Decrypting Crypto

Steve Weingart, CISA, CRISC
Manager of Public Sector Certifications
Aruba, A Hewlett Packard Enterprise Company

An Introduction to the Concept of Cryptography (and a few more details)

What is Cryptography?

From: crypt + graphia

1 : secret writing

2 : the enciphering and deciphering of messages in secret code or cipher

Why Do We Use Cryptography?

- To Hide Stuff!

but really… for

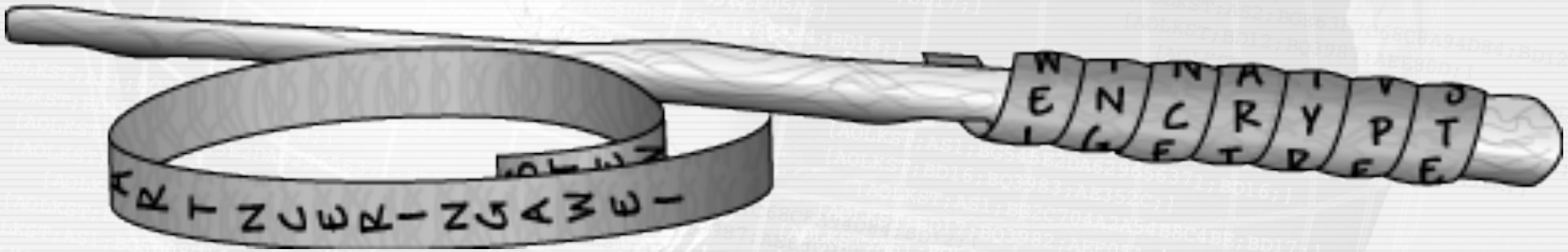
- Secrecy
- Integrity
- Authentication
- Non-repudiation
- And Many More Things…. (including protecting our monetary transactions and personal data)

But First… A Little History

- Cryptography dates back 4000 years to Egypt. One technique used a form of steganography

  A message was written on a belt wrapped around a staff, then the belt was unwound and worn by the deliverer, who also carried the staff. Upon delivery the belt was rewrapped around the staff to see the message.

Belt/Staff Steganography



… A little more history

- After getting started in Egypt, crypto just kept getting used
  - Mostly substitution codes i.e. a=1, b=2, etc
- Then came modern civilization and war
  - Crypto got serious!
  - One-time-pads, book codes, etc. were used to convey state/military secrets.
  - WWII fomented the birth of modern Cryptography (and computing, oh well, there went the neighborhood)

So, what is this crypto stuff, and how does it work?

- Steganography → hide the secret in something else
- Substitution Code → a=1, b=2…..
- Book Code → page 27, paragraph 3, word 5
- One-Time-Pad → a + #1, b + #2
- Mathematical/mechanical manipulation → Enigma, DES, AES, etc..

Steganography

- One of the oldest forms of cryptography
- Relies on hiding the information within another item
- The Egyptian example used a belt
- One modern example hides data in .JPG pictures

Substitution Codes

- The classic crypto!
- Arthur Conan Doyle's Sherlock Holmes had the "Dancing Men"
- Did you have a Captain Video Secret Decoder Ring???
- Super easily broken by letter frequency and examination.

Substitution Codes (cont)

- One special substitution code was never broken!
- Navaho code talkers transmitted the information in their language, which no one knew.
- To complicate frequency analysis, several words were used for the same things (a= apple or ant or aunt…..). This technique is still used and is called a polyalphabetic cipher

Book Codes

- Each word (or optionally each letter), is called out by a reference to a word in a book, usually by page, paragraph and word number.
- The parties share the book as a secret. If the book is not discovered, this code is unbreakable.
- Useful, easy, low tech!

One-Time-Pads

- A string of random numbers is generated, each party gets a copy
- By an agreed upon method, the message is encrypted by combining with the pad i.e. a=1, b=2, then add that to the pad number 32, 47, 63, 9, 11, etc., subtract each in order to decrypt
- Each number in the pad is only used once.
- If the pad is kept secret, and used only once, this cipher is unbreakable.
- A little higher tech than a book code
- But riskier, the pad is more obvious

Mathematical/Mechanical Manipulation

- This is Modern Cryptography
  - We have arrived!
- Symmetric Key Block/Stream Algorithms
  - AES, DES, 3DES, RC4, AES
- Public Key Algorithms
  - RSA, ECC
- Secure Hash Algorithms
  - MD5, SHA-1, SHA-256, SHA-3
- And many more….
  - But let's start here….

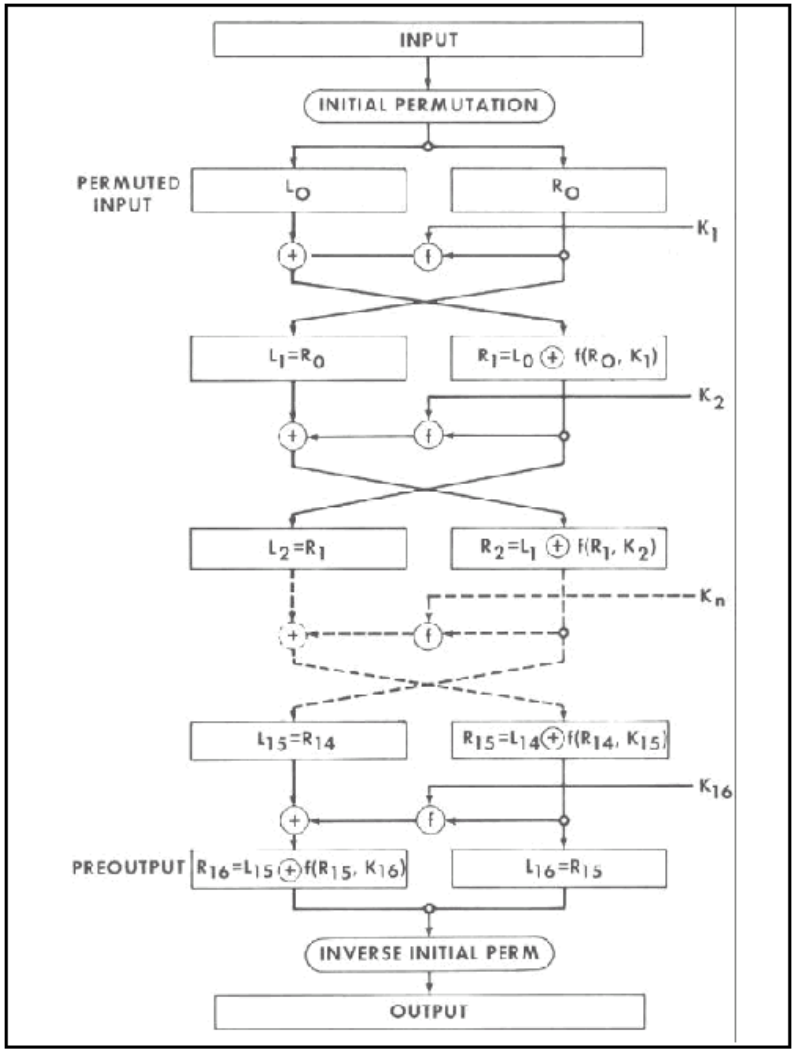
Modern Crypto

- Really started in WWII
  - Polyalphabetic substitution (mechanical rotors) Enigma is a very famous example
- Helped start modern computing as well
  - Computers were employed to both make and break ciphers
  - This lead to breakthroughs in computing

Symmetric Key Block/Stream Ciphers

- These algorithms are used to provide secrecy, they can also be used to check integrity, but the key must be kept secret for it to be of any value.
- DES, developed in the 1970's was the first commercial high quality cipher.
- These algorithms use a key (a string of bits), to both encrypt or decrypt (hence symmetric), a block of data at a time.
- Using a complex iterative algorithm, there is no way to analyze the data.
- The only known attack is brute force (trying every key), but with today's computing power, and the small (56 bit) key space for DES, it has become practical (< 24 hrs).
- So we had to go to a better algorithms.

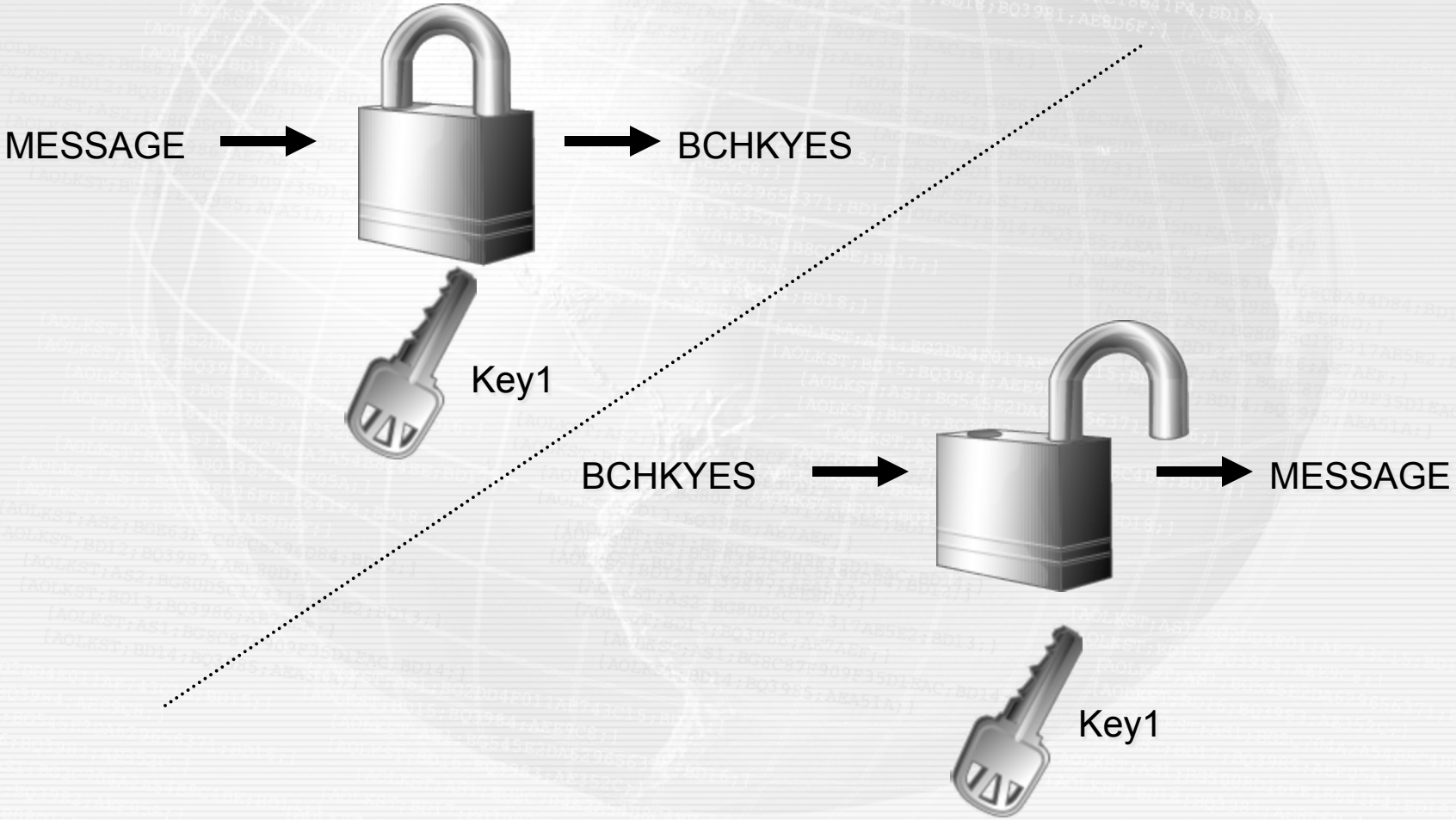
DES


INPUT
INITIAL PERMUTATION
PERMUTED INPUT
L_0
R_0
K_1
$L_1=R_0$
$R_1=L_0 \oplus f(R_0, K_1)$
K_2
$L_2=R_1$
$R_2=L_1 \oplus f(R_1, K_2)$
K_n
$L_{15}=R_{14}$
$R_{15}=L_{14} \oplus f(R_{14}, K_{15})$
K_{16}
PREOUTPUT
$R_{16}=L_{15} \oplus f(R_{15}, K_{16})$
$L_{16}=R_{15}$
INVERSE INITIAL PERM
OUTPUT

Symmetric Key Block/Stream Ciphers (cont)

- 3DES (or Triple-DES) iterates DES 3 times by encrypting, decrypting, encrypting to achieve higher strength (up to 168 bit key)
- AES is the official replacement for DES/ 3DES (up to 256 bit key)
- RC4 was commonly used for Web transactions, but like DES is historical at this point.

Basic Symmetric Encryption



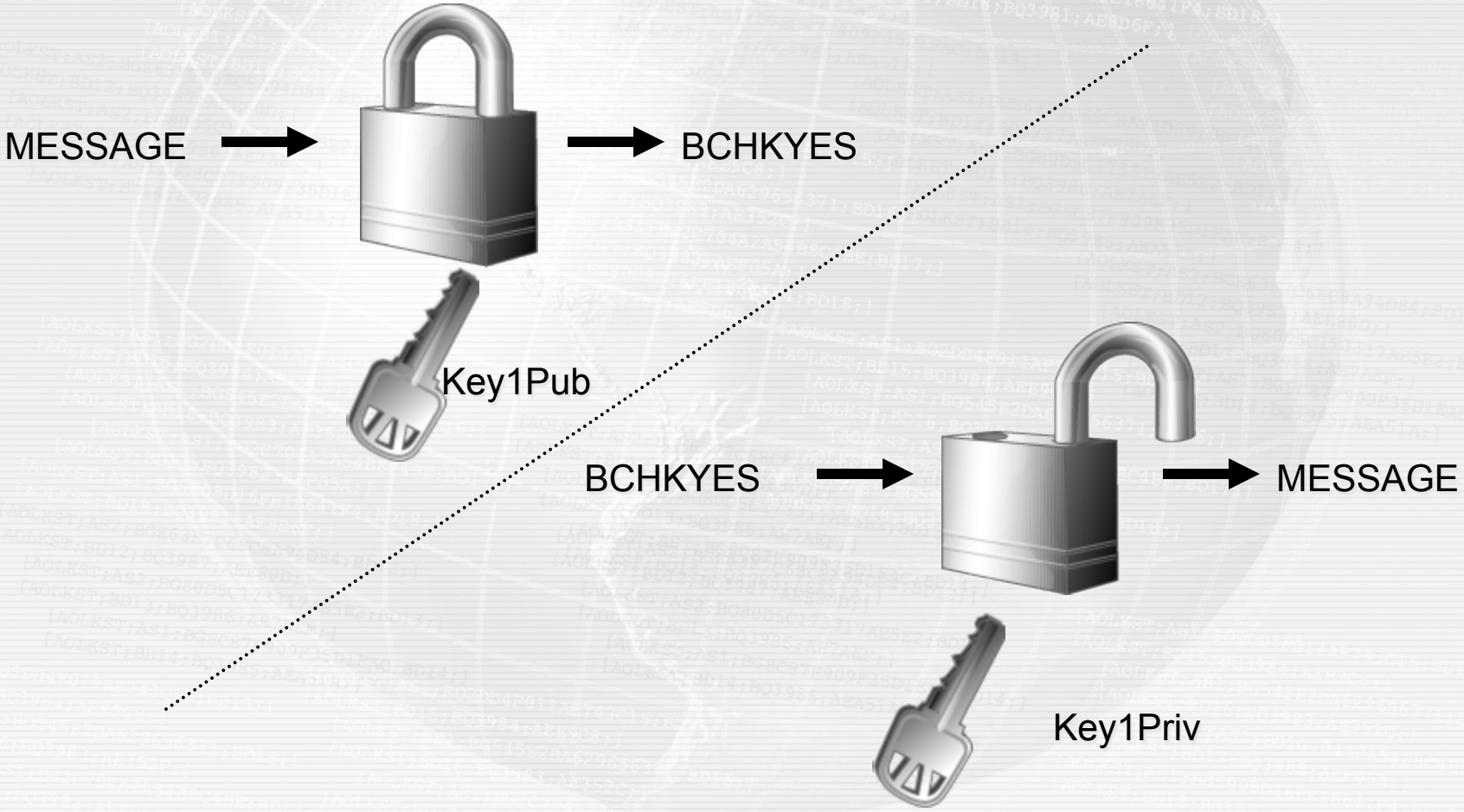
Public Key Algorithms

- Really Public/Private Key Algorithms
  - Asymmetric Key Algorithm
- This is a BIG thing!!!!!
- You can make part of your key public
  - Anyone can look up your public key and encrypt a message that ONLY you can decrypt
  - Not possible with symmetric key algorithms
- By mixing symmetric & asymmetric key algorithms, and hashing; we can create protocols that allow us to do all sorts of really useful things!

Public Key Algorithms (cont)

- RSA is the most commonly used public key algorithm
- ECC (elliptic curve cryptography) has gained more widespread use because it can have a small in code footprint (but it doesn't fair well against quantum)
- But.. Public Key Algorithms are computationally VERY intensive, so are typically used to transfer small amounts of information, like an AES key, or a signature, etc…

Basic Public Key Encryption



Secure Hash Algorithms

- Used to verify integrity
- Keyless (usually), one way, crypto
- Data in, fingerprint out, you CAN'T recover the data!
- ANY change to data makes the fingerprint change
- The SHA family are the common Hash algorithms.

OK, so now that we have these things, what do we do with them?

- With basic crypto tools, we can do some really useful things
  - Send secret messages
  - Send hashes of files for integrity proofing
  - Send a secret key using a public key algorithm
  - Enable non-repudiation using public/private keys
- And most importantly, we can develop protocols that allow us to do combine crypto functions to do even more useful things!

Crypto Protocols

- By combining basic crypto building blocks we can make easy to use, very powerful tools
- By using multiple algorithms and functions in a sequence, we can do very complex things securely
- The most important thing is that ALL parts of a protocol and the protocol itself have been reviewed by experts.
- Proprietary protocols are always a risk!

SSL, a crypto protocol example

- SSL (secure sockets layer)
  - This is how most web transactions were done (actually a lot still are)
  1. A random number generator is used to create a secret
  2. The client looks up the public key of the host and sends a secret to the host
  3. The (now) shared secret is used to create a session key that is used to block encrypt the exchange (and a hash algorithm verifies integrity)

TLS, an improved crypto protocol example

- SSL (secure sockets layer)
  - This is how most web transactions are done now

1. A random number generator is used to create a secret at each end
2. ***Both*** the client and the server look up the public key of each other and send the secrets to start the communication
3. The (now) shared secrets are used to create a session key that is used to block encrypt the exchange (and a hash algorithm verifies integrity)

A Brief Diversion…

PKI: Definition

Short for *public key infrastructure,* a system of digital certificates, Certificate Authorities, and other registration authorities that verify and authenticate the validity of each party involved in an Internet transaction. PKIs are currently evolving and there is no single PKI nor even a single agreed-upon standard for setting up a PKI. (but no one waited).

A PKI is also called a *trust hierarchy.*
(www.webopedia.com)

PKI

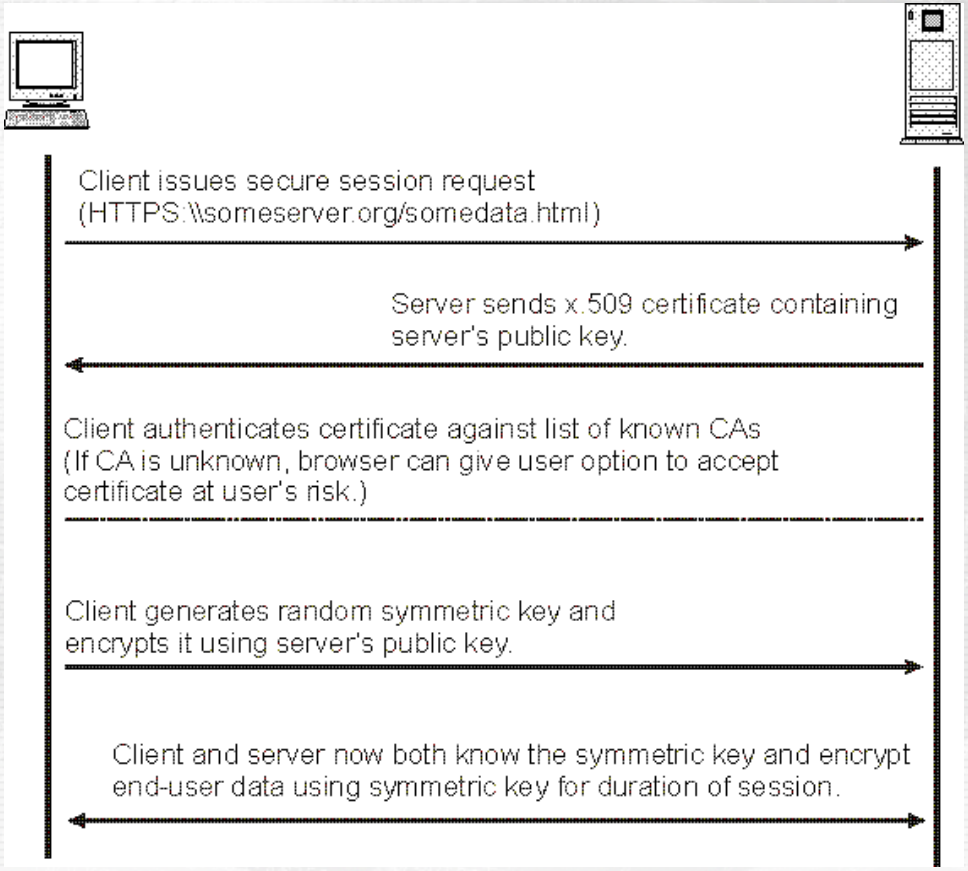
- It's a concept (like cryptography), not a thing!
- PKI protocols can be used/developed to perform remote key management/distribution
  - This is the Holy Grail of PKI! (the quest continues)
- The most common PKI is the way public key certificates are managed on the web now
  - A 'Certificate Authority' serves public keys
  - Each server keeps it's own private key
  - The Certificate Authority can be hierarchical

Now back to SSL & TLS

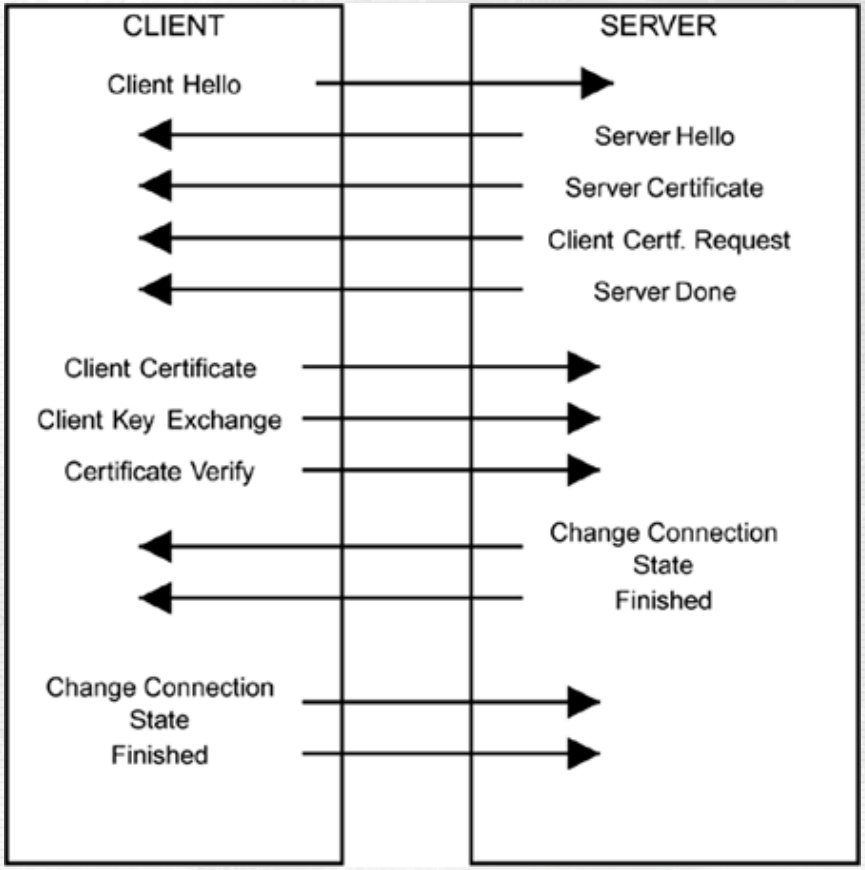
How SSL & TLS Work

- Earlier SSL and TLS were introduced, that was just a way to introduce PKI
- Now let's talk about the actual mechanism

How SSL Works



How TLS Works



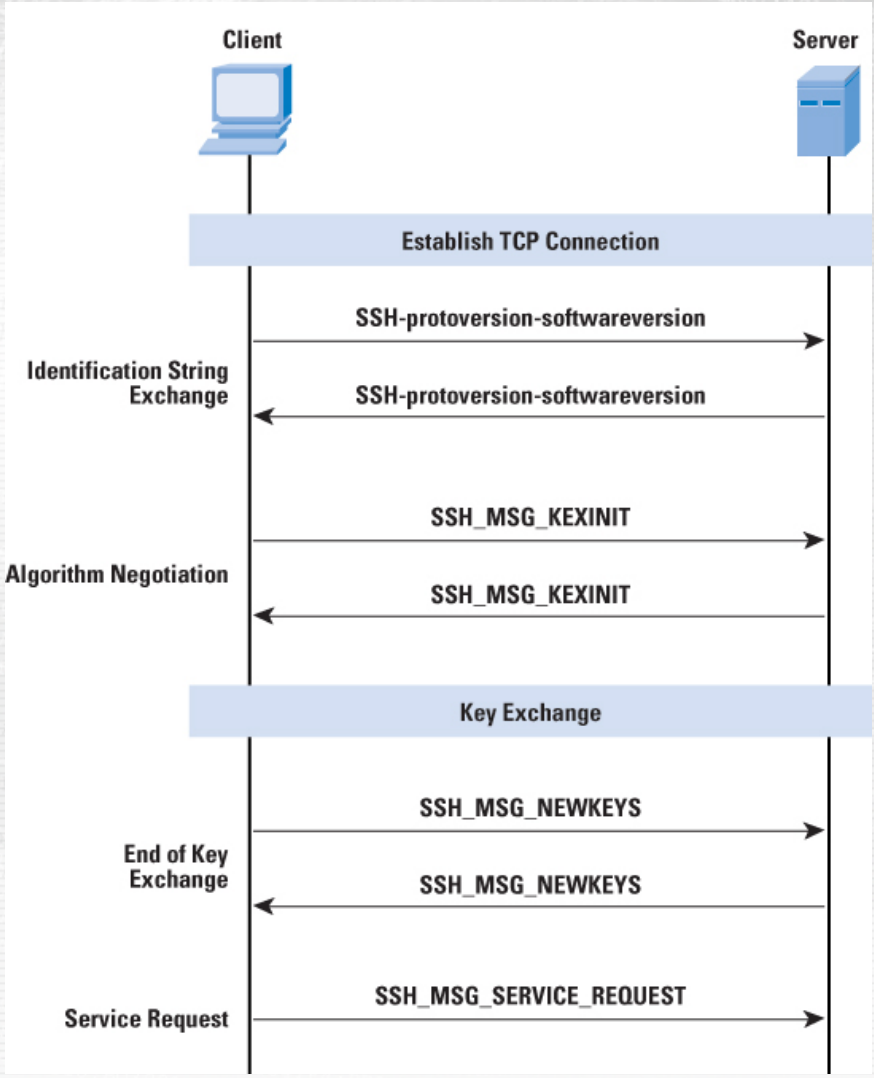
SSL v. TLS

- SSL is a *unidirectional* authentication, the server validates itself to the client
- TLS is a *bidirectional* authentication, both the server and the client validate themselves to each other.
- This prevents either end from spoofing its identity to the other

SSH: A Secure Terminal

- SSH is the standard mechanism to open a terminal session to another computer. It's usually used for management (but back in the day this was how most people connected to the Internet)
- It sets up a session by negotiating the version, then the crypto suite, then authentication then establishes a tunnel for the terminal session

SSH: A Secure Terminal



IPSec: A Secure Tunnel

- IPSec is really a collection of protocols that creates a secure tunnel between two entities.
- It sets up a session by negotiating the version, then the crypto suite, then authentication then establishes a tunnel

IPSec: A Secure Tunnel

- IPsec involves two security services:
  - Authentication Header (AH): This authenticates the sender and it discovers any changes in data during transmission.
  - Encapsulating Security Payload (ESP): This not only performs authentication for the sender but also encrypts the data being sent.

IPSec: A Secure Tunnel

- There are two modes of IPsec:
  - Tunnel Mode: This will take the whole IP packet to form secure communication between two places, or gateways.
  - Transport Mode: This only encapsulates the IP payload (not the entire IP packet as in tunnel mode) to ensure a secure channel of communication.

OK, Now That We Have an Idea What We Can Do With Cryptography, Let's Dig Into Some of the Details

Cryptographic Algorithms We Use All The Time

There are three types of cryptographic algorithms that we commonly use

1) Symmetric
2) Asymmetric, or Public Key
3) Hashes and MACs (Message Authentication Codes)

Symmetric Cryptographic Algorithms

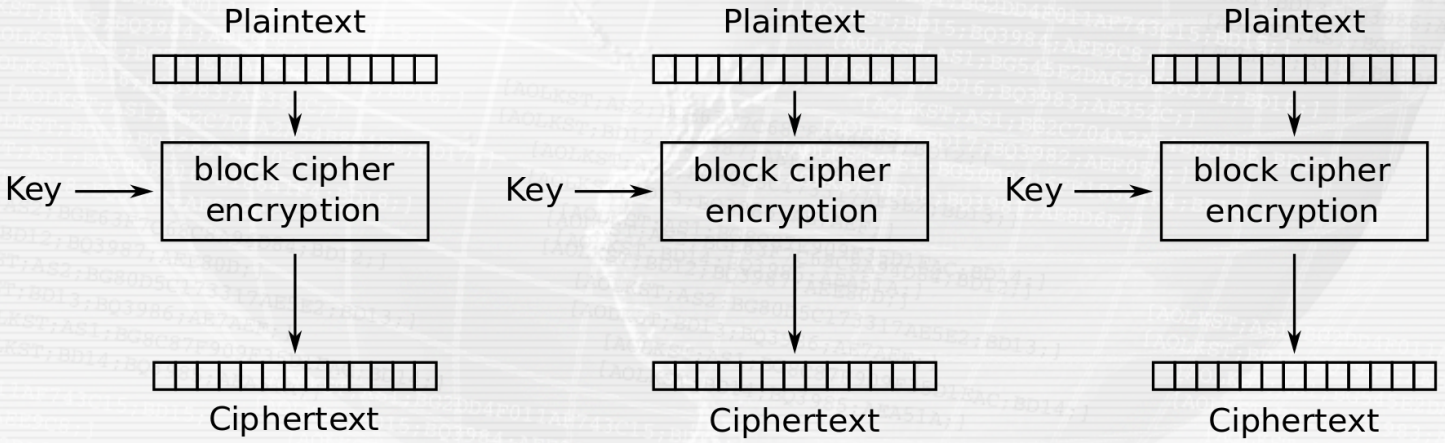
For the most part AES is the symmetric algorithm of choice in the US.

Triple-DES, the original DES, used with 3 different keys to encrypt, decrypt (under a second key), then encrypt again (under the third key), is also still an approved function.

Symmetric Cryptographic Algorithms: Modes

Symmetric cryptographic algorithms can be used in several different modes. Each of these modes have advantages (and disadvantages) for particular situations.

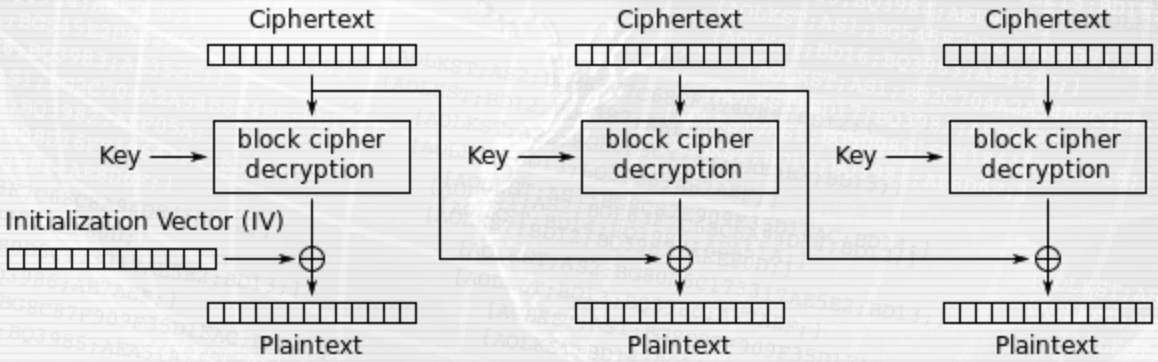
- ECB (Electronic Code Book)



Electronic Codebook (ECB) mode encryption

Symmetric Cryptographic Algorithms: Modes

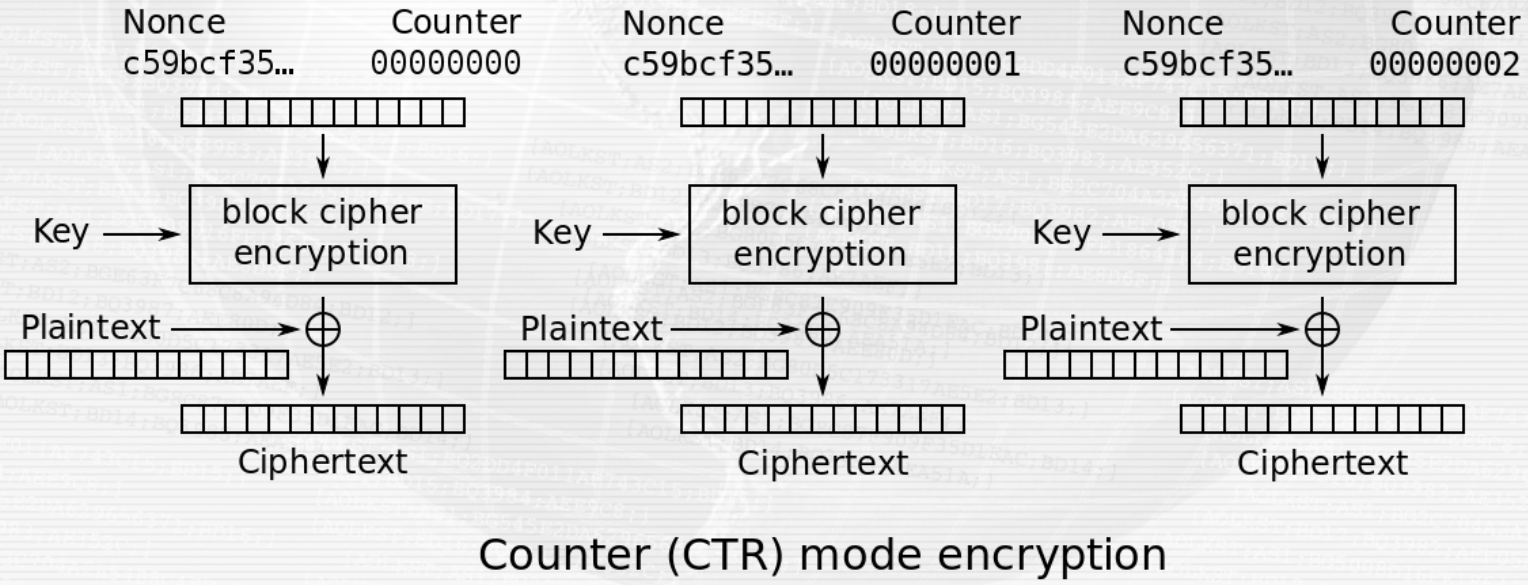
- CBC (Cipher Block Chaining)



Cipher Block Chaining (CBC) mode decryption

Symmetric Cryptographic Algorithms: Modes

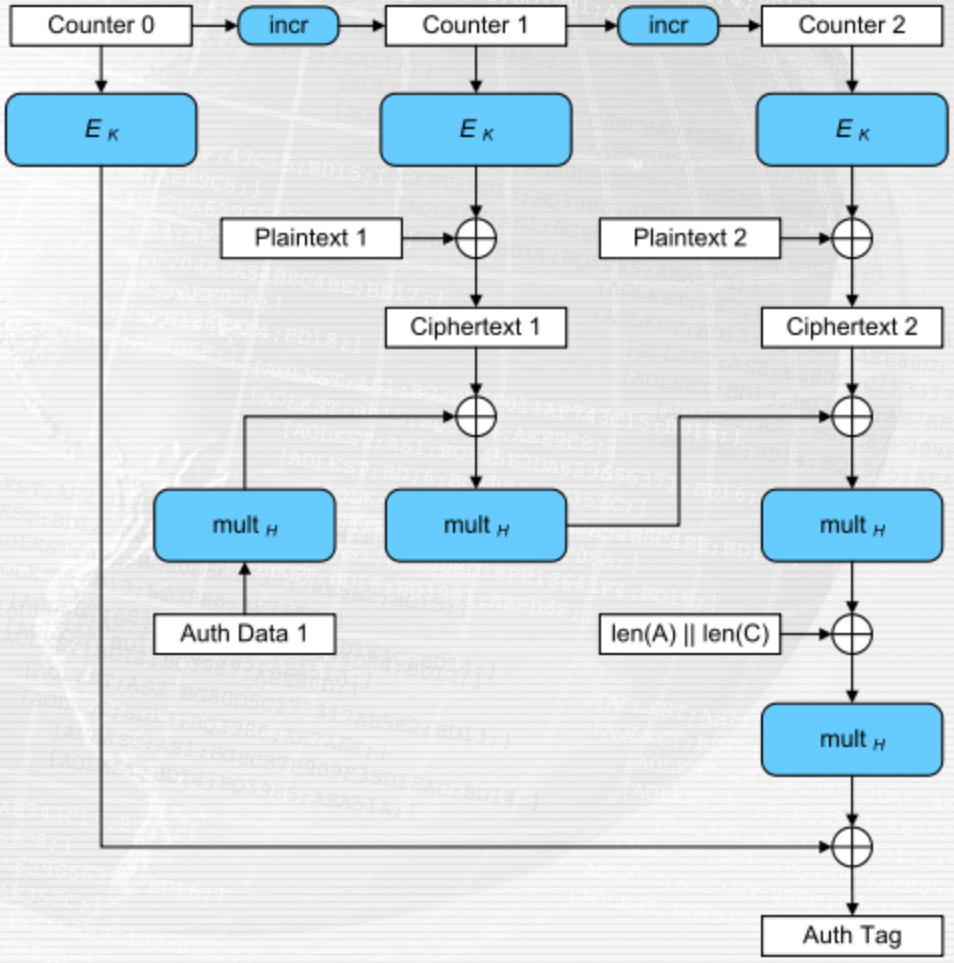
- CTR (Counter)



Counter (CTR) mode encryption

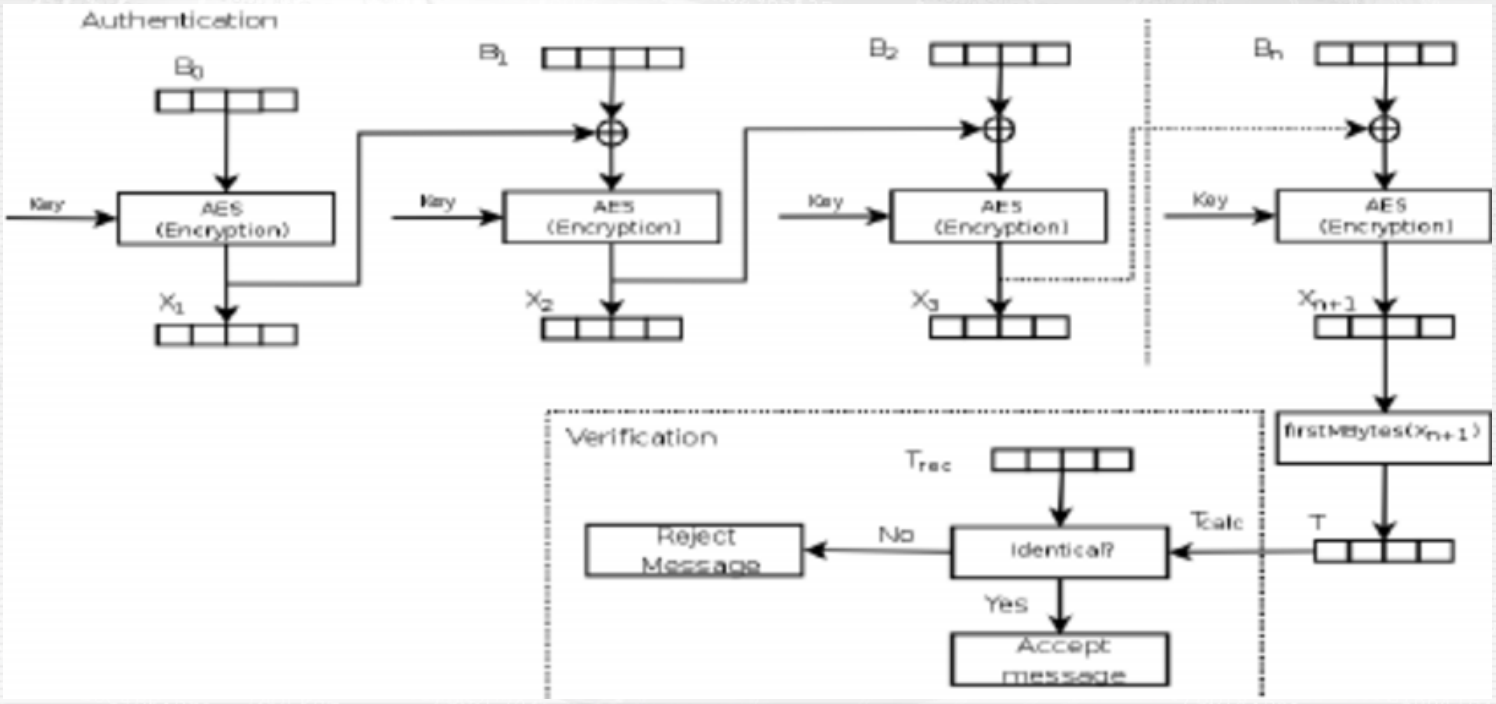
Symmetric Cryptographic Algorithms: Modes

- GCM
- Galois Counter Mode



Symmetric Cryptographic Algorithms: Modes

- CCM



Symmetric Cryptographic Algorithms: Modes

- There are several more modes
  - OFB
    - Output Feedback
  - CFB
    - Cipher Feedback
  - XTS
    - XEX-based tweaked-codebook mode with ciphertext stealing
    - Usually used for disk encryption

Hashes and MACs

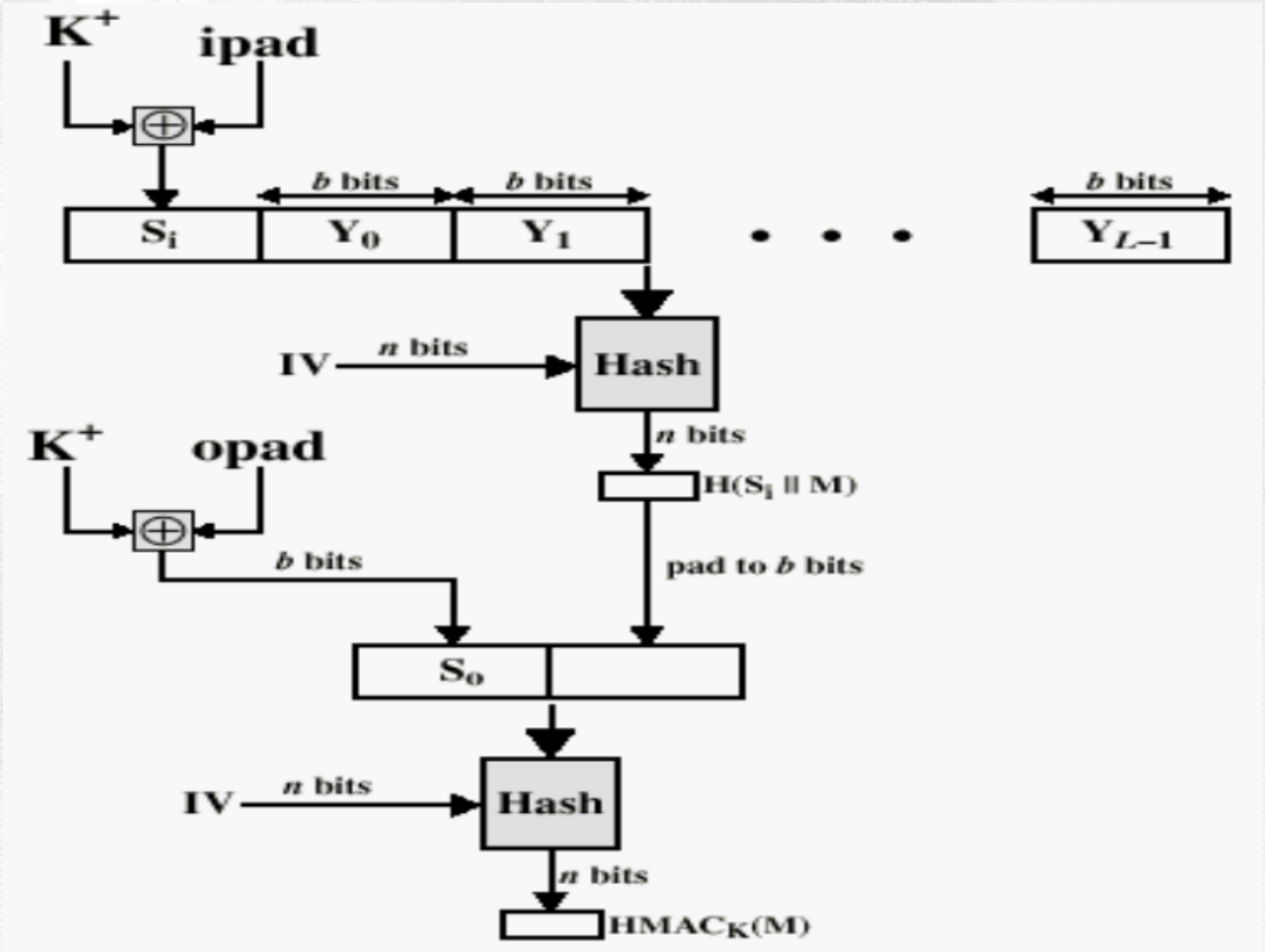
- Hashing creates a ‘unique’ signature for a block of data
  - But anyone can create the same hash for the same data
    - The SHA family is the most used
  - Message Authentication Codes create a keyed unique signature for a block of data
    - HMAC (hash message authentication code) which concatenates a key with the message, then hashes the result, is one popular method
    - CMAC (Cryptographic message authentication code) uses a cryptographic algorithm (i.e. AES) instead of a hash and keep the last block as the Auth code.

Hashes and MACs

- The SHA family is the most used, approved hash function
  - SHA-1, the original has some weaknesses and is used less often
  - SHA-224, 356, 384 and 512 are now the most commonly used
  - SHA-3 is a new alternative for diversity
- Now we'll look at some of the MACs (Message Authentication Codes

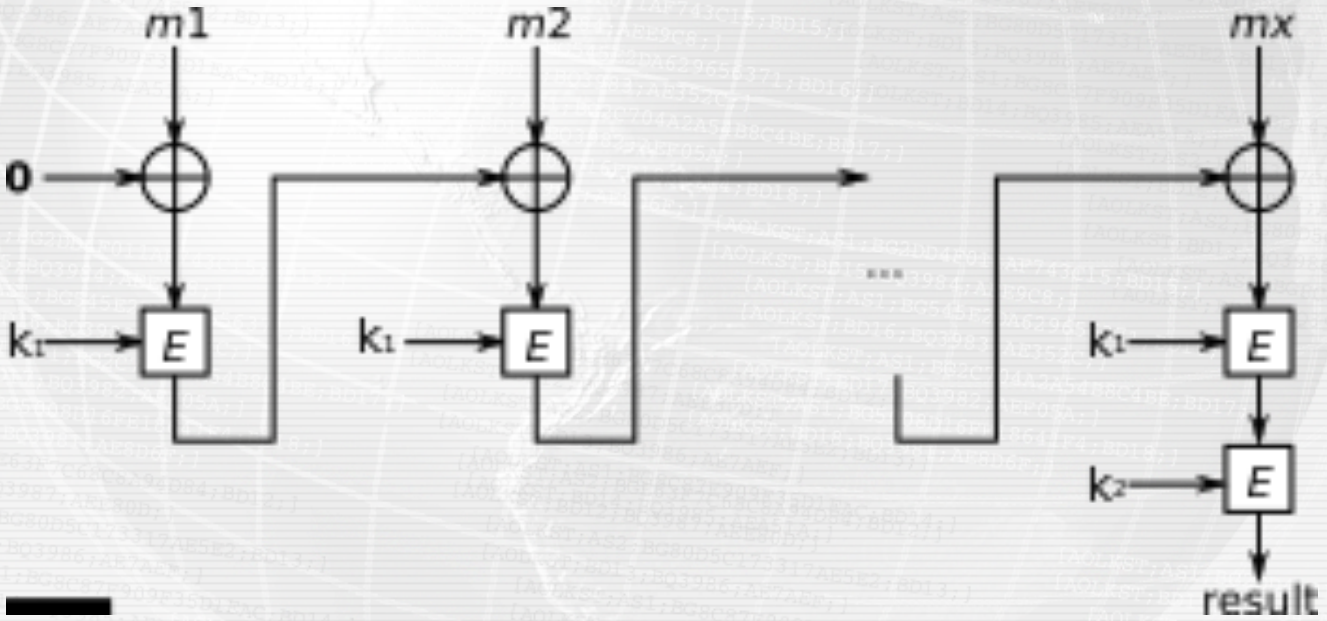
Hashes and MACs

- HMAC



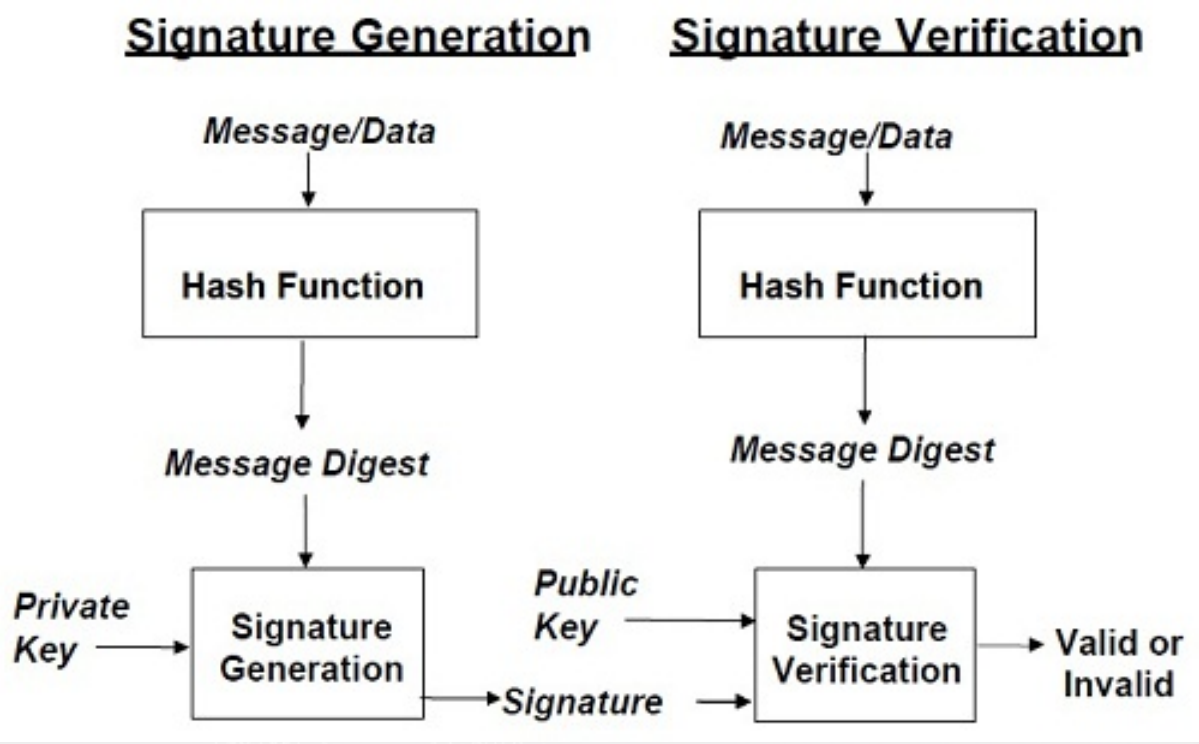
Hashes and MACs

- CMAC



Signatures

- Signatures cryptographically secure a hash for the highest level of integrity verification and non-repudiation
- RSA 2048 with SHA-256 is the current choice



Common Real World Use Scenarios

- Now we'll look at a collection of common things you may do using cryptography
- These are real world examples, from connecting to WiFi, to updating the firmware in your router, to sending a private message
- Everything we've talked about gets used here

Connect to WiFi (Bad, bad version)

1. Turn on WiFi
2. Look for an access point (AP)
3. Connect with no authentication or encryption
4. Hope you are connected to the Internet
5. Share your data with anyone ‘listening’

Connect to WiFi (Slightly better version)

1. Turn on WiFi
2. Look for an access point (AP)
3. Connect (WEP, WPA, WPA2/802.11i PSK)
4. Enter the password on the card on the wall
5. Hope you are connected to the Internet
6. Share your data with anyone ‘listening

Connect to WiFi (Decent home or Small Office version)

1. Turn on WiFi
2. Look for a particular access point (SSID)
3. Connect (WPA2/802.11i PSK)
4. Enter the password from your router or office
5. Connected to the Internet
6. Now you probably won't share your data with anyone 'listening'

Connect to WiFi (Enterprise version)

1. Turn on WiFi
2. Look for a particular access point (SSID)
3. Connect (WPA2/802.11i certificate)
4. The certificate that IT installed in your laptop or told you how to install authenticates you via RADIUS
5. Connected to the Internet
6. Now you very probably won't share your data with anyone 'listening'

Send Secret Message (simple basic version)

1. Create a Key
2. Encrypt the Message (Triple-DES or AES)
3. Deliver the Key to the Recipient *out of band*
4. Send the Message
5. Recipient Decrypts the Message with the Key

Send Secret Message (Internet Shopping Version)

1. Contact the https webserver with your browser
2. Under the covers, your browser and the server create an SSL or TLS session (nothing has to be out of band, because the public key makes it possible to send a secret in band)
3. Send the Message (your order and CC #)
4. They ship your new fill-in-the-blank

Send Secret Message (IPSec Version)

1. Connect to the enterprise network with IPSec
2. The person you want to send to does as well
3. Send the Message
4. The message is private on the network, but not to individuals on the network (Unless it is point-to-point)

Update Firmware (Router, etc)

1. The vendor, hashes the code and encrypts the hash with an RSA private key
2. You download the new firmware
3. The installer does a certificate verification, then decrypts the hash using the public key
4. The code is hashed locally and the value verified
5. If the hashes match, your new firmware is installed

Conclusion

- Cryptography is one of the main tools we use to achieve security
- It is COMPLICATED!
- Unless that is your chosen field, don't expect to understand all of it unless you plan to do a lot of studying (I had to accept that too)
- But, a basic understanding of the algorithms, uses and protocols goes a long way to making you crypto literate
- You can amaze your friends at parties!

Bibliography

- Handbook of Applied Cryptography; Menses, VanOorschot & Vanstone;
  - ISBN 0-8493-8523-7
- Applied Cryptography; Schneier;
  - ISBN 0-471-12845-7
- The Codebreakers; Kahn
  - ISBN 0-684-83130-9
- Many pages on the Web (searching is the best way to get specific answers)

Questions?

?

Thank You!

Steve Weingart

steve.weingart@hpe.com