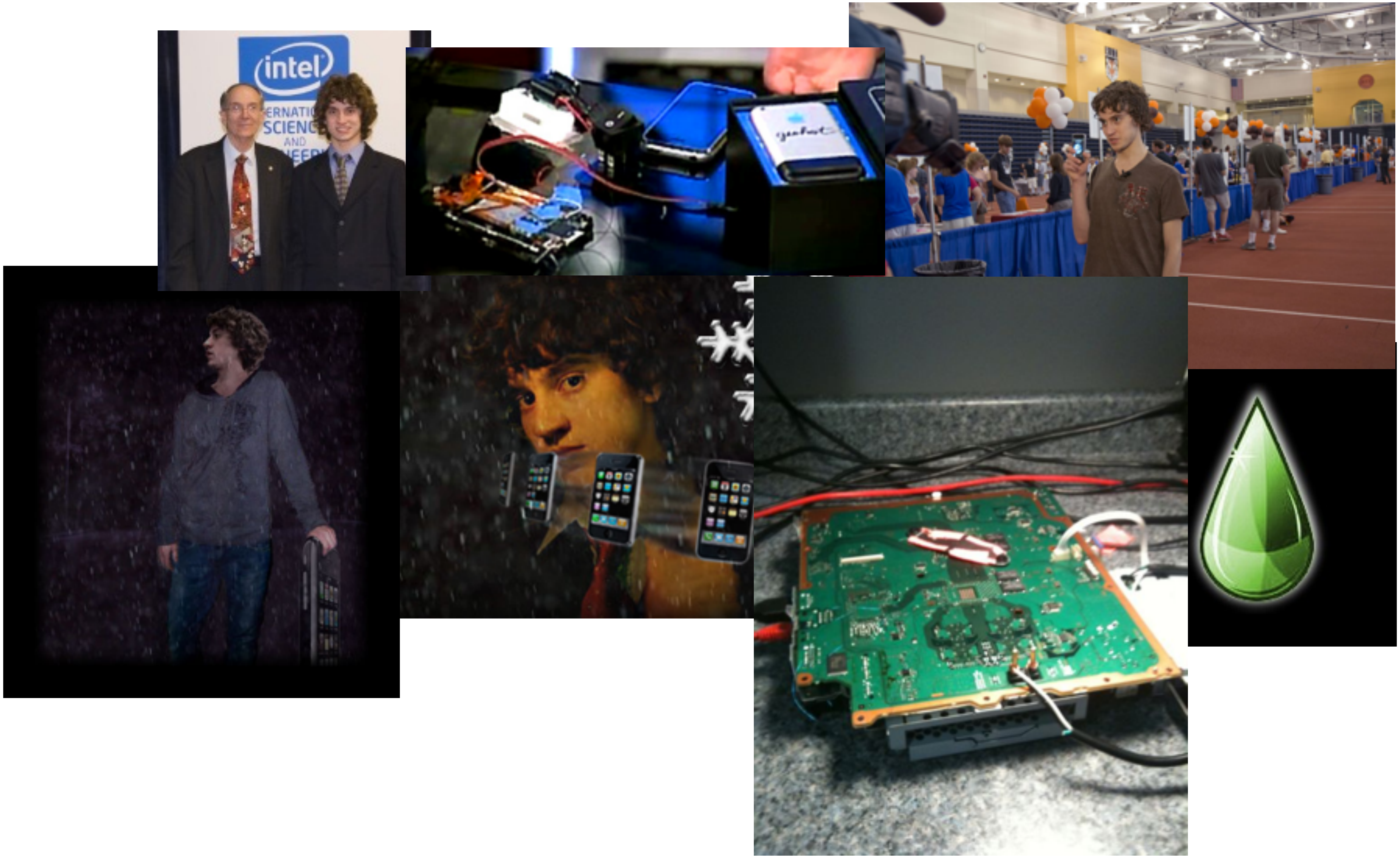
How to Break into Virtual Houses

George Hotz

Who am I?



What is security?

Allowing allowed things to do things while not allowing not allowed things to do things

The Locks on Virtual Houses

Cryptography

Can you read this?

KP ECUG QH UVCKTU, WUG HKTG.

Can you read this?

KP ECUG QH UVCKTU, WUG HKTG.

Algorithm is Caesar Cipher, Key is ‘B’

Can you read this?

KP ECUG QH UVCKTU, WUG HKTG.

Algorithm is Caesar Cipher, Key is ‘B’

IN CASE OF STAIRS, USE FIRE.

Monoalphabetic Substitution

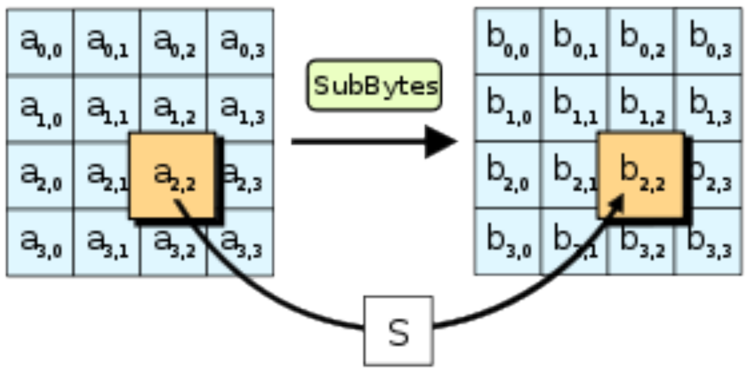
- {A,B,...Z} -> {P,L,...F}
- Frequency analysis
- Similar to codes used in WWII
- But letters are information, which can be expressed as numbers

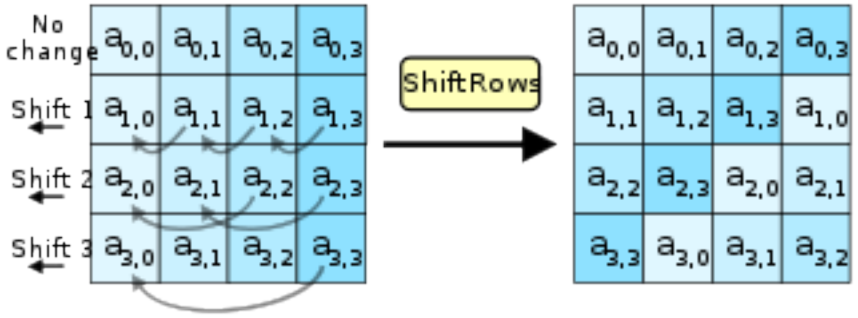
AES

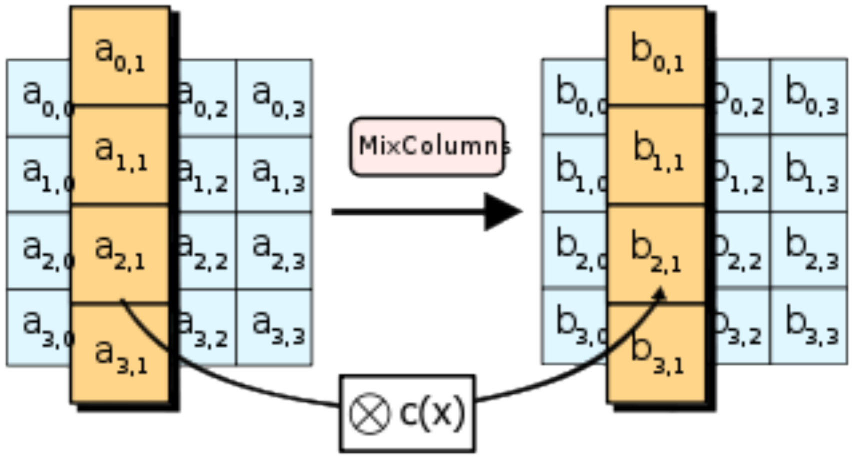
x and K are block size big

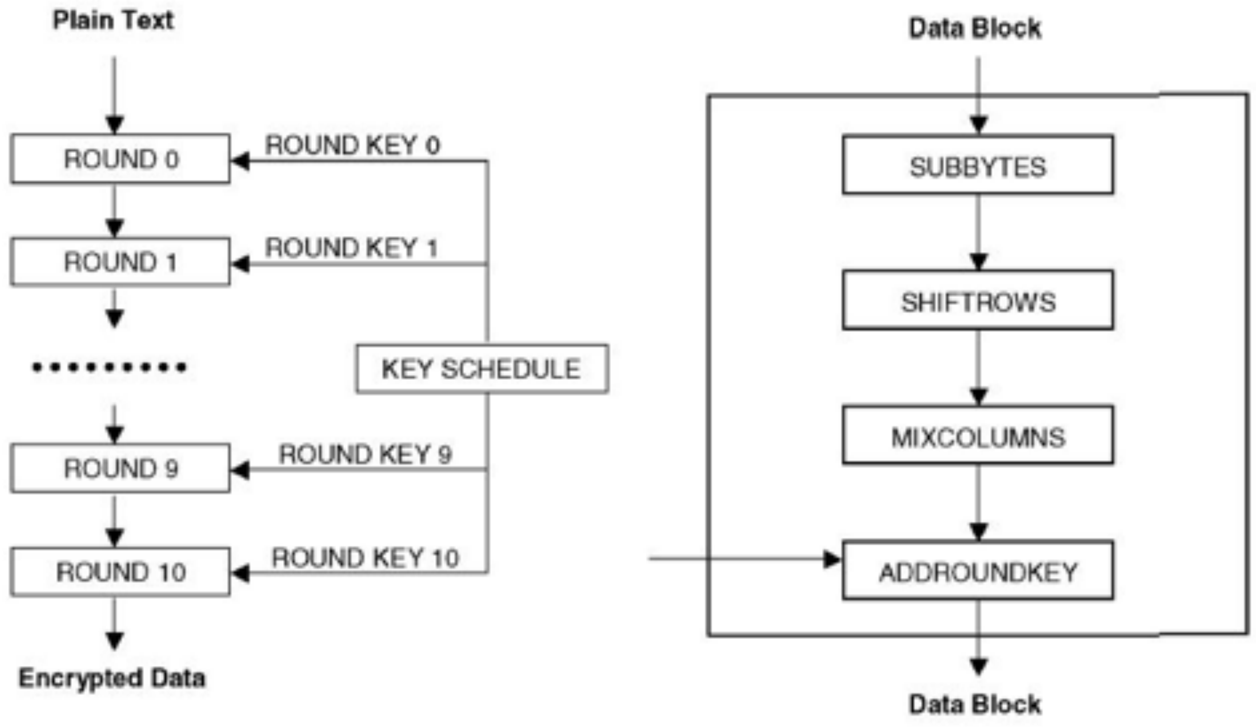
x -K> C(x)

C(x) -K> x


$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S


No change
Shift 1
Shift 2
Shift 3
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$


$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
MixColumns
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
$\otimes$ c(x)



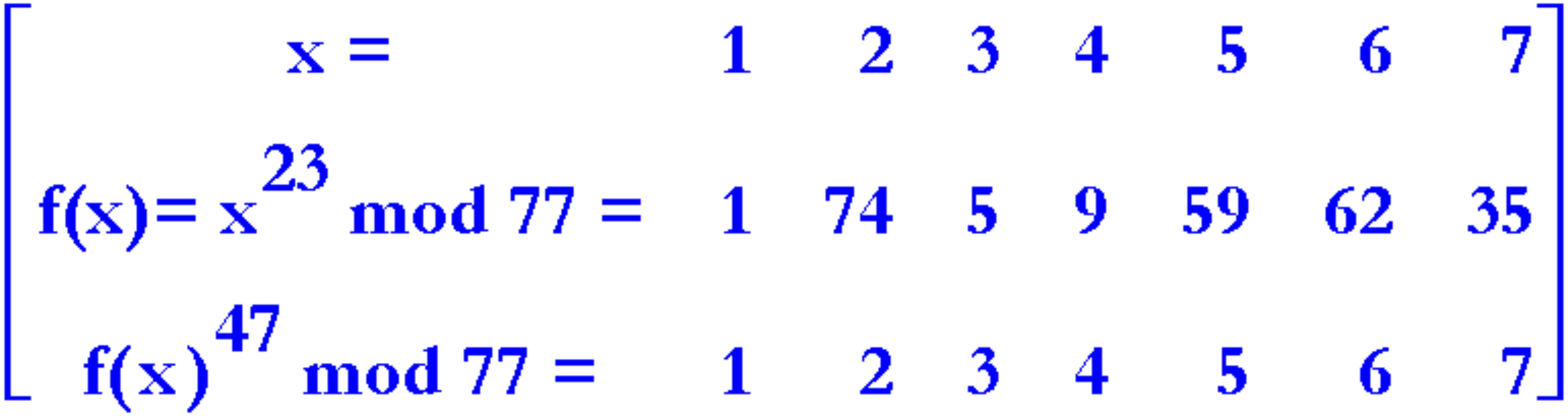
(a) (b)

Fig. 1. (a) The data-path for data block and key size of 128 bits, (b) generic structure of one internal round.

Key Exchange

- How do I get a key to you?
- Treasure chest metaphor
- public private key idea
  - you encrypt with my public key

RSA



$$\left[\begin{array}{rrrrrrrr} x = & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ f(x) = x^{23} \bmod 77 = & 1 & 74 & 5 & 9 & 59 & 62 & 35 \\ f(x)^{47} \bmod 77 = & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}\right]$$

p=7, q=11

77 = (7*11) is public modulus

(p-1)(q-1) = 60, big secret

47 is public key, gcd(47, 60) = 1

23 is private key, 47*x mod 60 = 1

by 47*x + 60*y = 1, x>0, y<0

Why secure?

Hash Functions

- SHA-1
- “hello” -> aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d
- x -> H(x) is easy
- H(x) -> x is hard
- How do you crack it?
    - Brute force
    - Rainbow tables

Recap

- Symmetric
    - Encryption where both sides know key
- Asymmetric
    - How do you move a key?
    - How do you trust people are who they claim to be?
- Hash Functions
    - Passwords, Integrity checks

How to make a jailbreak?

i.e. housebreaking

(wait that’s not right)

The Problem

- iPhone runs signed code
  - What is signed code? (show on board)
  - (you are all crypto experts right?)
- From boot
- Chains of trust

The Solution

- Look at inputs
- Send bad input to make magic

blackra1n

- Receive a usb_control_msg(0x21, 2)
  - memcpy(0, data, 0x2000);
- 0 is the exception vectors!
- Dropping payloads
- Hijacking the boot chain

Running up the chain

- Bad iBoot loads bad kernel
- Bad kernel loads bad applications
- Bad applications are fun

Crypto Oracles

- We encrypted everything you hacker scum!
- In the hardware even
- Secure?
- Rhymes with PS Tree

Snooze worthy security
for the lame webapp
your doomed startup
will create

Bunch of Noobs

- Data != Code, Data != Pointers
  - Stack overflows
  - SQL injection
  - XSS
- Check your inputs
  - Users are evil

Web Security is Boring

- If you really care I can explain
  - SQLi, XSS, XSRF
- Flash is a piece of crap
- So is Java
- So is IE
- So is PHP

Security

- You can’t win
- You can’t break even
- You can’t get out of the game

So join the Dark Side