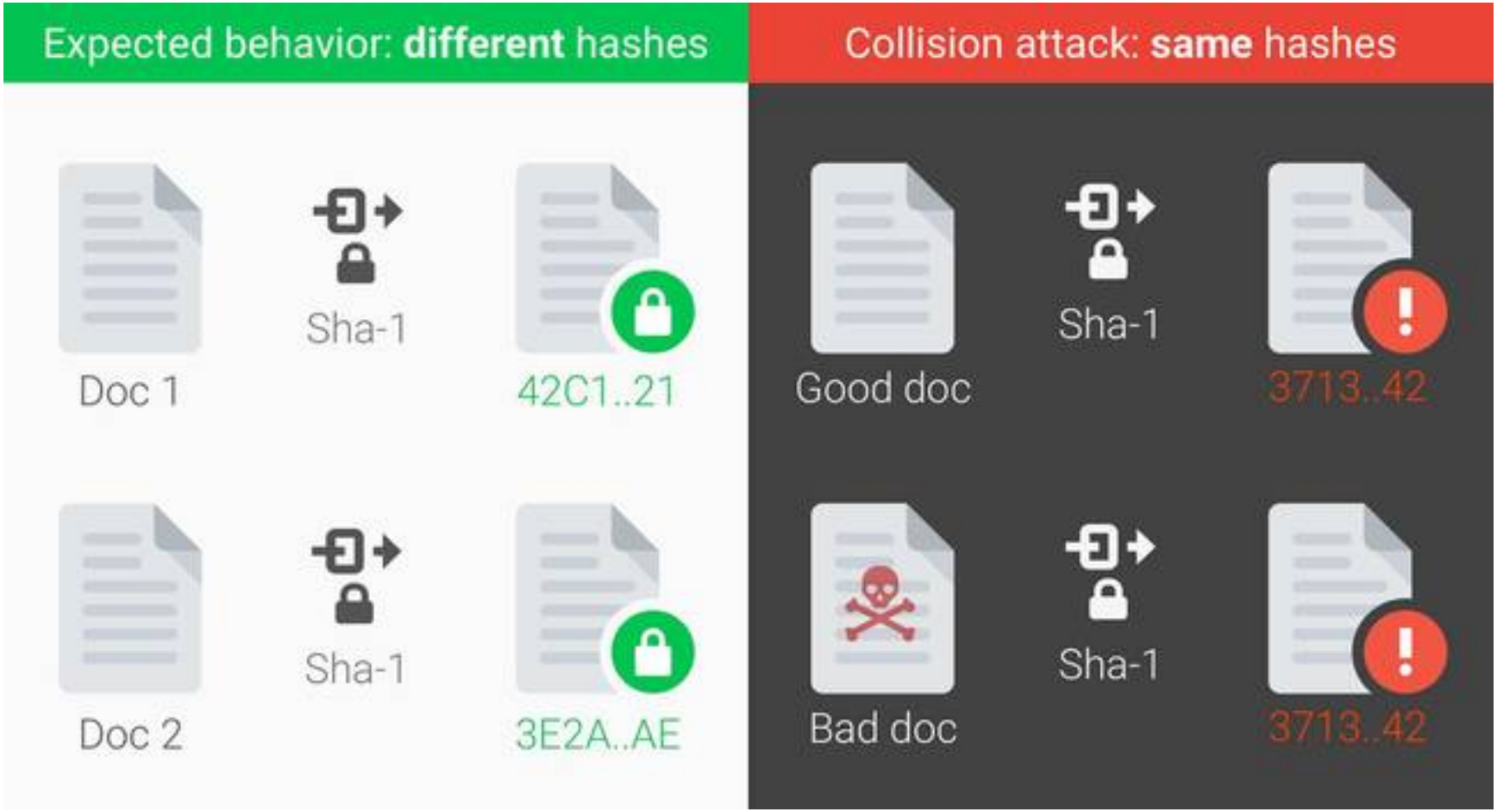
Key Management

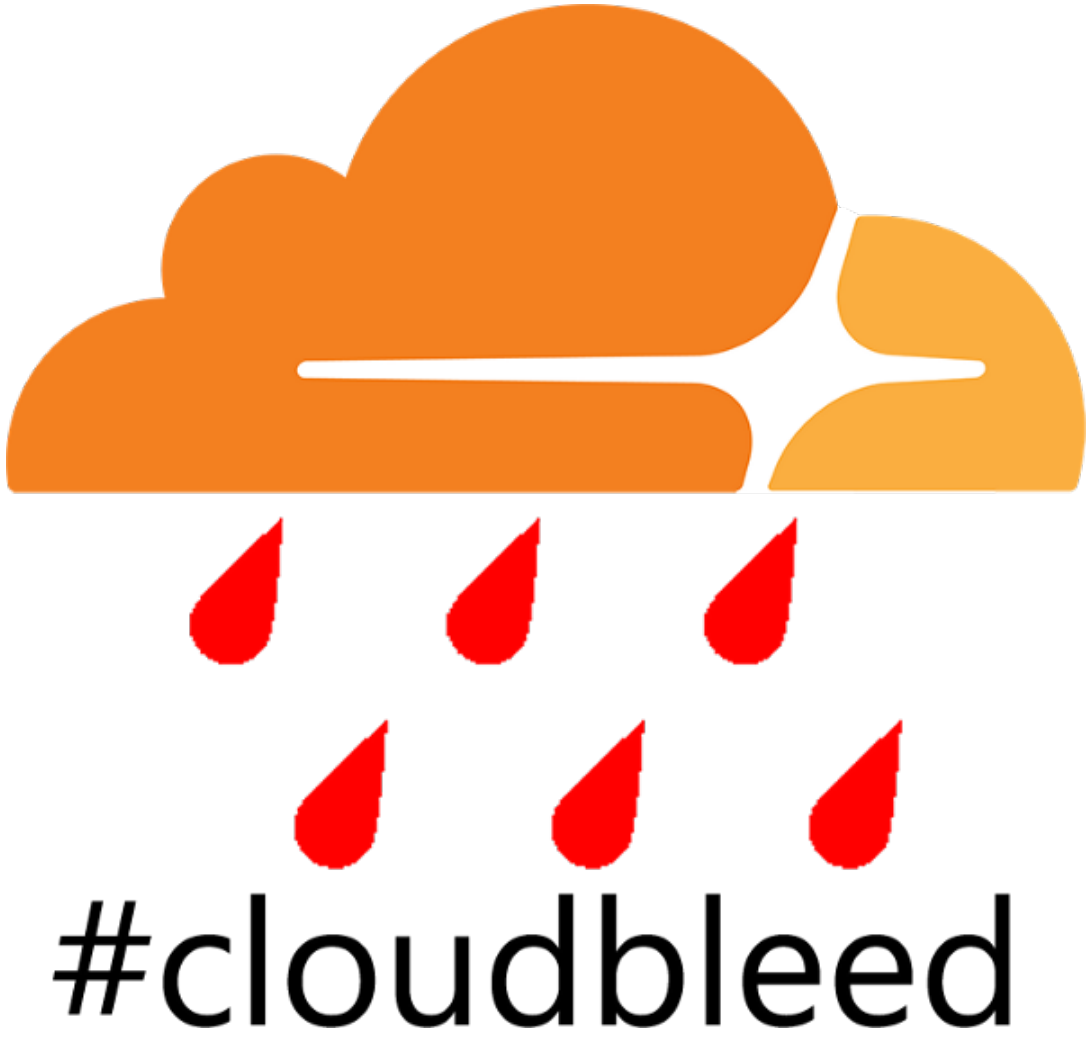
CS 5431 March 3, 2017

Since last time…

SHA-1 Collision



CloudBleed



CloudPets

On to Key Management…

Key Management

- key generation
- key exchange
- key storage
- key use
- key replacement

What sort of keys?

Security	Symmetric	FFC (e.g., DSA, DH)	IFC (e.g., RSA)	ECC (e.g., ECDSA)	DS, hash	HMAC, PRG	NIST Rec.
≤ 80	2TDEA	(1024,160)	k = 1024	f = 160-223	SHA-1		No
112	3TDEA	(2048, 224)	k = 2048	f = 224-255	SHA-224		until 2030
128	AES-128	(3072, 256)	k = 3072	f = 256-383	SHA-256	SHA-1	Yes
≥ 256	AES-256	(15360, 512)	k = 15360	f = 512+	SHA-512	SHA-256	Yes

Failure cases

Generating secure keys

- /dev/random
- /dev/urandom

Failure cases

```
int getRandomNumber()
{
    return 4;  // chosen by fair dice roll.
               // guaranteed to be random.
}
```

debian

Key Storage

- cryptographic module
- remote storage

Secret Sharing

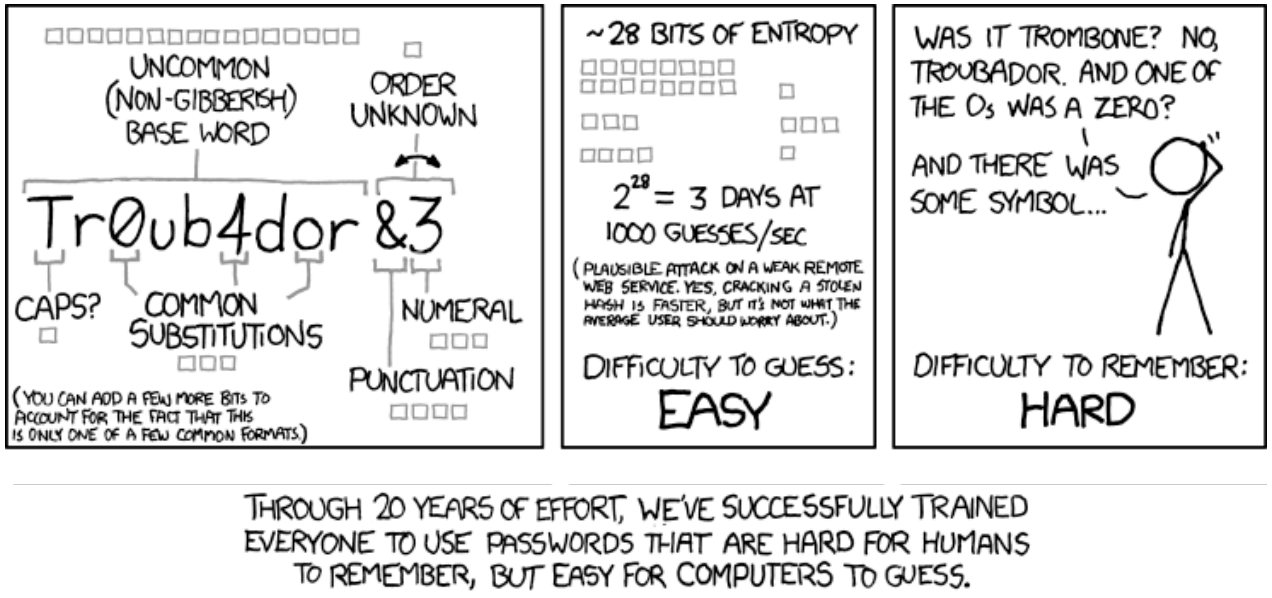
Key use

Don’t reuse keys

Key Replacement

Key Type	Cryptoperiod	
	Originator (OUP)	Recipient
Private Signature Key	1 to 3 years	-
Public Signature-Verification Key	Several years (depends on key size)	
Symmetric Authentication Key	$\leq$ 2 years	$\leq$ OUP + 3 years
Private Authentication Key	1 to 2 years	
Public Authentication Key	1 to 2 years	
Symmetric Data Encryption Keys	$\leq$ 2 years	$\leq$ OUP + 3 years
Symmetric Key Wrapping Key	$\leq$ 2 years	$\leq$ OUP + 3 years
Symmetric Master Key	About 1 year	
Private Key Transport Key	$\leq$ 2 years	
Public Key Transport Key	1 to 2 years	
Symmetric Authorization Key	$\leq$ 2 years	
Private Authorization Key	$\leq$ 2 years	
Public Authorization Key	$\leq$ 2 years	

Password-Based Encryption

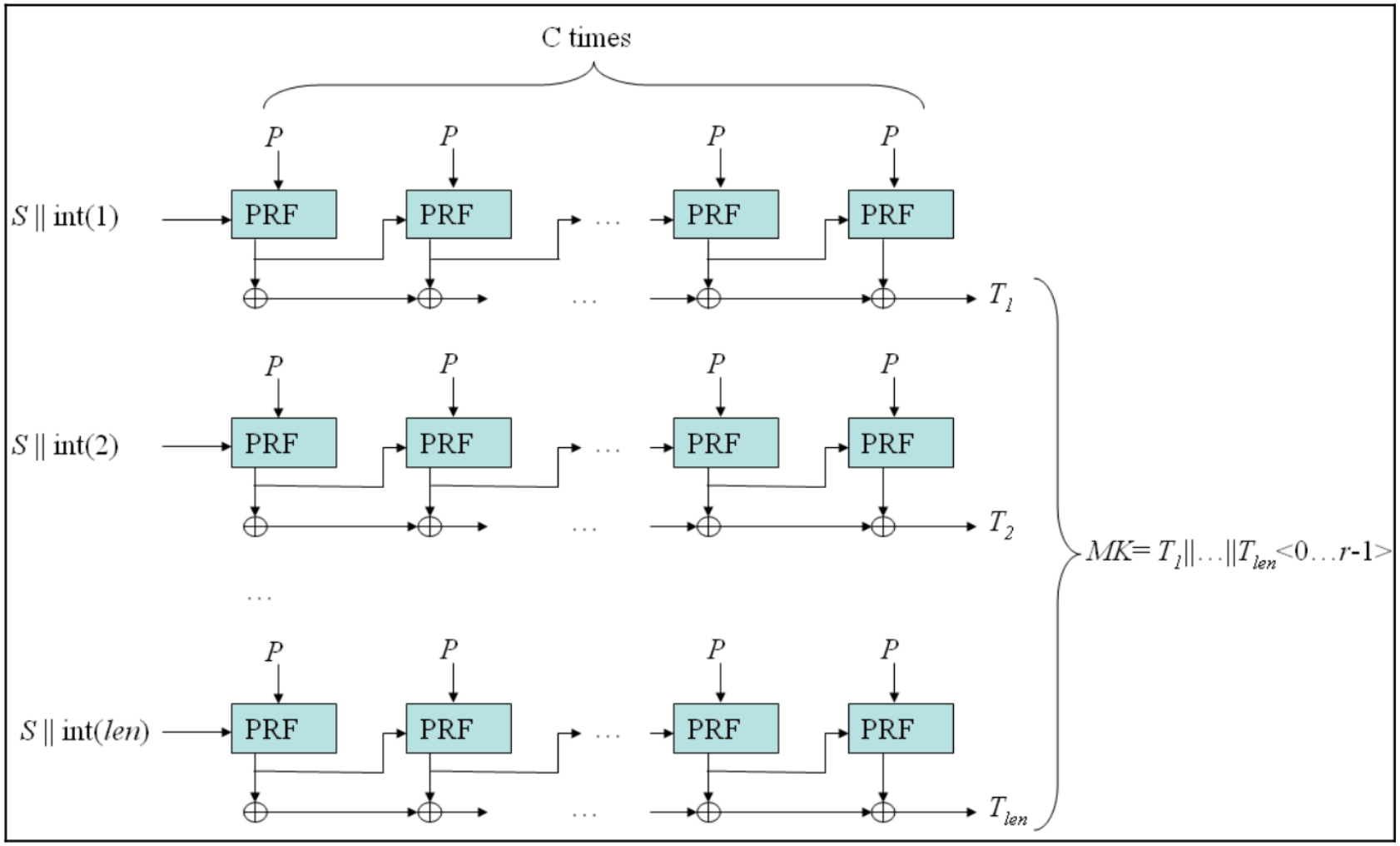


THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

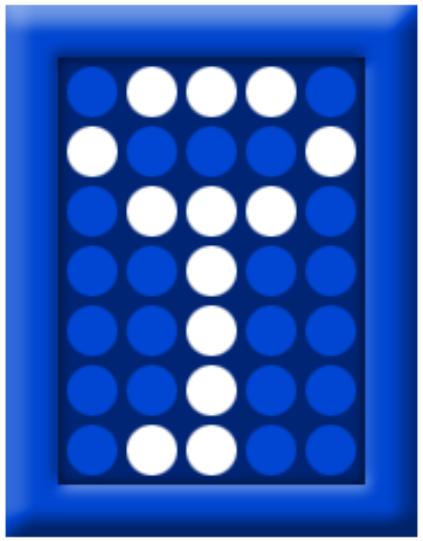
Fixing Password-Based Encryption

- slow hash functions
- salt

PBKDF2



PBE in practice



wuala

tresorit

ID-based encryption

- use ID as public key
- key generation authority generates secret keys for each ID
- Generalization: attribute-based encryption