Lecture 19
Cryptography

Cryptography

- some history
  - Caesar cipher, rot13
  - substitution ciphers, etc.
  - Enigma (Turing)
- modern secret key cryptography
  - DES, AES
- public key cryptography
  - RSA, digital signatures, cryptographic hashing
- cryptography in practice
  - e-commerce
  - Tor browser
  - Bitcoin
  - politics

Cryptography basics

- Alice & Bob want to exchange messages
  - keeping the content secret
  - though not the fact that they are communicating
- they need some kind of secret that scrambles messages
  - makes them unintelligible to bad guys but intelligible to good guys

- the secret is a "key" (like a password)
  - known only to the communicating parties
  - that is used to do the scrambling and unscrambling

  - for Caesar cipher, the "key" is the amount of the shift (A => D, etc.)
  - for substitution ciphers, the key is the permutation of the alphabet
  - for Enigma, key is wiring and position of wheels plus settings of patches

  - for modern ciphers, the key is a large integer used as part of an intricate algorithmic operation on the bits of the message

Modern secret key cryptography

- **messages encrypted and decrypted with a shared secret key**
  - usually the same key for both operations ("symmetric")
- **encryption/decryption algorithm is known to adversaries**
  - "security by obscurity" *does not work*
- **attacks**
  - decrypt specific message(s) by analysis
    various combinations of known or chosen plaintext and ciphertext
  - determine key by "brute force" (try all possible keys)
- **if key is compromised, all past and future messages are compromised**
- **big problem: key distribution**
  - need a secure way to get the key to both/all parties
    diplomatic pouches, secret agents, …
  - doesn't work when the parties don't know each other
  - or have no possible channel for exchanging a secret key
  - or when want to exchange secret messages with many different parties
    e.g., credit card numbers on Internet

The secrecy is in the key

"Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi."

(The system must not require secrecy, and it does not matter if it falls into the hands of the enemy.)

Auguste Kerckhoffs, "La cryptographie militaire",
Journal des sciences militaires, vol IX, pp 5-38, Janvier 1883.

- we have to assume that the bad guys know how the encryption and decryption operate
- “security by obscurity” does not work
- Claude Shannon: "The enemy knows the system."

DES and AES

- Data Encryption Standard (DES)
  - developed ~1977 by IBM, with NSA involvement
  - widely used, though lingering concerns about trap doors
  - by 1995, 56-bit key was much too short:
    could exhaustively test all keys in a few hours with special hardware
  - "triple DES" used 3 DES encryptions to increase effective key length
    but not enough to prevent brute-force attacks

- Advanced Encryption Standard (AES)
  - result of an international competition run by NIST (www.nist.gov/aes)
  - completely open: algorithms and analyses in public domain
  - Rijndael: winning algorithm selected October 2000
    approved as official US government standard
  - 128, 192, 256-bit keys
  - fast in both hardware and software implementations

Advanced Encryption Standard (AES)

- Rijndael
  - Joan Daemen & Vincent Rijmen, Belgium
  - 128, 192, 256-bit keys

The big problem: key distribution

- we need a secure way to get the key to all parties
  - diplomatic pouches, secret agents, steganography, …
- doesn't work when there is no channel for exchanging a secret key
- or when two parties don't know each other
- or if we need to exchange secret messages with many different parties
  - e.g., credit card numbers on Internet

Public key cryptography

- a fundamentally new idea
  - Diffie & Hellman (USA, 1976); invented earlier in England but kept secret
- each person has a (public key, private key) pair
  - the public and private keys are mathematically related
  - a message encrypted with one key can only be decrypted with the other key
- public key is published, visible to everyone
- private key is secret, known only to owner

- Alice sends a secret message to Bob by
  - encrypting it with *Bob's* public key
  - only Bob can decrypt it, using his private key
- Bob sends a secret reply to Alice by
  - encrypting it with *Alice's* public key
  - only Alice can decrypt it, using her private key
- Eve knows Alice and Bob are talking
  - but can't decrypt what they are saying

RSA public key cryptographic algorithm

- most widely used public key system
- invented by Ron Rivest, Adi Shamir, Len Adleman, 1977
  - patent expired Sept 2000, now in public domain

- based on (apparent) difficulty of factoring very large integers
  - "large" >= 1024 bits ~ 300 digits
  - public key based on product of two large (secret) primes
  - encrypting and decrypting require knowledge of the factors
- slow, so usually use RSA to exchange a secret "session key"
  - session key used for secret key encryption with AES
  - used by SSH for secure login
  - used by browsers for secure exchange of credit card numbers

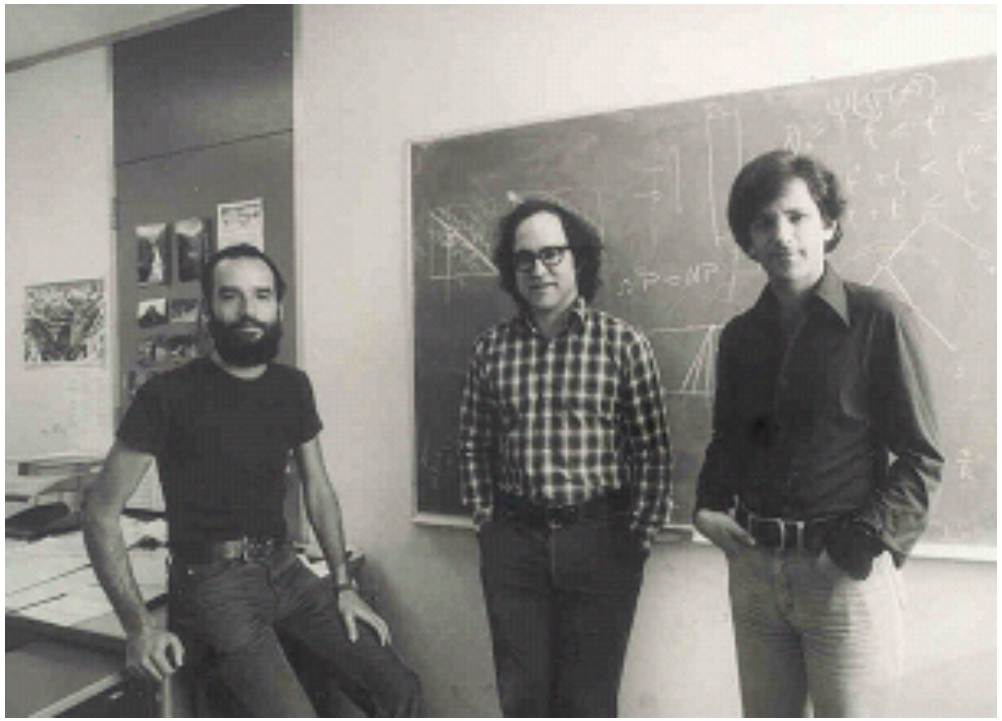
    https: http with encryption
  - SSL (Secure Sockets Layer) or TLS (Transport Layer Security)

    used to encrypt TCP/IP

Public key personae

Martin Hellman,
Whitfield Diffie

Adi Shamir,
Ron Rivest,
Len Adleman



Intuition for public-key crypto algorithms

- one-way function: easy to compute, hard to invert
  - multiplication vs factoring; expontiation vs logarithm

- need a trap door in one-way function that makes inverse easy

- if you know something extra
  - e.g., one of the factors or the base or exponent
  - then you can decrypt easily
- if you don't know the something extra
  - you can't decrypt easily

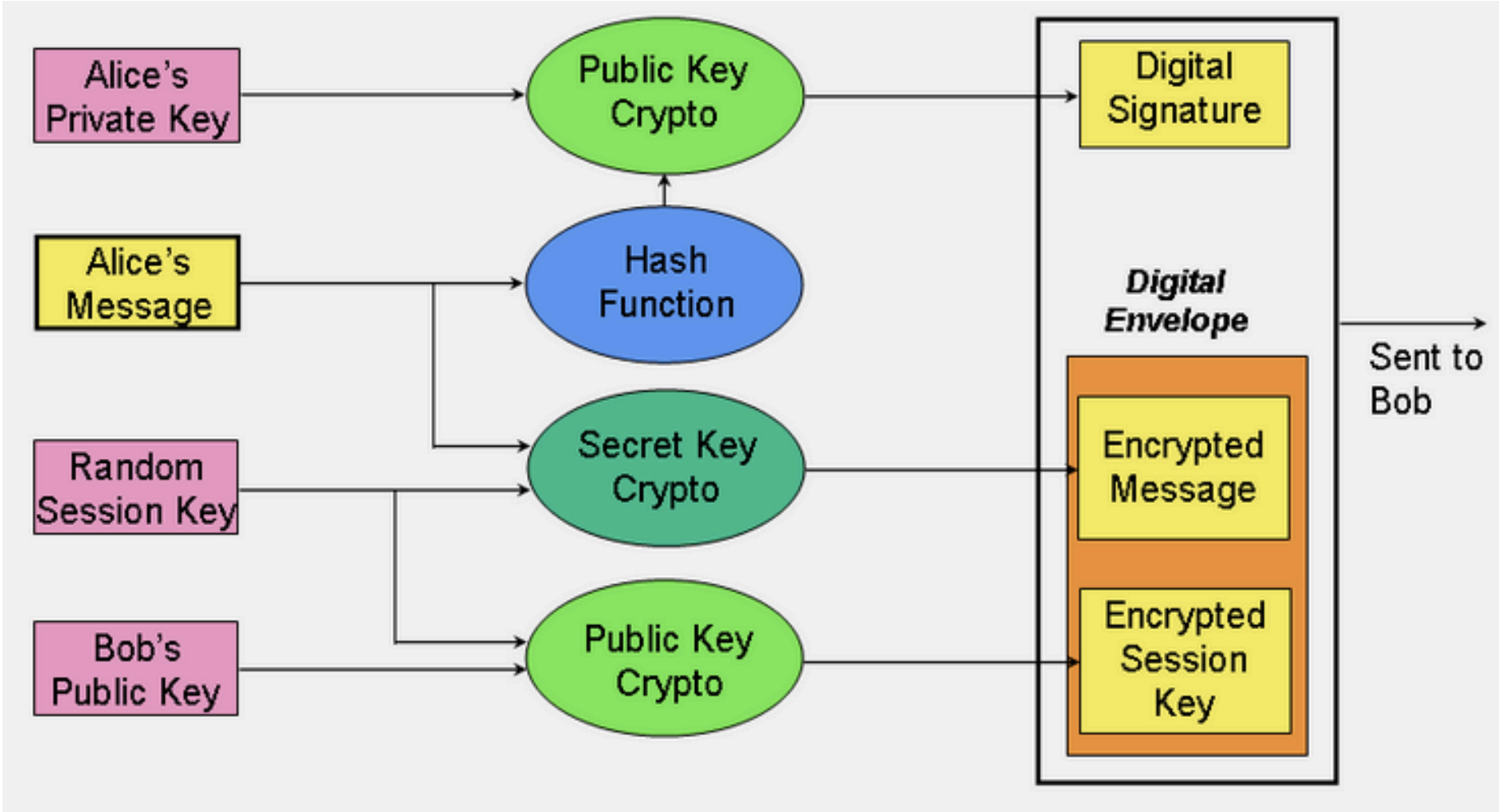
Digital signatures

- can use public key cryptography for digital signatures
  - if Alice encrypts a message with her private key
  - and it decodes properly with her public key
  - it had to be Alice who encoded it
- signature can be attached to a message
  - Alice encrypts a message with her private key
  - Alice encrypts the result with Bob's public key
  - only Bob can decrypt this (with his private key)
    but it won't make any sense yet
  - Bob then decrypts it with Alice's public key
  - if it decodes properly, it had to be Alice who encrypted it originally
- necessary properties of digital signatures
  - can only be done by the right person: can't be forged
  - can't re-use a signature to sign something else
  - signature attached to a document: signs specific contents
  - signature can't be repudiated

Cryptographic / secure hashing

- digital signature usually done by signing a ”secure hash" or “message digest” of a document, not the document itself
- secure hash algorithm reduces input data to a comparatively short number such that
  - any change to the original document produces a completely different hash
  - can't deduce the original document from the number
  - can't find another document that has the same hash
- current secure hash algorithms
  - MD5 (Rivest, MIT): 128 bits
  - SHA-1 (US government standard): 160 bits
  - SHA-2 (also standard): family of 224, 256, 384, or 512 bits
- international competition to create a new secure hash, SHA-3
  - analogous to AES competition (also run by NIST)
  - first round submissions 10/08, final round 12/10,
  - winner announced in Oct 2012; official 2015

Digital signature and envelope



RSA factoring challenge (1991; ended 2007)

RSA number	Decimal digits	Binary digits	Cash prize offered	Factored on	Factored by
RSA-100	100	330	US$1,000[4]	April 1, 1991[5]	Arjen K. Lenstra
RSA-110	110	364	US$4,429[4]	April 14, 1992[5]	Arjen K. Lenstra and M.S. Manasse
RSA-120	120	397	US$5,898[4]	July 9, 1993[6]	T. Denny *et al.*
RSA-129 [**]	129	426	US$100	April 26, 1994[5]	Arjen K. Lenstra *et al.*
RSA-130	130	430	US$14,527[4]	April 10, 1996	Arjen K. Lenstra *et al.*
RSA-140	140	463	US$17,226	February 2, 1999	Herman te Riele *et al.*
RSA-150	150	496		April 16, 2004	Kazumaro Aoki *et al.*
RSA-155	155	512	US$9,383[4]	August 22, 1999	Herman te Riele *et al.*
RSA-160	160	530		April 1, 2003	Jens Franke *et al.*, University of Bonn
RSA-170 [*]	170	563		December 29, 2009	D. Bonenberger and M. Krone [***]
RSA-576	174	576	US$10,000	December 3, 2003	Jens Franke *et al.*, University of Bonn
RSA-180 [*]	180	596		May 8, 2010	S. A. Danilov and I. A. Popovyan, Moscow State University[7]
RSA-190 [*]	190	629		November 8, 2010	A. Timofeev and I. A. Popovyan
RSA-640	193	640	US$20,000	November 2, 2005	Jens Franke *et al.*, University of Bonn
RSA-200 [*] ?	200	663		May 9, 2005	Jens Franke *et al.*, University of Bonn
RSA-210 [*]	210	696		September 26, 2013[8]	Ryan Propper
RSA-704 [*]	212	704	US$30,000	July 2, 2012	Shi Bai, Emmanuel Thomé and Paul Zimmermann
RSA-220 [*]	220	729		May 13, 2016	S. Bai, P. Gaudry, A. Kruppa, E. Thomé and P. Zimmermann
RSA-230 [*]	230	762		August 15, 2018	Samuel S. Gross, Noblis, Inc.
RSA-232	232	768			
RSA-768 [*]	232	768	US$50,000	December 12, 2009	Thorsten Kleinjung *et al.*
RSA-240 [*]	240	795		Dec 2, 2019[9]	F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimn
RSA-250	250	829			

Properties of public/private keys

- can't deduce the public key from the private, or vice versa
- can't find another encryption key that works with the decryption key
- keys are long enough that brute force search is infeasible

- nasty problems:
  - if a key is lost, all messages and signatures are lost
  - if a key is compromised, all messages and signatures are compromised
  - it's hard to revoke a key
  - it's hard to repudiate a key (and hard to distinguish that from revoking)

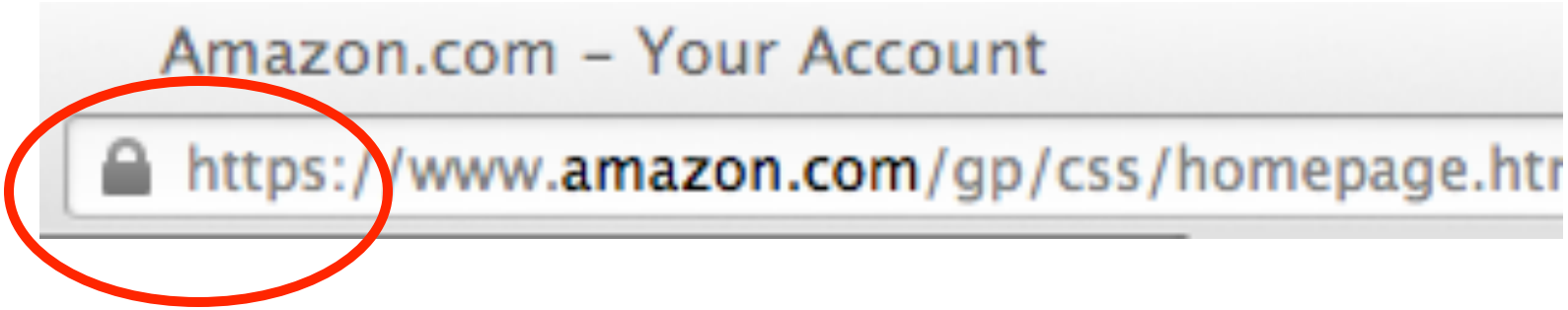
- authentication
  - how do you know who you are talking to? is that really Alice's public key?
  - public key infrastructure, web of trust, digital certificates

Encrypted transactions, online shopping

- browser says "prove that you're really Amazon"
- Amazon says "here's my signed certificate from a CA"
  - encrypted with the CA's private key
- browser decrypts certificate with CA's public key

- browser generates a random key, encrypts it with Amazon's public key, sends it to Amazon

- browser and Amazon use AES to talk securely



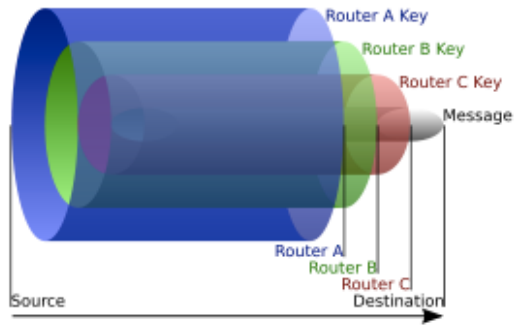
Tor: The Onion Router

- anonymous routing through the Internet using TCP
  - receiver can't determine the sender's address

- sender creates a random path through a network of Tor relays
  - path is changed frequently
- each part of the path is encrypted
  - separate encryption keys for each hop
- each relay only knows who gave it data and who it sends data to
  - no relay knows the whole path
- messages are wrapped up with nested encryptions, one for each component of the path
  - each relay removes one layer of encryption before passing it on
- potentially vulnerable to some attacks
  - traffic correlation at end points
  - exit nodes can be blocked or monitored

Data structure [edit]

An onion is the data structure formed by "wrapping" a message with successive layers of encryption to be decrypted ("peeled" or "unwrapped") by as many intermediary computers as there are layers before arriving at its destination. The original message remains hidden as it is transferred from one node to the next, and no intermediary knows both the origin and final destination of the data, allowing the sender to remain anonymous.[12]



In this example onion, the source of the data sends the onion to Router A, which removes a layer of encryption to learn only where to send it next and where it came from (though it does not know if the sender is the origin or just another node). Router A sends it to Router B, which decrypts another layer to learn its next destination. Router B sends it to Router C, which removes the final layer of encryption and transmits the original message to its destination.

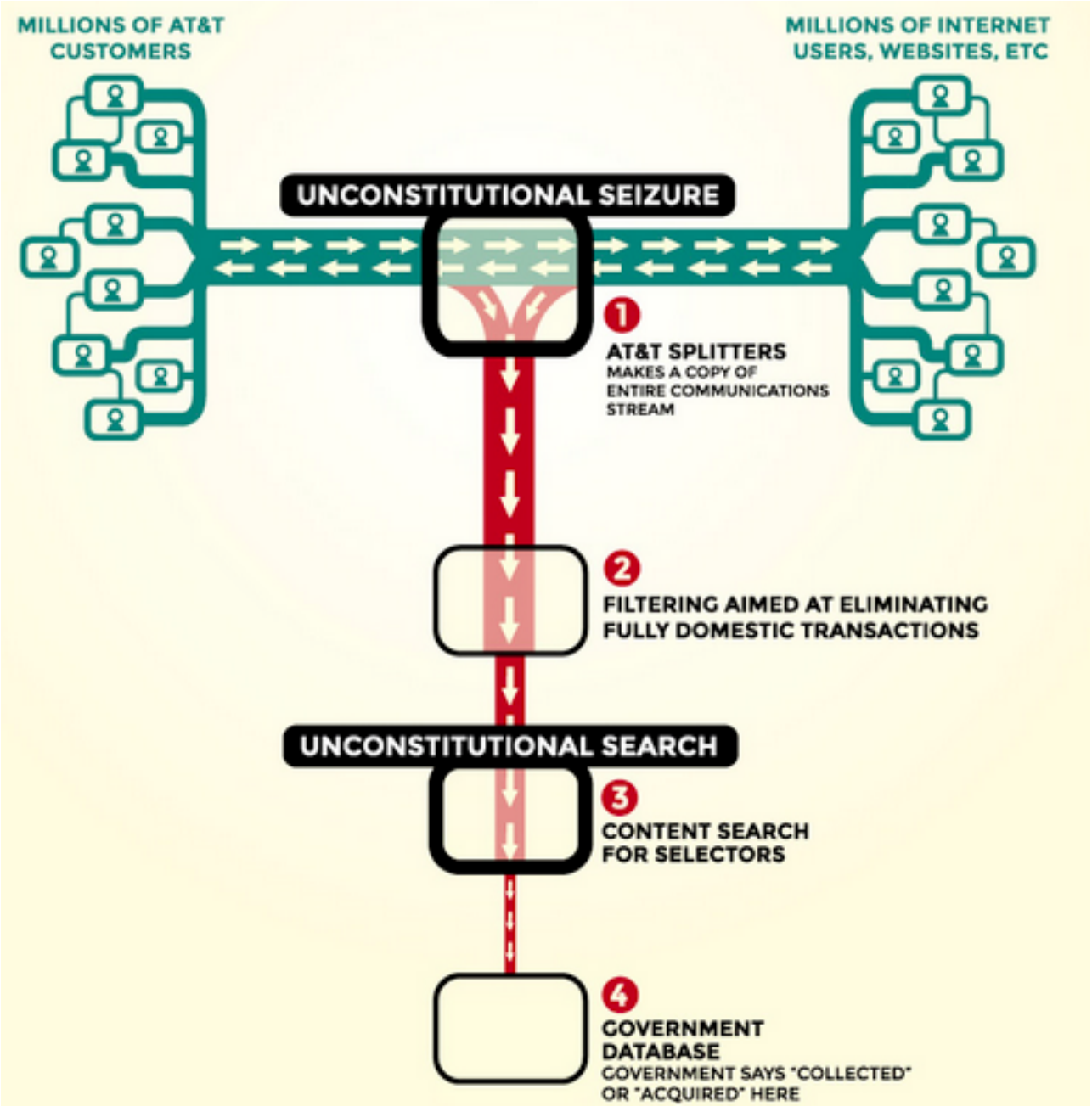
Onion creation and transmission [edit]

To create and transmit an onion, the originator selects a set of nodes from a list provided by a "directory node". The chosen nodes are arranged into a path, called a "chain" or "circuit", through which the message will be transmitted. To preserve the anonymity of the sender, no node in the circuit is able to tell whether the node before it is the originator or another intermediary like itself. Likewise, no node in the circuit is able to tell how many other nodes are in the circuit and only the final node, the "exit node", is able to determine its own location in the chain.[12]

Using asymmetric key cryptography, the originator obtains a public key from the directory node to send an encrypted message to the first ("entry") node, establishing a connection and a shared secret ("session key"). Using the established encrypted link to the entry node, the originator can then relay a message through the first node to a second node in the chain using encryption that only the second node, and not the first, can decrypt. When the second node receives the message, it establishes a connection with the first node. While this extends the encrypted link from the originator, the second node cannot determine whether the first node is the originator or just another node in the circuit. The originator can then send a message through the first and second nodes to a third node, encrypted such that only the third node is able to decrypt it. The third, as with the second, becomes linked to the originator but connects only with the second. This process can be repeated to build larger and larger chains, but is typically limited to preserve performance.[12]

When the chain is complete, the originator can send data over the Internet anonymously. When the final recipient of the data sends data back, the intermediary nodes maintain the same link back to the originator, with data again layered, but in reverse such that the final node this time removes the first layer of encryption and the first node removes the last layer of encryption before sending the data, for example a web page, to the originator.[12]

Domestic Internet Backbone Surveillance (eff.org)



(TS//SI//NF) FAA702 Operations

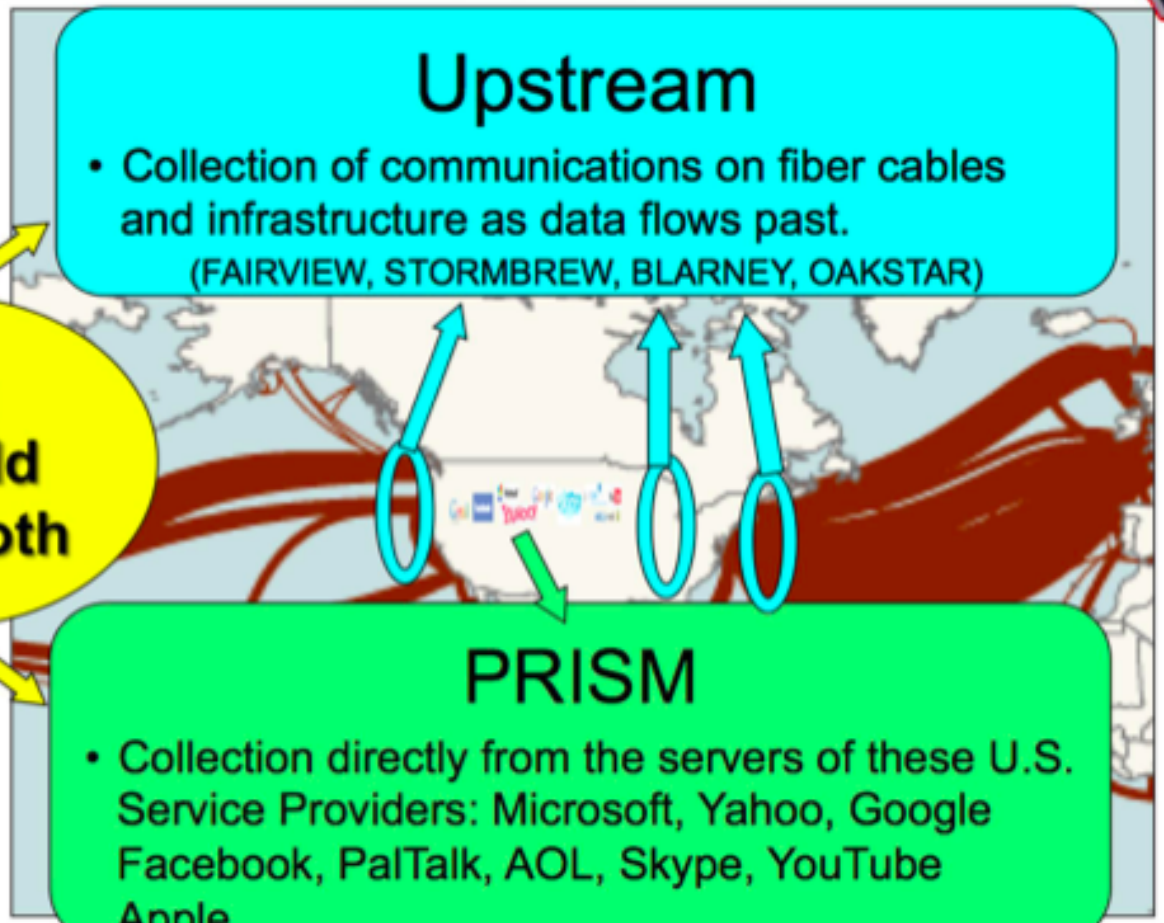
Two Types of Collection

Upstream

- Collection of communications on fiber cables and infrastructure as data flows past.

(FAIRVIEW, STORMBREW, BLARNEY, OAKSTAR)

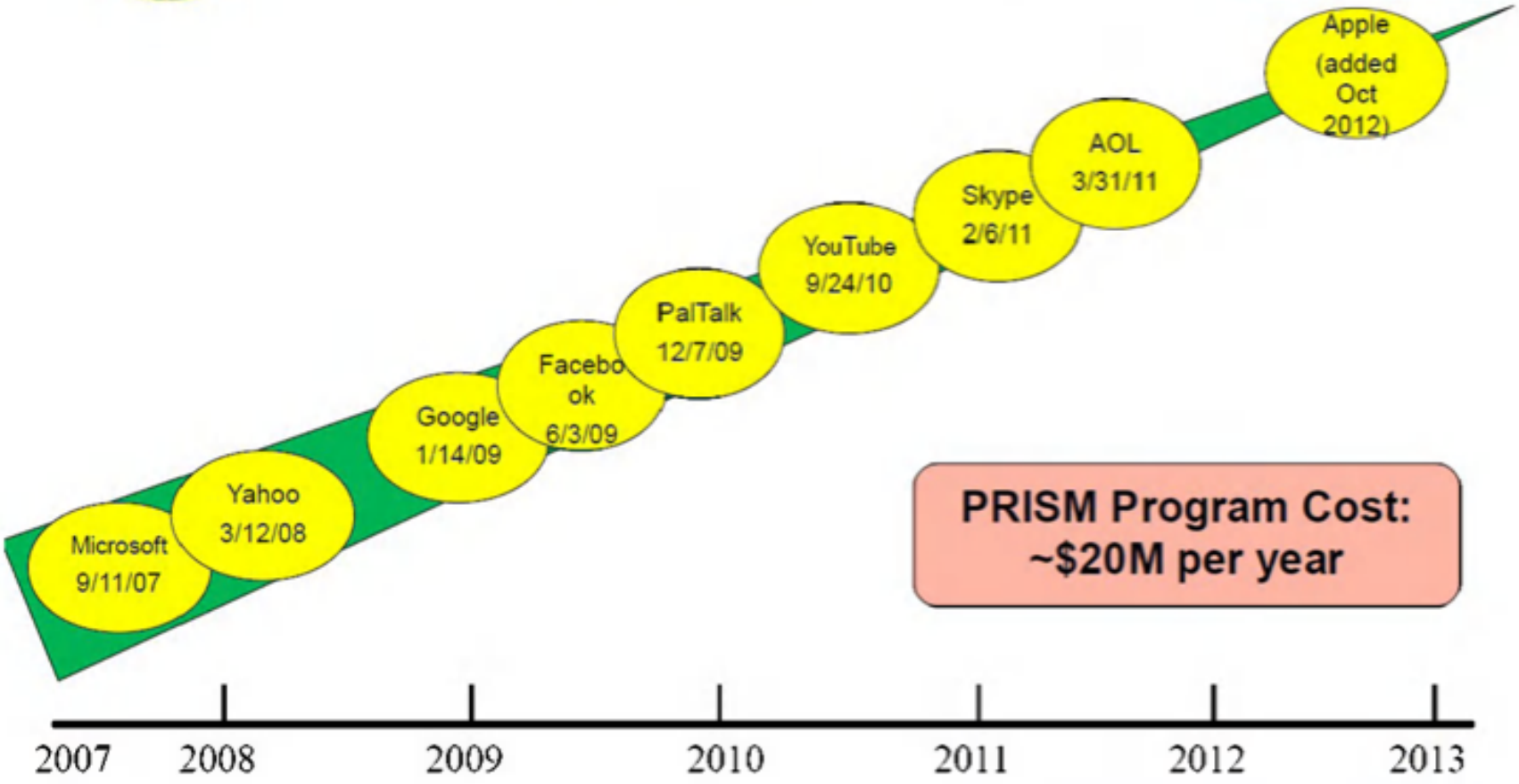
You Should Use Both



PRISM

- Collection directly from the servers of these U.S. Service Providers: Microsoft, Yahoo, Google Facebook, PalTalk, AOL, Skype, YouTube Apple.

(TS//SI//NF) Dates When PRISM Collection Began For Each Provider



PRISM Program Cost: ~$20M per year

(TS//SI//NF) PRISM Collection Details

Current Providers

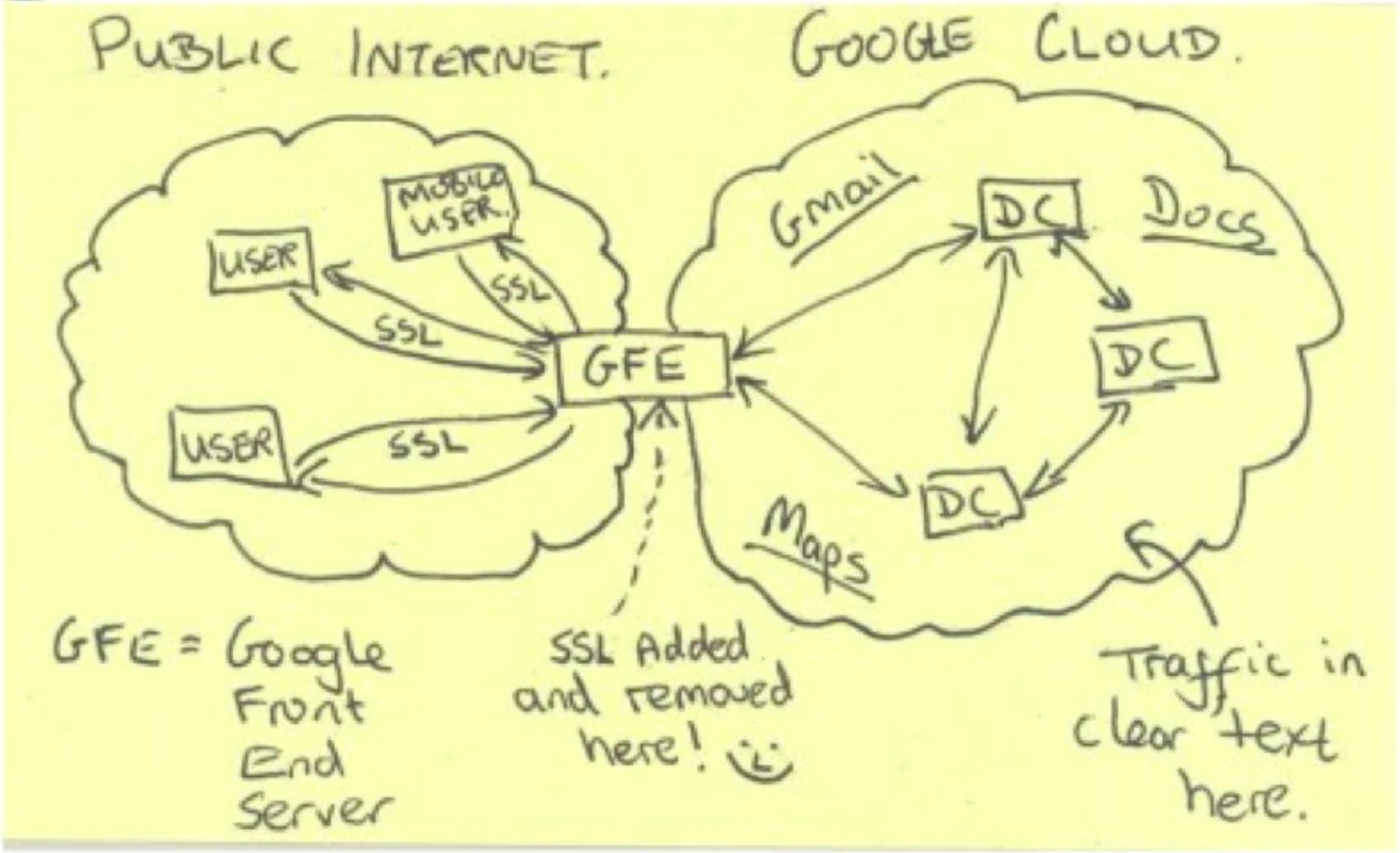
- Microsoft (Hotmail, etc.)
- Google
- Yahoo!
- Facebook
- PalTalk
- YouTube
- Skype
- AOL
- Apple

What Will You Receive in Collection (Surveillance and Stored Comms)? It varies by provider. In general:

- E-mail
- Chat – video, voice
- Videos
- Photos
- Stored data
- VoIP
- File transfers
- Video Conferencing
- Notifications of target activity – logins, etc.
- Online Social Networking details
- **Special Requests**

Complete list and details on PRISM web page:
Go PRISMFAA

Current Efforts - Google



Bitcoin digital currency (2009; see bitcoin.org et al)

- exists only in digital form: nothing physical like gold
  - no central authority or control
  - anonymous ownership and transfer
  - value fluctuates wildly
- how are bitcoins created?
- how is ownership validated & transferred without double spending?

- blockchain: shared public ledger of *all* transactions
- a transaction transfers value from one wallet to another
  - signed digitally by the sender
  - broadcast via peer to peer network so block chain can be updated
- "mining" confirms transactions by adding them to block chain
  - competitive distributed consensus algorithm
  - takes work to confirm; new bitcoins created as a reward
  - blocks are protected by cryptographic hashing; each new one depends on previous ones

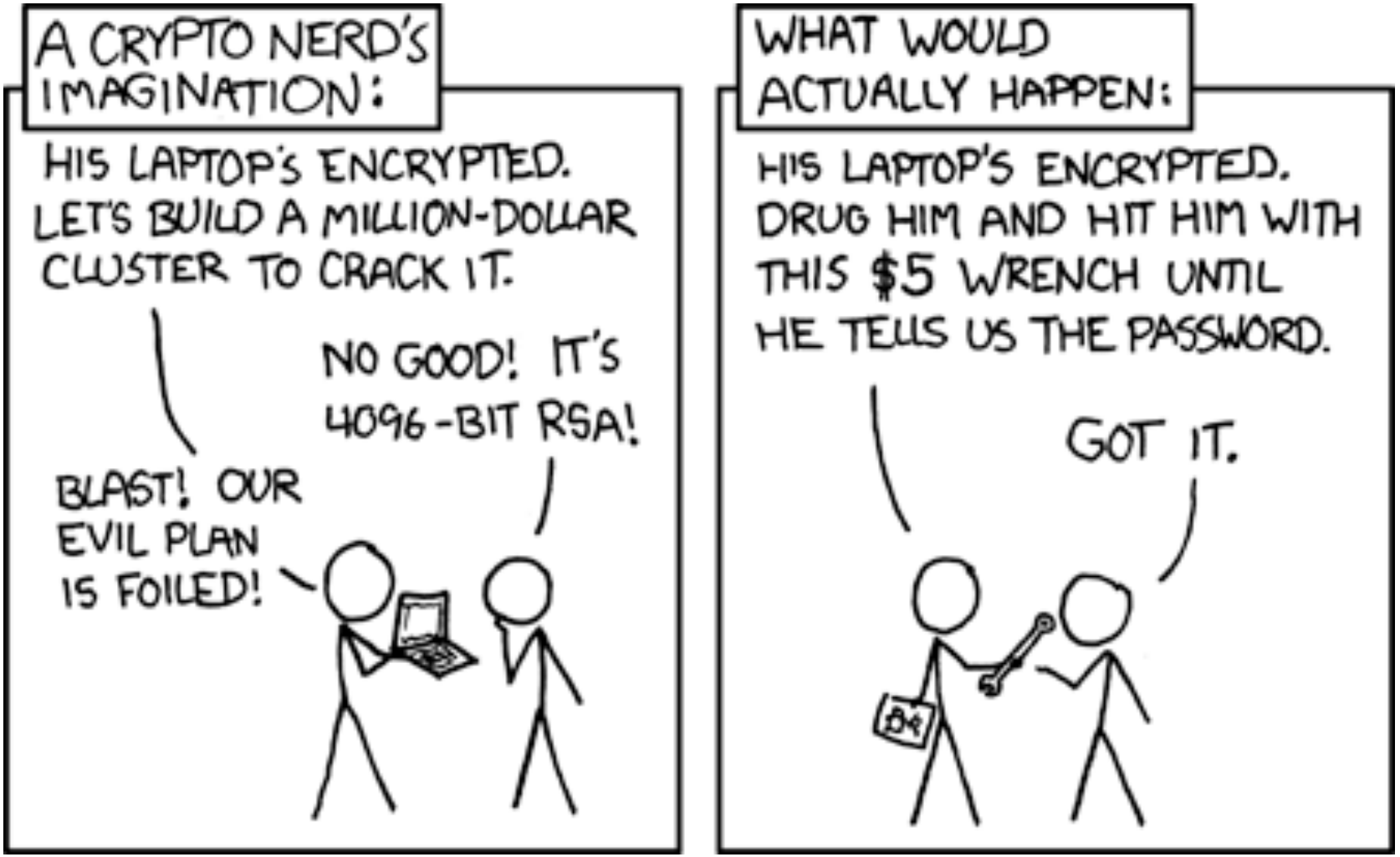
Crypto politics

- cryptographic techniques as weapons of war?
  - until recently, (strong) cryptography was classified as "munitions" in USA
  - falls under International Traffic in Arms Regulations and follow-ons
- export control laws prohibited export of cryptographic code
  - though it was ok to export books and T-shirts with code
    and everyone else in the world had it anyway
  - changed during 2000, but there are still restrictions
- does the government have the right/duty ...
  - to control cryptographic algorithms and programs?
  - to require trapdoors, key escrow, or similar mechanisms?
  - to prevent reverse-engineering of cryptographic devices?
  - to prevent research in cryptographic techniques?
- do corporations have the right ...
  - to prevent publication of cryptographic techniques?
  - to prevent reverse-engineering of cryptographic devices?
- how do we balance individual rights, property rights, & societal rights?

Summary of crypto

- secret/symmetric key algorithms: DES, AES
  - key distribution problem: everyone has to have the key
- public key algorithms: RSA, ...
  - solves key distribution problem, but authentication is still important
  - also permits digital signatures
  - much slower than secret key, so used mainly for key exchange
- security is entirely in the key
  - “security by obscurity” does not work: bad guys know everything
  - brute force attacks work if keys are too short or easy
- good cryptography is hard
  - you can't invent your own methods
  - you can't trust “secret” or proprietary methods
- people are the weak link
  - complicated or awkward systems will be subverted, ignored or misused
  - social engineering attacks are effective

    ignorance, incompetence, misguided helpfulness
- if all else fails, try bribery, burglary, blackmail, brutality


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Cryptography is important

- it protects our privacy and security
  - access to computers
  - email
  - online shopping, banking, taxes
  - electronic voting
  - ...

- it can restrict our rights and freedoms
  - digital rights management: limits on what we can do with music, movies, software, ...

- it helps good guys and bad guys alike