



Security Vulnerabilities

The devil is in the details

Errors, Bugs, and Failures

- Computers are composed of hardware whose behavior is determined by software (roughly...)
- Applications run on operating systems and interoperate through protocols
- Hardware and software are developed by humans and therefore aren't perfect
- A human **error** may introduce a **bug** (or fault)
 - The IEEE Standard Glossary of Software Engineering Terminology defines “fault” as “an incorrect step, process, or data definition in computer program”
- When a fault gets triggered, it might generate a **failure**

Windows

A fatal exception 0E has occurred at F0AD:42494C4C
the current application will be terminated.

- * Press any key to terminate the current application.
- * Press CTRL+ALT+DELETE again to restart your computer.
You will lose any unsaved information in all applications.

Press any key to continue

You need to restart your computer. Hold down the Power button for several seconds or press the Restart button.

Veuillez redémarrer votre ordinateur. Maintenez la touche de démarrage enfoncée pendant plusieurs secondes ou bien appuyez sur le bouton de réinitialisation.

Sie müssen Ihren Computer neu starten. Halten Sie dazu die Einschalttaste einige Sekunden gedrückt oder drücken Sie die Neustart-Taste.

コンピュータを再起動する必要があります。パワーボタンを数秒間押し続けるか、リセットボタンを押してください。

```
[ 0.682627] Failed to execute /init (error -2)
[ 0.682777] Kernel panic - not syncing: No working init found. Try passing i
nit= option to kernel. See Linux Documentation/admin-guide/init.rst for guidance
.
[ 0.682832] CPU: 1 PID: 1 Comm: swapper/0 Not tainted 4.16.6-2-CHAKRA #2
[ 0.682875] Hardware name: To Be Filled By O.E.M. To Be Filled By O.E.M./IMB-
A180, BIOS P1.00 10/09/2013
[ 0.682921] Call Trace:
[ 0.682974] dump_stack+0x5c/0x85
[ 0.683015] ? rest_init+0x50/0xd0
[ 0.683057] panic+0xe4/0x253
[ 0.683101] ? do_execveat_common.isra.39+0x87/0x830
[ 0.683142] ? rest_init+0xd0/0xd0
[ 0.683185] kernel_init+0xeb/0x100
[ 0.683228] ret_from_fork+0x22/0x40
[ 0.683305] Kernel Offset: 0xa000000 from 0xffffffff81000000 (relocation rang
e: 0xffffffff80000000-0xffffffffbfffffff)
[ 0.683354] ---[ end Kernel panic - not syncing: No working init found. Try
passing init= option to kernel. See Linux Documentation/admin-guide/init.rst for
guidance.
```

—

Security [Errors, Bugs, and Failures]

- A security error is made by a human
- As a consequence, a security bug is introduced in a program
 - A security bug is also called a “**vulnerability**”
- When a bug is triggered (or “exploited”) it generates a security failure
- As a consequence, the security policy of a system is violated and the system is compromised

No exact OS matches for host

Nmap run

sshnuke

Connecting

Attempting to exploit SSHv1 CRC32 ... successful.

Resetting root password to "210N0101".

System open: Access Level <9>

ssh 10.2.2.2 -l root

root@10.2.2.2's password:

RRF-CONTROL> disable grid nodes 21 - 48

Warning: Disabling nodes 21-48 will disconnect sector 11 (27 nodes)

ARE YOU SURE? (y/n) y

Grid Node 21 offline...

Grid Node 22 offline...

Grid Node 23 offline...



Other Security Problems

There is an overall concept of “system security” in terms of

- Privacy / Confidentiality
- Integrity / Consistency
- Availability

Some applications might work as designed but contain vulnerabilities ...

- ... when installed in systems with a conflicting security policy
 - “We allow students to have PHP applications in their web home directories”
- ... when configured insecurely
 - The service is protected by a 16 character password (set to AAAAAAAAAAAAAAAAAA)

Attacker



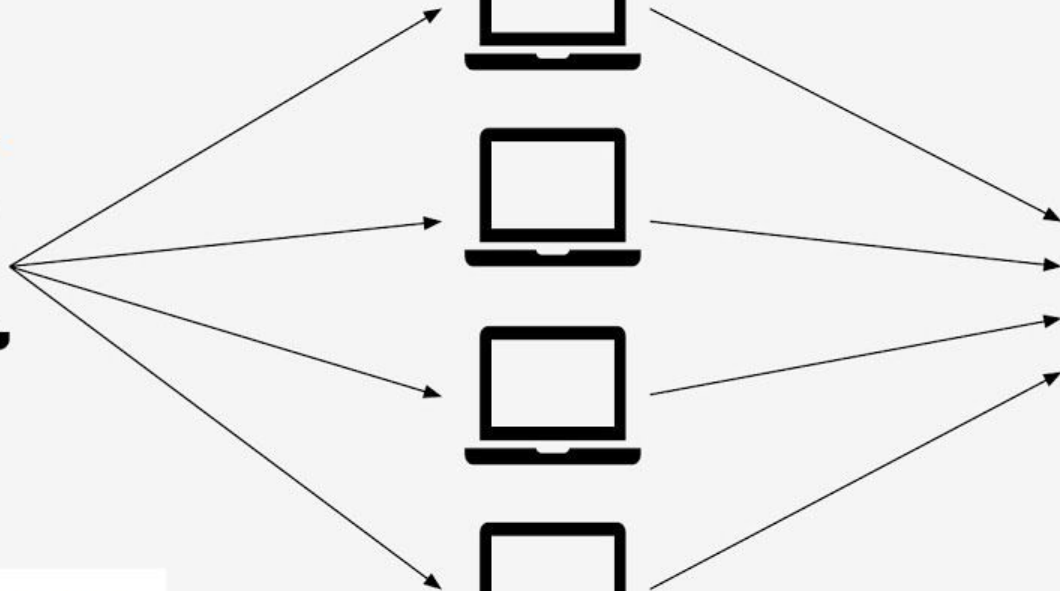
Slaves



Victim



DDoS Attack



The “solution” to the Security problem

- Strong authentication on both services and users
- Reliable authorization / access control
- Effective abuse control
- Secure design of protocols, operating systems, and applications
- Bug-free implementation of protocols, operating systems, and applications
- Perfect security policy
- Perfect policy enforcement
- ... and perfect users!

... and the real world

- Effective security protections are not deployed
- Administrators do not keep up with vendor updates/patches
- Sites do not monitor or restrict access to their internal hosts
- Organizations do not devote enough staff/resources to maintain security
- Users are not educated about security risks
- Sites do not implement policies (if they have one)

Security Analysis

- Security analysis is the process of determining the security posture of a system
 - With respect to a set of known design guidelines
 - With respect to a set of known security problems
 - With respect to its environment
- Security analysis answers the questions
 - Is it designed securely?
 - Is it implemented securely?
 - Is it deployed and configured securely?
- The security analysis process is difficult to automate and requires experience and skills (the backward / oblique mindset)



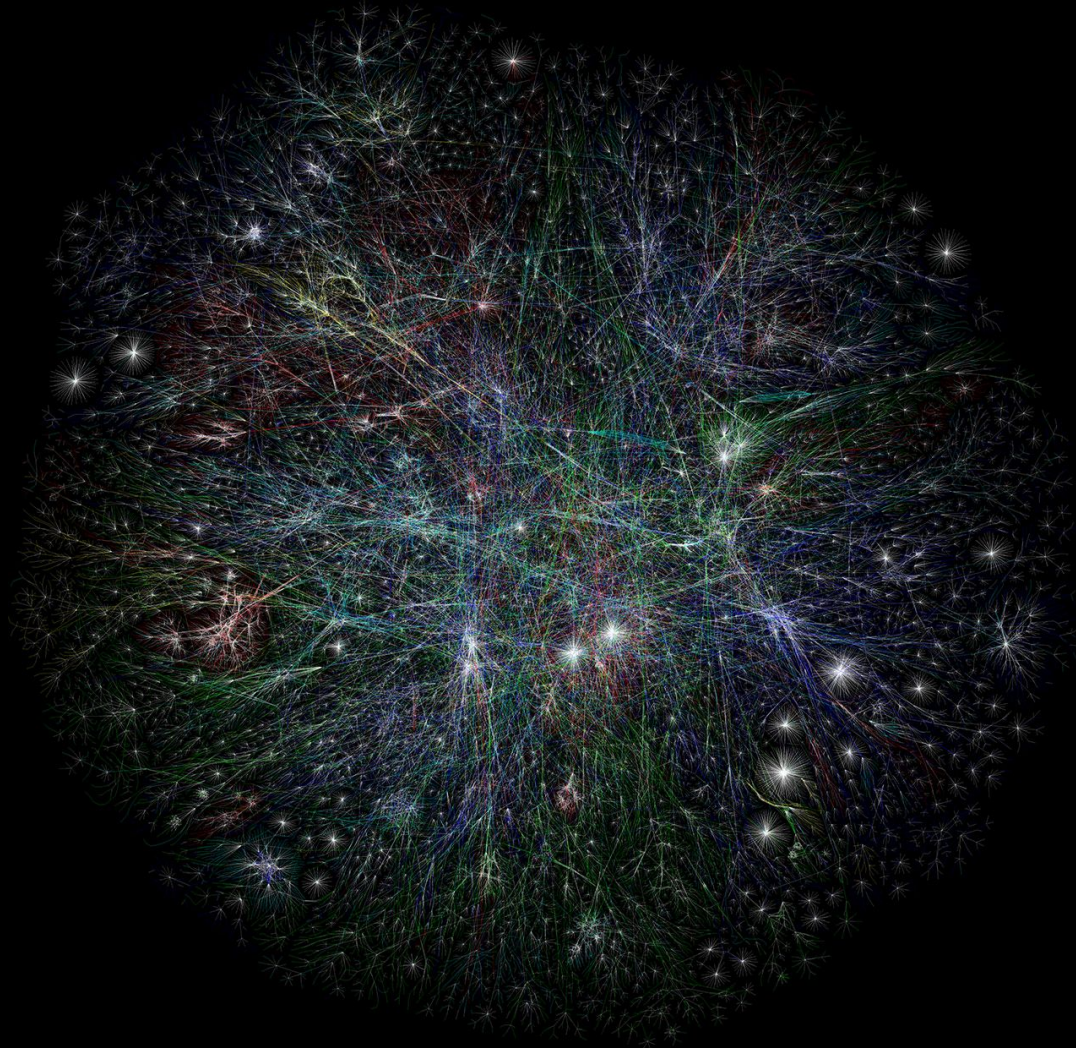
Brief History of Hacking

The internet

- A network of networks
- Composed of a set of autonomous subnetworks
- Open architecture
- Different Administrative domains with different (and possibly conflicting) goals
- Governments, companies, universities, organizations rely on the Internet to perform mission-critical tasks

History (90's)

- Fast growth (size and traffic volume)
- 1991: Tim Berners-Lee (CERN) creates the World-Wide Web
- 1993: The Mosaic browsers introduces the general public to the web
- The CGI specification (1993) supports web-based access to existing applications and services
- The Internet explodes



History (00-10's)

- The web becomes part of our everyday life
- JavaScript and asynchronous communication create a new application paradigm
- Web-based services and applications become the way in which we access, process, and store information
- Smartphones become the most used platform to access the web
- Everything becomes networked (more or less): Internet of Things (IoT) ...



Cap'n Crunch

- In 1972 John Draper finds that the whistle that comes with the Cap'n Crunch cereal produces a sound at the 2600 Hz frequency
- The 2600 frequency was used by AT&T to authorize long-distance calls

Phone Phreaking

- John Draper became Captain Crunch and built a blue box
- The blue box produced a number of different tones that could be used for in-band signaling
- Draper was eventually sentenced for five years' probation for toll fraud
- His story became an integral part of hacker culture

The German Hacker Incident

- Cliff Stoll, a system administrator at LBL in August 1986 was investigating a 75% discrepancy for CPU time
- He found out that an account had been created with no billing address
- He identified an intruder, instead of cutting him out he investigated the case
- He was using Emacs as a mailer, but in order to access e-mail, Emacs was using `movemail` to access `/var/spool/mail`, that required root access
- To allow this, `movemail` was set as `setuid root`
- The attacker was looking for “stealth”, “NORAD”, “nuclear”, so with the help of FBI they arrested Markus Hess in Hannover

The Internet Worm

- November 2, 1988: The “Internet worm”, developed by Robert T. Morris, was injected in the internet
- A mistake in the replication procedure led to unexpected proliferation
- The internet had to be “turned off”
- Damages were estimated in the order of several hundred thousand dollars
- RTM was sentenced to three years’ probation, a 10k\$ fine and 400 hours of community service



The Worm

A worm is a self-replicating program that spreads across a network of computers

The Morris Internet worm worked only on BSD UNIX

The worm consisted of two parts:

- A main program
- A bootstrap program

The Morris Internet Worm source code

This disk contains the complete source code of the Morris Internet worm program. This tiny, 99-line program brought large pieces of the Internet to a standstill on November 2nd, 1988.

The worm was the first of many intrusive programs that use the Internet to spread.

The Computer History Museum



Bootstrap program: gain remote privileged access

Finger buffer overflow:

```
char line[512];  
line[0] = '\\0';  
gets(line)
```

Sendmail: the DEBUG option allowed one to specify a number of commands to execute

- The bootstrap program (99 lines of C code) was transferred using a connection from the infecting machine

Main program

- Gathered information about the host's network interfaces and host with open connections to infect hosts
- Tried to break into hosts by using rsh, finger, sendmail
- Gathered more information on trusted hosts by examining
 - /etc/hosts.equiv
 - /.rhosts
 - ~/.forward in users home dirs
- Tried to rsh to the referenced hosts (password-cracking attack using the information contained in the password file, an internal dictionary of 432 words, and, eventually, the local UNIX dictionary)
- For each successful break-in the work was transferred

<https://pdos.csail.mit.edu/6.828/2018>

XV6
OPERATING SYSTEM

Add a new system call

RTM is now professor of
Operating Systems at MIT



Kevin Mitnick

- One of the most well-known hackers in the community
- 1982-1994: Sentenced many times for performing illegal activities
- 1994: California Department of Motor Vehicles issues \$1-million warrant for Mitnick's arrest

WANTED BY U.S. MARSHALS	
NOTICE TO ARRESTING AGENCY: Before arrest, validate warrant through National Crime Information Center (NCIC). United States Marshall Service NCIC entry number: (NCR) <u>WJ21460021</u>).	
NAME:	MITNICK, KEVIN DAVID
AKS(S):	MITNIK, KEVIN DAVID MEHRILL, BRIAN ALLEN
DESCRIPTION:	
Sex:	MALE
Race:	WHITE
Place of Birth:	VAN NUYS, CALIFORNIA
Date(s) of Birth:	08/06/63; 10/18/70
Height:	5'11"
Weight:	190
Eyes:	BLUE
Hair:	BROWN
Skintone:	LIGHT
Scars, Marks, Tattoos:	NONE KNOWN



Kevin Mitnick

Very sophisticated TCP-spoofing attack

The attack exploited the trust between hosts

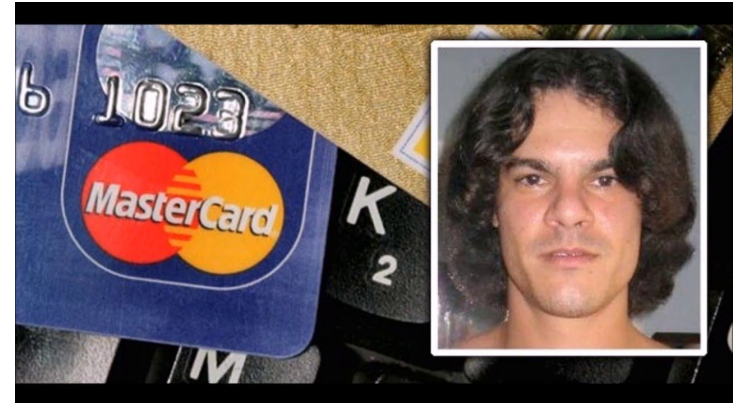
- x-terminal: diskless SPARC station running Solaris 1
- server: host providing boot image to x-terminal
- x-terminal allows unauthenticated login (and command execution requests) coming from server

The attack started with a Denial-of-Service attack against the server

The attack then progressed by impersonated the server wrt the x-terminals

Albert Gonzales' Case

- Responsible for running a credit card underground exchange (Shadowcrew)
- Shadowcrew was shut down by the FBI, but Albert escaped conviction by collaborating with law enforcement
- While working for the FBI he hacked into TJX companies and stole millions of credit card numbers
 - Used a composition of SQL injection and ARP spoofing attacks
- In 2010 he was sentenced to 20 years.



placeholder

For the plethora of high-profile hacking incidents

Hacking



What is a Hacker, anyway?

- First used at MIT in the 60s to describe “computer wizards”
- It has been eventually used to denote malicious hackers, that is, people that perform intrusions and misuse computer systems

Someone who lives and breathes computers, who knows all about computers, who can get a computer do anything. Equally important is the hacker's attitude. Computer programming must be a hobby, something done for fun, not out of a sense of duty or for the money.

(Brian Harvey, University of Berkley)

Ethics

- Is malicious hacking legal? NO
- Is it legal to discuss vulnerabilities and how they are actually exploited? YES, and it is a good thing, provided that...
 - The goal is to educate and increase awareness
 - The goal is to teach how to build a more secure computing environment
- A full disclosure policy has been advocated by many respected researchers, provided that:
 - The information disclosed has been already distributed to the parties that may provide a solution to the problem (e.g., vendors)
 - See: responsible vulnerability disclosure process (IETF Internet Draft)
 - The ultimate goal is to prevent similar mistakes from being repeated

Legal Hacking: Penetration Testing


Vulnerability analysis followed by exploitation

Assumptions and hypothesis derived from a black-box analysis

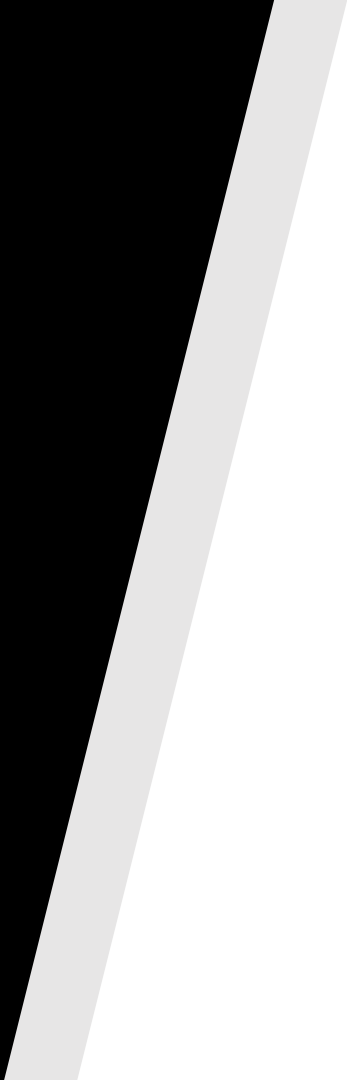
Pentesting is part of the larger security auditing/analysis process

Not a good way to ensure the security of a system

A comprehensive security analysis process takes into account many other aspects (e.g., source code analysis, policy analysis, social engineering)



Crypto (in-)securities



- We can try to attack the mathematical foundation of a cryptosystem
- If that doesn't work, we can try to attack the implementation



Side Channel Attacks



- We only want to sell even number of eggs
 - We want to use RSA to protect the orders
- (very sensitive information)

A parity problem

```
def check(c):  
    m = decrypt(c)  
    if is_even(m):  
        return "ok"  
    else:  
        return "err"
```

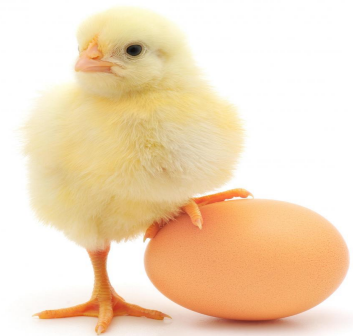
n = 15 (p = 3, q = 5)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----

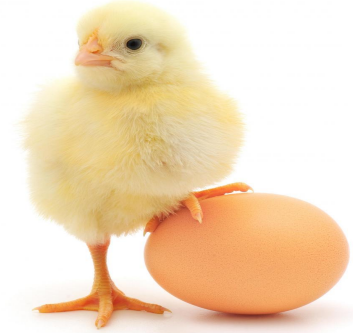


enc(m) →

← ok



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----



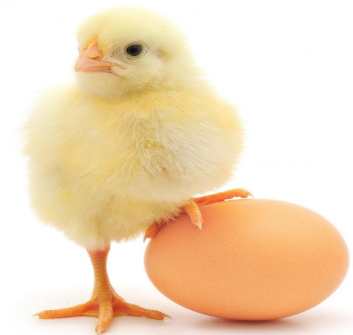
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----

m even



$\text{enc}(2 \cdot m)$

ok



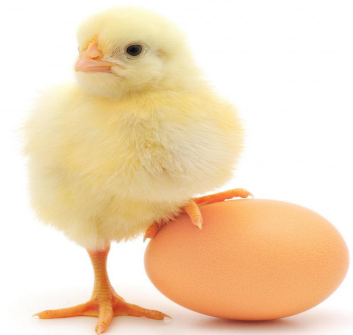
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----

Adaptive Ciphertext Attack



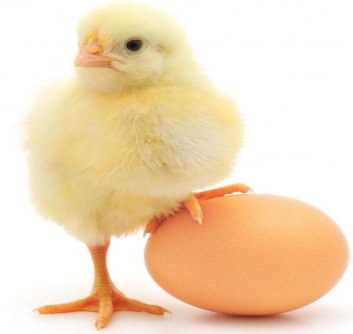
$\text{enc}(\textcolor{red}{2} \cdot m)$

ok



$2m$

$2m - n$



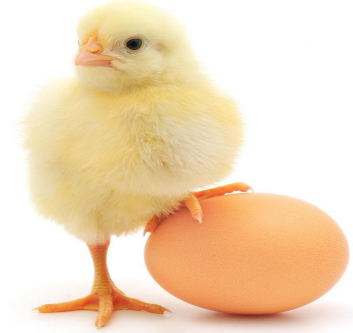
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----

$$m \in \{0, 2, 4, 6\}$$

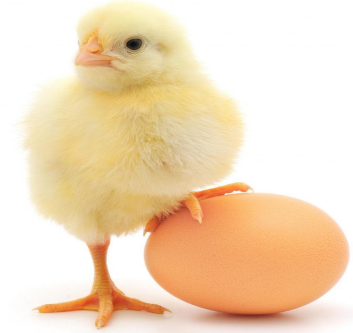


$\text{enc}(4 \cdot m)$

err



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----



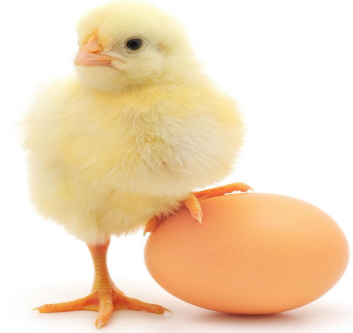
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----

$$m \in \{4, 6\}$$

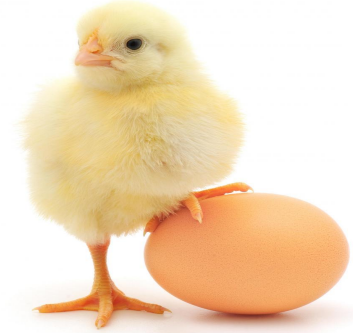


$\text{enc}(8 \cdot m)$

ok



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----

$$m = 4$$



How can we change the message?

$$enc(m) \rightarrow enc(2m)$$

$$(2^e \bmod_n) \cdot (m^e \bmod_n) = (2m)^e \bmod_n$$

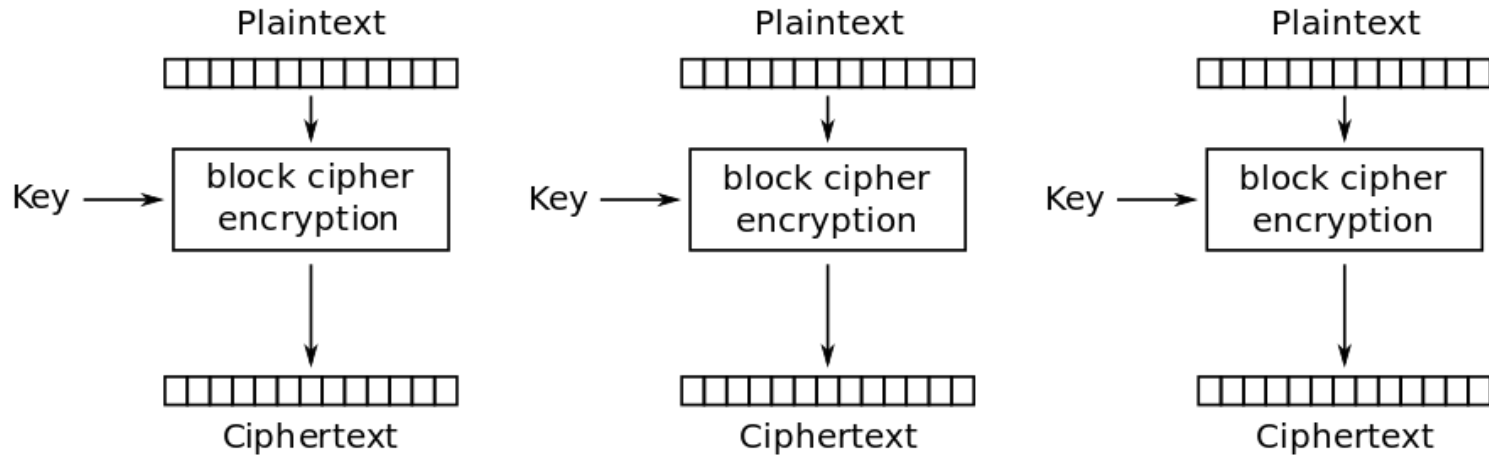
$$enc(2m) = enc(2) \cdot enc(m)$$

Multiplicative Property of RSA

Can we only hack farms?

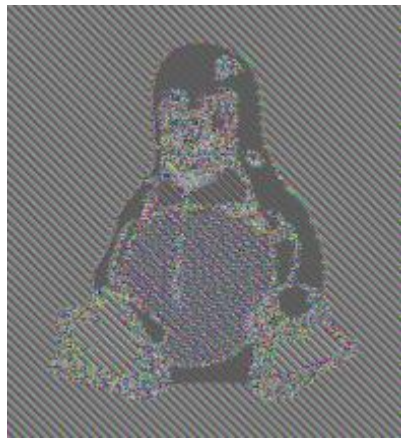
0002	RANDOM PAD	00	MESSAGE
-------------	-------------------	-----------	----------------

Broken by Bleichenbacher Attack (1998)



Electronic Codebook (ECB) mode encryption

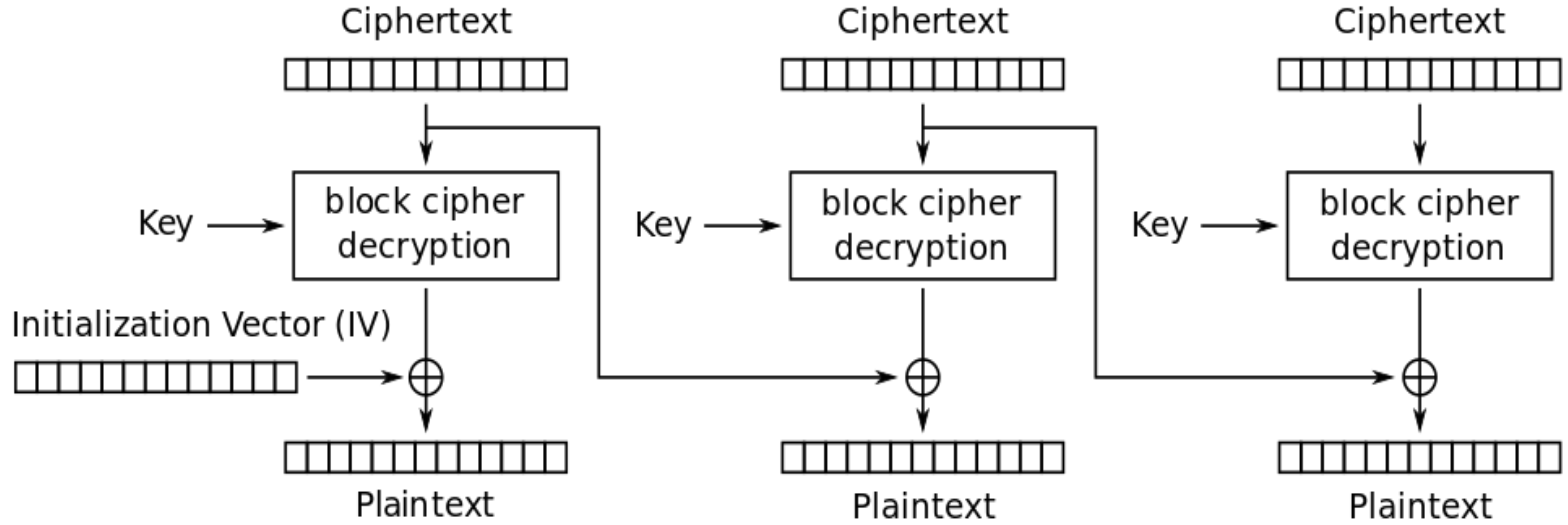
Electronic Codebook



ECB



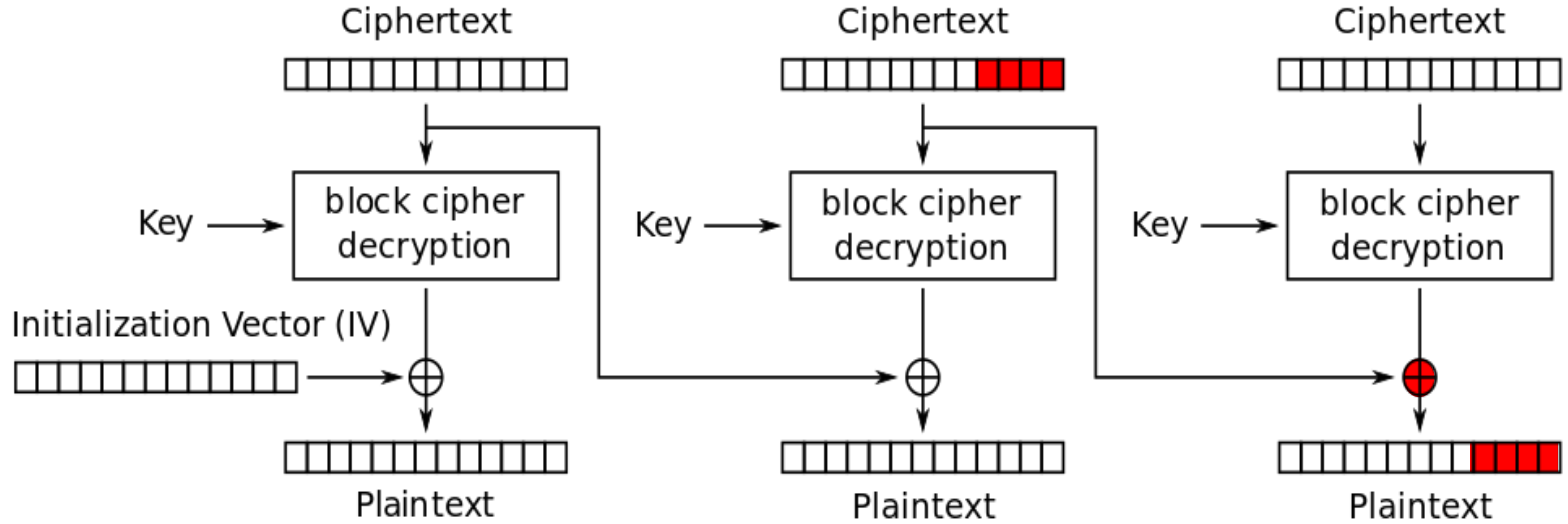
CBC



Cipher Block Chaining (CBC) mode decryption

Cipher Block Chaining

```
def cbc_mac(c):  
    m = decrypt(c)  
    if !pad_ok(m):  
        return "pad error"  
    if !mac_ok(m):  
        return "mac error"  
    ...
```

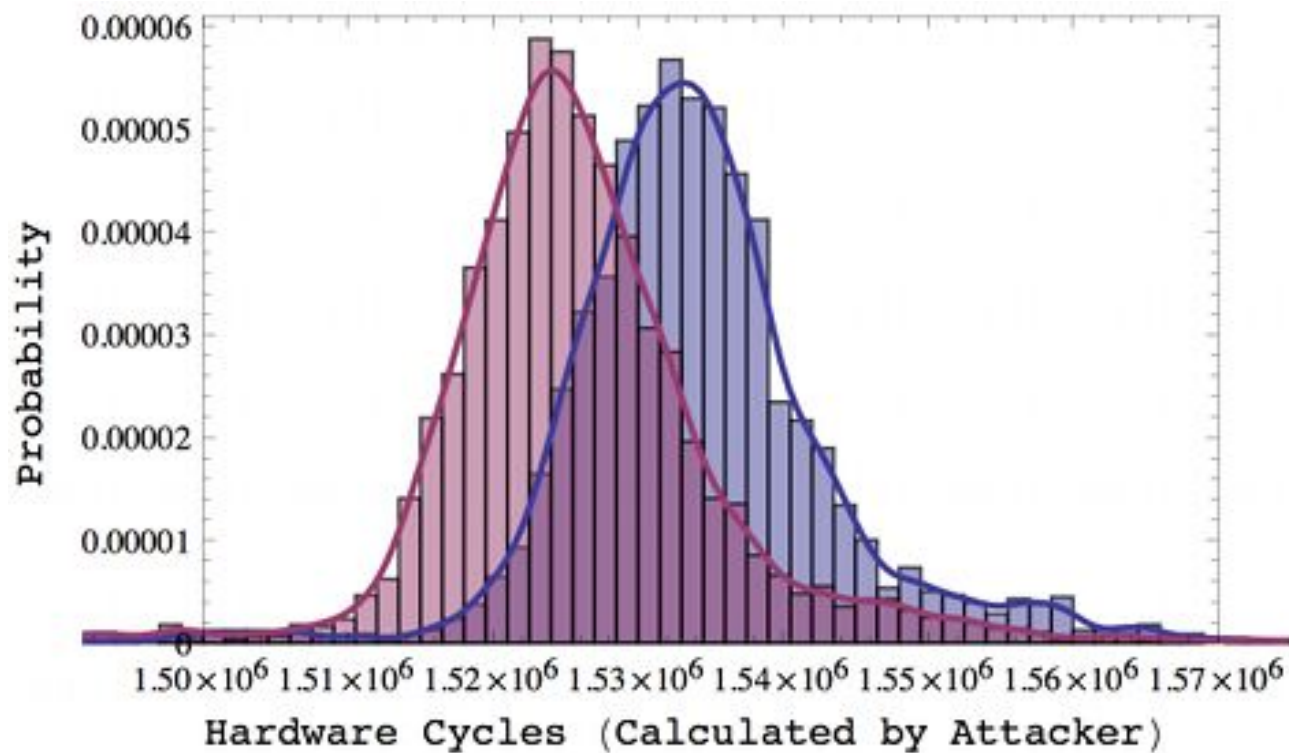


Cipher Block Chaining (CBC) mode decryption

https://www.infobytesec.com/down/paddingoracle_openjam.pdf

Padding Oracle Attack


```
def cbc_mac(c):  
    m = decrypt(c)  
    if !pad_ok(m) or !mac_ok(m):  
        return "error"  
    ...
```



Timing Attack

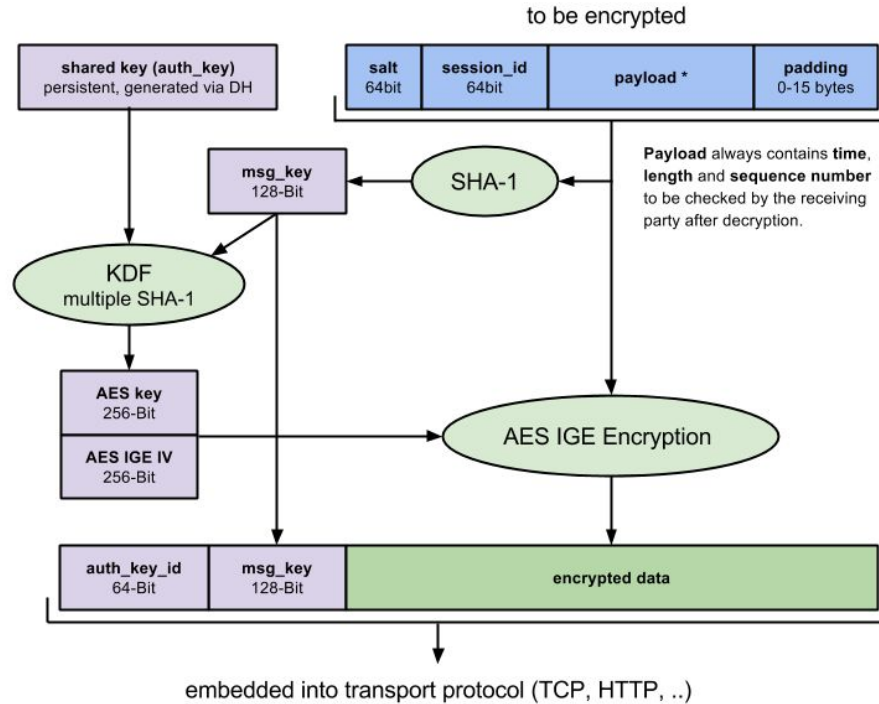
```
def cbc_mac(c):  
    m = decrypt(c)  
    if or(!pad_ok(m), !mac_ok(m)):  
        return "error"  
    ...
```

*"Never ever implement
your own cryptosystem"*

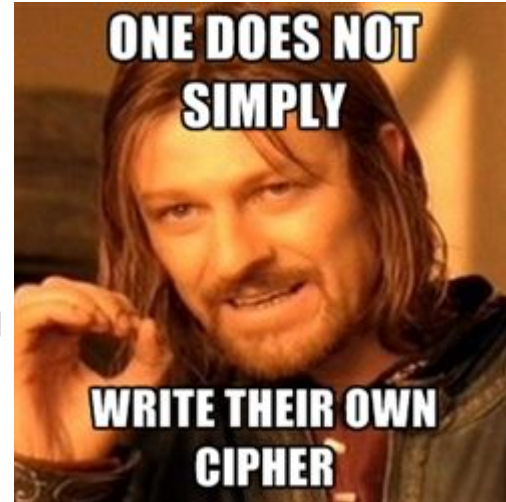
(Dan Boneh)



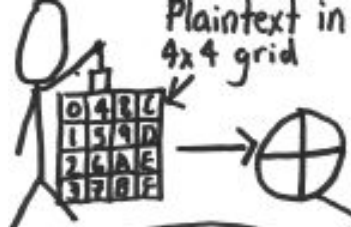
MTPROTO encryption



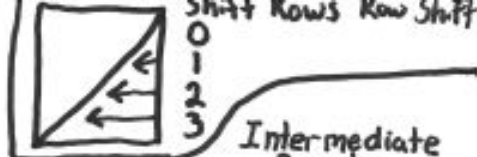
NB: After decryption, **msg_key** MUST be equal to SHA-1 of data thus obtained.



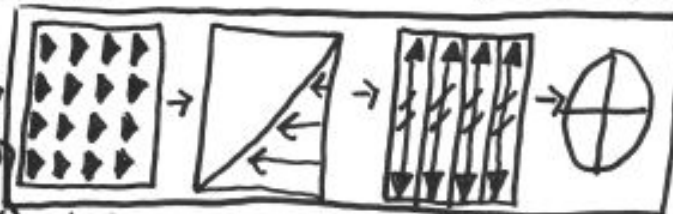
MTPROTO



AES Crib Sheet
(Handy for memorizing)



General Math
1.1B = AES Polynomial: $m(x)$
Fast Multiply
 $x^8 + x^4 + x^3 + x + 1$
 $x \cdot a(x) = (a \ll 1) \oplus (a_7 = 1) ? 1B : 00$
 $\log(x \cdot y) = \log(x) + \log(y)$
Use $(x+1) = 03$ for log base



Intermediate Rounds

#	Key
9	128
11	192
13	256



Final Round

?	?	?	?
?	?	?	?
?	?	?	?
?	?	?	?

Ciphertext

S-Box (SRD)

$SRD[a] = f(g(a))$

$g(a) = a^{-1} \text{ mod } m(x)$

For Think $53 \oplus 63^T$

5 is and 3 is $[0110 \ 0011]^T$

1111	000	a_7	0
0111	100	a_6	0
0011	110	a_5	0
0001	111	a_4	0
1000	111	a_3	0
1100	111	a_2	0
1100	111	a_1	0
1110	001	a_0	1

Key Expansion:



Other Columns:

S	B2	E1
0	6E	21
M	CB	86
E	B7	F2

Prev Col \oplus Col from Previous round key

Mix Columns:

2	1	1	3
2	1	1	3
3	2	1	1
1	1	3	2

$\begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}$

Inverse Mix

E	B	D	9
9	E	B	D
D	9	E	B
B	D	9	E

$\begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}$



Network Security



Network Sniffing

- Technique at the basis of many attacks
- The attacker sets his/her network interface in promiscuous mode
- Many protocols (FTP, POP, HTTP, IMAP) transfer information in clear
- Tools to collect, analyze, and reply traffic
- Routinely used for traffic analysis and troubleshooting
- Command line-tools:
 - tcpdump: collects traffic
 - tcpflow: reassembles TCP flows
 - tcpreplay: re-sends recorded traffic
- GUI tools:
 - Wireshark
 - Provides parsers for many protocols



Let's Encrypt



<https://www.yourdomain.com>

Sniffing

Spoofing

ARP spoofing

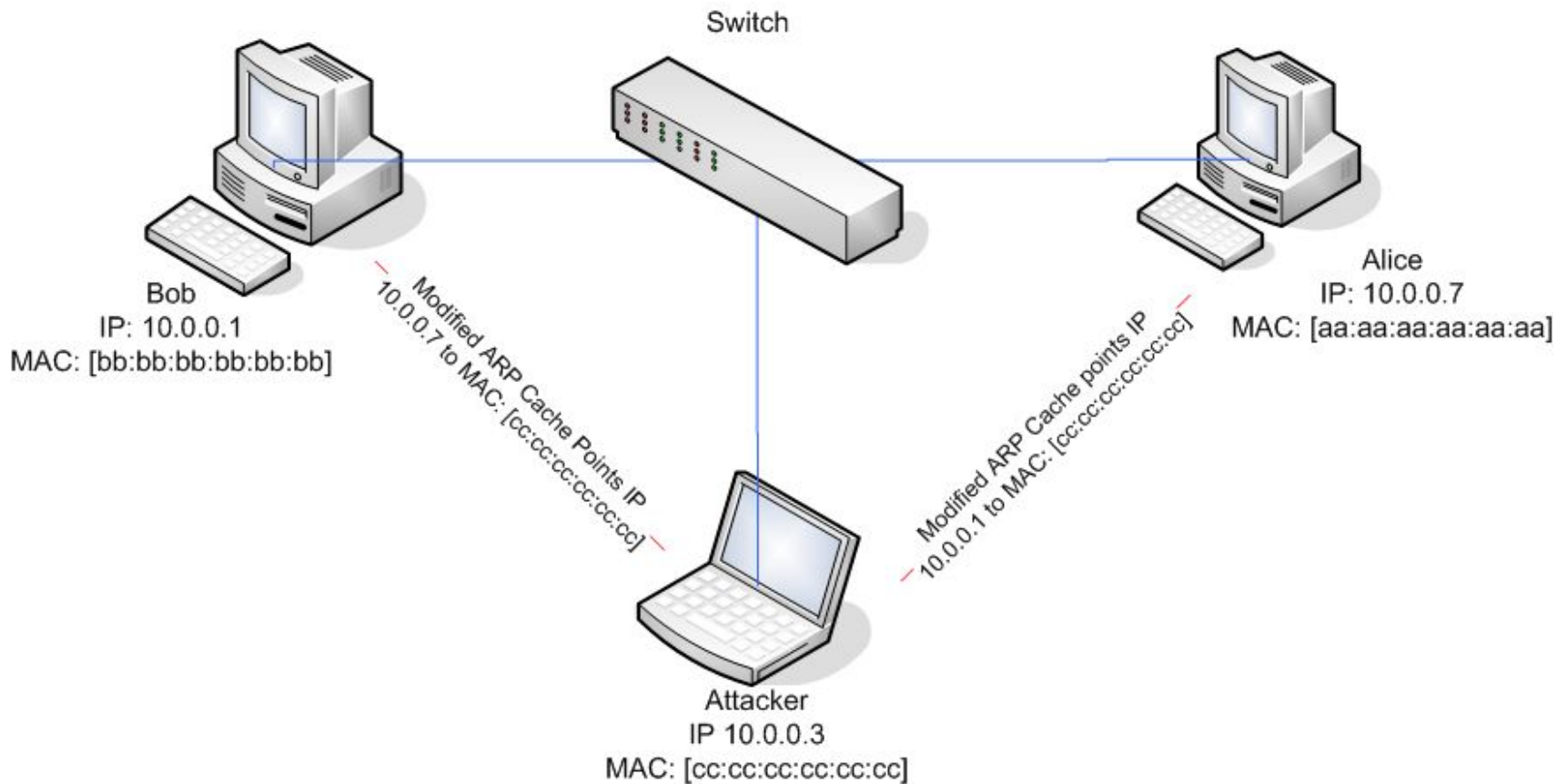
- The attacker sends wrong ARP replies to set himself as the other party
- Sniff all traffic between two host (man-in-the-middle)
- Tools:
 - Dsniff
 - Ettercap

IP Spoofing

- Forge a packet with the source IP address spoofed



Man In The Middle Attack



Man In The Middle Attack

Switched Environments

- Switched Ethernet does not allow direct sniffing
- MAC flooding
 - MAC address / port mappings
 - In some cases, flooding the switch with bogus MAC address will overflow the table's memory and revert from switch to hub
- MAC duplicating / cloning
 - Attacker configures her host to have the same MAC
 - The traffic is duplicated

Defenses

- Static ARP entries
- Ignore unsolicited ARP replies
- Monitor changes (arpwatch)
- Firewalls
- HTTPS