650.445/600.454 Practical Cryptographic Systems

Introduction

Instructor: Matthew Green

Intro

- What is a Practical Cryptographic System?
  - A security system
  - Uses cryptography
    - Many fascinating ways to get it wrong!
  - “Practical”:
    People actually use it & depend on it

DVD-Cracking Teen Acquitted

Associated Press 01.07.03

Bluetooth™

Bluetooth Hacking - How to hack a mobile phone using "Super Bluetooth Hack" (video tutorial)

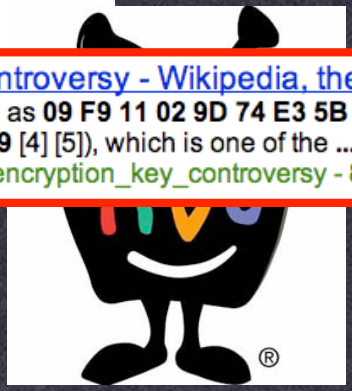
Cell phone, VoIP technologies lack security, experts say

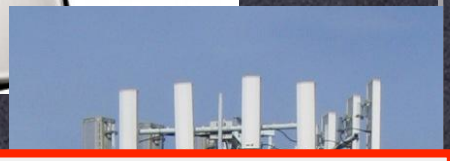
[AACS encryption key controversy - Wikipedia, the free encyclopedia]

... represented in hexadecimal as **09 F9 11 02 9D 74 E3 5B D8 41 56 C5 63 56 88 C0** [2] [3] (commonly referred to as **09 F9** [4] [5]), which is one of the ...

en.wikipedia.org/wiki/AACS_encryption_key_controversy - 87k - Cached - Similar pages -

Blu-ray Disc™





New attack cracks WEP in record time

The fact that 104-bit WEP has been cracked is in itself not newsworthy. The speed with which a new attack works is.

Researchers claim GSM calls can be hacked on cheap

Hacking Contactless Payment Cards

OpenSSL

Motivation

- Building (successful) systems requires more than cryptographic expertise
  - Though it's a prerequisite!
- It's cross-disciplinary:
  - Crypto
  - Information Security
  - Software Engineering
  - Hardware Engineering
  - UI, Policy, etc...

This class

- Not a traditional course in Cryptography
  - We'll cover the basics, but quickly
- Practice-oriented tutorial
  - examine how systems fail
  - how we can design against it
  - what can't we design against
- Driven by your questions & the news

What you'll come away with

- A grounding in cryptographic techniques
  - Strengths & weaknesses, applicability
  - A feel for the design/evaluation process
  - Introduction to standards (e.g., FIPS)
  - Enough to know where to look for more
- Knowledge of our own limitations
  - Building secure systems is hard (even for experts)

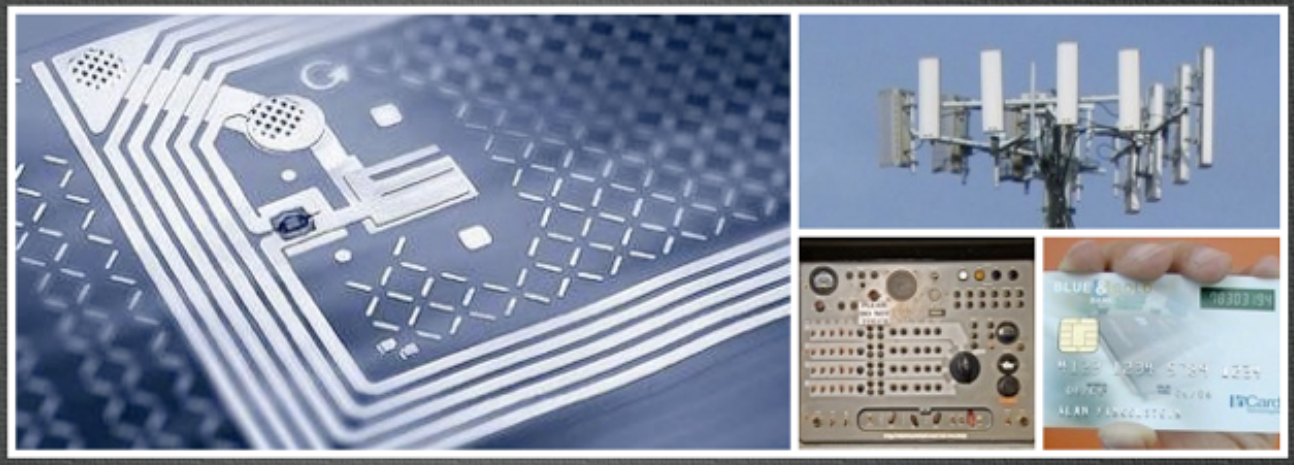
Grading, Text

- Grading Policy:
  - 40% Exams (Midterm & Final)
  - 40% Assignments
  - 10% Presentation
  - 10% Class participation
- Text:
  - Anderson: Security Engineering
    (an older version is online, see website)
- Website: spar.isi.jhu.edu/~mgreen/650.445

Programming

- The assignments in this class involve programming
  - I'll give you some latitude in languages (except for code review)
  - It's your responsibility to give us working assignments that compile/run
  - Anything other than a working assignment is a failure

650.445: PRACTICAL CRYPTOGRAPHIC SYSTEMS



READINGS & SUGGESTED PRESENTATION TOPICS

The following links elaborate on topics that will be covered in lecture, as well as many topics that would make excellent presentation subjects. It's by no means a complete list, and will be continuously updated– if you'd like to add something, please let me know.

Protocol Attacks

- Crosby, Goldberg, Johnson, Song, Wagner: Cryptanalyzing HDCP (2001)
- Schuler, Tews, Weinmann: Security of DECT (presentation only)
- Kohno: Analysis of WinZip Encryption
- Stubblefield, Ioannidis, Rubin: Breaking WEP

Side Channel Attacks

- Bar-el: Introduction to Side Channel Attacks (white paper)

Course Guidelines

- Do:
  - Read the news!
    Schneier, Slashdot, SecurityFocus, etc.
  - Bring up interesting topics & recent attacks you'd like to learn more about
- Don't:
  - Cheat***
  - Get me arrested

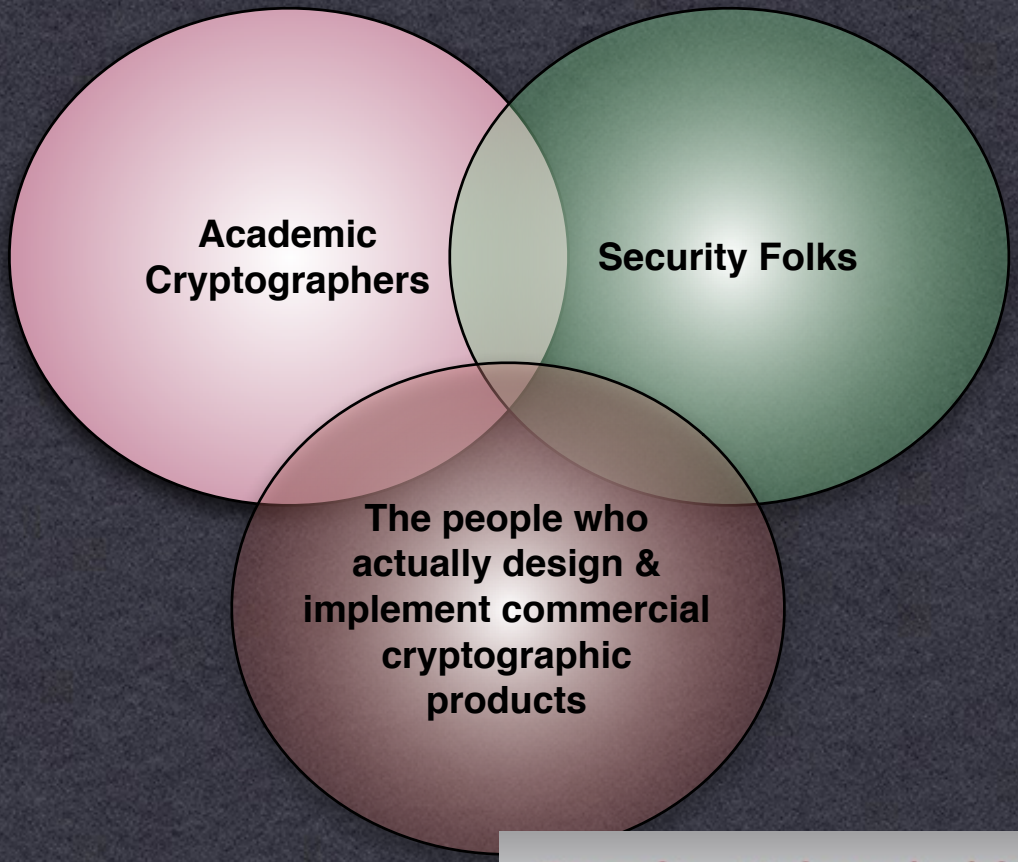
Readings

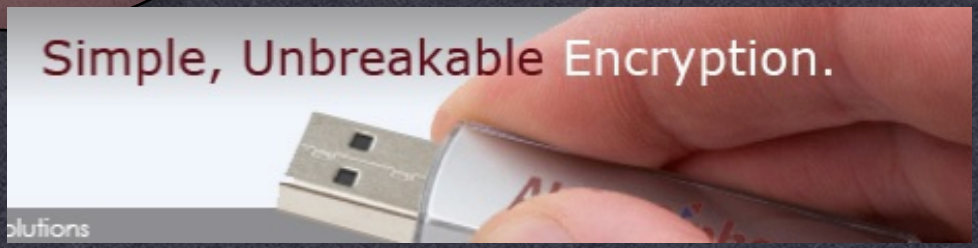
- Assigned each week
  - You must read them, be prepared to discuss in class
  - These will be on the exams
  - I may require you to write a summary of the materials -- we'll see how it goes

Incidentals

- Course mailing list
  - Goes to your @JHU.edu email
- Conferences
  - I might not be here, someone else probably will
  - I'm going to assign reading!

Today


Academic Cryptographers
Security Folks
The people who actually design & implement commercial cryptographic products


Simple, Unbreakable Encryption.

Security Failure

- When systems fail:
  - Researchers get published
  - $$$ lost
  - Private information compromised
  - People die (?)

Wireless Hackers Suspected In TJ Maxx Breach

By Martin H. Bosworth
ConsumerAffairs.com

May 7, 2007
Cyber-thieves using a telescoping wireless antenna to intercept payment information may be responsible for the "biggest data breach ever," investigators theori

Hackers can attack heart devic

Wireless implants vulnerable, UW scientist say

By TOM PAULSON
P-I REPORTER

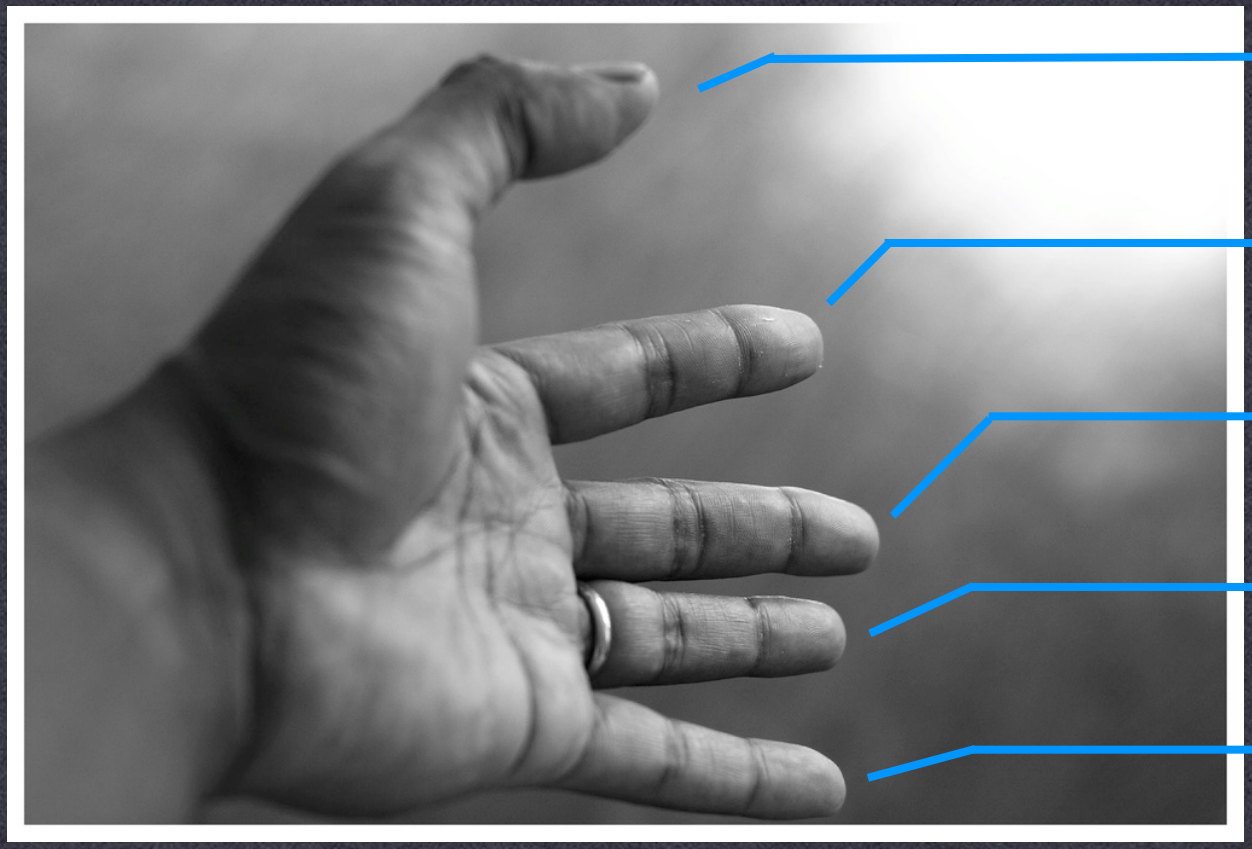
A Seattle computer scientist who helped expos
mess with electronic voting machines is part c
shown how new, wireless cardiac devices imp
heart patients also are vulnerable to electroni

Adding Math to List of Security Threats

By JOHN MARKOFF
Published: November 17, 2007

SAN FRANCISCO, Nov. 16 — One of the world's most prominent cryptographers issued a warning on Friday about a hypothetical incident in which a math error in a widely used computing chip places the security of the global electronic commerce system at risk.

Adi Shamir, a professor at the Weizmann Institute of Science in Israel, circulated a research note abou the problem to a small group of colleagues. He wrote that the



Concept

Primitives

Protocols

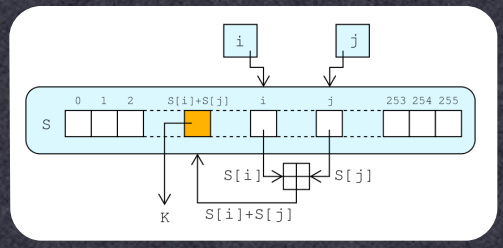
Implementation

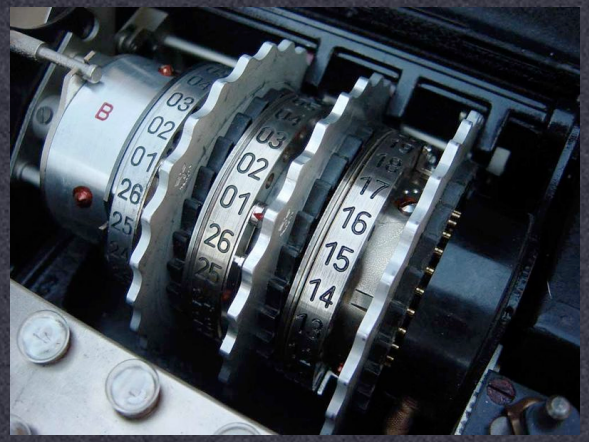
Usage

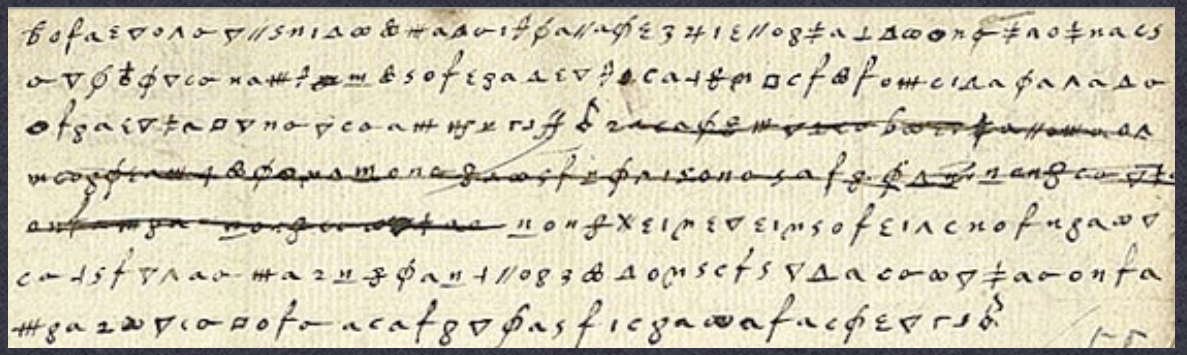
Primitives

Primitives

- Codes, ciphers, encryption schemes, MACs, etc.:
  - Classically, an attack on the "system" meant an attack on the primitive
  - History is littered with broken primitives





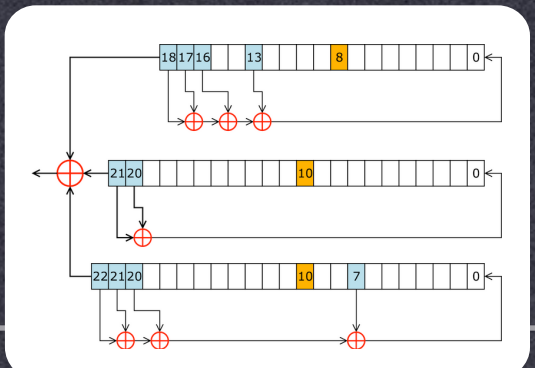


Primitives

- Practical Example: GSM encryption
  - A5/0: No encryption
  - A5/1: Based on LFSRs
  - A5/2: Weakened A5/1
  - A5/3 (KASUMI): New for 3G

GSM™
The wireless evolution

T-Mobile



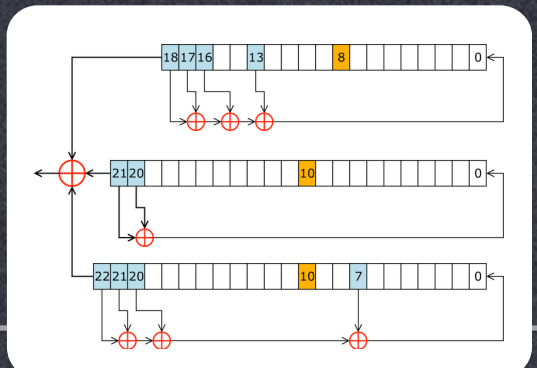
at&t

Primitives

- Practical Example: GSM encryption
  - A5/0: No encryption
  - A5/1: Broken
  - A5/2: Way Broken
  - A5/3 (KASUMI): Dented
    (and 3G vuln. to protocol attacks)
- Deliberately weak cipher design
  - Cost & politics

GSM™
The wireless evolution

T-Mobile



at&t

Primitives

- Practical Example: GSM encryption
  - A5/0: No encryption
  - A5/1: Broken
  - A5/2: Way Broken
  - A5/3 (KASUMI): De
    (and 3G vuln. to pr
- Deliberately weak
  - Cost & politics

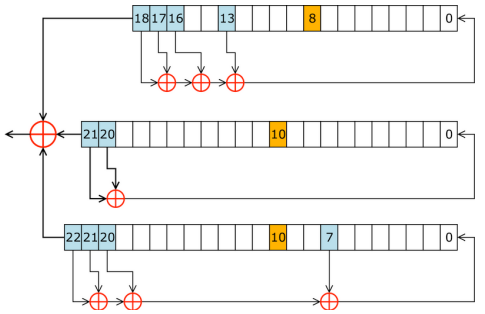
The New York Times

Cellphone Encryption Code Is Divulged

By KEVIN J. O'BRIEN
Published: December 28, 2009

BERLIN — A German computer engineer said Monday that he had deciphered and published the secret code used to encrypt most of the world's digital mobile phone calls, saying it was his attempt to expose weaknesses in the security of global wireless systems.

T-Mobile



at&t

Primitives

- Practical Example: GSM encryption
  - A5/0: No encryption
  - A5/1: Broken
  - A5/2: Way Broken
  - A5/3 (KASUMI): Dented
    (and 3G vuln. to protoco
- Deliberately weak ciph
  - Cost & politics

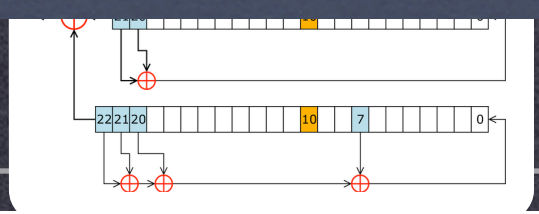
threatpost
The Kaspersky Lab Security News Service

January 11, 2010, 4:57PM

A Second GSM Cipher Falls

A group of cryptographers has developed a new attack that has broken Kasumi, the encryption algorithm used to secure traffic on 3G GSM wireless networks. The technique enables them to recover a full key by using a tactic known as a related-hey attack, but experts say it is not the end of the world for Kasumi.

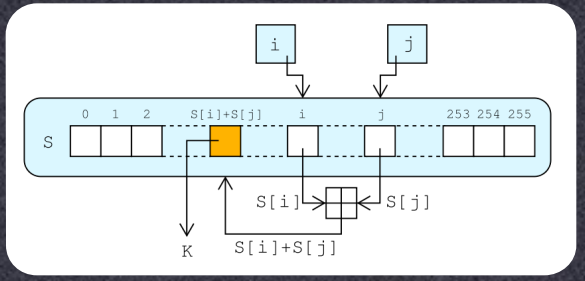
T-Mobile

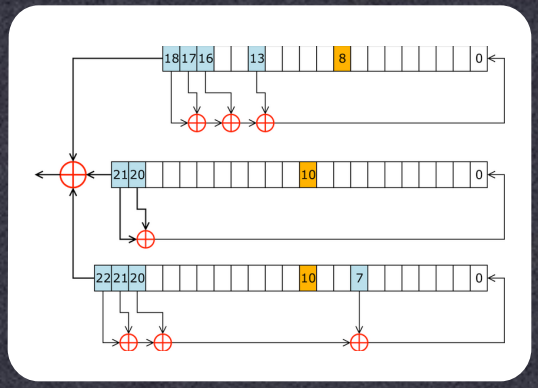


at&t

Primitives

- Typical problems:
  - Using the wrong ones (& homebrew crypto)
  - Or using the right ones... wrong
    - E.g., RC4 in WEP





Virtual Matrix Encryption (VME) is a data security method and apparatus that provides an exceptional degree of security at low computational cost. The data security arrangement differs from known data security measures in several fundamental aspects. Most notably, the content of the message is not sent with the encrypted data. Rather, the encrypted data consists of pointers to locations within a virtual matrix, a large (arbitrarily large), continuously-changing array of values.

Primitives

- Sometimes the “right” primitives stop being right...
  - The great Hash Function Adventure of 200X (MD5 broken, SHA1 -- sort of)

MD5 considered harmful today

Creating a rogue CA certificate

Primitives

- Sometimes the “right” primitives stop being right...
  - More recently:

Schneier on Security

A blog covering security and security technology.

« Risks of Cloud Computing | Main | Nuclear Self-Terrorizati

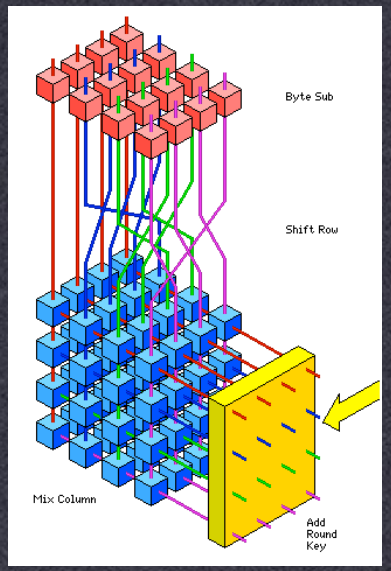
July 30, 2009

Another New AES Attack

A new and very impressive attack against AES has just been announced.

Over the past couple of months, there have been two (the second blogged about here) new cryptanalysis papers on AES. The attacks presented in the papers are not practical -- they're far too complex, they're related-key attacks, and they're against larger-key versions and not the 128-bit version that most implementations use -- but they are impressive pieces of work all the same.

This new attack, by Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, and Adi Shamir, is much more devastating. It is a completely practical attack against ten-round AES-256:



AES Image Source: http://www.quadibloc.com/crypto/co0404.htm

Protocols

Protocols

- Classical cryptographic protocol:

Encrypted Message

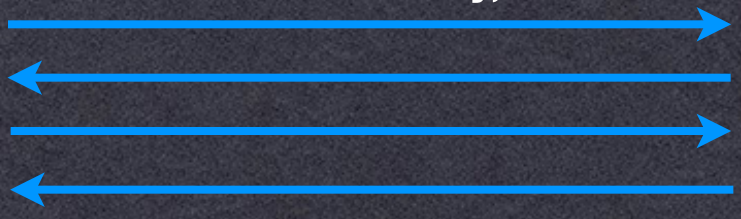
Attacker

Caesar, Centurion: Wikipedia. Barbarians by Flickr user skitzianist used under a Creative Commons license.

Protocols

- **Modern cryptographic protocol:**

Key Exchange, Validation,
Content Delivery, etc.



Attacker

Protocol examples:

- Vehicle remote control/immobilizer
  - Only legitimate owner can start the car/ unlock the doors, etc.

Protocol examples:

- Vehicle remote control/immobilizer
  - Early systems used fixed Serial Number

(SN)

Hello, SN

(SN)

Protocol examples:

- Vehicle remote control/immobilizer
  - Early systems used fixed Serial Number
  - Vulnerable to “replay attack”

(SN)

Hello, SN

Hello, SN

(SN)

Protocol examples:

- Solution: Challenge-Response
  - "Identification Friend or Foe"
  - Key is never broadcast over the air

(SN, k)

Challenge

MAC(k, Challenge)

(SN, k)

MITM

- Man in the Middle Attack
  - Route communications between car & keyfob
  - Don't have to break the protocol --- just abuse it

(SN, k)

Challenge

MAC(k, Challenge)

Challenge

MAC(k, Challenge)

(SN, k)

MITM

- Not just theoretical...
  - Anderson [Chap 2]
  - Military radars use a similar technique to identify friendly aircraft
  - How do we fix this?

S.A. Plane

Challenge

MAC(k, Challenge)

S.A. Radar

Challenge

MAC(k, Challenge)

Cuban Attacker

Round Trip Timing

- The case of DTCP-IP
  - Content transport protocol
  - Concern: prevent user from sharing content over the Internet
  - Ensure that Sink is within 7ms of Source

Source

Key Exchange

Challenge

Response

Verify ≤7ms RTT

Sink

Round Trip Timing

Establish Shared Key K

For attempt N = 0 to 1023

N, C = HASH1(K + N)

R = HASH2(K + N)

Check that R= HASH2(K + N)
and response time within 7 ms
(If not, retry)

Ok, I'm happy!

Check that C = HASH(K, N)

This check happens
way too late!

Implementation

Implementation

- Sadly, this is where most systems fail
  - Particularly if they're software-based

Oracle Security Alert #37
Created: 1 August, 2002
Updated: 5 August, 2002
Updated: 9 August, 2002
Updated: 24 September, 2002

OpenSSL Security Vulnerability

Description:

There are remotely exploitable buffer overflow vulnerabilities in OpenSSL versions prior to 0.9.6e.
These vulnerabilities may allow a remote attacker to execute arbitrary code or perform a denial-of-service (DoS) attack.

Console Hacking 2008: Wii Fail

Is implementation the enemy of design?

marcan and bushing
Team Twiizers

⚠ Vulnerability in Citrix Presentation Server could result in cryptographic settings not being correctly enforced

Implementation

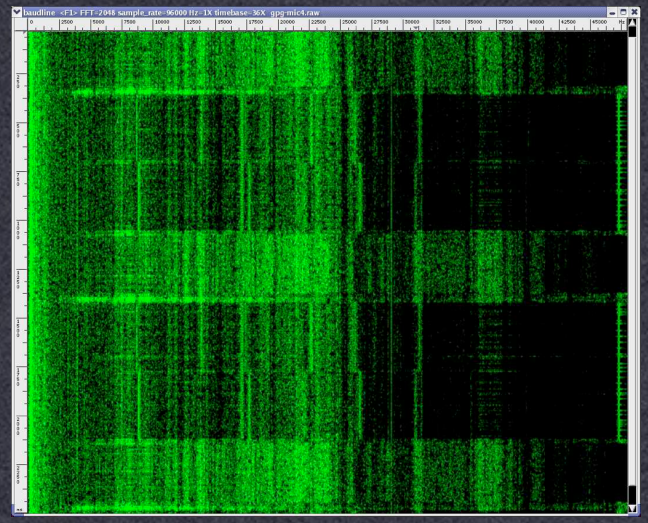
Console Hacking 2008: Wii Fail

Is implementation the enemy of design?

marcan and bushing
Team Twiizers

- Typical problems:
  - Poor protocol implementation
  - Bad PRNGs
  - Software vulnerabilities
  - Untrusted platforms
  - Side channel attacks
  - Weak hardware

USN-612-2: OpenSSH vulnerability



Source: Tromer, Acoustic Cryptanalysis: http://people.csail.mit.edu/tromer/acoustic/

Software

- Routine coding errors
  - Use strcmp() instead of memcmp()
  - Don’t check your buffer bounds
  - Don’t check your malloc() responses
  - Code anything secure on Windows
  - Write your own OpenSSL
  - Use the real OpenSSL...

Software

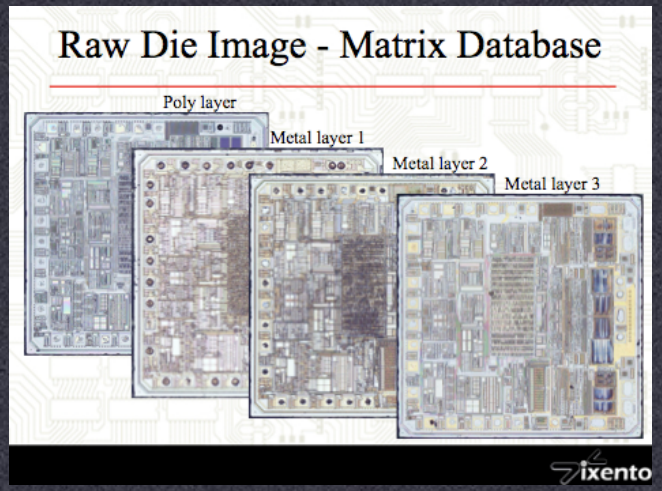
- More sophisticated issues
  - Which cryptographic libraries to use?
  - How to manage keys?
    (hint: not like this)

```
#define DESKEY ((des_key*)"F2654hD4")
```

  - Will keys be booted out into swap?

Hardware

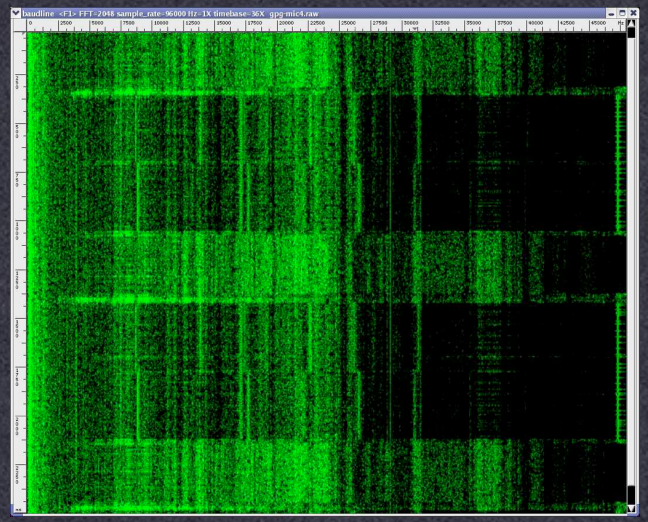
- May lead to a false sense of security
  - The notion that the bad guys can't crack open/ reverse engineer your system
- Tamper-evidence
  - Detect malicious activity
- Tamper-Resistance
  - Better
  - Depends on who's tampering, and how.



Thanks Quing Wang and iXento.

Side-Channel Attacks

- Even when perfectly implemented
  - System can leak information through a "side channel":
    EM, power consumption, audio, timing
  - E.g., recovering RSA keys via low-bandwidth audio -- using a cellphone!



Usage

Usage

- Unfortunately, users may be your greatest foe
  - Weak password choices,
    refusal to change defaults
  - Insistence on backdoors, fail-open mechanisms
  - Loss of key material, data
  - And so far we're talking about the honest users!

Usage

- Insider attacks:
  - Almost impossible to deal with
  - Ultimately relies on policy, vigilance
  - Where possible:
    - minimize trust
    - provide for renewability

Concept

Concept

- Certain things cannot be done
  - Perfect (software) DRM (i.e., user can watch/play/use the encrypted content, but can't decrypt it themselves)
  - Cryptographic software obfuscation
- Ok, if you understand:
  - These systems can at most slow down the attacker

Image source: http://chris.musgrave.org/projects/infiniteloop/index.html

Kerckhoffs' Principle(s)

- **Auguste Kerckhoffs (1835-1903)**

1. The system must be practically, if not mathematically, indecipherable;
2. It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience;
3. Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents;
4. It must be applicable to telegraphic correspondence;
5. It must be portable, and its usage and function must not require the concourse of several people;
6. Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

Kerckhoffs' Principle(s)

2. It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience;

"The enemy knows the System"
-- Claude Shannon's Maxim

Public Domain image from David Kahn's The Codebreakers, credited to "Eugen Drezen, Historio de la Mondo Lingvo (Leipzig, 1931), 102".

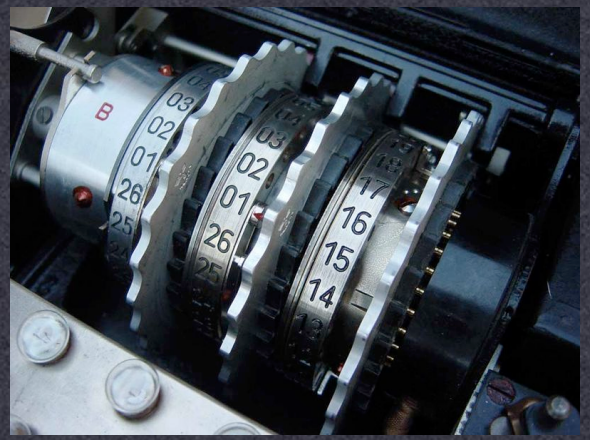
Don’t worry!

- I’m not all doom & gloom
  - We can do some things very well
  - Other things fairly well
  - Still others... well-ish
- We can certainly do better than most

Going forward:
The next few weeks

Part 1

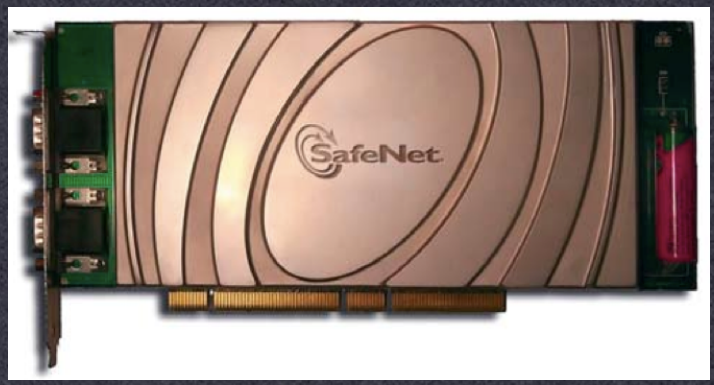
- Re-introduction to Crypto... at high speed:
  - Classical cryptography
  - Symmetric-key encryption & block ciphers
    - DES, the modes of operation
  - Public-key cryptography
    - Diffie-Hellman, RSA



Enigma image from Wikipedia, used under GFDL.

Part 2

- Exploiting Software:
  - Corrupting, overflowing and generally messing with software systems
- Physical security
  - Tamper-resistance
  - Hardware Security Modules
  - Fault attacks



Part 3

- Reductionist security & protocols
  - Proving the security of a construction
  - Analyzing protocols that fail
- Random number generation
- Security evaluation
  - What a security evaluation process looks like
  - The FIPS standards

A Note on Ethics

- We'll be discussing vulnerabilities in many systems
  - Some have been fixed
  - You might find more
  - It goes without saying: exploiting systems is often a crime. Be careful.
  - Important to disclose vulnerabilities responsibly

END