Password Cracking

CS 1660: Introduction to Computer Systems Security

Identification and Authentication (recap)

- A subject should provide a unique identifier
- Authentication is the act of confirming the truth of an attribute of a datum or entity
- There are three authentication factors:
  - Knowledge: Something you know
  - Ownership: Something you have
  - Inherence: Something you are

Authorization and more…

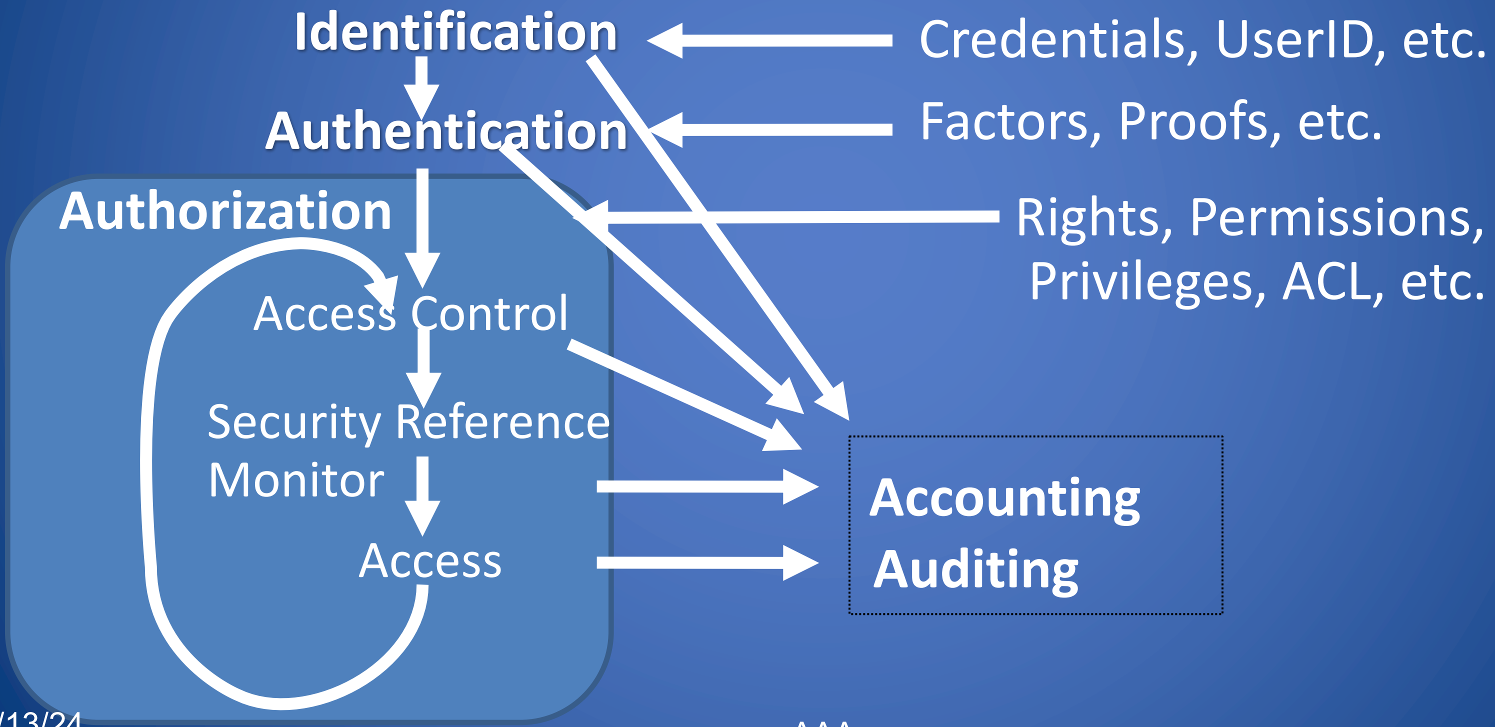
Authorization

- Once a subject is Authenticated, access should be authorized
- Authorization is the function of specifying access rights to resources (access control)
- More formally, "to authorize" is to define access policy: permissions, rights, etc.

AAA and more…

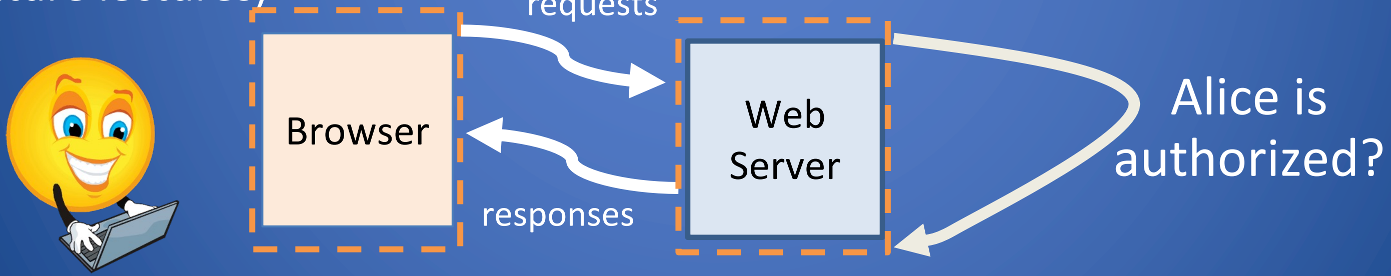
Identification, **Authentication**, **Authorization**, **Accounting**, Auditing

– AAA Working Group, IETF



Authorization on the web

- Alice logs in (i.e. authenticates)
- The web server is now aware of who is logged in
- Alice attempts to access a course
- The application checks to see if Alice has the authorization for the course…
  - If so, Alices receives the requested information
  - If not, Alice has a denied access response
- Authorization could be just for reading or writing or execute (more in the future lectures)

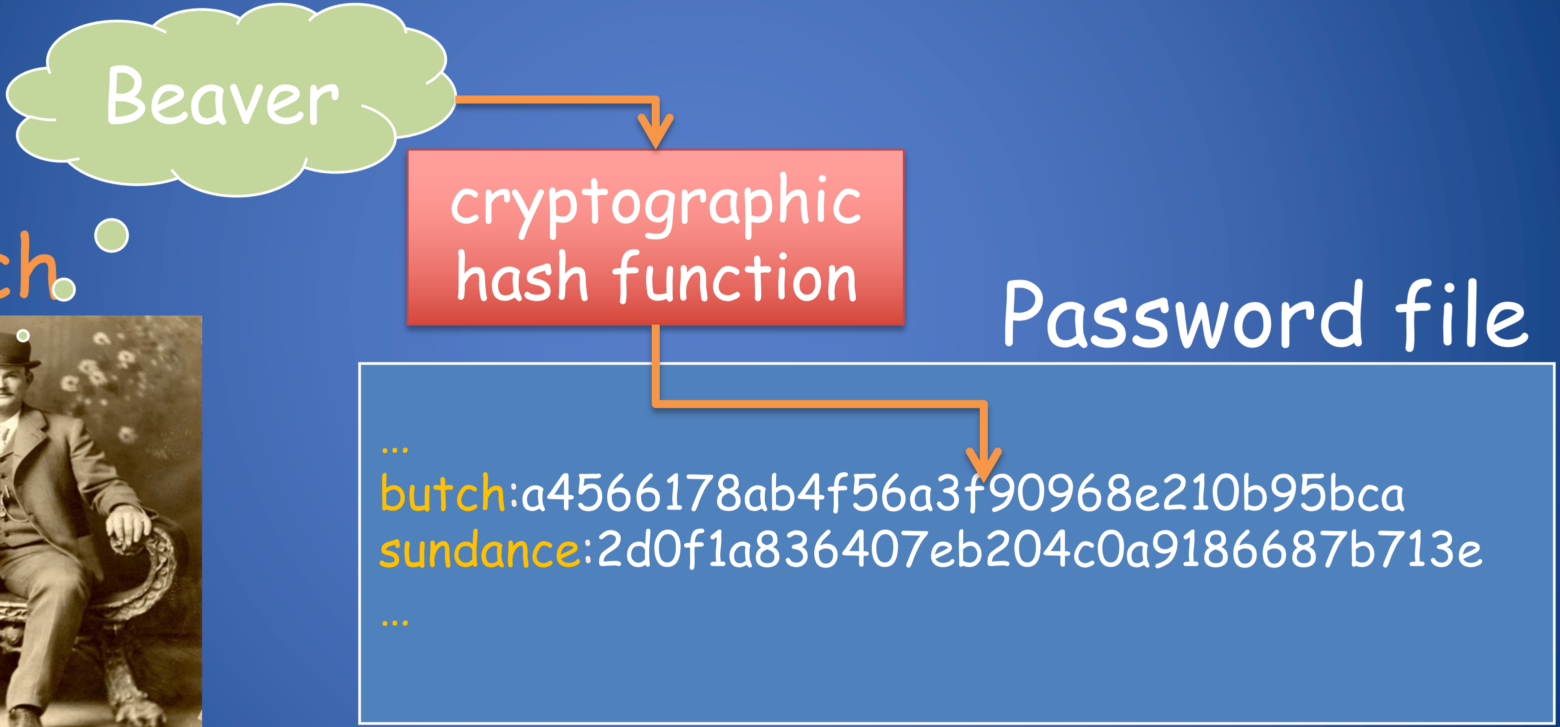


Alice has already logged in

Password Cracking

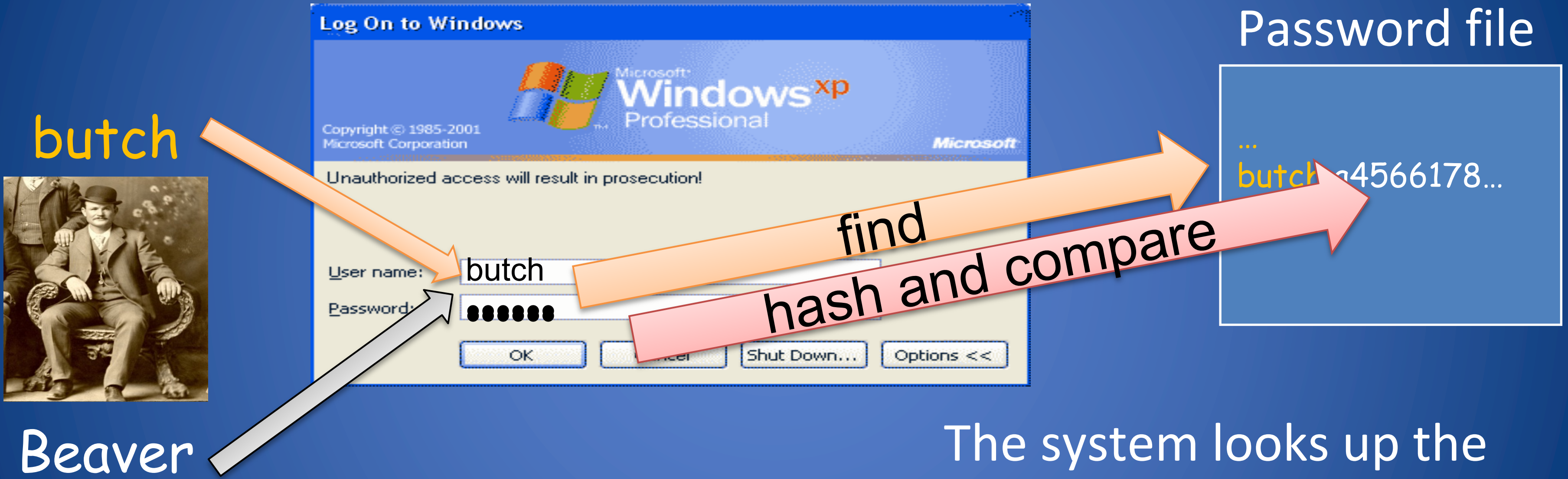
Storing Passwords

butch



```
…
butch:a4566178ab4f56a3f90968e210b95bca
sundance:2d0f1a836407eb204c0a9186687b713e
…
```

Standard Authentication in OS



The user inserts username and password into the login window

The system looks up the username compares the hash of the password with the stored hash

Most common scenario

- The hacker is able to get usernames and password hashes

Location of password file

- Windows (32 bit): C:\WINDOWS\system32\config\SAM
- Linux: /etc/passwd and /etc/shadow
- Mac OS X: /var/db/dslocal/nodes/Default/users/<username>.plist

Password Cracking Methods

- Brute force
  - Try all passwords in a given space
  - Eventually succeeds given enough time and CPU power
- Dictionary
  - Precompute hashes of a set of likely passwords
  - Store (hash, password) pairs sorted by hash
  - Fast look up for password given the hash
  - Requires large storage and preprocessing time
- Rainbow table
  - Partial dictionary of hashes
  - More storage, shorter cracking time

Cracking passwords with Hashcat

What is Hashcat?

Hashcat is the self-proclaimed free fastest password recovery tool.

Benchmarks:

```
./hashcat -b
```

Brute Force

- Try all passwords in a given space
  - Parallelizable
- Eventually succeeds given enough time and computing power
  - Best done with GPUs and specialized hardware (FPGAs, or Asic)
- Large computational effort for each password cracked

Brute Force Cracking

- The attacker has 60 days to crack a password by exhaustive search
- How many hash computations per second are needed?

– 5 characters:	1,415
– 6 characters:	133,076
– 7 characters:	12,509,214
– 8 characters:	1,175,866,008
– 9 characters:	110,531,404,750

Dictionary Attack

- Precompute hashes of a set of likely passwords
  - Parallelizable
- Store (hash, password) pairs sorted by hash
- Fast look up for password given the hash
- Requires large storage and preprocessing time

Setup: Dictionary Attack

STEP 1: Make a plaintext password file of bad passwords (called `wordlist`):

```
bernardo12345
letmein
zaq1zaq1
```

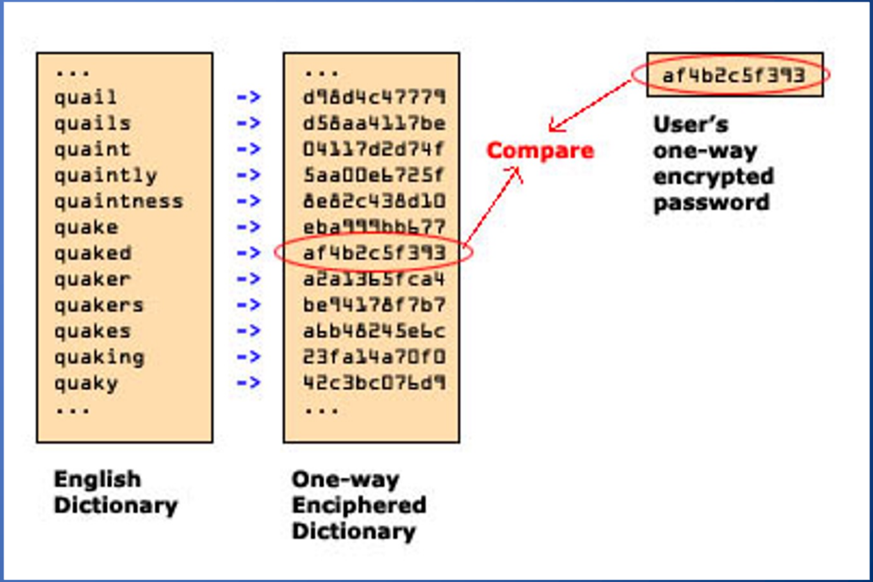
STEP 2: Generate MD5 hashes:

```
for i in $(cat wordlist); do
       echo -n "$i" | md5sum | tr -d " *-"; done > hashes
```

STEP 3: Get a dictionary file. (We’re using rockyou.txt which lists most common passwords from the RockYou hack a couple years back…)

Dictionary Attack

```
./hashcat -m 0 -a 0 hashes
rockyou.txt
```



Source: http://web.mit.edu/dheera/www/blur_dictattack.jpg

Hashtypes: -m [NUM]

0 : MD5 [8743b52063cd84097a65d1633f5c74f5]

100 : SHA1 [b89eaac7e61417341b710b727768294d0e6a277b]

1000 : NTLM

1400 : SHA256

Hash-Mode 1800 (sha512crypt $6$, SHA512 (Unix)

[$6$Lw5wXCCssJp3Ei8S$413/7AdrNLD.T/waBp61ItYXa0eUSzQQp3/iM.oiuTCecZAR79GBom2yJlTgC.5Q5p5DHInYY/9AXjRIQ5r6K1]

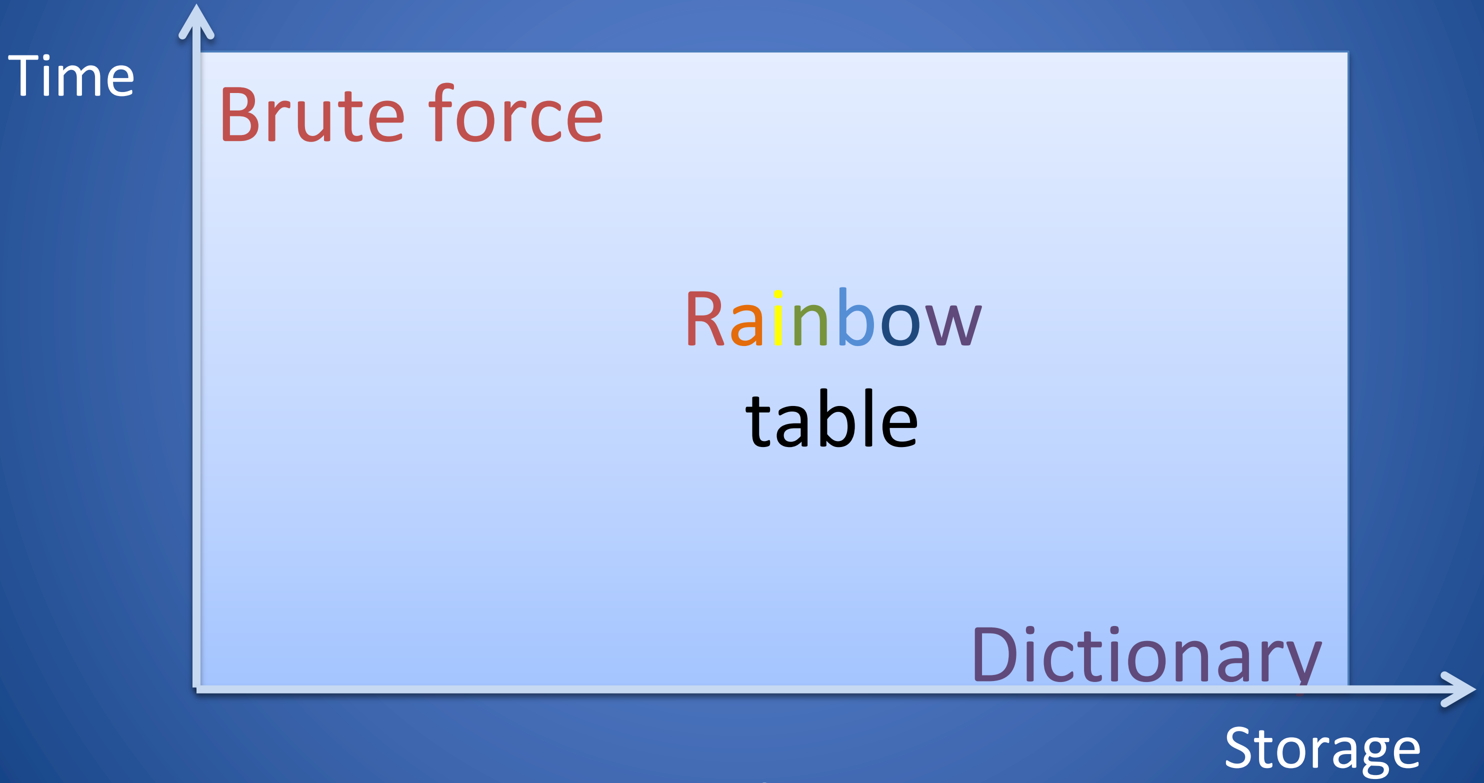
https://hashcat.net/wiki/doku.php?id=example_hashes

Time-memory Trade-Off

1980 - Martin Hellman

- –In this kind of trade-off, you reduce the time you need to crack a password by using a large amount of memory
  - •**Benefits**: It would seem more efficient to do the brute-forcing once, store the result, and then use this stored result to accelerate the cracking on any machine.
  - •**Flaws**: this kind of database takes tens of memory’s terabytes.

Password Cracking Tradeoff



Attack Modes: -a [NUM]

- 0: Straight (Dictionary attack)
  1: Combination (concatenating words from multiple wordlists)
  2: Toggle-Case (toggling case of characters. You can use rules for this)
  3: Brute-force (trying all characters from given charsets, per position)
  4: Permutation (dictionary generates permutations of itselt)
  5: Table-Lookup (Disabled in later hashcat implementations)
  8: Prince (Intelligent guessing implemented by hashcat)
- https://hashcat.net/wiki/doku.php?id=hashcat

Mask Attack (Brute-Force)

Built-in charsets:

?l = abcdefghijklmnopqrstuvwxyz
?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ
?d = 0123456789
?s = !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
?a = ?l?u?d?s
?b = 0x00 - 0xff

Run (try all lowercase passwords of length 5):

```
./hashcat -a 3 hashes ?l?l?l?l?l
```

Source: https://www.4armed.com/blog/perform-mask-attack-hashcat/

Rule Based Attack

If you have a list of passwords:
password
mysecret
qwerty

What kind of rules are there?

- You can create a rule file with these rules:

  sa@

Rule 1) Substitute 'a' for '@'
p@ssword

$1 $2 $3
Rule 2) Add 123 to end:
password123
mysecret123
qwerty123

$c
Rule 3) Capitalize first word:
Password
Mysecret
Qwerty

Source: https://www.4armed.com/blog/hashcat-rule-based-attack/

Rule Based Attack

Run:

```
./hashcat -m 0 hashes rockyou.txt -r ruledemo --debug-mode=1 --debug-file=matched.rule
```

Running `cat matched.rule` will show you all the rules that matched so you can build better rules.

There also lots of built-in rules provided in the default hashcat installation

Intelligent Guessing Methods

- Try the top N most common passwords
  - Check out several lists of passwords on Daniel Miessler's github
- Dictionary of words, names, places, notable dates
- Combinations of the above
- Replace and intersperse digits, symbols
- Syntax model
  - e.g., two dictionary words with some letters replaced by numbers: `elitenoob, e1iten00b, 31it3n00b, …`
- Markov chain model or a trained neural network

Intelligent Guessing

- A 10-character randomly selected password would take years of CPU time to crack
- For any scheme that involves guessing, the time to crack is reduced by guessing intelligently
- Key insight: not all passwords are equally likely
- Idea: try most likely passwords first

eHarmony Hack

In 2012, 1.5 Million passwords were stolen from eHarmony and published online. As a CS166 student you can now hack like the pros and recover some eHarmony passwords using hashcat.

eHarmony uses MD5: so try `-m 0`

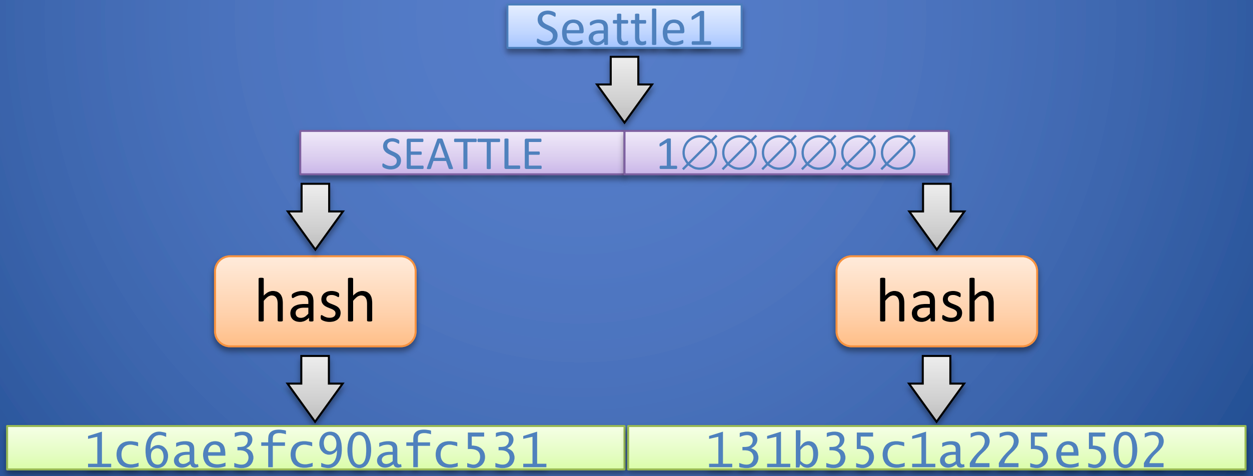
Warning before running command: some passwords in here are offensive

Source: http://www.adeptus-mechanicus.com/codex/hcateasy/hcateasy.php http://www.nydailynews.com/life-style/eharmony-passwords-hacked-1-5-million-users-dating-site-data-compromised-article-1.1091568

Server side:
Password algorithm issue

Windows XP Password Hashing

- LAN Manager Hash
  - Convert password to uppercase, truncated to length 14
  - Split into two 7-charcter halves
  - Compute 64-bit hash of each half



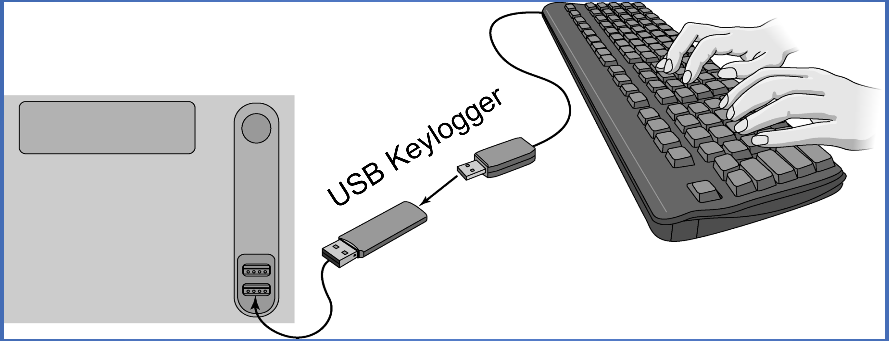
LAN Manager Hash Weaknesses

- Small password space
  - Equivalent to two uppercase passwords of 7 characters
  - About 6.7 trillion possible passwords
- Attack performed with rainbow table
  - 1.4GB storage
  - 14 seconds recovery time
  - 99.9% success rate

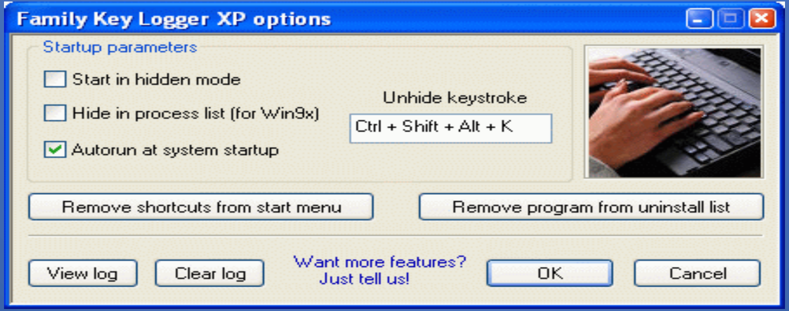
If Cracking does not Work

Other attacks…

Keyloggers



Hardware



Family Key Logger XP options

Startup parameters

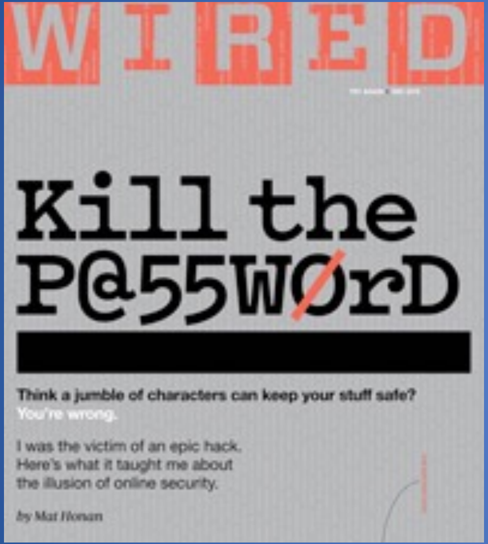
- [ ] Start in hidden mode
- [ ] Hide in process list (for Win9x)
- [x] Autorun at system startup

Unhide keystroke: Ctrl + Shift + Alt + K

Remove shortcuts from start menu
Remove program from uninstall list
View log
Clear log
Want more features? Just tell us!
OK
Cancel

Software

Password are still important..



WIRED

Kill the P@55WØrD

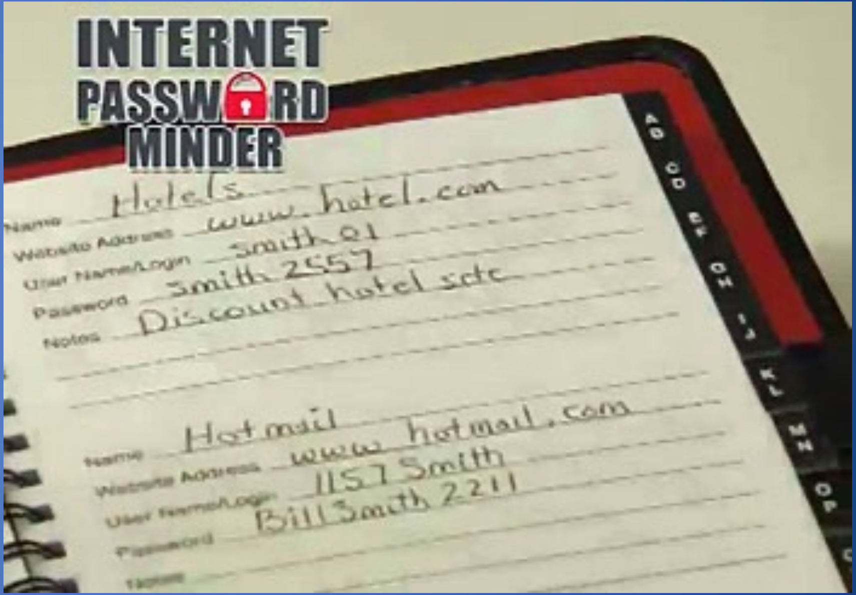
Think a jumble of characters can keep your stuff safe? You're wrong.

I was the victim of an epic hack. Here's what it taught me about the illusion of online security.

by Mat Honan

"Hackers destroyed my entire digital life in the span of an hour" 12 - 2012

Mat Honan Wired senior writer



INTERNET PASSWORD MINDER

Name	Hotels
Website Address	www.hotel.com
User Name/Login	Smith01
Password	Smith2557
Notes	Discount hotel site
Name	Hotmail
Website Address	www.hotmail.com
User Name/Login	1157Smith
Password	BillSmith2211
Notes	

- http://y2u.be/Srh_TV_J144

What We Have Learned

- Password authentication
  - Principles and attack vectors
- Password storage methods
  - Use salted hashes
- Password cracking
  - Brute-force
  - Dictionary precomputation
  - Intelligent guessing
- Demos
- Ethical and legal issues
  - Compelled password disclosure

BREAK!

5 4 3 2 1

Rainbow Table details

Tradeoffs

- Comparison of simple cracking methods
  - Brute-force: no preprocessing, no storage, very slow cracking
  - Dictionary: very slow preprocessing, huge storage, very fast cracking
- Rainbow tables
  - Tunable tradeoff between storage space and cracking time
  - More storage, faster cracking
- Cost comparison in terms of number of cryptographic hash values stored and computed

Method	**Storage cost**	**Preprocessing cost**	**Cracking cost**
Brute-force	~ 0	~ 0	n
Dictionary	n	n	~ 0
Rainbow table $mt^2 = n$	mt	mt^2	$t^2/2$

Reduction Function

- A reduction function maps a hash value to a pseudorandom password from a given password space
- Example reduction function p = R(x)
  - –Consider 256-bit hash values and 8-character passwords from an alphabet of 64 symbols $a_1, a_2, \ldots, a_{64}$
  - –Split x into 48-bit blocks $x_1, x_{2,} \ldots, x_5$ and one 16-bit block x_6
  - –Compute $y = x_1 \oplus x_2 \ldots \oplus x_5$
  - –Split y into 6-bit blocks $y_1, y_{2,} \ldots, y_8$
  - –Let $p = a_{y_1}, a_{y_{2,}} \ldots, a_{y_8}$
- Above method can be generalized to arbitrary password spaces

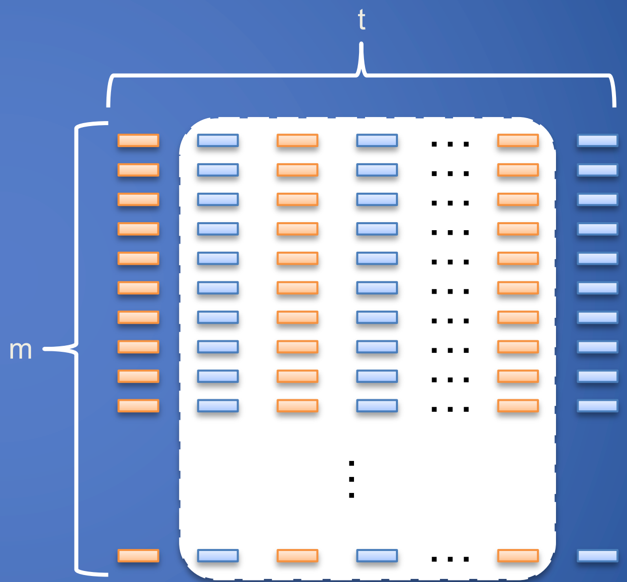
Password Chain

- Sequence of alternating passwords and hash values
  - Start with a random password p_1
  - Use cryptographic hash function H and reduction function R
  - $x_i = H(p_i)$
  - $p_{i+1} = R(x_i)$
  - End with a hash value x_t

$$\begin{tikzcd} \text{mycat6} \arrow[r, "H"] & \text{b56f19c8d4} \arrow[r, "R"] & \text{twotrains} \arrow[r, "H"] & \cdots \arrow[r, "H"] & \text{43be7901cd} \\ p_1 & x_1 & p_2 & & x_t \end{tikzcd}$$

Hellman’s Method

- Starting from m random passwords, build a table of m password chains, each of length t
- The expected number of distinct passwords in a table is $\Omega(mt)$
- Compressed storage:
  - For each chain, keep only the first password, p, and the last hash value, z
  - Store pairs (z, p) in a dictionary D indexed by hash value z



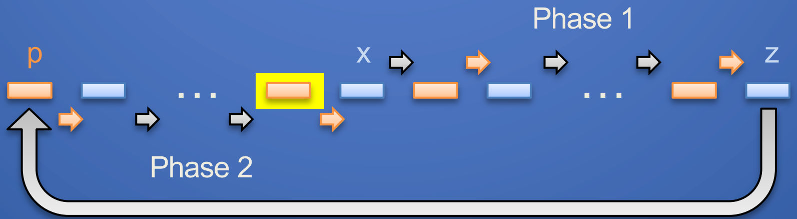
Classic Password Recovery

- Recovery of password with hash value x
- Step 1: traverse the suffix of the chain starting at x

```
– y = x
– while p = D.get(y) is null
    • y = H(R(y)) // advance
    • if i++ > t return “failure” // x is not in the table
```

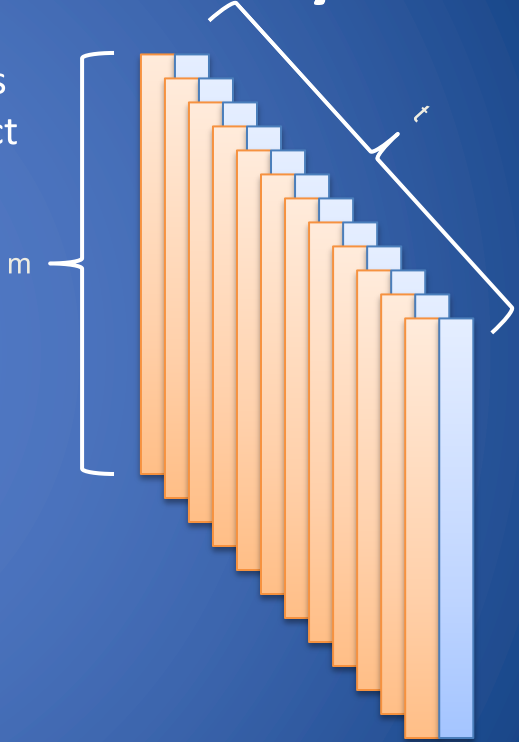
- Step 2: traverse the prefix of the chain ending at x

```
– while y = H(p) ≠ x
    • p = R(y) // advance
    • if j++ > t return “failure” // x is not in the table
– return p // password recovered
```



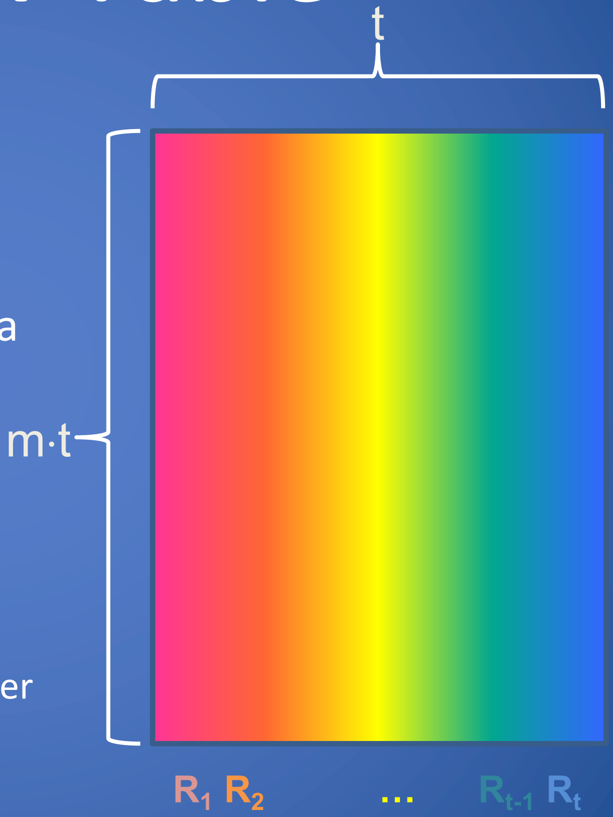
High-Probability Recovery

- Collisions in the reduction function are the cause of recovery issues
- To mitigate the impact of collisions, use multiple tables with distinct reduction functions
- Password space of size n covered with high probability by using t tables, where
  - Each tables uses a different reduction function
  - $m \cdot t^2$ is O(n)
- Performance
  - Storage cost: mt cryptographic hash values
  - Recovery cost: t^2 cryptographic hash computations and t^2 dictionary lookups
- Example
  - $m = t = n^{1/3}$
  - $n = 1,000,000,000$
  - $n^{2/3} = 1,000,000$



Rainbow Table

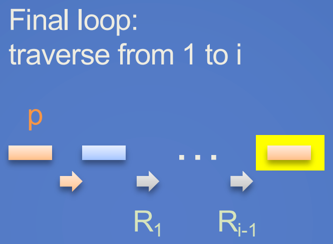
- Instead of t different tables, use a single table with
  - O(m·t) chains of length t
  - Different reduction function at each step
- Visualizing the reduction functions with a gradient of colors yields a rainbow
- Performance
  - Storage cost: mt cryptographic hash values, similar to previous method
  - Recovery cost: $t^2/2$ cryptographic hash computations and t dictionary lookups, lower than previous method

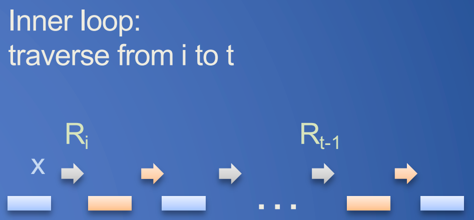


Rainbow Password Recovery

Recovery of password with hash value x

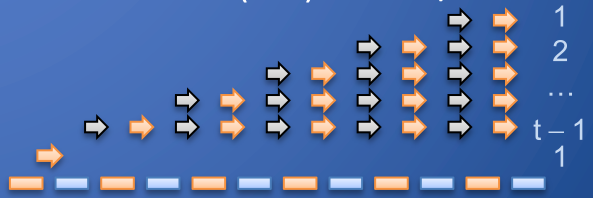
```
for i = t, (t – 1), … , 1
    // Traverse from i to t
    y = x
    for j = i, …, t - 1
        y = H(R_j(y)) // advance
    if p = D.get(y) is not null
        // i is the candidate position
        for j = 1 … i - 1 // Traverse chain from 1 to i
            p = R_j(H(p)) // advance
        if H(p) = x   return p // password recovered
        else return "failure" // x is not in the table
return "failure" // x is not in the table
```





Worst-case number of hash computations

$1 + 2 + \ldots + (t - 1) + 1 \approx t^2/2$



Entropy

Entropy

- Amount of uncertainty in a situation
- Fair Coin Flip
  - Maximum uncertainty
- Biased Coin Flip
  - More bias $\rightarrow$ Less uncertainty

Entropy

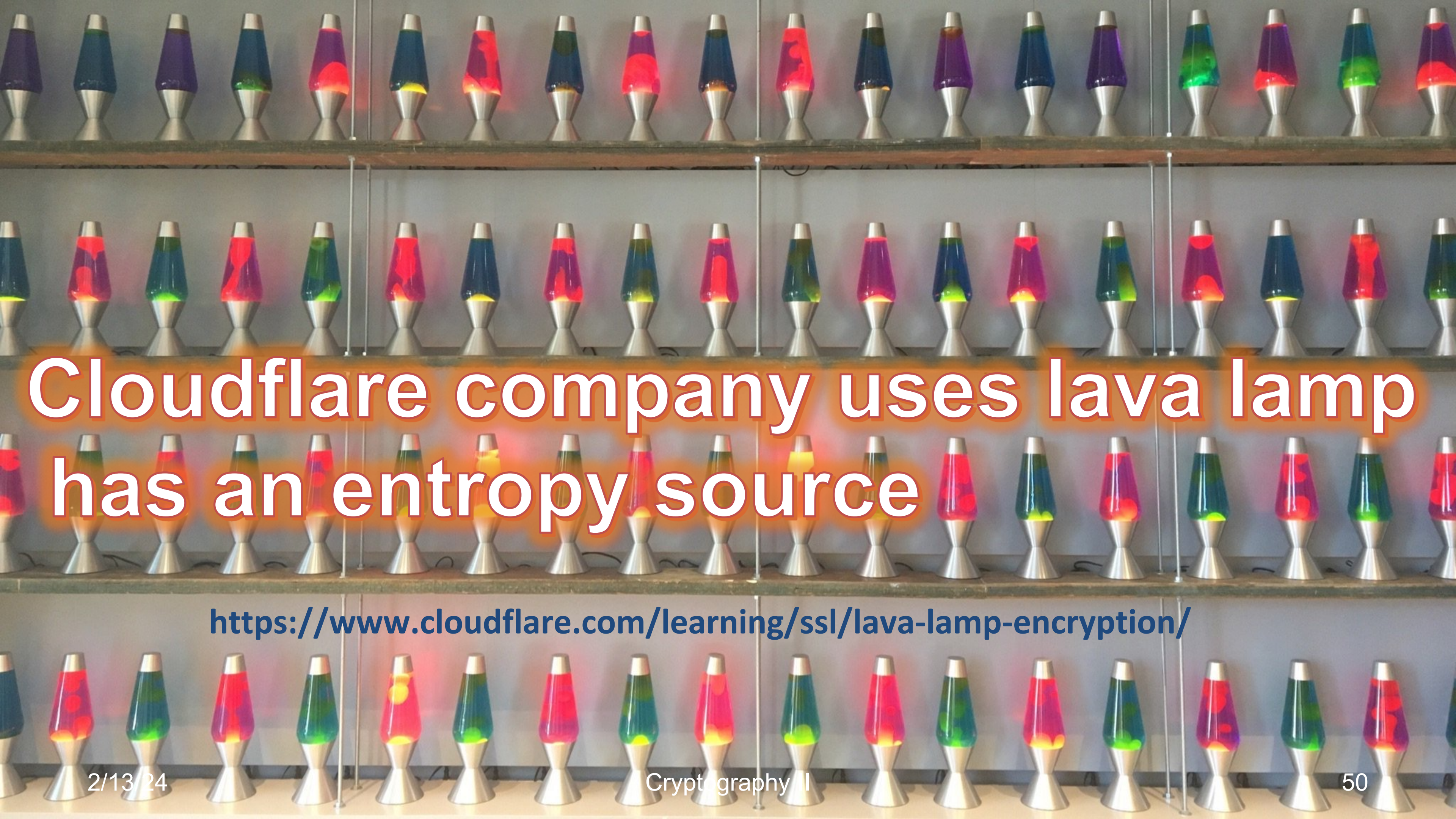
- Computers need a source of uncertainty (entropy) to generate random numbers.
  - Cryptographic keys.
  - Protocols that need coin flips.
- Which are sources of entropy in a computer?
  - Mouse and keyboard movements or thermal noise of processor.
  - Unix like operating systems use dev/random and dev/urandom as randomness collector

Random Number in Practice

- We need random numbers but...

  "Anyone who considers arithmetical methods of producing random numbers is, of course, in a state of sin." - John von Neumann

- Bootup state is predictable and entropy from the environment may be limited:
  - Temperature is relatively stable
  - Oftentimes the mouse/keyboard motions are predictable
- Routers often use network traffic
  - Eavesdroppers.
- Electromagnetic noise from an antenna outside of a building
- Radioactive decay of a 'pellet' of uranium
- Lava lamps…



Cloudflare company uses lava lamp has an entropy source

https://www.cloudflare.com/learning/ssl/lava-lamp-encryption/

Random Oracle Model

- A random oracle is a theoretical model (black box) that:
  - Answers to a query Q with a random number
  - Answers same way every time it receives same query Q
- This is often not an accurate representation of reality...

Exceptional Access to Encrypted Data

So what can we do with crypto?

- Confidentiality: can communicate secretly
- Authenticity: can verify identity, integrity of messages from another party

These are very powerful concepts!

Before computers…

Before modern computing, access to cryptography was mainly limited to governments

Vigenère cipher (1866)

Enigma Machine (1940s)

ROMULUS (1960s)

Now..

Cryptography is open—
anyone can use it.

What does that mean?

“Crypto Wars”

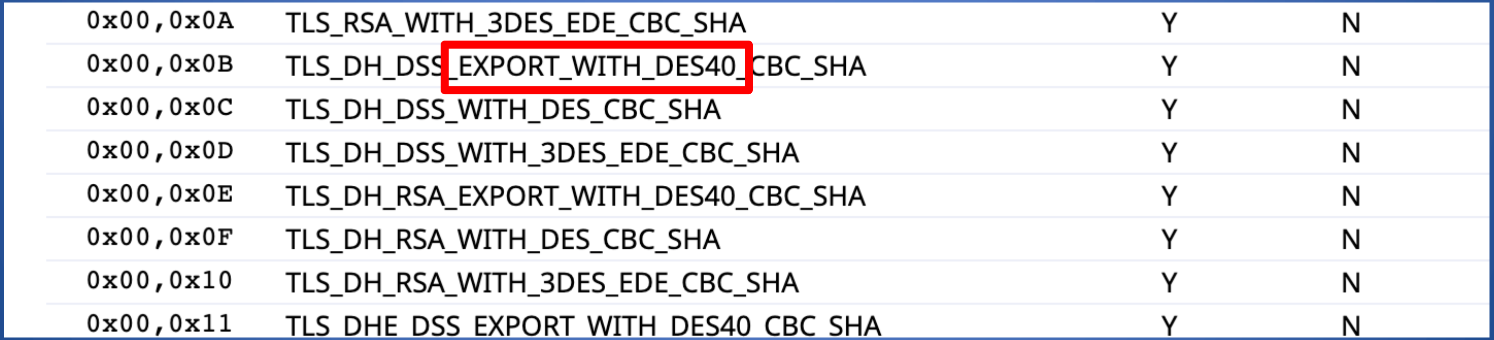
In US, cryptography was subject to ”export controls” until 1996-1997

- Categorized similarly to weapons, military technology
- Limited sharing outside US, often required using weaker systems

WARNING
This shirt is classified as a munition and may not be exported from the United States, or shown to a foreign national
RSA
encryption in perl
Machine readable version of program

Example: TLS cipher suites

- TLS is the main protocol that secures web traffic
- Early cipher modes for TLS had weaker “export” versions (eg. DES key size reduced 56 => 40 bits!)



0x00,0x0A	TLS_RSA_WITH_3DES_EDE_CBC_SHA	Y	N
0x00,0x0B	TLS_DH_DSS_EXPORT_WITH_DES40_CBC_SHA	Y	N
0x00,0x0C	TLS_DH_DSS_WITH_DES_CBC_SHA	Y	N
0x00,0x0D	TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA	Y	N
0x00,0x0E	TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA	Y	N
0x00,0x0F	TLS_DH_RSA_WITH_DES_CBC_SHA	Y	N
0x00,0x10	TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA	Y	N
0x00,0x11	TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	Y	N

- This can still affect badly-configured TLS versions!

For Law Enforcement

Exceptional access: should a government be able to access encrypted data if they have a warrant?

- Can law enforcement compel someone to disclose your password?
- Should cryptosystems have built-in exceptions?

Key Escrow

Idea: Build exceptional access into algorithm, or how it's deployed

- Off-device: government keeps a copy of key
- On-device: store key in such a way government can obtain it

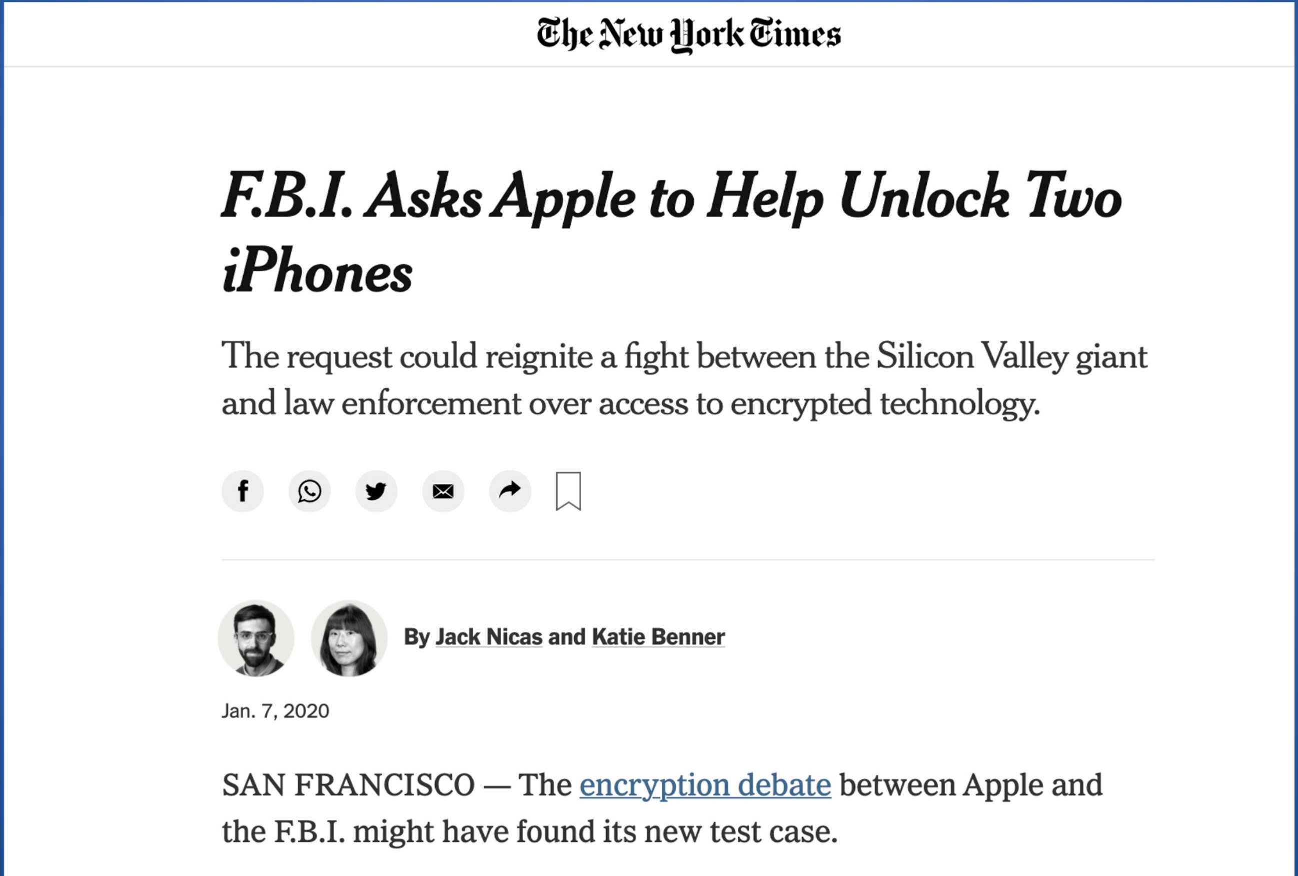
Problems?

Clipper Chip (~1993)

- Device proposed for encrypting landline telephone communications, promoted by NSA
- Used then-undisclosed algorithm that designed for using key escrow

- Never actually adopted
- Vulnerabilities discovered in key escrow mechanism

FBI vs. Apple (2015)



The New York Times

F.B.I. Asks Apple to Help Unlock Two iPhones

The request could reignite a fight between the Silicon Valley giant and law enforcement over access to encrypted technology.

By Jack Nicas and Katie Benner

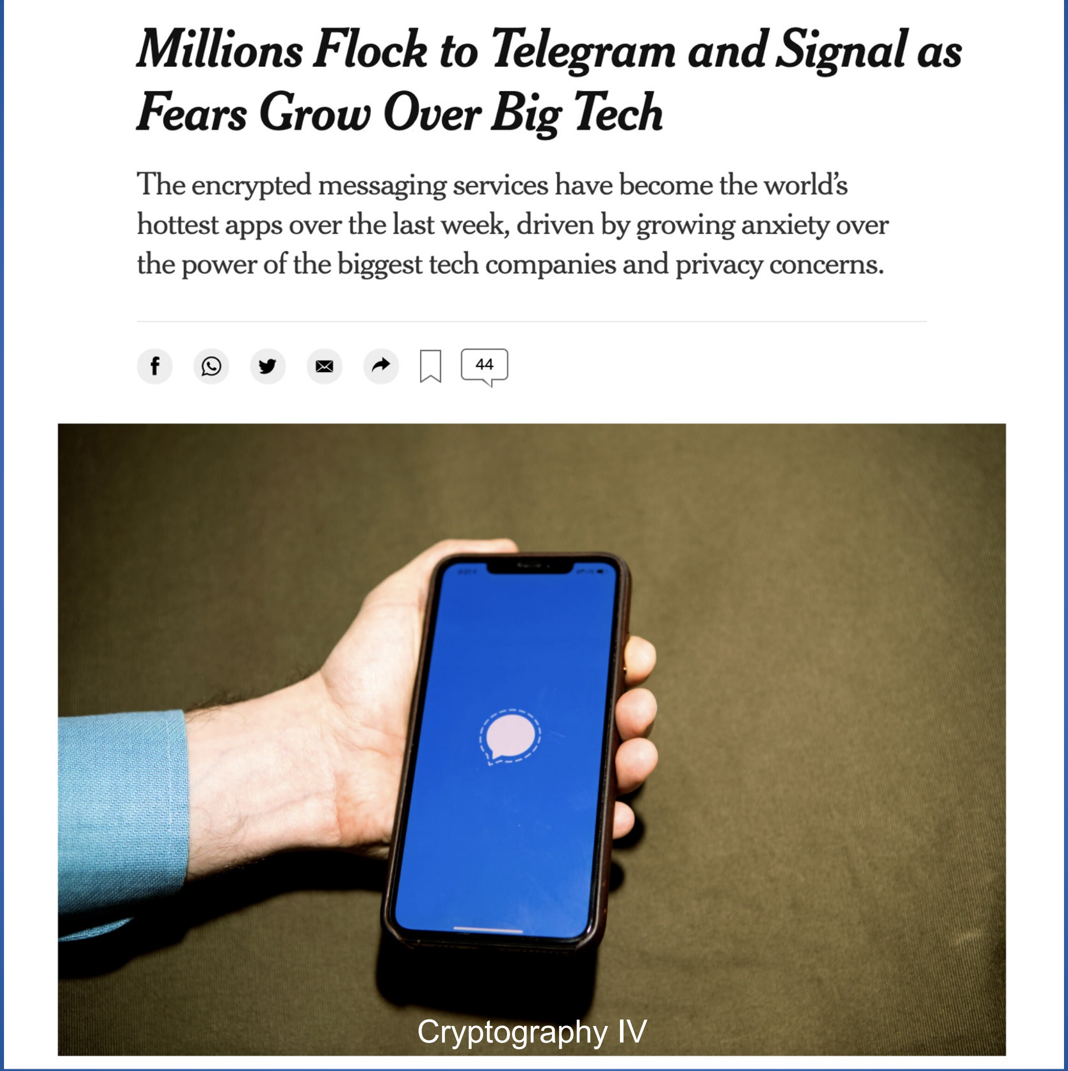
Jan. 7, 2020

SAN FRANCISCO — The encryption debate between Apple and the F.B.I. might have found its new test case.

If we allow exceptional access…

What risks does this create?

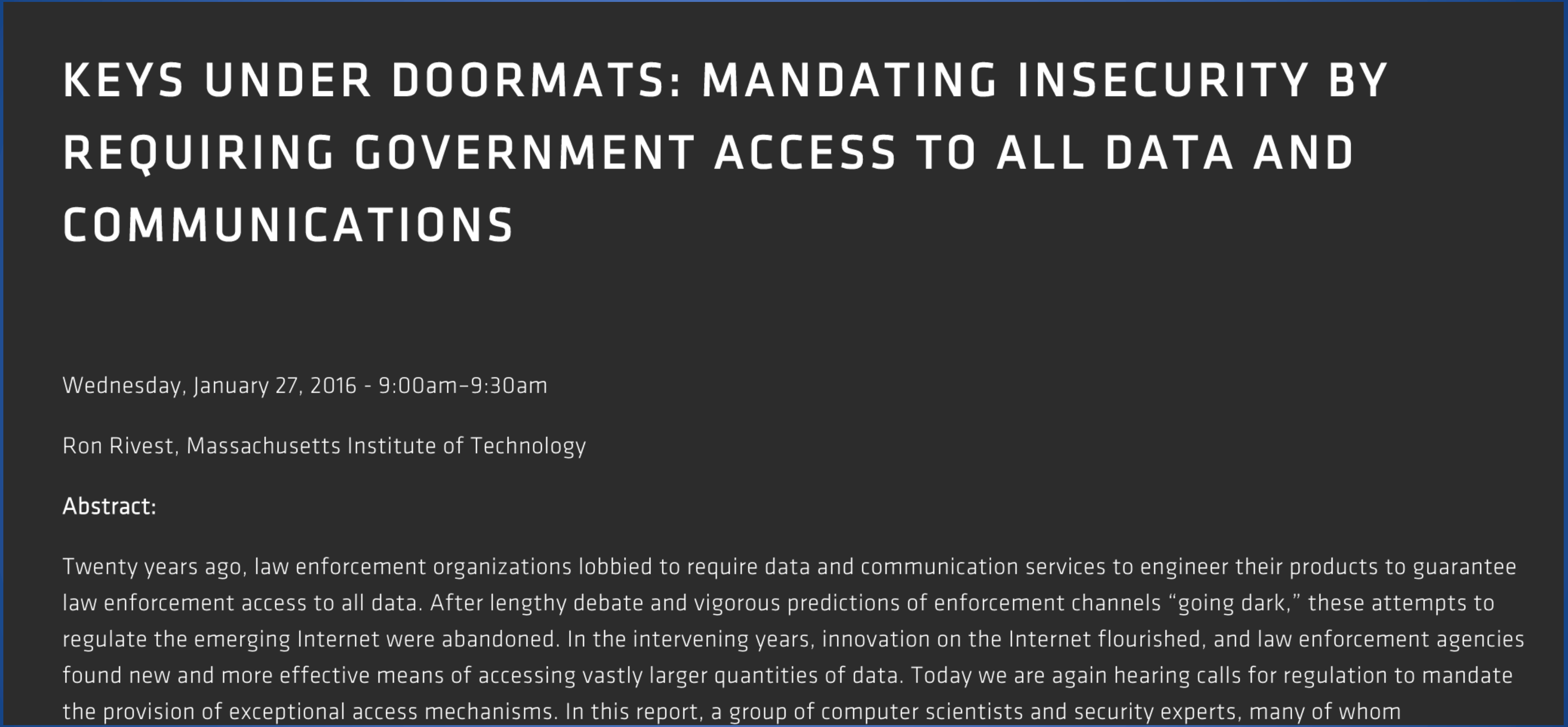
- Could a malicious actor use the system?
- Can we trust the government/company/etc to use the system properly?
- Does it add complexity?
  - More complexity => more that can go wrong
- Will users just switch to other, more secure systems?



Millions Flock to Telegram and Signal as Fears Grow Over Big Tech

The encrypted messaging services have become the world's hottest apps over the last week, driven by growing anxiety over the power of the biggest tech companies and privacy concerns.

44



KEYS UNDER DOORMATS: MANDATING INSECURITY BY REQUIRING GOVERNMENT ACCESS TO ALL DATA AND COMMUNICATIONS

Wednesday, January 27, 2016 - 9:00am–9:30am

Ron Rivest, Massachusetts Institute of Technology

Abstract:

Twenty years ago, law enforcement organizations lobbied to require data and communication services to engineer their products to guarantee law enforcement access to all data. After lengthy debate and vigorous predictions of enforcement channels “going dark,” these attempts to regulate the emerging Internet were abandoned. In the intervening years, innovation on the Internet flourished, and law enforcement agencies found new and more effective means of accessing vastly larger quantities of data. Today we are again hearing calls for regulation to mandate the provision of exceptional access mechanisms. In this report, a group of computer scientists and security experts, many of whom

What We Have Learned

- Authorization
- Password cracking
  - –Brute-force
  - –Dictionary precomputation
  - –Intelligent guessing
- Rainbow table
- Exceptional Access to Encrypted Data
- Entropy