Introduction to Cryptography

Gary C. Kessler

Vermont Internet Crimes Against Children Task Force
Burlington, VT

April 26, 2012

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.
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.

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.
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
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

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.

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
Transposition Cipher

Columnar transposition cipher

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

CRYPTOTODA
YISALOTMOR
ECOMPLEXTH
ANITUSEDTO
BENSIVEGFL

ciphertext: cyea bric neys oinp amts tlpuiool svtteeo mxdg dott farh olx

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)
Secure Communications

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

Hash Functions

© 1998-2012, Gary C. Kessler
Hash Functions

- **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:LqAi7Mdyg/HcQ:503:100:Alex Insley:/home/alex:/bin/bash
gary:FkJXupRyPqY4s:501:100:Gary Kessler:/home/gary:/bin/bash
todd:edGqQHAgv7g6:506:101:Todd Pritsky:/home/todd:/bin/bash
sarah:Jbw6BwE4XoUHo:504:101:Sarah Antone:/home/schedule:/bin/bash
josh:FhH0ONcJPutlg:505:101:Joshua Kessler:/home/webroot:/bin/bash

*Clear-text passwords is one good reason to image RAM!*
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.html#hash
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?

Using 2a9e402f3b2a4db8826606d527a27609, the MD5 hash of a disk drive.

© 1998-2012, Gary C. Kessler
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
**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

---

**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!!**
## 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)

## 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
Key Length and SKC

<table>
<thead>
<tr>
<th>Attacker</th>
<th>Budget</th>
<th>Tool</th>
<th>Time Per Recovered Key</th>
<th>Key Length For Protection In Late-1995</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pedestrian hacker</td>
<td>Tiny $400</td>
<td>PC</td>
<td>1 week</td>
<td>45</td>
</tr>
<tr>
<td>Small business</td>
<td>$10K</td>
<td>FPGA</td>
<td>12 min.</td>
<td>50</td>
</tr>
<tr>
<td>Corporate Dept.</td>
<td>$300K</td>
<td>FPGA</td>
<td>24 sec. 0.18 sec.</td>
<td>60</td>
</tr>
<tr>
<td>Big Company</td>
<td>$300M</td>
<td>ASIC</td>
<td>7 sec. 5 ms</td>
<td>70</td>
</tr>
<tr>
<td>Government</td>
<td>$300M</td>
<td>ASIC</td>
<td>0.2 ms 0.12 sec.</td>
<td>75</td>
</tr>
</tbody>
</table>

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

Source: Blaze, et al., 1996

© 1998-2012, Gary C. Kessler

http://www.aescrypt.com/
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)
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)

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 $Y$.

Select $N=7$, $G=4$

Choose $X_A=2$

$Y_A = G^{X_A} \mod N$

$= 4^2 \mod 7$

$= 2$

$K_A = Y_B^{X_A} \mod N$

$= 1^2 \mod 7$

$= 1$

Choose $X_B=3$

$Y_B = G^{X_B} \mod N$

$= 4^3 \mod 7$

$= 1$

$K_B = Y_A^{X_B} \mod N$

$= 2^3 \mod 7$

$= 1$

Secret Key
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 mod n
- To decrypt ciphertext C with private key:
  - M = C^d mod n
- Of course, either key can be used first...

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} mod 15
  - C = 0x2c24
- Decrypt
  - Private key value is (d,n) = (3,15)
  - M_i = C_i^3 mod 15
  - M = 0x8384
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 \( i \) so \( P = iQ \).

- 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.

Sample Cryptosystems

Poll question #3
Sample Hybrid Cryptosystem

Alice’s Private Key -> Public Key Crypto Engine -> Digital Signature
Alice’s Message -> Hash Function
Random Secret Key -> Secret Key Crypto Engine
Bob’s Public Key -> Public Key Crypto Engine

Sent to Bob

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/AwUBNFUdO5WOcz5SftuEEQjx/ACaAgR07+vvDU6XWELV/GANjAAgBtUAnjg3Sdfw23qmZICLjPw7jP0Y8/M=jUAU
-----END PGP SIGNATURE-----
Case Study: PGP Encryption

Version: PGP for Personal Privacy 5.0
MessageID: DAdVB3wpBr3YRunzWVvhK5gBKBXOb/m

qANQ1R1BwU4D/T1T68XXuiUQCADfj3o4b4aFYBcNumA7hR1Wvz9rbv2BR6WB6USyZ8BIEFTjygCd96qF38sp9IqiJ1KNaNzfx2GLR WKFPzwhUXkB+AA5+1q5G/ELBvRac9XeafypbabZ6z6koQp+eEOXASaJ7eEFPdxzZTT7dVyiuxxUBBYNLN8bphdrzvZ/9Ak4/ClnLiJKn05/2UNESZB0e+31cvITMmFga+yVr5hXqocavFOKiin3hv7+Vx88uLem2/fQH2eGgVqkzVqXv8e0W5zuvw+y1WCh39wG3GQyOMkJiZIRI7axWnoU93KCmpR86OV4rDRA5uGL9ioSvze+q8XqubA6gdkKko6/2B4dctznLfw1L2YBS+dzFDw5desMFS0J7JkecAS4ANB9ja9K+f7PTAsesCBNETd49BT0FFTWWaAfegLYcPrcn4s3ErlUqvL3ozsFR4plcnHu6sa3ZJKTBbridoA3Vp9qG3hxfqfNyOlgAkmJJuQ50b9TehF88yC/e/VqUFdwb+QTrAz6NpIxi/x0FfOInhc/bBw7pDLXBFNaXhd1LqRQPrdmnWskKnOSarxq4GjpRTPo4hpCRJ7sJ5a7tZ09HPTXFG6iRjJw0aw47AR5nvKLEKoAJW5HaDKLruiWldtN4XeCwWVsfrjR32ez76W8aLpAK87GZEYTZeBx dv+1H0wWyT/y1cZQ/El5USEFp40KWF4uqquPeelOFeFMBo4Cvu6yH2XD/18Ft/53YWIebvdiCgsoaabK3jEdfGExce63zD10m

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 DES; XP also supports 3DES
      - Windows XP SP1, Server 2003, Vista, and Server 2008 default to AES; also support DES and 3DES
      - Windows 7 and Server 2008 R2 default to AES, SHA, and ECC; also support DES 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
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:\>
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
  - Ultimaco
- A case for live forensics...

© 1998-2012, Gary C. Kessler

Pre-Boot Logon

© 1998-2012, Gary C. Kessler
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
Plausible Deniability

Normal TrueCrypt volume

Normal TrueCrypt volume with a hidden container

Contents of /var/vm/sleepimage
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](http://www.jadsoftware.com/go/?page_id=167)

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

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).
Secure Communication Protocols

- Secure MIME (S/MIME)
- Secure Sockets Layer (SSL)
  - https, ftps, pop3s, 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

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
**Kerberos**

1) Once per Kerberos session, client connects to AS to obtain TGS session key and ticket-granting ticket (TGT).

2) Once per application session, client connects to TGS to obtain application session key (ASK) & application server's secret key.

3) The authenticated client sends its ticket, ASK, & encrypted application server's secret key to Application Server, and initiates connection.
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(?)

Sample Browser Certificate
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)

HTTP Over SSL

- Client
  - SSL
  - TCP/IP Stack
- Server
  - SSL
  - TCP/IP Stack
- Port 443
- URL form: https://www.example.com
SSL Handshake Protocol

CLIENT
- ClientHello
- Certificate*
- ClientKeyExchange
- Certificate
- changecipherspec
- Finished
- Application Data

SERVER
- ServerHello
- Certificate*
- Certificate Request*
- ServerKeyExchange*
- Certificate
- changecipherspec
- Finished
- Application Data

* optional or situation-dependent message

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!

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
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-->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
- ElcomSoft
  - Password Recovery Bundle
- AccessData
  - Password Recovery Toolkit (PRTK)
  - Distributed Network Attack (DNA)
  - Portable Office Rainbow Table (PORT)

© 1998-2012, Gary C. Kessler

A Little Crypto Humor...

http://xkcd.com/538/
Feb. 2009

© 1998-2012, Gary C. Kessler
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)

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

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

© 1998-2012, Gary C. Kessler
# Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DES</td>
<td>Triple DES</td>
</tr>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard (NIST)</td>
</tr>
<tr>
<td>ANS</td>
<td>American National Standard</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate authority</td>
</tr>
<tr>
<td>CPS</td>
<td>Certification practice statement</td>
</tr>
<tr>
<td>CRL</td>
<td>Certificate Revocation List</td>
</tr>
<tr>
<td>DES</td>
<td>Data Encryption Standard</td>
</tr>
<tr>
<td>DoB</td>
<td>Date of birth</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of service</td>
</tr>
<tr>
<td>DSA</td>
<td>Digital Signature Algorithm (NIST)</td>
</tr>
<tr>
<td>ECC</td>
<td>Elliptic Curve Cryptography</td>
</tr>
<tr>
<td>EFF</td>
<td>Electronic Frontier Foundation</td>
</tr>
<tr>
<td>EFS</td>
<td>Encrypting File System (W2K)</td>
</tr>
<tr>
<td>FIPS</td>
<td>Federal Information Processing Standard</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol (IETF)</td>
</tr>
<tr>
<td>HMAC</td>
<td>Hashed message authentication code</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol (IETF)</td>
</tr>
<tr>
<td>https</td>
<td>HTTP over SSL</td>
</tr>
<tr>
<td>IDEA</td>
<td>International Data Encryption Algorithm</td>
</tr>
<tr>
<td>IE</td>
<td>Internet Explorer (MS)</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IMAP</td>
<td>Internet Message Access Protocol (IETF)</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet service provider</td>
</tr>
<tr>
<td>ITU-T</td>
<td>International Telecommunication Union, Telecommunication Standardization Sector</td>
</tr>
<tr>
<td>IV</td>
<td>Initial vector</td>
</tr>
<tr>
<td>MAC</td>
<td>Message authentication code</td>
</tr>
<tr>
<td>MD2/4/5</td>
<td>Message Digest 2, 4, &amp; 5</td>
</tr>
<tr>
<td>MIPS</td>
<td>Millions of Instructions per second</td>
</tr>
<tr>
<td>MS</td>
<td>Microsoft</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>NNTP</td>
<td>Network News Transport Protocol (IETF)</td>
</tr>
<tr>
<td>NSA</td>
<td>National Security Agency</td>
</tr>
<tr>
<td>OS</td>
<td>Operating system</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal digital assistant</td>
</tr>
<tr>
<td>PGP</td>
<td>Pretty Good Privacy</td>
</tr>
<tr>
<td>PKC</td>
<td>Public key cryptography</td>
</tr>
<tr>
<td>PKI</td>
<td>Public key infrastructure</td>
</tr>
<tr>
<td>POP</td>
<td>Post Office Protocol (IETF)</td>
</tr>
<tr>
<td>RA</td>
<td>Registration Authority</td>
</tr>
<tr>
<td>R2/2/5</td>
<td>Rivest Cipher (or Ron's Code) 2, 4, and 5</td>
</tr>
<tr>
<td>RFC</td>
<td>Request for Comments (IETF)</td>
</tr>
<tr>
<td>RSA</td>
<td>Rivest, Shamir, Adleman</td>
</tr>
<tr>
<td>SCUBA</td>
<td>Self-contained underwater breathing apparatus</td>
</tr>
<tr>
<td>SHA</td>
<td>Secure Hash Algorithm (NIST)</td>
</tr>
<tr>
<td>SKC</td>
<td>Secret-key cryptography</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer (Netscape)</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol (IETF)</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security (IETF)</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>WDE</td>
<td>Whole disk encryption</td>
</tr>
<tr>
<td>W2K</td>
<td>Windows 2000 (MS)</td>
</tr>
<tr>
<td>XOR</td>
<td>Exclusive OR</td>
</tr>
</tbody>
</table>

© 1998-2012, Gary C. Kessler 74

---

# Acronyms and Abbreviations (cont.)

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

© 1998-2012, Gary C. Kessler 75
Questions? Comments? Queries?