Class 36: Public Key Crypto

CS150: Computer Science
University of Virginia
Computer Science

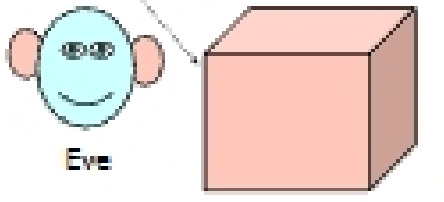
David Evans
http://www.cs.virginia.edu/cs150

Login Process

Terminal

```
login: alyssa
password: fido
```

login sends
<"alyssa", "fido">



Password Problems

- Need to store the passwords
  - Dangerous to rely on database being secure

Lecture 31, Recap Now

- Need to transmit password from user to host
  - Dangerous to rely on Internet being confidential

Today

Hashed Passwords

UserID	Password
alyssa	*f*("fido")
ben	*f*("schemer")
dave	*f*("Lx.Ly.x")

Dictionary Attacks

- Try a list of common passwords
  - All 1-4 letter words
  - List of common (dog) names
  - Words from dictionary
  - Phone numbers, license plates
  - All of the above in reverse
- Simple dictionary attacks retrieve most user-selected passwords
- Precompute $H(x)$ for all dictionary entries

(at least) 86% of users are dumb and dumber

Single ASCII character	0.5%
Two characters	2%
Three characters	14%
Four alphabetic letters	14%
Five same-case letters	21%
Six lowercase letters	18%
Words in dictionaries or names	15%
Other (possibly good passwords)	**14%**

(Morris/Thompson 79)

Salt of the Earth

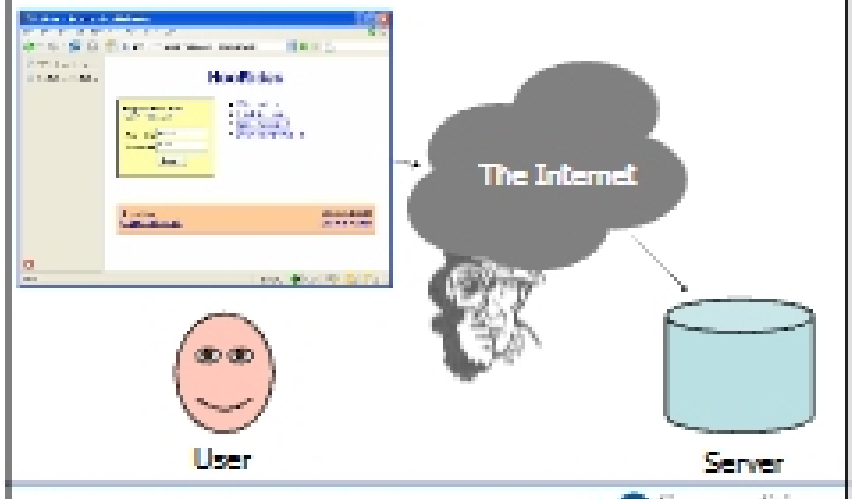
Salt: 12 random bits

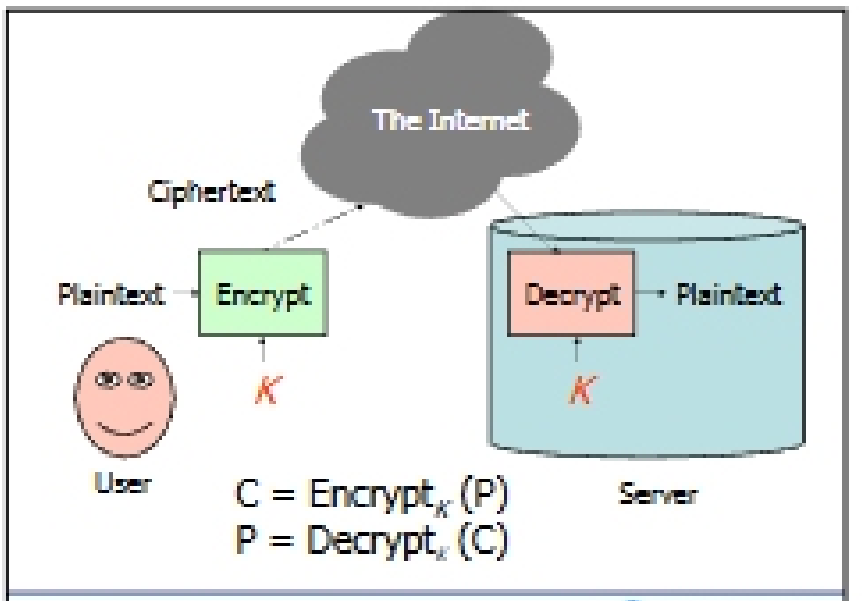
UserID	Salt	Password
alyssa	1125	crypt ("Lx.Ly.x", 1125)
ben	2437	crypt ("schemer", 2437)
dave	932	crypt ("Lx.Ly.x", 932)

How much harder is the off-line dictionary attack?

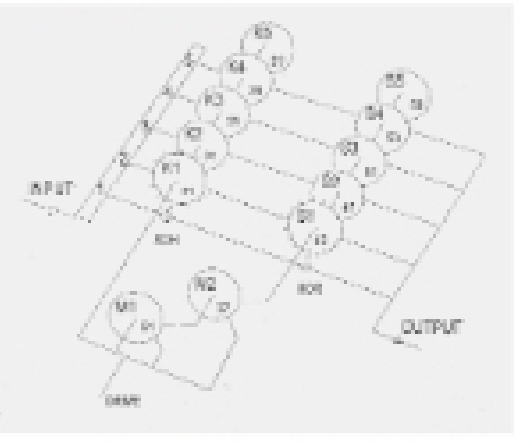
In HooRides.net we use the user name as the salt.
Is this better or worse?

Sending Passwords





PS4: Lorenz Cipher



From http://www.codesandciphers.org.uk/lorenz/fish.htm

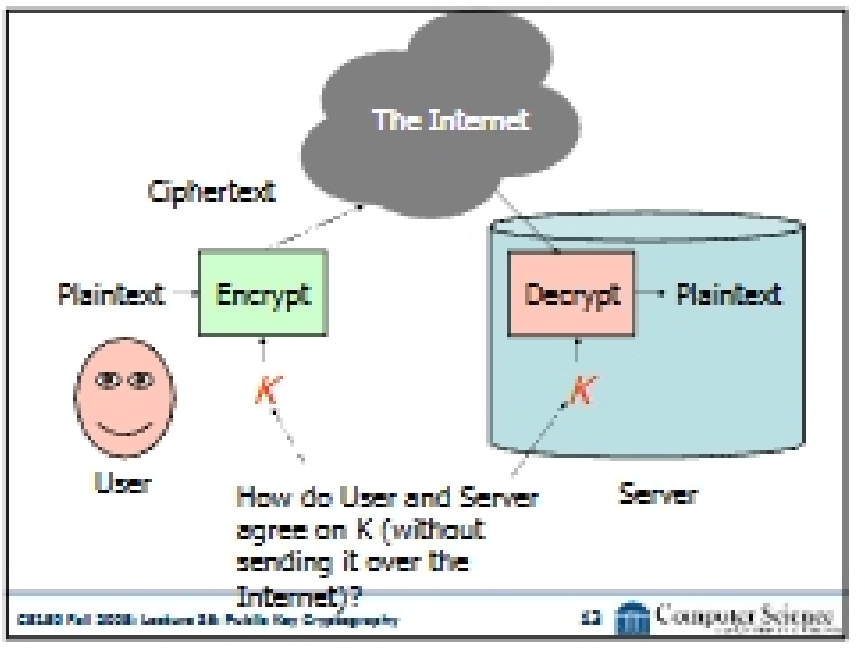
Modern Symmetric Ciphers

A billion billion is a large number, but it's not that large a number. Whitfield Diffie

- Same idea but:
  - Use digital logic instead of mechanical rotors
  - Larger keys (random bits, not rotor alignments)
    - PS4 = 5^5; Lorenz ≈ 5^{12} < 10^9
    - Modern ≥ 128 bits > 10^{37}
  - Encrypt blocks of letters at a time

Modern Ciphers

- AES (Rijndael) successor to DES selected 2001
- 128-bit keys, encrypt 128-bit blocks
- Brute force attack (around 10^{32} times harder than Lorenz)
  - Try 1 Trillion keys per second
  - Would take 10790283070806000000 years to try all keys!
  - If that's not enough, can use 256-bit key
- No known techniques that do better than brute force search



Key Agreement Demo

(Animated version at end of slides.)

Asymmetric Cryptosystems

- Need a hard problem (like symmetric cryptosystems)
- With a trap door: if you know a secret, the hard problem becomes easy

One-Way Functions

- Easy to compute, hard to invert
- Trap-door one way function:
  - $D(E(M)) = M$
  - E and D are easy to compute.
  - Revealing E doesn't reveal an easy way to compute D.
  - Hence, anyone who knows E can encrypt, but only someone who knows D can decrypt

RSA [Rivest, Shamir, Adelman 78]

One-way function:
multiplication is easy, factoring is hard
Trap-door: number theory (Euler and Fermat)

Security of RSA

- n is public, but not p and q where $n = pq$
- How much work is factoring n?

  Number Field Sieve (fastest known factoring algorithm) is:

$$O(e^{1.9223((\ln(n))^{1/3}(\ln(\ln(n)))^{2/3})})$$

n ~200 digits – would take quintillions of years

The movie Sneakers is about what happens if someone discovers a $O(n^k)$ factoring algorithm.