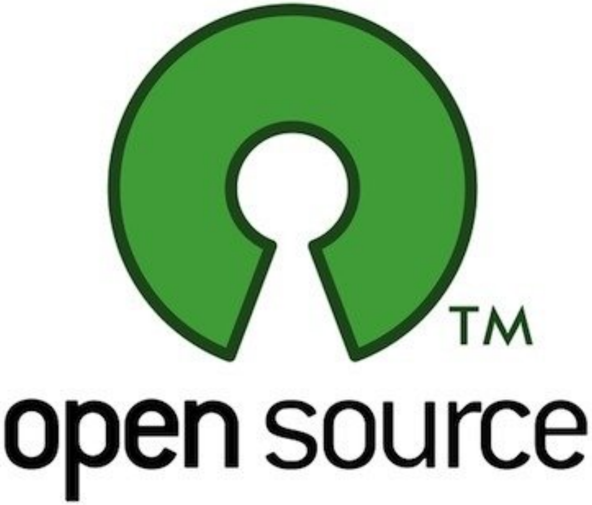
# GOOG13

# JAVA CRYPTOGRAPHY DEEP DIVE:

## TAMING THE BEAST

@ABSTRACTJ


TM
open source

redhat®

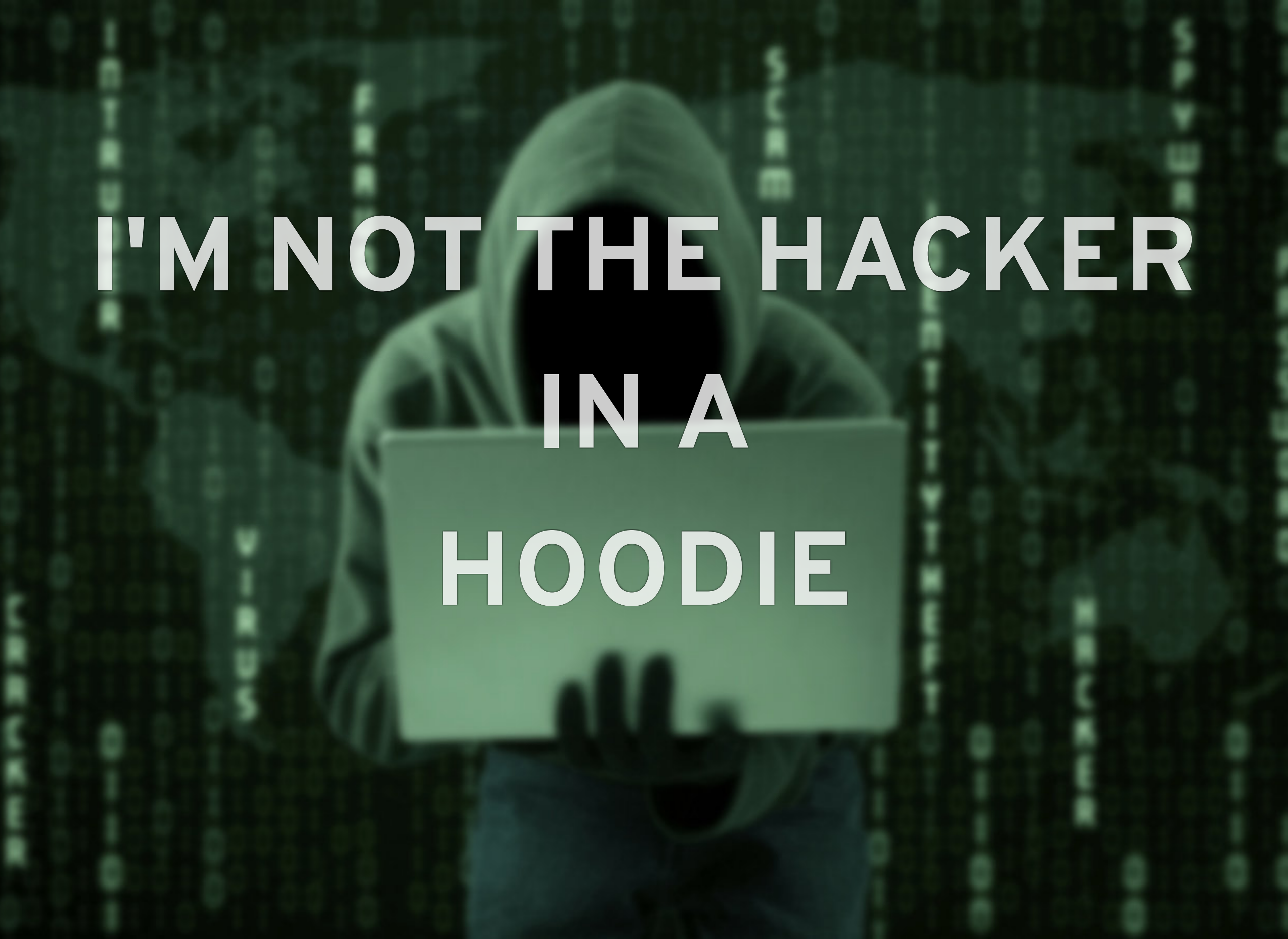
JBoss®
by Red Hat

AeroGear

# SECURITY

# &

# CRYPTOGRAPHY

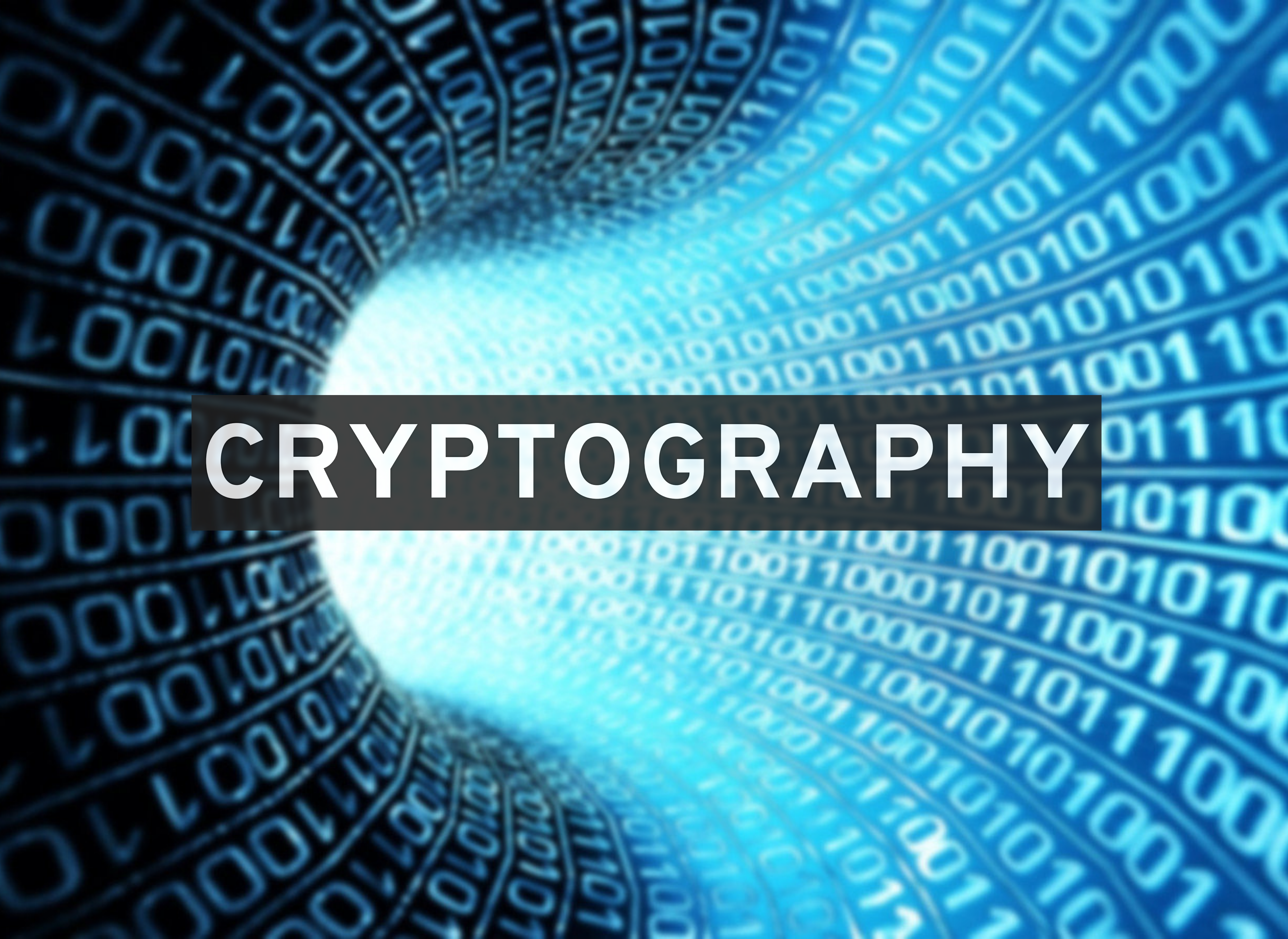


I'M NOT THE HACKER IN A HOODIE

# I'M ONLY VERY, VERY CURIOUS

# DISCLAIMER

# SECURITY IS ABOUT FEELING VS REALITY



CRYPTOGRAPHY

> The study of codes, or the art of writing and solving them.

Oxford Dictionaries

WHAAAAT ?!
memegenerator.net

> Cryptography is the art and science of encryption

Cryptography Engineering

WHAAAAT ?!
memegenerator.net

# HISTORICALLY FOCUSED ON SECRET COMMUNICATIONS

# VIGENÈRE CIPHER

## ~ 1553, Rome

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| k = | j | a | v | a | a | o | e |
| m = | m | o | r | n | i | n | g |
| c = | v | o | m | n | w | a | k |

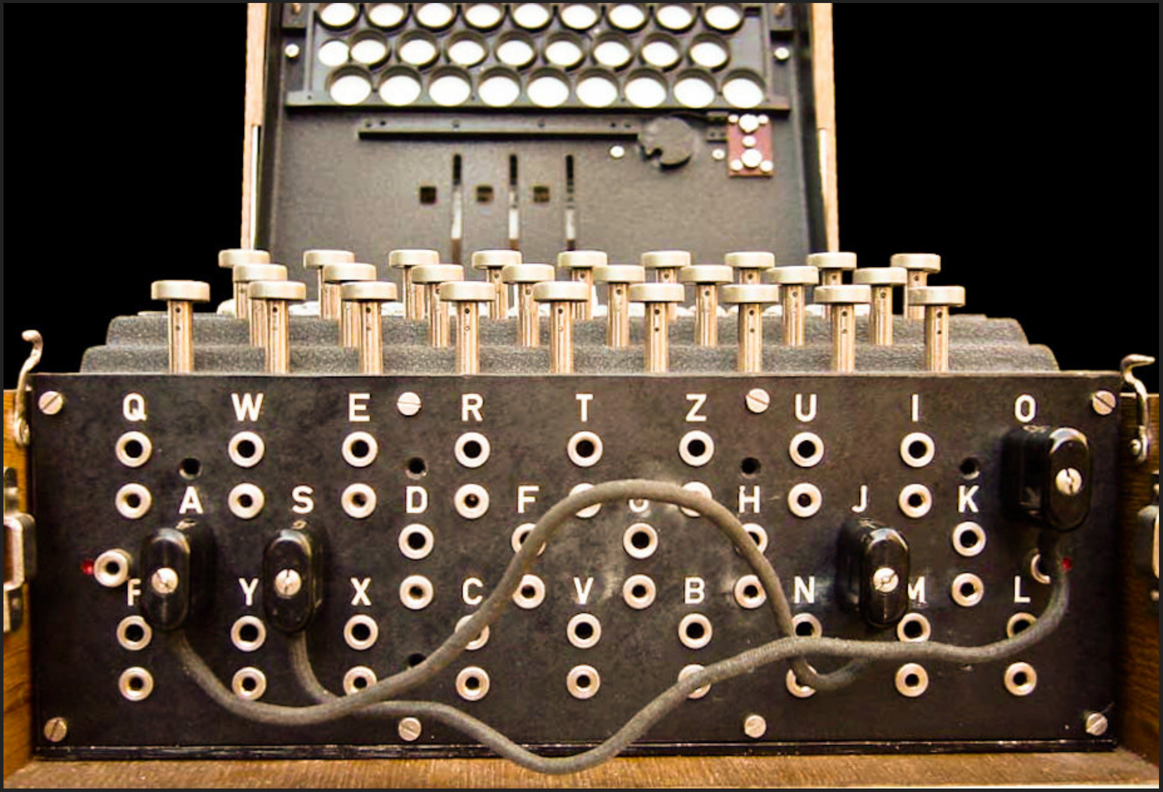
mod 26

# KERCKHOFF'S PRINCIPLE

"A cryptosystem should be secure even if everything about the system, except the key, is public knowledge."

# ENIGMA

(1920)



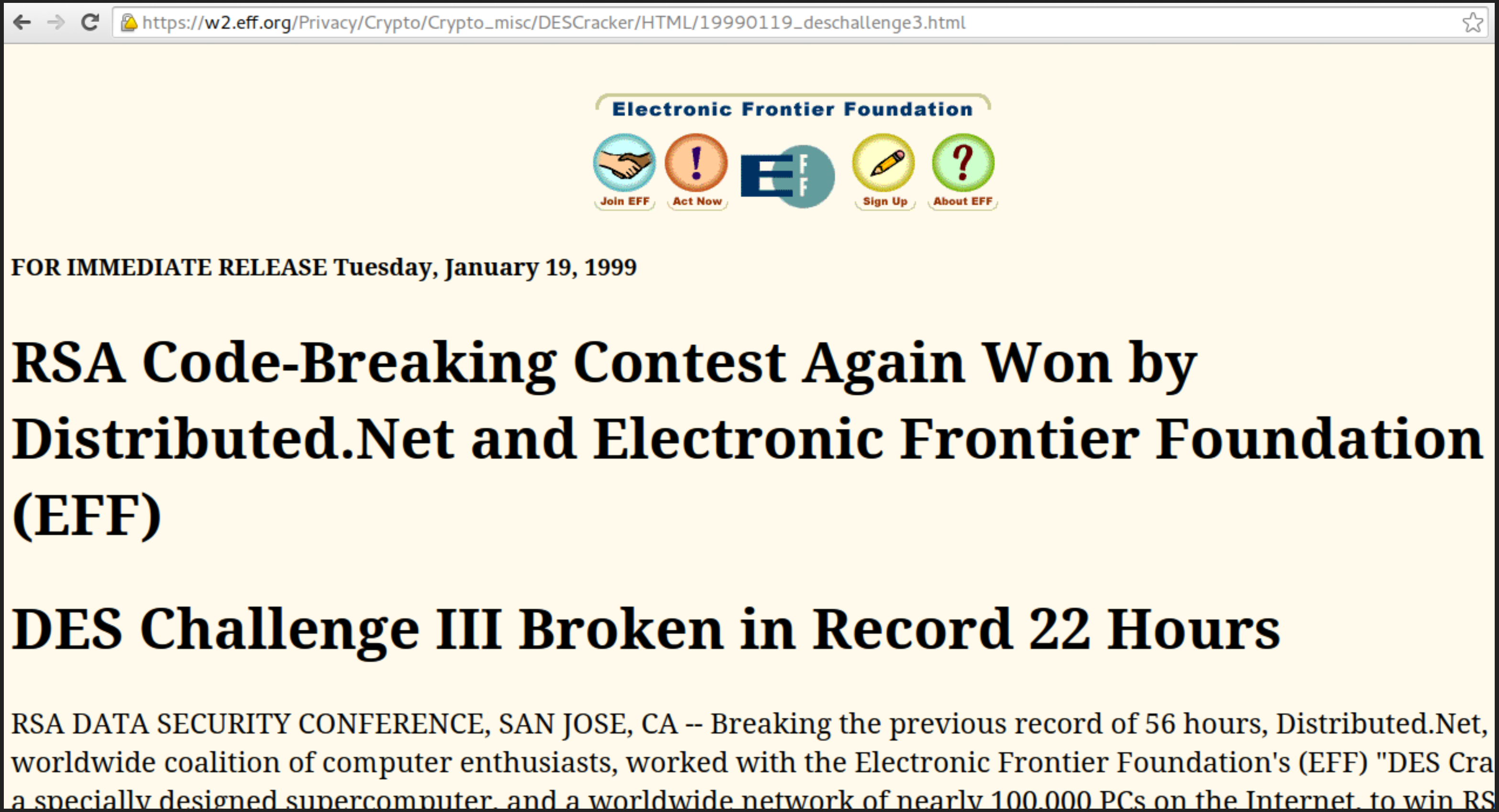
# DES

(1974)

- Key size $2^{56}$, block size 64 bits
- Short key sizes can be subject of brute force
- Should be avoided when possible
- Broken in 22 hours
  - See: https://goo.gl/KgluCi

# DES

(1974)



https://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19990119_deschallenge3.html
Electronic Frontier Foundation
Join EFF
Act Now
EFF
Sign Up
About EFF

**FOR IMMEDIATE RELEASE Tuesday, January 19, 1999**

## RSA Code-Breaking Contest Again Won by Distributed.Net and Electronic Frontier Foundation (EFF)

## DES Challenge III Broken in Record 22 Hours

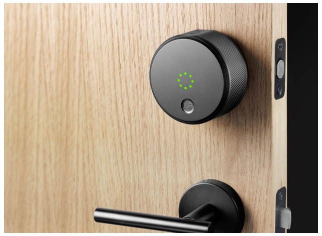
RSA DATA SECURITY CONFERENCE, SAN JOSE, CA -- Breaking the previous record of 56 hours, Distributed.Net, worldwide coalition of computer enthusiasts, worked with the Electronic Frontier Foundation's (EFF) "DES Cra
a specially designed supercomputer, and a worldwide network of nearly 100,000 PCs on the Internet, to win RS

# TODAY

# HARDWARE IS NO LONGER A PROBLEM

# DAILY BASIS

Island
Ridge ending
Short Break



JUSTIÇA ELEITORAL
www.tse.jus.br

4920

Enter Passcode
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
0

HTTPS Everywhere

# BROADER SCOPE

DATA INTEGRITY

SECRECY

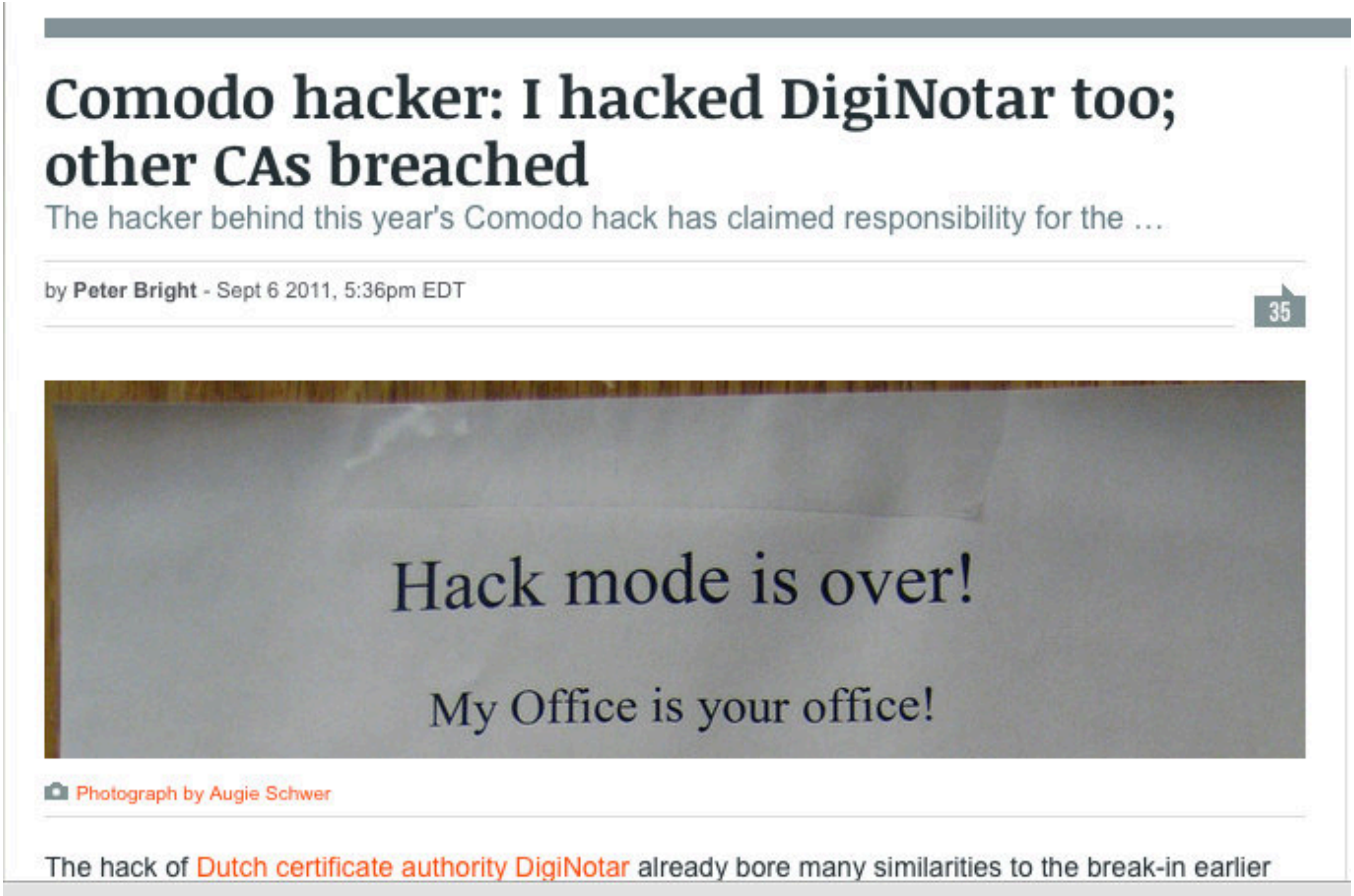
SEVERAL PROTOCOLS

AUTHENTICITY

IT WAS
SUPPOSED TO BE
SIMPLE

# BUT MOST PART OF THE TIME IS LIKE

IT'S REALLY HARD TO GET IT RIGHT



# Comodo hacker: I hacked DigiNotar too; other CAs breached

The hacker behind this year's Comodo hack has claimed responsibility for the …

by **Peter Bright** - Sept 6 2011, 5:36pm EDT

35

Photograph by Augie Schwer

The hack of Dutch certificate authority DigiNotar already bore many similarities to the break-in earlier

Forbes / Security

OCT 20, 2015 @ 12:30 PM 5,923 VIEWS

# 'No Excuses' As Western Digital Leaves Gaping Crypto Flaws In Hard Drives

**Thomas Fox-Brewster,** FORBES STAFF

*I cover crime, privacy and security in digital and physical forms.*

FOLLOW ON FORBES (154)

FULL BIO

Some serious cryptographers have bloodied foreheads today. They've been facepalming rather vociferously

September 21, 2015

# Leaked D-Link security key allows hackers to disguise malware as legit

Share this article:

*A leak of a major technology company's security key has been discovered, allowing hackers to convince Windows that their malware is legit.*

A company has accidentally released a key that allows hackers to issue malware, disguised as legitimate software.

In February, D-Link, a Taiwanese networking equipment company, published one of its private keys, allowing its software to be recognised as legitimate.

The leaked key has been published since february of this year

bartvb, a user of Tweakers, a Dutch news outlet, discovered the leak late last week before reporting it. The key was discovered when it appeared in one of D-link's open-source firmware downloads for its DCS-5020L surveillance camera.

ASHLEY MADISON BREACH

# Ashley Madison hackers publish compromised records

ASHLEY MADISON®
Life is short. Have an affair.®
Get started by telling us your relationship status:
Please Select
See Your Matches »
anonymous
100% SECRET

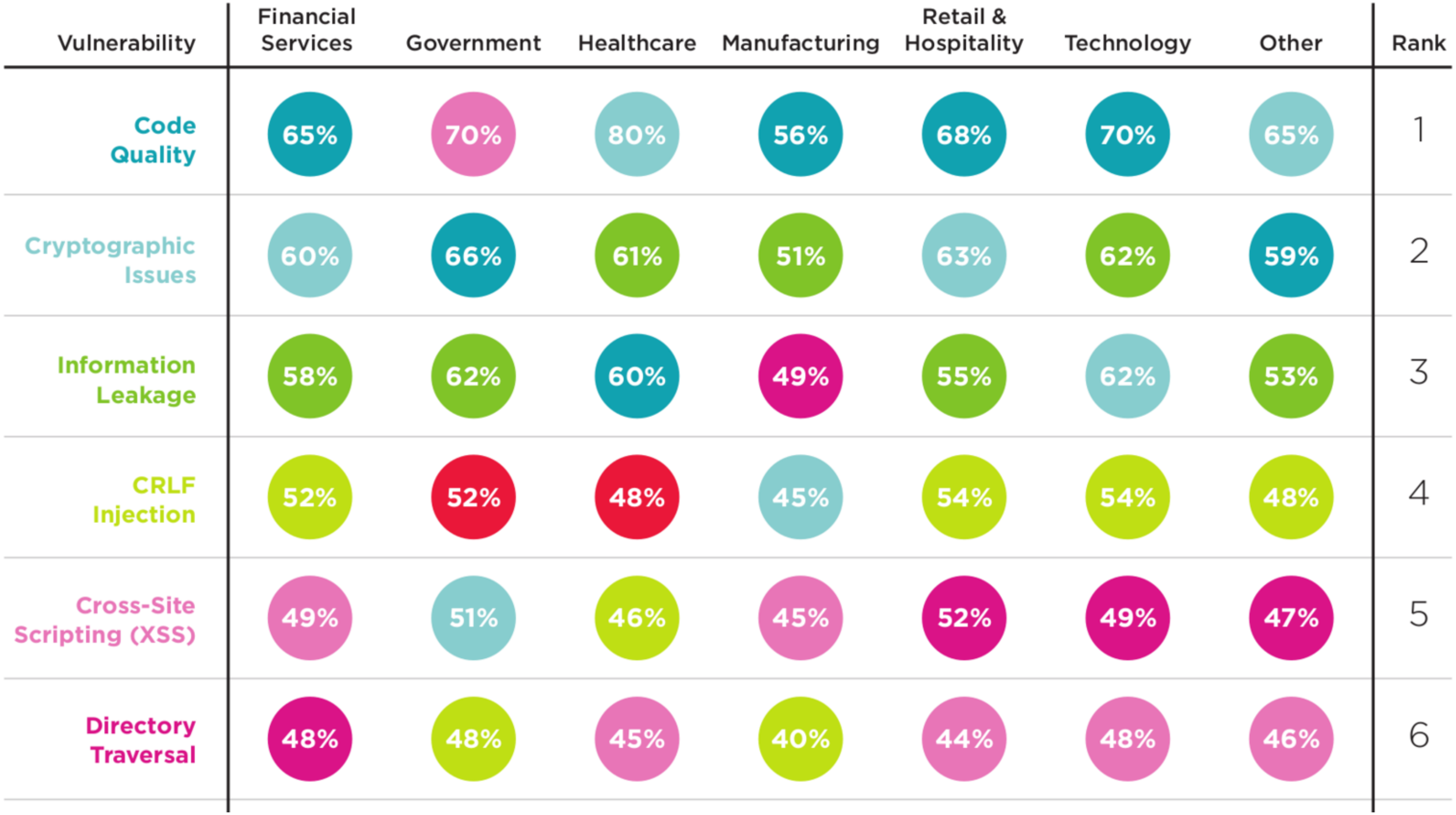
# HEARTBLEED

# SURVEILLANCE

"Security is the Jar Jar Binks of software development.

Martin Boßlet

# PROBLEM OR SOLUTION?

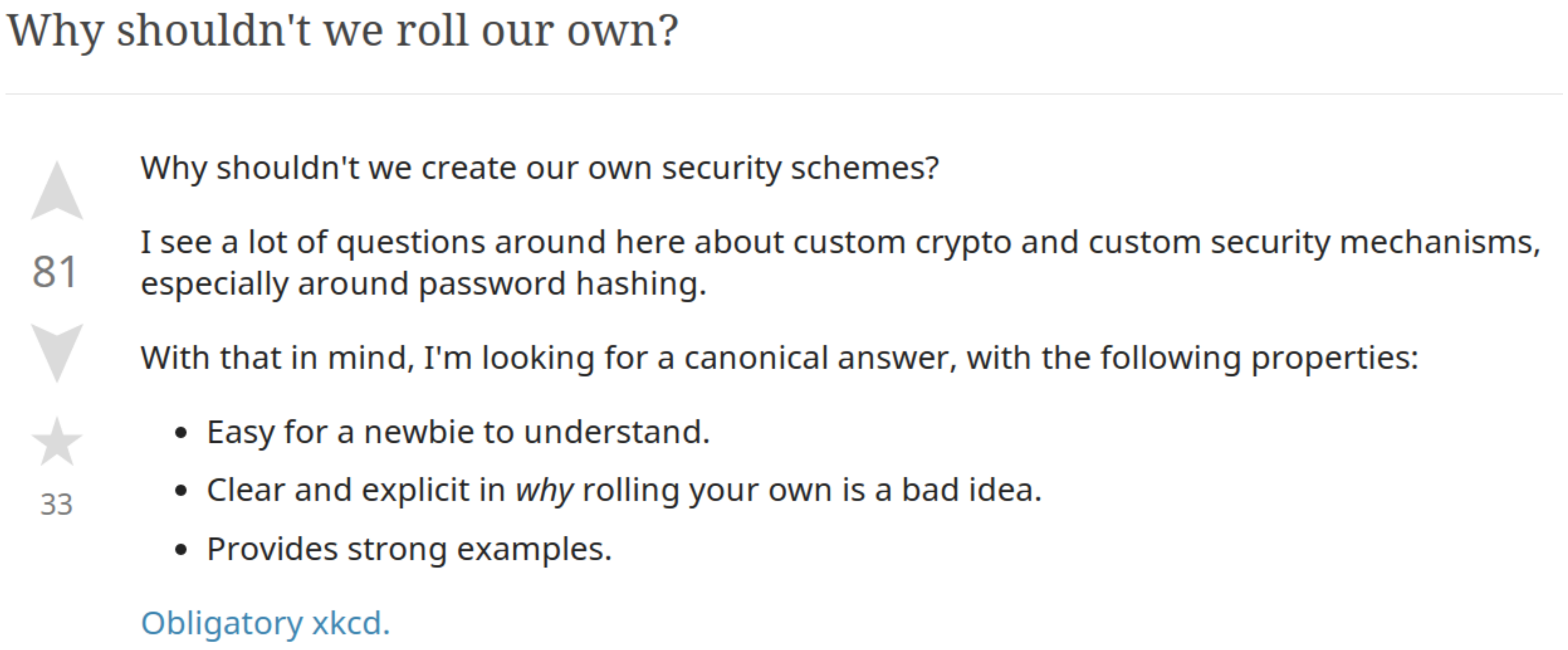


| Vulnerability | Financial Services | Government | Healthcare | Manufacturing | Retail & Hospitality | Technology | Other | Rank |
|---|---|---|---|---|---|---|---|---|
| Code Quality | 65% | 70% | 80% | 56% | 68% | 70% | 65% | 1 |
| Cryptographic Issues | 60% | 66% | 61% | 51% | 63% | 62% | 59% | 2 |
| Information Leakage | 58% | 62% | 60% | 49% | 55% | 62% | 53% | 3 |
| CRLF Injection | 52% | 52% | 48% | 45% | 54% | 54% | 48% | 4 |
| Cross-Site Scripting (XSS) | 49% | 51% | 46% | 45% | 52% | 49% | 47% | 5 |
| Directory Traversal | 48% | 48% | 45% | 40% | 44% | 48% | 46% | 6 |

Source: Veracode

BOOKS

ARE AN AMAZING
SOURCE TO LEARN



## Why shouldn't we roll our own?

Why shouldn't we create our own security schemes?

I see a lot of questions around here about custom crypto and custom security mechanisms, especially around password hashing.

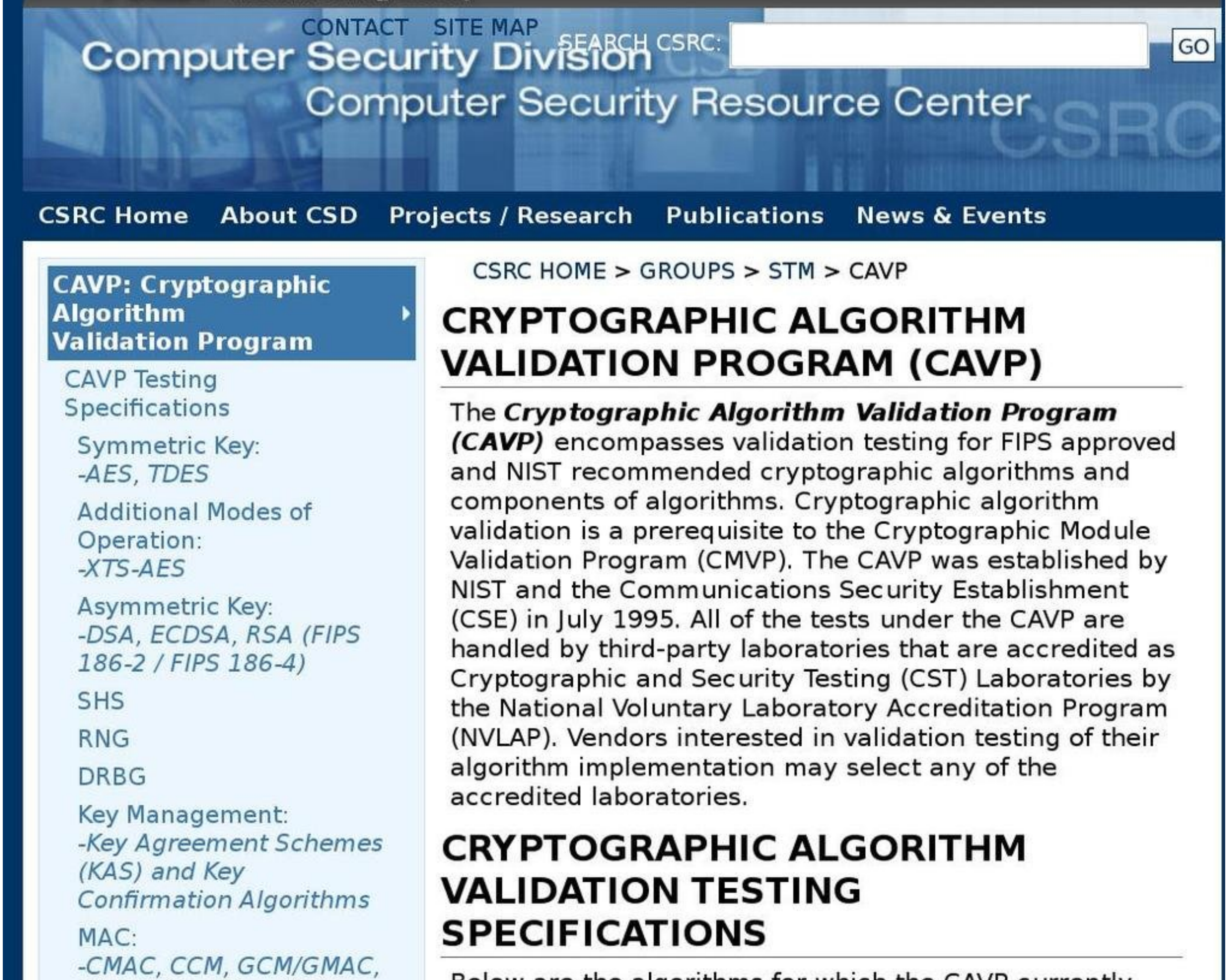
With that in mind, I'm looking for a canonical answer, with the following properties:

- Easy for a newbie to understand.
- Clear and explicit in *why* rolling your own is a bad idea.
- Provides strong examples.

Obligatory xkcd.

**Source: Stackoverflow**

# DON'T ROLL YOUR OWN CRYPTO


CONTACT
SITE MAP
Computer Security Division
SEARCH CSRC:
GO
Computer Security Resource Center
CSRC
CSRC Home
About CSD
Projects / Research
Publications
News & Events
CAVP: Cryptographic Algorithm Validation Program
CAVP Testing Specifications
Symmetric Key:
-AES, TDES
Additional Modes of Operation:
-XTS-AES
Asymmetric Key:
-DSA, ECDSA, RSA (FIPS 186-2 / FIPS 186-4)
SHS
RNG
DRBG
Key Management:
-Key Agreement Schemes (KAS) and Key Confirmation Algorithms
MAC:
-CMAC, CCM, GCM/GMAC,
CSRC HOME > GROUPS > STM > CAVP

# CRYPTOGRAPHIC ALGORITHM VALIDATION PROGRAM (CAVP)

The ***Cryptographic Algorithm Validation Program (CAVP)*** encompasses validation testing for FIPS approved and NIST recommended cryptographic algorithms and components of algorithms. Cryptographic algorithm validation is a prerequisite to the Cryptographic Module Validation Program (CMVP). The CAVP was established by NIST and the Communications Security Establishment (CSE) in July 1995. All of the tests under the CAVP are handled by third-party laboratories that are accredited as Cryptographic and Security Testing (CST) Laboratories by the National Voluntary Laboratory Accreditation Program (NVLAP). Vendors interested in validation testing of their algorithm implementation may select any of the accredited laboratories.

# CRYPTOGRAPHIC ALGORITHM VALIDATION TESTING SPECIFICATIONS

Below are the algorithms for which the CAVP currently

# HOW CRYPTO IS DONE NOWADAYS?

# FULL CONTROL

GOOD

BAD

# LET'S GET OUR HANDS DIRTY

# THE BADLY DESIGNED NOTE APP

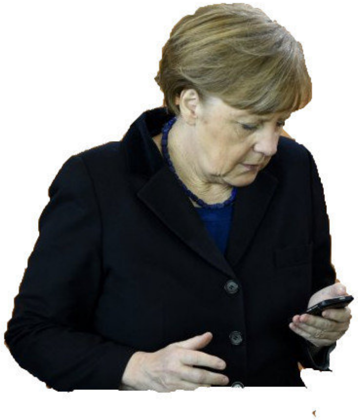
# TOOLS

Java 8

org.json

BouncyCastle

Hmmm, I wish I had a journal app to share my notes



ALICE

Yes, we can!

BOB

Must be nice!

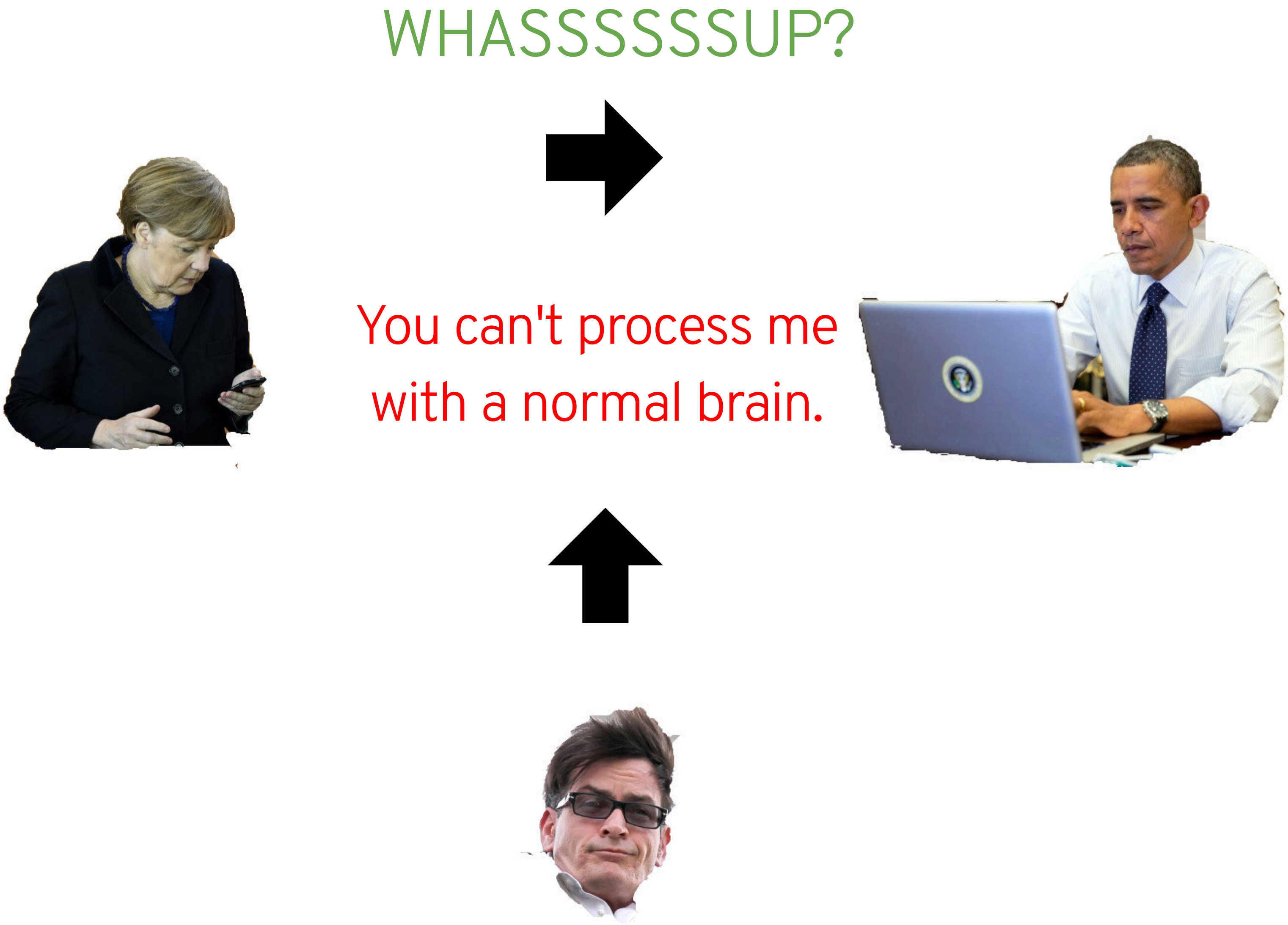
EVE

# #1 STORY

AS A USER OF THIS APP, ALICE WANTS TO BE ABLE TO CREATE NEW ENTRIES

WHASSSSSSUP?

WHASSSSSSUP?



WHASSSSSSUP?

You can't process me with a normal brain.

# #2 STORY

AS A USER OF THIS APP, BOB WANTS TO BE ABLE TO VERIFY THE INTEGRITY OF ALICE'S FILES

# CWE-327

## USE OF A BROKEN OR RISKY CRYPTOGRAPHIC ALGORITHM

# MD5

## ARE THEY DIFFERENT?

# SHA-1

https://malicioussha1.github.io/

# BUT WHAT ABOUT INCLUDING A SALT?

# CWE-916

# PASSWORD HASH WITH INSUFFICIENT COMPUTATIONAL EFFORT

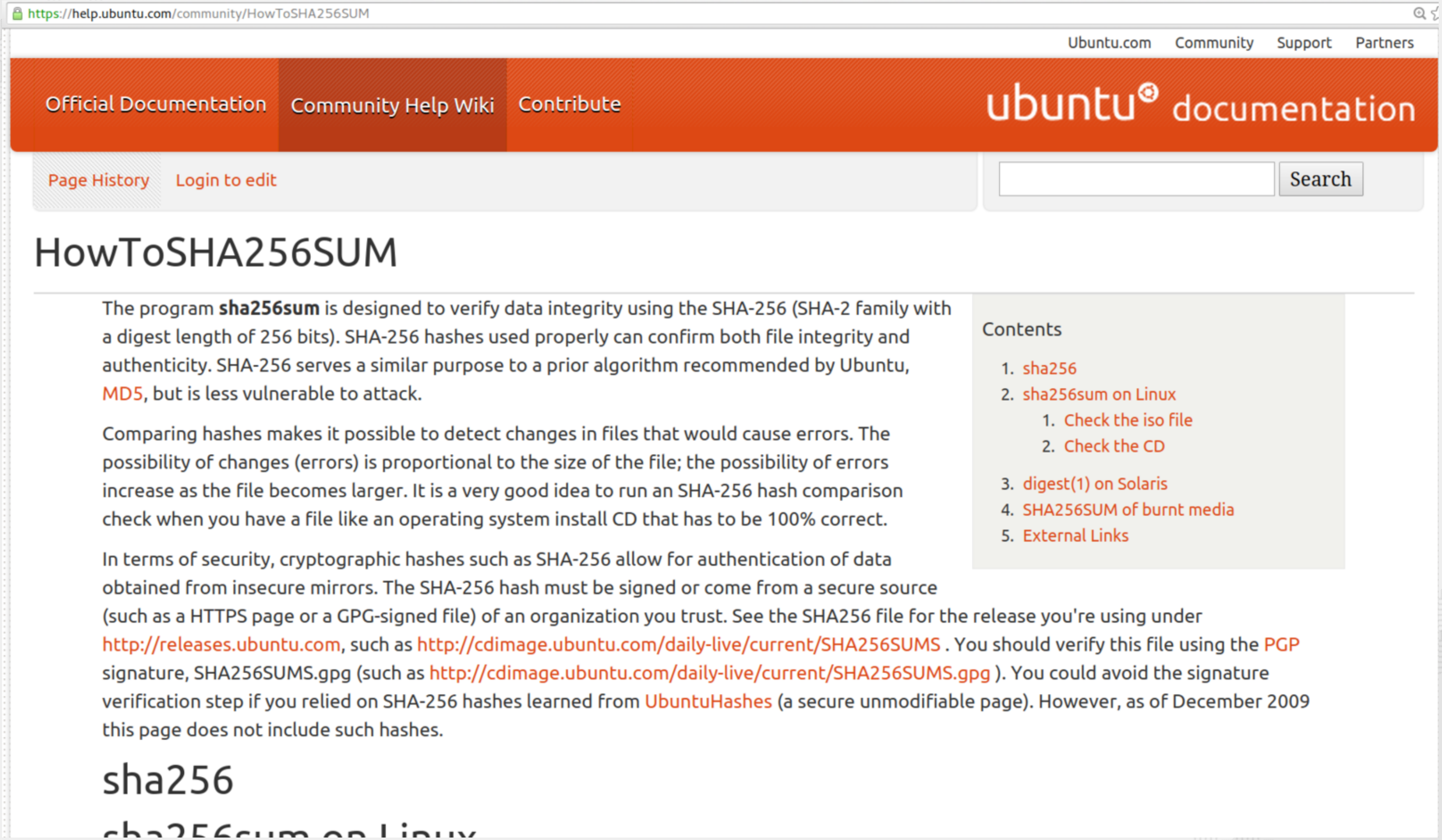
SHA-224

SHA-256

SHA-384

SHA-512

**ARE ALL GOOD CHOICES**



# HowToSHA256SUM

The program **sha256sum** is designed to verify data integrity using the SHA-256 (SHA-2 family with a digest length of 256 bits). SHA-256 hashes used properly can confirm both file integrity and authenticity. SHA-256 serves a similar purpose to a prior algorithm recommended by Ubuntu, MD5, but is less vulnerable to attack.

Comparing hashes makes it possible to detect changes in files that would cause errors. The possibility of changes (errors) is proportional to the size of the file; the possibility of errors increase as the file becomes larger. It is a very good idea to run an SHA-256 hash comparison check when you have a file like an operating system install CD that has to be 100% correct.

In terms of security, cryptographic hashes such as SHA-256 allow for authentication of data obtained from insecure mirrors. The SHA-256 hash must be signed or come from a secure source (such as a HTTPS page or a GPG-signed file) of an organization you trust. See the SHA256 file for the release you're using under http://releases.ubuntu.com, such as http://cdimage.ubuntu.com/daily-live/current/SHA256SUMS . You should verify this file using the PGP signature, SHA256SUMS.gpg (such as http://cdimage.ubuntu.com/daily-live/current/SHA256SUMS.gpg ). You could avoid the signature verification step if you relied on SHA-256 hashes learned from UbuntuHashes (a secure unmodifiable page). However, as of December 2009 this page does not include such hashes.

Contents

1. sha256
2. sha256sum on Linux
    1. Check the iso file
    2. Check the CD
3. digest(1) on Solaris
4. SHA256SUM of burnt media
5. External Links

## sha256

## sha256sum on Linux

# #3 STORY

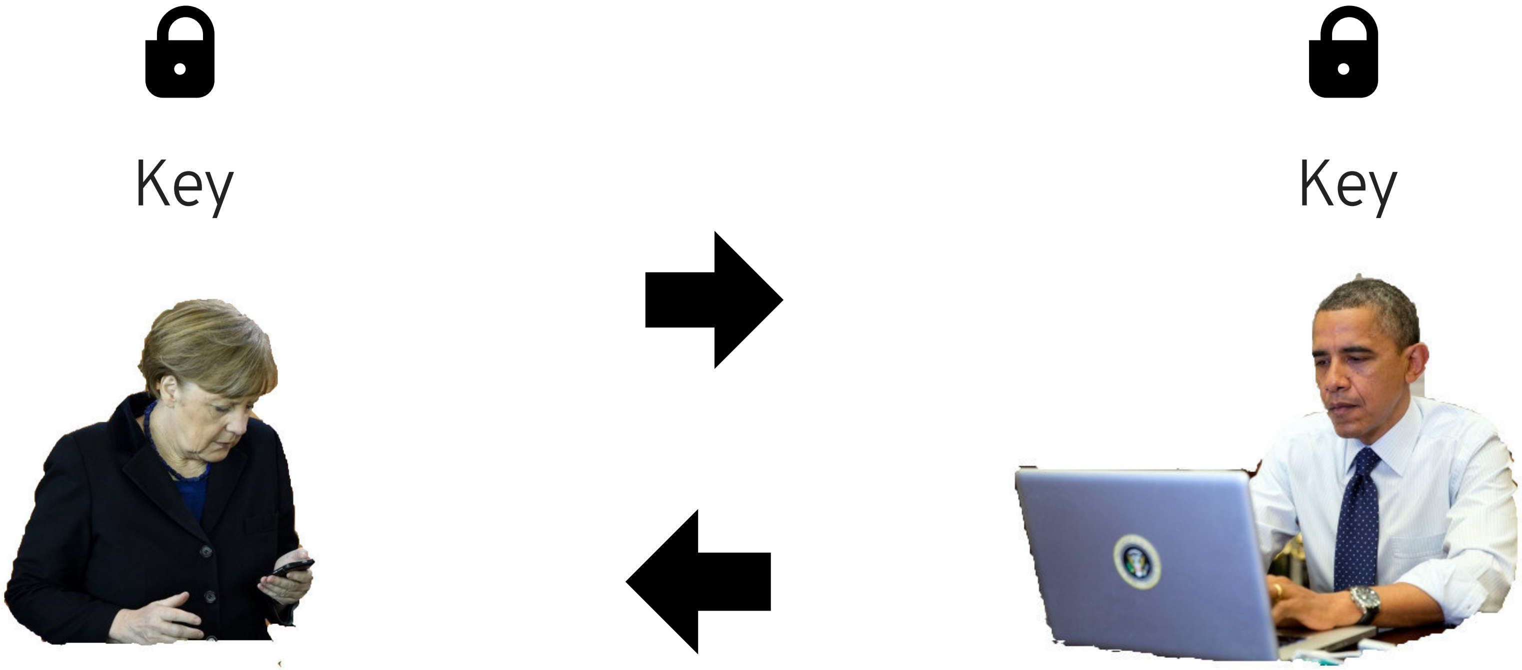
AS A PARANOID, I WOULD LIKE NOT ONLY INTEGRITY, BUT ALSO AUTHENTICITY

# #4 STORY

AS A USER OF THIS APP, I WANT TO ADD INTEGRITY, AUTHENTICITY AND SECRECY AND HIDE MY ENTRIES FROM NSA

# PADDING

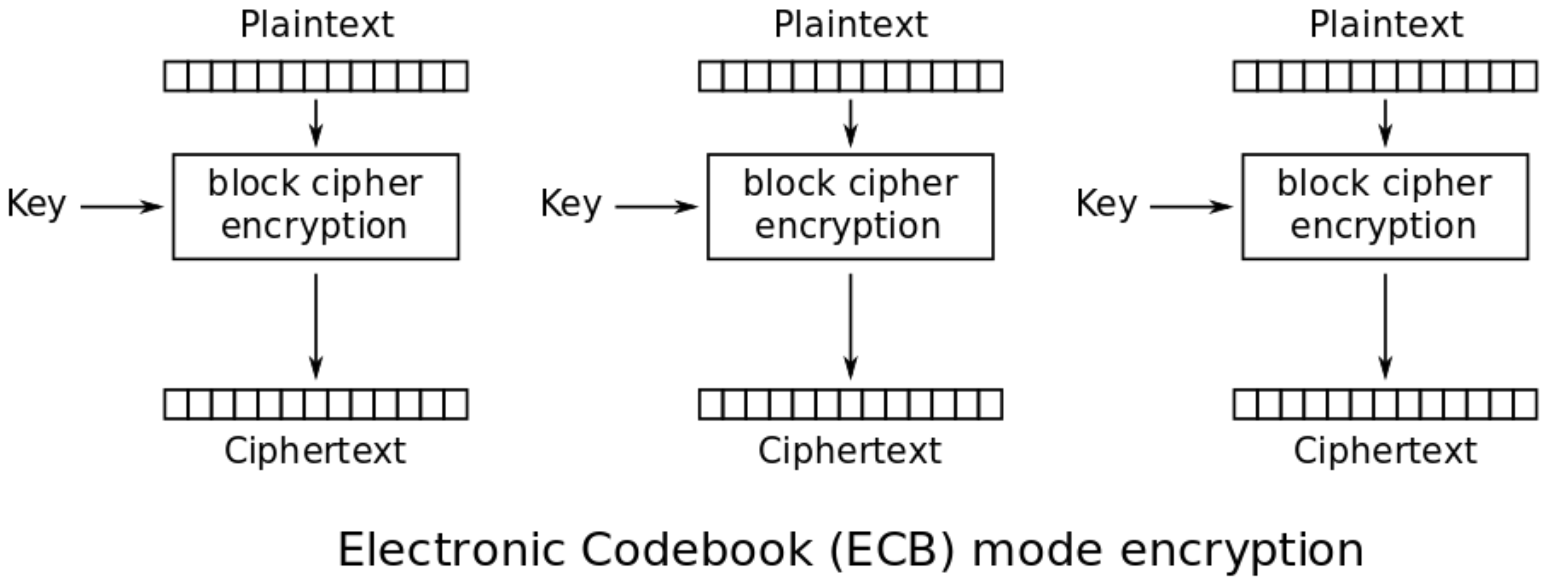
WE USE A "PADDING SCHEME" TO FILL THE LAST BLOCK UNTIL IT MEETS THE CIPHER BLOCK SIZE



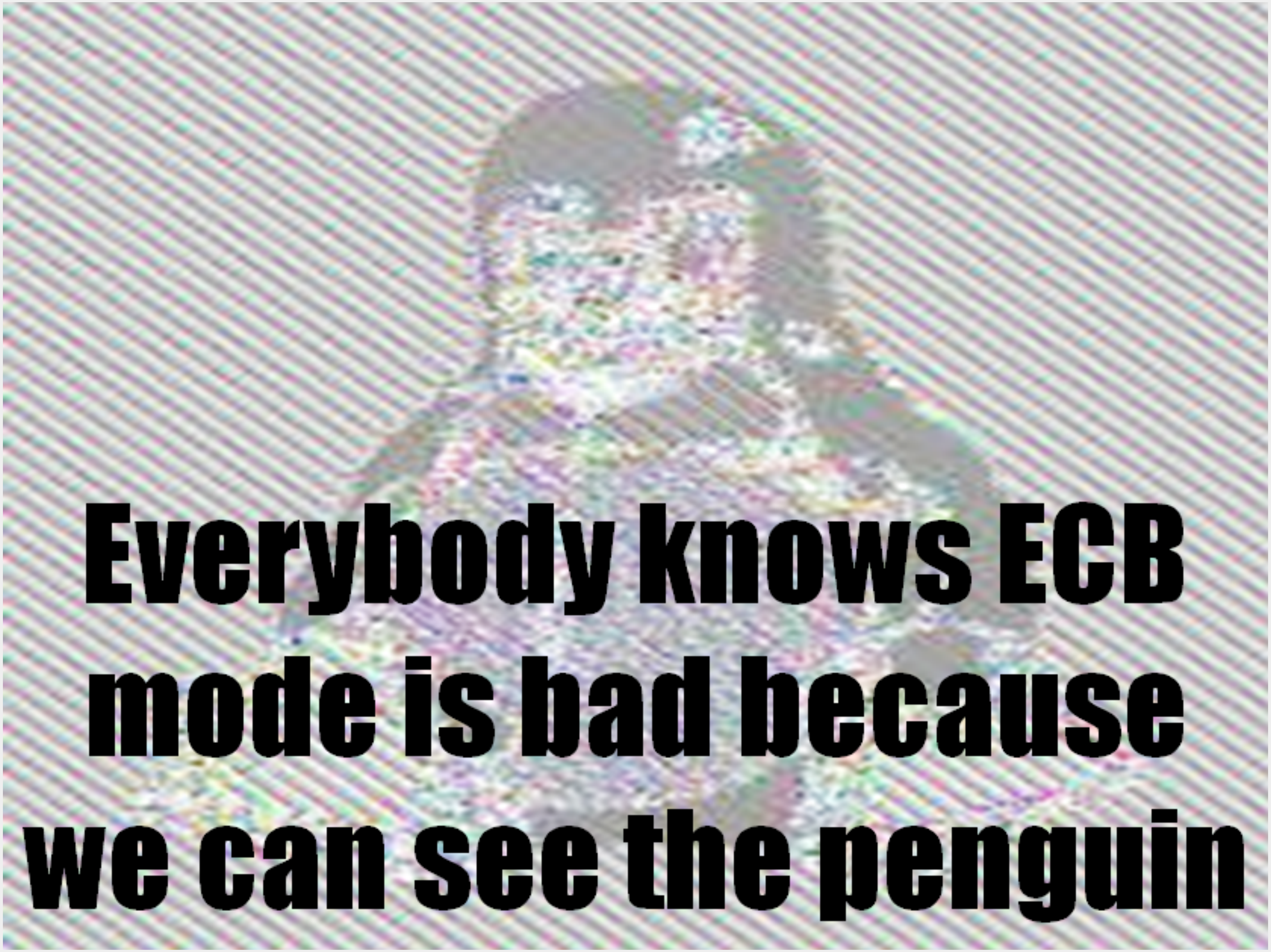
Key

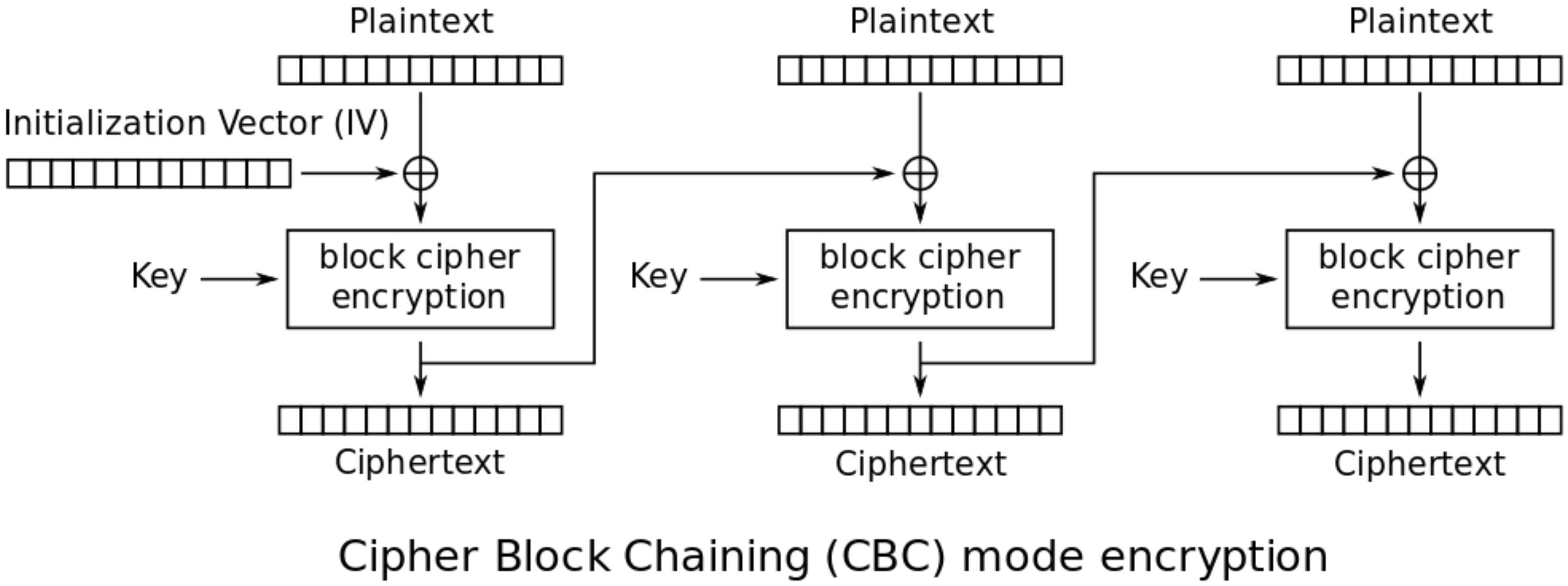
Key

# MODES OF OPERATION

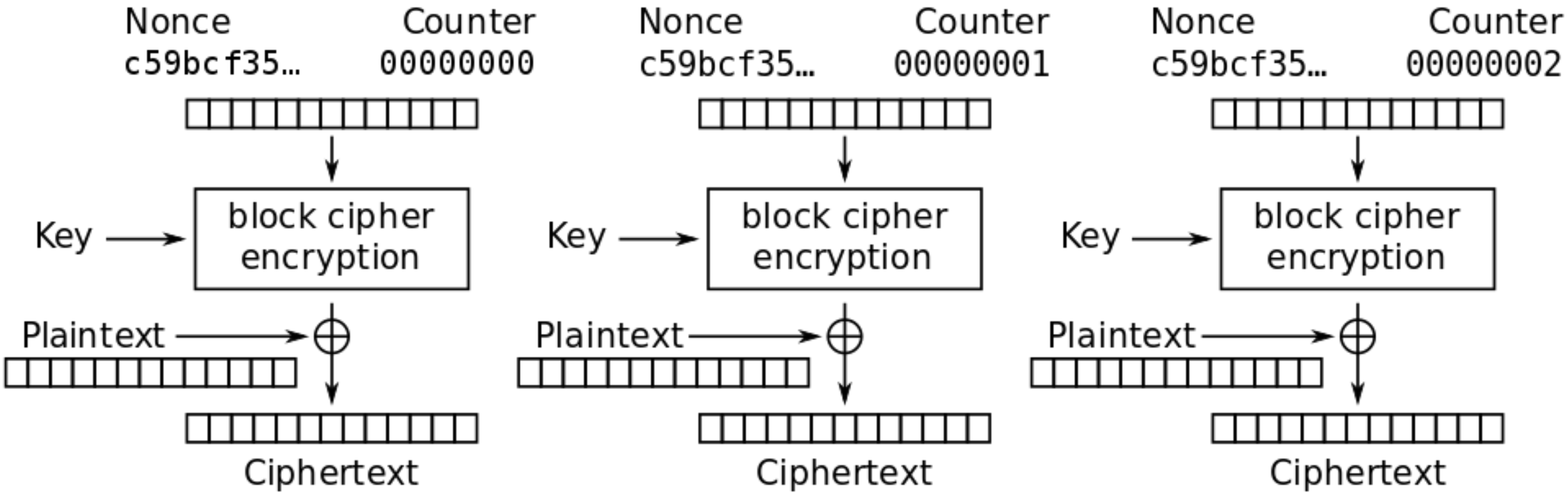


Source: Wikipedia


Everybody knows ECB
mode is bad because
we can see the penguin

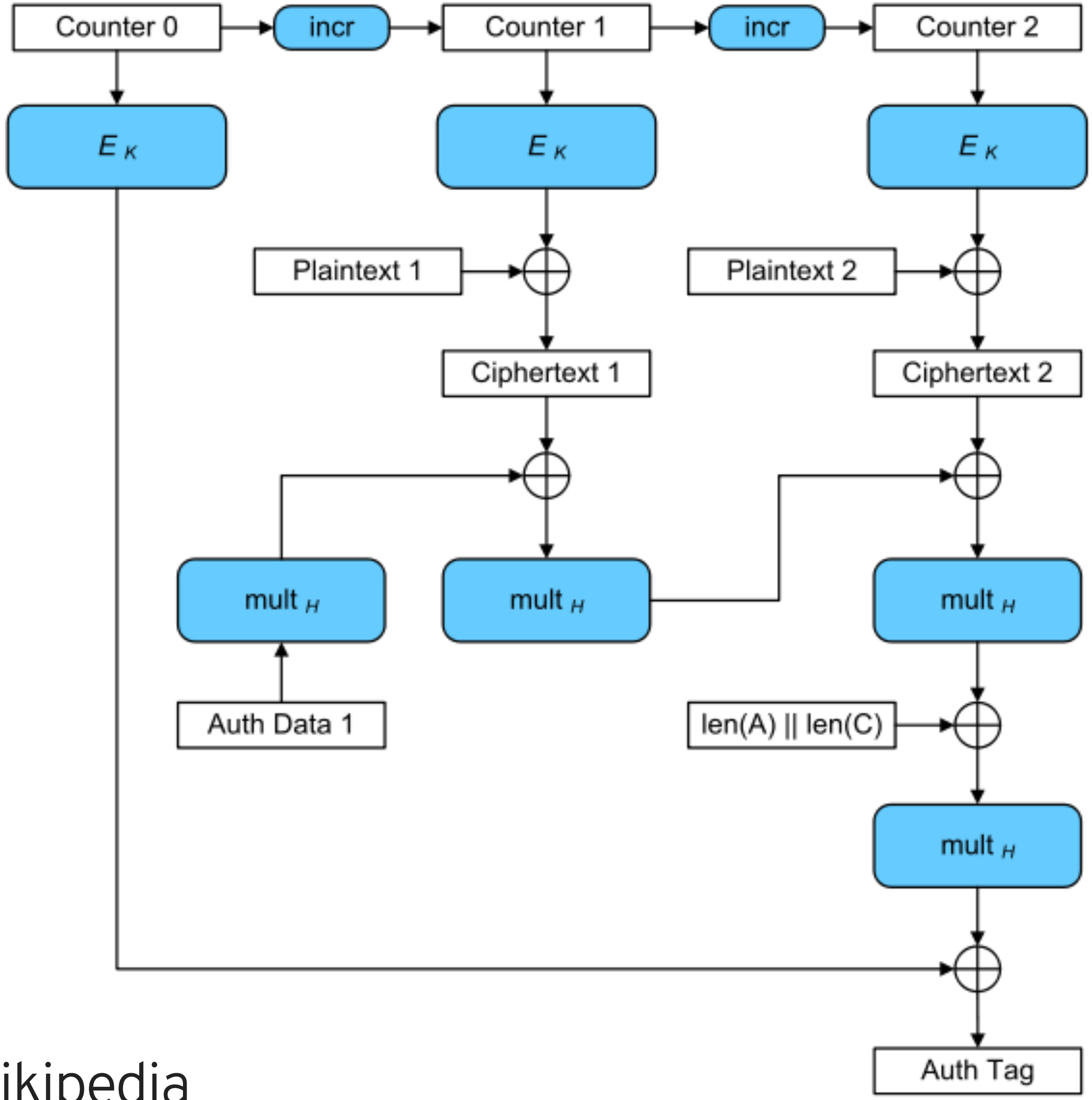


Source: Wikipedia



Counter (CTR) mode encryption

Source: Wikipedia



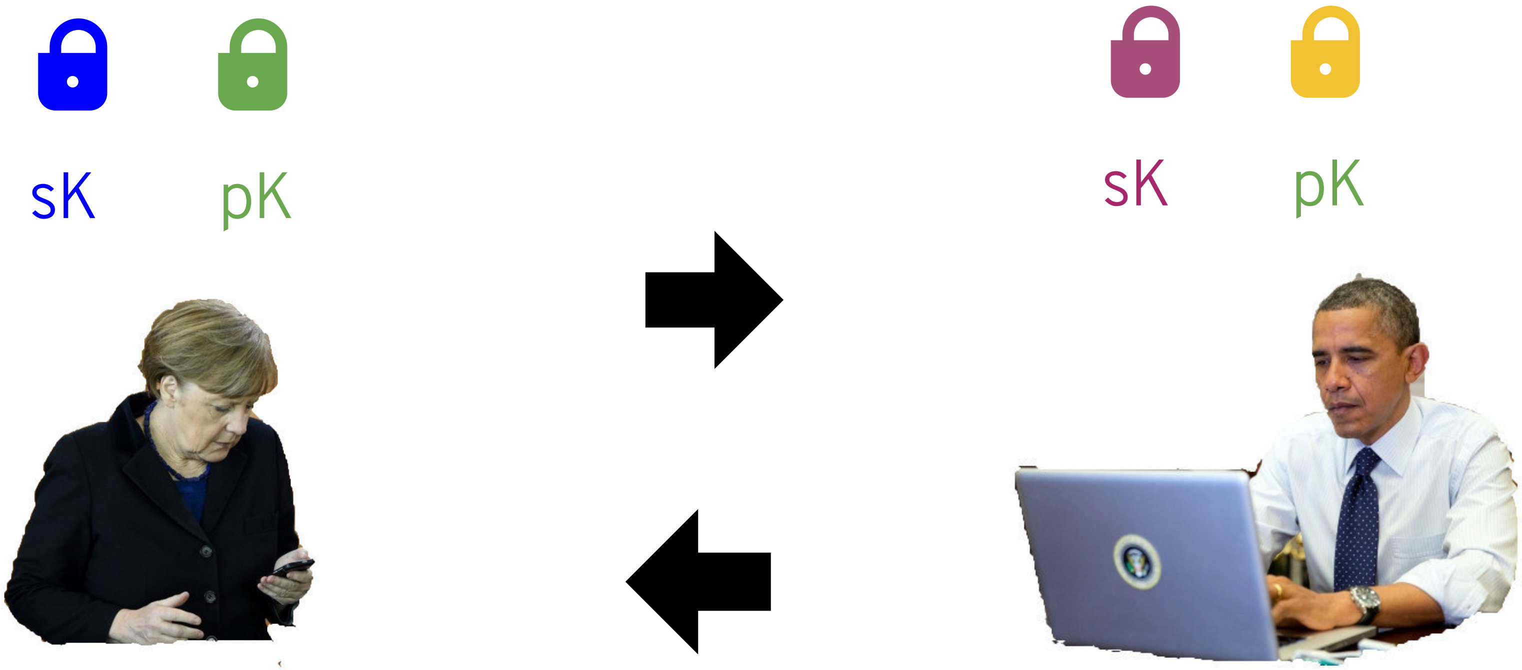
Source: Wikipedia

# #5 STORY

AS A USER I WANT TO HAVE PASSWORD PROTECTED ENTRIES

# #6 STORY

AS SOMEONE VERY SOCIAL, I WANT TO SHARE MY ENTRIES WITH A FRIEND WITHOUT EXPOSING MY KEYS



sK pK

sK pK

redhat®

# THANK YOU!

http://abstractj.org

https://aerogear.org