File contents encrypted using FEK and encryption algorithm
 - Windows 2000 and XP default to DESX; XP also supports 3DES
 - Windows XP SP1, Server 2003, Vista, and Server 2008 default to AES; also support DESX and 3DES
 - Windows 7 and Server 2008 R2 default to AES, SHA, and ECC; also support DESX and 3DES
 - FEK stored with file, encrypted with user's RSA public key (and, optionally, recovery agent's RSA public key)
 - » When the file is opened:
 - FEK recovered for decryption using RSA private key, which can be stored on external floppy disk or smart card
 - If private key lost, files may be accessed using RA's private key
 - Key tied to username prior to Win XP SP2; now uses user password
 - » **A pre-encryption backup file is *deleted* after encryption**

© 1998-2012, Gary C. Kessler

39

EFS



© 1998-2012, Gary C. Kessler

40

The cipher command (Windows)

```
C:\> cipher /u /n
```

```
Encrypted File(s) on your system:
```

```
C:\My Documents\document_s\SCR09.GIF
C:\My Documents\document_s\SCR10.GIF
C:\My Documents\Word\GKS\phish\1_real1.png
C:\My Documents\Word\GKS\phish\2_real2.png
C:\My Documents\Word\GKS\phish\3_bogus1.png
C:\My Documents\Word\GKS\phish\4_bogus2.png
C:\My Documents\Word\GKS\phish\5_has_data.png
C:\My Documents\Word\GKS\phish\62.193.219.166.html.txt
C:\My Documents\Word\GKS\phish\6_after_submit.png
C:\My Documents\Word\GKS\phish\7_tcp_stream.png
C:\My Documents\Word\GKS\phish\commnatlbank.acp
C:\My Documents\Word\GKS\phish\Re Help with investigation.txt
```

```
C:\>
```

© 1998-2012, Gary C. Kessler

41

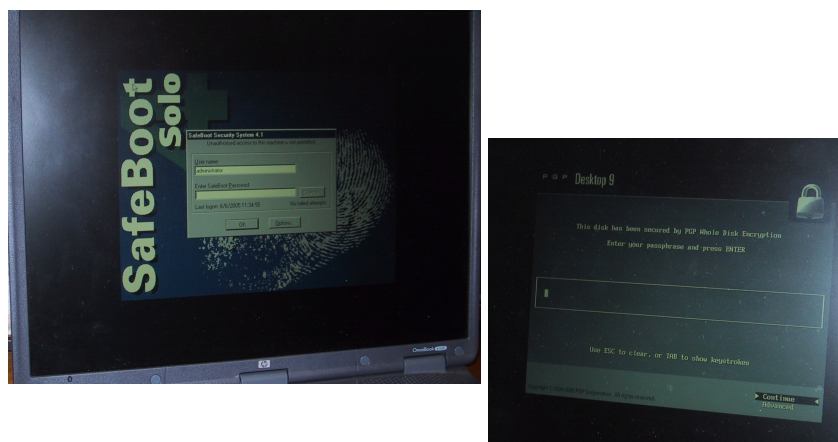
Whole Disk Encryption (WDE)

- WDE encrypts entire drive
 - » WDE modifies boot sector 0 to go to alternate loader
 - » Files are available to user after a password is entered when logging on
 - Files are encrypted when viewed by forensics software if disk drive is powered down
- Some disk encryption software
 - » Windows Vista
 - Full volume encryption; Vista encrypts data partition, but not boot partition
 - » PGP 9.0
 - » Pointsec
 - » Safeboot
 - » Utimaco
- A case for live forensics...

© 1998-2012, Gary C. Kessler

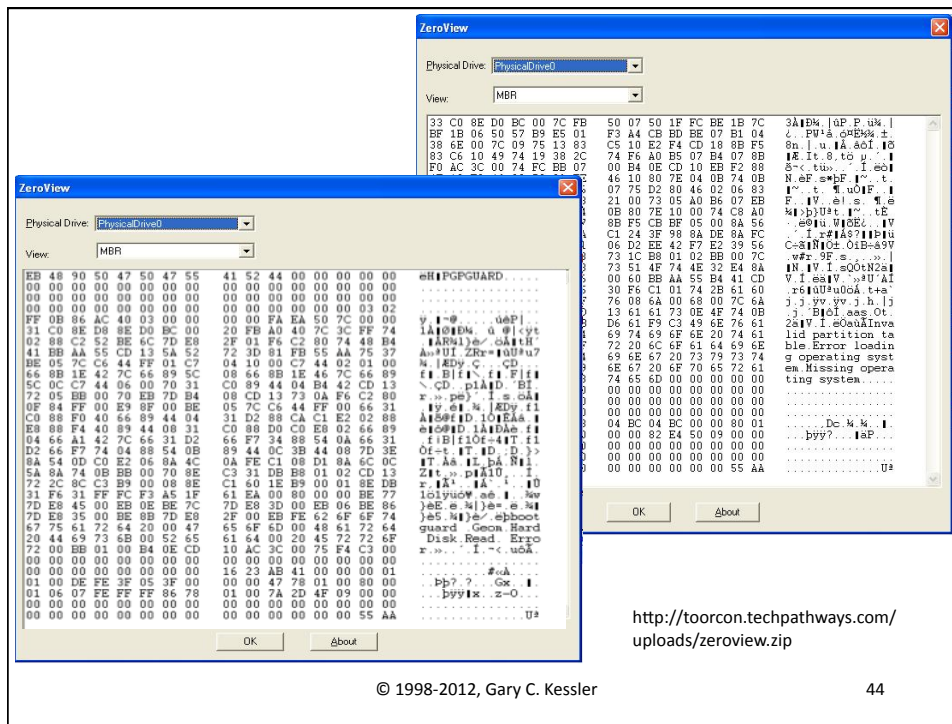
42

Pre-Boot Logon



© 1998-2012, Gary C. Kessler

43



Windows Vista and Windows 7

- Vista and Win7 encryption can cause forensics examiners some problems...
 - » Hardware-enabled full-volume encryption
 - » Windows Vista Enterprise and Ultimate
 - Encryption **not** on by default
 - » Use BitLocker drive encryption tied to Trusted Platform Module (TPM) chip or USB flash drive for key storage

Other Crypto Schemes

- TrueCrypt
 - » Open source encryption for Windows, MacOS, or Linux
 - » Virtual encrypted disk using AES, Serpent, or Twofish
 - » Can create hidden encrypted volume
 - » First released in 2004
- FileVault
 - » File encryption for Macs, using AES
 - » Password derived from user's login password

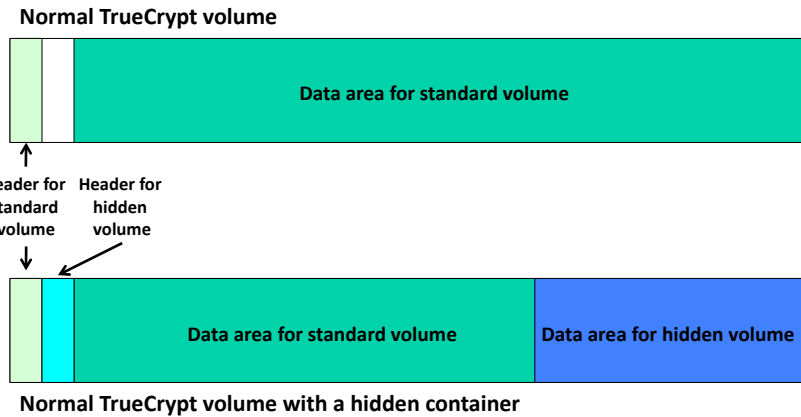


The screenshot shows the TrueCrypt application interface. The main window has a menu bar (Volumes, System, Favorites, Tools, Settings, Help) and a toolbar (Homepage). Below the menu bar is a table with columns: Drive, Volume, Size, Encryption algorithm, and Type. The 'Volume' column shows 'E:\James'. Below the table are buttons for 'Create Volume', 'Volume Properties...', and 'Wipe Cache'. A secondary window is open, showing a table with columns: Slot, Volume, Size, Mount Directory, and Type. The table has 12 slots, with slot 7 selected and highlighted in red. Below the table are buttons for 'Create Volume', 'Volume Properties...', and 'Wipe Cache'. A 'Volume Properties' dialog box is open, showing a 'Volume' field with the value '/Volumes/JAMESTC/James', a 'Select File...' button, and a checked 'Never save history' checkbox. Below the dialog box are buttons for 'Dismount', 'Mount All Devices', 'Dismount All', and 'Close'.

Slot	Volume	Size	Mount Directory	Type
1				
2				
3				
4				
5				
6				
7	/Volumes/JAMESTC/James	1.5 GB	/Volumes/JIMMY	Normal
8				
9				
10				
11				
12				

© 1998-2012, Gary C. Kessler

Plausible Deniability



The screenshot shows a Mac OS X desktop environment. A terminal window is open, displaying the output of a command to list the contents of a VM image. The output shows a list of files and directories, including a hidden container. A blue box highlights the text "Contents of /var/vm/sleepimage". The desktop background is a purple and blue space-themed wallpaper. The dock at the bottom contains various application icons. The top of the screen shows the menu bar and system status icons.

```
Terminal - bash - 111x47
bp1ist00
Zpassphrase|disgracedface
-file://localhost/Users/.Face/Face.sparsebundle
% ?
Sean-Cavanaughs-MacBook-Pro:desktop Face$ cat After\ Secure\ VM.txt | grep -n -C 3 disgracedface
751577--face
751578--managedUser
751579--password
751580--disgracedface
751581--shell
751582--/bin/basoulndnmount
751583--username
--
797081--verbose
797082--sinur
797083--bp1ist00
797084--Zpassphrase|disgracedface
797085--file://localhost/Users/.Face/Face.sparsebundle
797086--% ?
797087--p7XP
Sean-Cavanaughs-MacBook-Pro:desktop Face$ cat Before\ Secure\ VM.txt | grep -n -C 3 disgracedface
999442--face
999443--managedUser
999444--password
999445--disgracedface
999446--shell
999447--/bin/basoulndnmount
999448--username
--
1090863--verbose
1090864--o.in-ur
1090865--bp1ist00
1090866--Zpassphrase|disgracedface
1090867--file://localhost/Users/.Face/Face.sparsebundle
1090868--% ?
1090869--p0PP
--
1165236--face
1165237--
1165238--managedUser
1165239--password
1165240--disgracedface
1165241--shell
1165242--/bin/bash
1165242--shoulndnmount
Sean-Cavanaughs-MacBook-Pro:desktop Face$
```

EDD

- JADsoftware's Encrypted Disk Detector
 - » Tests for BitLocker, PGP, and TrueCrypt encrypted drives and volumes (partitions)
 - » http://www.jadsoftware.com/go/?page_id=167

© 1998-2012, Gary C. Kessler

50

EDD D:\EDD.exe

```
Encrypted Disk Detector v1.0
Copyright (c) 2009 Jad Saliba
http://www.jadsoftware.com
// This program comes with no guarantee or warranty,
all risk is assumed by the user. //

* Checking physical drives on system... *
PhysicalDrive0, Partition 1 --- OEM ID: NTFS
PhysicalDrive1, Partition 1 --- OEM ID: NTFS
* Completed checking physical drives on system. *

No TrueCrypt, PGP, or Bitlocker encrypted volumes detectable by EDD were found.
Press any key to continue...
(Use 'DiskID /batch' to bypass this prompt next time)
```

Two physical drives, both with NTFS;
no encrypted volume.

EDD D:\EDD.exe

```
Encrypted Disk Detector v1.0
Copyright (c) 2009 Jad Saliba
http://www.jadsoftware.com
// This program comes with no guarantee or warranty,
all risk is assumed by the user. //

* Checking physical drives on system... *
PhysicalDrive0, Partition 1 --- OEM ID: NTFS
PhysicalDrive1, Partition 1 --- OEM ID: r06
PhysicalDrive1, Partition 1 might be an encrypted volume,
or contains a damaged boot sector.
* Completed checking physical drives on system. *

Press any key to continue...
(Use 'DiskID /batch' to bypass this prompt next time)
```

Two physical drives. The first volume
uses NTFS and is not encrypted. The
second volume is detected as possibly
encrypted (in fact, it is a TrueCrypt
volume).

Screen shots from <http://www.jadsoftware.com/home/edd.htm>

© 1998-2012, Gary C. Kessler

51

Secure Communication Protocols

- Secure MIME (S/MIME)
- Secure Sockets Layer (SSL)
 - » https, ftps, pops, smtps, ...
- Secure Electronic Transactions (SET)
- Secure HTTP (S-HTTP)
- Transaction Internet Protocol (TIP)
- Simple Authentication and Security Layer (SASL)
- Pretty Good Privacy (PGP)
- IP Security Protocol (IPsec)
- Kerberos
- Server Gated Cryptography (SGC)
- Transport Layer Security (TLS)
- Secure Shell (SSH)
- Authenticated POP (APOP)

Do *not* trust “secret” cryptographic protocols (e.g., Skipjack!). The safety is in the choice (and length) of the *key*, not the secrecy of the *algorithm* (Kerckhoffs' Principle, 1883).

Sidebar: Other Considerations

- Poor implementation and/or management of keys works in the investigator's favor!
 - » Unprotected files on the computer with usernames/passwords
 - » Never underestimate the value of a good interview
 - Particularly in the very early stages on scene
- Think carefully before pulling the plug!!
 - » Encryption is a case for live imaging in the field

Trust in Cryptosystems



© 1998-2012, Gary C. Kessler

54

Poll question #4

Trust Models

- When using cryptography, how can you trust the entity that gives you a key?
 - » PGP Web of trust
 - » Kerberos trusted server and SKC for *a priori* relationships
 - » PKI trusted third parties and PKC for anyone-to-anyone communication

© 1998-2012, Gary C. Kessler

55

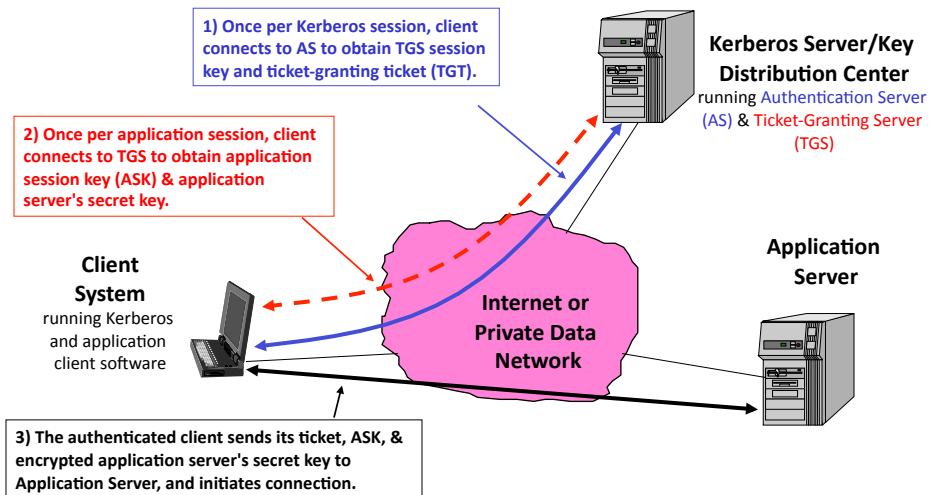
PGP Web of Trust

Keys	Validity	Trust	Size	Description
Gary C. Kessler <kumquat@sover.net>	●	██████████	2048/1024	DH/DSS key pair
George Bakos <alpinista@bigfoot.com>	●	██████████	3072/1024	DH/DSS public key
George Bakos <alpinista@bigfoot.com>	●	██████████	768	RSA public key
Michael Schirling <mschirli@dps.state.vt.us>	●	██████████	2048/1024	DH/DSS public key
Mich Kabay <mkabay@compuserve.com>	●	██████████	768	RSA public key
Mich Kabay <mkabay@compuserve.com>	●	██████████		User ID
Mich Kabay <mkabay@compuserve.c...>	●	██████████		RSA exportable signature
Robert G. Moskowitz <rgm@icsa.net>	●	██████████		RSA trusted introducer signature
Gary C. Kessler <kumquat@sover.net>	●	██████████		DSS exportable signature
Microsoft Security Response Center <secure@...>	●	██████████	2048	RSA public key
N. Todd Pritsky <todd@hill.com>	●	██████████	2048/1024	DH/DSS public key
N. Todd Pritsky <todd@hill.com>	●	██████████		User ID
N. Todd Pritsky <todd@hill.com>	●	██████████		DSS exportable signature
Gary C. Kessler <kumquat@sover.net>	●	██████████		DSS exportable signature
PGP Support Key DSS <pgpsupport@pgp.com>	●	██████████	1024/1024	DH/DSS public key
Pretty Good Privacy, Inc. Corporate Key	●	██████████	2048/1024	DH/DSS public key
Robert G. Moskowitz <rgm@icsa.net>	●	██████████	1024	RSA public key
The SANS Institute <sans@sans.org>	●	██████████	1024/1024	DH/DSS public key
The SANS Institute <sans@sans.org>	●	██████████		User ID
The SANS Institute <sans@sans.org>	●	██████████		DSS exportable signature
Unknown Signer, Key ID is 0a7C6E575F	●	██████████		DSS exportable signature

© 1998-2012, Gary C. Kessler

56

Kerberos



© 1998-2012, Gary C. Kessler

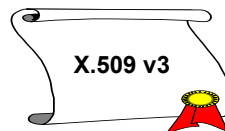
57

Public Key Infrastructure

- How can a party's public key be found if not known locally? Where is the key stored?
- How does a recipient verify that a public key really belongs to the sender *and* that it is being used for a legitimate purpose?
- When does a public key expire?
- How can a key be revoked in case of loss or compromise?

Certificates

- *Certificates* bind a public key to an individual, position, or other entity, and provide
 - » Identification
 - » Expiration date
 - » Issuing authority
 - » Serial number
 - » Policies about how the user was identified
 - » Limitations on how the key may be used



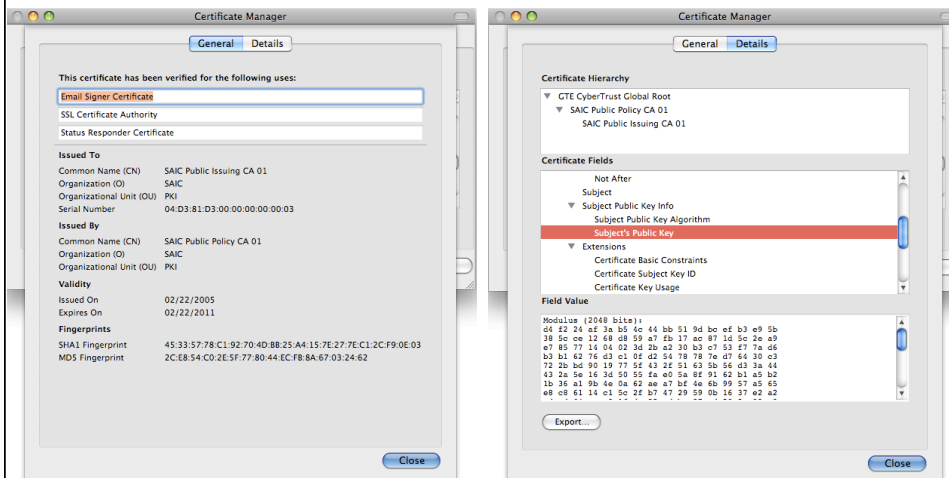
Certificates in Real-Life...

- Certificates identify us, what we are allowed to do, issuer, validity period, etc.
 - » Driver's license: Name, DOB, address, type of vehicle, issuing state, valid period, serial number, photo(?), organ donation(?)
 - » Credit card: Name, serial number, valid period, issuer
 - » SCUBA certification: Name, DOB, serial number, level of training, certification date, instructor, issuing agency, photo(?)

© 1998-2012, Gary C. Kessler

60

Sample Browser Certificate



© 1998-2012, Gary C. Kessler

61

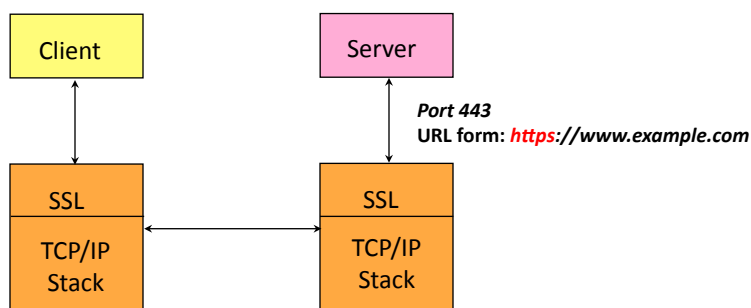
Secure Sockets Layer

- Originally developed by Netscape Communications
 - » SSL v2.0 (**deprecated**) and v3.0 (**weak**)
 - » Transport Layer Security (TLS) v1.0 ~ "SSL v3.1" (RFC 2246)
 - **Theoretical vulnerability described in 2002 made practical in 2011!**
 - TLS v1.1 (RFC 4346) and TLS v1.2 (RFC 5246)
- Provides privacy, integrity, client/server authentication
- Application-independent
 - » Can be used with HTTP, Telnet, FTP, NNTP, IMAP, POP3 over TCP
 - » Datagram TLS (v1.2, RFC 6347) operates over UDP
- Two main protocols
 - » SSL Handshake Protocol (parameter negotiation)
 - » SSL Record Protocol (data transfer)

© 1998-2012, Gary C. Kessler

62

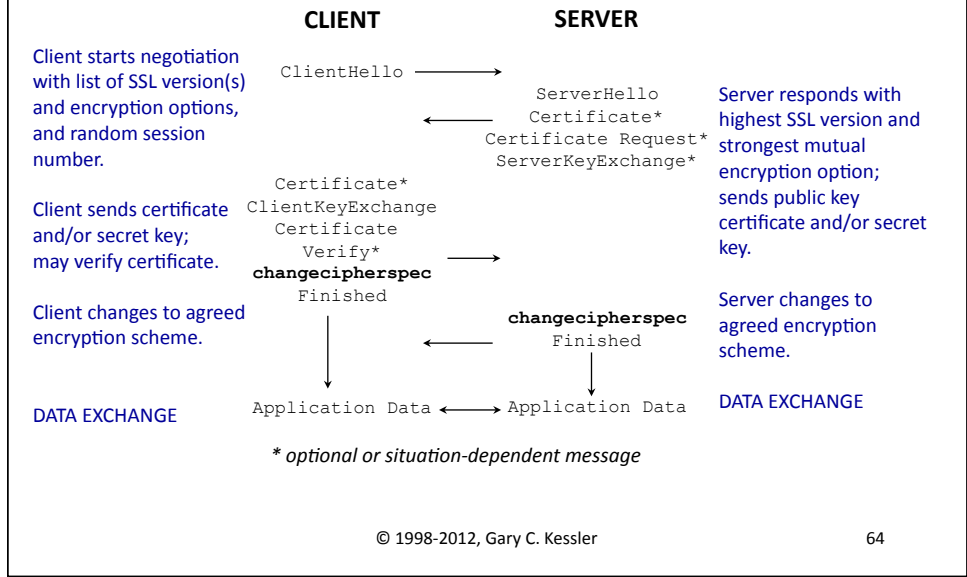
HTTP Over SSL



© 1998-2012, Gary C. Kessler

63

SSL Handshake Protocol



Limitations of PKI

- A digital signature does **not** prove that Alice signed a message, but that her private key did
 - » Good cryptographic algorithms can be bypassed by viruses, malicious code, abuse/misuse by users, and other real-world events
- Users do not generally check the source or validity of received certificates

Summary and Closure!



© 1998-2012, Gary C. Kessler

66

Detecting Encryption

- Cryptography provides *secret* communication but not necessarily *hidden*
 - » Use of crypto does not form a covert communications channel
 - » Encrypted messages and files can be detected by a third party

© 1998-2012, Gary C. Kessler

67


Example Detection Statistics (FTK)

- Most encryption detection schemes are testing for randomness; high randomness suggests use of encryption
 - » *Arithmetic Mean*: Calculated by summing all of the bytes in a file and dividing by the file length; if random, the value should be ~ 1.75 .
 - » *Chi-Squared Error Percent*: This distribution is calculated for a byte stream in a file; the value indicates how frequently a truly random number would exceed the calculated value.
 - » *Entropy*: Describes the information density (per Shannon) of a file in bits/character; as entropy $\rightarrow 8$, there is more randomness.
 - » *MCPI Error Percent*: The Monte Carlo algorithm uses statistical techniques to approximate the value of π ; A high error rate implies more randomness.
 - » *Serial Correlation Coefficient*: Indicates the amount to which each byte is an e-mail relies on the previous byte. A value close to 0 indicates randomness.

Crypto Attack Methods

- Password guessing
- Known plaintext
- Chosen plaintext
- Known ciphertext
- Dictionary attack
- Brute force attack
- Side channel attacks

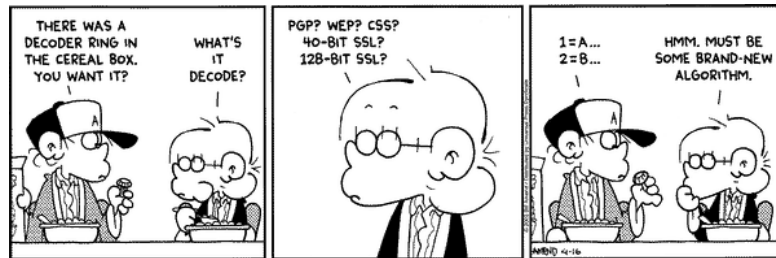
Crypto Breaking Tools

- Rainbow tables
- *distributed.net* 
- Rack attack
- Passware
 - » Password Recovery Bundle
- ElcomSoft
 - » Password Recovery Bundle
- AccessData
 - » Password Recovery Toolkit (PRTK)
 - » Distributed Network Attack (DNA)
 - » Portable Office Rainbow Table (PORT)

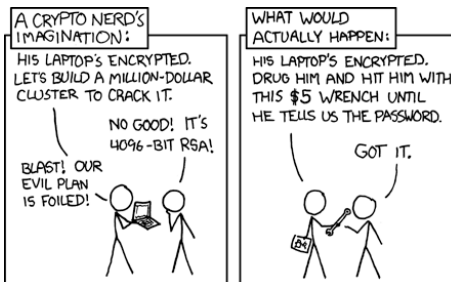
© 1998-2012, Gary C. Kessler

70

A Little Crypto Humor...



Foxtrot, 4/16/2003



<http://xkcd.com/538/>, Feb. 2009

© 1998-2012, Gary C. Kessler

71

Additional References

- *Cryptography Engineering: Design Principles and Practical Applications*, Ferguson, Schneier, & Kohno
- *The Code Book: The Evolution of Secrecy from Mary Queen of Scots to Quantum Cryptography*, Singh
- *Classical and Contemporary Cryptology*, Spillman
- *Malicious Cryptography*, Young & Yung

- Counterpane (www.counterpane.com)
- Cryptography Research (www.cryptography.com)
- RSA's Crypto FAQ (www.rsa.com/rsalabs/node.asp?id=2152)
- GCK's crypto overview paper (www.garykessler.net/library/crypto.html) and crypto links (www.garykessler.net/library/securityurl.html#crypto)

© 1998-2012, Gary C. Kessler

72

Author Contact Information

Gary C. Kessler, Ph.D., CCE, CISSP
GARY KESSLER ASSOCIATES
2 Southwind Drive
Burlington, VT 05401

mobile: +1 802-238-8913
e-mail: gck@garykessler.net
gkessler@bpdvt.org
Skype: [gary.c.kessler](https://www.skype.com/user/gary.c.kessler)

<http://www.garykessler.net>
<http://www.vtinternetcrimes.org>



© 1998-2012, Gary C. Kessler

73

Acronyms and Abbreviations

3DES	Triple DES	ISP	Internet service provider
AES	Advanced Encryption Standard (NIST)	ITU-T	International Telecommunication Union, Telecommunication Standardization Sector
ANS	American National Standard	IV	Initial vector
CA	Certificate authority	MAC	Message authentication code
CPS	Certification practice statement	MD2/4/5	Message Digest 2, 4, & 5
CRL	Certificate Revocation List	MIPS	Millions of instructions per second
DES	Data Encryption Standard	MS	Microsoft
DOB	Date of birth	NIST	National Institute of Standards and Technology
DoS	Denial of service	NNTP	Network News Transport Protocol (IETF)
DSA	Digital Signature Algorithm (NIST)	NSA	National Security Agency
ECC	Elliptic Curve Cryptography	OS	Operating system
EFF	Electronic Frontier Foundation	PDA	Personal digital assistant
EFS	Encrypting File System (W2K)	PGP	Pretty Good Privacy
FIPS	Federal Information Processing Standard	PKC	Public key cryptography
FTP	File Transfer Protocol (IETF)	PKI	Public key infrastructure
HMAC	Hashed message authentication code	POP	Post Office Protocol (IETF)
HTTP	Hypertext Transfer Protocol (IETF)	RA	Registration Authority
https	HTTP over SSL	RC2/4/5	Rivest Cipher (or Ron's Code) 2, 4, and 5
IDEA	International Data Encryption Algorithm	RFC	Request for Comments (IETF)
IE	Internet Explorer (MS)	RSA	Rivest, Shamir, Adleman
IETF	Internet Engineering Task Force		
IMAP	Internet Message Access Protocol (IETF)		

© 1998-2012, Gary C. Kessler

74

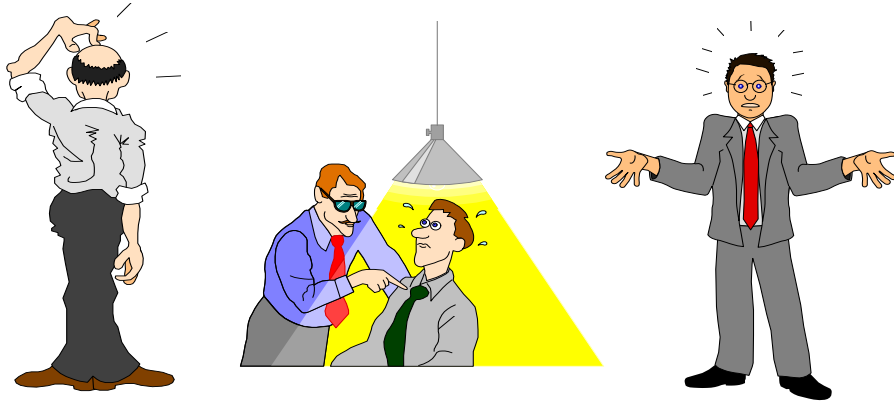
Acronyms and Abbreviations (cont.)

SCUBA	Self-contained underwater breathing apparatus
SHA	Secure Hash Algorithm (NIST)
SKC	Secret-key cryptography
SSL	Secure Sockets Layer (Netscape)
TCP	Transmission Control Protocol (IETF)
TLS	Transport Layer Security (IETF)
URL	Uniform Resource Locator
WDE	Whole disk encryption
W2K	Windows 2000 (MS)
XOR	Exclusive OR

© 1998-2012, Gary C. Kessler

75

Questions? Comments? Queries?



© 1998-2012, Gary C. Kessler

76