

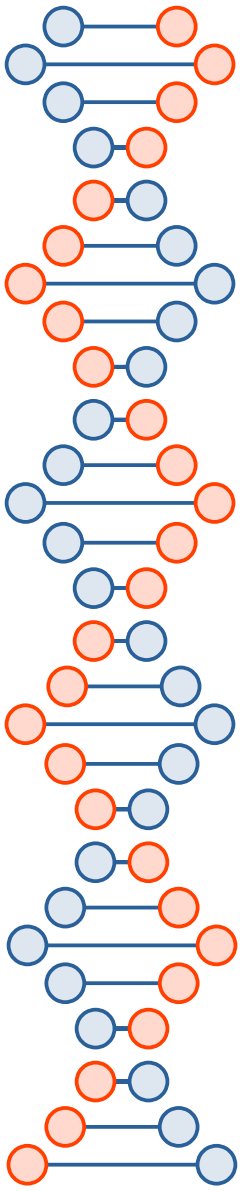


Hash functions

`jedimaestro@asu.edu`

Why hash functions?

- Speed
- Error detection (*e.g.*, checksum)
- Security and privacy





Why cryptographic hash functions?

- Unique identifier for an object
- Integrity of an object
 - *E.g.*, message authentication codes
- Digital signatures
- Passwords
- Proof of work

Example

Input

Fox

The red fox
jumps over
the blue dog

The red fox
jumps over
the blue dog

The red fox
jumps over
the blue dog

The red fox
jumps over
the blue dog

cryptographic
hash
function

cryptographic
hash
function

cryptographic
hash
function

cryptographic
hash
function

cryptographic
hash
function

Digest

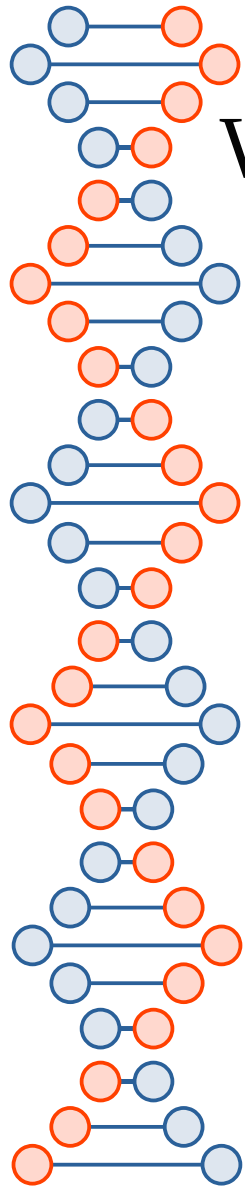
DFCD 3454 BBEA 788A 751A
696C 24D9 7009 CA99 2D17

0086 46BB FB7D CBE2 823C
ACC7 6CD1 90B1 EE6E 3ABC

8FD8 7558 7851 4F32 D1C6
76B1 79A9 0DA4 AEFE 4819

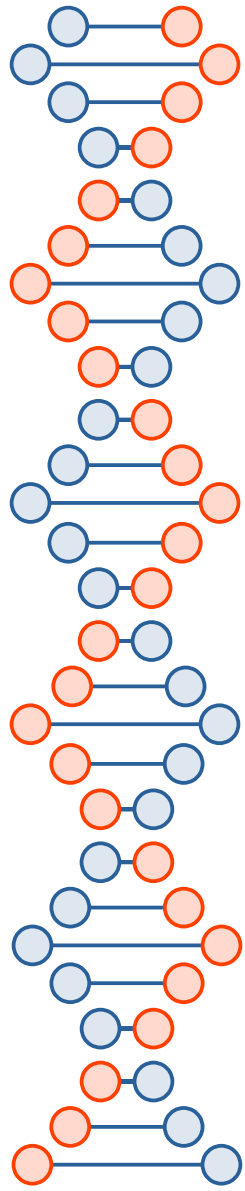
FCD3 7FDB 5AF2 C6FF 915F
D401 C0A9 7D9A 46AF FB45

8ACA D682 D588 4C75 4BF4
1799 7D88 BCF8 92B9 6A6C



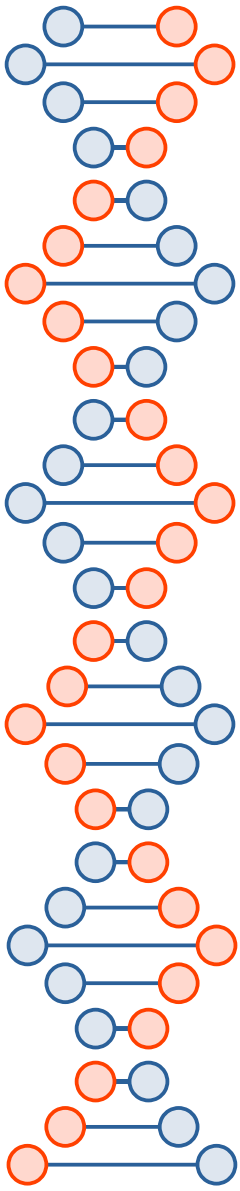
What makes a hash function cryptographic?

- One-way function
- Deterministic (same input, same output)
- Infeasible to find message that digests to specific hash value
- Infeasible to find two messages that digest to the same hash
- Avalanche effect (small change in message leads to big changes in digest---digests seemingly uncorrelated)
- *Still want it to be quick*



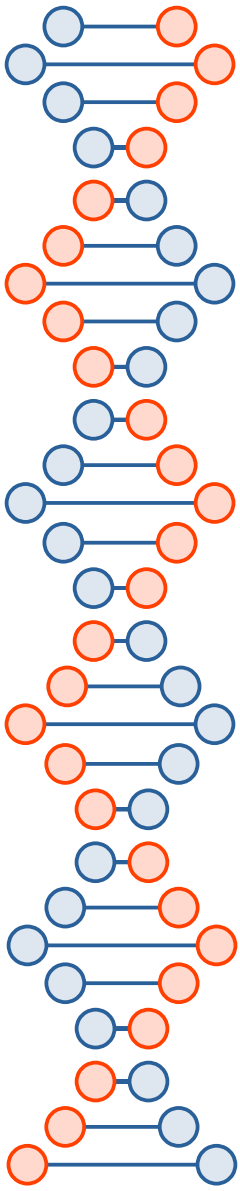
Algorithms

- MD5: 128-bit digest, seriously broken
- SHA-1: 160-bit digest, not secure against well-funded adversaries
- SHA-3: 224 to 512 bit digest, adopted in August of 2015
- CRC32: not cryptographic, very poor choice



Algorithms

- MD5: 128-bit digest, seriously broken
- SHA-1: 160-bit digest, not secure against well-funded adversaries
- SHA-3: 224 to 512 bit digest, adopted in August of 2015
- CRC32: not cryptographic, very poor choice



Property #1

- Pre-image resistance
- Given h , it should be infeasible to find m such that $h = \text{hash}(m)$

Neither MD5 nor SHA-3 are broken in this way, but MD5 digests are small.



Property #2

- Second pre-image resistance
- Given a message m_1 , it should be infeasible to find another message m_2 such that...
 $hash(m_1) = hash(m_2)$

Neither MD5 nor SHA-3 are broken in this way, but MD5 digests are small.



Property #3

- Collision resistance
- It should be infeasible to find two messages, m_1 and m_2 such that...
 $hash(m_1) = hash(m_2)$

SHA-3 is not broken in this way, MD5 broken in seconds on your laptop, SHA-1 with \$100K or so.

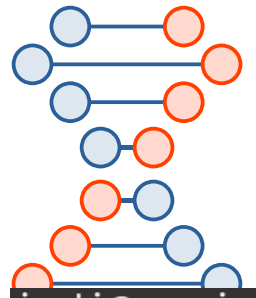


Wang Xiaoyun




- Tsinghua University
- Contributed a lot of ideas to cracking MD5, SHA-0, and SHA-1

Length extension attack



```
jedi@mariposa:~$ echo "password='lDEnr45#d3'&donut=choc&quantity=1" | md5sum
91a9fc74a98997dba291a26a91c9648e -
jedi@mariposa:~$ echo "password='lDEnr45#d3'&donut=choc&quantity=100" | md5sum
8fdd2d4515bcba887b1b80a653f21e0c -
```



```
jedi@mariposa:~$ echo "password=[REDACTED]&donut=choc&quantity=1" | md5sum
91a9fc74a98997dba291a26a91c9648e -
jedi@mariposa:~$ echo "password=[REDACTED]&donut=choc&quantity=100" | md5sum
8fdd2d4515bcba887b1b80a653f21e0c -
```

MD5 and SHA-1 vulnerable, SHA-3 is not

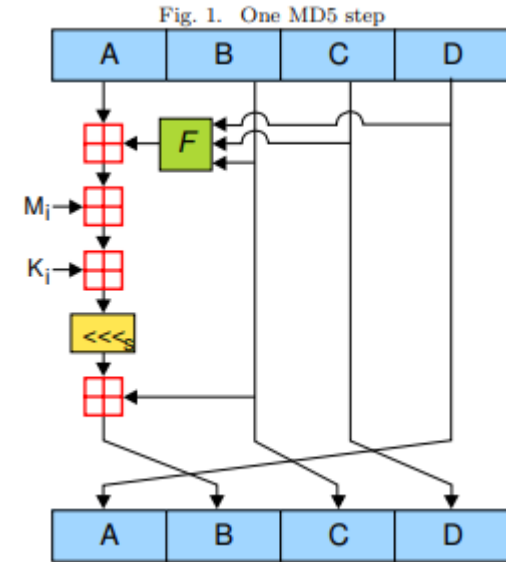


Length extension attack

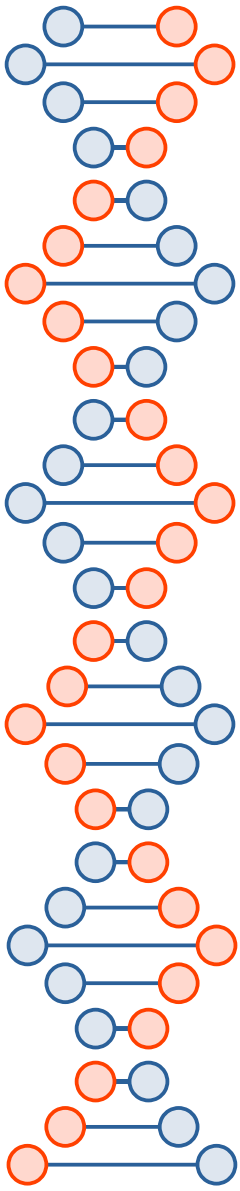
- One issue is if the attacker doesn't know the password
- Another issue is if the password is different but the attacker finds a collision later on
- MD5 and SHA-1 are vulnerable, SHA-3 is not

MD5

- Pad to multiple of 512 bits
- 4 rounds
- 4 32-bit words at a time
- Concatenate them at the end for a 128-bit digest
- F is non-linear, varies by round

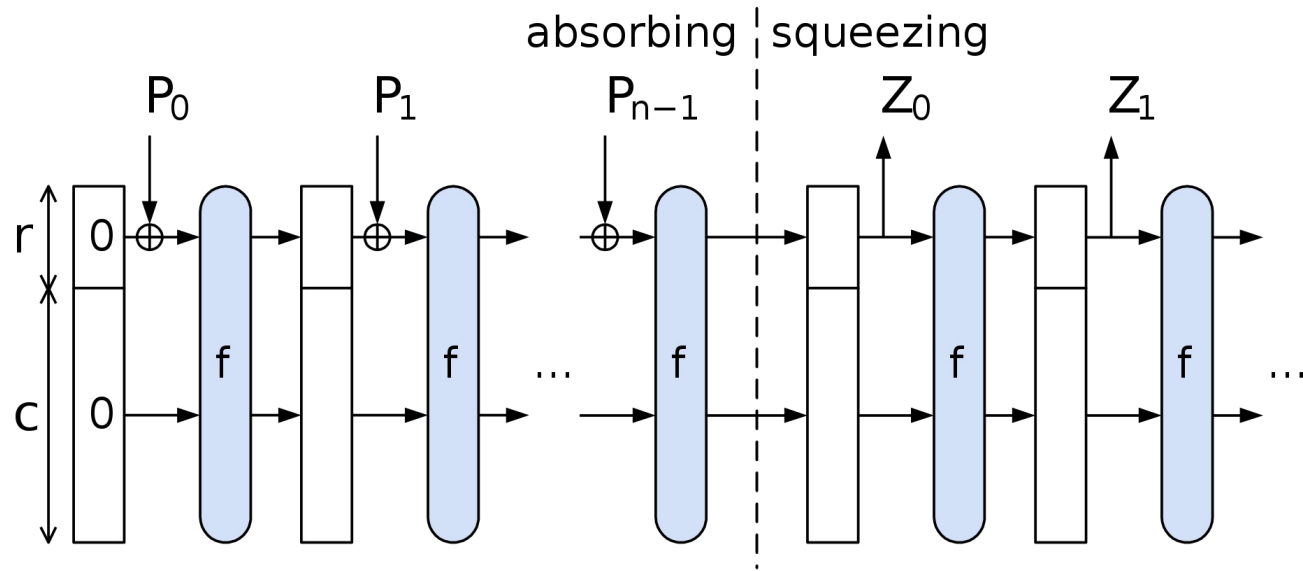


Round (i)	$F(X, Y, Z)$	g
0	$(X \wedge Y) \vee (\neg X \wedge Z)$	i
1	$(X \wedge Z) \vee (Y \wedge \neg Z)$	$(5 \times i + 1) \bmod 16$
2	$(X \oplus Y \oplus Z)$	$i(3 \times i + 5) \bmod 16$
3	$(Y \oplus (X \vee \neg Z))$	$(7 \times i) \bmod 16$

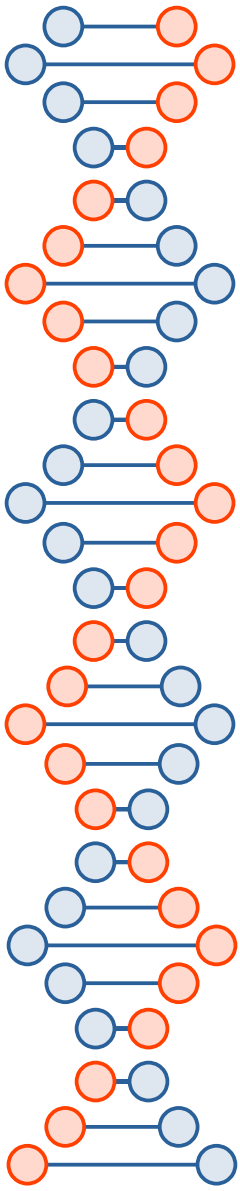


SHA-3

- Sponge construction, 1600 bits of internal state



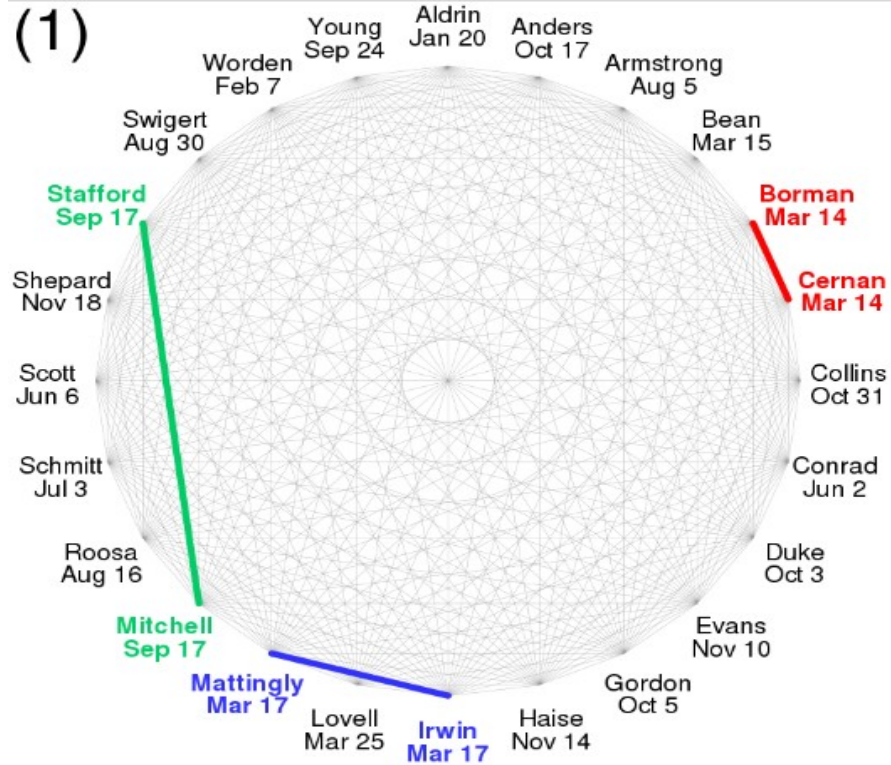
<https://en.wikipedia.org/wiki/SHA-3>



Birthday attack

- Probability of collision is 1 in 2^n , but the expected number of hashes until two of them collide is $\sqrt{2^n}=2^{n/2}$
 - Why? Third try has two opportunities to collide, fourth has three opportunities, fifth has six, and so on...

24 people, same birthday?



Chosen-prefix collision attack

- Given two prefixes p_1 and p_2 , find m_1 and m_2 such that $hash(p_1 || m_1) = hash(p_2 || m_2)$
- p_1 and p_2 could be domain names in a certificate, images, PDFs, etc. ... any digital image.



Ingredients for a practical chosen prefix attack on MD5

- Collision attack on MD5
 - That works for any initialization vector (so you can put bits in front)
- Length extension attack
 - So you can put identical bits on the end
- Birthday attack
 - So you can bridge the prefix to a block that meets the requirements of the collision attack

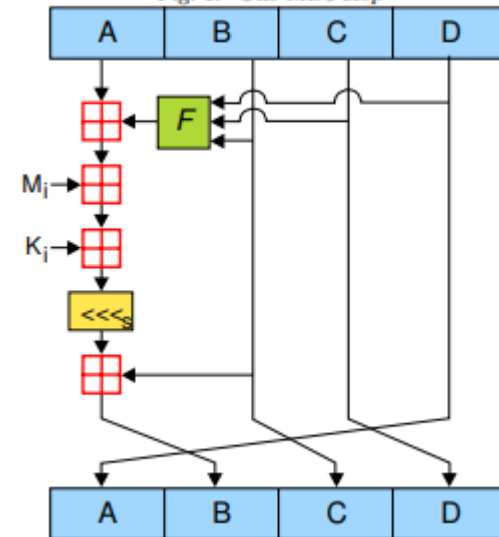
MD5 collision attack by Wang and Yu

$$C_0 = (0, 0, 0, 0, 2^{31}, 0, 0, 0, 0, 0, 0, 2^{15}, 0, 0, 2^{31}, 0)$$

and

$$C_1 = (0, 0, 0, 0, 2^{31}, 0, 0, 0, 0, 0, 0, -2^{15}, 0, 0, 2^{31}, 0)$$

Fig. 1. One MD5 step



Round (i)	$F(X, Y, Z)$	g
0	$(X \wedge Y) \vee (\neg X \wedge Z)$	i
1	$(X \wedge Z) \vee (Y \wedge \neg Z)$	$(5 \times i + 1) \bmod 16$
2	$(X \oplus Y \oplus Z)$	$i(3 \times i + 5) \bmod 16$
3	$(Y \oplus (X \vee \neg Z))$	$(7 \times i) \bmod 16$

<http://koclab.cs.ucsb.edu/teaching/cren/project/2008/savage.pdf>



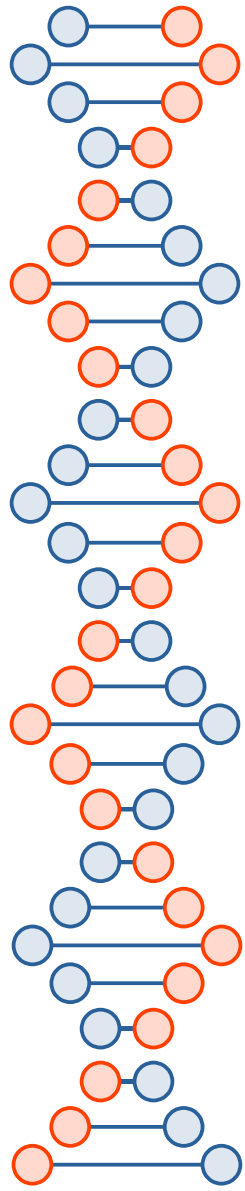
An example

Both have digest 79054025255fb1a26e4bc422aef54eb4

```
d131dd02c5e6eec4693d9a0698aff95c2fcab58712467eab4004583eb8fb7f89
55ad340609f4b30283e488832571415a085125e8f7cdc99fd91dbd7280373c5b
d8823e3156348f5bae6dacd436c919c6dd53e2b487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080a80d1ec69821bcb6a8839396f9652b6ff72a70
```

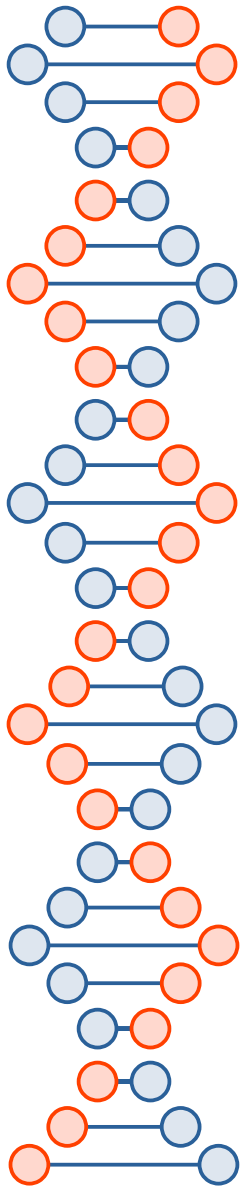
```
d131dd02c5e6eec4693d9a0698aff95c2fcab50712467eab4004583eb8fb7f89
55ad340609f4b30283e4888325f1415a085125e8f7cdc99fd91dbd7280373c5b
d8823e3156348f5bae6dacd436c919c6dd53e23487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080280d1ec69821bcb6a8839396f965ab6ff72a70
```

<https://www.mscs.dal.ca/~selinger/md5collision/>



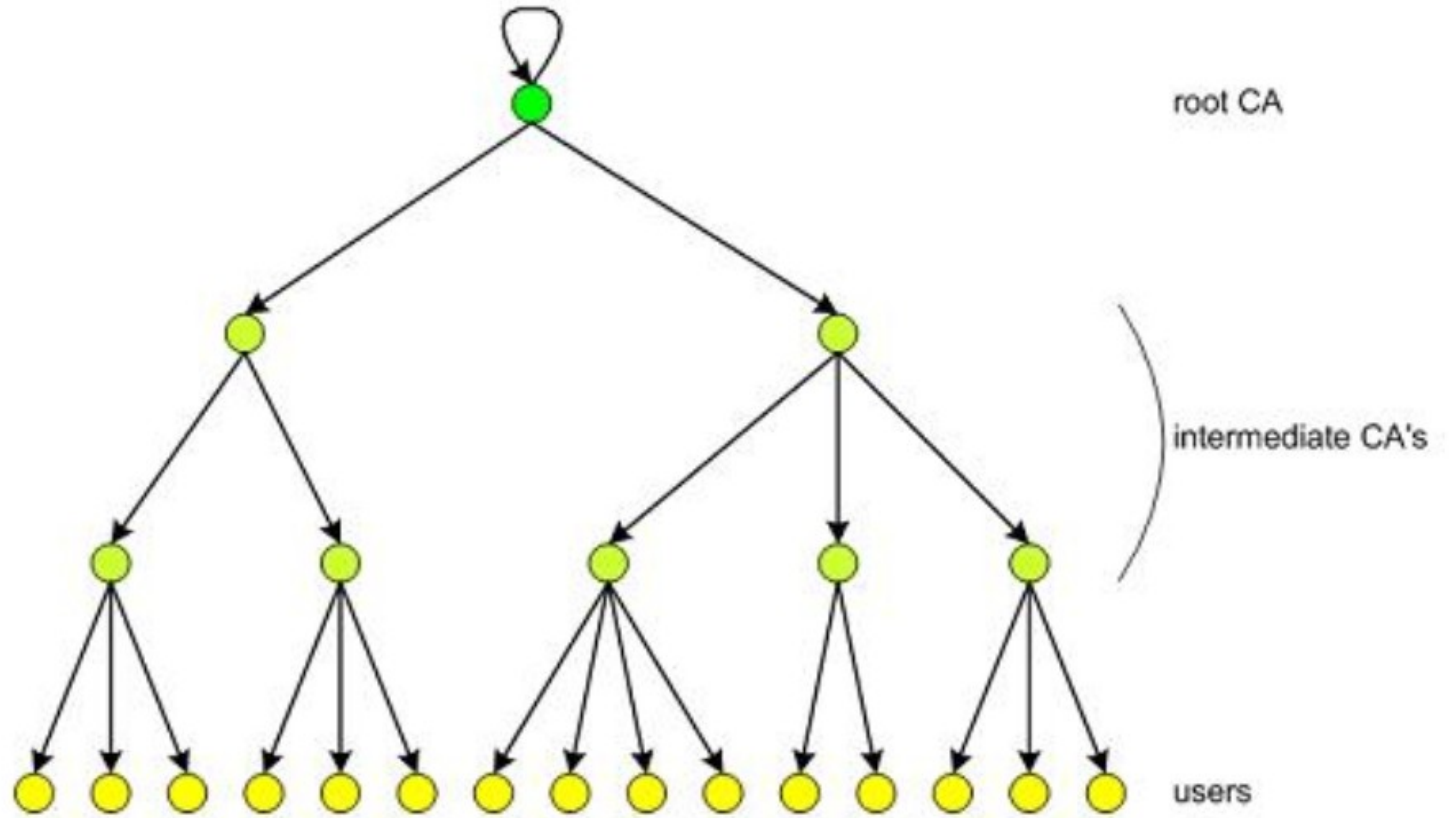
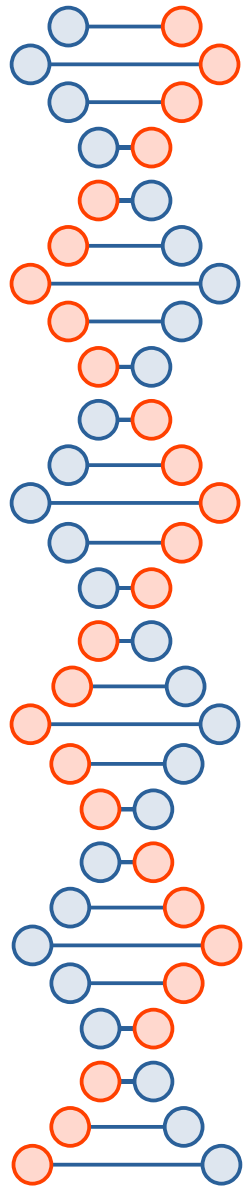
Short Chosen-Prefix Collisions for MD5 and the Creation of a Rogue CA Certificate

Marc Stevens¹, Alexander Sotirov²,
Jacob Appelbaum³, Arjen Lenstra^{4,5}, David Molnar⁶,
Dag Arne Osvik⁴, and Benne de Weger⁷



legitimate website certificate		rogue CA certificate	
serial number	chosen prefixes	serial number	tumor
commercial CA name		commercial CA name	
validity period		validity period	
domain name		rogue CA name	
		1024 bit RSA public key	
		v3 extensions	
		“CA = TRUE”	
2048 bit RSA public key	collision bits		
v3 extensions	identical suffixes		
“CA = FALSE”			

Fig. 1. The to-be-signed parts of the colliding certificates



Slide from MD5 Considered Harmful Today, Creating a rogue CA certificate by Sotirov *et al.*



References

- [Cryptography Engineering] *Cryptography Engineering: Design Principles and Applications*, by Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno. Wiley Publishing, 2010.
- Lots of images and info plagiarized from Wikipedia