CSCI E-11

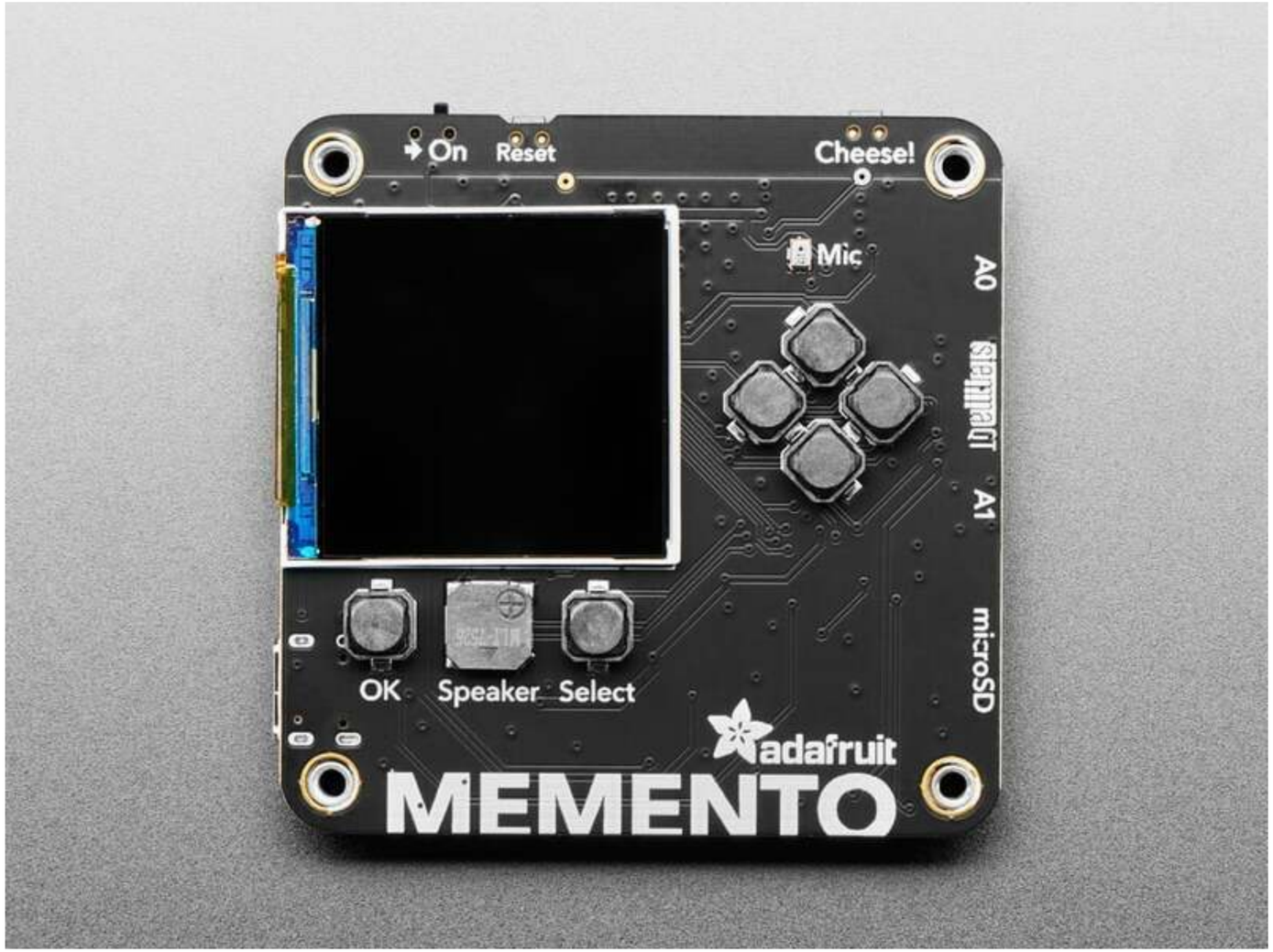
Class 6

March 3, 2026

Encryption and Private Computation

Harvard
Extension School
HARVARD DIVISION OF
CONTINUING EDUCATION

Does anyone not have a Memento yet? (Or on the way?)



A Comparative Long-Term Study of Fallback Authentication

A Comparative Long-Term Study of Fallback Authentication Schemes

Leona Lassak
Ruhr University Bochum
Germany
leona.lassak@rub.de

Philipp Markert
Ruhr University Bochum
Germany
philipp.markert@rub.de

Maximilian Golla
CISPA Helmholtz Center for Information Security
Germany
golla@cispa.de

Elizabeth Stobert
Carleton University
Germany
elizabeth.stobert@carleton.ca

Markus Dürmuth
Leibniz University Hannover
Germany
markus.duermuth@itsec.uni-hannover.de

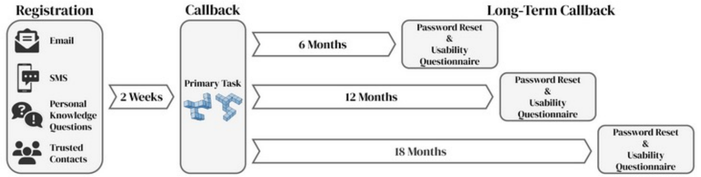


Figure 1: Structure of the conducted long-term study ($n = 97$). Participants were assigned to 1 of 4 fallback schemes (email, SMS, PKQ, trusted contacts) and a recall time (6, 12, 18 months). The 2-week callback served the purpose of minimizing dropout rates.

ABSTRACT

Fallback authentication, the process of re-establishing access to an account when the primary authenticator is unavailable, holds critical significance. Approaches range from secondary channels like email and SMS to personal knowledge questions (PKQs) and social authentication. A key difference to primary authentication is that the duration between enrollment and authentication can be much longer, typically months or years. However, few systems have been studied over extended timeframes, making it difficult to know how well these systems truly help users recover their accounts. We also lack meaningful comparisons of schemes as most prior work examined two mechanisms at most. We report the results of a long-term user study of the usability of fallback authentication over 18 months to provide a fair comparison of the four most commonly used fallback authentication methods. We show that users prefer email and SMS-based methods, while mechanisms based on PKQs and trustees lag regarding successful resets and convenience.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3642889

CCS CONCEPTS

• **Security and privacy → Authentication**; **Usability in security and privacy**.

KEYWORDS

fallback authentication, email, SMS, personal knowledge questions

ACM Reference Format:
Leona Lassak, Philipp Markert, Maximilian Golla, Elizabeth Stobert, and Markus Dürmuth. 2024. A Comparative Long-Term Study of Fallback Authentication Schemes. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3613904.3642889

1 INTRODUCTION

Fallback authentication (also called account recovery, backup, emergency, or recovery authentication) is the mechanism for restoring access to an account if the primary authenticator becomes unavailable. It plays a central role in real-world account management, i.e., a study by Bonneau et al. showed that almost all account owners of the surveyed sample needed to reset their primary authenticator at least once [13]. Administrators have to deal with forgotten passwords and lost security tokens regularly [41, 75, 82], highlighting the need for usable mechanisms. Manual account recovery is usually a last resort as it comes at the highest cost for providers [23].

Encryption

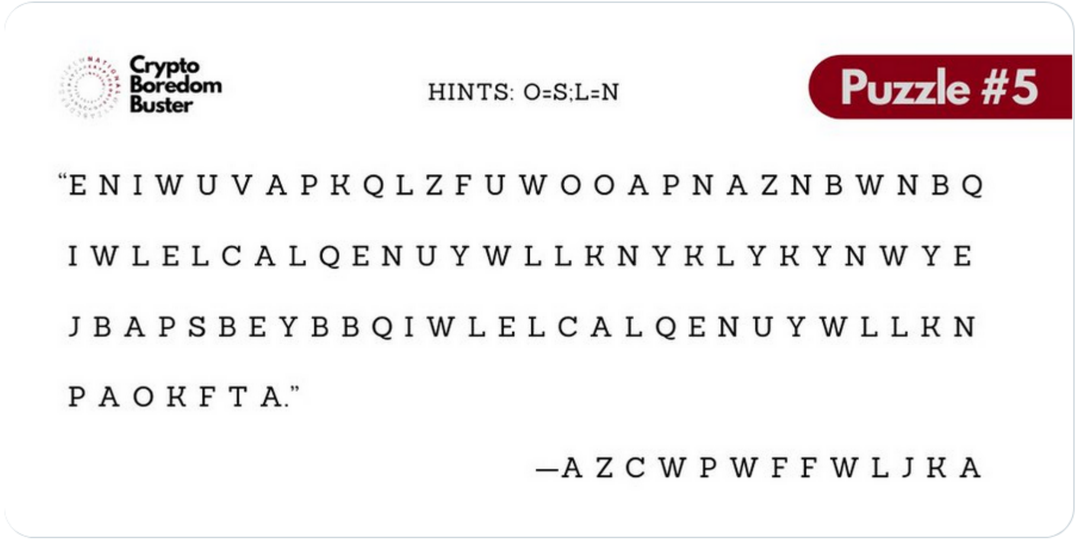
A puzzle from the NSA

NSA/CSS
@NSAGov

Many famous and not so famous people in history have written memorable things about the creative human mind. Some were authors who dabbled in ciphers.

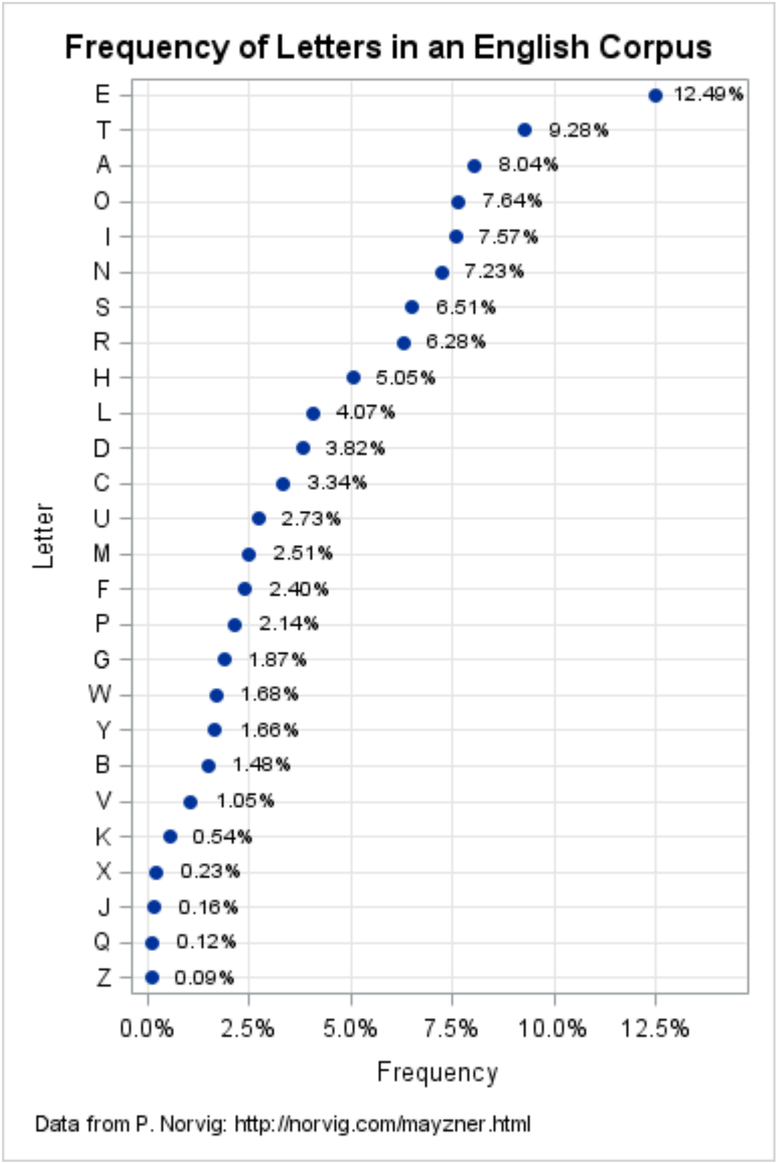
Can you decipher the quote and who said it? No spaces this time.

Puzzle 4 Secret Message: Innovation is key



3:11 PM · Apr 28, 2020 · Sprout Social

Cryptanalysis using English letter frequencies.



First Letter	Second Letter by frequency
A	**N** **T** R L S
C	O
D	E I
E	**R** **S** **N** **D** **A** T C L
H	**E** A I
I	**N** T S C
L	E L I A
M	E A
N	**T** **D** G E A
O	**N** **R** U F
R	**E** O A I T
S	**T** E A I O
T	**H** **O** I E A
V	E

Weight by frequency 0.5-0.75 0.75-1.0 **1.0-1.5** **1.5-2.0** **>2.0%**

The Solution to the NSA Puzzle

NSA/CSS
@NSAGov

Replying to @NSAGov

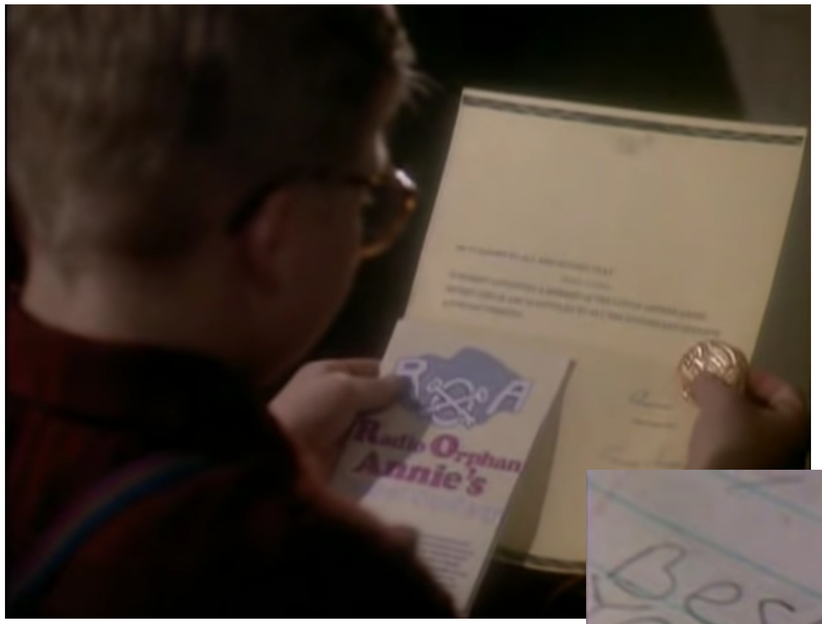
Puzzle 5 Solution: 'It may be roundly asserted that human ingenuity cannot concoct a cipher which human ingenuity cannot resolve' - Edgar Allan Poe

3:31 PM · May 5, 2020 · Twitter for iPhone

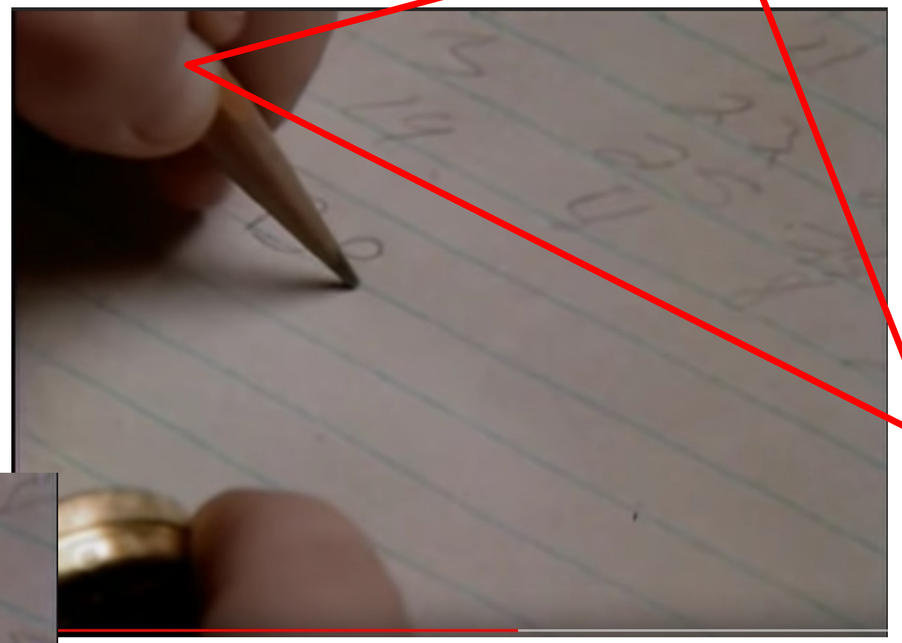
23 Retweets **4** Quote Tweets **68** Likes

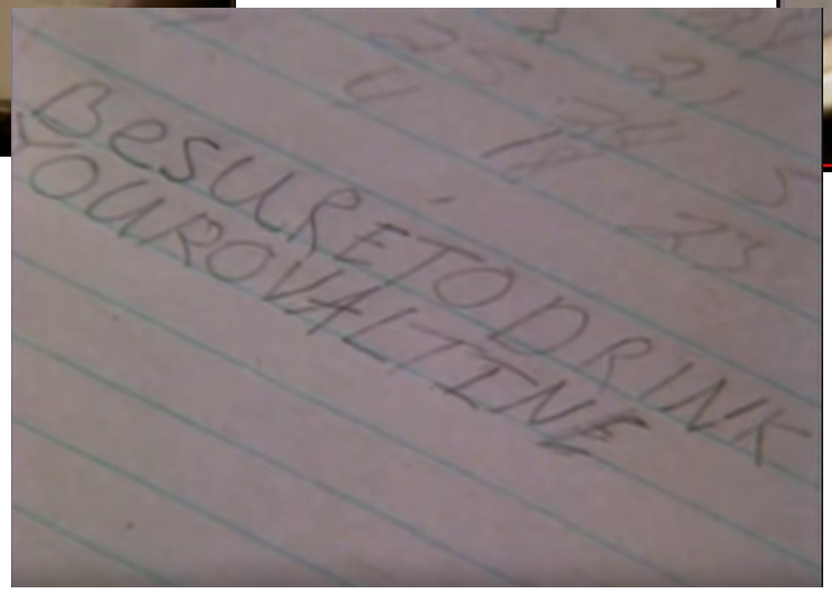
Components of a Cryptosystem

A Key



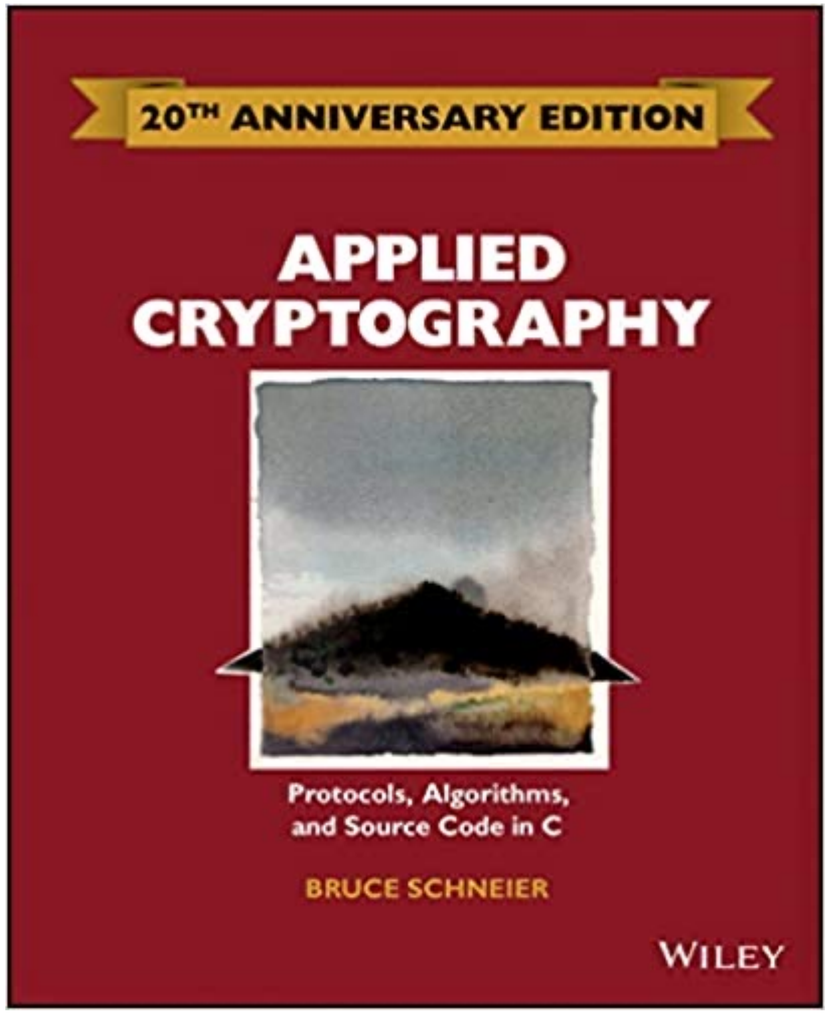
Ciphertext





Message/Plaintext

There is a lot to cover!



Product details

Publisher : Wiley; 20th edition (March 30, 2015)

Language : English

Hardcover : 784 pages

ISBN-10 : 1119096723

ISBN-13 : 978-1119096726

Item Weight : 3.23 pounds

Dimensions : 7.7 x 2 x 9.3 inches

Best Sellers Rank: #426,951 in Books (See Top 100 in Books)

#204 in Web Encryption

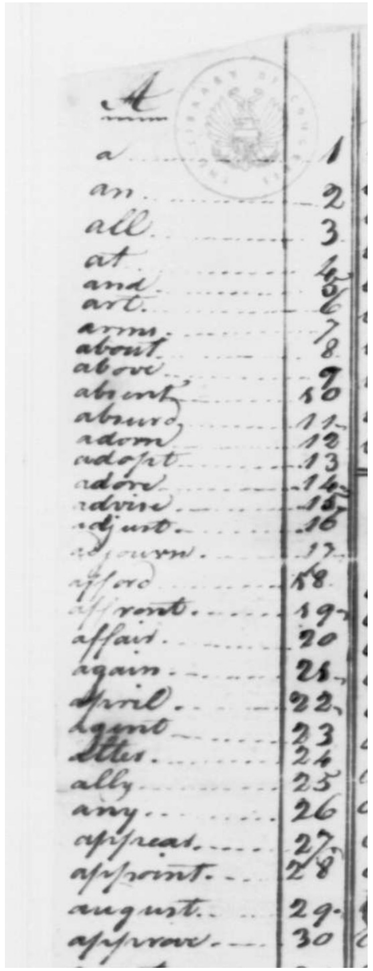
#209 in Computer Cryptography

#3,666 in Computer Science (Books)

Customer Reviews: ★★★★½ 59 ratings

Classical Cryptography

Codes — Communication with a prearranged secret



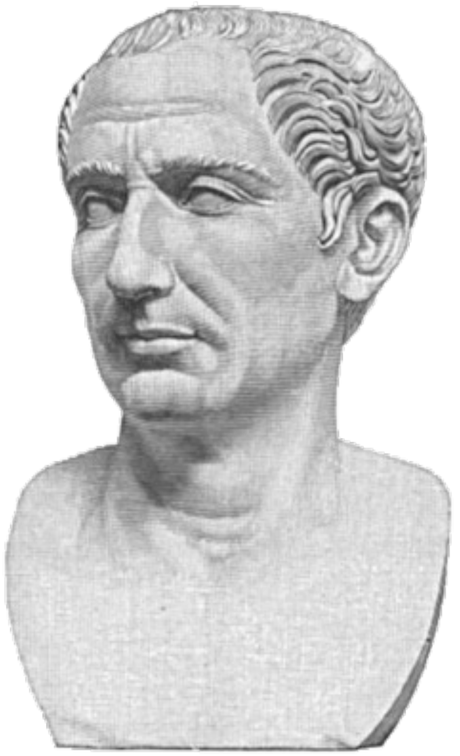
	A	219	gun	436	offer	655	value
1	a	220	go	437	office	656	virtue
2	an	221	gain	438	onset	657	visit
3	all	222	guide	439	order	658	valiant
4	at	223	gold	440	over	659	victory
5	and	224	glory	441	obstruct	660	vigilant
6	art	225	gunner	442	obtain	661	vigorous
7	arms	226	gloomy	443	observe	662	violent
8	about	227	govern	444	occur	663	volenteer
9	above	228	grandieure	445	offense	664	valuable
10	absent	229	guilty	446	ommit	665	voluntary

https://www.mountvernon.org/george-washington/the-revolutionary-war/spying-and-espionage/the-culper-code-book

The “Caesar Cipher” substitutes each letter with the third next letter.

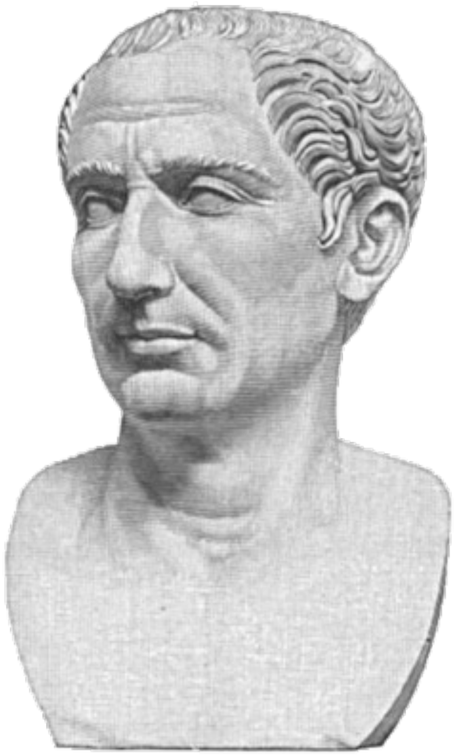
Omnia Gallia est divisa in partes tres

RPQND LDOOND HVZ GNBNVD NQ SDUZHV ZUHV



With a shift of 13, the same operation reverses the code.

```
Omnia Gallia est divisa in partes tres

Bzavn Tnyyvn rfg qvivfn va cnegrf gerf

Omnia Gallia est divisa in partes tres
```



Change the offset on every character and it gets harder still.

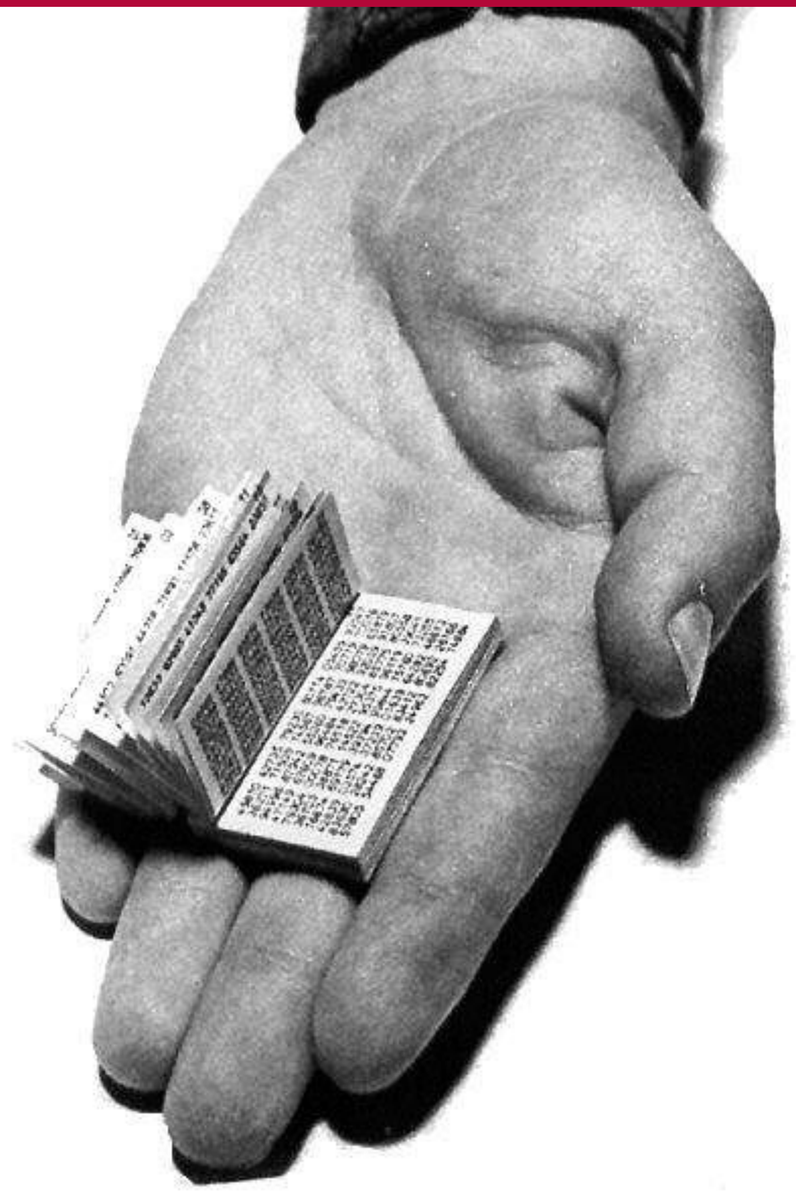
Confederate Army Cipher Disk

German Enigma Machine

Images from https://introtcs.org/public/lec_19_cryptography.html

Change the offset for every character and you have a “one time pad.”

A	ABCDEFGHIJKLMNOPQRSTUVWXYZ ZYXWVUTSRQPONMLKJIHGFEDCBA
B	ABCDEFGHIJKLMNOPQRSTUVWXYZ YXWVUTSRQPONMLKJIHGFEDCBAZ
C	ABCDEFGHIJKLMNOPQRSTUVWXYZ XWVUTSRQPONMLKJIHGFEDCBAZY
D	ABCDEFGHIJKLMNOPQRSTUVWXYZ WVUTSRQPONMLKJIHGFEDCBAZYX
E	ABCDEFGHIJKLMNOPQRSTUVWXYZ VUTSRQPONMLKJIHGFEDCBAZYXW
F	ABCDEFGHIJKLMNOPQRSTUVWXYZ UTSRQPONMLKJIHGFEDCBAZYXWV
G	ABCDEFGHIJKLMNOPQRSTUVWXYZ TSRQPONMLKJIHGFEDCBAZYXWVU
H	ABCDEFGHIJKLMNOPQRSTUVWXYZ SRQPONMLKJIHGFEDCBAZYXWVUT
I	ABCDEFGHIJKLMNOPQRSTUVWXYZ RQPONMLKJIHGFEDCBAZYXWVUTS
J	ABCDEFGHIJKLMNOPQRSTUVWXYZ QPONMLKJIHGFEDCBAZYXWVUTSR
K	ABCDEFGHIJKLMNOPQRSTUVWXYZ PONMLKJIHGFEDCBAZYXWVUTSRQ
L	ABCDEFGHIJKLMNOPQRSTUVWXYZ ONMLKJIHGFEDCBAZYXWVUTSRQP
M	ABCDEFGHIJKLMNOPQRSTUVWXYZ NMLKJIHGFEDCBAZYXWVUTSRQPO
N	ABCDEFGHIJKLMNOPQRSTUVWXYZ MLKJIHGFEDCBAZYXWVUTSRQPON
O	ABCDEFGHIJKLMNOPQRSTUVWXYZ LKJIHGFEDCBAZYXWVUTSRQPONM
P	ABCDEFGHIJKLMNOPQRSTUVWXYZ KJIHGFEDCBAZYXWVUTSRQPONML
Q	ABCDEFGHIJKLMNOPQRSTUVWXYZ JIHGFEDCBAZYXWVUTSRQPONMLK
R	ABCDEFGHIJKLMNOPQRSTUVWXYZ IHGFEDCBAZYXWVUTSRQPONMLKJ
S	ABCDEFGHIJKLMNOPQRSTUVWXYZ HGFEDCBAZYXWVUTSRQPONMLKJI
T	ABCDEFGHIJKLMNOPQRSTUVWXYZ GFEDCBAZYXWVUTSRQPONMLKJIH
U	ABCDEFGHIJKLMNOPQRSTUVWXYZ FEDCBAZYXWVUTSRQPONMLKJIHG
V	ABCDEFGHIJKLMNOPQRSTUVWXYZ EDCBAZYXWVUTSRQPONMLKJIHGF
W	ABCDEFGHIJKLMNOPQRSTUVWXYZ DCBAZYXWVUTSRQPONMLKJIHGFE
X	ABCDEFGHIJKLMNOPQRSTUVWXYZ CBAZYXWVUTSRQPONMLKJIHGFED
Y	ABCDEFGHIJKLMNOPQRSTUVWXYZ BAZYXWVUTSRQPONMLKJIHGFEDC
Z	ABCDEFGHIJKLMNOPQRSTUVWXYZ AZYXWVUTSRQPONMLKJIHGFEDCB



Transposition: changing the order of symbols in a text.

Start with plaintext

Arrange plaintext in *m* character rows of fixed length

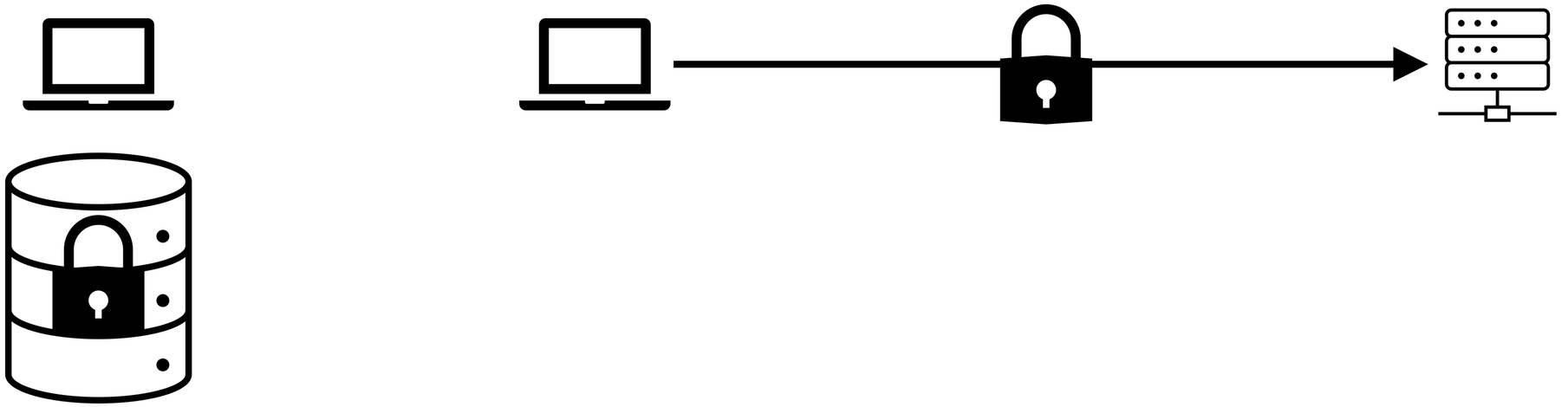
Read text out by columns

Double Transposition — Even more security!

```
LAUNCH      LTSAHI      LUCTEI
THEMIS  →   UELNME  →   SLSANH
SILES.      CISHS.      HMSIE.
```

Applications of Encryption

Encryption at rest and in transit

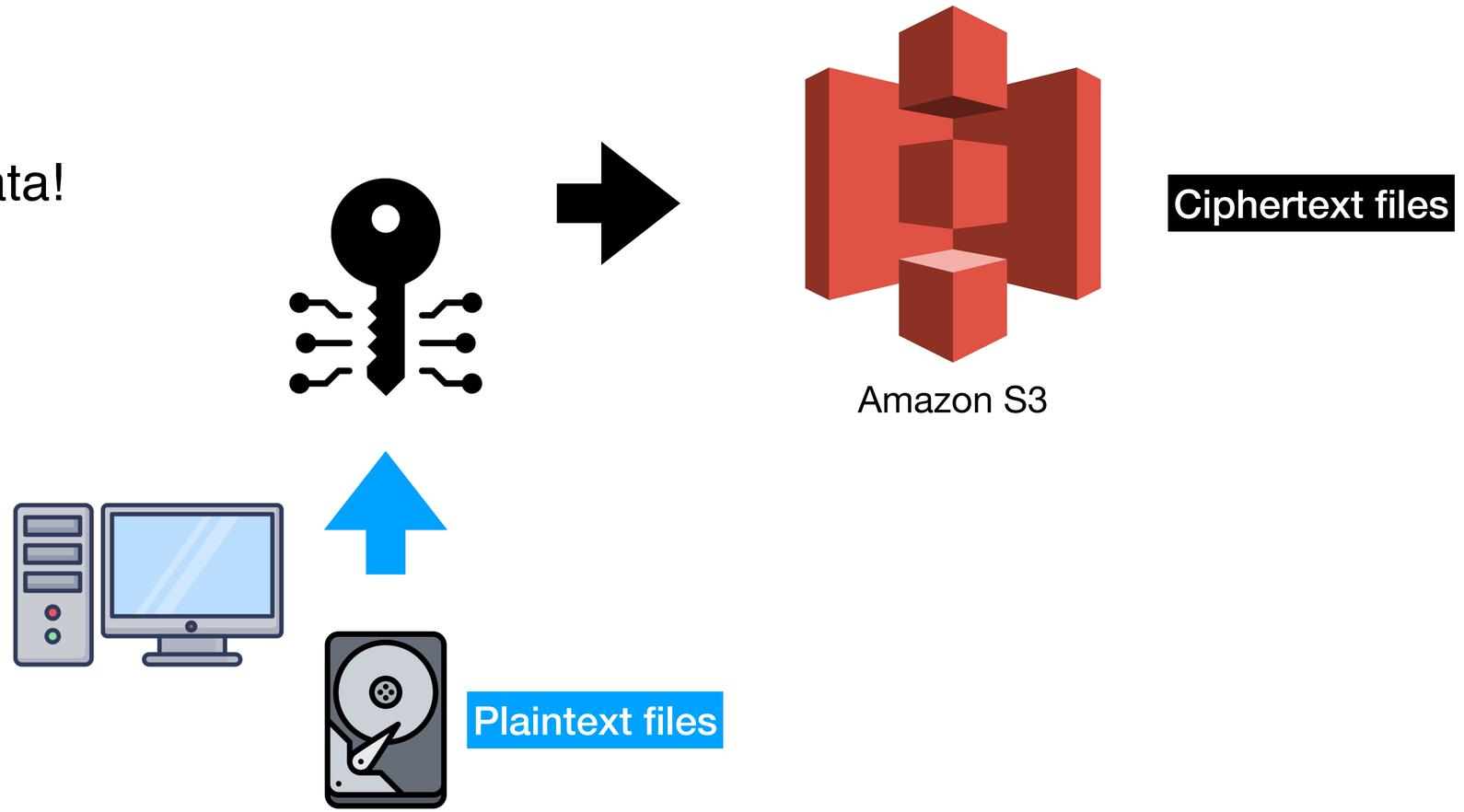


Remote secure storage

Encrypt before write.

Decrypt after read.

Remote system can’t read your data!

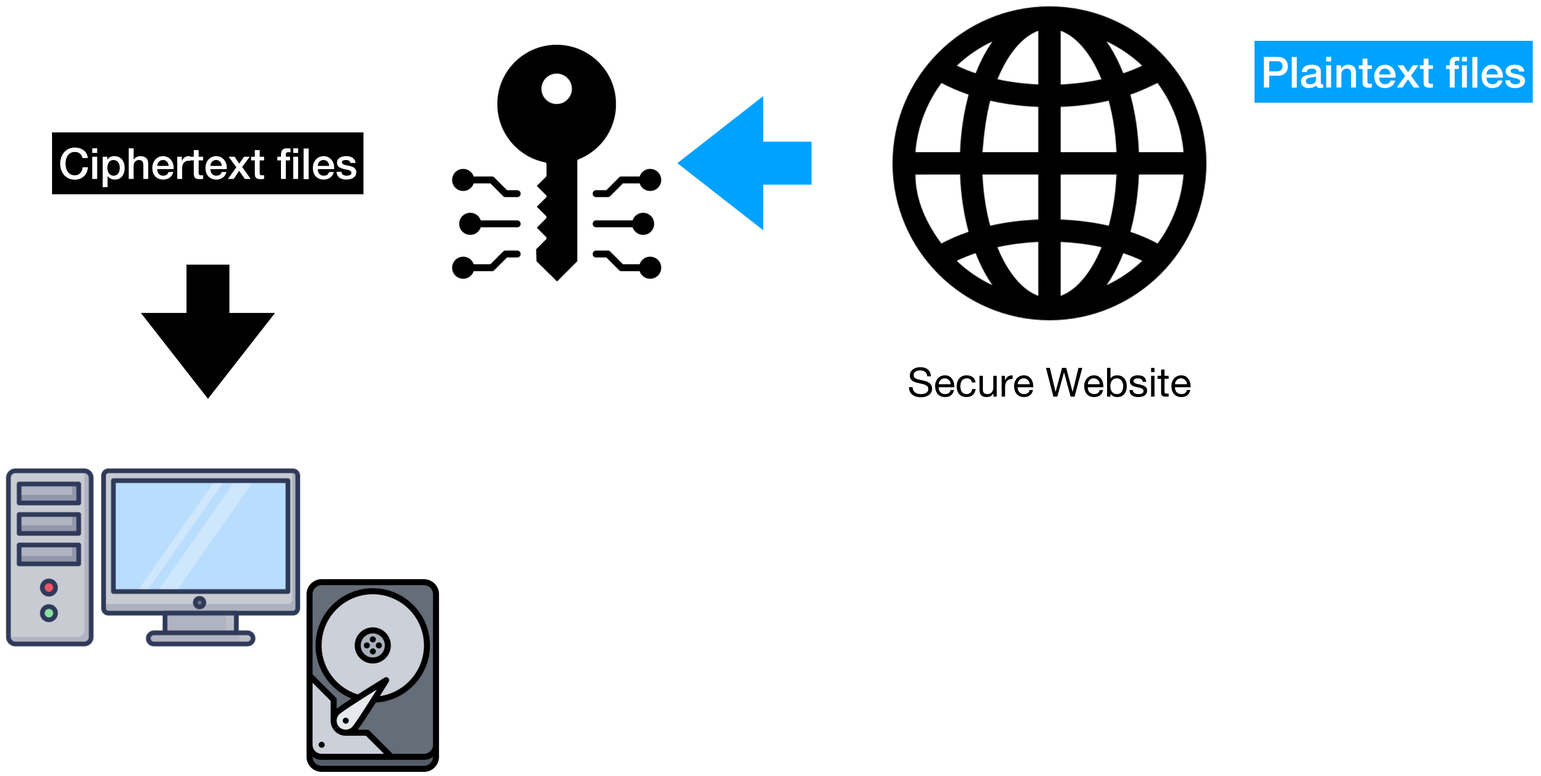


Storing (someone else's) encrypted data on your system

Website encrypts data and sends it to you.

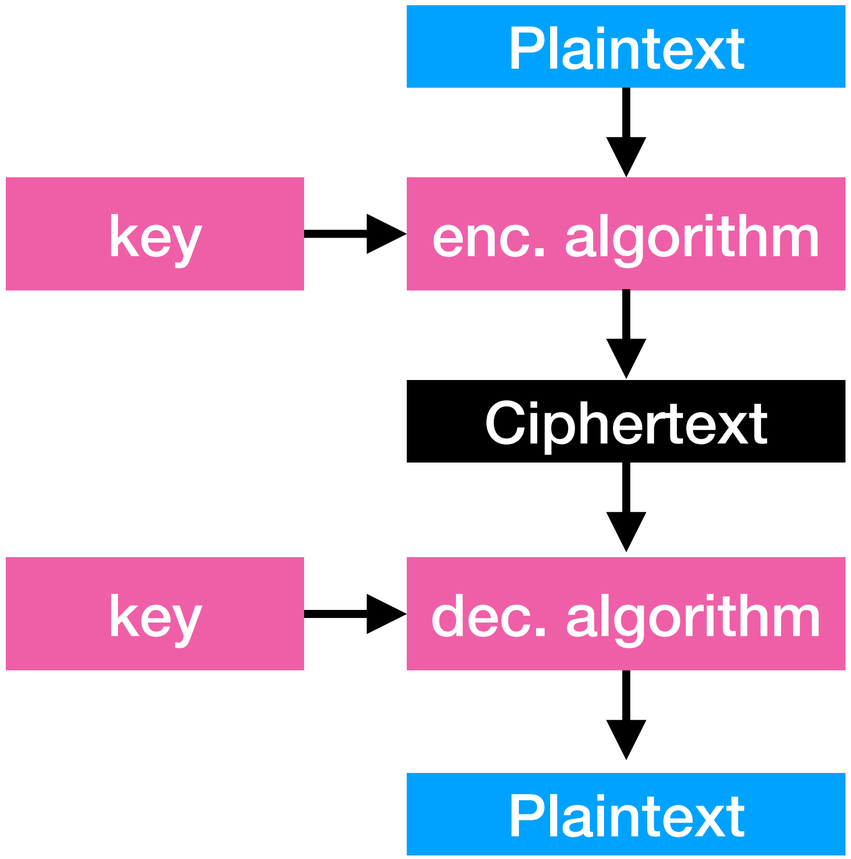
Website requests encrypted data and decrypts when you send it back.

Used for cookies



Modern Symmetric Encryption Algorithms

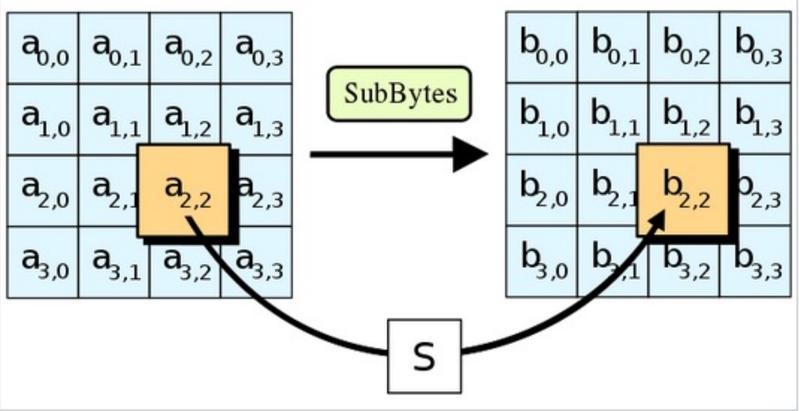
Symmetric Encryption



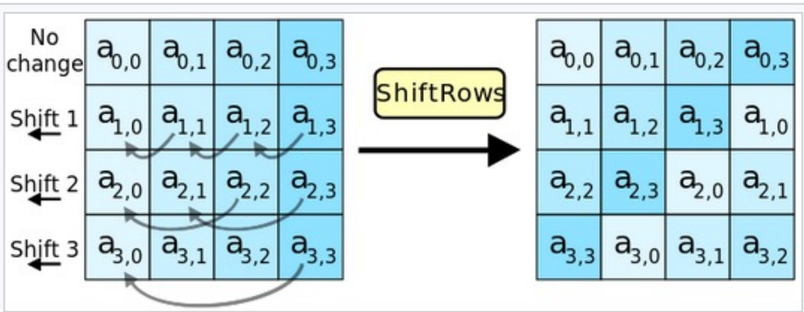
$$M' = f_{encrypt}(k, M)$$

$$M = f_{decrypt}(k, M')$$

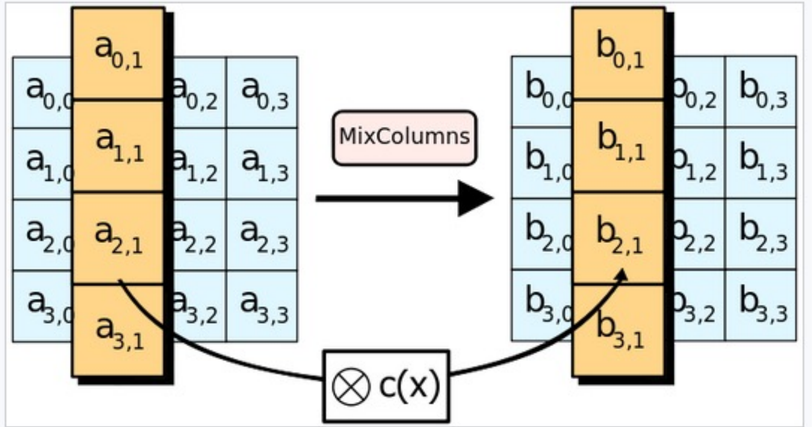
Modern Encryption: AES-256



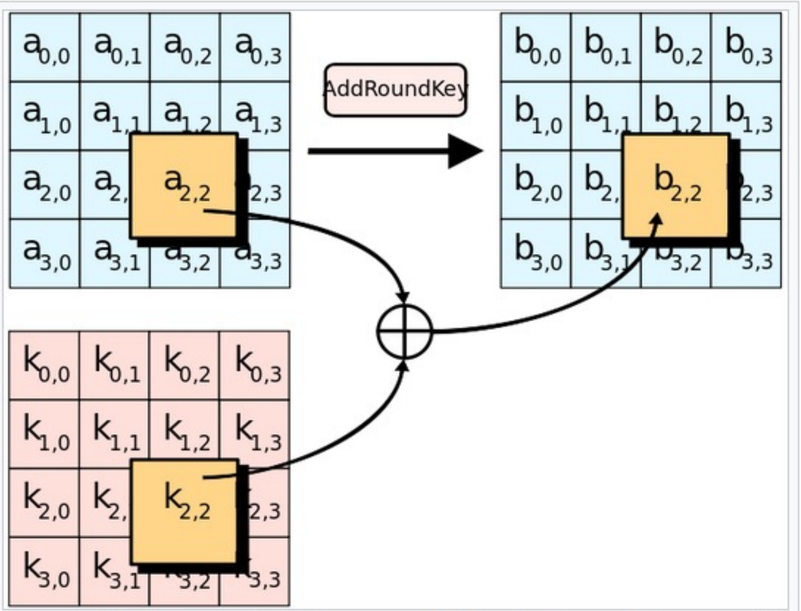
In the `SubBytes` step, each byte in the state is replaced with its entry in a fixed 8-bit lookup table, S; $b_{ij} = S(a_{ij})$.



In the `ShiftRows` step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs incrementally for each row.

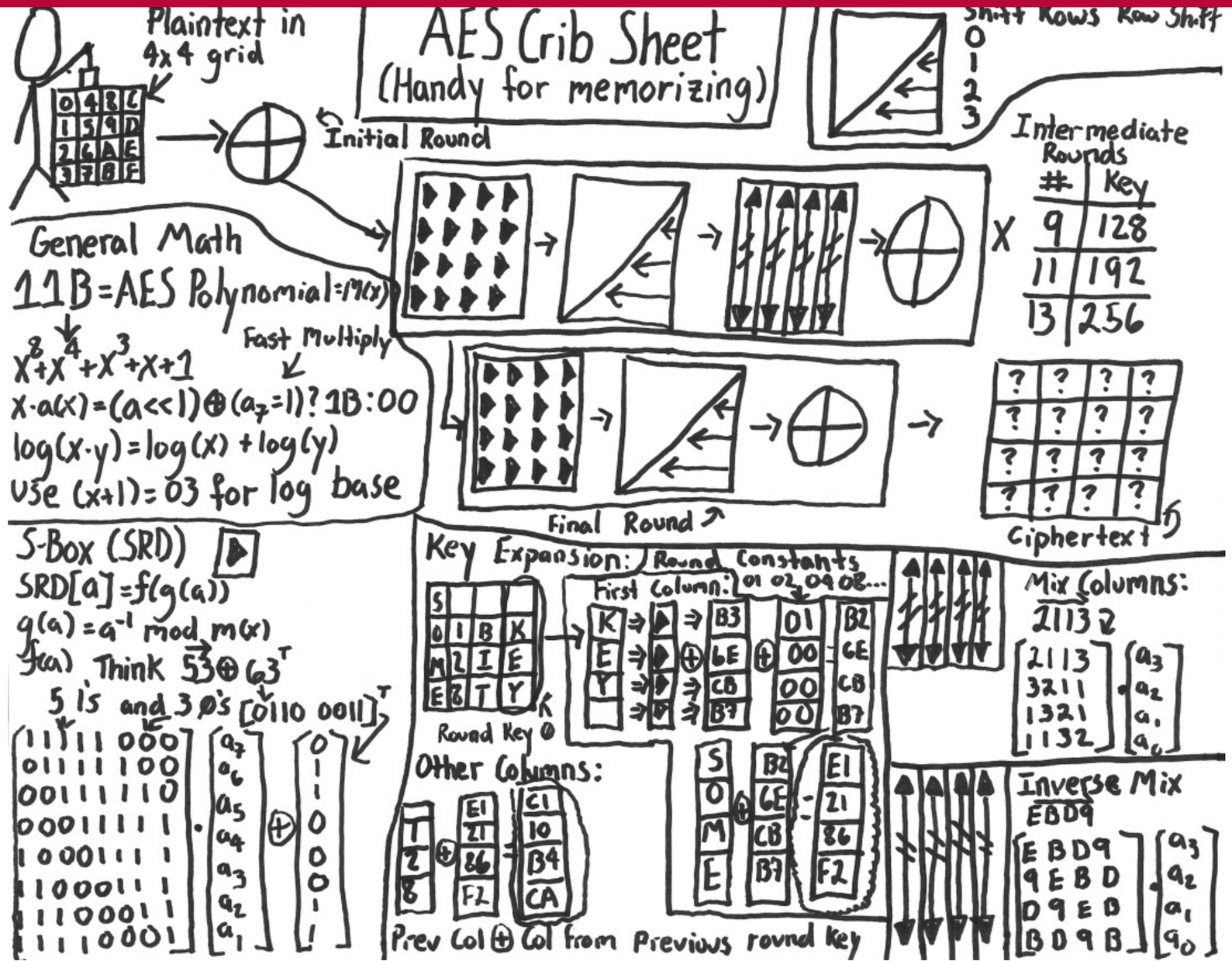


In the `MixColumns` step, each column of the state is multiplied with a fixed polynomial $c(x)$.



In the `AddRoundKey` step, each byte of the state is combined with a byte of the round subkey using the XOR operation ($\oplus$).

Someone’s attempt to draw all of AES-256



Stream Ciphers; Block Ciphers; Modes of Operation

Stream ciphers — encrypt 1 byte at a time

Block ciphers — encrypt 1 block at a time (block size parameter)

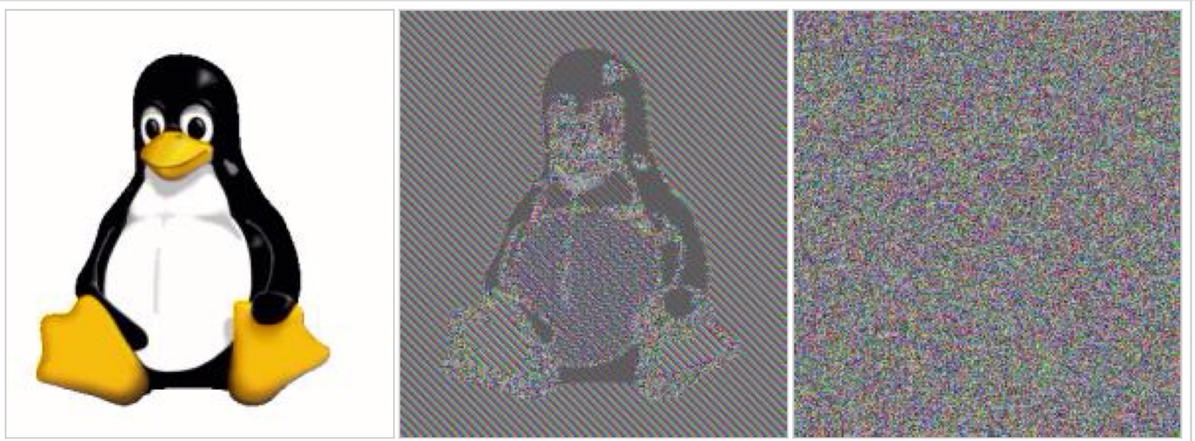
Modes of operation — describes how to put the blocks together

Electronic Code Book (ECB) mode
- each block is encrypted the same

Cipher Block Changing (CBC) mode
- first block XORed with Initialization Vector (IV)
- each block XORed with previous block

Counter Mode (CTR)
- Each block XORed with block #

Authenticated encryption
- Detect message corruption.
- The better ones are patented



Original image | Encrypted using ECB mode | Modes other than ECB result in pseudo-randomness

The image on the right is how the image might appear encrypted with CBC, CTR or any of the other more secure modes—indistinguishable from random noise. Note that the random appearance of the image on the right does not ensure that the image has been securely encrypted; many kinds of insecure encryption have been developed which would produce output just as 'random-looking'.

Security of symmetric algorithms

56-bit keys have been brute-forced (EFF; 1998)

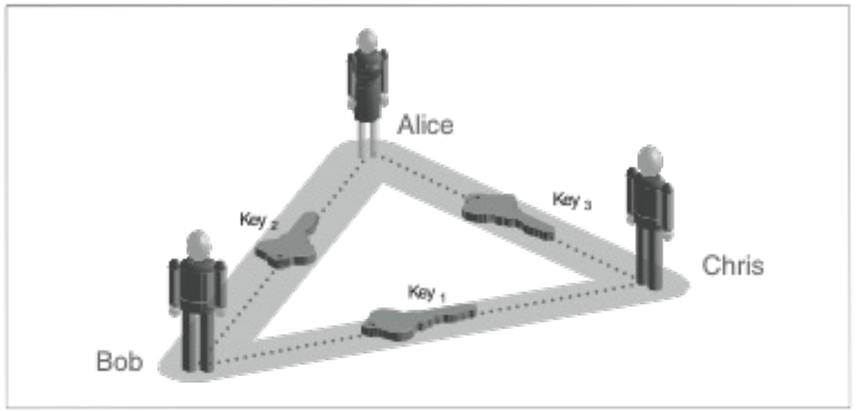
64-bit keys were brute forced (distributed.net; 2006)

80-bit keys can be brute-forced

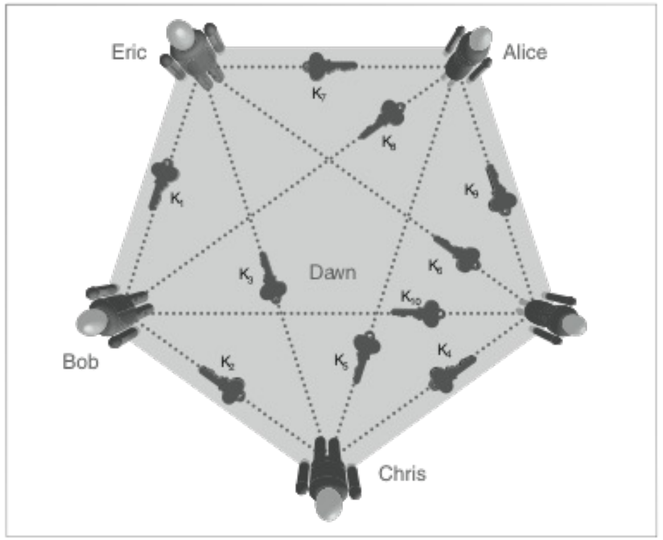
128-bit keys might be brute-forced with a quantum computer.

256-bit keys cannot be brute forced even with quantum computing

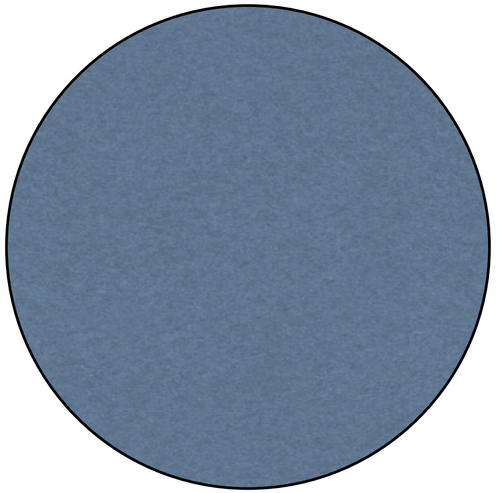
Symmetric encryption has a key problem.



3 people, 3 keys



5 people, 10 keys



1000 people, 499,550 keys!

$$\#\,keys = \frac{(n)(n-1)}{2}$$

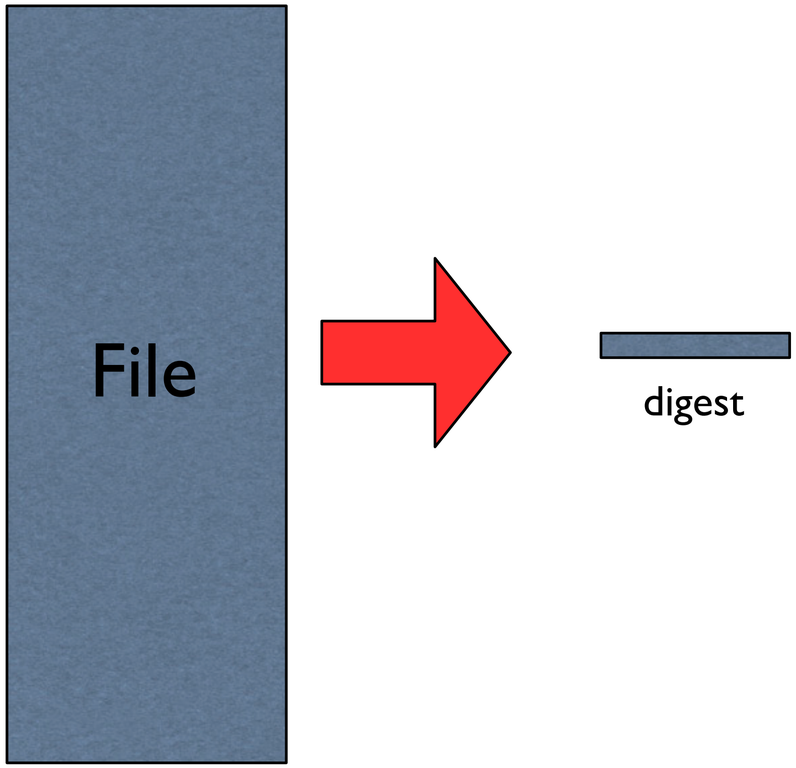
Hash Functions

Message Digests

Message Digests make a “fingerprint” of a file.

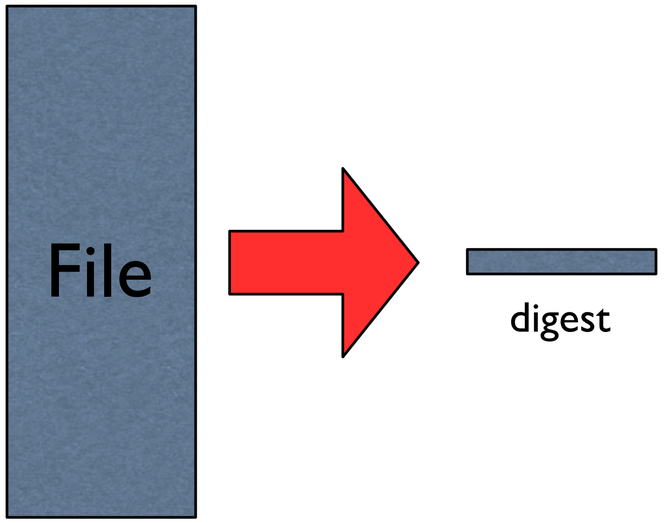
Input: 1-264 bytes

Output: 128, 160, 256 bits or more (really not need for more)



Message Digests — The same input always produces the same hash.

“We the people of the United States, in order to form a more perfect union, establish justice, insure domestic tranquility, provide for the common defense, promote the general welfare, and secure the blessing of liberty to ourselves and our posterity, do ordain and establish the Constitution of the United States of America.”




```
$ echo -n "We the people of the United States, in order to form a more perfect union, establish justice, insure domestic tranquility, provide for the common defense, promote the general welfare, and secure the blessing of liberty to ourselves and our posterity, do ordain and establish the Constitution of the United States of America." \
| openssl sha256

SHA2-256(stdin)= cd098fc401f242aa6bd99bc39b7ed273ee1c5741386853e219e069fb8c426f8a
$
```

Message Digests — change one bit, and half* the output bits change.

```
$ echo -n "We the people of the United States, in order to form a more perfect union, establish justice, insure domestic tranquility, provide for the common defense, promote the general welfare, and secure the blessing of liberty to ourselves and our posterity, do ordain and establish the Constitution of the United States of America." \
| openssl sha256

SHA2-256(stdin)= cd098fc401f242aa6bd99bc39b7ed273ee1c5741386853e219e069fb8c426f8a
$


$ echo -n "We the people of the United States, in order to form a more perfect union, establish justice, insure domestic tranquility, provide for the common defense, promote the general welfare, and secure the blessing of liberty to ourselves and our posterity, do ordain and establish the Constitution of the United States of America!" \
| openssl sha256
SHA2-256(stdin)= 1a2dd0fd16fb801265a82f2cd7284cd7658e3138ed1e5d5abaffd4ac1e82083a
$


* - ideally
```

Properties of a good Message Digest

- Digest cannot be predicted from the input.
- It's hard to find an input that produces a specific digest ("preimage resistance")
- **second preimage resistance** — For a given message M1, it's hard to find a second message M2 that has the same hash.
- **collision resistance** — It's hard to find two message with the same hash.

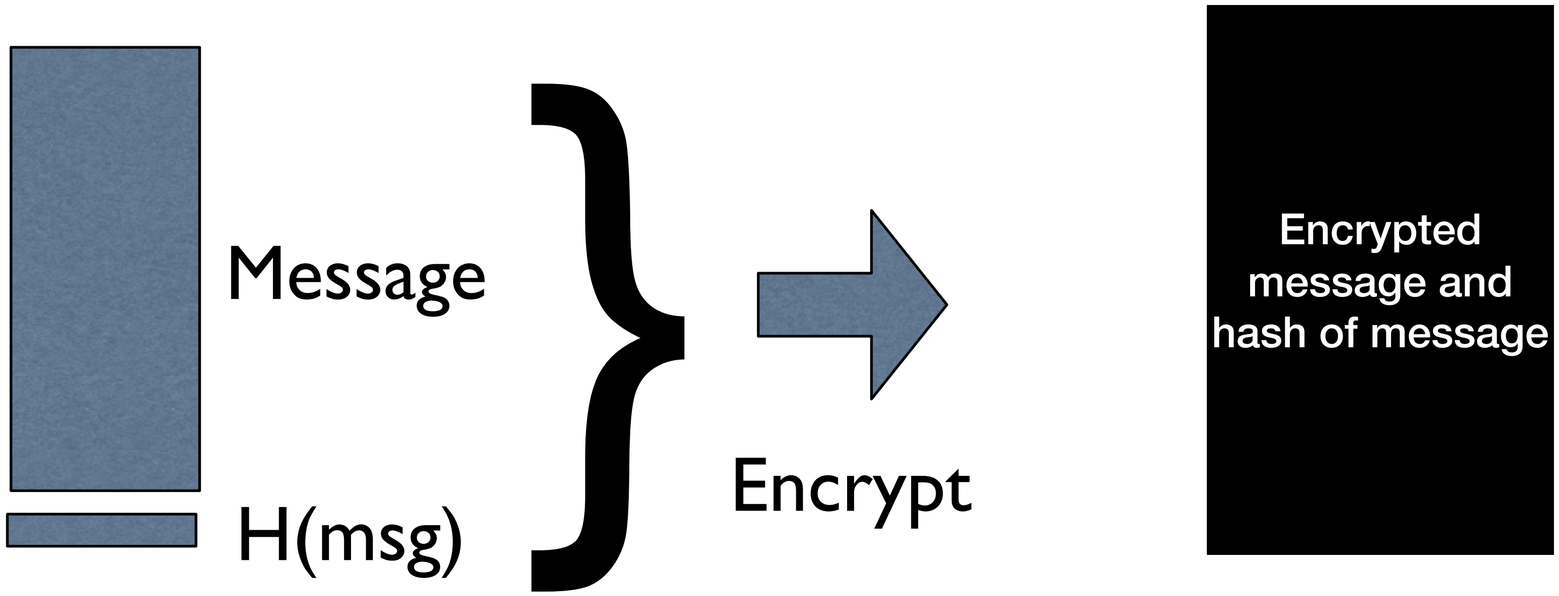
https://crypto.stackexchange.com/questions/1173/what-are-preimage-resistance-and-collision-resistance-and-how-can-the-lack-ther

Now we can trust that the encrypted data wasn't modified!

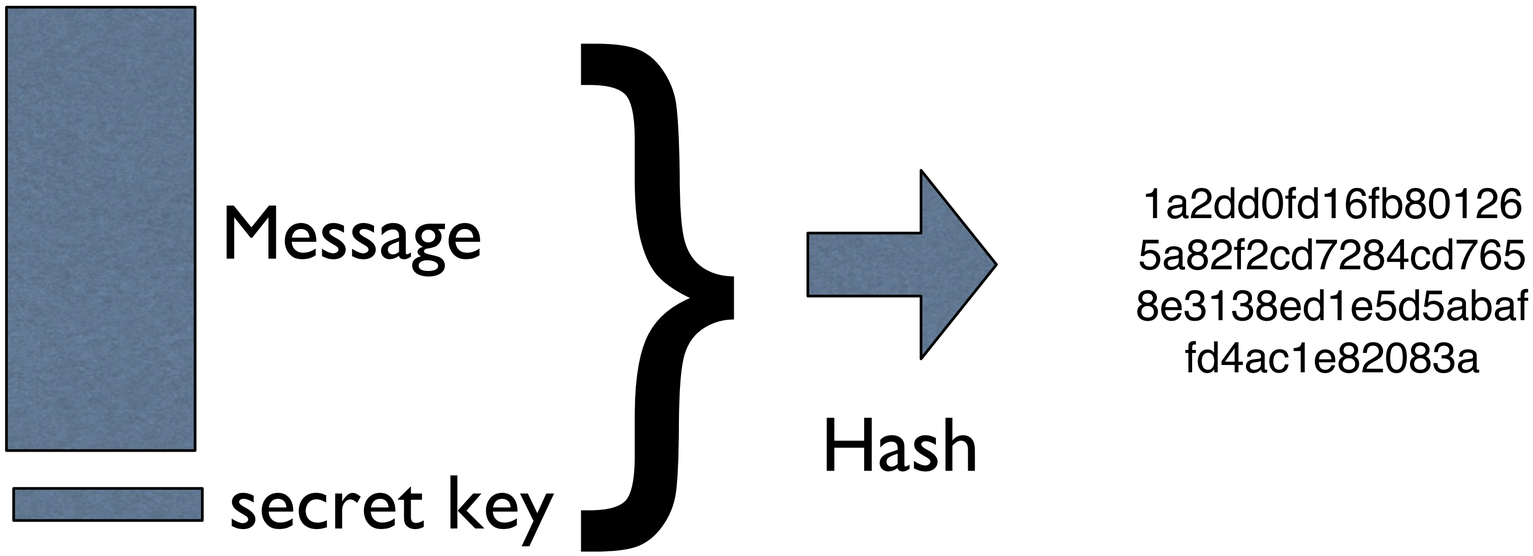
Compute the hash of the data

Encrypt the data and the hash

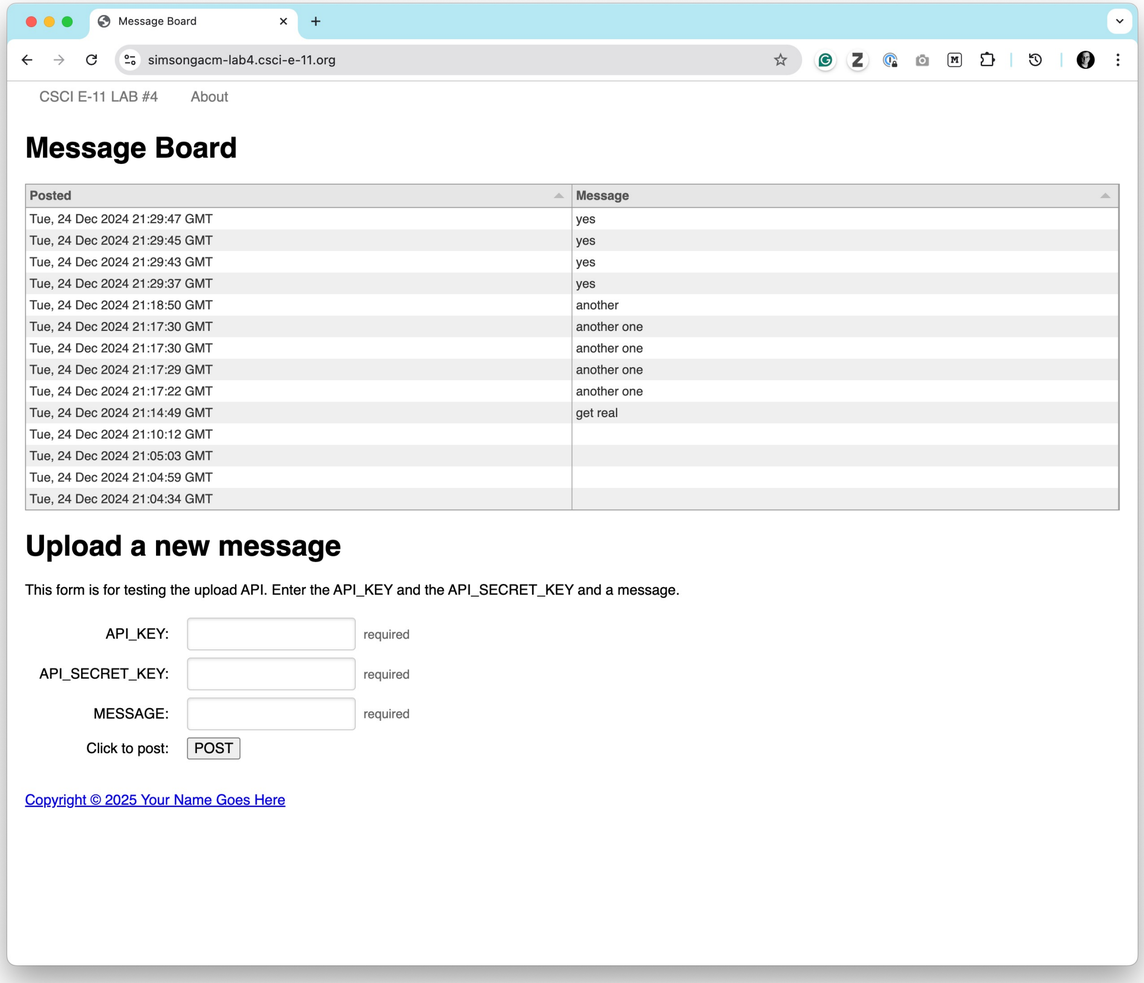
Store the package on the remote system.

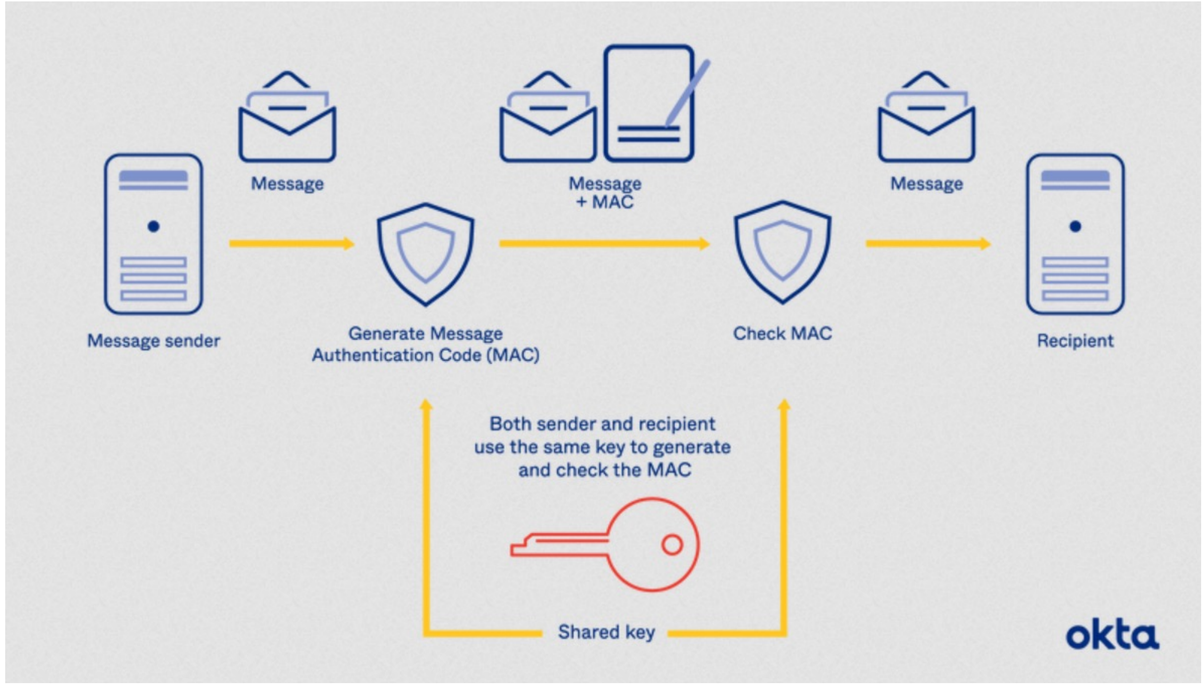


Hash-Based Message Authentication Code (HMAC)



HMACs are used as the Lab4 password hash





Hashing as proof of work

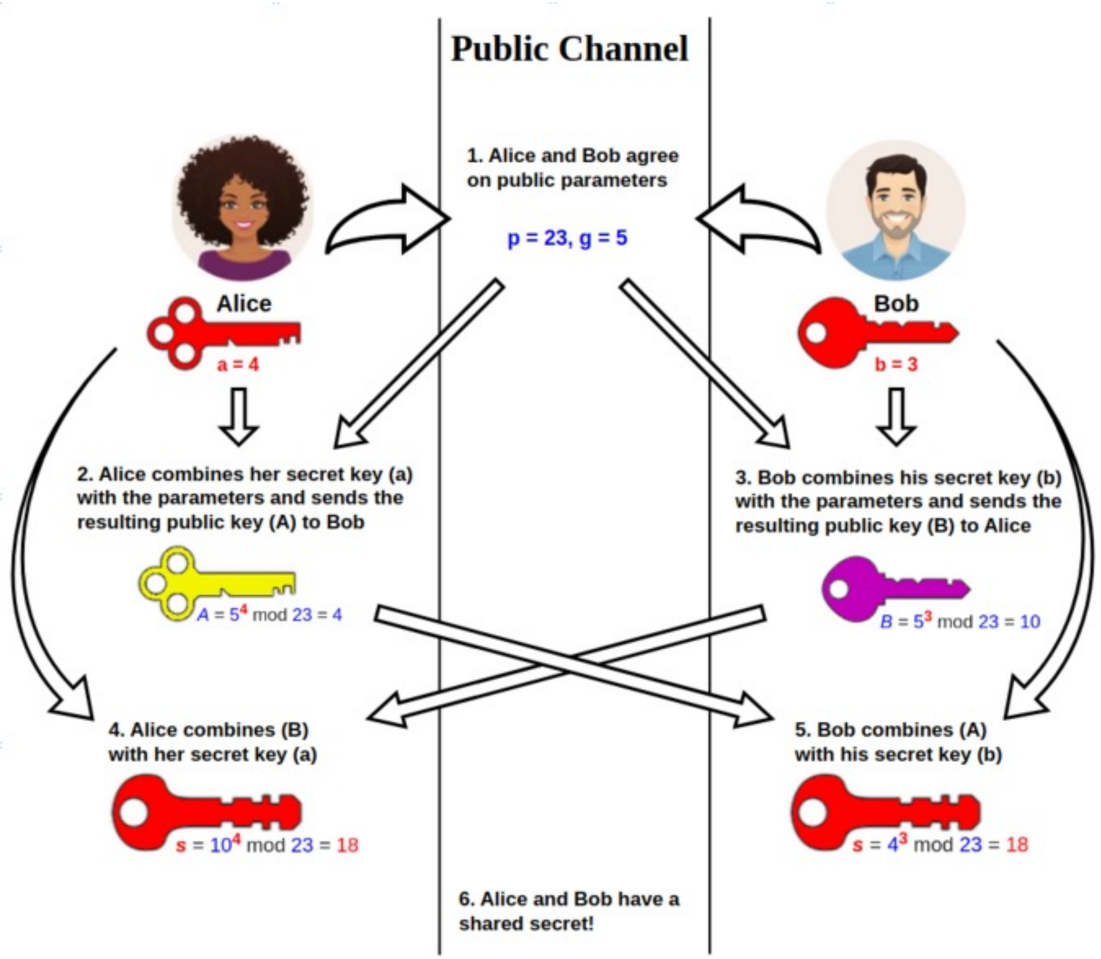
Here is a program to find a number that, when hashed, produces a hash with ’n’ leading zeros:

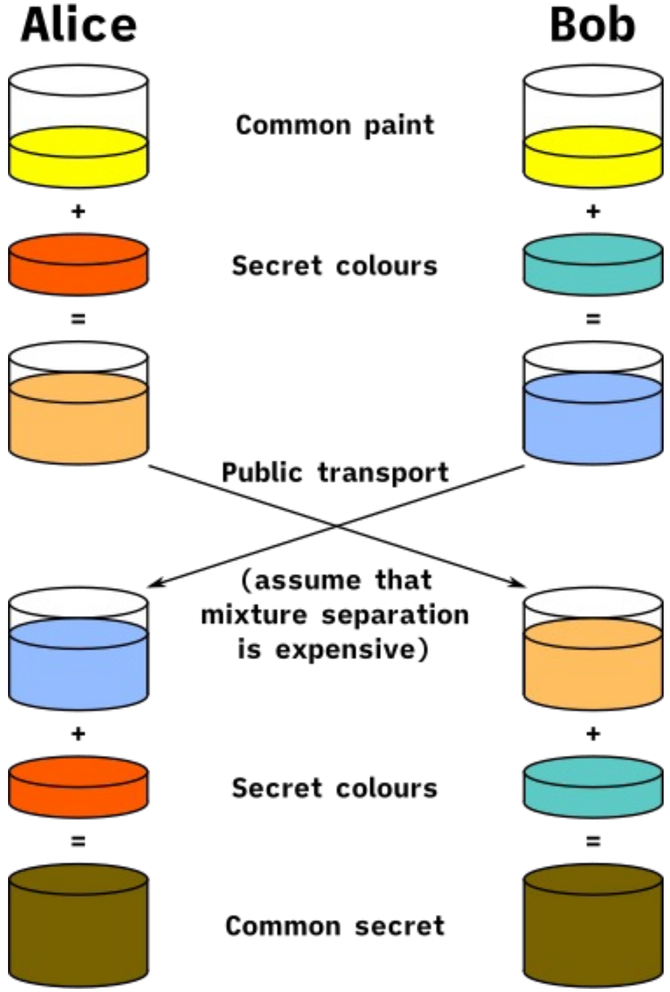
```
"""
find a number that, when hashed, produces as SHA256 hash with a specific number of leading zeros.
"""

import argparse
import hashlib

if __name__=="__main__":
    parser = argparse.ArgumentParser(formatter_class=argparse.ArgumentDefaultsHelpFormatter)
    parser.add_argument(“--verbose",action='store_true')
    parser.add_argument("zeros",type=int)
    args = parser.parse_args()
    zeros = "0" * args.zeros

    found = 0
    n     = 0
    while not found:
        hasher = hashlib.sha256()
        hasher.update(str(n).encode('utf-8'))
        hexdigest = hasher.hexdigest()
        found = hexdigest[0:args.zeros] == zeros
        if found or args.verbose:
            print(f"SHA256('{n}')={hexdigest}")
        n += 1
```

Public Key (Asymmetric) Cryptography

Can Alice communicate securely to Bob if they haven’t exchanged a secret?

Diffie-Hellman Key Exchange



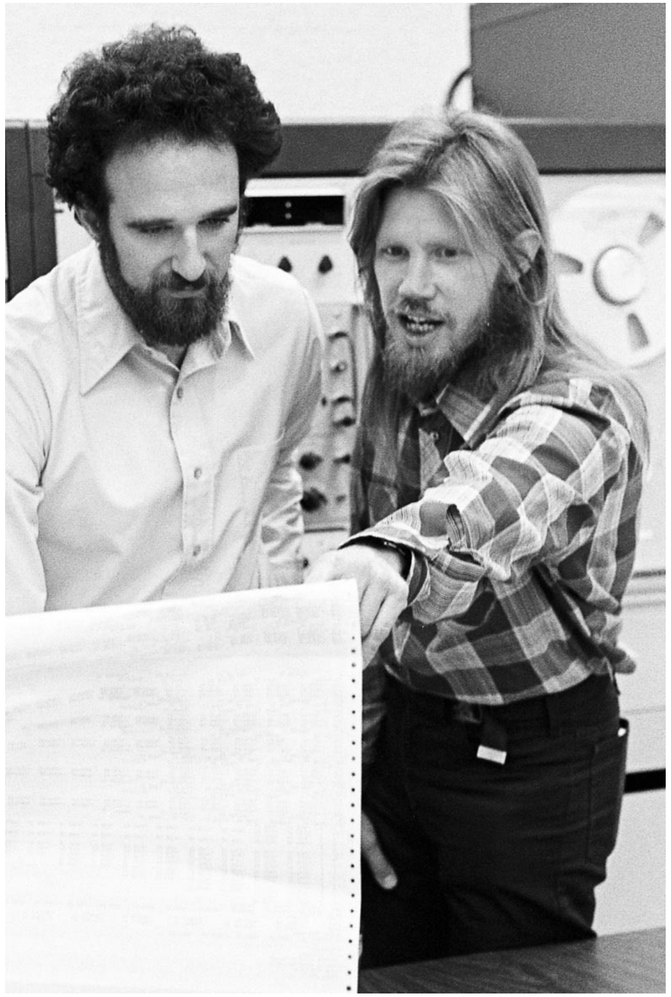


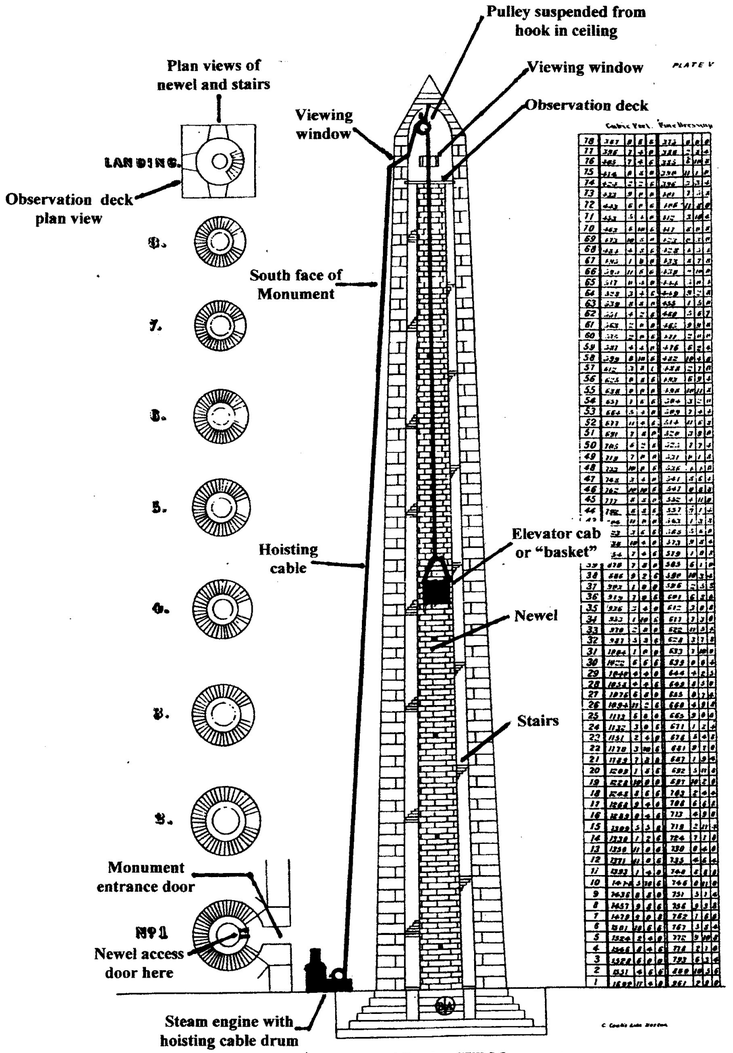
Math Quiz: Choose One

1. What are the prime factors of 1,920,179?

2. 1583 x 1213 = ?

One Way Functions



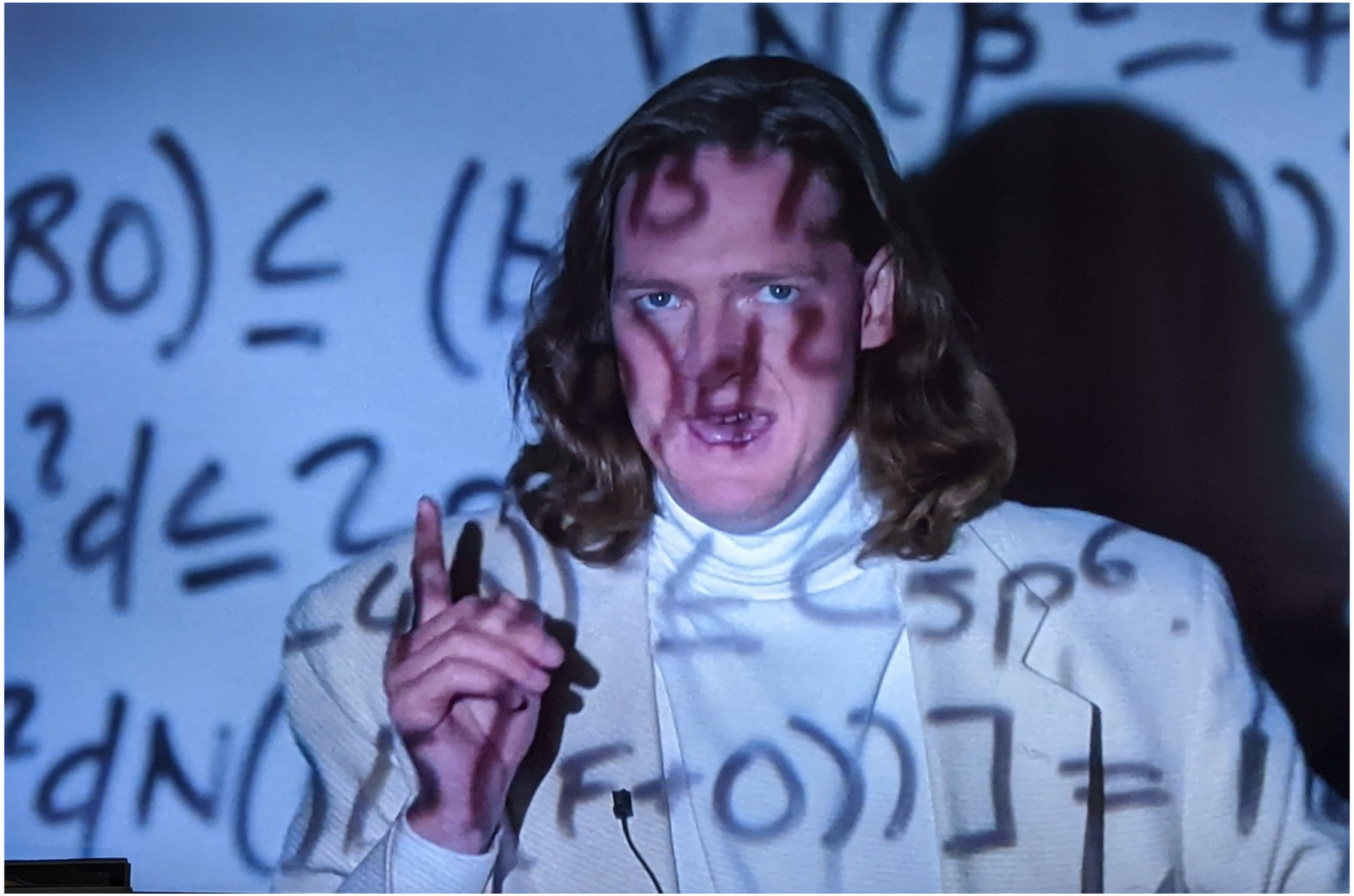


Section of Bunker Hill Monument

Reproduced from Solomon Willard, *Plans and Sections of the Obelisk on Bunker's Hill* (Boston, 1843), Plate V

Fig. 2. Conjectural diagram of the Bunker Hill Monument elevator operation.

The Rivest-Shamir-Adleman (RSA) Algorithm



Asymmetric (public key) encryption

Two keys: one to encrypt, one to decrypt!

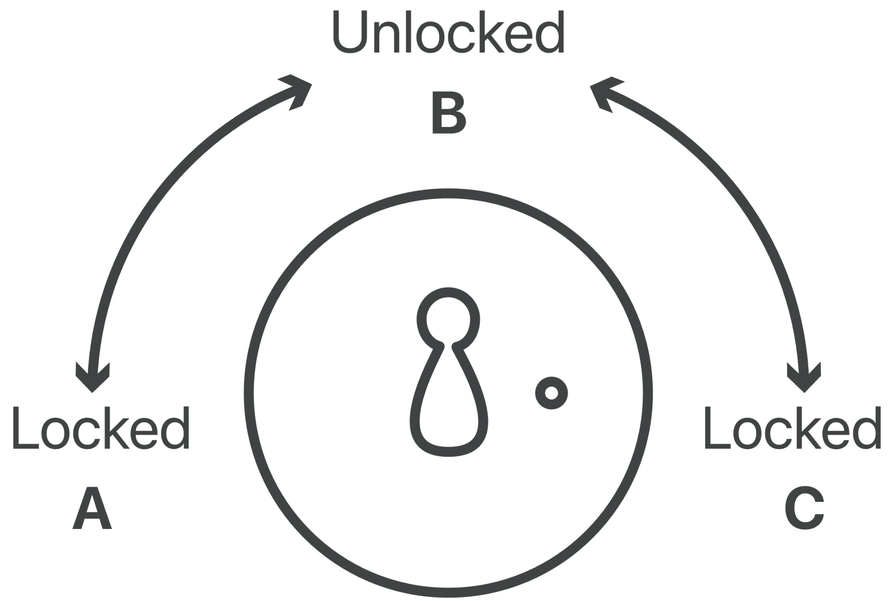
M’ = f(M,K1)

M = f’(M’,K2)

Public Key Encryption Uses

Encryption: Locked with receiver’s public key, unlocked with receiver’s private key

Digital signature and non-repudiability: hashed/encrypted by sender’s private key, verified by sender’s public key



Asymmetric Public Key Cryptography

Public key *e*

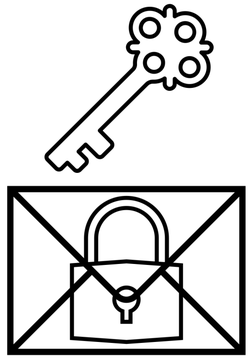
Private key *d*

Alice generates a key pair derived from two prime numbers. One she makes public and one she keeps secret.

Bob has a secret message he Wants to send to Alice

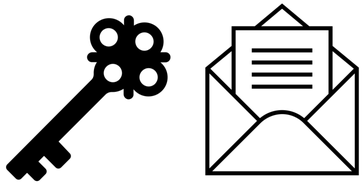
Bob encrypts his message using Alice's public key and transmits it to Alice

Alice's public key *e*



Private key *d*

Alice decrypts the message using her private key



Private key *d*

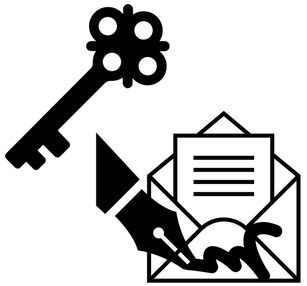
2. Alice wants to send a message to Bob and prove it came from her

Public key *e*

Private key *d*

Alice signs her message using her private key and the message content

Private key *d*



Alice’s public key *e*

Bob uses Alice’s public key to confirm the signature

Private key *d*

Alice’s public key *e*

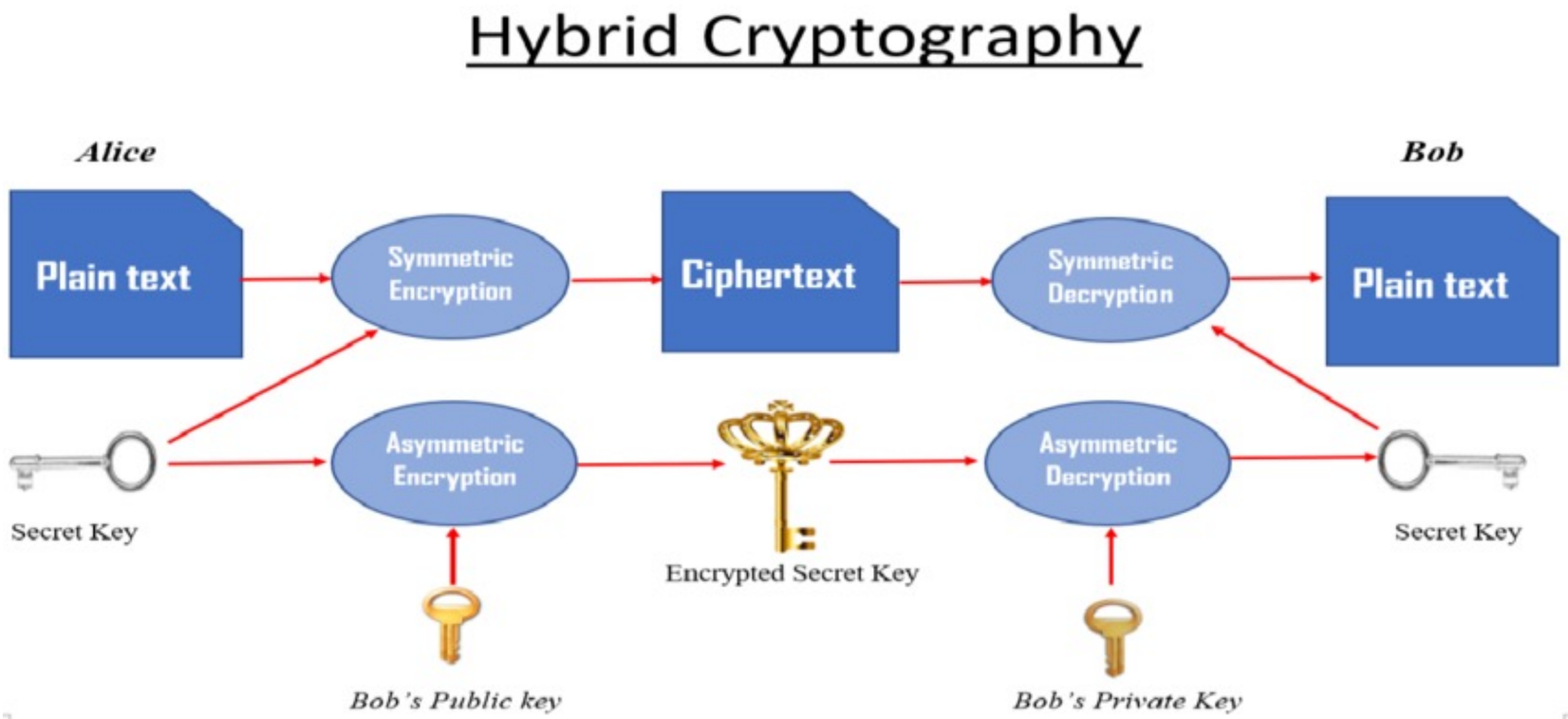
Secret Key vs. Public Key

	secret key	public key
algorithm type	symmetric	asymmetric
basis	substitution and transposition	math
speed	fast	slow
encrypts	blocks of data	numbers
uses	encrypting files	encrypting email

Hybrid Systems — Sending encrypted files

Most public key systems are actually hybrid systems

- Use Diffie-Hellman or RSA to exchange a 128-bit session key
- Use AES to encrypt bulk information



https://www.researchgate.net/publication/363656300_Secure_File_Storage_On_Cloud_Using_Hybrid_Cryptography

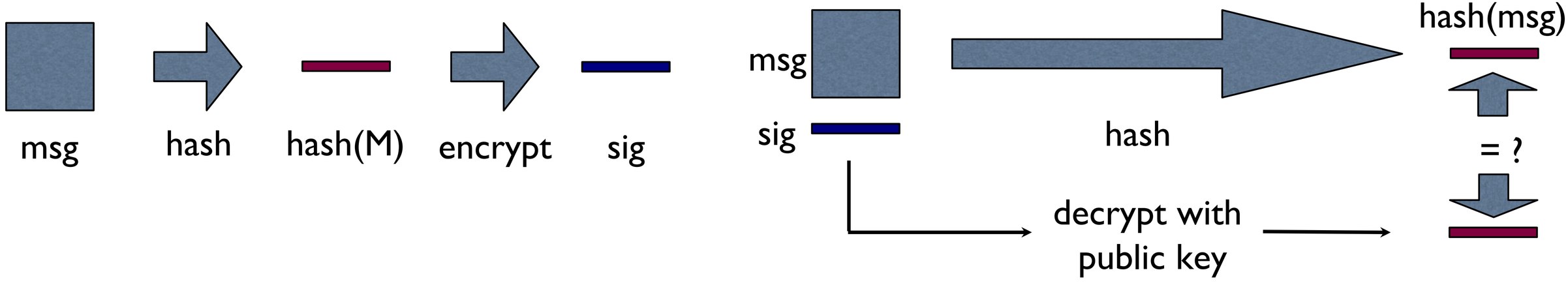
Hybrid Systems — Digital Signatures

Signing a signature:

- Hash the document
- Encrypt the hash with the private key
- Send both the document and the encrypted hash

Verifying the signature:

- Decrypt the document's encrypted hash with the public key
- Hash the document (without the hash)
- Compare the decrypted hash to the computed hash



Cryptosystem Design

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

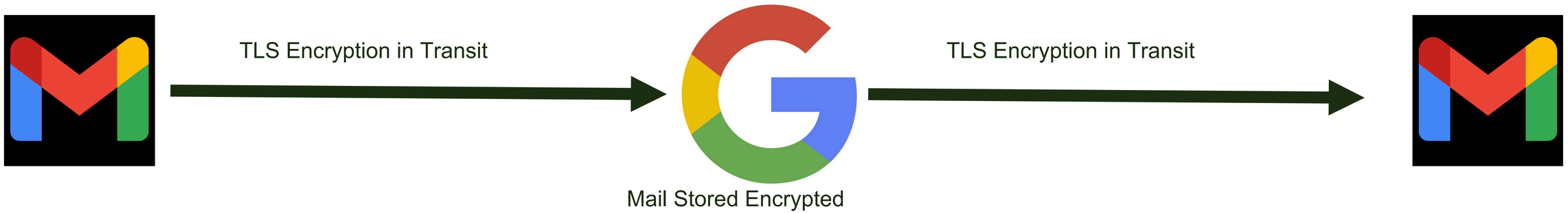
- Auguste Kerckhoffs, 1883

According to Steve Bellovin, the NSA version is “assume that the first copy of any device we make is shipped to the Kremlin.”

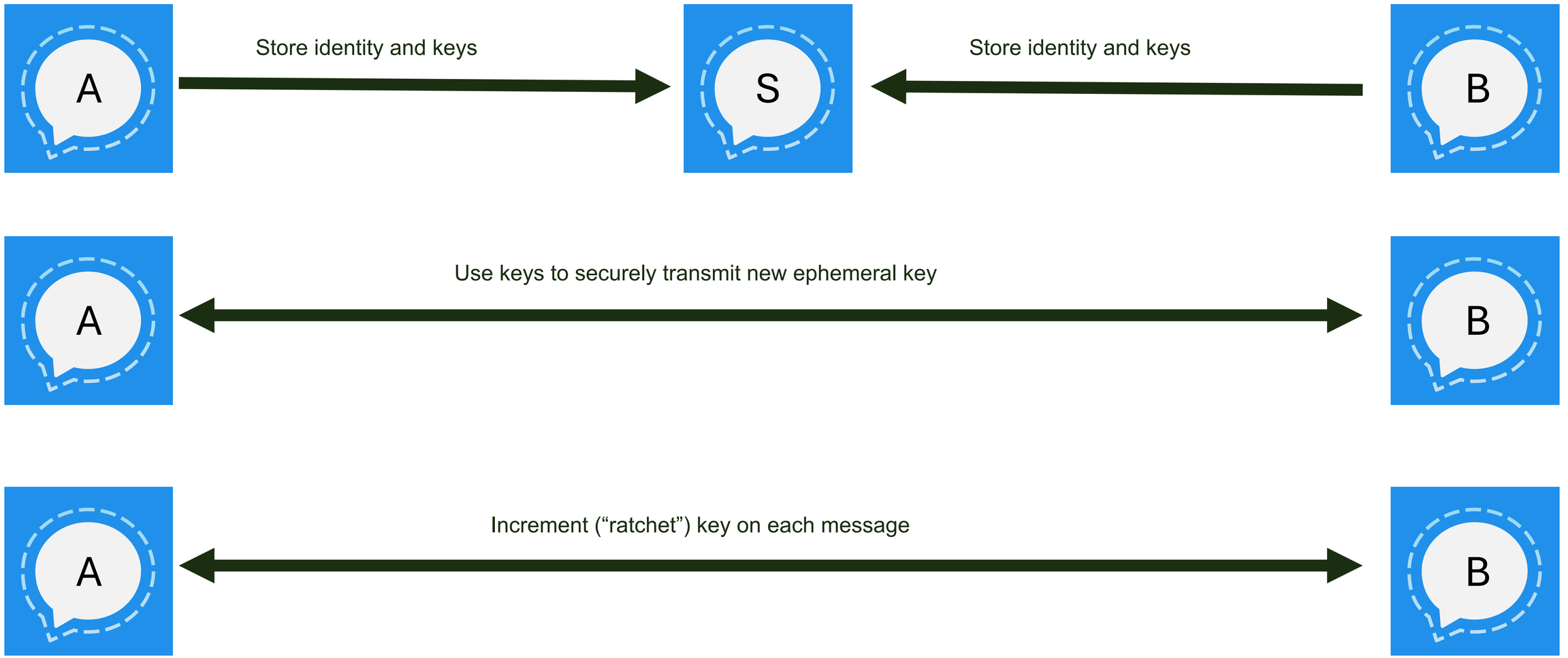
Q: What are some vulnerabilities that can occur in cryptographic systems?

- Simple design flaws can introduce vulnerabilities
- Bugs (or sabotaged) code
- Unsafely padding short messages, e.g., with all 1s or 0s
- Resending identical messages
- Poor use of sequence numbers
- Poor choice of random number generators
- Deliberate choice of weak system configuration parameters
- Quantum computers can crack private keys.
- Back doors.

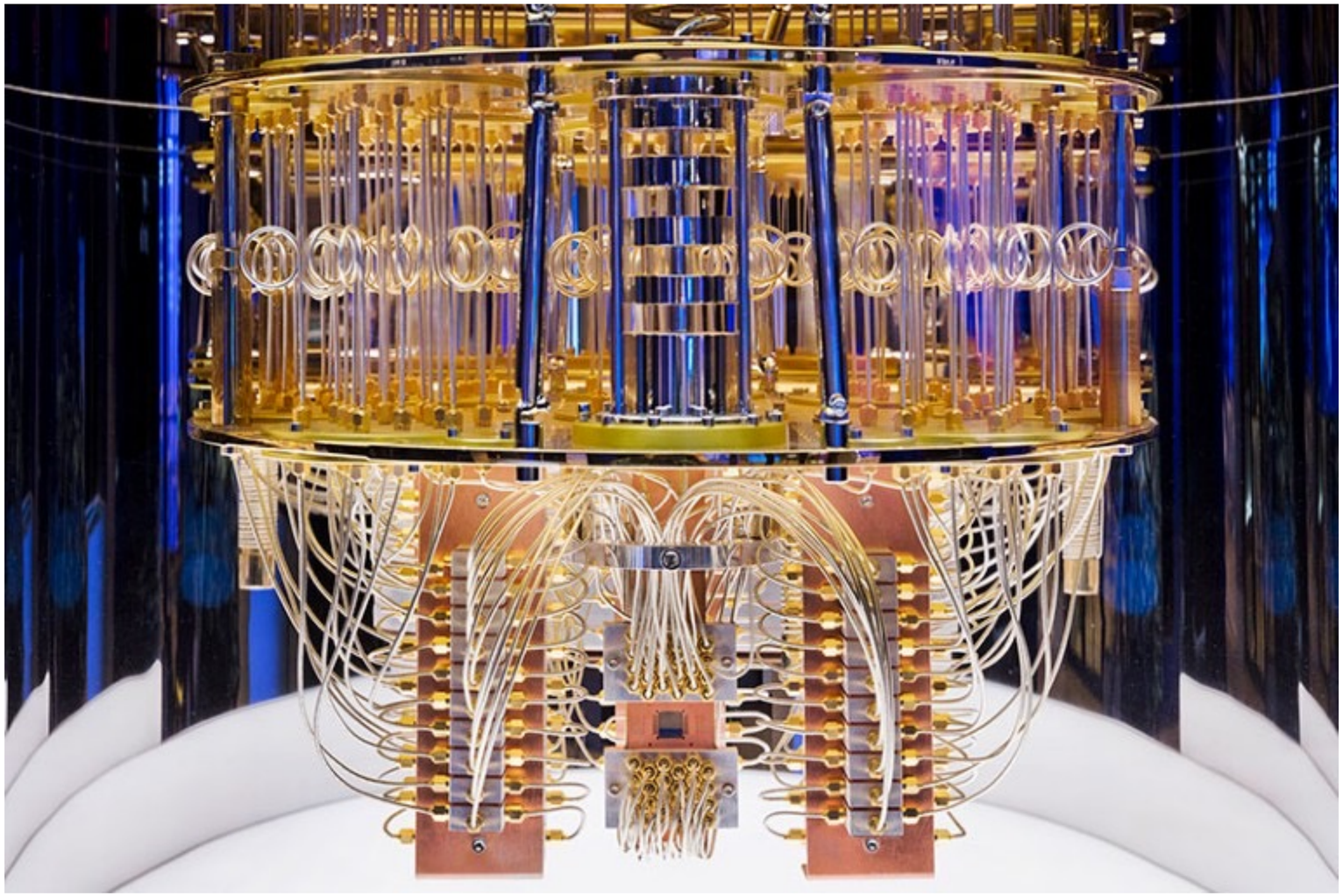
Encrypted in Transit v. End-to-End Encryption



End-to-End Encryption in Signal



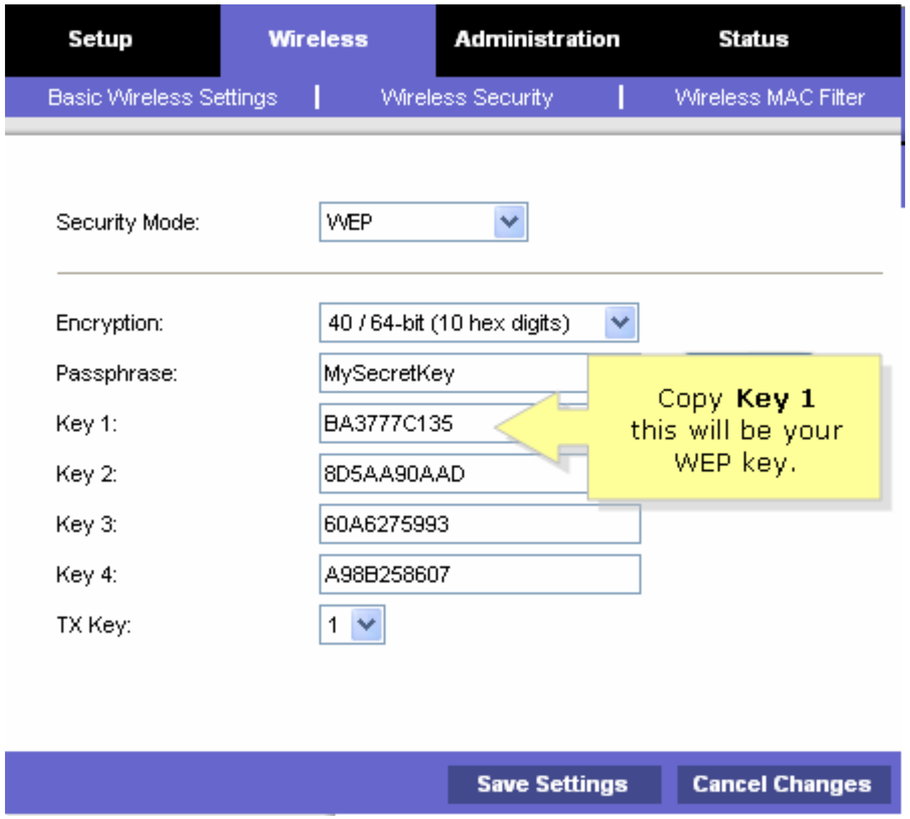
Will quantum computers break encryption?



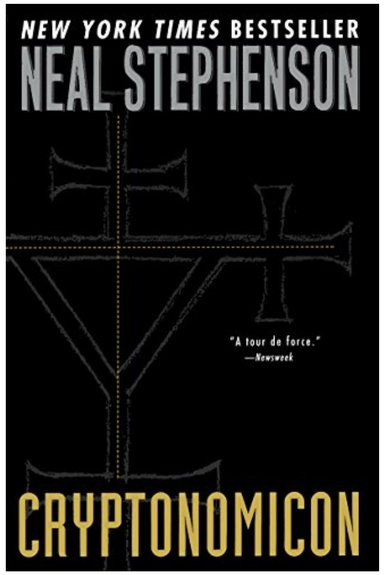
https://www.nature.com/articles/d41586-021-03476-5

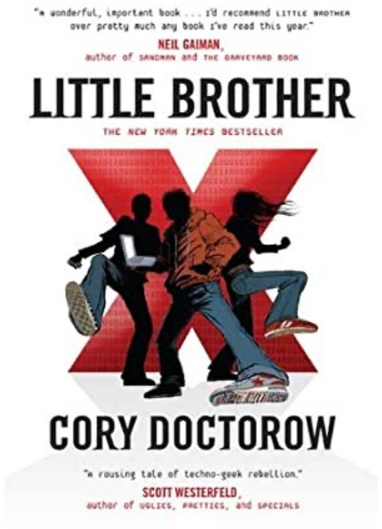
Implementation Details

When It Isn’t Random Enough

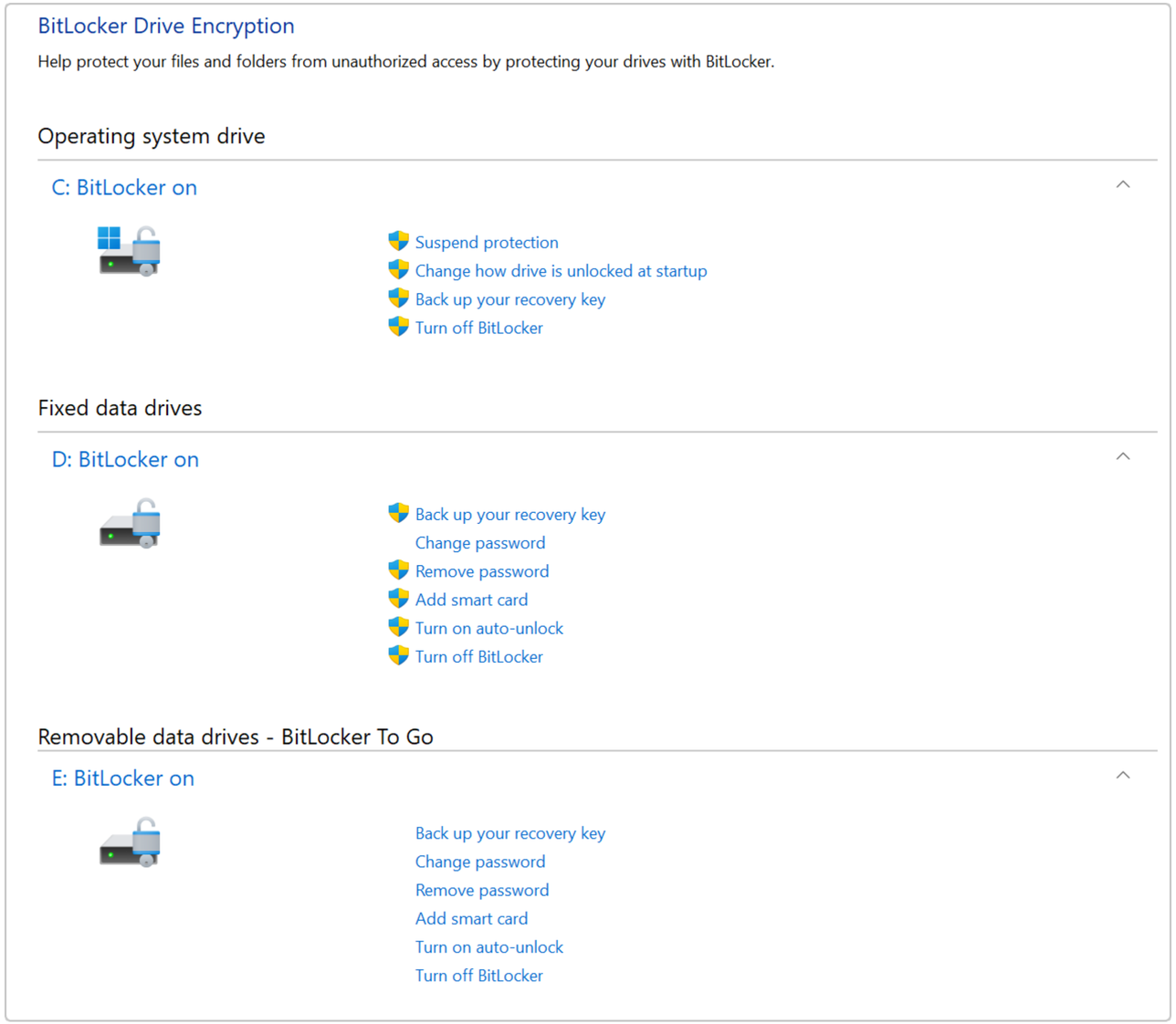


Want to Learn More?





News story of the week



DAILY COVER

Microsoft Gave FBI Keys To Unlock Encrypted Data, Exposing Major Privacy Flaw

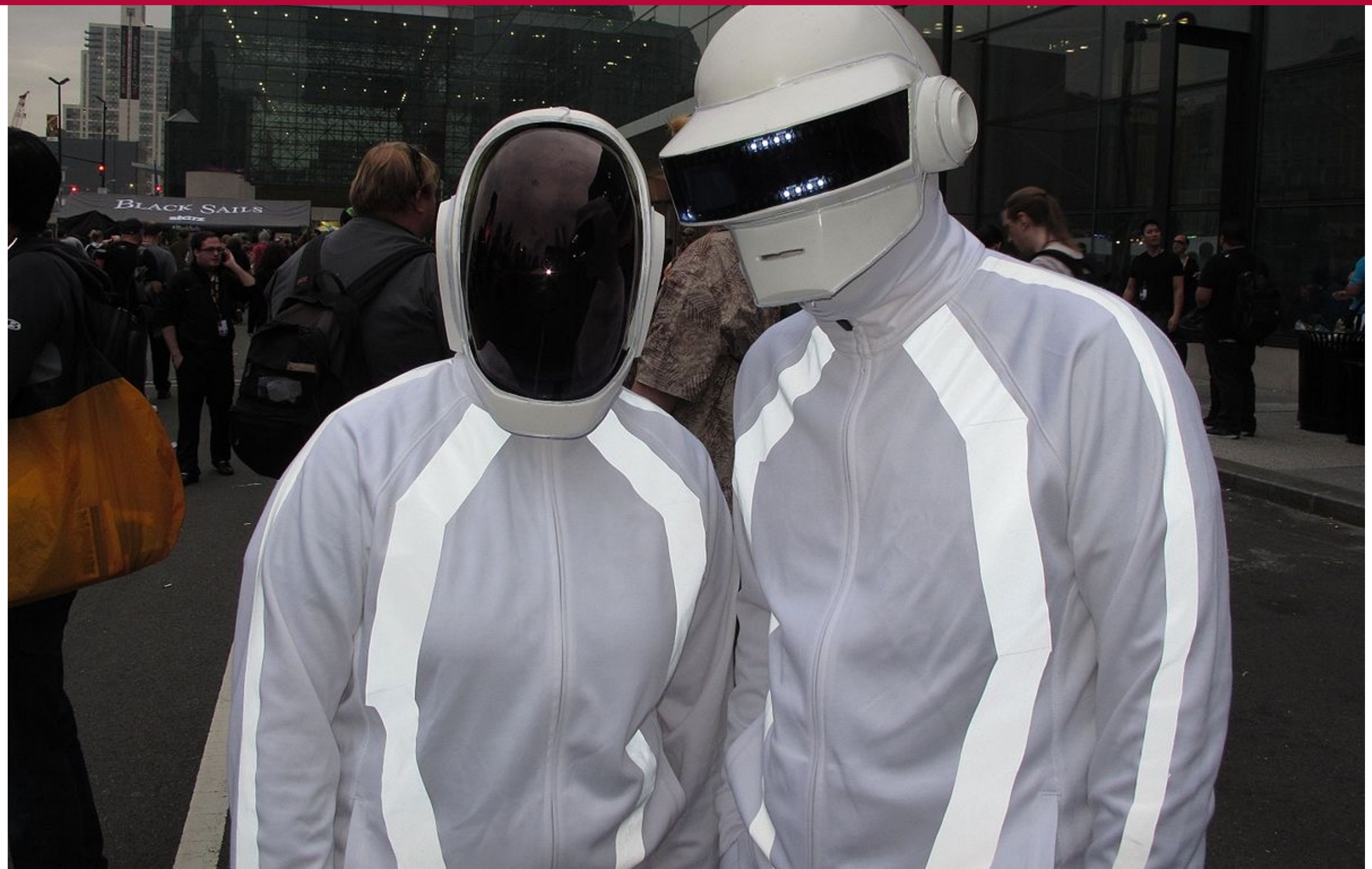
The tech giant said it receives around 20 requests for BitLocker keys a year and will provide them to governments in response to valid court orders. But companies like Apple and Meta set up their systems so such a privacy violation isn't possible.

By Thomas Brewster, Forbes Staff. Senior writer at Forbes covering cybercri...

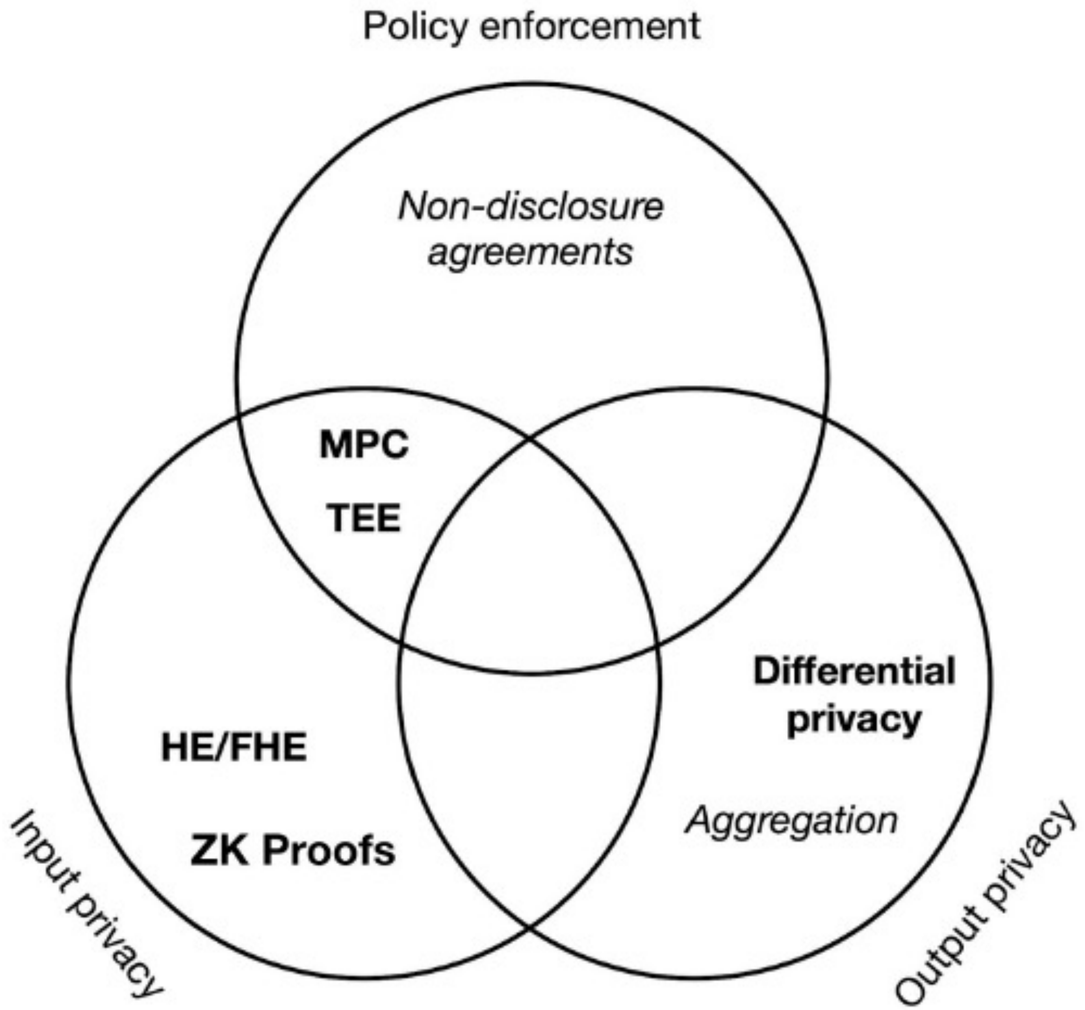
Follow Author

Published Jan 23, 2026, 06:30am EST

Private Computation



PETs can protect data from being accessed, control how it's used, or limit output disclosures.



Source: UN Handbook on Privacy-Preserving Computing Techniques

Zero Knowledge Proof: Prove something is true without revealing it.

Prover

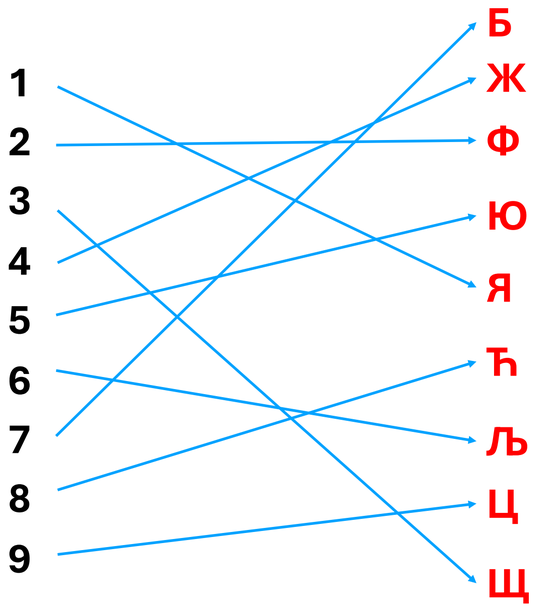
Verifier

Zero Knowledge Proofs (as told using Sudoku)

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

I randomize the symbols I use and write down the complete solution

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9



You pick any two squares that are constrained wrt each other, I reveal the random symbols

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3			7				
6			1	9	5			
	9	8					6	
8				6	Ю			3
4			8		3			1
7				2				6
	6				Ж	2	8	
			4	1	9			5
				8			7	9

Cryptographers have figured out how to do this in one step

SNARK: a Succinct ARgument of Knowledge

A **succinct preprocessing argument system** is a triple (S, P, V):

- **S**$(C) \rightarrow$ public parameters (S_p, S_v) for prover and verifier
- **P**$(S_p, \boldsymbol{x}, \boldsymbol{w}) \rightarrow$ **short** proof π ; $|\pi| = O(\log(|C|), \lambda)$
- **V**$(S_v, \boldsymbol{x}, \boldsymbol{\pi})$ **fast to verify** ; time(V) = $O(|x|, \log(|C|), \lambda)$

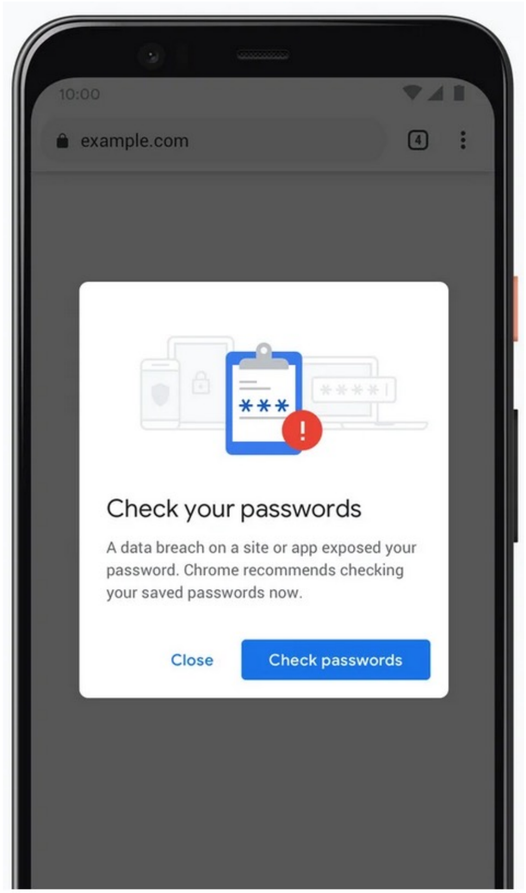
SNARK: (S, P, V) is **complete**, **knowledge sound**, and **succinct**

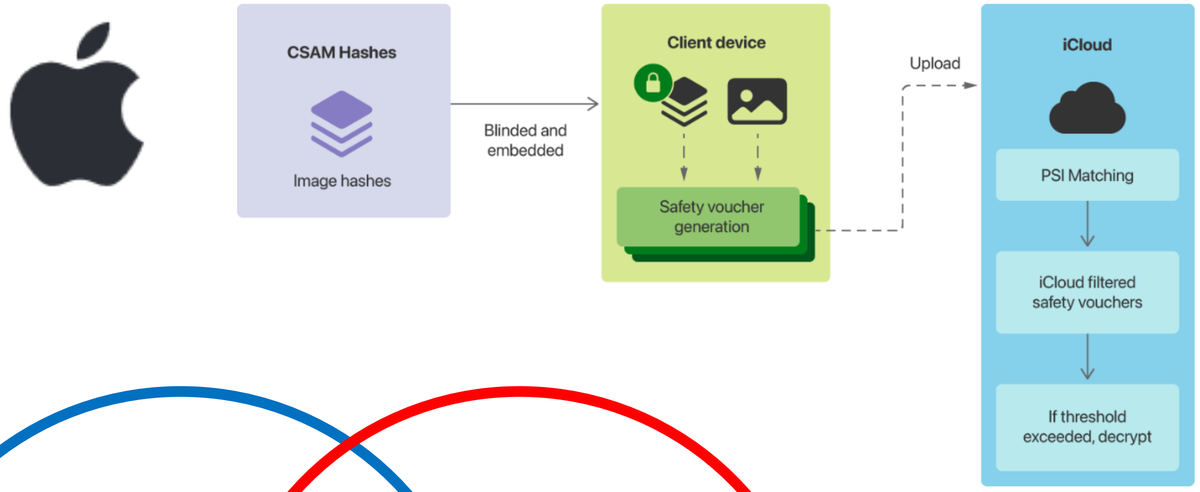
zk-SNARK: (S, P, V) is a SNARK and is **zero knowledge**

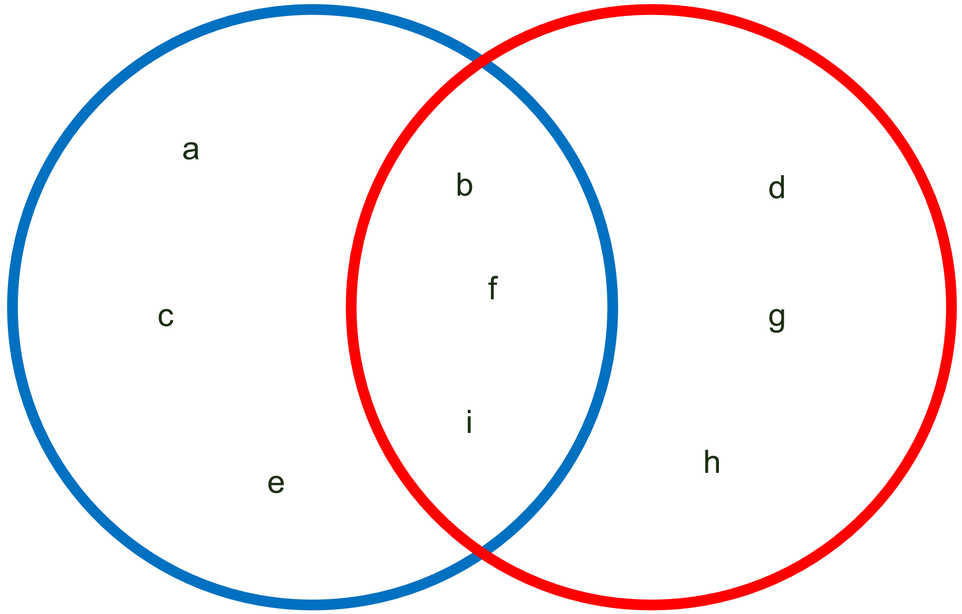
The Millionaire’s Problem

?

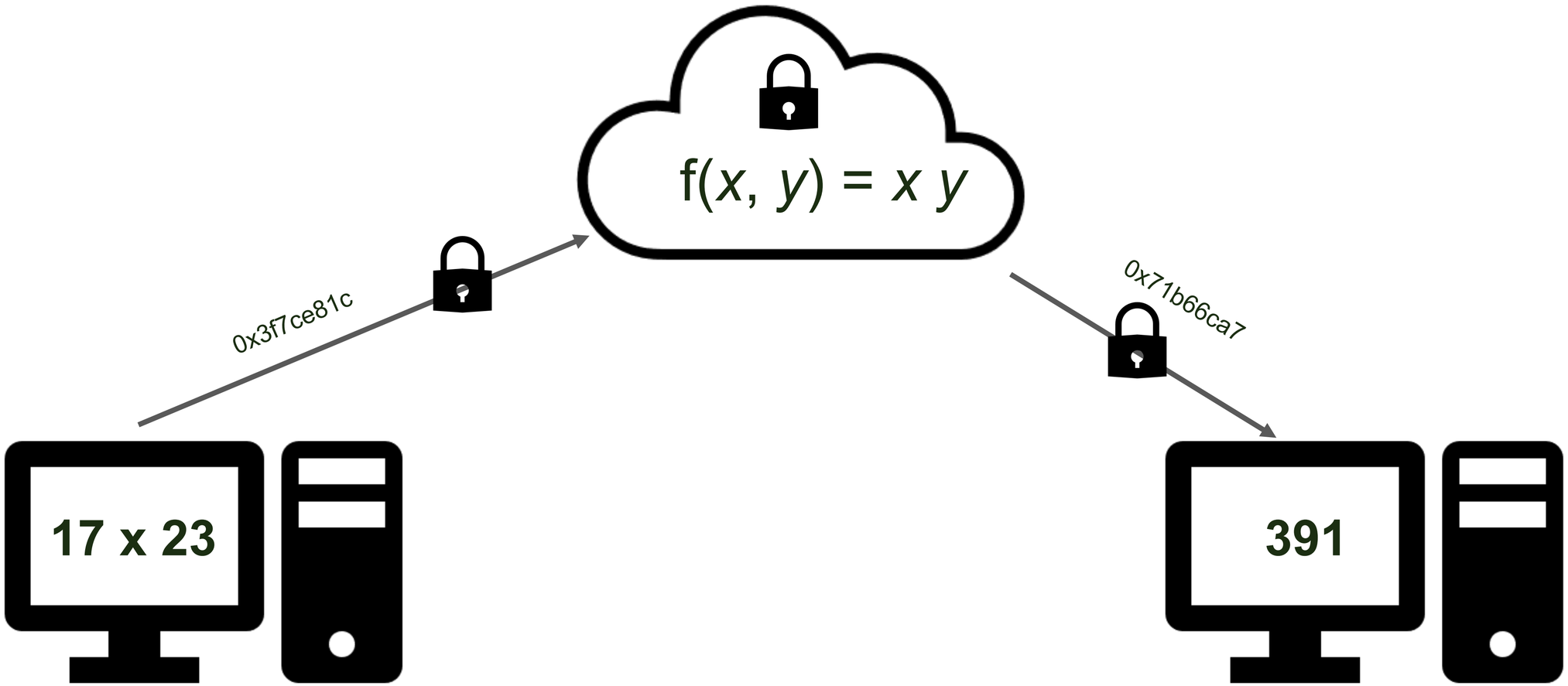
Private Set Intersection lets two parties discover overlap without revealing other members.



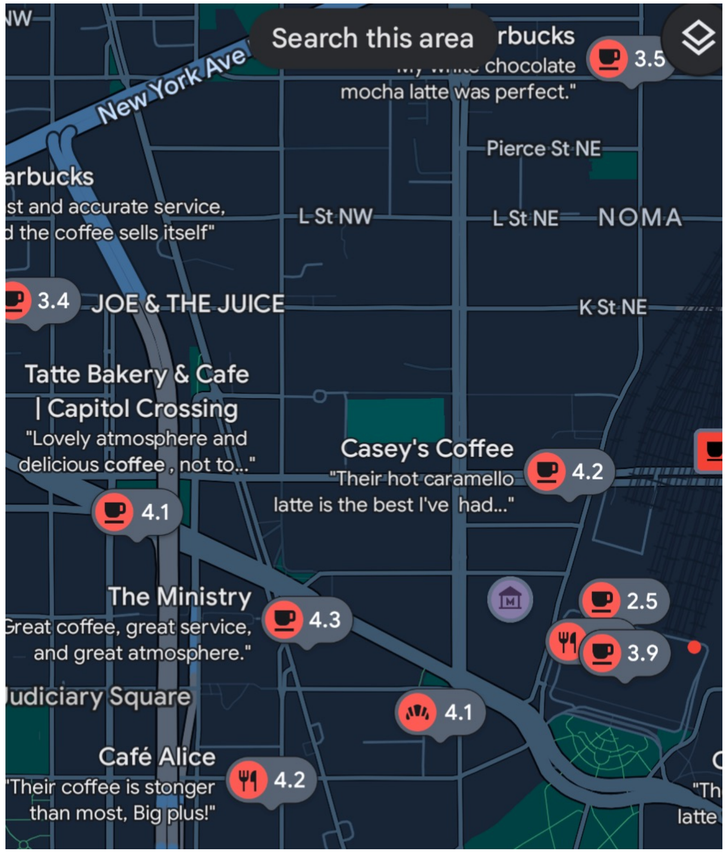


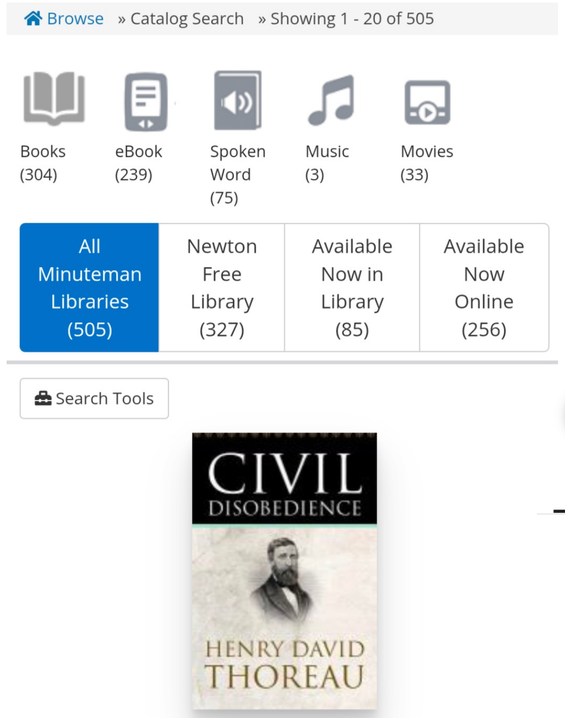


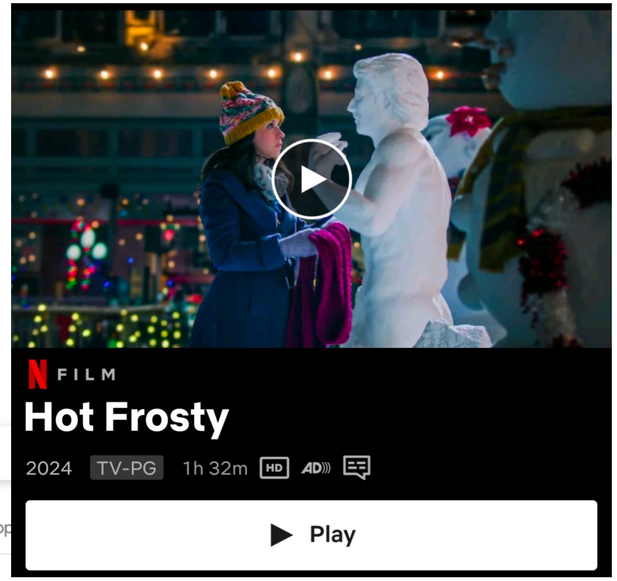
Homomorphic Encryption



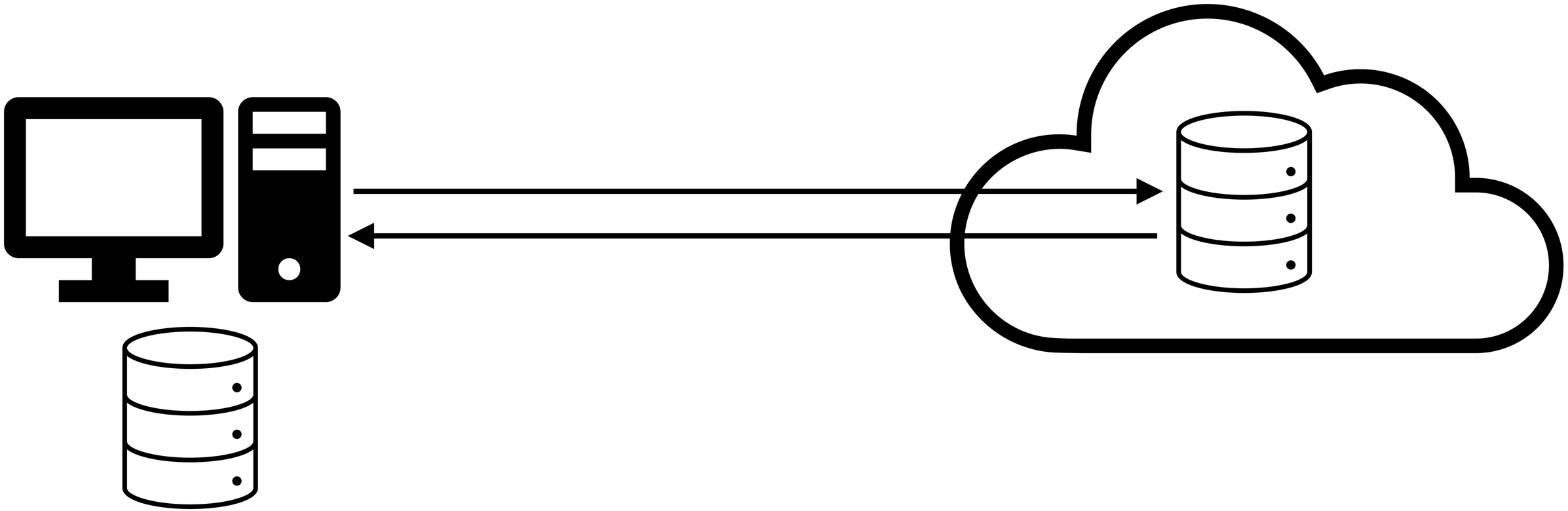
Private Information Retrieval



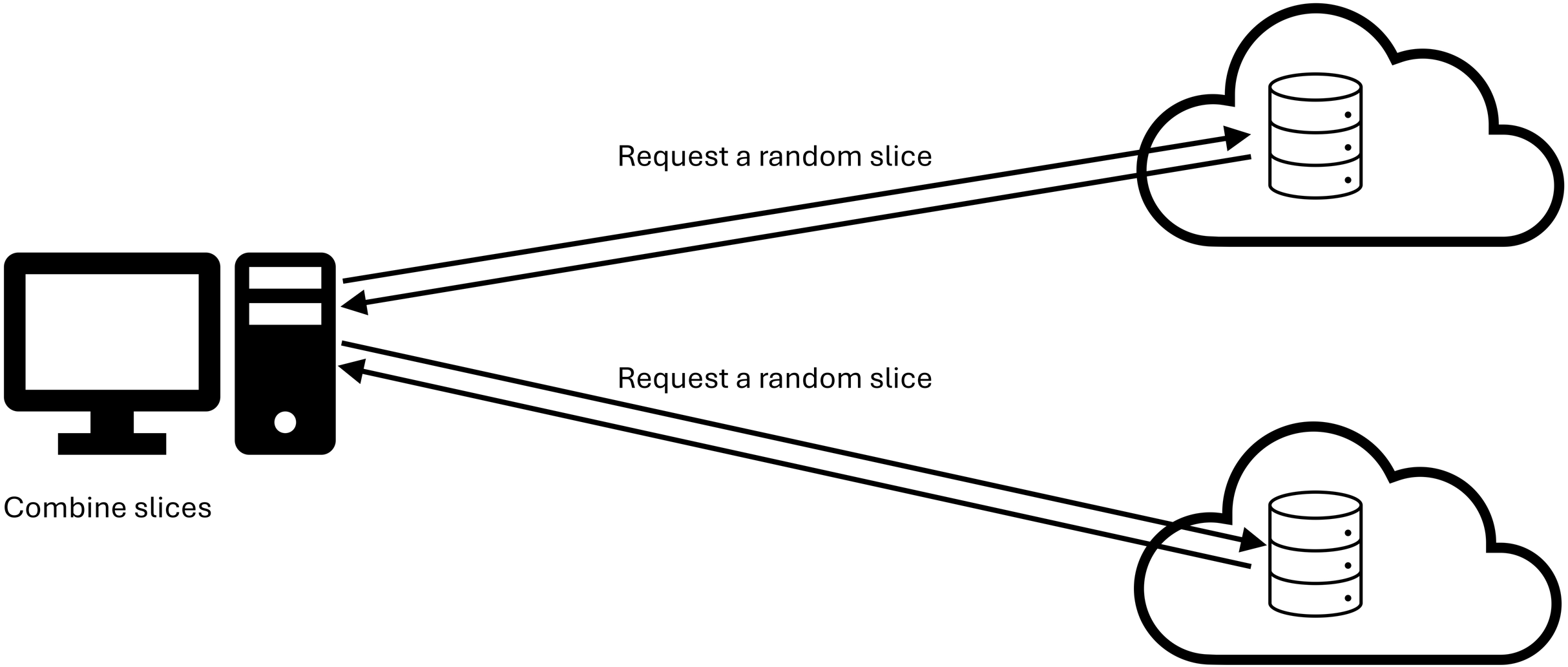




PIR the easy way!



PIR a harder way



Next Week's Paper: The Complex Path to Quantum Resistance

The Complex Path to Quantum Resistance

IS YOUR ORGANIZATION PREPARED?

ATEFEH MASHATAN AND DOUGLAS HEINTZMAN

There is a new technology on the horizon that will forever change the information security and privacy industry landscape. Quantum computing, together with quantum communication, will have many beneficial applications but will also be capable of breaking many of today's most popular cryptographic techniques that help ensure data protection—in particular, confidentiality and integrity of sensitive information. These techniques are ubiquitously embedded in today's digital fabric and implemented by many industries such as finance, health care, utilities, and the broader information communication technology (ICT) community. It is therefore imperative for ICT executives to prepare for the transition from quantum-vulnerable to quantum-resistant technologies.

This transition will be particularly complex, time-

For next week:

- Read “The Complex Path to Quantum Resistance”
  - Comment/respond on Perusall
  - Take the reading quiz
- Participate in the Ed Discussion board!
  - Share a news story, ask a question, answer a question, or continue a conversation in class
- Finish Lab 4 (Due March 9)
  - DON’T PUT IT OFF
- Spring Forward! Don’t forget Daylight Saving Time
  - Begins March 8 at 2:00 AM