The Somewhat Simplified Solitaire Algorithm

Lester I. McCann
mccann@cs.arizona.edu

Computer Science Department
The University of Arizona
Tucson, AZ

ACM SIGCSE Nifty Assignments Panel
March 4, 2006
Who Is This Guy?
Who Is This Guy?

Best-selling Author Neal Stephenson

http://www.nealstephenson.com
What Has He Written?

(among others)
Cryptonomicon

- A Combination of Historical & Modern-Day Fiction

(c) 1999
Cryptonomicon

- A Combination of Historical & Modern-Day Fiction
- Threads Joined By Cryptography

(c) 1999
Cryptonomicon

- A Combination of Historical & Modern-Day Fiction
- Threads Joined By Cryptography
- And After ~ 800 pages . . .

(c) 1999
Cryptonomicon

A Combination of Historical & Modern-Day Fiction

Threads Joined By Cryptography

And After ~ 800 pages . . .

. . . The Pontifex Transform Is Used

(c) 1999
Pontifex == Solitaire

- In reality, Pontifex is really security expert Bruce Schneier’s Solitaire cryptosystem.
- Schneier describes it in Cryptonomicon’s appendix

www.schneier.com
Solitaire? A Cryptosystem??
Solitaire? A *Cryptosystem*??

No, not *that* Solitaire . . .
Bruce Schneier’s Solitaire

- So named because it is based on manipulations of playing cards
Bruce Schneier’s Solitaire

- So named because it is based on manipulations of playing cards
  - Who would question an innocent deck of cards?
As Tested on MythBusters!

by Ricky Jay, (c) 1977
Bruce Schneier’s Solitaire

- So named because it is based on manipulations of playing cards
  - Who would question an innocent deck of cards?
    ... OK, we’ll ignore that.
Bruce Schneier’s Solitaire

- So named because it is based on manipulations of playing cards
  - Who would question an innocent deck of cards?
    - ...OK, we’ll ignore that.

- Sender and Receiver begin with matched decks

- Each application of Solitaire generates a sequence of keystream values, each in the range [1..26]

- Roughly:
  - Plaintext + keystream = Ciphertext
  - Ciphertext - keystream = Plaintext
Keystream Algorithm: Step 1 of 5

Step 1: Exchange ‘A’ Joker with Following Card
Keystream Algorithm: Step 2 of 5

Step 2: Exchange ‘B’ Joker with Following Two Cards
Keystream Algorithm: Step 3 of 5

Step 3: “Triple Cut”
Keystream Algorithm: Step 4 of 5
Keystream Algorithm: Step 4 of 5

Step 4: Needs More Words Than I Have Space!
Keystream Algorithm: Step 5 of 5

Step 5:
Keystream Algorithm: Step 5 of 5

Step 5: 1st Card’s Value
Keystream Algorithm: Step 5 of 5

Step 5: 1st Card’s Value + 1 ⇒ Index
Keystream Algorithm: Step 5 of 5

Step 5: 1\text{st Card's Value} + 1 \Rightarrow \text{Index} \Rightarrow \text{Keystream Value} = 4
Encryption

Plaintext: N I F T Y

Letter Values: 14 9 6 20 25

Keystream Sequence: 4 2 4 1 5

Sums: 18 11 10 21 30

Wrap: 18 11 10 21 4

Ciphertext: R K J T D
Decryption

Ciphertext: R K J T D

Letter Values: 18 11 10 21 4
Keystream Sequence: 4 2 4 1 5
Differences: 14 9 6 20 -1
Wrap: 14 9 6 20 25
Plaintext: N I F T Y
Why “Somewhat Simplified”?

- Schneier has links to implementations in ~ 12 languages
Why “Somewhat Simplified”? 

- Schneier has links to implementations in ∼ 12 languages

- My Standard Adjustments:
  - Steps 1 and 2: No special bottom-of-deck behavior
    - Have students assume that the deck is circular
Why “Somewhat Simplified”?

- Schneier has links to implementations in ~ 12 languages
- My Standard Adjustments:
  - Steps 1 and 2: No special bottom-of-deck behavior
    - Have students assume that the deck is circular
  - Use a different deck; for example:
    - Half-deck (only two suits)
    - Pinochle deck (need to add jokers)
Why “Somewhat Simplified”? 

- Schneier has links to implementations in \(~ 12\) languages

- My Standard Adjustments:
  - Steps 1 and 2: No special bottom-of-deck behavior
    - Have students assume that the deck is circular
  - Use a different deck; for example:
    - Half-deck (only two suits)
    - Pinochle deck (need to add jokers)

⇒ Unwise cryptographically . . . but so what?
Adoption Issues

- Skill Prerequisites:
  - List Manipulation
  - Char $\leftrightarrow$ ASCII
  - Text File I/O (?)
Adoption Issues

- **Skill Prerequisites:**
  - List Manipulation
  - Char ↔ ASCII
  - Text File I/O (?)

- **Implementation Decisions:**
  - Arrays or Linked Lists?
  - Card Representation?
  - Must state be retained?
Adoption Issues

- **Skill Prerequisites:**
  - List Manipulation
  - Char ⇔ ASCII
  - Text File I/O (?)

- **Implementation Decisions:**
  - Arrays or Linked Lists?
  - Card Representation?
  - Must state be retained?

:: Applicable to CS0, CS1, CS2, ...
So Why Is This “Nifty”?

- Flexible — Can assign entire system or just parts
- Provides a gentle introduction to cryptosystems
- Encourages distributed testing (message exchange)
- Would be a fun algorithm to animate
So Why Is This “Nifty”? 

- Flexible — Can assign entire system or just parts
- Provides a gentle introduction to cryptosystems
- Encourages distributed testing (message exchange)
- Would be a fun algorithm to animate

- Just *might* encourage students to read a novel! 😊
Image Credits

- Neal Stephenson: Bela Bollobas
- Bruce Schneier: dk.compulenta.ru
- Stephenson book covers: barnesandnoble.com
- Klondike: AisleRot 2.10.0 / Jonathan Blandford
- Cards As Weapons: amazon.com
- Card Images: david.bellot.free.fr
- UA Campus: The UA Computer Science Webcam
Any *Quick* Questions?

mccann@cs.arizona.edu

These full-screen PDF slides were created in LaTeX using the `prosper` class.