# Fun with numbers

Breaking the NRIC check digit algorithm

**Ngiam Shih Tung**

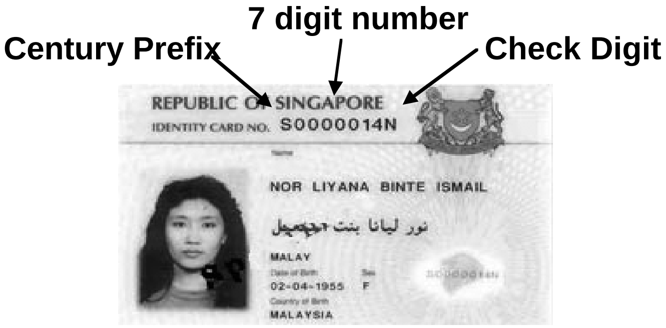
**December 22, 2003**

# Introduction

- The algorithm for computing the check digit for Singapore identity card numbers is unpublished
- Algorithm is partially described in various open sources
- Objective of this exercise is to elucidate the complete algorithm from internet resources and “virtual experimentation”

# UIN/FIN structure

- The National Registration Identity Card (NRIC) number is the Unique Identification Number (UIN) or Foreigner Identification Number (FIN)



Century Prefix

7 digit number

Check Digit

- Century prefix
  - **S, T - 19th and 20th letters of alphabet for UINs issued in 19xx and 20xx respectively**
  - **F, G - Foreigners (not 7th and 8th century !)**
- Check digit (official reference)
  - **Computed from first eight characters of UIN/FIN**
  - **Detects data entry errors**

☆ **How do we calculate this ?**

# UIN/FIN algorithm

- Government will release UIN/FIN algorithm for computing check digit, BUT
  - "**Application is open ONLY to Singapore-based organisations with the *legitimate* need for the UIN/FIN validation.**"
  - "Your application is subject to our final approval and our decision shall be final"
  - License agreement requires:
    - *"The Licensee agrees to take all reasonable steps to protect the Licensed Material from* ***unauthorised*** *copying, adaptation or use."*
  - License fee

| | |
|---|---|
| Algorithm | $200 |
| Sample code | $400 |

Source: ICA website (http://app.ica.gov.sg/related_links/uin_fin/unifin_faq.asp)

# IP Analysis

## Can the government really prohibit unauthorised use ?

- Copyright
  - Source code is subject to copyright
  - Algorithms are *not* subject to copyright
- Patent
  - Algorithms are patentable, but
    - Patent must be published
    - Prior art probably exists in this case
    - Patent, if any is long expired (> 20 yrs)
- Trade Secret
  - May be protectable under the license agreement
  - BUT, no secret if the information is already publicly available or obtained via a different route

# Modulo 11 checksum

- Algorithm for S-series (old-style) NRIC numbers is well-known*

***7-digit NRIC number*** ***Weights***

$$d = \left[(d_1\ d_2\ d_3\ d_4\ d_5\ d_6\ d_7) \bullet (2\ 7\ 6\ 5\ 4\ 3\ 2\ )\right] \bmod 11$$

$$= (\ 2d_1 + 7d_2 + 6d_3 + 5d_4 + 4d_5 + 3d_6 + 2d_7\ ) \bmod 11$$

Lookup d:

| *d* | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Check digit | A | B | C | D | E | F | G | H | I | Z | J |

1

- Does this work for F, G, T-prefix UIN/FINs ?

* e.g. soc.culture.singapore newgroup postings (1995)

# Reverse Engineering the FIN algorithm

- Find a large set of FINs then reverse engineer the check digits to determine weights and mapping of checksum to letters
- MOM publishes a list of Registered Safety Officers on its website

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| F | 8 | 1 | 7 | 9 | 5 | 9 | 9 | K | 10 |
| F | 5 | 5 | 3 | 3 | 3 | 9 | 7 | K | 10 |
| F | 7 | 7 | 8 | 3 | 9 | 8 | 0 | K | 10 |
| F | 5 | 5 | 6 | 4 | 4 | 3 | 8 | K | 10 |
| F | 5 | 5 | 5 | 8 | 2 | 8 | 3 | K | 10 |
| F | 2 | 4 | 1 | 3 | 0 | 7 | 6 | L | 9 |
| F | 2 | 4 | 0 | 7 | 5 | 3 | 6 | L | 9 |
| F | 5 | 5 | 9 | 3 | 2 | 0 | 4 | L | 9 |
| F | 7 | 3 | 4 | 8 | 9 | 4 | 8 | L | 9 |
| F | 2 | 5 | 2 | 9 | 7 | 7 | 9 | L | 9 |
| F | 7 | 3 | 4 | 2 | 5 | 6 | 0 | M | 8 |
| F | 7 | 7 | 0 | 8 | 0 | 3 | 3 | M | 8 |
| F | 2 | 3 | 1 | 5 | 9 | 6 | 4 | M | 8 |
| F | 8 | 1 | 7 | 9 | 5 | 9 | 8 | M | 8 |
| F | 1 | 9 | 2 | 0 | 2 | 6 | 2 | M | 8 |
| F | 5 | 5 | 6 | 0 | 5 | 4 | 2 | N | 7 |
| F | 8 | 1 | 0 | 4 | 0 | 4 | 9 | N | 7 |
| F | 8 | 3 | 2 | 6 | 1 | 9 | 2 | N | 7 |
| F | 1 | 9 | 2 | 1 | 7 | 4 | 3 | N | 7 |
| F | 8 | 1 | 3 | 1 | 2 | 5 | 2 | N | 7 |
| F | 7 | 7 | 7 | 2 | 7 | 1 | 7 | P | 6 |
| F | 0 | 8 | 2 | 3 | 1 | 6 | 0 | P | 6 |

FINs extracted from MOM website

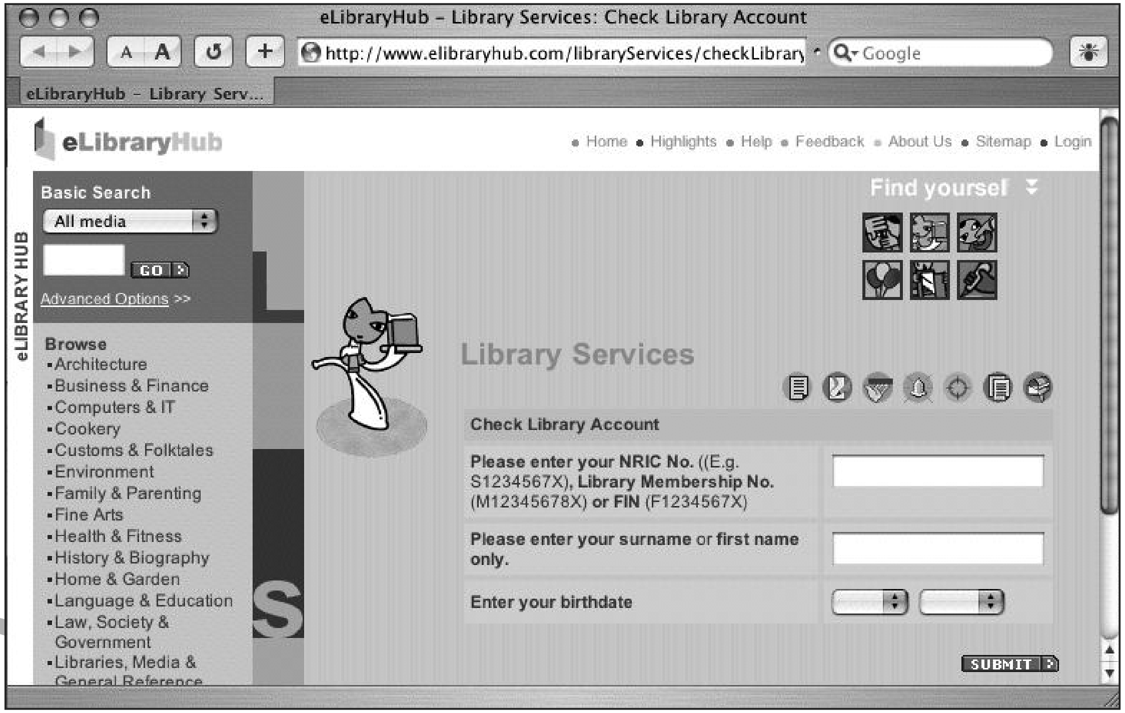
Checksums calculated using formula ①

- 48 out of 1,287 Safety officers are foreigners with FINs
- By inspection, same algorithm and same weights are used but with different check letters:

| ***d*** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Check digit** | **K** | **L** | **M** | **N** | **P** | **Q** | **R** | **T** | **U** | **W** | **X** |

# 21st century UINs - T & G prefix

- Difficult to obtain large list of T-and G-series UINs
  - Children born and foreigners registered during or after 2000
- Solution: Use a brute force approach and rely on the National Library web interface to check accuracy of guess



eLibraryHub – Library Services: Check Library Account

http://www.elibraryhub.com/libraryServices/checkLibrary

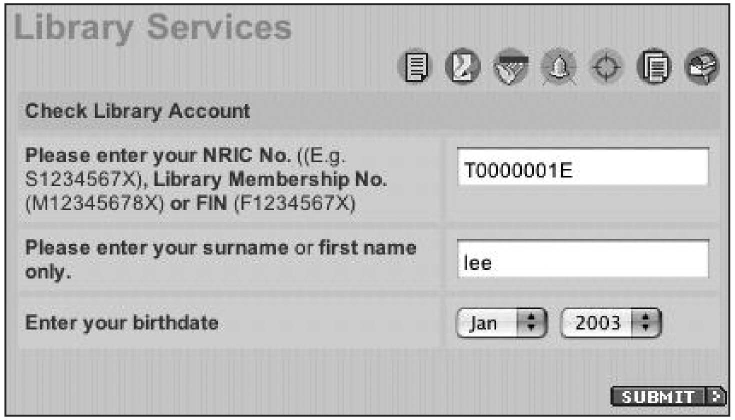
Check Library Account

| | |
|---|---|
| **Please enter your NRIC No.** ((E.g. S1234567X), **Library Membership No.** (M12345678X) **or FIN** (F1234567X) | |
| **Please enter your surname** or **first name only.** | |
| **Enter your birthdate** | |

SUBMIT

# Virtual Experiment

## Verifying UIN/FIN check digits



Library Services

Check Library Account

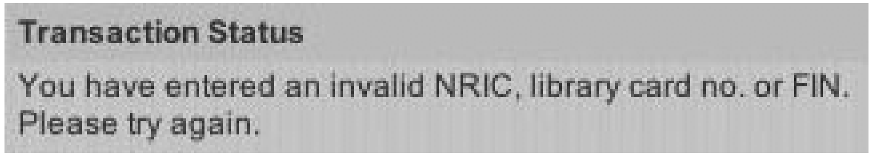
| | |
|---|---|
| **Please enter your NRIC No.** ((E.g. S1234567X), **Library Membership No.** (M12345678X) **or FIN** (F1234567X) | T0000001E |
| **Please enter your surname** or **first name only.** | lee |
| **Enter your birthdate** | Jan 2003 |

*NLB Online Services*

**Enter Test UIN/FIN**
Guess check digit (letter)
corresponding to IC number

**Enter any name / birth month**

Error ①



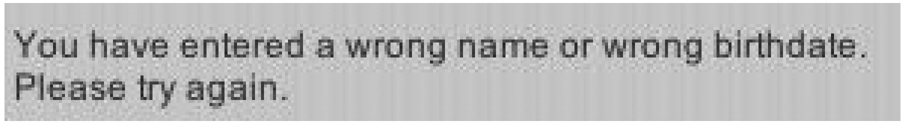
**Transaction Status**

You have entered an invalid NRIC, library card no. or FIN. Please try again.

**Guess incorrect**

or

Error ②



You have entered a wrong name or wrong birthdate. Please try again.

**Guess correct !**

# 21st century UIN/FIN check digit

- By exhaustive search, we conclude for T-prefix UINs
  - Same weighting factors and modulo 11 algorithm is used but
  - Mapping of check digits is shifted 4 places

| *d* | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| S prefix | A | B | C | D | E | F | G | H | I | Z | J |
| T prefix | H | I | Z | J | A | B | C | D | E | F | G |

Shift 4 places

- Similar shift is observed for G-prefix FINs

| *d* | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| F prefix | K | L | M | N | P | Q | R | T | U | W | X |
| G prefix | T | U | W | X | K | L | M | N | P | Q | R |

Shift 4 places

# Universal UIN/FIN Check Digit Algorithm

- For any UIN/FIN of format

$P\ d_1 d_2 d_3 d_4 d_5 d_6 d_7\ C$ where

P = Century prefix {S, T, F or G}

$d_i$ = Number, i = 1..7

C = Check Digit (letter)

$$d = \left\{ d_0 + \left[ (d_1\ d_2\ d_3\ d_4\ d_5\ d_6\ d_7) \bullet (2\ 7\ 6\ 5\ 4\ 3\ 2\ ) \right] \right\} \bmod 11$$

$d_0$ = 0 for P = S or F

= 4 for P = T or G

Check digit is determined by prefix and value of d

| d | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| UIN (S,T prefix) | A | B | C | D | E | F | G | H | I | Z | J |
| FIN (F,G prefix) | K | L | M | N | P | Q | R | T | U | W | X |

# References

- UIN algorithm described in chapter 3 of course notes for NUS Coding Theory course (http://www.math.nus.edu.sg/~ma3218)
  - S & T prefix algorithm confirmed
- No known public references to F, G-prefix FIN algorithm

## Other checksum implementations

- Hong Kong Identity Card http://www.ghs.edu.hk/webtec/lindacws/CS/notes/theory/Data%20Control.pdf
  - HKID uses numerical check digit, e.g. B255241(3)
  - Check digit given by modulo 11 checksum with weights (8, 7, 6, 5, 4, 3, 2) where letter prefix is converted to number A=1, B=2, etc.
  - Use X if remainder is 10
- International Standard Book Number (ISBN) http://en.wikipedia.org/wiki/ISBN
  - ISBN is 9 digit number with check digit given by modulo 11 checksum
  - Weights (1, 2, 3, 4, 5, 6, 7, 8, 9)
  - Use X if remainder is 10

# Points to Ponder

- Why modulo 11 ?
  - For numerical check digit, using modulo 11 allows checksum to be written as single digit (10 = X)
  - For alphabetic check digit, modulo 26 is more likely to detect errors

- Why weights (2, 7, 6, 5, 4, 3, 2) ?
  - Is there an optimal weighting scheme (compare to HKID, ISBN weighting factors) ?

- Why ABCDEFGHI*ZJ* for S-prefix UINs ?

- Will there be U-series UINs in 2200 ?