National Australia Bank

GSE
GUIDE SHARE EUROPE
UK REGION

Understanding Passwords

Nigel Pentland
National Australia Group
Room: Nurburgring
Session: **DB**

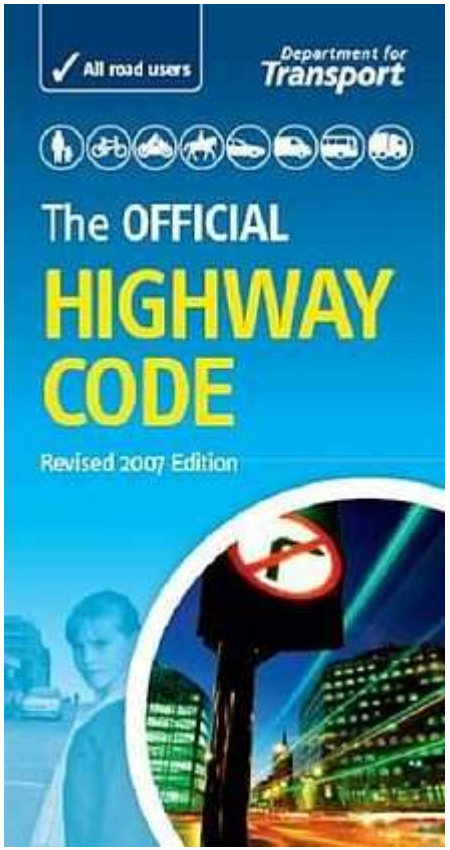
Nigel Pentland

Senior Security Analyst

nigel.pentland@eu.nabgroup.com

0141 223 3179

Road Safety analogy



- Accidents do happen
- They’re not always your fault
- Practice ‘defensive driving’

The standard *Safe Practices for Motor Vehicle Operations*, ANSI/ASSE Z15.1, defines **defensive driving** as "driving to save lives, time, and money, in spite of the conditions around you and the actions of others."[1] This definition is taken from the National Safety Council's Defensive Driving Course.

Don’t

Don’t share / re-use passwords across different web sites / systems !

4 digit PINs

DataGenetics

The Data

almost **3.4 million** four digit passwords. Every single one of the of the 10,000 combinations of digits from 0000 through to 9999 were represented in the dataset.

The most popular password is **1234** …

… it's staggering how popular this password appears to be. Utterly staggering at the lack of imagination …

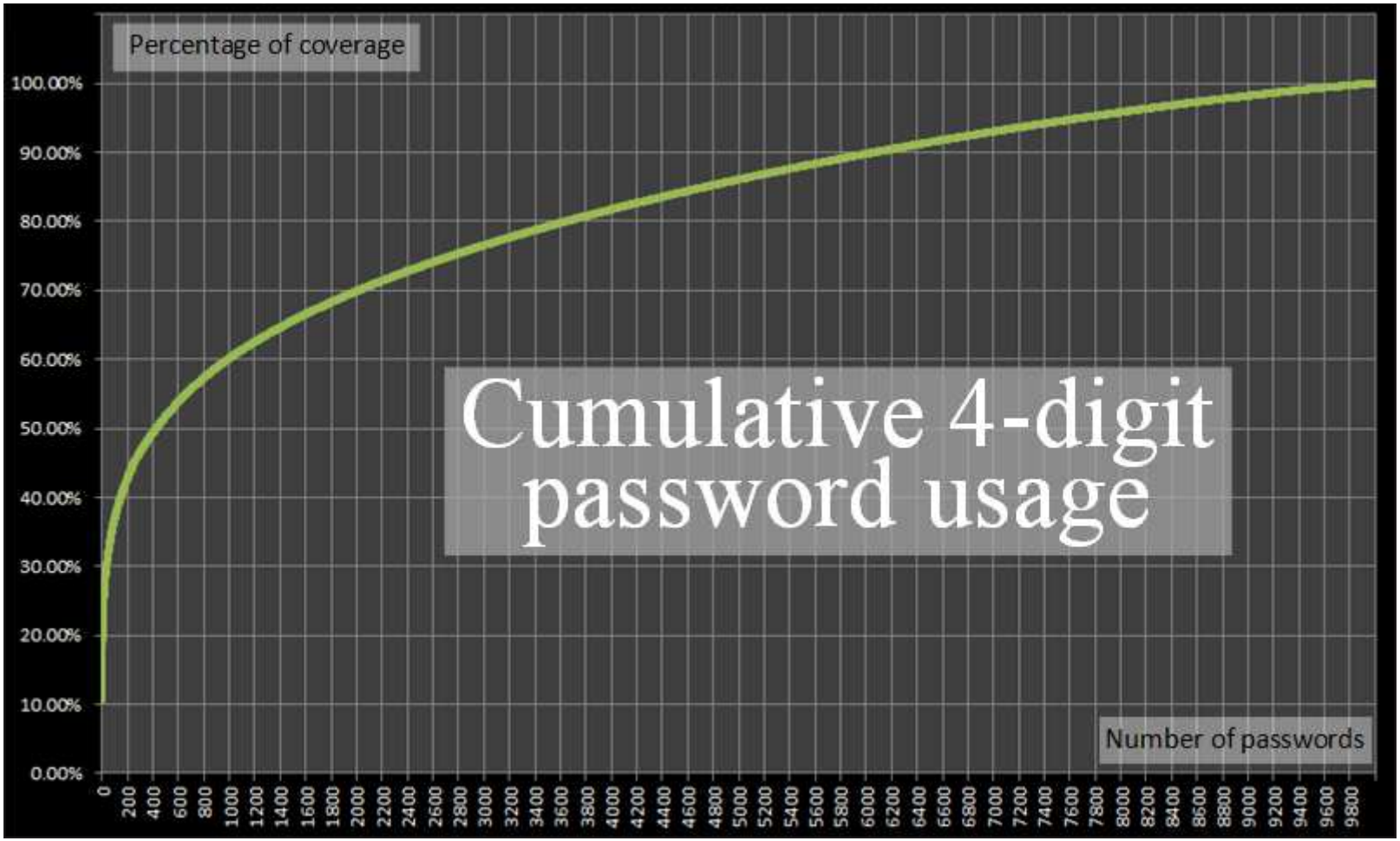
… nearly **11%** of the **3.4 million** passwords are **1234** !!!

	PIN	Freq
#1	1234	10.713%
#2	1111	6.016%
#3	0000	1.881%
#4	1212	1.197%
#5	7777	0.745%
#6	1004	0.616%
#7	2000	0.613%
#8	4444	0.526%
#9	2222	0.516%
#10	6969	0.512%
#11	9999	0.451%
#12	3333	0.419%
#13	5555	0.395%
#14	6666	0.391%
#15	1122	0.366%
#16	1313	0.304%
#17	8888	0.303%
#18	4321	0.293%
#19	2001	0.290%
#20	1010	0.285%

Cumulative Frequency

DataGenetics

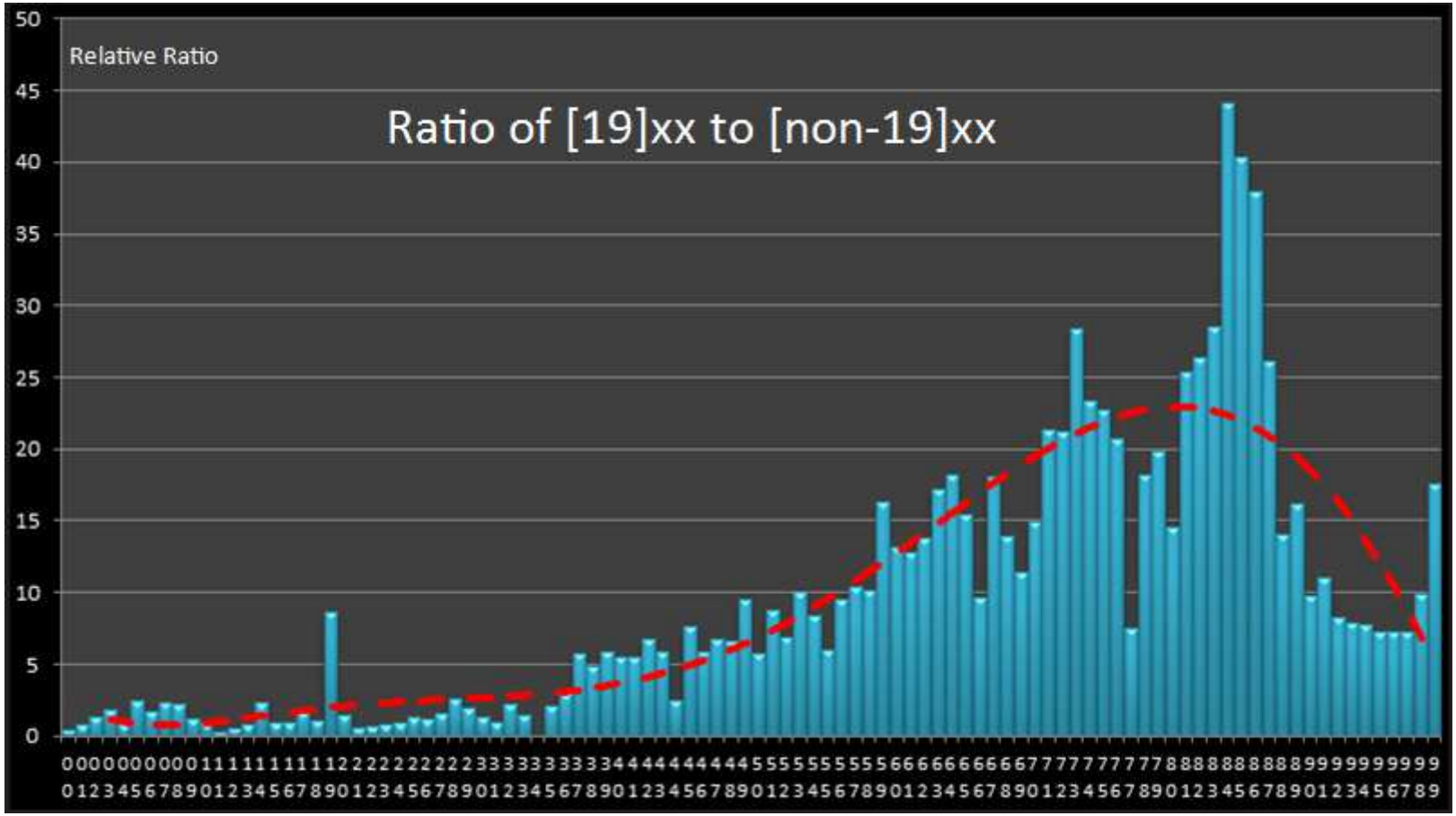
Statistically, one third of all codes can be guessed by trying just 61 distinct combinations!



Memorable Years

DataGenetics

Note: repeating couplets such as 1919



LinkedIn cracked!

6.5 million SHA1 password hashes

Point being, it's no longer if your password becomes compromised, you have to plan on the expectation it will get compromised !

Yes, mine was in there!

Password = **aPDbxGu8**
SHA1 = 51359d602ca2000352aafe06e2baf7e39932f9f9

It's now a 16 random character password !

June 2012

Importance of hashing algorithm strength plus speed

To appreciate just how poor a password hashing choice these unsalted algorithms are, consider this: It took independent security researcher Jeremi Gosney about **six days** to crack more than **90 percent** of the **6.5 million SHA1** hashes exposed in the LinkedIn breach.

He recovered a fifth of the plaintext passwords in just 30 seconds. In the following two hours, he cracked another one-third of them. One day into the exercise, he had recovered a total of 64 percent, and in the five days that followed he cracked another 26 percent.

Importance of hashing algorithm strength plus speed

A key part of his success—besides his **500 million strong word list** and a computer equipped with four AMD Radeon HD6990 graphics cards—was the decision by LinkedIn engineers to hash passwords using **SHA1**.

The algorithm uses a single cryptographic iteration to convert plaintext, allowing Gosney's system to cycle through more than **15.5 billion guesses per second**.

By contrast, algorithms specifically designed to protect passwords are engineered to require significantly more time and computation to convert plaintext into hashes.

For instance, SHA512crypt, which is included in Mac OS X and most Unix-based operating systems, passes text through 5,000 iterations, a hurdle that would have limited Gosney to slightly less than 2,600 guesses per second.

Looking back, one month on…

LinkedIn slapped with $5 million class action suit over leaked passwords

In June 2012 Cryptographic hashes of approximately 6.4 million LinkedIn user passwords were stolen by hackers who then published the stolen hashes online.[29] In response to the incident, LinkedIn asked its users to change their passwords.[30]Security experts criticized LinkedIn for not salting their password file, and instead using a single iteration of SHA-1.[31]

National Institute of Standards and Technology
U.S. Department of Commerce

3 October 2012 — NIST selects Keccak for SHA-3

Guido Bertoni

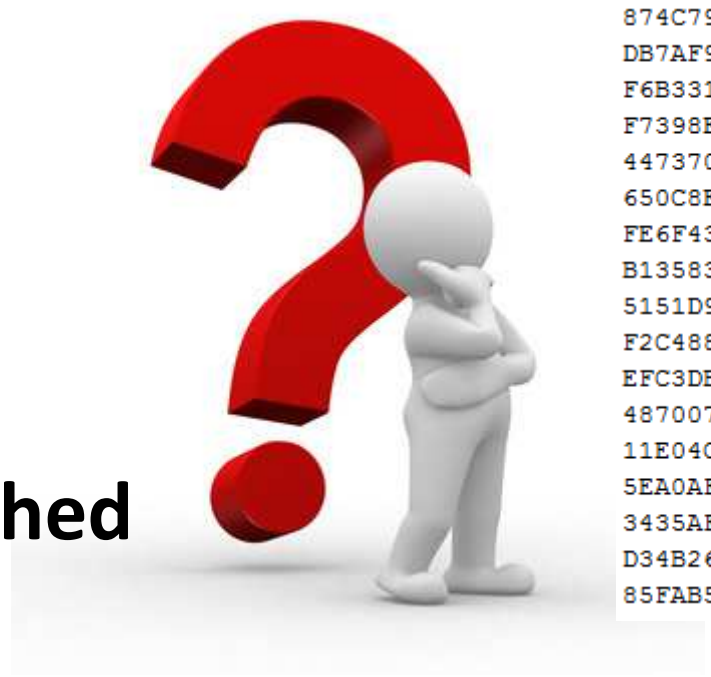
Joan Daemen

Michaël Peeters

Gilles Van Assche

How are passwords stored?

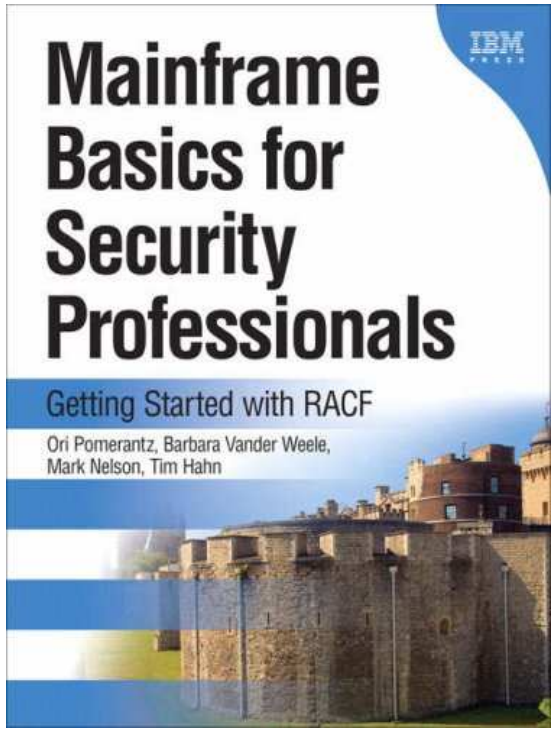
- In clear
- Obfuscated
- Encrypted
- **Securely hashed**



```
874C79D762026B90
DB7AF90E1C74A067
F6B331E3DBDF9EC0
F7398E354727A526
44737030636AA1F2
650C8E06B079ED4A
FE6F43FF1ACEFAC0
B13583B8AAF9A927
5151D9283DCA514C
F2C48856CA6CB5D9
EFC3DB74EB5C3506
4870077C92B608D8
11E04C6EA4325372
5EA0AF5C64CC6BE6
3435AF9C3BEACBD3
D34B268BF122BC6E
85FAB5935EA49C78
```

Important point being, there ***are*** varying levels of security!

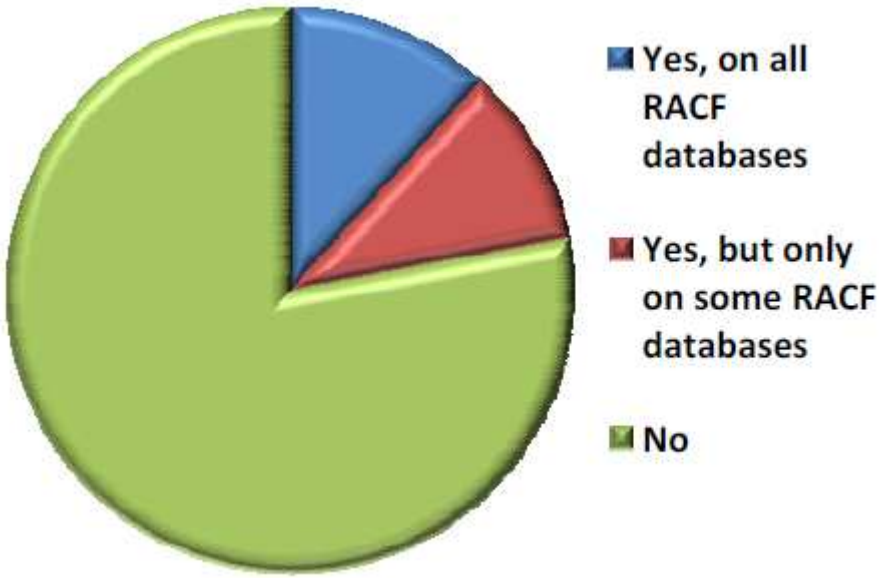
IBM mainframe (RACF)



- Stores passwords **securely hashed**
- **Well** chosen password is **secure**
- **Poorly** chosen password is **vulnerable**

RSH RACF Survey

Are mixed case passwords enabled on your mainframe?



Robert S Hansel – RSH Consulting – Survey taken January 2012

Password cracking…

8 character RACF password, but **securely hashed**
Speed of trying approximately 3 million guesses per minute (on a **PC**!)

Guessable?

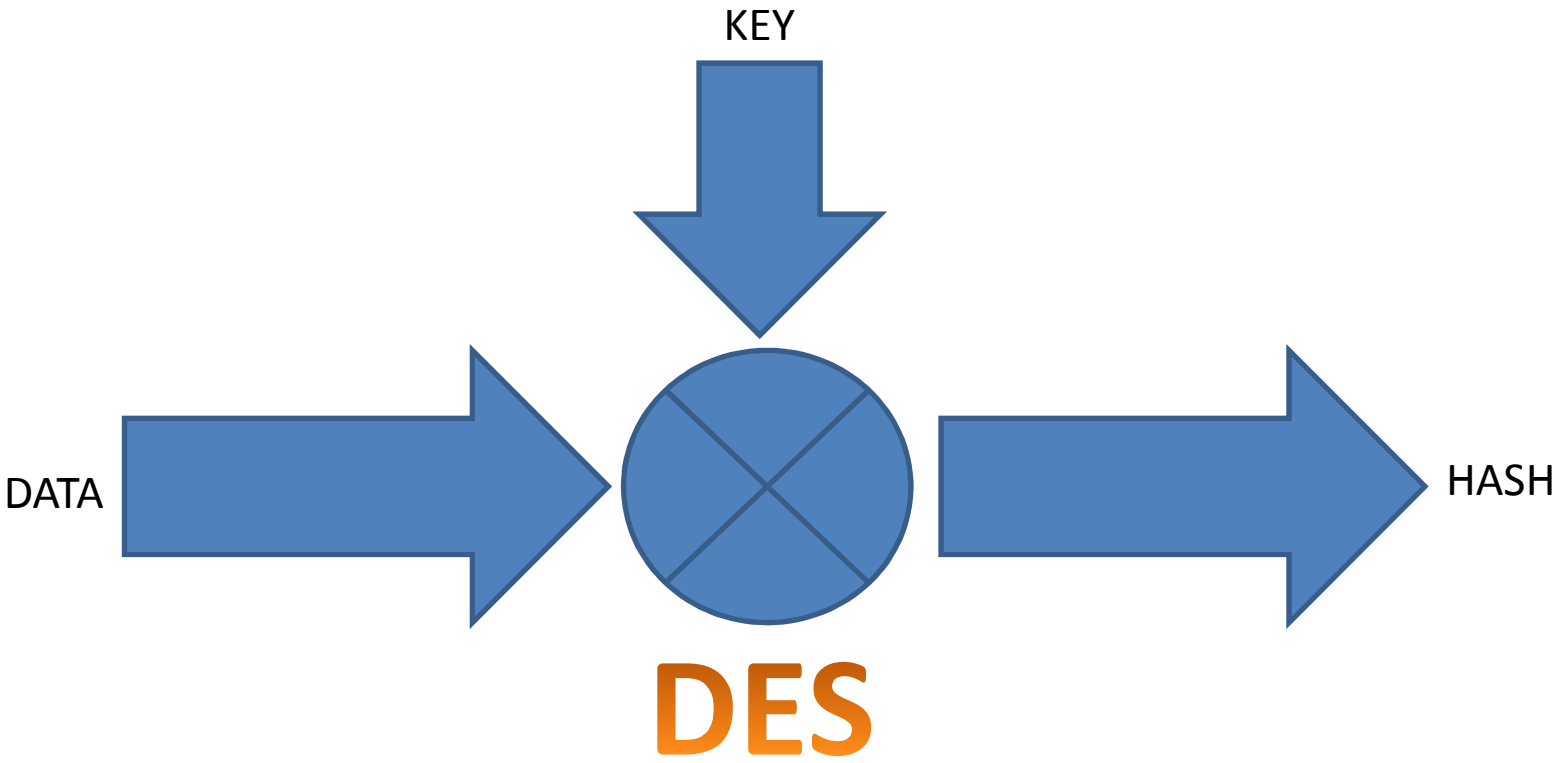
Password dictionary 100,000 words
Add up to 2 numeric digits on end
Time taken to crack a poor password
is about **2 minutes**

For dictionary words think
words, names, places, etc.

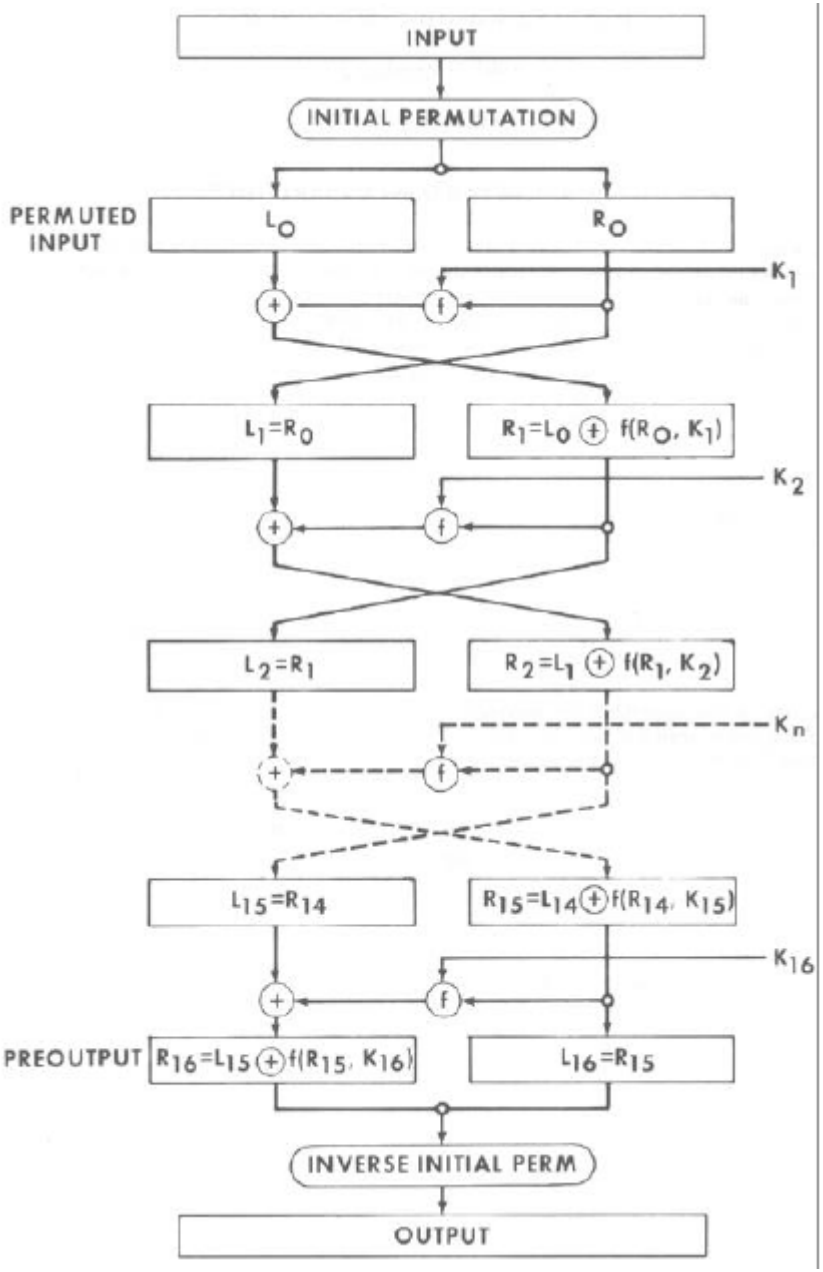
Not guessable?

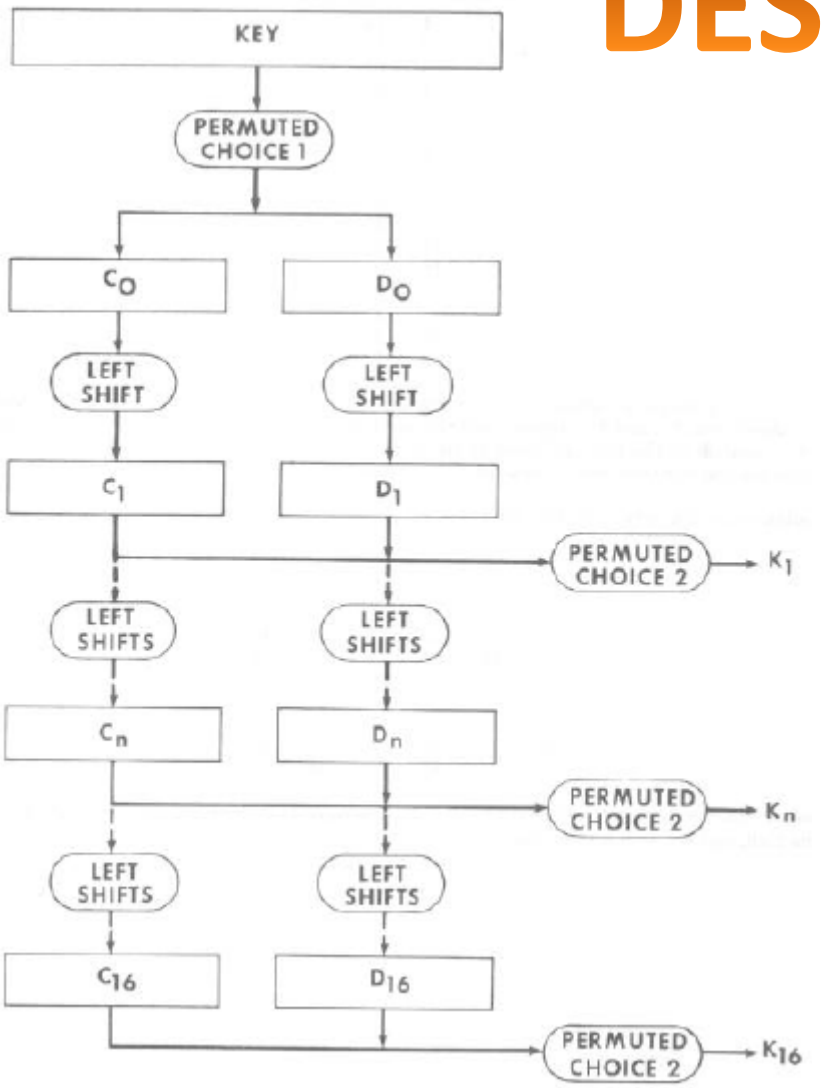
Time to try every combination
of letters and numbers
(36 characters, i.e. ignoring
special characters, just taking
uppercase letters plus numbers)
is about **2 years**

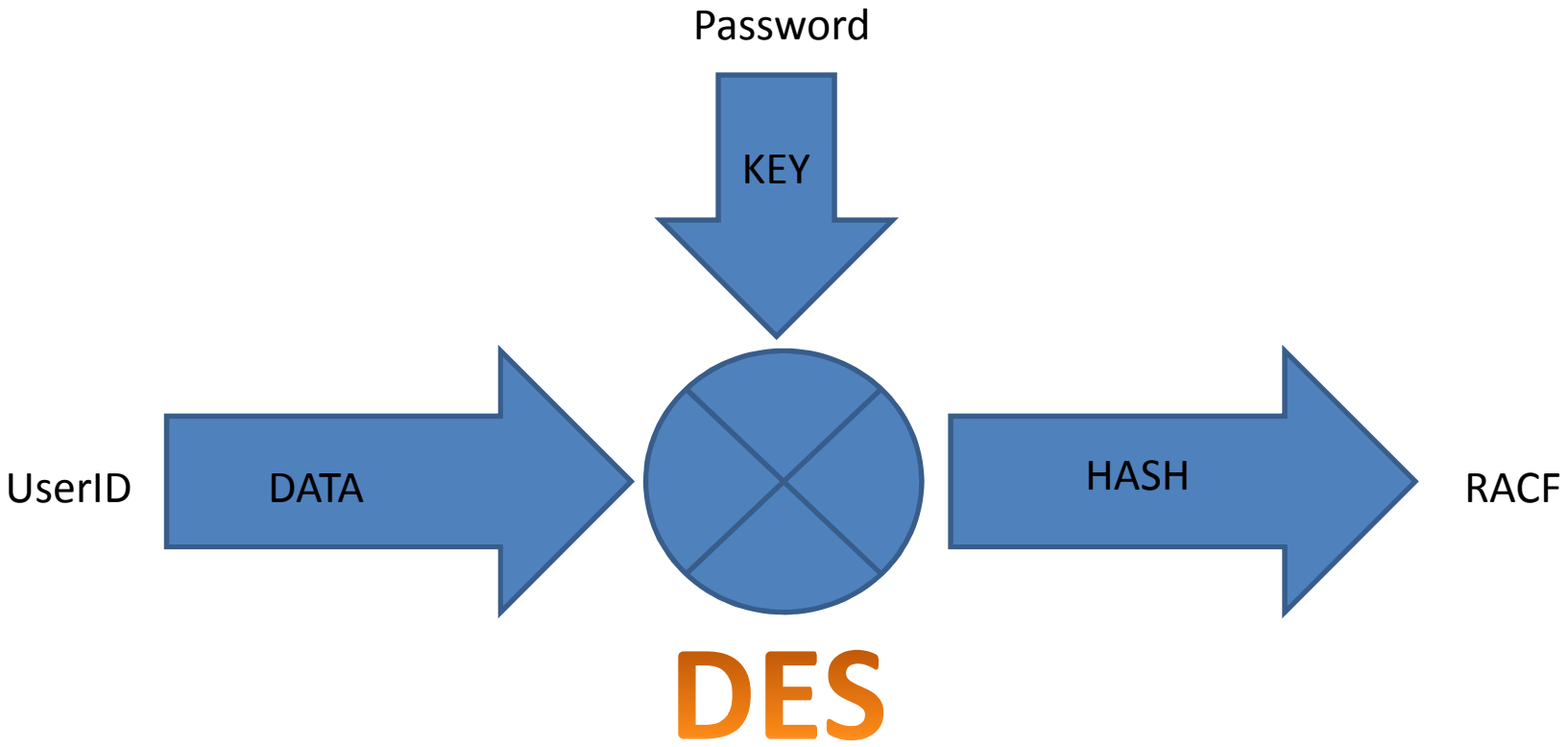
So, how *does*, RACF store passwords?

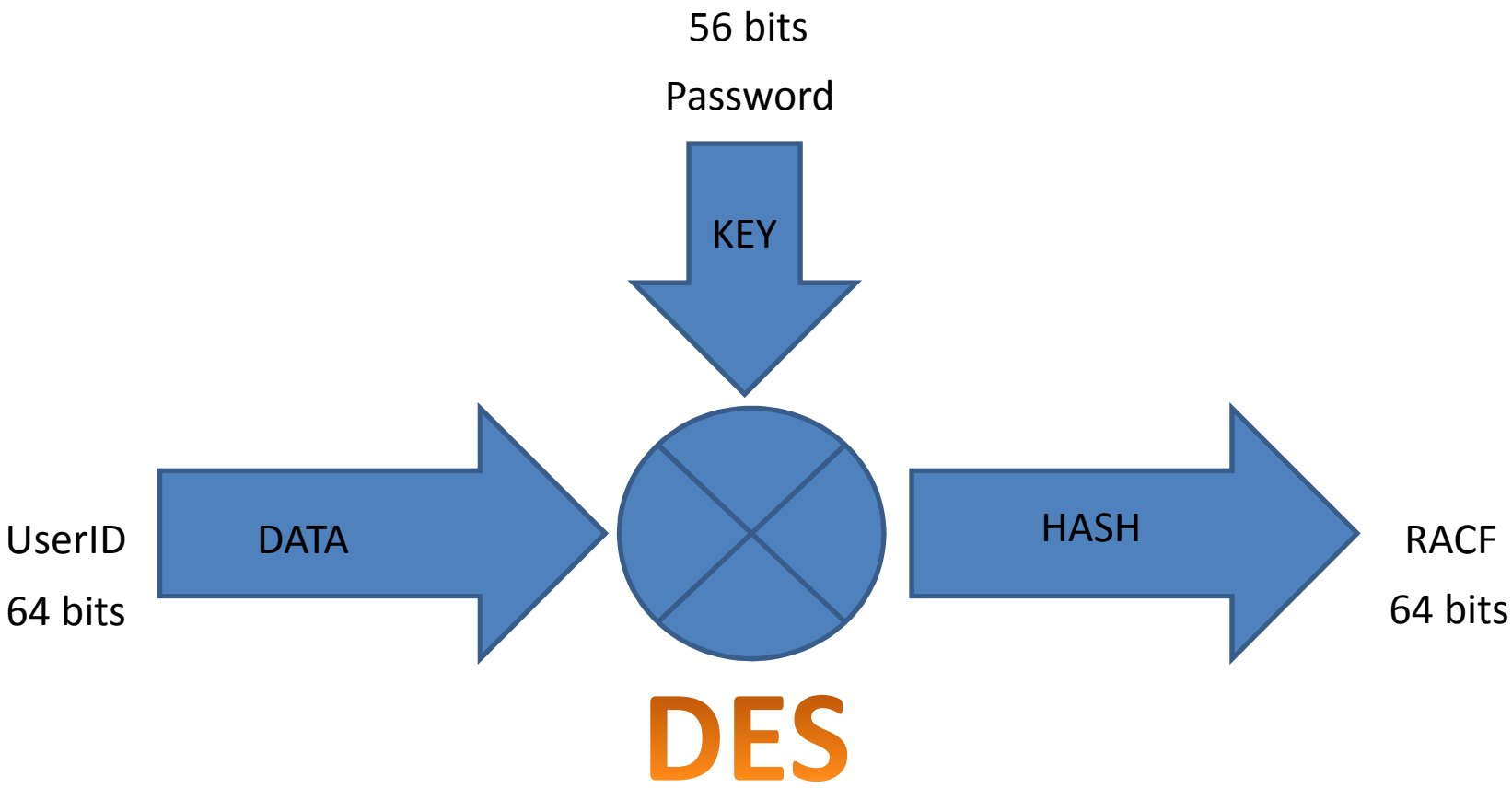

KEY
DATA
HASH
DES

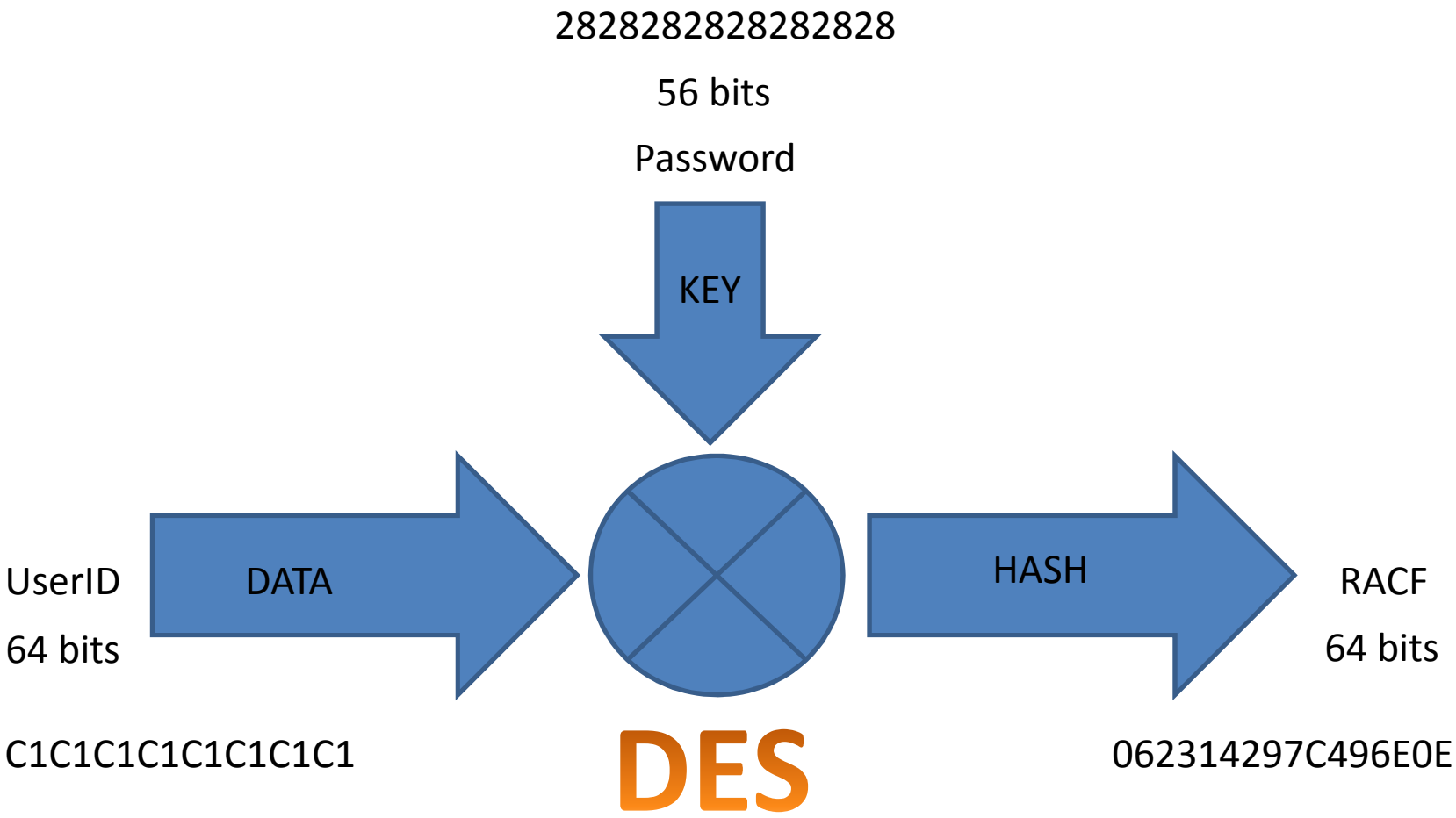
DES


INPUT
INITIAL PERMUTATION
PERMUTED INPUT
L_0
R_0
K_1
$L_1=R_0$
$R_1=L_0 \oplus f(R_0, K_1)$
K_2
$L_2=R_1$
$R_2=L_1 \oplus f(R_1, K_2)$
K_n
$L_{15}=R_{14}$
$R_{15}=L_{14} \oplus f(R_{14}, K_{15})$
K_{16}
PREOUTPUT
$R_{16}=L_{15} \oplus f(R_{15}, K_{16})$
$L_{16}=R_{15}$
INVERSE INITIAL PERM
OUTPUT


KEY
PERMUTED CHOICE 1
C_0
D_0
LEFT SHIFT
LEFT SHIFT
C_1
D_1
PERMUTED CHOICE 2
K_1
LEFT SHIFTS
LEFT SHIFTS
C_n
D_n
PERMUTED CHOICE 2
K_n
LEFT SHIFTS
LEFT SHIFTS
C_{16}
D_{16}
PERMUTED CHOICE 2
K_{16}


Password
KEY
UserID
DATA
HASH
RACF
DES


56 bits
Password
KEY
UserID
64 bits
DATA
HASH
RACF
64 bits
DES



UserID = AAAAAAAA
Password = AAAAAAAA

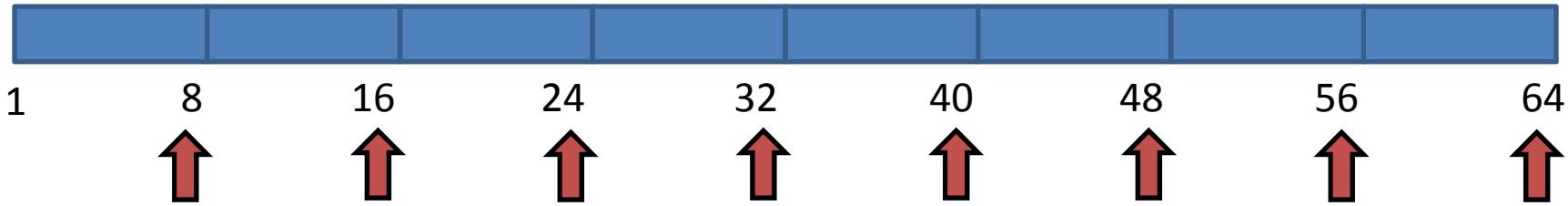
EBCDIC (A) = C1

Password = AAAAAAAA

Becomes = 2828282828282828

In other words why/how does A map to 28 for the password?

Numbering notation for bits in a DES key, showing where the parity bits are



Parity bits, not used during DES encrypt / decrypt, hence **56 bit actual key**

So each EBCDIC character of password is represented by 7 bits

OK – let's take a closer look at the possible EBCDIC characters

A	C1	11000001		V	E5	11100101
B	C2	11000010		W	E6	11100110
C	C3	11000011		X	E7	11100111
D	C4	11000100		Y	E8	11101000
E	C5	11000101		Z	E9	11101001
F	C6	11000110				
G	C7	11000111		0	F0	11110000
H	C8	11001000		1	F1	11110001
I	C9	11001001		2	F2	11110010
J	D1	11010001		3	F3	11110011
K	D2	11010010		4	F4	11110100
L	D3	11010011		5	F5	11110101
M	D4	11010100		6	F6	11110110
N	D5	11000101		7	F7	11110111
O	D6	11000110		8	F8	11111000
P	D7	11010111		9	F9	11111001
Q	D8	11011000				
R	D9	11011001				
S	E2	11100010				
T	E3	11100011				
U	E4	11100100				

So, looking at the 8 bit EBCDIC Representations of these characters Is there an easy way to reduce to 7 bits?

But the parity bit is the right most bit and the redundant bit is left most?

Shift left !

So, let’s try this with our example…

A = C1 = 11000001

Shift left = 10000010 = 82

But we are expecting a value of hex 28 = 00101000

So, not quite!

So, could it have been obfuscated?

Standard obfuscation is XOR 55 , but it would make more sense to obfuscate before shifting

A = C1 = 11000001
55 = 01010101
XOR = 10010100
SHL = 00101000
HEX = 28

OK – now we all know how to take a RACF database, compute a password to find out if it is a match, for the password held DES encrypted in the database.

But what software **already exists** out there to do this for us?

1997 - PWCHECK and PWCHECK-PRO
Kurt Meiser / Peter Goldis

2000 - Cracker
Thierry Falissard

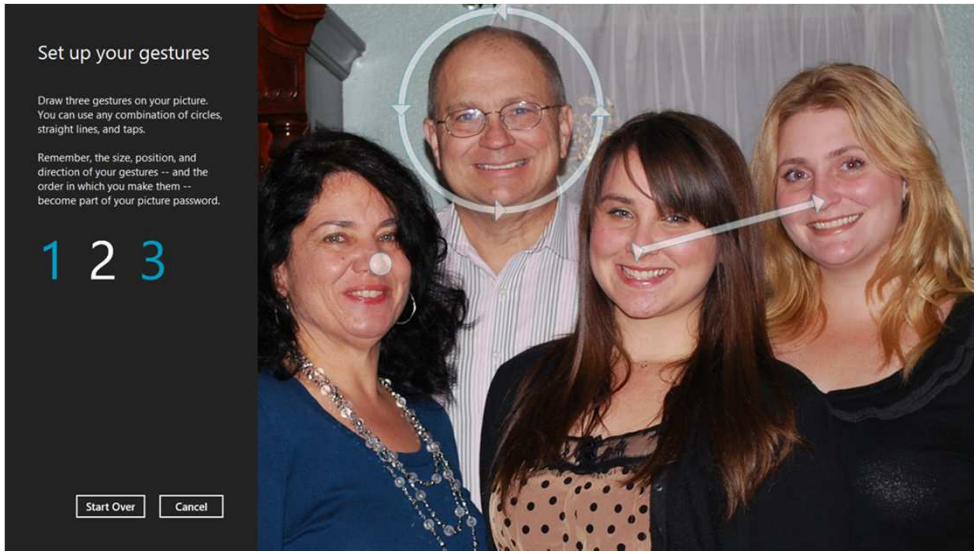
2000 - CRACF
2008 - WEAKWORD
2012 - RACFSNOW
Nigel Pentland

2012 - John the Ripper
(year RACF functionality was added to JtR)

Windows 8

Two new types of password

- four digit PIN
- picture password



Release date announced as 26th October 2012

Ford Key Free Login

- Uses Smart Phone for all passwords
- Car security has come a long way
- Concept ***videoware***

References

http://www.datagenetics.com/blog/september32012/

http://arstechnica.com/security/2012/08/passwords-under-assault/

http://thepasswordproject.com/passpal

http://thepasswordproject.com/leaked_password_lists_and_dictionaries

‘Bad passwords never die
in fact, they don’t even fade away.’

National Australia Bank

GSE
GUIDE SHARE EUROPE
UK REGION

Understanding Passwords

Nigel Pentland
National Australia Group
Room: Nurburgring
Session: **DB**