

Who Moved my Rock?

Post-Quantum Cryptography and
its Impact on Higher Education

Who am I?

Brian Epstein (he/him) - bepstein@ias.edu

Institute for Advanced Study - ias.edu/security

IT Manager, Network and Security
Chief Information Security Officer

Mastodon: infosec.exchange/@ep

X: [@epepepep](https://twitter.com/epepepep)





Quantum Computers Could Crack Encryption Sooner Than Expected With New Algorithm

 By **Edd Gent** > October 2, 2023



FEATURED



Mice Just Passed the Mirror Test. Here's What That Says About Our Sense of Self

December 7, 2023

[Load more >](#)

China's new quantum code-breaking algorithm raises concerns in the US

The new algorithm could render mainstream encryption powerless within years.



Baba Tamim

Published: Jan 12, 2023 06:56 AM EST

INNOVATION



TRENDING: [Reinvent the Customer Journey With Enterprise Decisioning](#) •

Euro Security Watch with Mathew J. Schwartz

Tracking security and privacy trends across UK, Europe and beyond



[Encryption & Key Management](#), [Security Operations](#)

Researcher Claims to Crack RSA-2048 With Quantum Computer

As Ed Gerck Reads Research Paper, Security Experts Say They Want to See Proof

Mathew J. Schwartz ([@euroinfosec](#)) • November 1, 2023

GET DAILY EMAILS

Covering topics in
fraud, and inform

Email address

By submitting this
GDPR Statement

RESOURCES

FORBES > INNOVATION > ENTERPRISE TECH

Debunking The Top 5 Quantum Computing Myths

Bernard Marr Contributor ⓘ

Follow



1

Aug 23, 2023, 02:58am EDT


Member-only story

NASA Just Shut Down Quantum Computer After Something Insane Happened!

Top highlight

Houston, We Have a Problem!



The Pareto Investor  · [Follow](#)

3 min read · Nov 9



8.5K



158



Guest Post: Harvest Now, Decrypt Later? The Truth Behind This Common Quantum Theory

Insights

Jeffrey Duran • February 7, 2023





Disrupted

Microsoft, Quantinuum claim breakthrough in quantum computing

By Stephen Nellis

April 3, 2024 5:11 PM EDT · Updated 22 days ago



Aa





Paper 2024/555

Quantum Algorithms for Lattice Problems

Yilei Chen ^{ID}, Tsinghua University, Shanghai Artificial Intelligence Laboratory, Shanghai Qi Zhi Institute

Abstract

We show a polynomial time quantum algorithm for solving the learning with errors problem (LWE) with certain polynomial modulus-noise ratios. Combining with the reductions from lattice

Meta

Availa



Publica

Pr

Contact

Characters

Hem - angry, impatient, unwilling to move

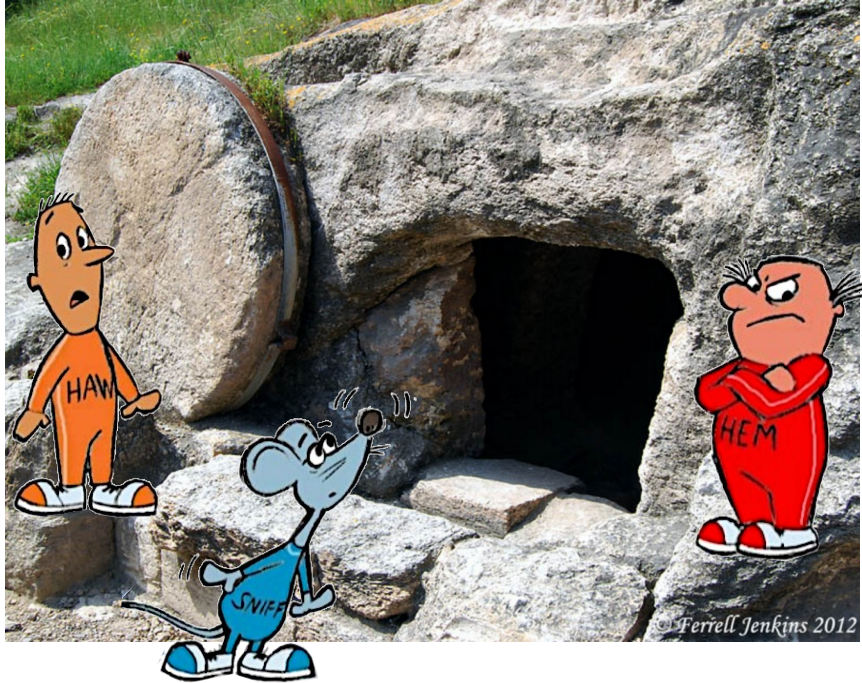
Haw - scared stiff, but thinks

Sniff - always using their tools

Scurry - always on the move to discover new things



Protecting things



First Attempt at Security

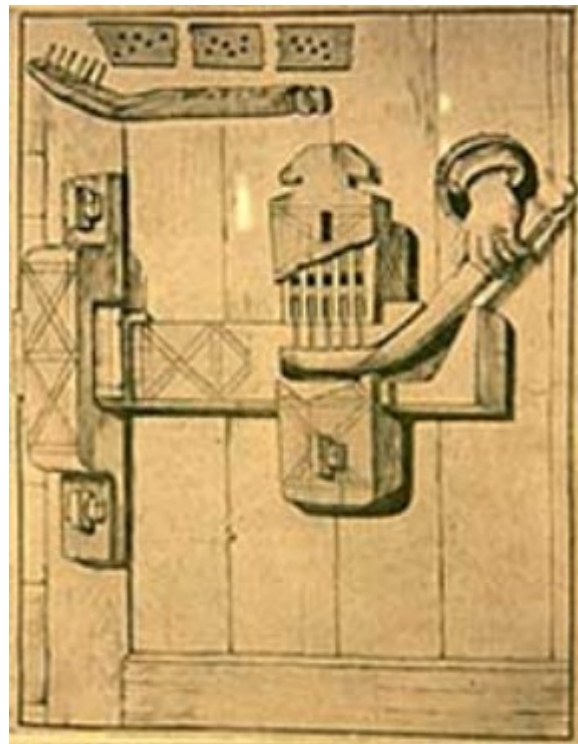
- Put large boulders in front of our caves
- Someone figured out how to move them
- Repetitive journey - find the better lock



Protectors



Locks and Lockpicks



Locks and Lockpicks



Value of things and information

- Information became valuable
 - War plans
 - Hidden treasure
 - Hunting / farming grounds
- Necessary to transmit
 - Get strategy to front lines
 - Remote allies
- Easy to intercept

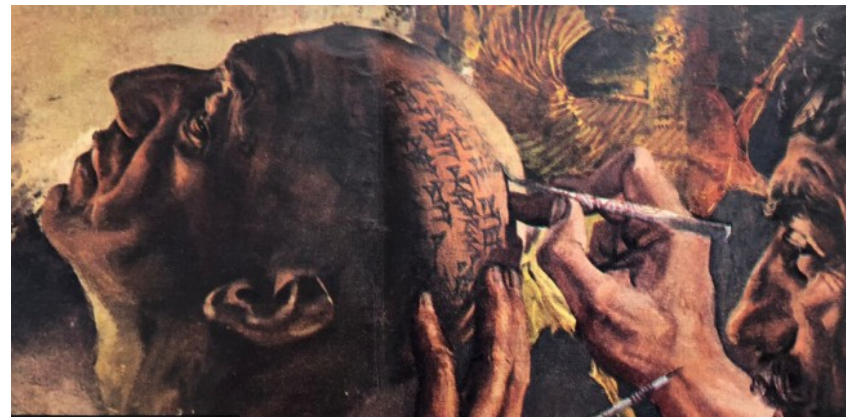


Hiding info in plain sight

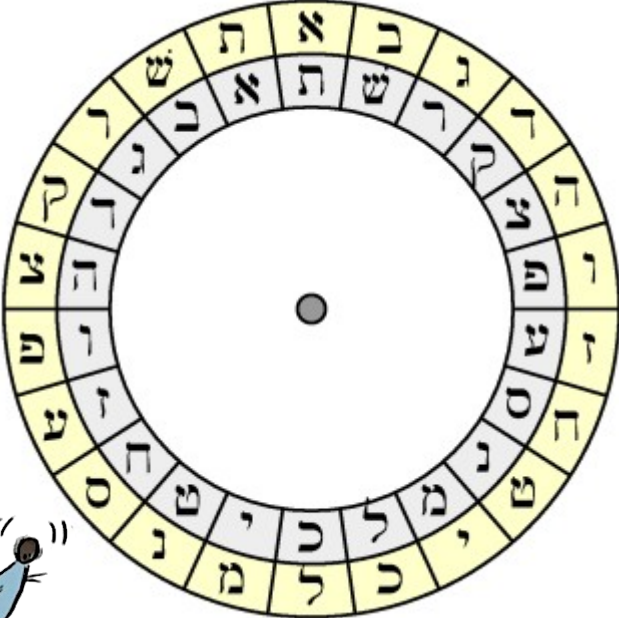
- Novel schemes
- Useless when discovered
- Secrets can be bought



Hiding info in plain sight



Encoding

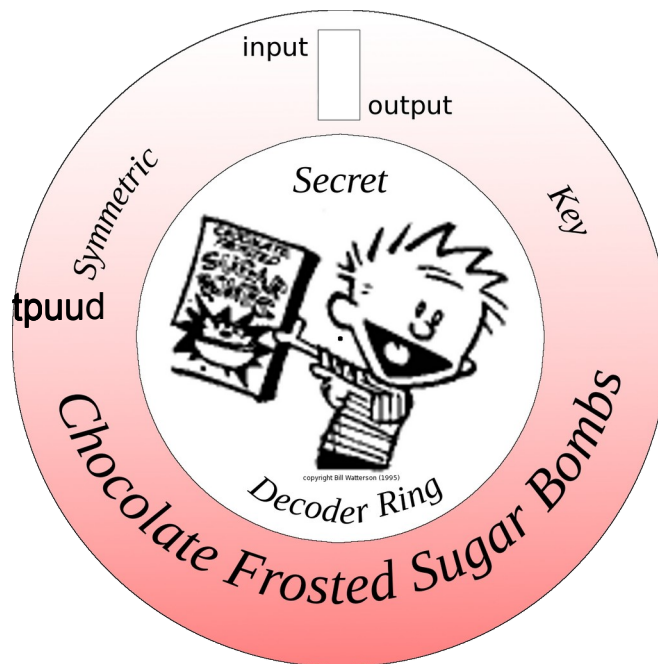


Symmetric Key Cryptography

- Single key for both encryption and decryption
- Simple
- Powerful
- Key distribution issues
- Number of keys difficult

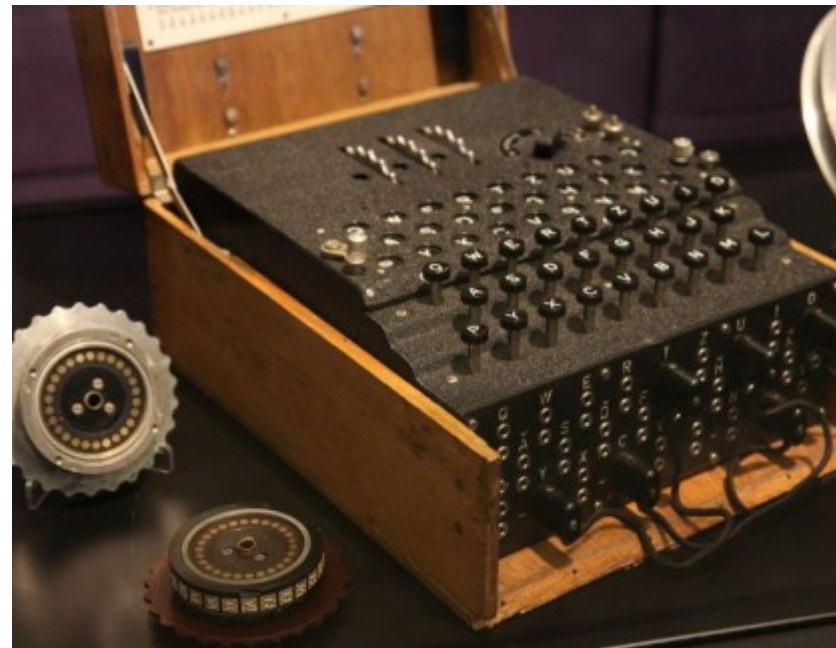
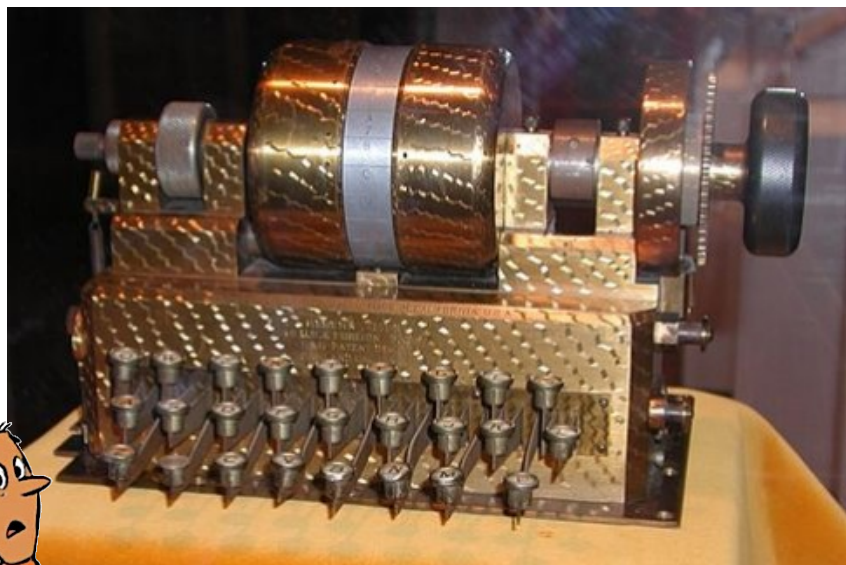
Symmetric Key Cryptography

tpuud

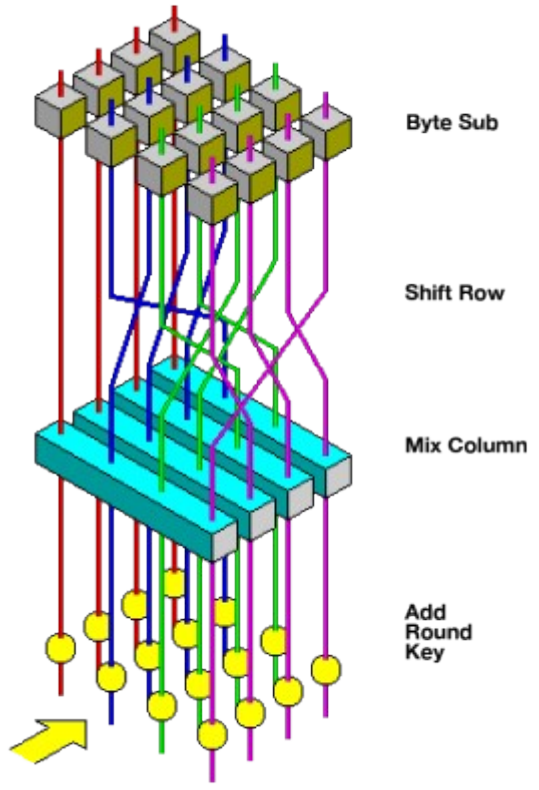
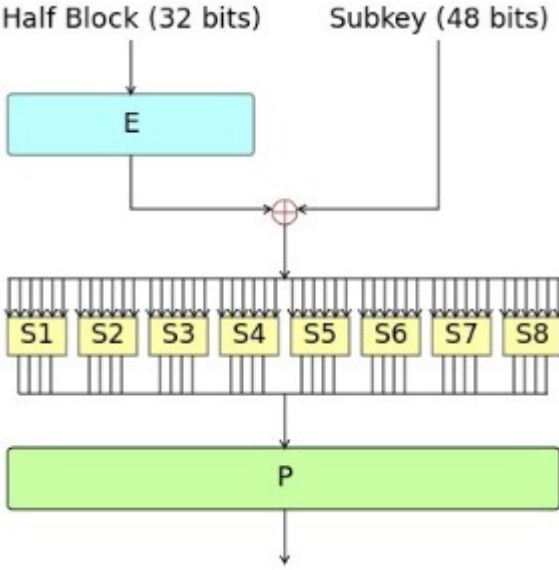


Jimmy

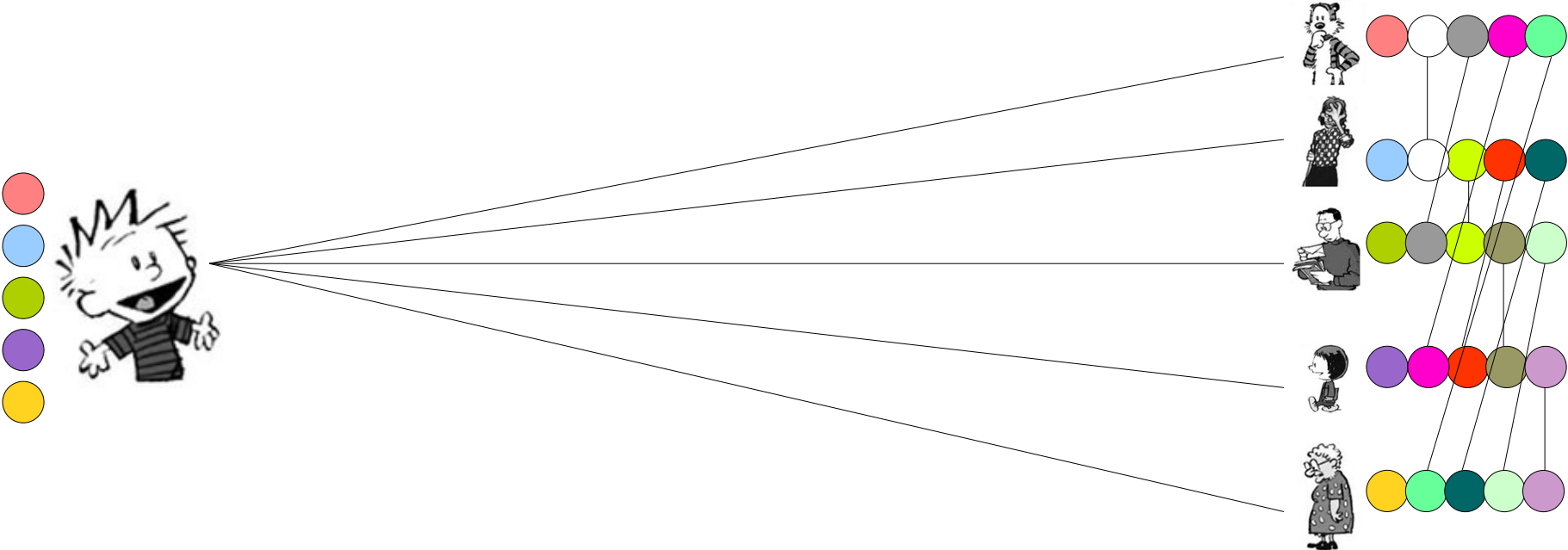
Symmetric Key Cryptography



Symmetric Key Cryptography



Symmetric Key Cryptography

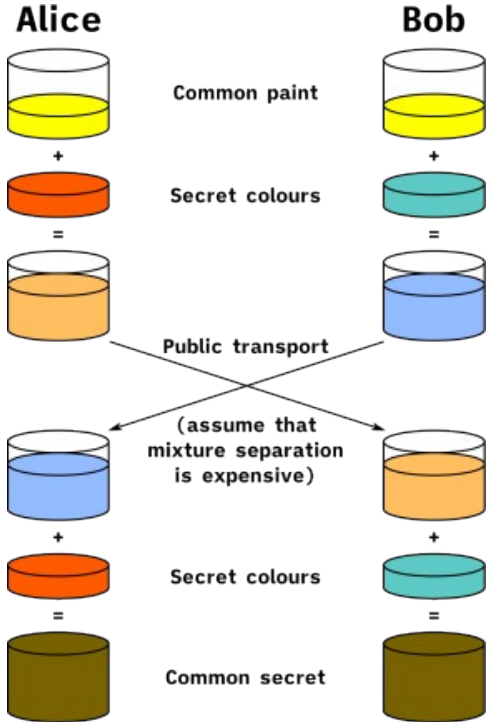


$$n*(n-1)/2 = 6*(5-1)/2 = 30/2 = 15 \text{ unique keys}$$



Key Exchange

- Generate a unique secret key
- Never transfer it over the wire

Key Exchange (Diffie-Hellman)



Public Key Cryptography/ECDH

 <i>Bob picks</i> $\beta = 7$ <i>Bob computes</i> $B = 7G = (4,7)$ <i>Bob Receives</i> $A = (6,7)$ <i>Bob computes</i> $\beta A = 7(5G) = 35G = 17G = (7,10)$	<i>Common parameters</i> $y^2 = (x^3 + 2 \cdot x + 3) \text{ mod } 13$ $+ N = 18$ $G = (7,3)$ $B = (4,7)$ $A = (6,7)$	 <i>Alice picks</i> $\alpha = 5$ <i>Alice computes</i> $A = 5G = (6,7)$ <i>Alice Receives</i> $B = (4,7)$ <i>Alice computes</i> $\alpha B = 5(7G) = 35G = 17G = (7,10)$
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



Asymmetric Key Cryptography

- Split public and private keys
- Key distribution easier
- Number of keys manageable
- Slower
- Authentication issues
- Relies on a one-way function (Theoretical Math)

One way function



January 23, 2015

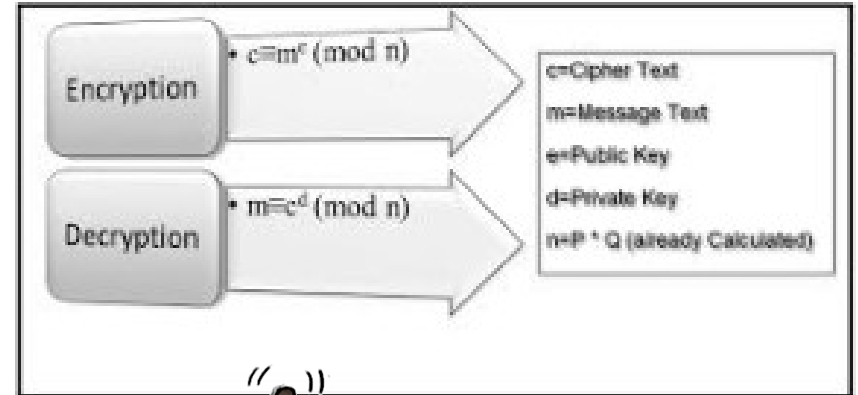
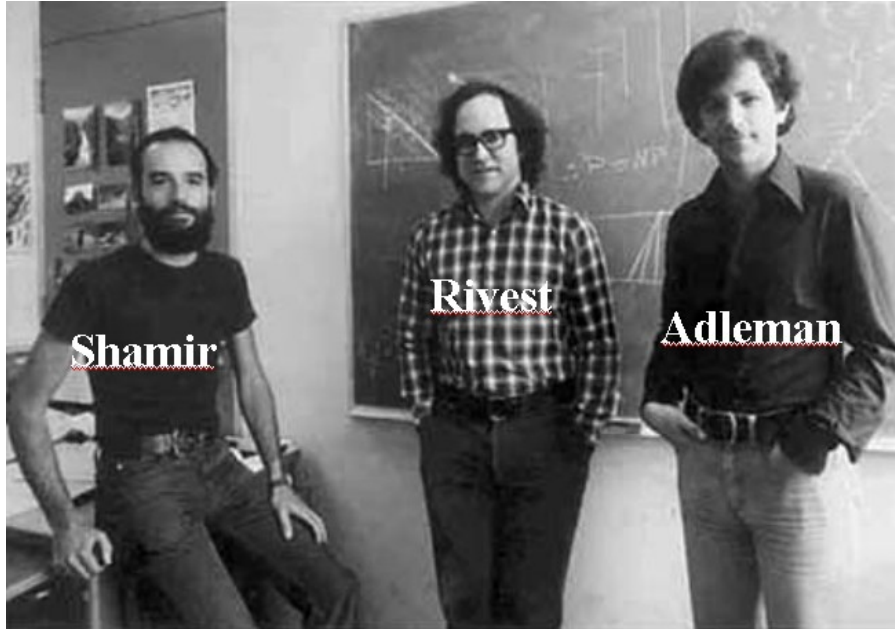
UCI, fellow chemists find a way to unboil eggs

Share

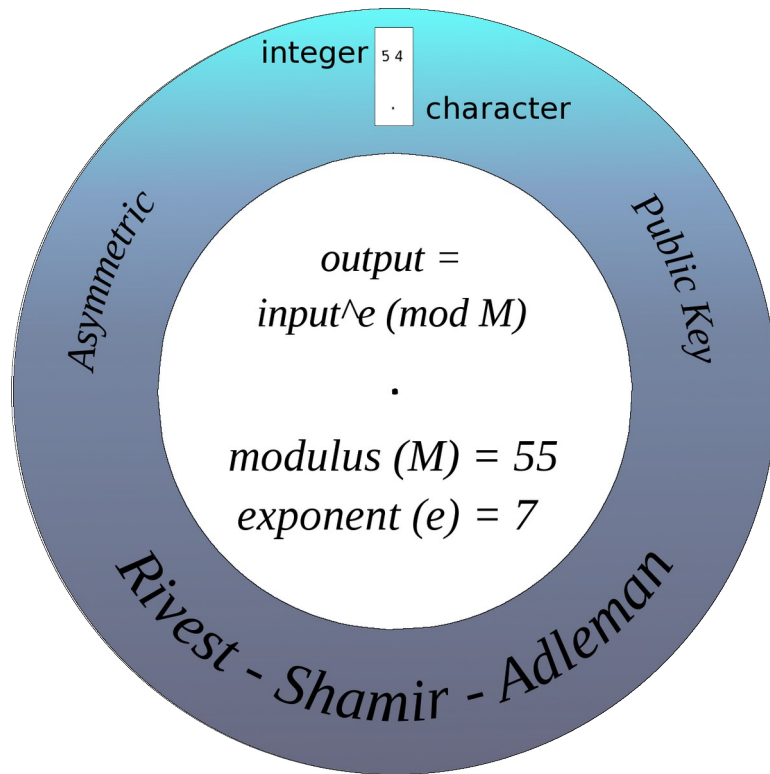


Irvine, Calif., Jan. 23, 2015 – UC Irvine and Australian chemists have figured out how to unboil egg whites – an innovation that could dramatically reduce

RSA



One way function for RSA



Jimmy

14 13 18 18 36

One way function for RSA

14 13 18 18 36

14 13 18 18 36

One way function for RSA

~~14~~ 13 18 18 36

13

18

18

36

One way function for RSA

$$14^7 \bmod 55 = 105413504 \bmod 55 = 9 = \text{"g"}$$

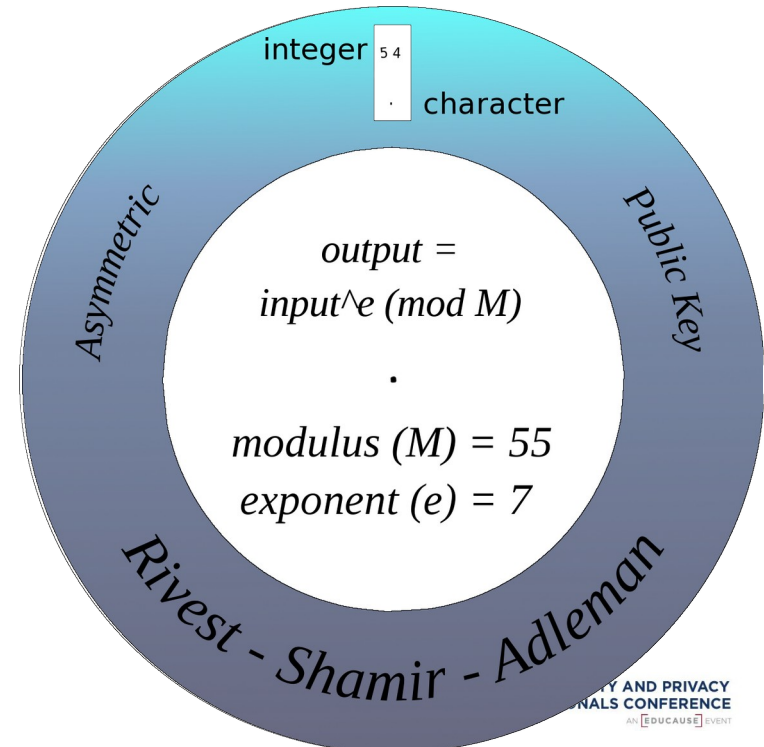
$$13^7 \bmod 55 = 7 = \text{"e"}$$

$$18^7 \bmod 55 = 17 = \text{"l"}$$

$$18^7 \bmod 55 = 17 = \text{"l"}$$

$$36^7 \bmod 55 = 31 = \text{"v"}$$

$$E(\text{Jimmy}) = \text{gellv}$$



One way function for RSA

$$9^{23} \bmod 55 = 14 = \text{"J"}$$

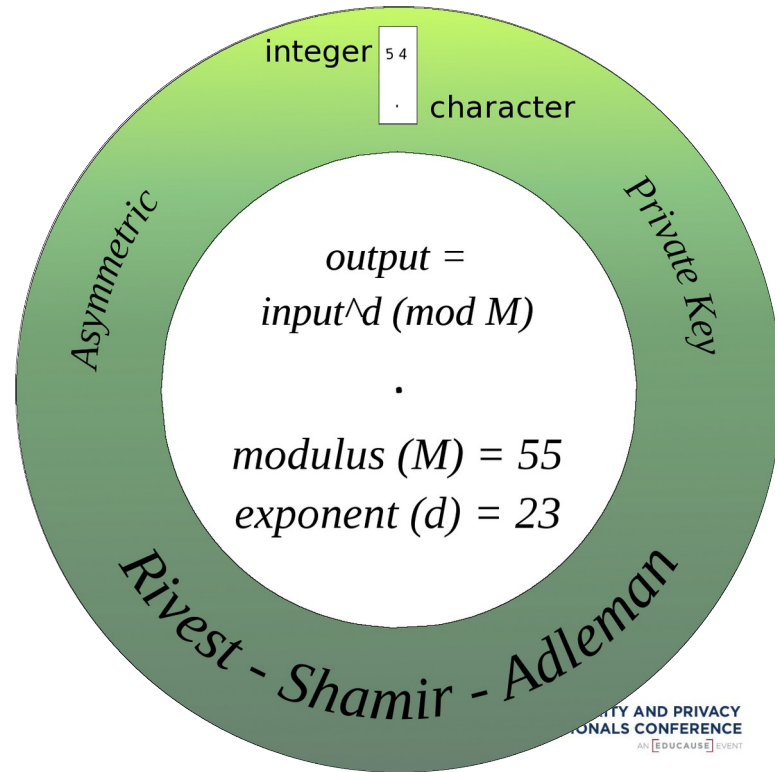
$$7^{23} \bmod 55 = 13 = \text{"i"}$$

$$17^{23} \bmod 55 = 18 = \text{"m"}$$

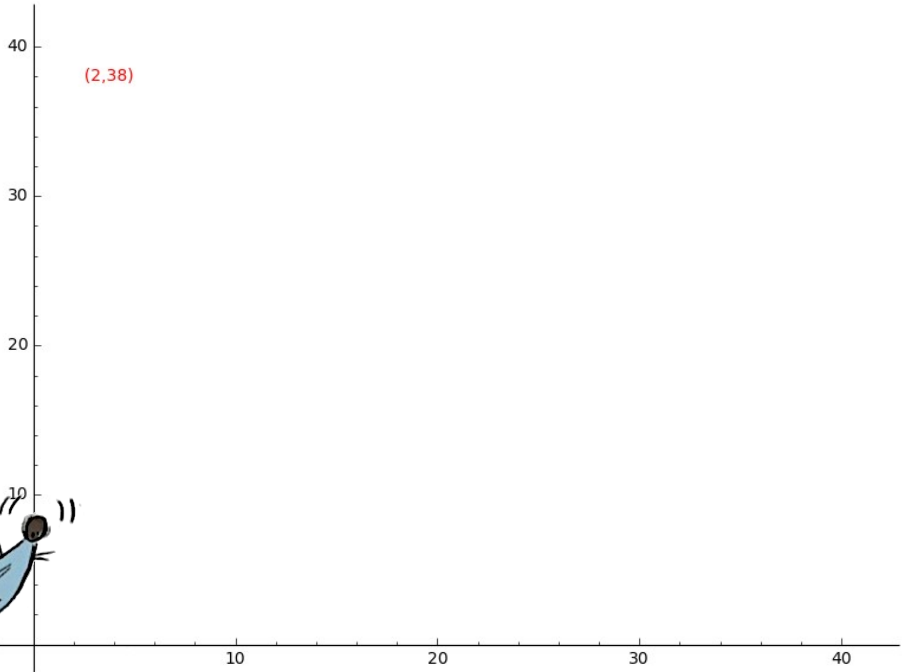
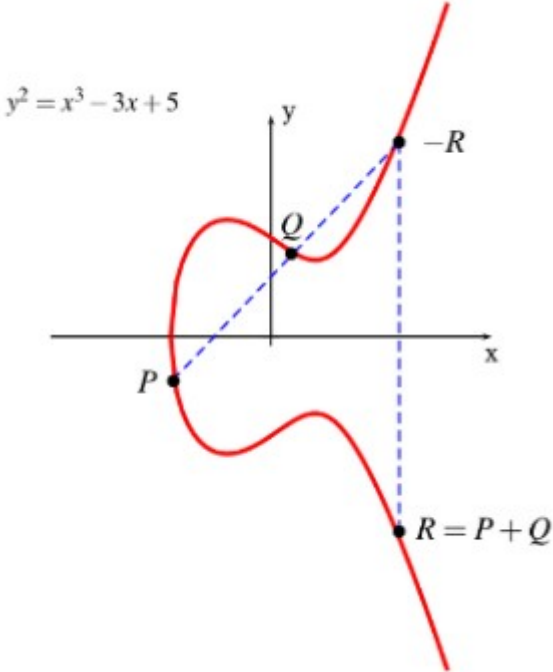
$$17^{23} \bmod 55 = 18 = \text{"m"}$$

$$31^{23} \bmod 55 = 36 = \text{"y"}$$

D(gellv) = Jimmy

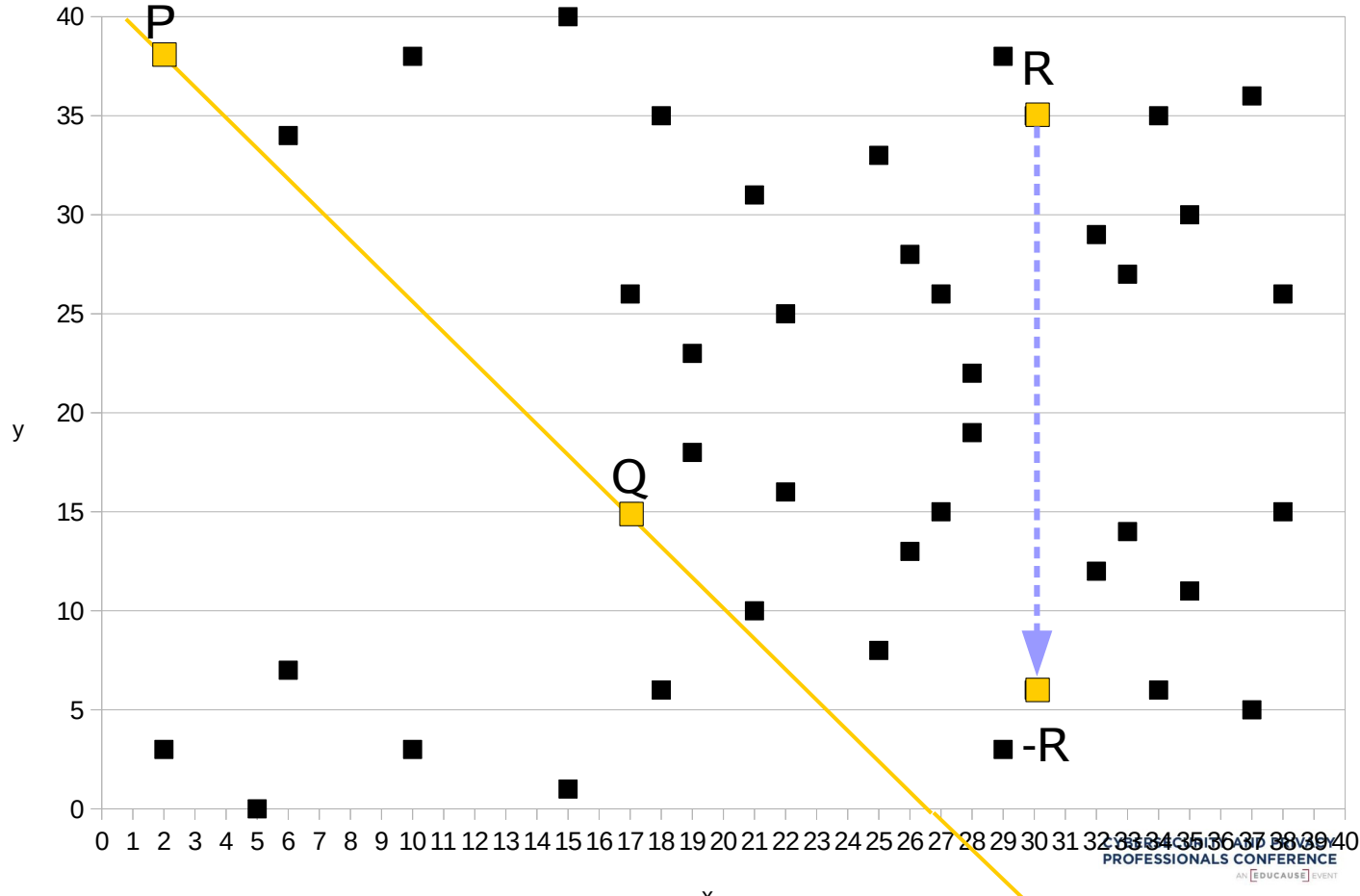


Elliptic Curve Cryptography



$$x \quad | \quad y^2 \pmod{41} \equiv x^3 - x + 3 \pmod{41}$$

2	3; 38
5	0
6	7; 34
10	3; 38
15	1; 40
17	15; 26
18	6; 35
...	...



$$P \oplus Q \oplus R = 0$$

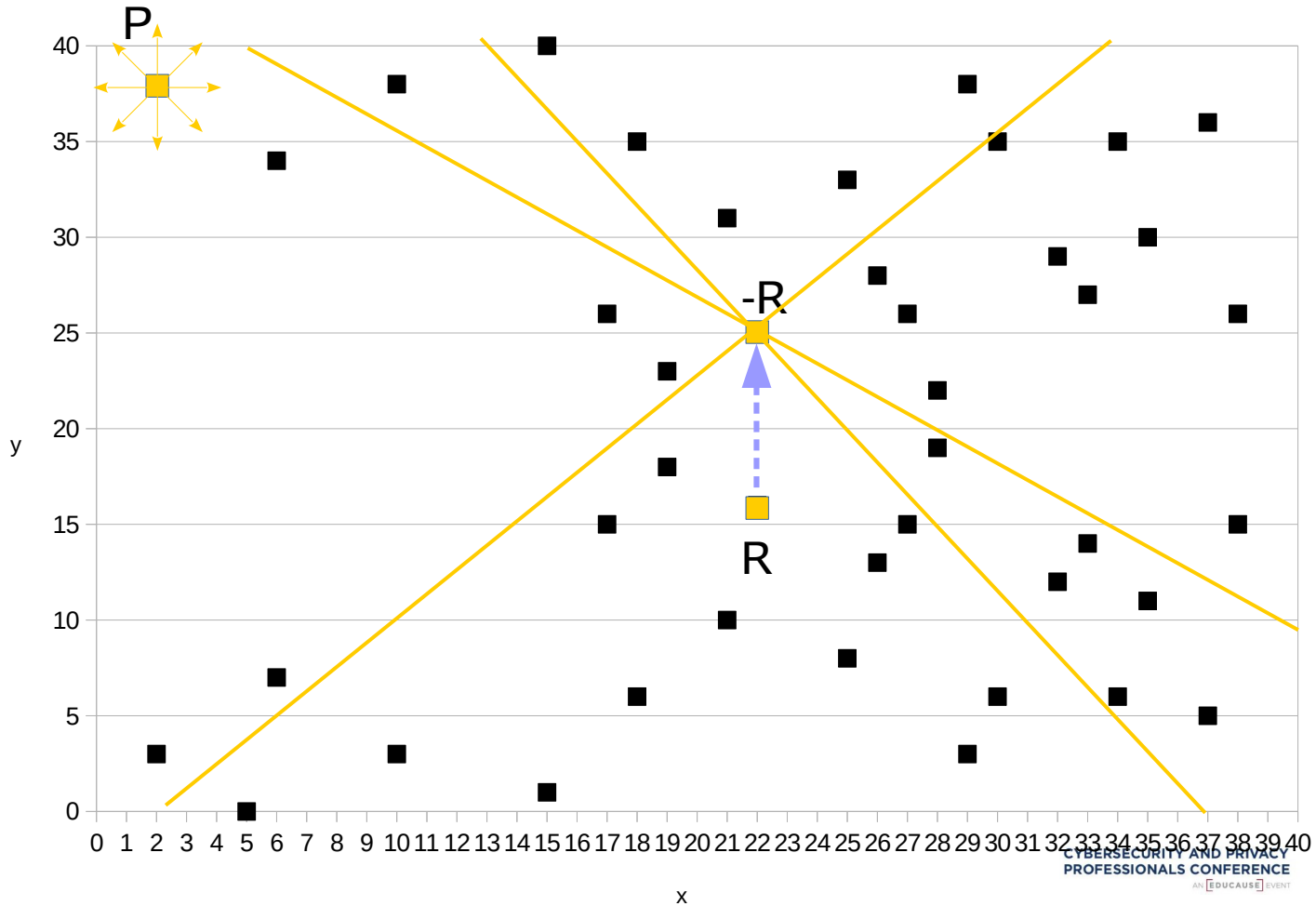
$$P \oplus Q = -R$$

$$x \quad | \quad y^2 \pmod{41} \equiv x^3 - x + 3 \pmod{41}$$

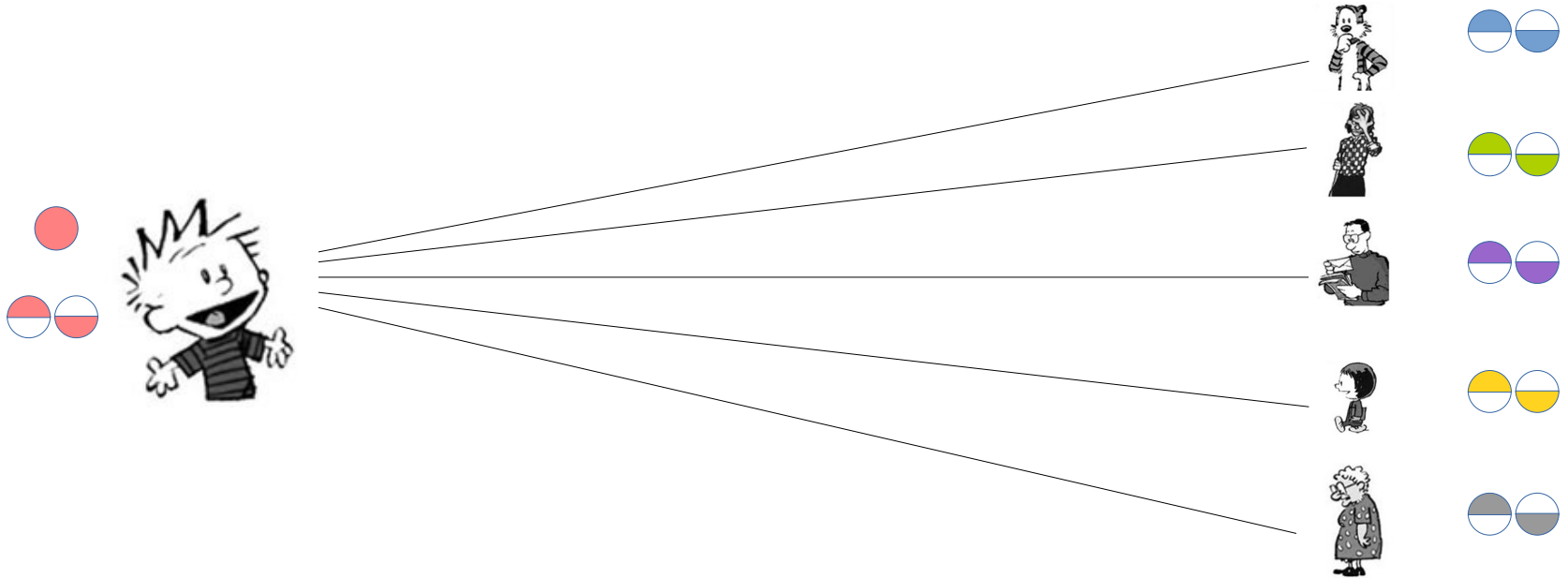
2	3; 38
5	0
6	7; 34
10	3; 38
15	1; 40
17	15; 26
18	6; 35
...	...

$$P + Q = -R$$

$$P + Q + R = 0$$



Asymmetric Key Cryptography



$$2 * n = 2 * 6 = 12 \text{ unique keys}$$

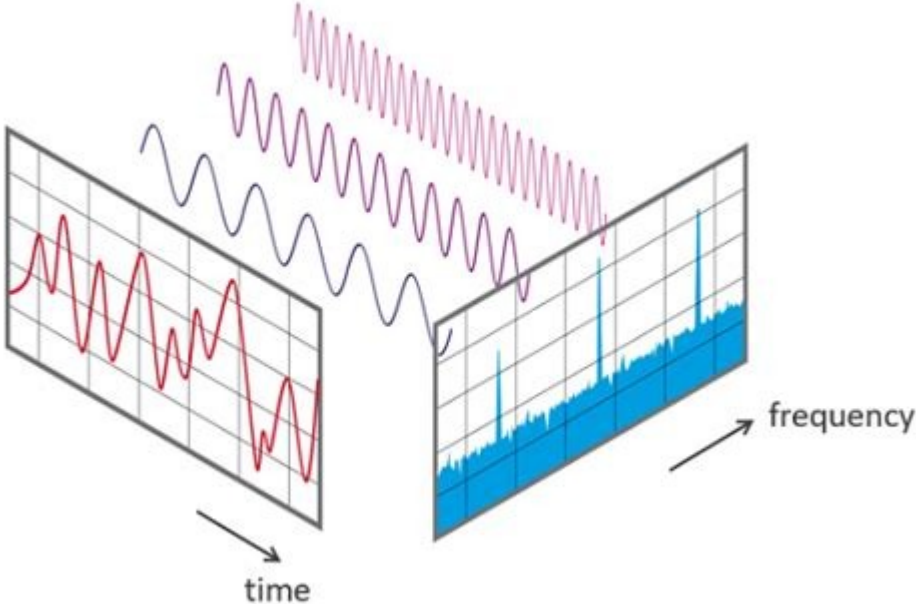
Quantum Computers

- Use qubits instead of classical bits
- Qubits can be in more than one state at the same time
- The state of a qubit is unknown until you observe it
- Qubits are fragile and interference can put them into an error state
- Error rates slow down the development of quantum computers

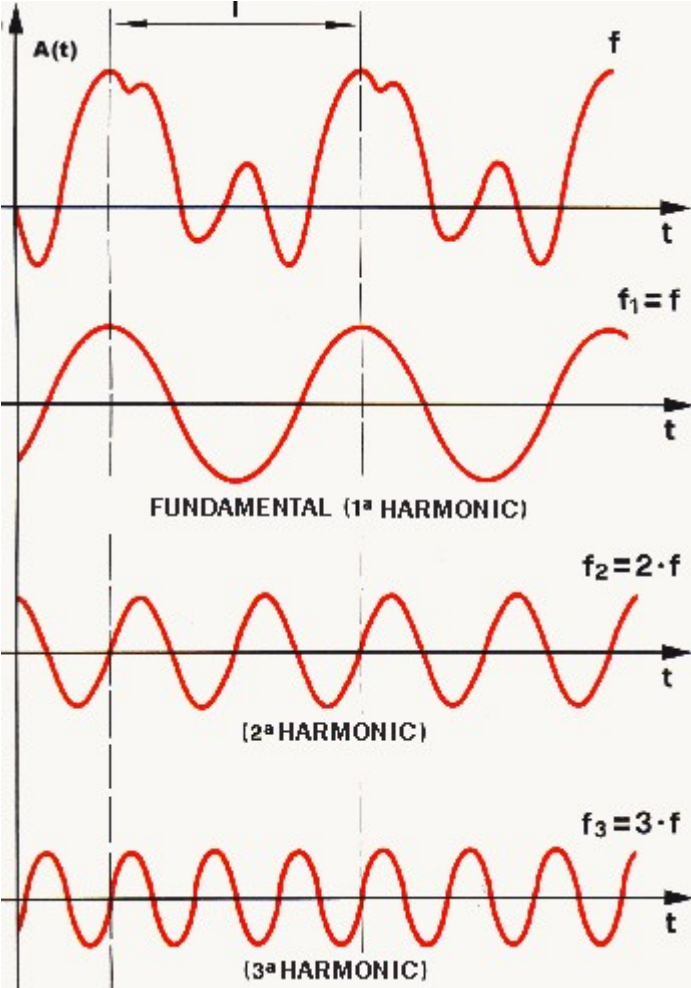
Shor's Algorithm

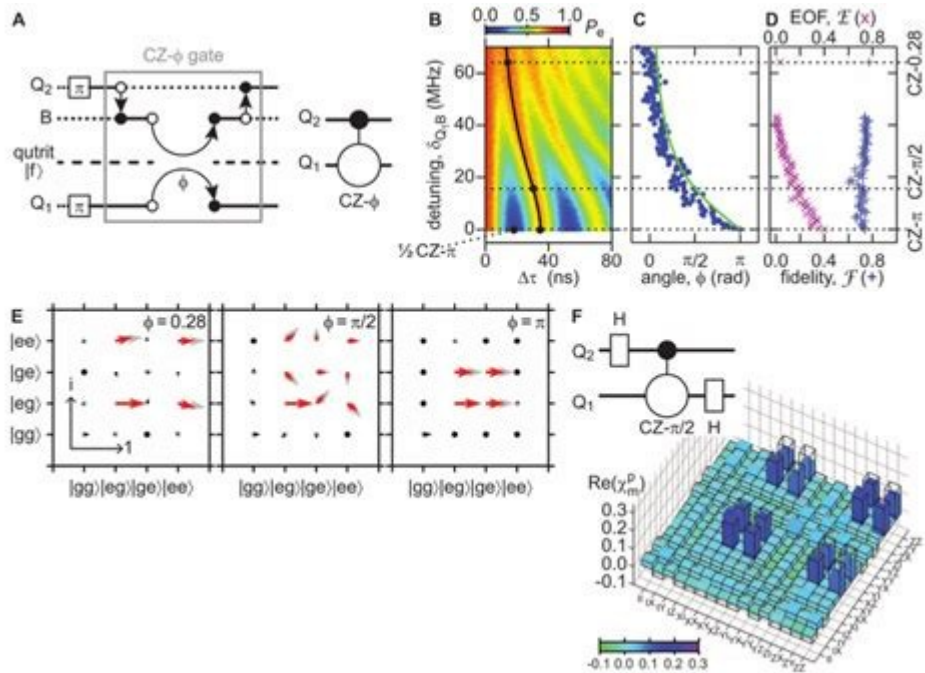
- Makes factoring large semiprimes obtainable
- Another algorithm for solving the discrete logarithm problem
- And yet another for the period finding problem
- With proper quantum computer, RSA, DH, ECC, ECDH all become obsolete

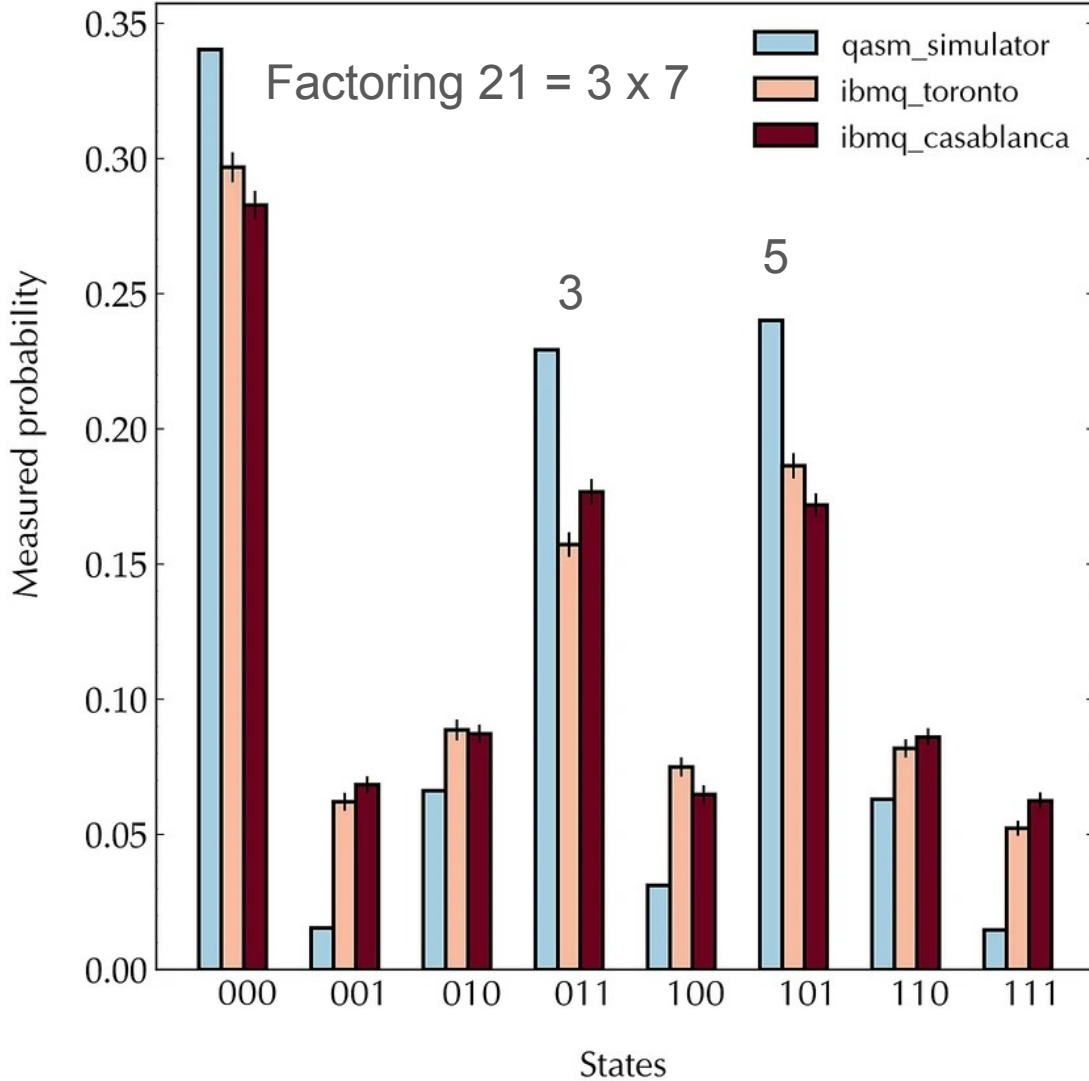
Fourier Transform



Fourier Transform







GENNISQ

noisy intermediate-scale quantum era



GENISQ

noisy intermediate-scale quantum era

not that worried, bra



Privacy24



CYBERSECURITY
PROFESSIONAL
CONFERENCE
EVENT

1977

Rivest, Shamir, and Adleman designed the RSA algorithm which uses the idea that prime factorization is difficult.

2001

IBM shows that Shor's algorithm can work on a quantum computer with 7 qubits. It was able to factor 15 into 3 x 5.

2012

NIST formally begins the Post Quantum Cryptography Project.

Prime factorization of 21 into 3 x 7 was achieved.



Peter Shor designs an algorithm that can factor integers into its prime counterparts quickly by utilizing a quantum computer with a large number of qubits.

1994

NIST publishes a survey for protocol designers. The survey says, "it does not appear inevitable that quantum computing will end cryptographic security as we know it."

2009

NIST h...
Wo...
NSA s...
transitioni...
resistant a...
the not too...

2

begins the
sum
project.
on of 21
chieved.

2016

NIST publishes 1st report on PQC (8105). Deadlines for submissions of round one due in 2017.

2021 & 2022

NIST holds 3rd & 4th PQC Workshop.

Draft standards made, algorithms selected for QSC (CRYSTALS-KYBER, CRYSTALS-DILITHIUM, FALCON, SPHINCS+).



NIST holds 1st PQC Workshop.

NSA states that transitioning to quantum resistant algorithms is in the not too distant future.

NIST holds 2nd PQC Workshop.

Attempt to factor 35 into primes fails due to accumulating errors.

NIST announced that it will integrate the four selected post-quantum algorithms into U.S. encryption standards

2015

2019

2024

1977

Rivest, Shamir, and Adleman designed the RSA algorithm which uses the idea that prime factorization is difficult.

2001

IBM shows that Shor's algorithm can work on a quantum computer with 7 qubits. It was able to factor 15 into 3 x 5.

2012

NIST formally begins the Post Quantum Cryptography Project.

Prime factorization of 21 into 3 x 7 was achieved.



Peter Shor designs an algorithm that can factor integers into its prime counterparts quickly by utilizing a quantum computer with a large number of qubits.

1994

NIST publishes a survey for protocol designers. The survey says, "it does not appear inevitable that quantum computing will end cryptographic security as we know it."

2009

NIST h...
Wo...
NSA s...
transitioni...
resistant a...
the not too...

2

begins the
sum
project.
on of 21
chieved.

2016

NIST publishes 1st report on PQC (8105). Deadlines for submissions of round one due in 2017.

2021 & 2022

NIST holds 3rd & 4th PQC Workshop.

Draft standards made, algorithms selected for QSC (CRYSTALS-KYBER, CRYSTALS-DILITHIUM, FALCON, SPHINCS+).



NIST holds 1st PQC Workshop.

NSA states that transitioning to quantum resistant algorithms is in the not too distant future.

NIST holds 2nd PQC Workshop.

Attempt to factor 35 into primes fails due to accumulating errors.

NIST announced that it will integrate the four selected post-quantum algorithms into U.S. encryption standards

2015

2019

2024

Quantum Factored

4 bit semiprime (2001): 15

5 bit semiprime (2012): 21

6 bit semiprime (2019 - failed): 35

Need to factor

1024 bit semiprime:

14858031842529041742675223662020065615490358969220662
89933239251279096441074379066840500275518447762785965
12160601382625659982180758999544221868254722043619979
84526745656869867322663775380667065617182042243040564
49791211661181323805868000257522596407301217381568222
96246476504443847811940638639921190244907229

Need to factor

2048 bit semiprime:

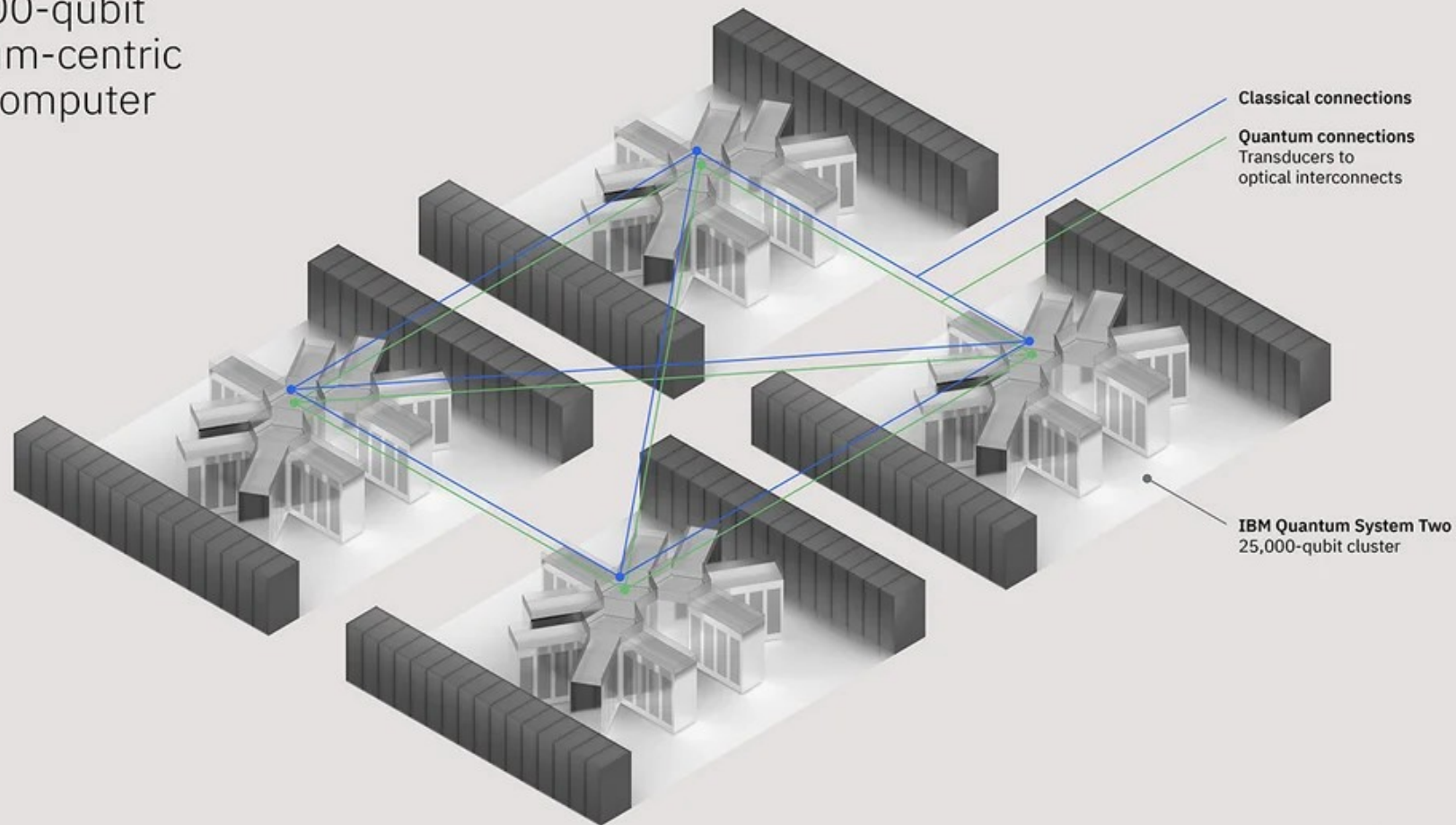
23042155144807033264822777505847352979234760665383921887127664352
78228085219412095936558643453832808918618527312570608206987897806
778266867076836343848261613715911228213968789316742345746645880253
12480684919203629910806547215276202762168939550018927857695365726
726743981630638931221130356279390701114143002852846597092746910840
713994794061711394675366463723772973275703764980539097704296549591
25201782358647860679882638386416987498248726270464804391684357189
38652132696815444783098028151462344977286590749544794608585623582
763434406818849621131943279250579658150209579438838292900340862945
2106790235491702341492875849

Need to factor

4096 bit semiprime:

8950429693485782969819159103361636787324531247595331455865576049528957603177763165691586761508274
8612428210789254796713033291402777424383964474525675340059592677596345495367727186879078511383357
66406911679216800465068765481022358269511066011043214241154792284849342476806301462838794988489104
4979706015770085049633781304347756129377783540108661429560151911005773387375216115635302836880772
7894927096506522360815072003186182464302107329240801651864447332432227480806121176546660163477961
2079087301378180468550758603979708720992301281518897296205081608907825465936077824761157221996661
9098316259052074129972637777508871680607010649649306287447241547902349676376787491756994866954495
0361493264214164541577312583731983304075229287865529835971053244670543287945198597007230421194138
5987622009399852416597534382289605254398863710160238544899455274270168012833272387402646230414545
4385691054574768115474820051683248135374589610152876639436419219968226769548413114519425094561652
9523978521590091466213527524706661778437750247039074736641799486281849406242236762484545429225759
1818867787433314406963341724404237652561568658515029294655137151461974536474688780183636286034894
24424030802769700952150094244596873055650438391281230331589881854277

100,000-qubit
quantum-centric
supercomputer
—
2033

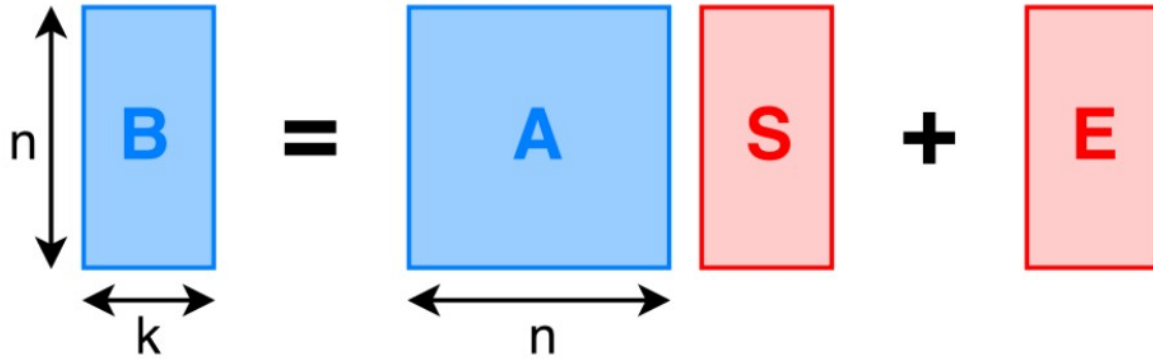


Post-Quantum Cryptography (PQC) - where are we?

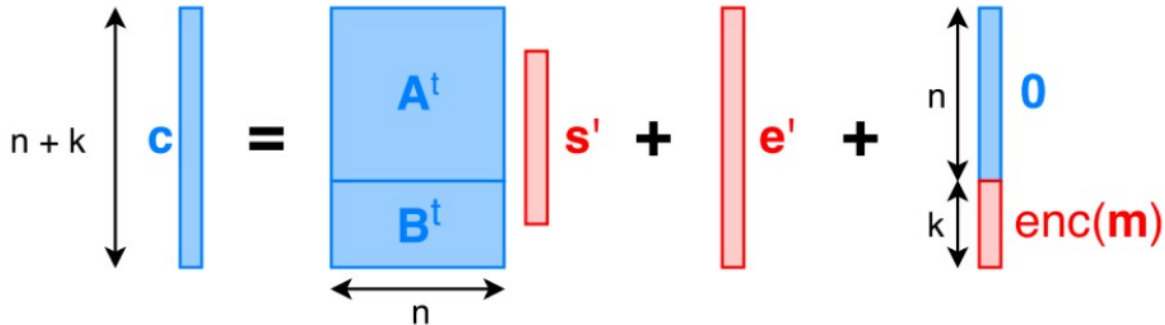
2022 NIST approves PQC encryption and signature algorithms

- CRYSTALS-Kyber (general encryption)
- CRYSTALS-Dilithium (signature)
- FALCON (signatures for smaller applications)
- SPHINCS+ (backup based on a different mathematical model)

Public parameter $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n \times n})$
KeyGen()



Encrypt_b($m \in \{0, 1\}$)



Post-Quantum Cryptography (PQC) - where are we?

- X25519Kyber768Draft00 readily available to test
 - Google Chrome
 - Firefox
 - BoringSSL
 - Nginx
- Cloudflare Research - <https://pq.cloudflare.com/research/>
- Signal - <https://signal.org/blog/pqxdh/>
- Apple iMessage - <https://security.apple.com/blog/imessage-pq3/>



Experiments x +

Chrome | chrome://flags

kyber Reset all

Experiments

119.0.6045.199

Available Unavailable

- TLS 1.3 hybridized **Kyber** support
This option enables a combination of X25519 and Kyber in TLS 1.3. – Mac, Windows, Linux, ChromeOS, Android, Fuchsia, Lacros
[#enable-tls13-kyber](#) Enabled

Cloudflare Research: Post-Quantum Key Agreement



On essentially all domains served (1) through **Cloudflare**, including this one, **we have enabled** hybrid post-quantum key agreement. We are also **rolling out support** for post-quantum key agreement for connection from Cloudflare to origins (3).

You are using *X25519Kyber768Draft00* which is **post-quantum secure**.

Deployed key agreements

Available with TLSv1.3 including HTTP/3 (QUIC)

Key agreement	TLS identifier
<i>X25519Kyber768Draft00</i>	<code>0x6399</code> (recommended) and <code>0xfe31</code> (obsolete)
<i>X25519Kyber512Draft00</i>	<code>0xfe30</code>

X25519Kyber[x]Draft00 is a **hybrid** of *X25519* and *Kyber[x]Draft00* (in that order).

Software support

- **Chrome 116+** if you turn on *TLS 1.3 hybridized Kyber support* (`enable-tls13-kyber`) in `chrome://flags`.



Overview

Main origin

Reload to view details

Security overview



This page is secure (valid HTTPS).

Certificate - valid and trusted

The connection to this site is using a valid, trusted server certificate issued by E1.

[View certificate](#)

Connection - secure connection settings

The connection to this site is encrypted and authenticated using TLS 1.3, X25519Kyber768Draft00, and AES_128_GCM.

Resources - all served securely

All resources on this page are served securely.

Console What's New X

Highlights from the Chrome 119 update

Improved @property section in Elements > Styles

You can now edit the @property rule in Elements > Styles.

Updated list of devices



What do we need to do?

1. Don't panic.
2. Educate yourself and your company on the risks.
3. Understand your environment.
 - a. Where do you use encryption?
 - b. What type of encryption do you use?
 - c. Can you update? Are your vendors working on implementing PQC?
 - d. Realize that you should be auditing your encryption usage anyway.



```

[1]ep:~$ nmap -sV --script ssl-enum-ciphers --script ssl-cert www.example.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-12 20:54 EST
Nmap scan report for www.example.com (93.184.216.34)
Host is up (0.0055s latency).
Other addresses for www.example.com (not scanned):
2606:2800:220:1:248:1893:25c8:1946
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http   Edgecast CDN httpd (nyb/1DCD)
|_ http-server-header: ECS (nyb/1DCD)
443/tcp   open  ssl/http Edgecast CDN httpd (nyb/1DCD)
| ssl-enum-ciphers:
| TLSv1.0:
|   ciphers:
|   compressors:
|     NULL
|   cipher preference: server
| TLSv1.1:
|   ciphers:
|     TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|     TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp256r1) - A
|     TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|     TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
|     TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 2048) - A
|     TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 2048) - A
|     TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|     TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 2048) - A
|     TLS_DHE_RSA_WITH_SEED_CBC_SHA (dh 2048) - A
|   compressors:
|     NULL
|   cipher preference: server
| TLSv1.2:
|   ciphers:
|     TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|     TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (secp256r1) - A
|     TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A
|     TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A

```

```

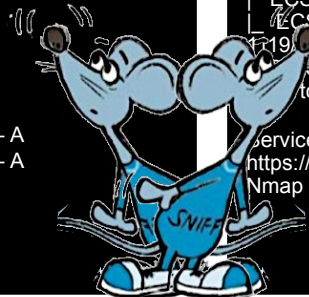
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (secp256r1) - A
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (secp256r1) - A
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp256r1) - A
| compressors:
|   NULL
| cipher preference: server
|
| TLSv1.3:
|   ciphers:
|     TLS_AKE_WITH_AES_256_GCM_SHA384 (secp256r1) - A
|     TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (secp256r1) - A
|     TLS_AKE_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|   cipher preference: server
|   least strength: A
|   ssl-cert: Subject: commonName=www.example.org/organizationName=InternetxC2I
xA0Corporation\xC2xA0for\xC2\xA0Assigned\xC2\xA0NamesxC2\xA0andxC2I
xA0Numbers/stateOrProvinceName=California/countryName=US
| Subject Alternative Name: DNS:www.example.org, DNS:example.net,
DNS:example.edu, DNS:example.com, DNS:example.org, DNS:www.example.com,
DNS:www.example.edu, DNS:www.example.net
| Issuer: commonName=DigiCert TLS RSA SHA256 2020
CA1/organizationName=DigiCert Inc/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-01-13T00:00:00
| Not valid after: 2024-02-13T23:59:59
| MD5: 749bbbeb4a6cb23c205c9850b35bed6a
| SHA-1: f2aad73d32683b716d2a7d61b51c6d5764ab3899
| http-server-header:
|   ECS (nyb/1D2E)
|   ECS (nyb/1DCD)
| 19) closed bnetgame
| 5) open h323q931?
| tcp closed rtmp

```

```

| service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 189.80 seconds

```



What do we need to do?

1. Plan

- a. What can you upgrade?
- b. What can't you upgrade?
- c. Risk Analysis

2. Execute your Plan

3. Automation

- a. ACME - Automatic Certificate Management Environment
- b. SCEP - Simple Certificate Enrollment Protocol
- c. REST/API - Check with your certificate provider, InCommon supports this



CYBER RISK

DARKREADING TECHNOLOGY

News, news analysis, and commentary on the latest trends in cybersecurity technology.

Google Proposes Reducing TLS Cert Life Span to 90 Days

Organizations will likely have until the end of 2024 to gain visibility and control over their keys and certificates.

#Cyl



Dark Reading Staff, Dark Reading

March 14, 2023

🕒 2 Min Read



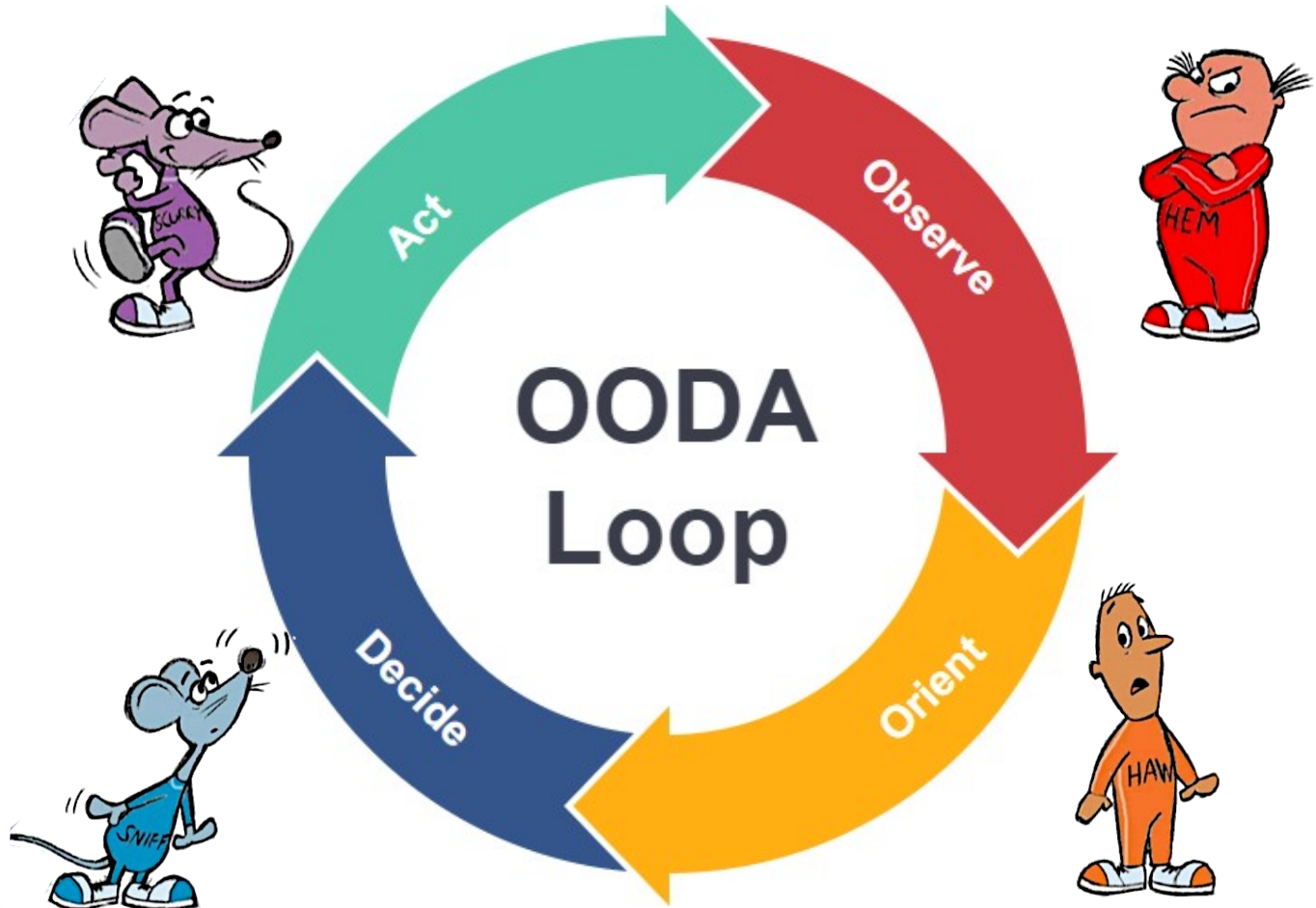
Latest Articles in DR Technology

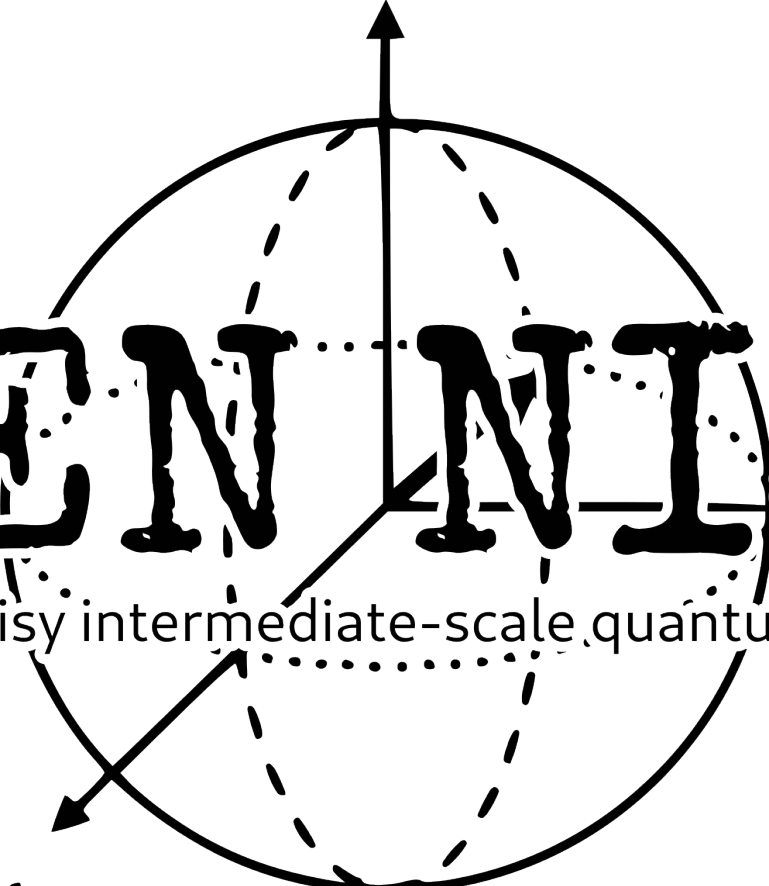
ACY
ENCE
EVENT

Module-Lattice-based Key-Encapsulation Mechanism FIPS-203

13. Qualifications. In applications, the security guarantees of a KEM only hold under certain conditions (see NIST SP 800-227 [1]). One such condition is the secrecy of several values, including the randomness used by the two parties, the decapsulation key, and the shared secret key itself. Users shall, therefore, guard against the disclosure of these values. While it is the intent of this standard to specify general requirements for implementing ML-KEM algorithms, **conformance to this standard does not ensure that a particular implementation is secure**. It is the responsibility of the implementer to ensure that any module that implements a key establishment capability is designed and built in a secure manner. Similarly, the use of a product containing an implementation that conforms to this standard does not guarantee the security of the overall system in which the product is used. The responsible authority in each agency or department shall ensure that an overall implementation provides an acceptable level of security. NIST will continue to follow developments in the analysis of the ML-KEM algorithm with its other cryptographic algorithm standards, **NIST will formally reevaluate this standard every five years**. Both this standard and possible threats that reduce the security provided through the use of this standard will undergo review by NIST as appropriate, taking into account newly available analysis and technology. In addition, the awareness of **any breakthrough in technology or any mathematical weakness of the algorithm will cause NIST to reevaluate** this standard and provide necessary revisions.







GENNISQ

noisy intermediate-scale quantum era

not that worried, bra /

Who Moved my Rock?

Post-Quantum Cryptography and its Impact on Higher Education

We live in an age of misinformation, fear, uncertainty, and doubt. I put together this talk to discuss where we are with Post-Quantum Cryptography and whether it is time to panic or not.

Who am I?

Brian Epstein (he/him) - bepstein@ias.edu

Institute for Advanced Study - ias.edu/security

IT Manager, Network and Security
Chief Information Security Officer

Mastodon: infosec.exchange/@ep

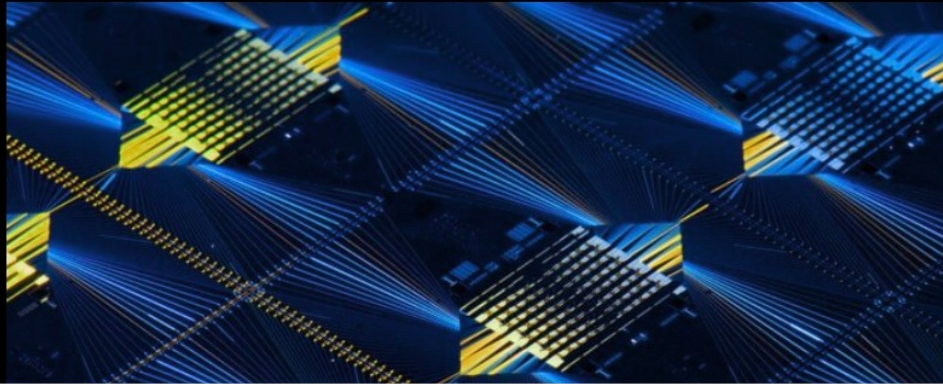
X: [@epepepep](https://twitter.com/epepepep)



My name is Brian Epstein, my pronouns are he him. I'm not a cryptographer, but I have been a practitioner of cryptography for over two decades in my career in IT. I currently work at the Institute for Advanced Study in Princeton, NJ and serve the role as IT Manager for Network and Security and the Chief Information Security Officer. The Institute for Advanced Study promotes the disinterested pursuit of knowledge unburdened from distraction for our scholars.

Quantum Computers Could Crack Encryption Sooner Than Expected With New Algorithm

By **Edd Gent** > October 2, 2023



FEATURED



Mice Just Passed the Mirror Test. Here's What That Says About Our Sense of Self

December 7, 2023

[Load more >](#)

It's hard to determine what is truth and what is fiction in today's quantum news.

China's new quantum code-breaking algorithm raises concerns in the US

The new algorithm could render mainstream encryption powerless within years.



Baba Tamim

Published: Jan 12, 2023 06:56 AM EST

INNOVATION



Headlines like these come out on a daily basis.

TRENDING: Reinvent the Customer Journey With Enterprise Decisioning •

Euro Security Watch with Mathew J. Schwartz

Tracking security and privacy trends across UK, Europe and beyond



GET DAILY

Covering top
fraud, and li

Email a

By submitti
GDPR State

Encryption & Key Management , Security Operations

Researcher Claims to Crack RSA-2048 With Quantum Computer

As Ed Gerck Readies Research Paper, Security Experts Say They Want to See Proof

Mathew J. Schwartz (@euroinfosec) • November 1, 2023

RESOURCE

#CybersecPrivacy24

CYBERSECURITY AND PRIVACY
PROFESSIONALS CONFERENCE
AT EDUCAUSE

Sometimes they point to stories that are difficult to believe.

FORBES > INNOVATION > ENTERPRISE TECH

Debunking The Top 5 Quantum Computing Myths

Bernard Marr Contributor

Follow



Aug 23, 2023, 02:58am EDT

Other times, you see helpful articles that try to dispel the myths.

★ Member-only story

NASA Just Shut Down Quantum Computer After Something Insane Happened!

Top hig

Houston, We Have a Problem!



The Pareto Investor  · Follow

3 min read · Nov 9



8.5K



158



And sometimes you just see articles that are plain fake.

Guest Post: Harvest Now, Decrypt Later? The Truth Behind This Common Quantum Theory

Insights

Jeffrey Duran • February 7, 2023



What should we believe and what should we be skeptical about?

Disrupted

Microsoft, Quantinuum claim breakthrough in quantum computing

By Stephen Nellis

April 3, 2024 5:11 PM EDT · Updated 22 days ago



This article came out just last month.



Paper 2024/555

Quantum Algorithms for Lattice Problems

Yilei Chen ^{ID}, Tsinghua University, Shanghai Artificial Intelligence Laboratory, Shanghai Qi Zhi Institute

Abstract

We show a polynomial time quantum algorithm for solving the learning with errors problem (LWE) with certain polynomial modulus-noise ratios. Combining with the reductions from lattice

And this other one caused a bit of a panic soon after. It claimed to be able to break the new Quantum Safe Cryptography we will discuss in this talk. A week later, the author revealed that a “bug” was found in the paper and removed the claim.

So what should we be doing to protect ourselves and our institutions?

Characters

Hem - angry, impatient, unwilling to move

Haw - scared stiff, but thinks

Sniff - always using their tools

Scurry - always on the move to discover new things



#CybersecPrivacy24

ERSECURITY AND PRIVACY
PROFESSIONALS CONFERENCE
EDUCAUSE

For this talk we'll follow a cast of characters and their adventures through history. We borrow them from the popular book by Spencer Johnson, "Who Moved My Cheese".

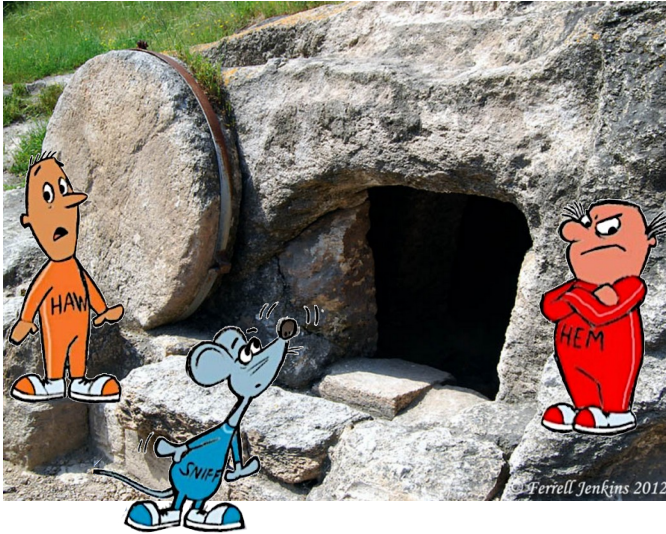
Hem - angry, impatient, unwilling to move

Haw - scared stiff, but thinks

Sniff - always using their tools

Scurry - always on the move to discover new things

Protecting things



First Attempt at Security

- Put large boulders in front of our caves
- Someone figured out how to move them
- Repetitive journey - find the better lock



Our first attempts <Click> at security involved hiding our stuff by moving a big boulder in front of our caves <Click>. This took a lot of work to roll the boulder and get just right.

However <Click>, someone figured out how to move our rock and get to our stuff anyway.

<Click> This scared Haw who said, "What will we do now?"

<Click> Hem was mad and said "We'll move the rock back of course. They won't figure it out again!"

<Click> Sniff knew this was a mistake, though, and

<Click> Scurry said, "Let's find a new way to protect our stuff!"

So, this is the story we have repeated throughout history <Click>. We create a lock, someone breaks it, so we make a new lock.

Let's visit some of these innovations over time.

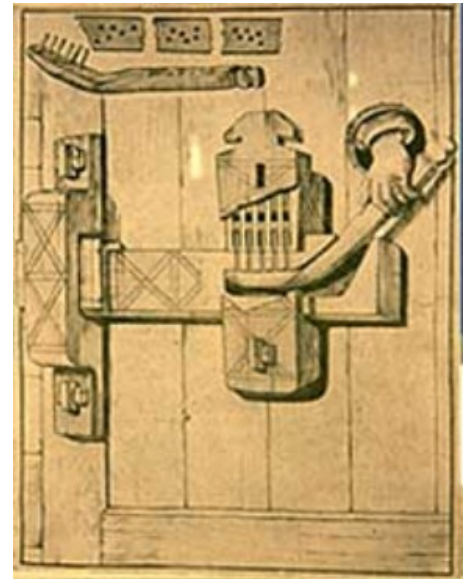
Protectors



We first utilized protectors to guard against a thief. Human guards were error prone and unreliable, so we started with <Click> man's best friends trained to attack intruders.

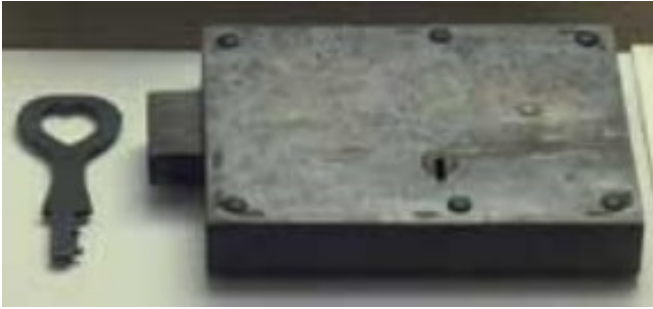
<Click> The story goes that in India, they placed their valuables in pools filled with partially starved crocodiles. In both cases, drugging or killing the animals was an effective means of bypassing the lock.

Locks and Lockpicks



Moving to a non-living protection method was the next evolution. At first we used <Click> seal locks that would be able to show evidence of tampering. Although a deterrent, it didn't actually prevent someone from gaining access. <Click> In the 8th century BC in Persia one of the first wooden locks was used in a method very similar to a tumbler lock that we use today.

Locks and Lockpicks



#CybersecPrivacy24

CYBERSECURITY AND PRIVACY
PROFESSIONALS CONFERENCE
AT EDUCAUSE

As materials improved, we moved to <Click> metallic tumbler locks and <Click> rim locks. Both proved tricky for criminals to bypass, but not impossible. Today, we see many hobbyists picking locks like these for fun.

Value of things and information

- Information became valuable
 - War plans
 - Hidden treasure
 - Hunting / farming grounds
- Necessary to transmit
 - Get strategy to front lines
 - Remote allies
- Easy to intercept



At some point in history, we started to realize that in many circumstances, information can be much more valuable than possessions.

Information also needs to be transferred in some way or another for it to be useful.

This led to problems with information being intercepted and used for malicious purposes.

Hiding info in plain sight

- Novel schemes
- Useless when discovered
- Secrets can be bought



Which lead us to invent methods to hide information in plain sight.

<Click>

Many of these were novel schemes. They worked, but

<Click>

Once discovered how they worked, they were useless.

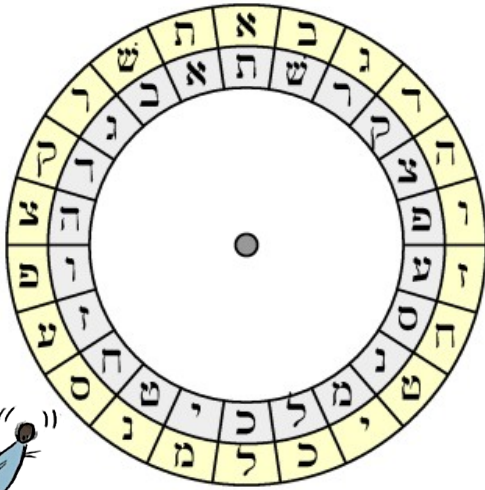
And of course, <Click> secrets can be bought.

Hiding info in plain sight



During the Classical Era <Click>, the Spartans were known to use Scytale (skit-a-lee) Cipher's to transmit message by a leather belt wrapped around a pole. There are even classical writers <Click> who indicate that in 500 BC, Histiaeus tattooed secret messages onto his slaves shaven heads, waited until their hair grew back, and then sent them to deliver the messages in secret.

Encoding



In other parts of the world, they were coming up with their own ways of encoding secret messages, <Click> like Atbash in Israel in 500 BC, and the <Click> Polybius Square in 200 BC.

All of these methods were trivial to break once discovered how they worked. So, our heroes had to move on to something new.

Symmetric Key Cryptography

- Single key for both encryption and decryption
- Simple
- Powerful
- Key distribution issues
- Number of keys difficult

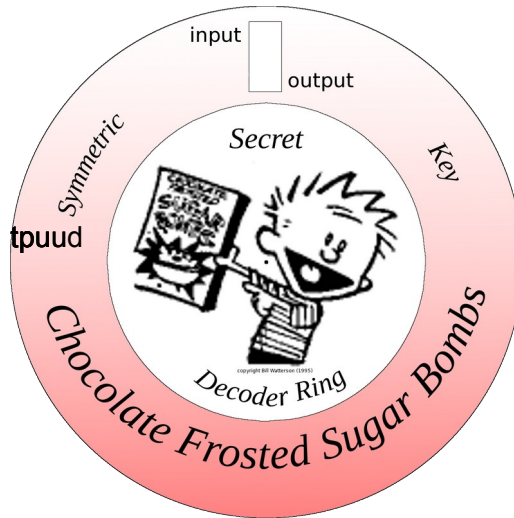
How can we hide information in a way that is known without it being discoverable? Much like a lock, we need a key that is kept secret and only the keyholders can <Click> encrypt and decrypt the message.

Having a system like this gives us many desirable <Click> properties <Click> that can help transfer secret messages.

However, how do we keep those <Click> keys safe? How do we make sure only those that should, have the keys. And how to you <Click> keep track of all those keys? Every conversation you have with a different person needs a new key!

Symmetric Key Cryptography

tpuud

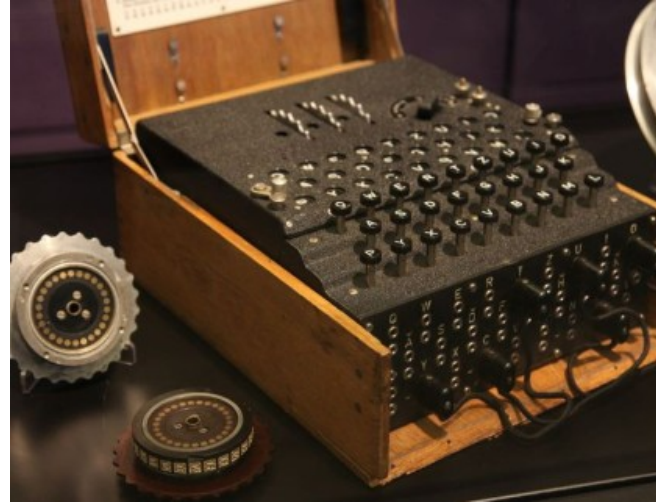
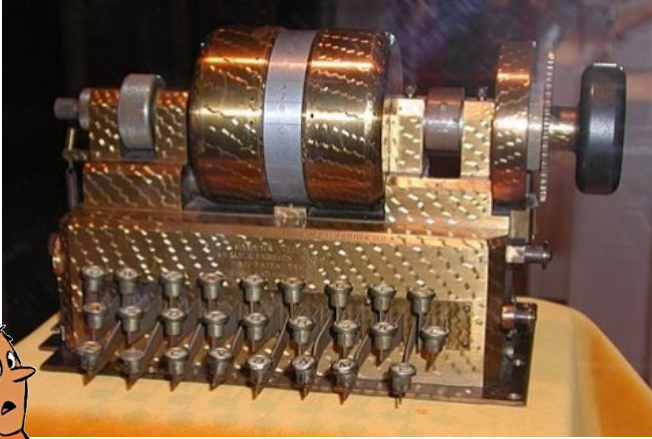


Jimmy

Here we've created a simple substitution cipher
<Click><Click> where we decrypt our simple
message <Click> by rotating the dial <Click> and
writing down each letter.<Click><Click><Click>

The nice thing about this type of cipher is that the
same key can be used for encrypting as well
<Click>, which you can see when I reverse the
process.

Symmetric Key Cryptography

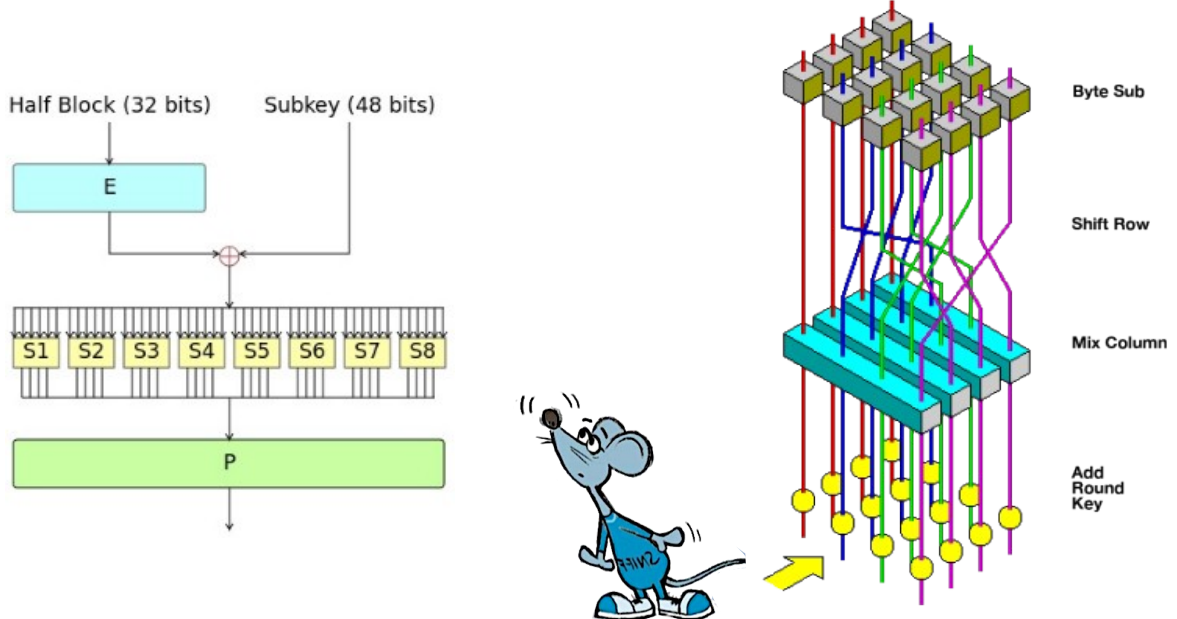


Privacy24

CYBERSECURITY AND PRIVACY
PROFESSIONALS CONFERENCE
AT EDUCAUSE

Before modern computers, many types of machines were created in the early 20th century. [Click](#) the Hebern Rotor Machine in 1921 failed as product, only selling a few machines to the US Navy in 1931. A more famous rotor machine was invented by the Germans in 1923. [Click](#) The Enigma Machine was actively used for transmitting secrets during WWII. In 1932, the Enigma machine was defeated by a Polish mathematician.

Symmetric Key Cryptography



#CybersecPrivacy24

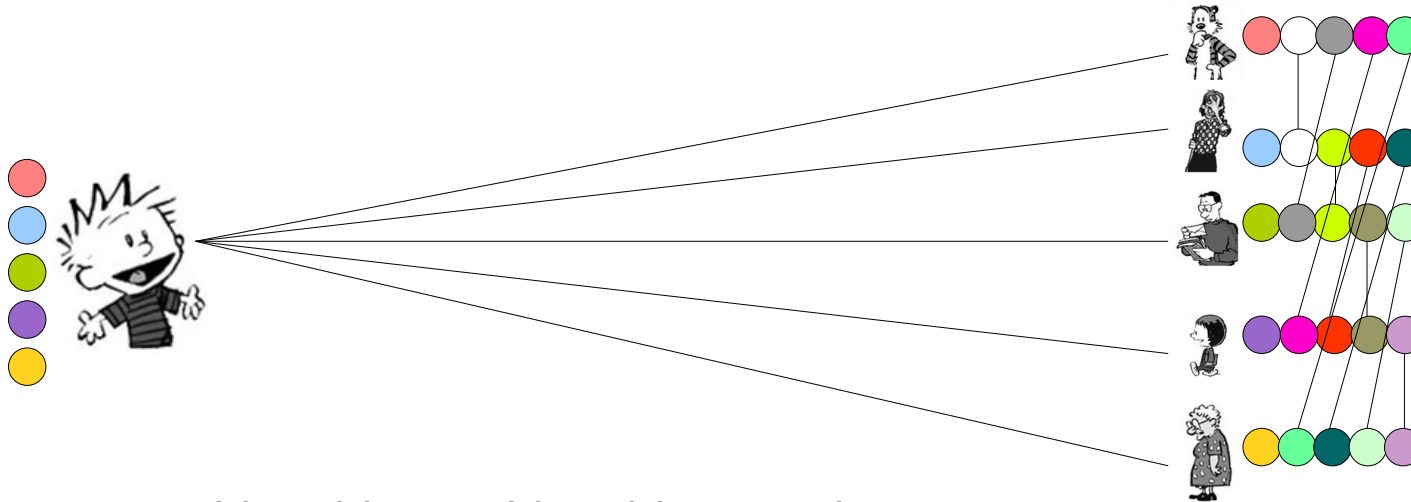
CYBERSECURITY AND PRIVACY
PROFESSIONALS CONFERENCE
AT EDUCAUSE

The Data Encryption Standard (DES) <Click> was created in 1975 and utilized mathematics for the scheme to encrypt and decrypt based on a private key. The NSA approved DES, but with a small key size that they could brute force if necessary. By 1999, public organizations showed that they too could break DES leading us to the use of triple DES (3DES), which is vulnerable to some theoretical attacks.

This brought us to the adoption <Click> of the Advanced Encryption Standard (AES) in 2001. AES-256 is considered to be quantum resistant as well.

Either way, the key distribution remains an issue even with AES-256, which leads us to our next topic.

Symmetric Key Cryptography



$$n*(n-1)/2 = 6*(5-1)/2 = 30/2 = 15 \text{ unique keys}$$

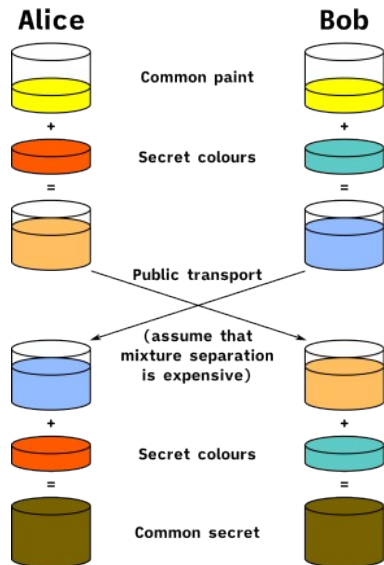
May not seem significant, but remember, for 100 people, you'll need almost 5000 keys (4,950). Keeping track of all those keys turns into a nightmare.

Key Exchange

- Generate a unique secret key
- Never transfer it over the wire

Another problem with Symmetric Key Cryptography is how to share the key securely. One method is to utilize a key exchange protocol. It allows you to generate a one time use, random secret key. And then transfer enough information to so that both parties can know the secret key without ever having transferred it over the wire.

Key Exchange (Diffie-Hellman)



Public Key Cryptography/ECDH

<p><i>Bob picks</i> $\beta = 7$</p> <p><i>Bob computes</i> $B = 7G = (4,7)$</p> <p><i>Bob Receives</i> $A = (6,7)$</p> <p><i>Bob computes</i> $\beta A = 7(5G) = 35G = 17G = (7,10)$</p>	<p><i>Common parameters</i> $y^2 = (x^3 + 2 \cdot x + 3) \pmod{13}$</p> <p>$+ N = 18$</p> <p>$G = (7,3)$</p> <p>$B = (4,7)$</p> <p>$A = (6,7)$</p>	<p><i>Alice picks</i> $\alpha = 5$</p> <p><i>Alice computes</i> $A = 5G = (6,7)$</p> <p><i>Alice Receives</i> $B = (4,7)$</p> <p><i>Alice computes</i> $\alpha B = 5(7G) = 35G = 17G = (7,10)$</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Whitfield Diffie and Martin Hellman came up with such a <Click> scheme in 1976 utilizing discrete logarithms. In 1985 <Click>, a method of using Elliptic Curves along with Diffie-Hellman was created as well. Both are popular today. However, there are times when you want to encrypt and decrypt more than just a secret key. In those cases, Asymmetric Cryptography is used.

Asymmetric Key Cryptography

- Split public and private keys
- Key distribution easier
- Number of keys manageable
- Slower
- Authentication issues
- Relies on a one-way function (Theoretical Math)

Using the idea of splitting a key into a public half shared by all, and a private half kept secret corrected a lot of issues surrounding key distribution. Now you only needed one private key and you could share your public key with the world. Unfortunately, this type of encryption is slower and is typically only used to share a secret key that is used for symmetric key encryption.

Another issue with this is verifying the authenticity of a key holder, which is a problem we solve with a Public Key Infrastructure, or PKI.

The beauty of Asymmetric Key cryptography is the use of a one-way function which uses a mathematical construct.

One way function



Easy to make scrambled eggs, but it is practically impossible to de-scramble eggs without some sort of a magic de-scrambler.

We want something similar, where it is easy to go one way, but nearly impossible to go back the other way without some other piece of knowledge.

January 23, 2015

UCI, fellow chemists find a way to unboil eggs

Share

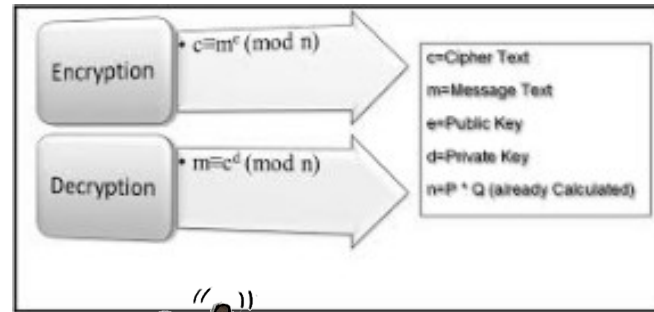
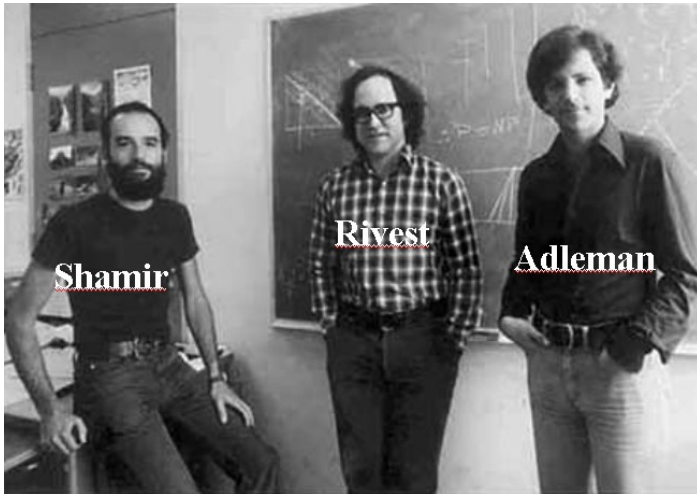


Irvine, Calif., Jan. 23, 2015 – UC Irvine and Australian chemists have figured out how to unboil egg whites – an innovation that could dramatically reduce

Anyone here from University of California, Irvine? Don't spoil my analogy! *

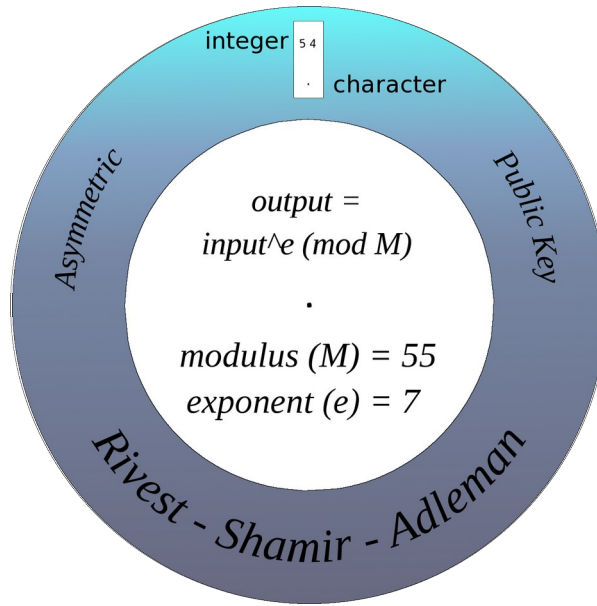
* <http://news.uci.edu/press-releases/uci-fellow-chemists-find-a-way-to-unboil-eggs/>

RSA



In 1977, <Click> Ron Rivest, Adi Shamir and Leonard Adleman created the RSA algorithm. <Click> It relied on the one-way function of factoring large composite numbers made by multiplying two primes together, also know as a semi-prime. Multiplying is easy, factoring is hard.

One way function for RSA



Jimmy

14 13 18 18 3

One way function for RSA

14 13 18 18 36

14 13 18 18 36

One way function for RSA

~~144~~ 13 18 18 36

13

18

18

36

One way function for RSA

$$14^7 \bmod 55 = 105413504 \bmod 55 = 9 = \text{"g"}$$

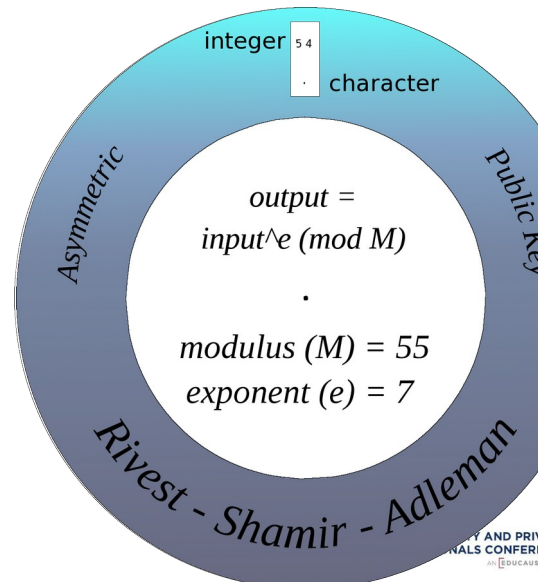
$$13^7 \bmod 55 = 7 = \text{"e"}$$

$$18^7 \bmod 55 = 17 = \text{"l"}$$

$$18^7 \bmod 55 = 17 = \text{"l"}$$

$$36^7 \bmod 55 = 31 = \text{"v"}$$

$$E(\text{Jimmy}) = \text{gellv}$$



One way function for RSA

$$9^{23} \bmod 55 = 14 = \text{"J"}$$

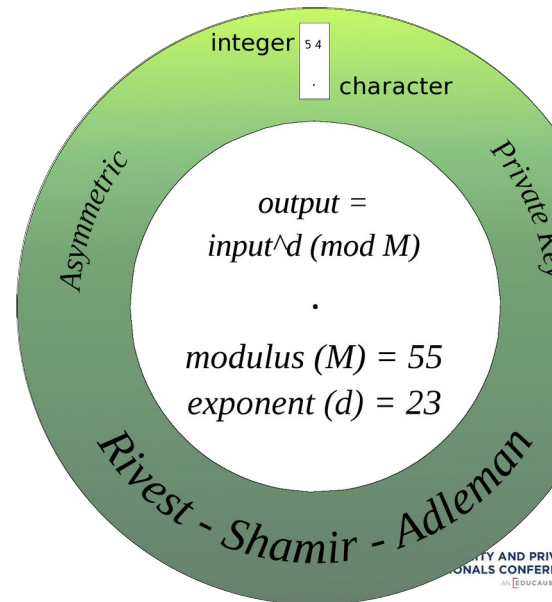
$$7^{23} \bmod 55 = 13 = \text{"i"}$$

$$17^{23} \bmod 55 = 18 = \text{"m"}$$

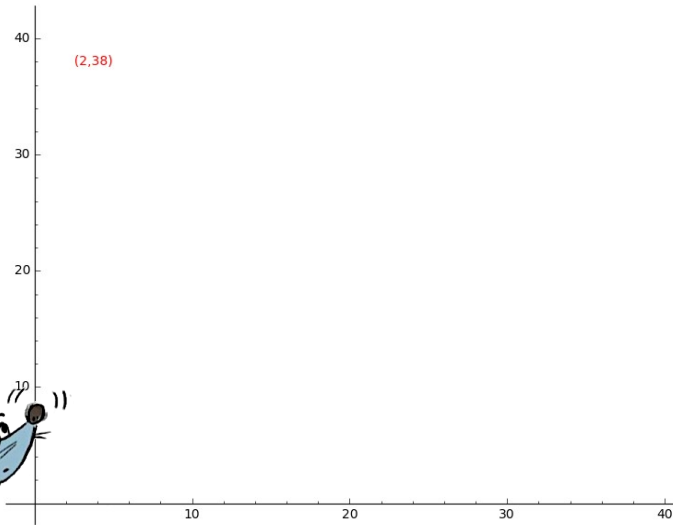
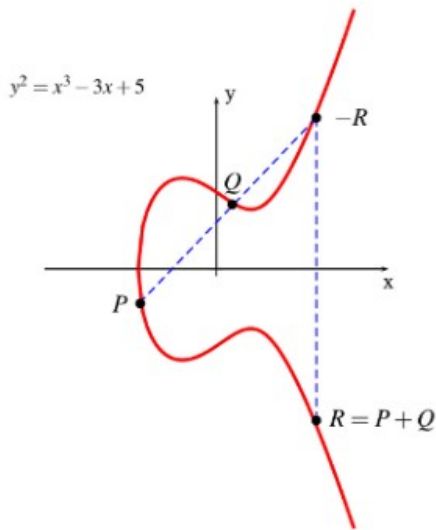
$$17^{23} \bmod 55 = 18 = \text{"m"}$$

$$31^{23} \bmod 55 = 36 = \text{"y"}$$

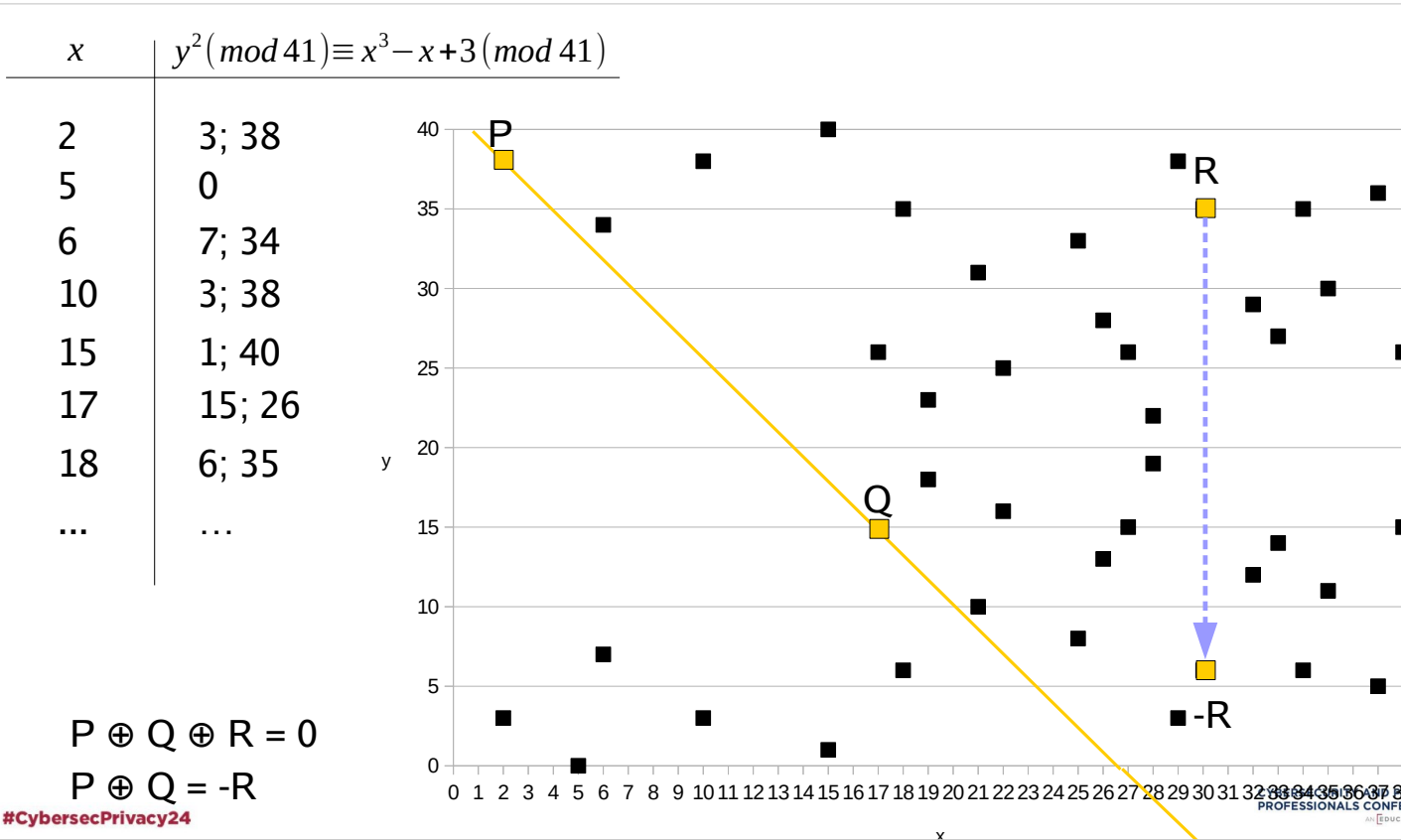
D(gellv) = Jimmy



Elliptic Curve Cryptography

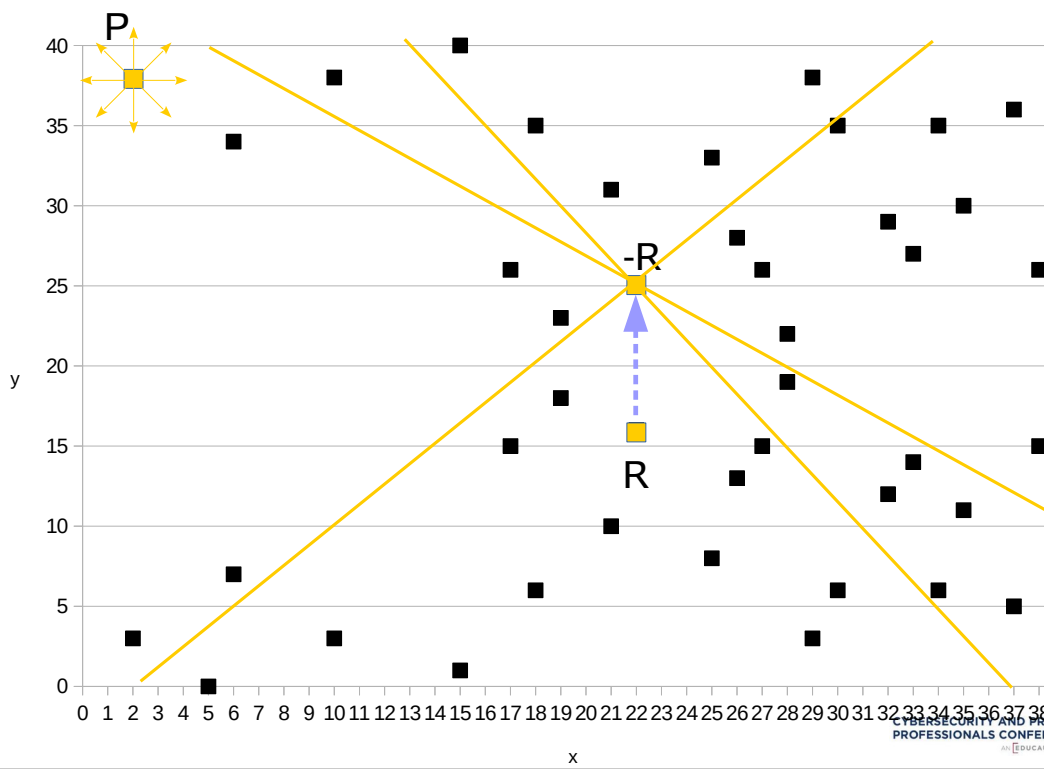


In 1985, Elliptic Curve Cryptography was created which relies on the mathematical properties of an Elliptic Curve. It's smaller keys and faster mathematics make it ideal for use on small devices without a lot of computing power.



The one way function for ECC involves plotting the graph in a finite field and then doing what is called point addition. By drawing a line through points P&Q you can jump back to the top of the graph and eventually draw through point R. Then you can flip across the center axis to find the negative R value which is the answer.

x	$y^2(\text{mod } 41) \equiv x^3 - x + 3(\text{mod } 41)$
2	3; 38
5	0
6	7; 34
10	3; 38
15	1; 40
17	15; 26
18	6; 35
...	...



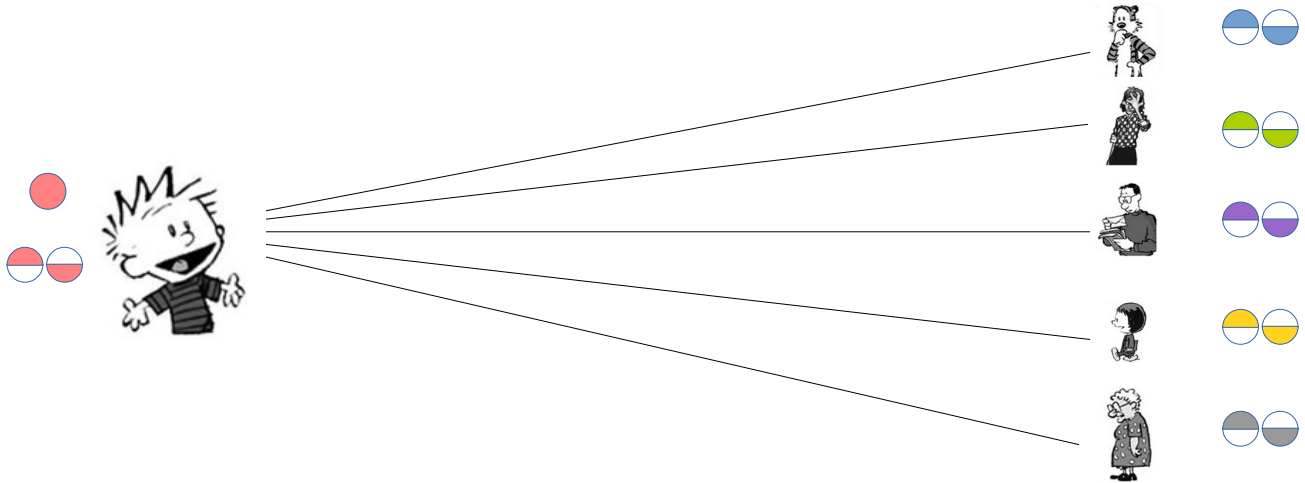
$P + Q = -R$
 $P + Q + R = 0$

#CybersecPrivacy24

CYBERSECURITY AND PRIVACY PROFESSIONALS CONFERENCE

Reversing this flow, however is difficult. Even if you know point P and point R, you don't know which direction to draw in order to find your point Q. This is the one way function of ECC.

Asymmetric Key Cryptography



$$2 * n = 2 * 6 = 12 \text{ unique keys}$$

So for asymmetric key cryptography, sharing keys between Calvin <Click> and his friends <Click> becomes much more manageable. Calvin creates a key <Click> and splits it into two <Click><Click>, a public half and a private half. His friends then repeat this process <C5>. Sharing public keys <Click> is easy as it doesn't matter if they are stolen.

<Click> Remember, for 100 people Symmetric crypto would require about 5000 keys (4950).

For Asymmetric crypto, we only need 200.

Quantum Computers

- Use qubits instead of classical bits
- Qubits can be in more than one state at the same time
- The state of a qubit is unknown until you observe it
- Qubits are fragile and interference can put them into an error state
- Error rates slow down the development of quantum computers

Quantum computing uses quantum mechanical properties such as superposition and entanglement to solve problems. Much like a bit in classical computers, quantum computers use qubits to represent data. Qubits have the special property of being able to exist in multiple states at the same time. Once observed the qubits reveal their final state in a probabilistic manner. Qubits are prone to errors due to many factors, which slows down the efficacy of quantum computers.

Shor's Algorithm

- Makes factoring large semiprimes obtainable
- Another algorithm for solving the discrete logarithm problem
- And yet another for the period finding problem
- With proper quantum computer, RSA, DH, ECC, ECDH all become obsolete

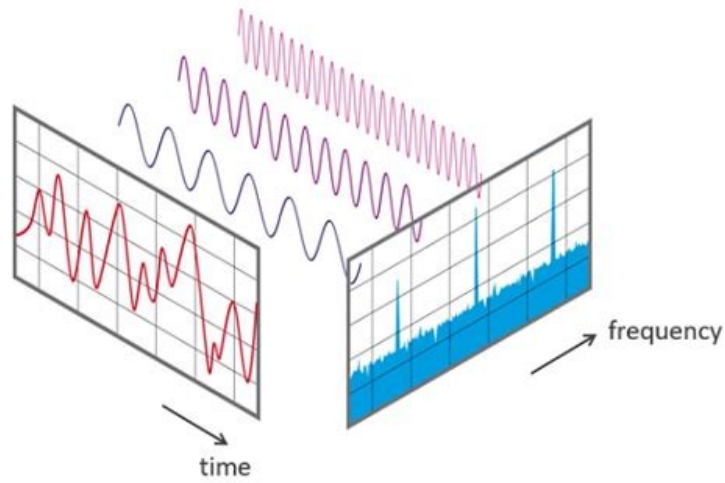
Peter Shor created an algorithm in 1994 that used a theoretical quantum computer to <Click> drastically reduce the time needed to factor semiprimes back to their original primes.

He created other algorithms to solve the <Click> discrete logarithm and <Click> period finding problem in a shorter amount of time as well.

With the correct hardware, <Click> these algorithms could render all of the previously discussed key exchange and asymmetric key cryptography systems unusable.

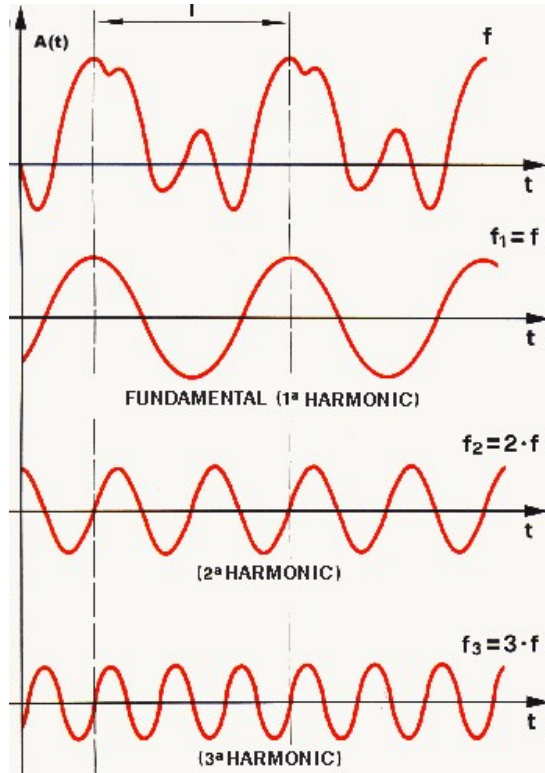
The concepts of Shor's algorithms are complex, but here is a brief overview of how they work.

Fourier Transform

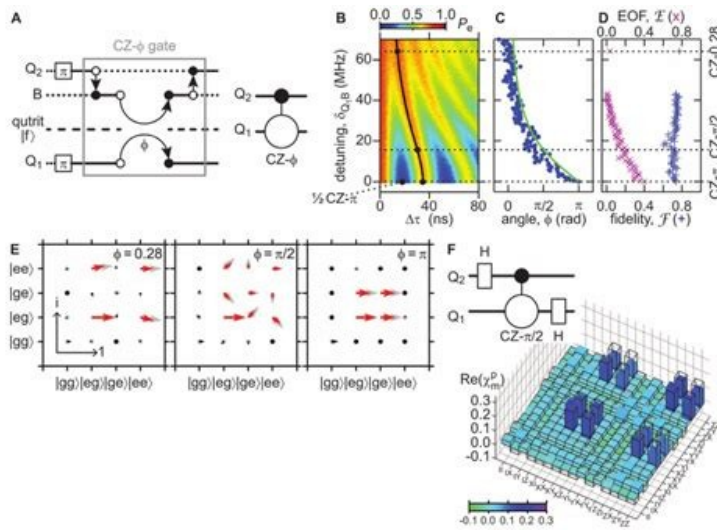


A Fourier Transform is a way to convert a complex wave into its component waves. On the left of this slide we see a complex wave over time being transformed into a frequency graph on the right.

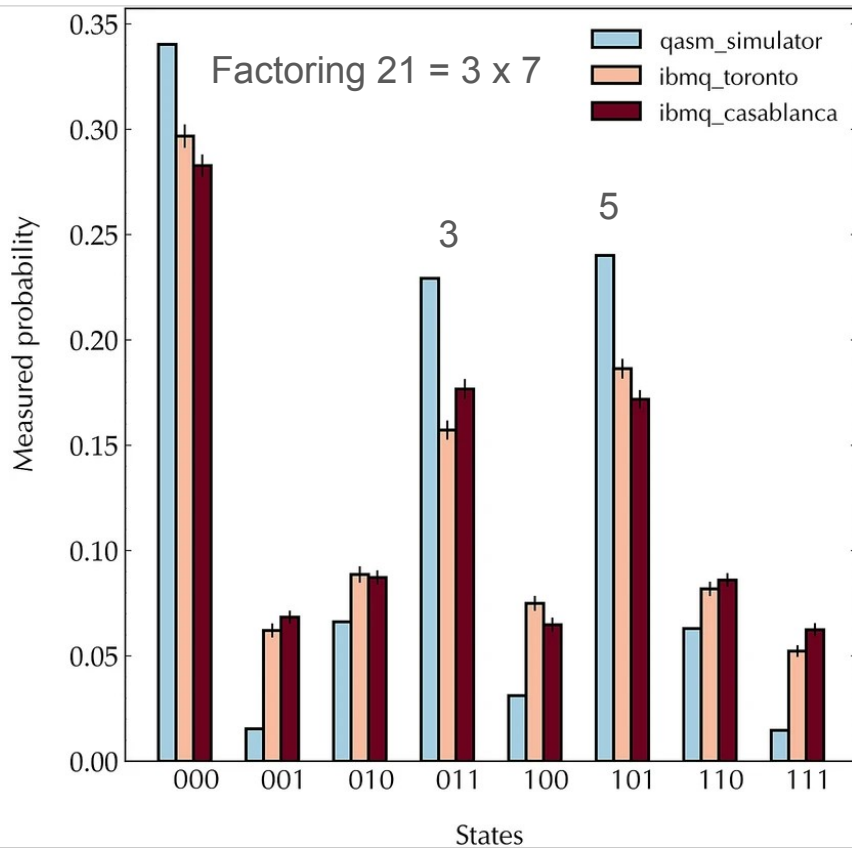
Fourier Transform



This is useful in breaking down waves of any kind, like sound waves to determine how they are formed.



Utilizing Quantum Fourier Transform, we can sequence the probability amplitudes for all the possible outcomes upon measurement. This ability to test everything with a single measurement shows us the power of quantum computers. Notice, that unlike discrete systems, quantum computers and qubits give us a probable answer which may change over multiple runs. The number of runs is very small, though, compared to the processing that a classical computer would need.



Let's talk about the errors that happen with quantum computers. Here is the results of a 5 qubit quantum computer attempting to factor 21 into its component primes 3 and 7. As you can see, the measured probability of 0 is the highest given, with 3 and 5 coming in close second and third. The 0 result is an error state and should be ignored. Given 4 as the initial guess, the IBM casablanca test came out favoring 3 as one of the primes, which was the desired result. This experiment is considered as being successful in factoring of the semiprime 21 even though there are lots of error conditions to consider.



GEN NISQ

noisy intermediate-scale quantum era



Privacy24



CYBERSECURITY
PROFESSIONAL

So that leads us to the question, are quantum computers ready to start breaking the Internet? It has been described that we are in the <Click> Noisy Intermediate-Scale Quantum era. I like to say that everyone alive today is part of generation NISQ, or GEN-NISQ. This is a time of innovation and investment into building better and more stable quantum computers. But, we still have time before quantum computers will be breaking the encryption that keeps us all safe today.

GENNISQ

noisy intermediate-scale quantum era

not that worried, bra



Privacy24



CYBERSECURITY
PROFESSIONAL

So why aren't I more worried about Quantum computers? To explain, we'll let's take a look at our timeline and what quantum computers have accomplished so far.

1977

Rivest, Shamir, and Adleman designed the RSA algorithm which uses the idea that prime factorization is difficult.

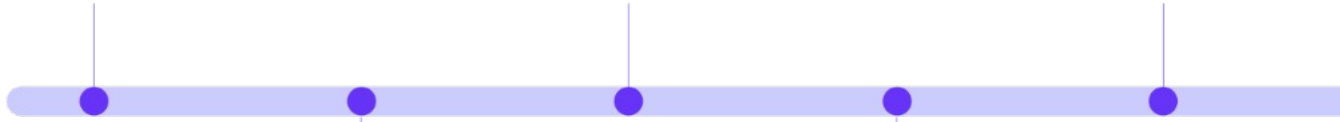
2001

IBM shows that Shor's algorithm can work on a quantum computer with 7 qubits. It was able to factor 15 into 3 x 5.

2012

NIST formally begins the Post Quantum Cryptography Project.

Prime factorization of 21 into 3 x 7 was achieved.



Peter Shor designs an algorithm that can factor integers into its prime counterparts quickly by utilizing a quantum computer with a large number of qubits.

1994

NIST publishes a survey for protocol designers. The survey says, "it does not appear inevitable that quantum computing will end cryptographic security as we know it."

2009

NIST...
transi...
resist...
the no...

As you can see from this timeline, the RSA algorithm was created over 47 years ago. 17 years later, Peter Shor developed the algorithm that a quantum computer could use to break RSA. Only 7 years later, IBM was able to use a 7 qubit computer to factor a 4 bit number, 15, into its prime factors 3 and 5. 11 years after that a 5 bit number, 21, was factored into 3 x 7. Another 7 years goes by and in 2019 an attempt to factor a 6 bit number, 35, failed. There is a number of larger, specially chosen numbers that have been factored by a process called quantum annealing. This has not been proven as a general solution to the problem, though.

In the meantime, in 2009, NIST starts a survey to begin developing the next generation of quantum safe encryption algorithms. Fast forward to 2023, and NIST has completed several rounds of discovery and testing of various quantum safe algorithms and chosen its candidates. In 2024, we expect those algorithms to be fully ratified and that vendors will start releasing them in their products. Cloudflare, Signal, and Apple come to mind as vendors that have already adopted Quantum Safe Cryptography.

gins the
um
project.
on of 21
hieved.

2016

NIST publishes 1st report on PQC (8105). Deadlines for submissions of round one due in 2017.

2021 & 2022

NIST holds 3rd & 4th PQC Workshop.

Draft standards made, algorithms selected for QSC (CRYSTALS-KYBER, CRYSTALS-DILITHIUM, FALCON, SPHINCS+).

NIST holds 1st PQC Workshop.

NSA states that transitioning to quantum resistant algorithms is in the not too distant future.

NIST holds 2nd PQC Workshop.

Attempt to factor 35 into primes fails due to accumulating errors.

NIST announced that it will integrate the four selected post-quantum algorithms into U.S. encryption standards

2015

2019

2024

CYBERSECURITY AND PRIVACY
PROFESSIONALS CONFERENCE
AT EDUCAUSE

#CybersecPrivacy24

As you can see from this timeline, the RSA algorithm was created over 47 years ago.

17 years later, Peter Shor developed the algorithm that a quantum computer could use to break RSA. Only 7 years later, IBM was able to use a 7 qubit computer to factor a 4 bit number, 15, into its prime factors 3 and 5. 11 years after that a 5 bit number, 21, was factored into 3 x 7. Another 7 years goes by and in 2019 an attempt to factor a 6 bit number, 35, failed. There is a number of larger, specially chosen numbers that have been factored by a process called quantum annealing. This has not been proven as a general solution to the problem, though.

In the meantime, in 2009, NIST starts a survey to begin developing the next generation of quantum safe encryption algorithms. Fast forward to 2023, and NIST has completed several rounds of discovery and testing of various quantum safe algorithms and chosen its candidates. In 2024, we expect those algorithms to be fully ratified and that vendors will start releasing them in their products. Cloudflare, Signal, and Apple come to mind as vendors that have already adopted Quantum Safe Cryptography.

1977

Rivest, Shamir, and Adleman designed the RSA algorithm which uses the idea that prime factorization is difficult.

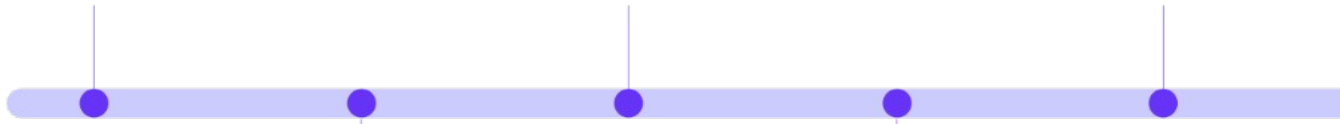
2001

IBM shows that Shor's algorithm can work on a quantum computer with 7 qubits. It was able to factor 15 into 3 x 5.

2012

NIST formally begins the Post Quantum Cryptography Project.

Prime factorization of 21 into 3 x 7 was achieved.



Peter Shor designs an algorithm that can factor integers into its prime counterparts quickly by utilizing a quantum computer with a large number of qubits.

1994

NIST publishes a survey for protocol designers. The survey says, "it does not appear inevitable that quantum computing will end cryptographic security as we know it."

2009

NIST
transi
resist
the no

As you can see from this timeline, the RSA algorithm was created over 47 years ago. 17 years later, Peter Shor developed the algorithm that a quantum computer could use to break RSA. Only 7 years later, IBM was able to use a 7 qubit computer to factor a 4 bit number, 15, into its prime factors 3 and 5. 11 years after that a 5 bit number, 21, was factored into 3 x 7. Another 7 years goes by and in 2019 an attempt to factor a 6 bit number, 35, failed. There is a number of larger, specially chosen numbers that have been factored by a process called quantum annealing. This has not been proven as a general solution to the problem, though.

In the meantime, in 2009, NIST starts a survey to begin developing the next generation of quantum safe encryption algorithms. Fast forward to 2023, and NIST has completed several rounds of discovery and testing of various quantum safe algorithms and chosen its candidates. In 2024, we expect those algorithms to be fully ratified and that vendors will start releasing them in their products. Cloudflare, Signal, and Apple come to mind as vendors that have already adopted Quantum Safe Cryptography.

gins the
um
project.
on of 21
hieved.

2016

NIST publishes 1st report on PQC (8105). Deadlines for submissions of round one due in 2017.

2021 & 2022

NIST holds 3rd & 4th PQC Workshop.

Draft standards made, algorithms selected for QSC (CRYSTALS-KYBER, CRYSTALS-DILITHIUM, FALCON, SPHINCS+).

NIST holds 1st PQC Workshop.

NSA states that transitioning to quantum resistant algorithms is in the not too distant future.

NIST holds 2nd PQC Workshop.

Attempt to factor 35 into primes fails due to accumulating errors.

NIST announced that it will integrate the four selected post-quantum algorithms into U.S. encryption standards

2015

2019

2024

#CybersecPrivacy24

CYBERSECURITY AND PRIVACY
PROFESSIONALS CONFERENCE
AT EDUCAUSE

As you can see from this timeline, the RSA algorithm was created over 47 years ago.

17 years later, Peter Shor developed the algorithm that a quantum computer could use to break RSA. Only 7 years later, IBM was able to use a 7 qubit computer to factor a 4 bit number, 15, into its prime factors 3 and 5. 11 years after that a 5 bit number, 21, was factored into 3 x 7. Another 7 years goes by and in 2019 an attempt to factor a 6 bit number, 35, failed. There is a number of larger, specially chosen numbers that have been factored by a process called quantum annealing. This has not been proven as a general solution to the problem, though.

In the meantime, in 2009, NIST starts a survey to begin developing the next generation of quantum safe encryption algorithms. Fast forward to 2023, and NIST has completed several rounds of discovery and testing of various quantum safe algorithms and chosen its candidates. In 2024, we expect those algorithms to be fully ratified and that vendors will start releasing them in their products. Cloudflare, Signal, and Apple come to mind as vendors that have already adopted Quantum Safe Cryptography.

Quantum Factored

4 bit semiprime (2001): 15

5 bit semiprime (2012): 21

6 bit semiprime (2019 - failed): 35

So, to review, this is what we've accomplished. <Click><Click><Click>Ok, this last one was a failure.

Need to factor

1024 bit semiprime:

14858031842529041742675223662020065615490358969220662
89933239251279096441074379066840500275518447762785965
12160601382625659982180758999544221868254722043619979
84526745656869867322663775380667065617182042243040564
49791211661181323805868000257522596407301217381568222
96246476504443847811940638639921190244907229

Here you can see the size of the numbers that need to be factored for us to be truly at risk.

Need to factor

2048 bit semiprime:

23042155144807033264822777505847352979234760665383921887127664352
78228085219412095936558643453832808918618527312570608206987897806
778266867076836343848261613715911228213968789316742345746645880253
12480684919203629910806547215276202762168939550018927857695365726
726743981630638931221130356279390701114143002852846597092746910840
713994794061711394675366463723772973275703764980539097704296549591
25201782358647860679882638386416987498248726270464804391684357189
38652132696815444783098028151462344977286590749544794608585623582
763434406818849621131943279250579658150209579438838292900340862945
2106790235491702341492875849

Roll of 2048bit semiprime example.

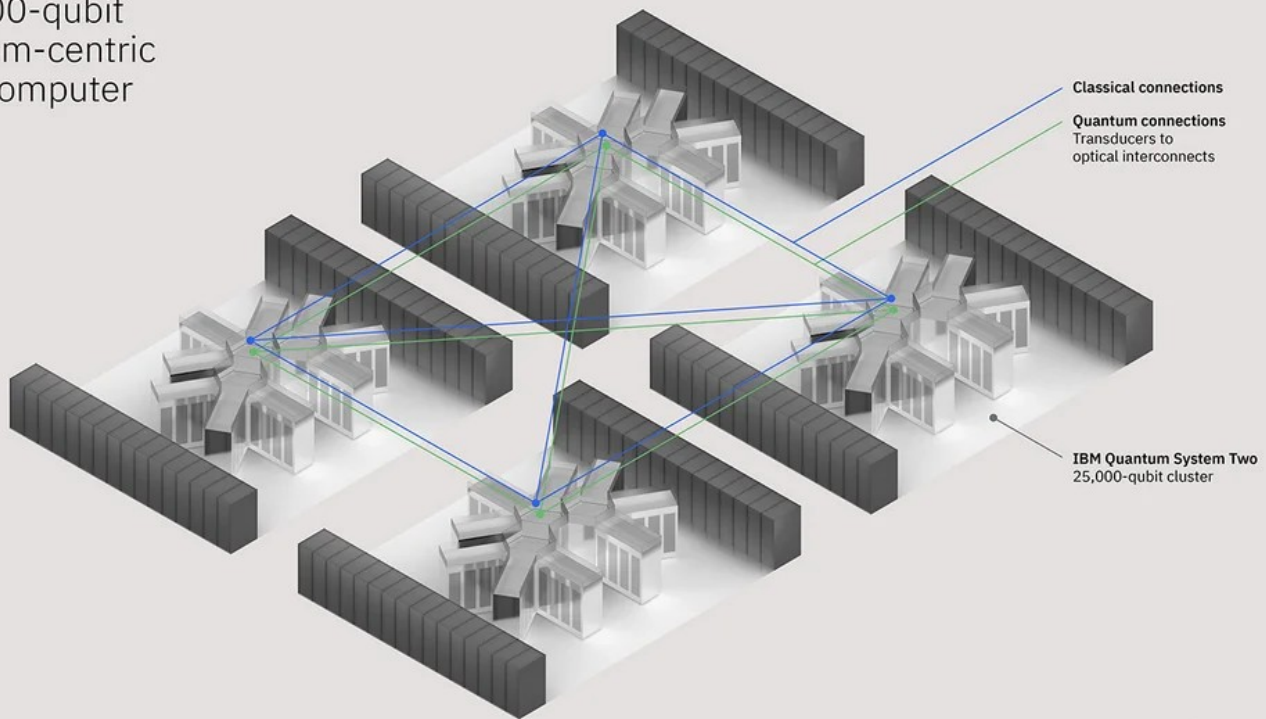
Need to factor

4096 bit semiprime:

8950429693485782969819159103361636787324531247595331455865576049528957603177763165691586761508274
8612428210789254796713033291402777424383964474525675340059592677596345495367727186879078511383357
66406911679216800465068765481022358269511066011043214241154792284849342476806301462838794988489104
4979706015770085049633781304347756129377783540108661429560151911005773387375216115635302836880772
7894927096506522360815072003186182464302107329240801651864447332432227480806121176546660163477961
2079087301378180468550758603979708720992301281518897296205081608907825465936077824761157221996661
909831625905207412997263777508871680607010649649306287447241547902349676376787491756994866954495
0361493264214164541577312583731983304075229287865529835971053244670543287945198597007230421194138
5987622009399852416597534382289605254398863710160238544899455274270168012833272387402646230414545
4385691054574768115474820051683248135374589610152876639436419219968226769548413114519425094561652
9523978521590091466213527524706661778437750247039074736641799486281849406242236762484545429225759
1818867787433314406963341724404237652561568658515029294655137151461974536474688780183636286034894
24424030802769700952150094244596873055650438391281230331589881854277

4096 bit semiprimes are not the largest number I've seen used for modern cryptography, either. I believe we will someday have quantum computers capable of factoring large numbers like this.

100,000-qubit
quantum-centric
supercomputer
—
2033



IBM Quantum

IBM plans on having a quantum-centric supercomputer in 2033 with 100,000 qubits. Is that enough to break RSA? It depends on how error prone those qubits are.

So, what about those Post-Quantum Cryptography algorithms that NIST has been working on?

Post-Quantum Cryptography (PQC) - where are we?

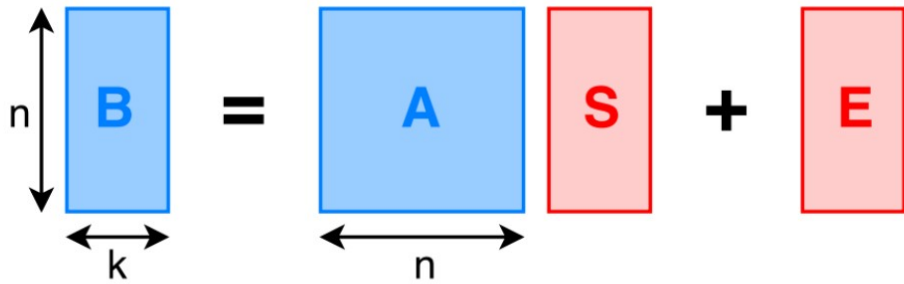
2022 NIST approves PQC encryption and signature algorithms

- CRYSTALS-Kyber (general encryption)
- CRYSTALS-Dilithium (signature)
- FALCON (signatures for smaller applications)
- SPHINCS+ (backup based on a different mathematical model)

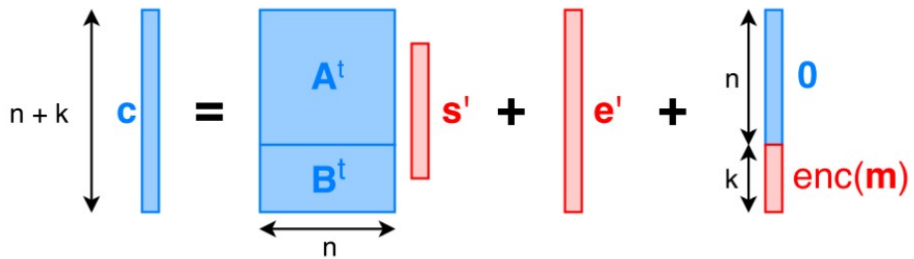
<Click> NIST made its approvals of the Quantum-Safe Algorithms <Click> CRYSTALS-Kyber for encryption, <Click> CRYSTALS-Dilithium, FALCON, and SPHINCS+ for signatures.

Public parameter $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n \times n})$

$\text{KeyGen}()$



$\text{Encrypt}_b(m \in \{0, 1\})$



CRYSTALS-Kyber for example, uses techniques created in 2005 called Learning with Errors. Instead of being based on factoring semiprimes, discrete logarithms, or elliptic curves, it uses lattices and introduces errors to make discovery difficult mathematically, even for large scale quantum computers without interference issues.

Post-Quantum Cryptography (PQC) - where are we?

- X25519Kyber768Draft00 readily available to test
 - Google Chrome
 - Firefox
 - BoringSSL
 - Nginx
- Cloudflare Research - <https://pq.cloudflare.com/research/>
- Signal - <https://signal.org/blog/pqxdh/>
- Apple iMessage - <https://security.apple.com/blog/imessage-pq3/>

These algorithms have been encoded for use in various applications for testing. And Cloudflare has released a tool where you can test. As mentioned before, Signal has adopted QSC, and so has Apple for its iMessage application.



Experiments

Chrome | chrome://flags

kyber

Reset all

119.0.6045.199

Available Unavailable

- TLS 1.3 hybridized **Kyber** support
This option enables a combination of X25519 and Kyber in TLS 1.3. – Mac, Windows, Linux, ChromeOS, Android, Fuchsia, Lacros
[#enable-tls13-kyber](#) Enabled

Enabling Kyber support in the latest versions of Google Chrome (Microsoft Edge, Samsung Internet, Opera, Brave, etc), for example, is easy to do.

Experiments Post-Quantum Key Agree

← → ↻ 🏠 🔒 pq.cloudflareresearch.com

Cloudflare Research: Post-Quantum Key Agreement

On essentially all domains served (1) through **Cloudflare**, including this one, **we have enabled** hybrid post-quantum key agreement. We are **rolling out support** for post-quantum key agreement for connection from Cloudflare to origins (3).

You are using `X25519Kyber768Draft00` which is **post-quantum secure**.

Deployed key agreements

Available with TLSv1.3 including HTTP/3 (QUIC)

Key agreement	TLS identifier
<code>X25519Kyber768Draft00</code>	<code>0x6399</code> (recommended) and <code>0xfe31</code> (obsolete)
<code>X25519Kyber512Draft00</code>	<code>0xfe30</code>

`X25519Kyber[xy]Draft00` is a **hybrid** of `X25519` and `Kyber[xy]Draft00` (in that order).

Software support

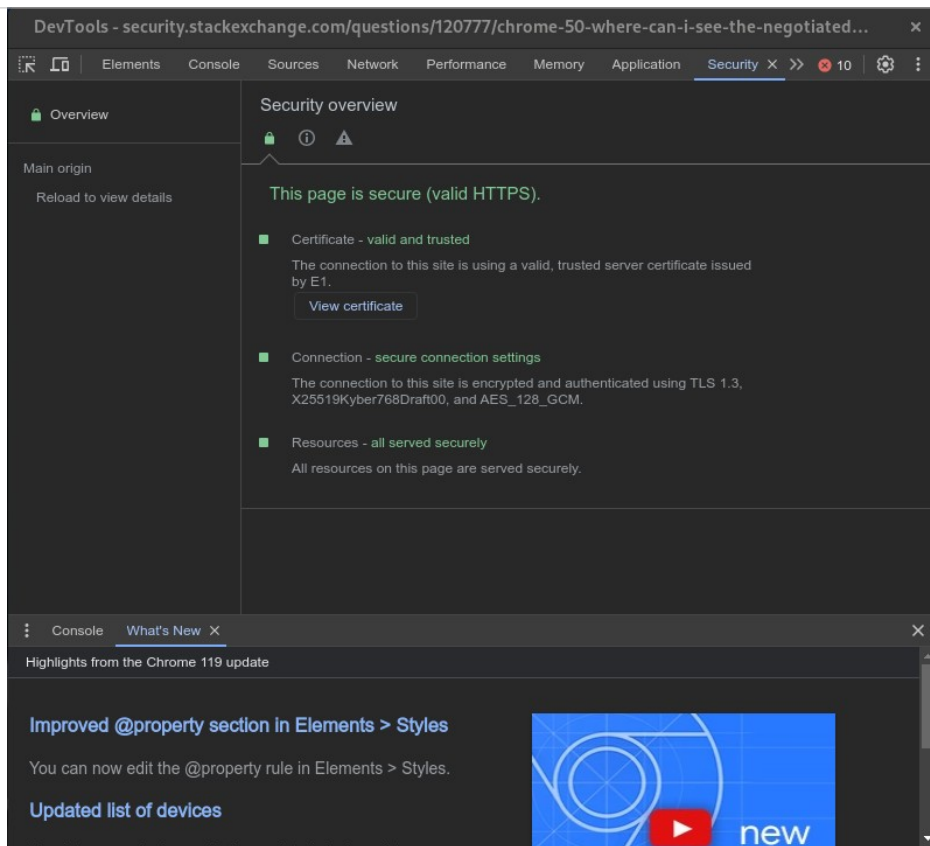
- Chrome 116+** if you turn on *TLS 1.3 hybridized Kyber support* (`enable-tls13-kyber`) in `chrome://flags`.

#CybersecPrivacy24

CYBERSECURITY AND PRIVACY PROFESSIONALS CONFERENCE AT EDUCAUSE



Once enabled, you can test your browser against Cloudflare's test site. Quantum-Safe Cryptography is here!



Here we have Stack Exchange which is using Kyber as well! You can access developer tools via <CTRL><SHIFT>-I in chrome and then look at the security tab to see what cipher you are using.

Sadly, this information is not available via the Chrome extension API, so we won't get a nice extension to tell us if we are quantum safe or not. You'll have to use the developer's tools.

What do we need to do?

1. Don't panic.
2. Educate yourself and your company on the risks.
3. Understand your environment.
 - a. Where do you use encryption?
 - b. What type of encryption do you use?
 - c. Can you update? Are your vendors working on implementing PQC?
 - d. Realize that you should be auditing your encryption usage anyway.



#CybersecPrivacy24

SECURITY AND PRIVACY PROFESSIONALS CONFERENCE AT EDUCAUSE

<Click> As with all things, don't panic, and don't let your company panic. Good leadership will show that calm, collective forward movement will prevail.

<Click> Educate yourself and your leadership on the risks you face. Make sure this is treated as a priority to plan for now. You have time, but don't become complacent.

<Click> Start auditing your environment now to understand your exposure. <Click> You may be surprised to find out where you use encryption, <Click> and the type of encryption you use. Are your certificates already expired? <Click> Are you still using certificates with SHA-1 signatures. <Click>

Nothing on this list should be new to a seasoned IT professional. Technology changes all the time and we have to constantly balance upgrading and updating against risk and productivity.

There are easy to use tools to scan your environment to see what cryptographic algorithms you are using and the health of your certificates.

```
[1]ep:~$ nmap -sV --script ssl-enum-ciphers --script ssl-cert www.example.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-12 20:54 EST
Nmap scan report for www.example.com (93.184.216.34)
Host is up (0.0055s latency).
Other addresses for www.example.com (not scanned):
2606:2800:220:1:248:1893:25c8:1946
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Edgecast CDN httpd (nyb/1DCD)
|_http-server-header: ECS (nyb/1DCD)
443/tcp   open  ssl/http  Edgecast CDN httpd (nyb/1DCD)
|_ssl-enum-ciphers:
|_TLSv1.0:
|_  ciphers:
|_  compressors:
|_    NULL
|_  cipher preference: server
|_TLSv1.1:
|_  ciphers:
|_    TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|_    TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp256r1) - A
|_    TLS_DHE_RSA_WITH_AES_128_CBC_SHA (dh 2048) - A
|_    TLS_DHE_RSA_WITH_AES_256_CBC_SHA (dh 2048) - A
|_    TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (dh 2048) - A
|_    TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (dh 2048) - A
|_    TLS_RSA_WITH_AES_256_CBC_SHA (rsa 2048) - A
|_    TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (rsa 2048) - A
|_    TLS_DHE_RSA_WITH_SEED_CBC_SHA (dh 2048) - A
|_  compressors:
|_    NULL
|_  cipher preference: server
|_TLSv1.2:
|_  ciphers:
|_    TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|_    TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (secp256r1) - A
|_    TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A
|_    TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
```

```

|_    TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (secp256r1) - A
|_    TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (secp256r1) - A
|_    TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (secp256r1) - A
|_    TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (secp256r1) - A
|_  compressors:
|_    NULL
|_  cipher preference: server
|_TLSv1.3:
|_  ciphers:
|_    TLS_AKE_WITH_AES_256_GCM_SHA384 (secp256r1) - A
|_    TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (secp256r1) - A
|_    TLS_AKE_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|_  cipher preference: server
|_  least strength: A
|_  ssl-cert: Subject: commonName=www.example.org/organizationName=InternetxC2
xA0Corporation\xC2xA0forxC2xA0Assigned\xC2xA0Names\xC2xA0andxC2
xA0Numbers/stateOrProvinceName=California/countryName=US
|_  Subject Alternative Name: DNS:www.example.org, DNS:example.net,
DNS:example.edu, DNS:example.com, DNS:example.org, DNS:www.example.com,
DNS:www.example.edu, DNS:www.example.net
|_  Issuer: commonName=DigiCert TLS RSA SHA256 2020
CA1/organizationName=DigiCert Inc/countryName=US
|_  Public Key type: rsa
|_  Public Key bits: 2048
|_  Signature Algorithm: sha256WithRSAEncryption
|_  Not valid before: 2023-01-13T00:00:00
|_  Not valid after: 2024-02-13T23:59:59
|_  MD5: 749bbbeb4a6cb23c205c9850b35bed6a
|_  SHA-1: f2aad73d32683b716d2a7d61b51c6d5764ab3899
|_  http-server-header:
|_    ECS (nyb/1D2E)
|_    ECS (nyb/1DCD)
|_  Tr19:
|_    closed bnetgame
|_    open h323q931?
|_    tcp closed rtmp
|_  service detection performed. Please report any incorrect results at
https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 189.80 seconds
```



#C

Here is an example of a simple scan of a host with the free open source tool, nmap. You can easily expand this to your entire environment and the script automatically scans around 1000 known ports that serve TLS and STARTTLS. It supports the majority of protocols, too.

What do we need to do?

1. Plan

- a. What can you upgrade?
- b. What can't you upgrade?
- c. Risk Analysis

2. Execute your Plan

3. Automation

- a. ACME - Automatic Certificate Management Environment
- b. SCEP - Simple Certificate Enrollment Protocol
- c. REST/API - Check with your certificate provider, InCommon supports this

Based on your encryption audit, <Click>make a plan. Find out <Click> what can be upgraded, and <Click> what can't. Do a <Click> risk analysis of the things that are too old to be updated, or just can't. Do you have a replacement plan for those items? <Click> then execute your plan when the technology becomes available.

<Click> Also, remember that automation for certificate deployment is important to consider. Chromium, the backend engine to most modern browsers (with the notable exception of Firefox) is proposing to reduce TLS Cert Life Span from 398 days to 90 days. This means you'll have to replace your certs every 90 days.


CYBER RISK

DARKREADING
TECHNOLOGY

News, news analysis, and commentary on the latest trends in cybersecurity technology.

Google Proposes Reducing TLS Cert Life Span to 90 Days

Organizations will likely have until the end of 2024 to gain visibility and control over their keys and certificates.

#Cyl  Dark Reading Staff, Dark Reading

🕒 2 Min Read

[Latest Articles in DR Technology](#)

Being able to deploy these in a large environment will really need to rely on automation. Setting this up now will mean it is all that much easier to deploy PQC when it is made available.

Although no official timeline has been released, it has been suggested that they may push this change before the end of 2024. This is something that I'm worried about.

Module-Lattice-based Key-Encapsulation Mechanism FIPS-203

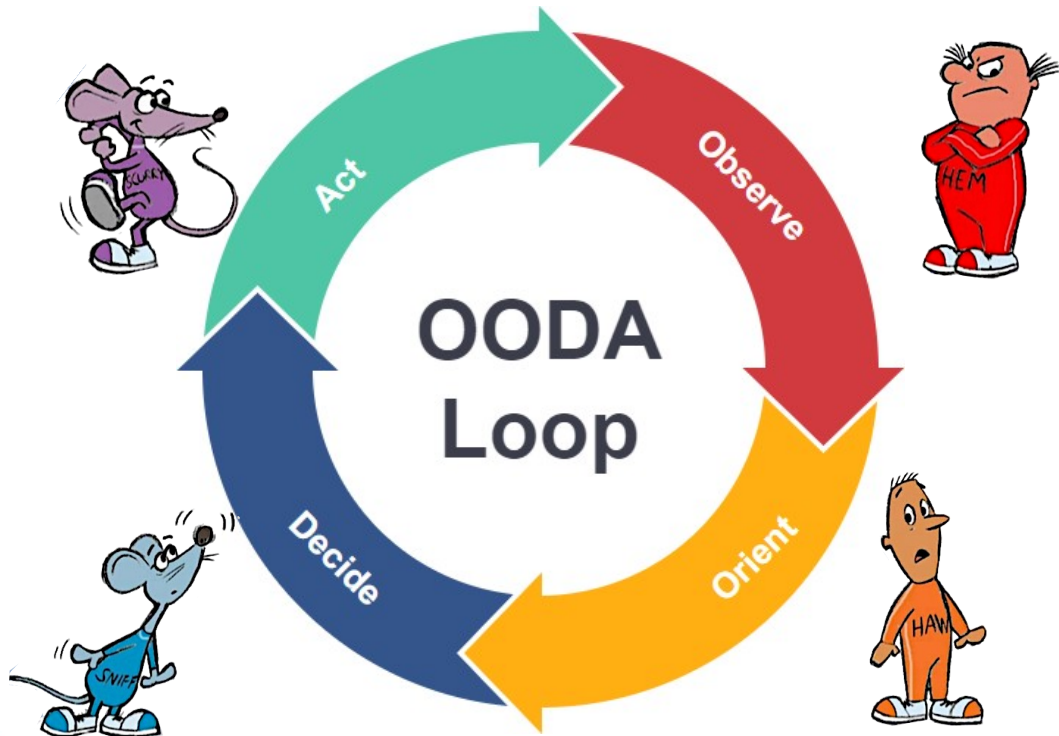
13. Qualifications. In applications, the security guarantees of a KEM only hold under certain conditions (see NIST SP 800-227 [1]). One such condition is the secrecy of several values, including the randomness used by the two parties, the decapsulation key, and the shared secret key itself. Users shall, therefore, guard against the disclosure of these values. While it is the intent of this standard to specify general requirements for implementing ML-KEM algorithms, **conformance to this standard does not ensure that a particular implementation is secure**. It is the responsibility of the implementer to ensure that any module that implements a key establishment capability is designed and built in a secure manner. Similarly, the use of a product containing an implementation that conforms to this standard does not guarantee the security of the overall system in which the product is used. The responsible authority in each agency or department shall ensure that an overall implementation provides an acceptable level of security. NIST will continue to follow developments in the analysis of the ML-KEM algorithm with its other cryptographic algorithm standards, **NIST will formally reevaluate this standard every five years**. Both this standard and possible threats that reduce the security provided through the use of this standard will undergo review by NIST as appropriate, taking into account newly available analysis and technology. In addition, the awareness of **any breakthrough in technology or any mathematical weakness of the algorithm will cause NIST to reevaluate** this standard and provide necessary revisions.



#CybersecPrivacy24

CYBERSECURITY AND PRIVACY
PROFESSIONALS CONFERENCE
AT EDUCAUSE

And remember, we are all in this boat together. NIST describes Module-Lattice-based Key-Encapsulation Mechanism in FIPS-203. Note that NIST already plans to re-evaluate the protocol every 5 years or if needed sooner. Once again, repeating the cyclical nature of information security



The OODA loop (observe, orient, decide, act) was developed by military strategist and United States Air Force Colonel John Boyd. Remaining agile and following the steps of Observe <Click>, Orient <Click>, Decide <Click>, and Act <Click> will keep us always moving in the right direction.



Thank you!