CSE 484 / CSE M 584: Computer Security and Privacy

Cryptography [Intro]

Fall 2017

Franziska (Franzi) Roesner
franzi@cs.washington.edu

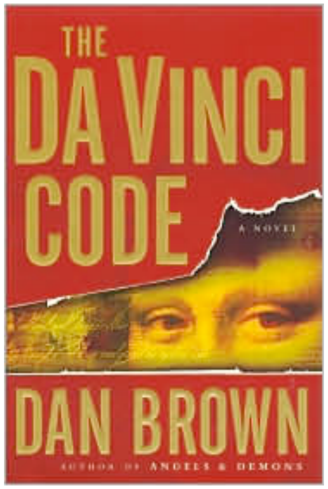
Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, Ada Lerner, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

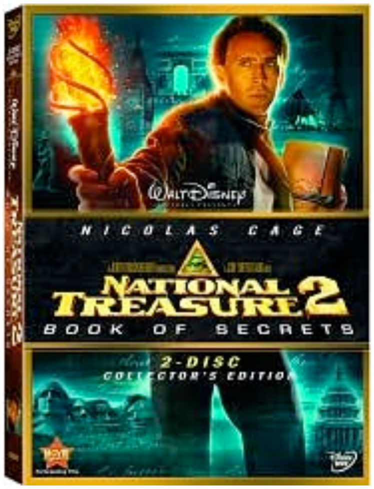
Admin

- Reminders:
  - Lab #1 checkpoint due tonight
    - Submit md5 hashes to dropbox
    - **Back up your sploit files!**
    - Only one person needs to submit
    - Include group name
  - You can pick up worksheets in my office

Cryptography and Security

- Art and science of *protecting* our *information*.
  - Keeping it **confidential**, if we want privacy.
  - Protecting its **integrity**, if we want to avoid forgeries.

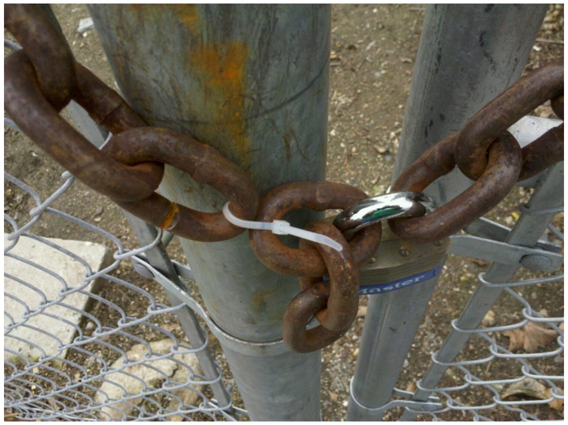




Images from Wikipedia and Barnes & Noble

Some Thoughts About Cryptography

- Cryptography only one small piece of a larger system
- Must protect entire system
  - Physical security
  - Operating system security
  - Network security
  - Users
  - Cryptography (following slides)



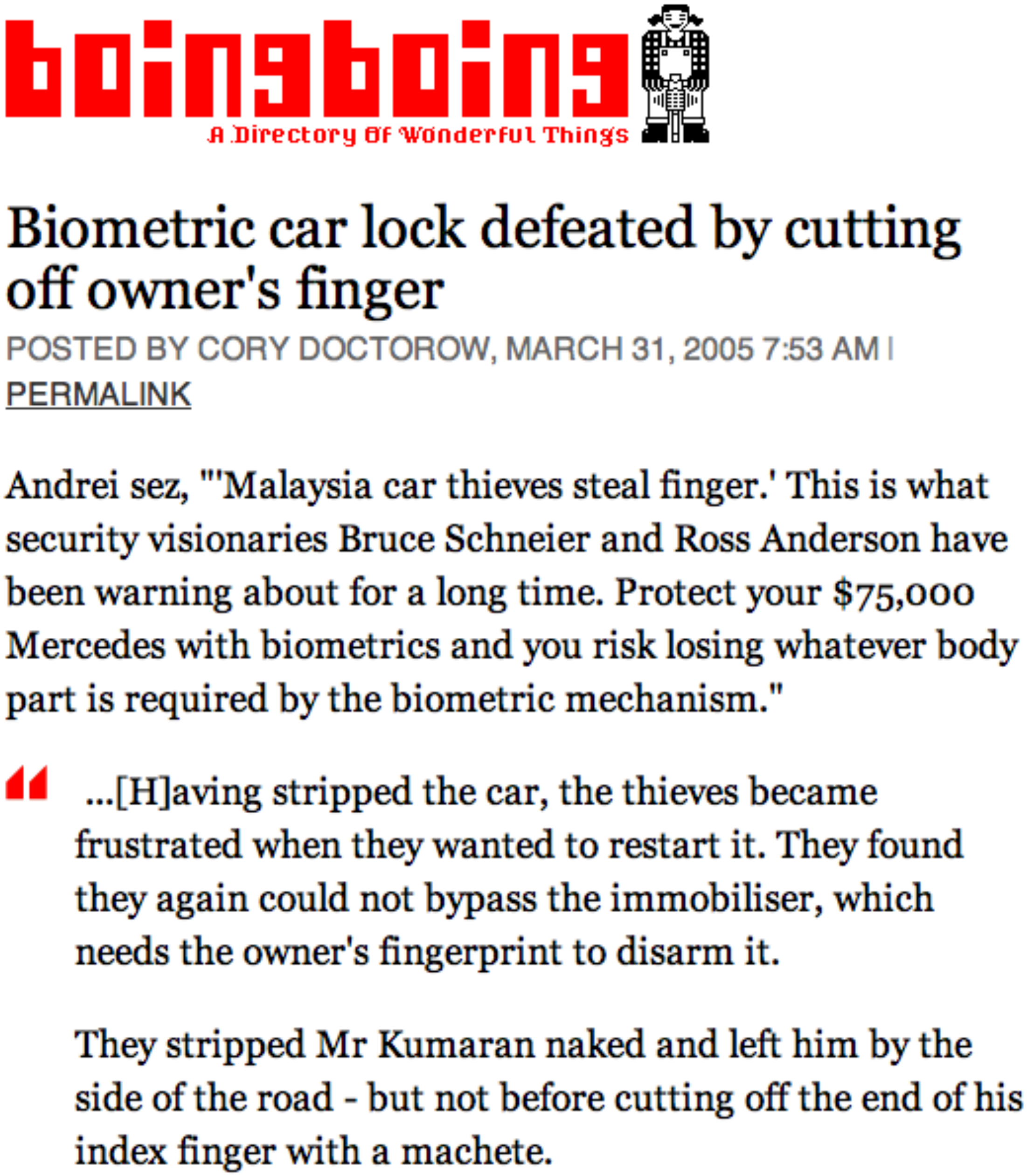
- Recall the weakest link
- Famous quote: "Those who think that cryptography can solve their problems don't understand cryptography and don't understand their problems."

Improved Security, Increased Risk

- RFIDs in car keys:
  - RFIDs in car keys make it harder to hotwire a car
  - Result: Car jackings increased

Improved Se

- RFIDs in car keys:
  - RFIDs in car keys
  - Result: Car jackin



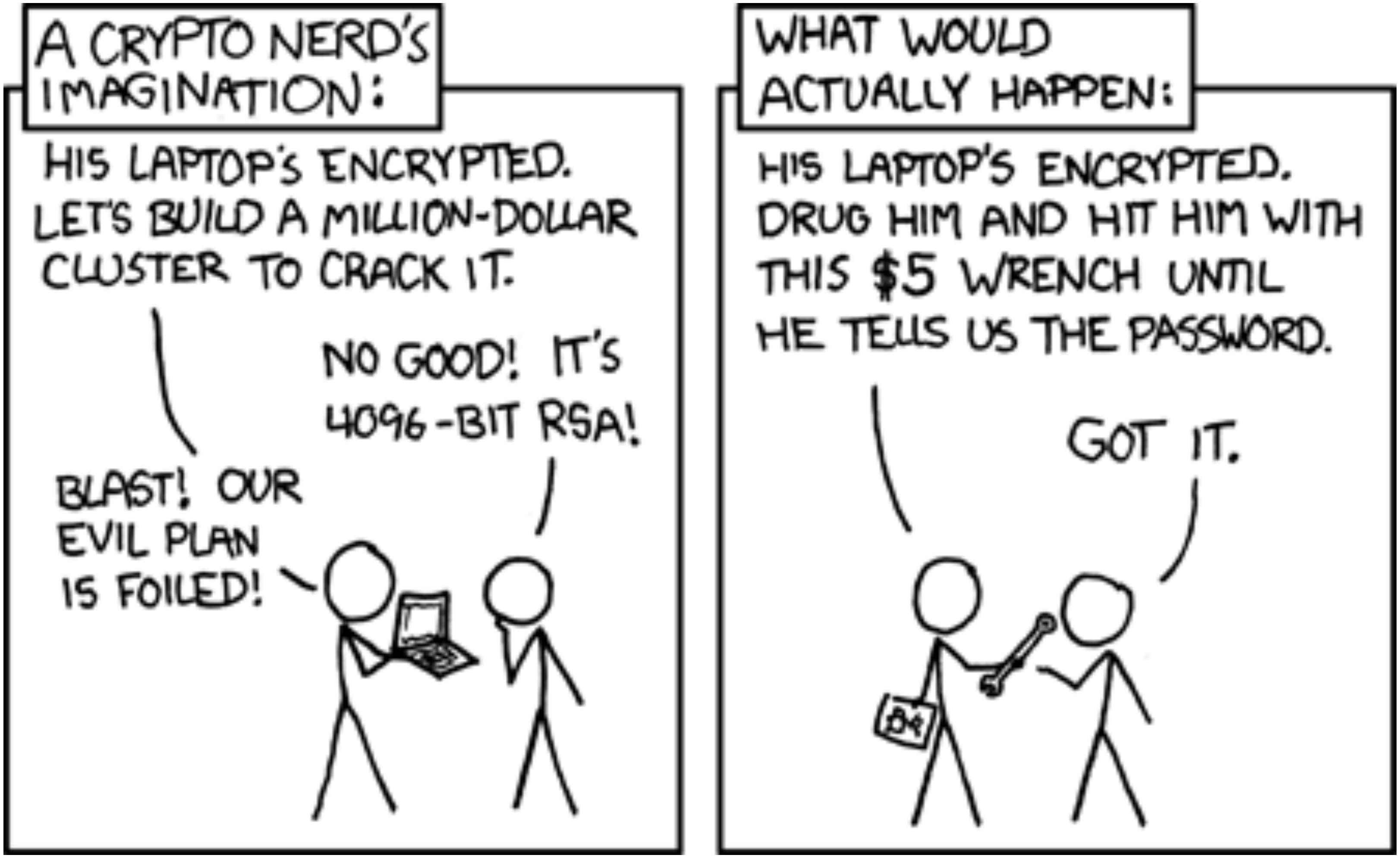
Biometric car lock defeated by cutting off owner's finger

POSTED BY CORY DOCTOROW, MARCH 31, 2005 7:53 AM | PERMALINK

Andrei sez, "'Malaysia car thieves steal finger.' This is what security visionaries Bruce Schneier and Ross Anderson have been warning about for a long time. Protect your $75,000 Mercedes with biometrics and you risk losing whatever body part is required by the biometric mechanism."

> ...[H]aving stripped the car, the thieves became frustrated when they wanted to restart it. They found they again could not bypass the immobiliser, which needs the owner's fingerprint to disarm it.
>
> They stripped Mr Kumaran naked and left him by the side of the road - but not before cutting off the end of his index finger with a machete.

XKCD: http://xkcd.com/538/


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.
$4

Kerckhoff’s Principle

- Security of a cryptographic object should depend only on the secrecy of the secret (private) key.
- Security should not depend on the secrecy of the algorithm itself.

Ingredient: Randomness

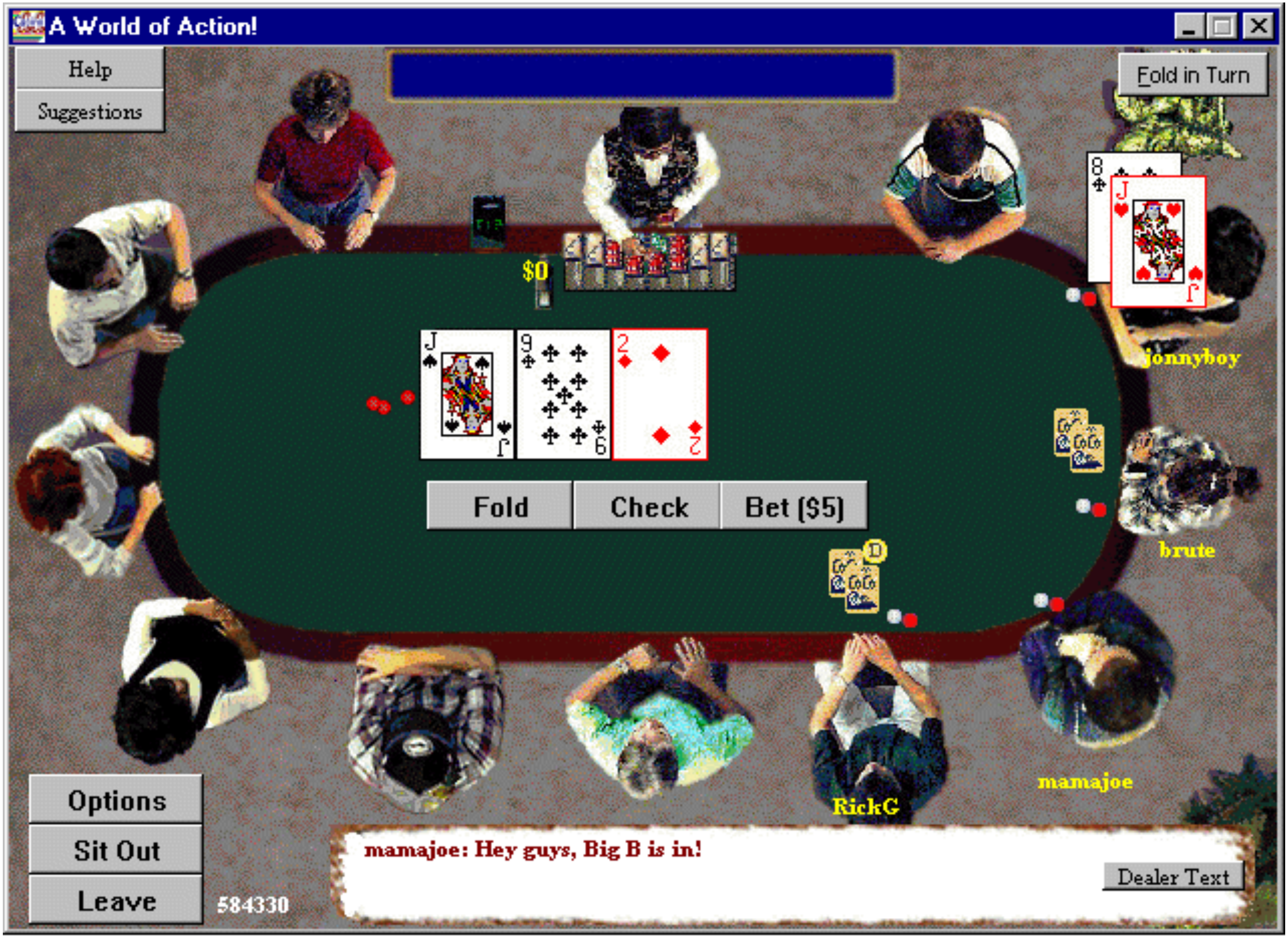
- Many applications (especially security ones) require randomness
- Explicit uses:
  - Generate secret cryptographic keys
  - Generate random initialization vectors for encryption
- Other “non-obvious” uses:
  - Generate passwords for new users
  - Shuffle the order of votes (in an electronic voting machine)
  - Shuffle cards (for an online gambling site)

C’s rand() Function

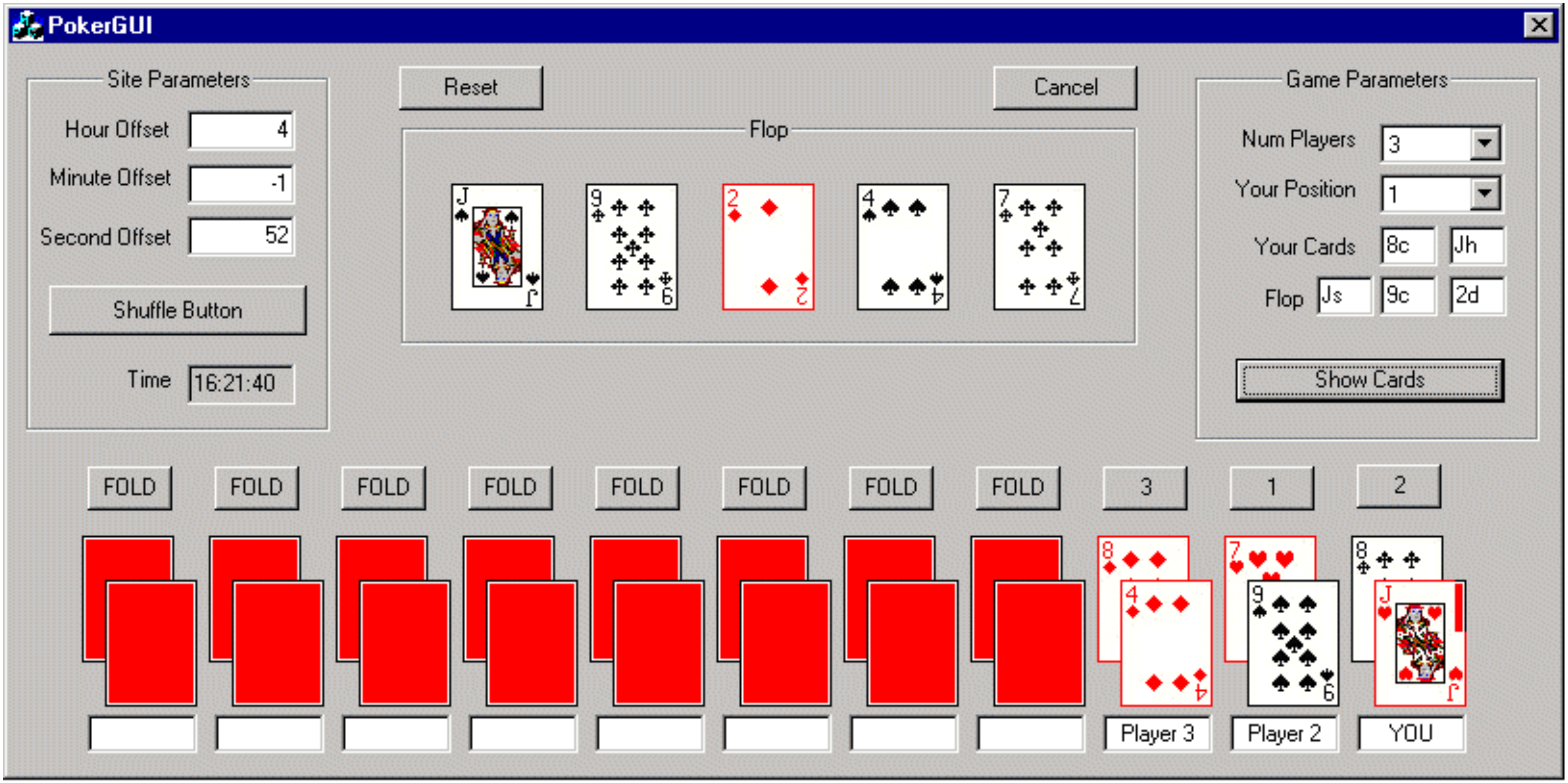
- C has a built-in random function: rand()

```
unsigned long int next = 1;
/* rand:  return pseudo-random integer on 0..32767 */
int rand(void) {
    next = next * 1103515245 + 12345;
    return (unsigned int)(next/65536) % 32768;
}
/* srand:  set seed for rand() */
void srand(unsigned int seed) {
    next = seed;
}
```

- Problem: don’t use rand() for security-critical applications!
  - Given a few sample outputs, you can predict subsequent ones



mamajoe: Hey guys, Big B is in!



Hour Offset 4

Minute Offset -1

Second Offset 52

Time 16:21:40

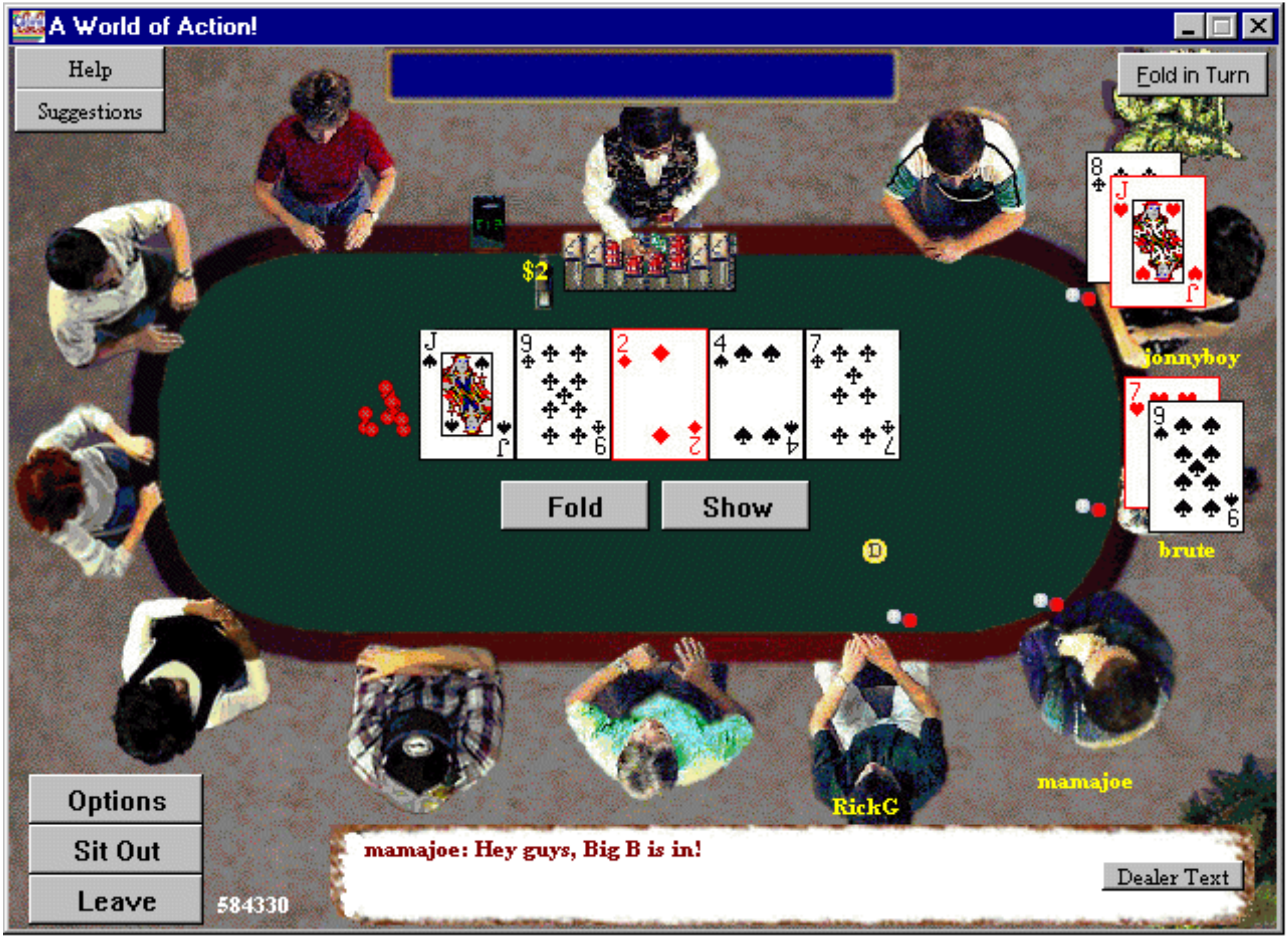
Num Players 3

Your Position 1

Your Cards 8c Jh

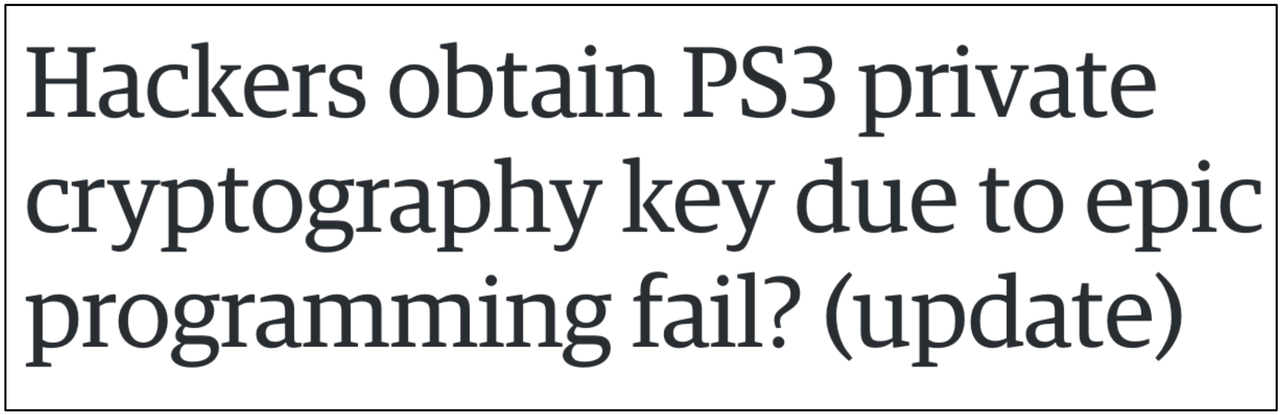
Flop Js 9c 2d

More details: “How We Learned to Cheat at Online Poker: A Study in Software Security”
http://www.cigital.com/papers/download/developer_gambling.php



mamajoe: Hey guys, Big B is in!

PS3 and Randomness



Hackers obtain PS3 private cryptography key due to epic programming fail? (update)

http://www.engadget.com/2010/12/29/hackers-obtain-ps3-private-cryptography-key-due-to-epic-programm/

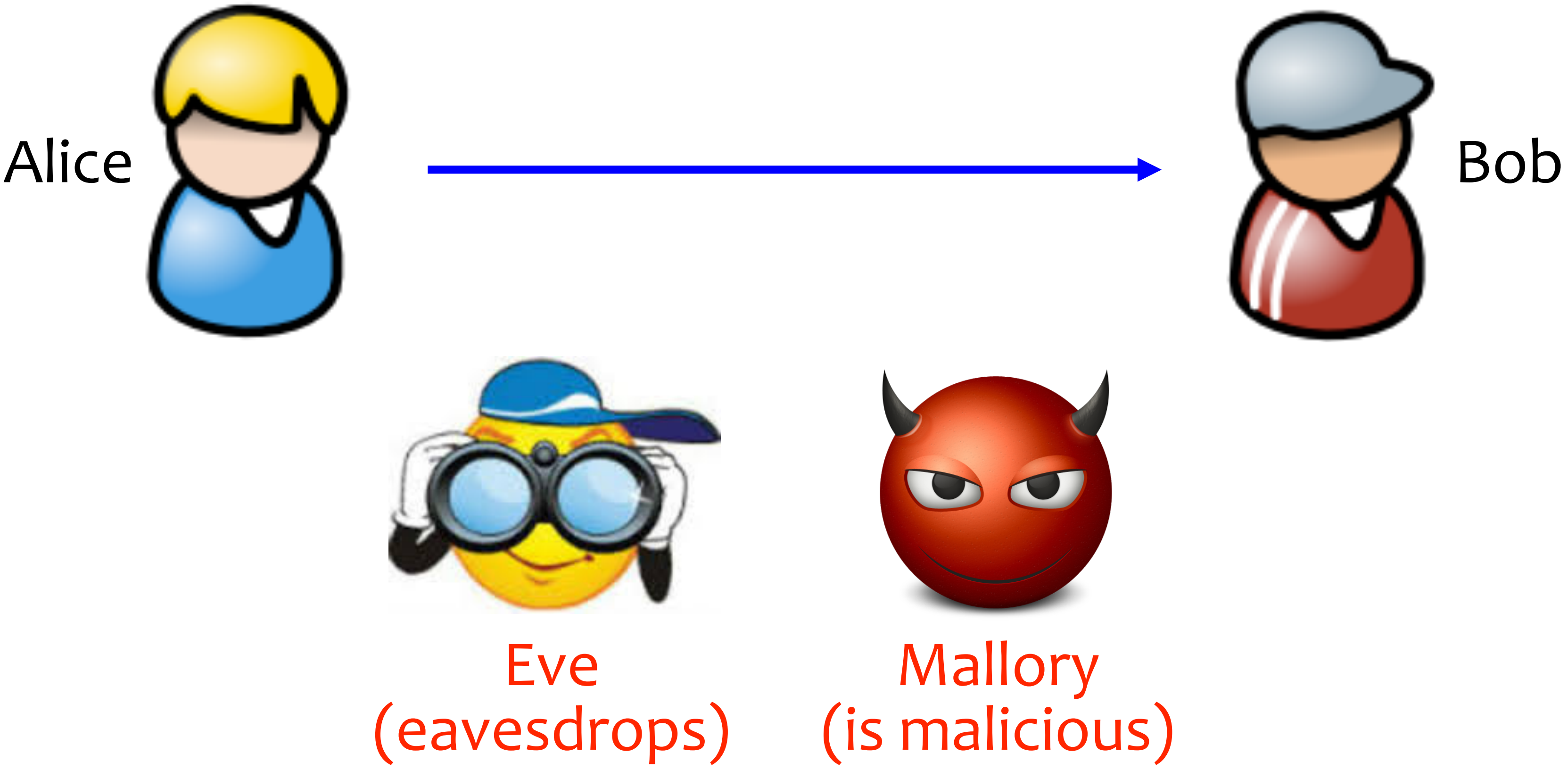
- 2010/2011: Hackers found/released private root key for Sony's PS3
- Key used to sign software – now can load any software on PS3 and it will execute as "trusted"
- Due to bad random number: same "random" value used to sign all system updates

Obtaining Pseudorandom Numbers

- For security applications, want "cryptographically secure pseudorandom numbers"
- Libraries include cryptographically secure pseudorandom number generators (CSPRNG)
- Linux:
  - /dev/random
  - /dev/urandom - nonblocking, possibly less entropy
- Internally:
  - Entropy pool gathered from multiple sources
    - e.g., mouse/keyboard timings

Alice and Bob

- Archetypical characters

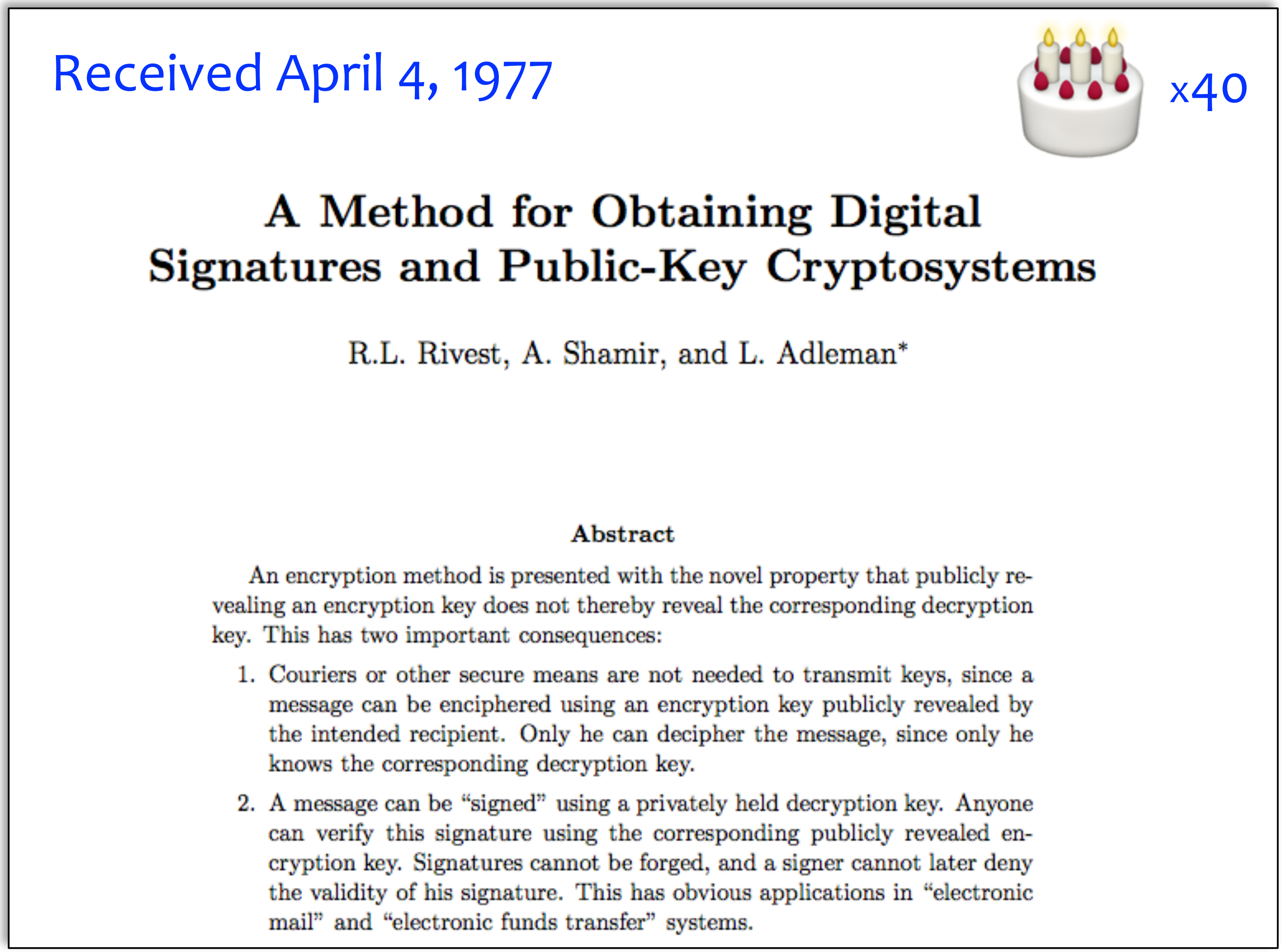


Alice

Bob

Eve
(eavesdrops)

Mallory
(is malicious)



Received April 4, 1977

🎂 x40

A Method for Obtaining Digital Signatures and Public-Key Cryptosystems

R.L. Rivest, A. Shamir, and L. Adleman*

Abstract

An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences:

1. Couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key.
2. A message can be "signed" using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in "electronic mail" and "electronic funds transfer" systems.

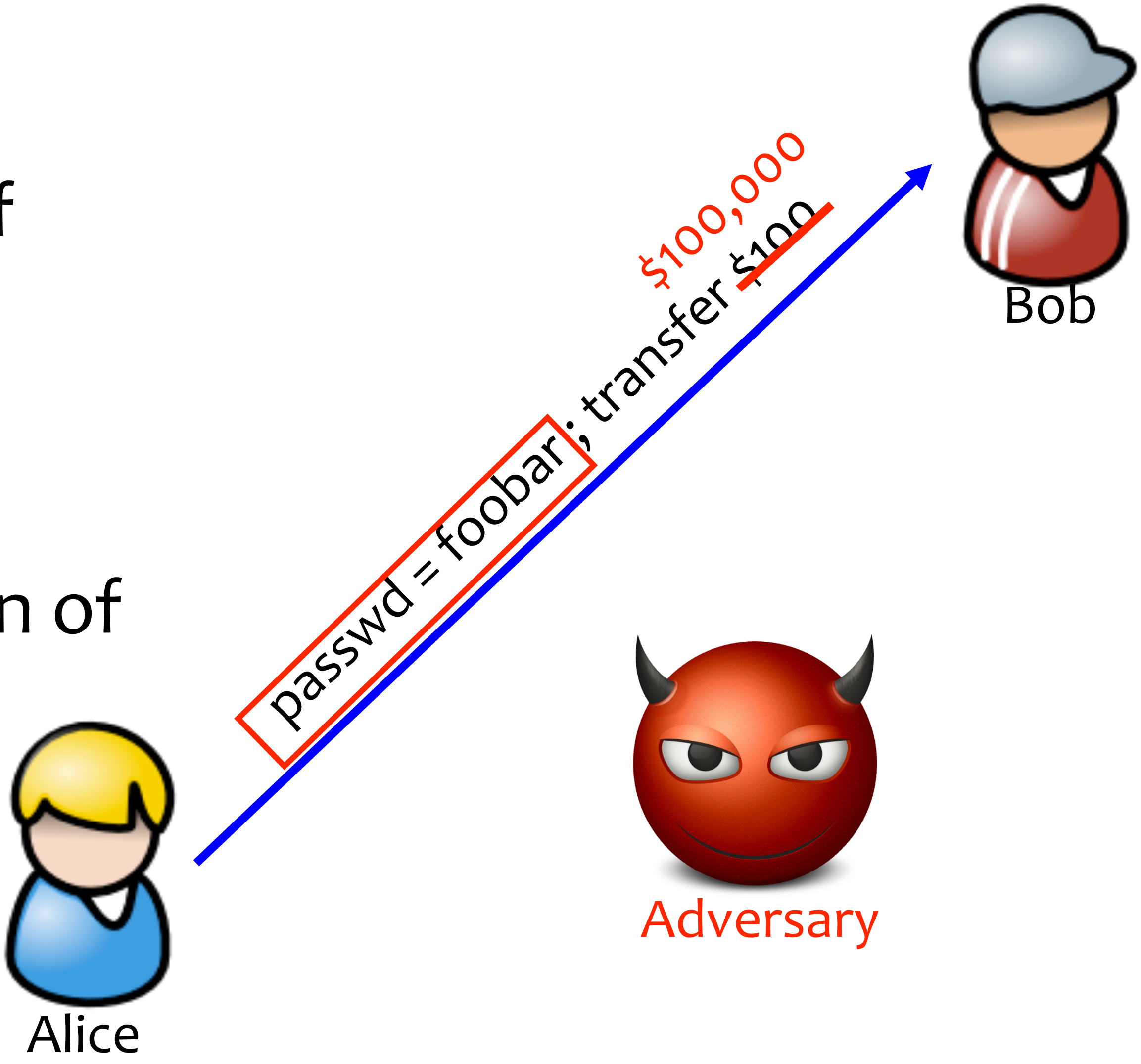
Common Communication Security Goals

Privacy of data:

Prevent exposure of information

Integrity of data:

Prevent modification of information



passwd = foobar; transfer ~~$100~~ $100,000

Bob

Alice

Adversary

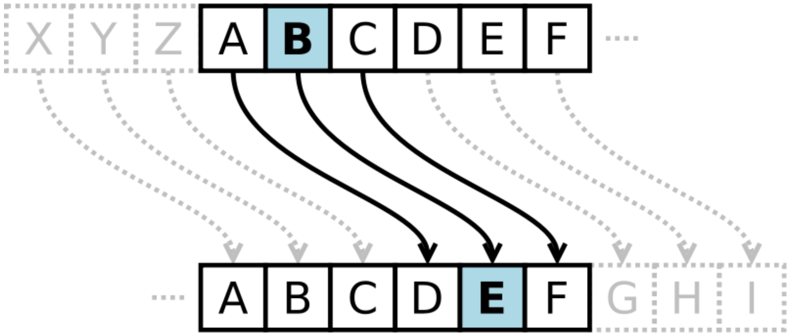
History

- Substitution Ciphers
  - Caesar Cipher
- Transposition Ciphers
- Codebooks
- Machines

- Recommended Reading: **The Codebreakers** by David Kahn and **The Code Book** by Simon Singh.

History: Caesar Cipher (Shift Cipher)

- Plaintext letters are replaced with letters a fixed shift away in the alphabet.



- Example:
  - Plaintext: `The quick brown fox jumps over the lazy dog`
  - Key: Shift 3

    `ABCDEFGHIJKLMNOPQRSTUVWXYZ`

    `DEFGHIJKLMNOPQRSTUVWXYZABC`
  - Ciphertext: `WKHTX LFNEU RZQIR AMXPS VRYHU WKHOD CBGRJ`

History: Caesar Cipher (Shift Cipher)

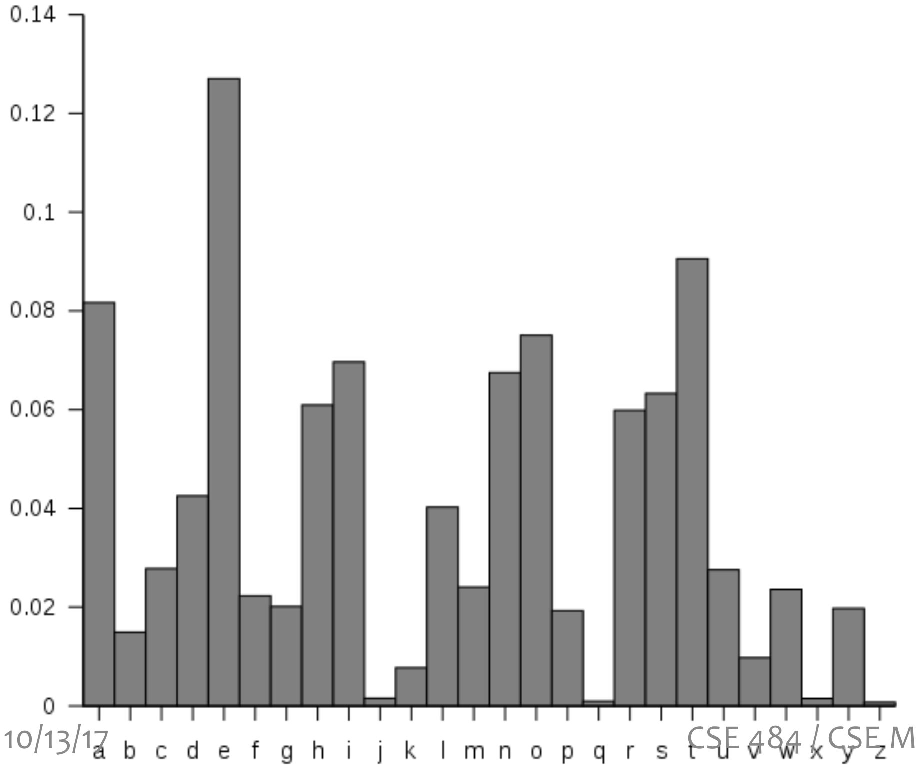
- ROT13: shift 13 (encryption and decryption are symmetric)
- What is the key space?
  - 26 possible shifts.
- How to attack shift ciphers?
  - Brute force.

History: Substitution Cipher

- Superset of shift ciphers: each letter is substituted for another one.
- Add a secret key
- Example:
  - Plaintext: `ABCDEFGHIJKLMNOPQRSTUVWXYZ`
  - Cipher: `ZEBRASCDFGHIJKLMNOPQTUVWXY`
- "State of the art" for thousands of years

History: Substitution Cipher

- What is the key space? 26! ~= 2^88
- How to attack?
  - – Frequency analysis.



Bigrams:

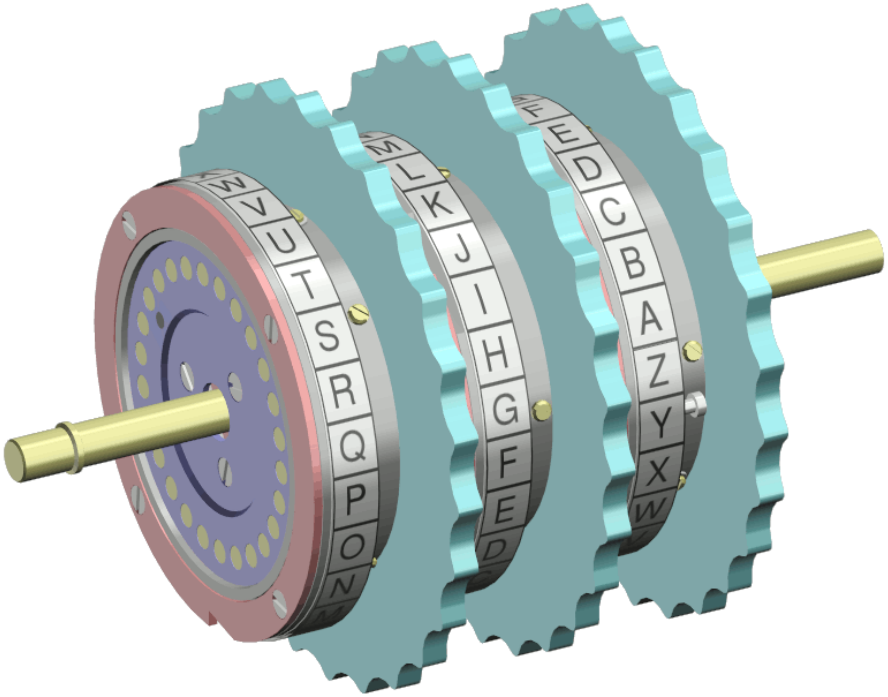
th	1.52%	en	0.55%	ng	0.18%
he	1.28%	ed	0.53%	of	0.16%
in	0.94%	to	0.52%	al	0.09%
er	0.94%	it	0.50%	de	0.09%
an	0.82%	ou	0.50%	se	0.08%
re	0.68%	ea	0.47%	le	0.08%
nd	0.63%	hi	0.46%	sa	0.06%
at	0.59%	is	0.46%	si	0.05%
on	0.57%	or	0.43%	ar	0.04%
nt	0.56%	ti	0.34%	ve	0.04%
ha	0.56%	as	0.33%	ra	0.04%
es	0.56%	te	0.27%	ld	0.02%
st	0.55%	et	0.19%	ur	0.02%

Trigrams:

1. the
2. and
3. tha
4. ent
5. ing
6. ion
7. tio
8. for
9. nde
10. has
11. nce
12. edt
13. tis
14. oft
15. sth

History: Enigma Machine

Uses rotors (substitution cipher) that change position after each key.



Key = initial setting of rotors

Key space?

26^n for n rotors

How Cryptosystems Work Today

- Layered approach:
  - **Cryptographic primitives**, like block ciphers, stream ciphers, hash functions, and one-way trapdoor permutations
  - **Cryptographic protocols**, like CBC mode encryption, CTR mode encryption, HMAC message authentication
- Public algorithms (Kerckhoff's Principle)
- Security proofs based on assumptions (not this course)

- Don't roll your own!