Warning - Ethics, Law, and University Policies

To defend a system you need to be able to think like an attacker, and that includes understanding techniques that can be used to compromise security. However, using those techniques in the real world may violate the law or the University's rules, and it may be unethical. Under some circumstances, even probing for weaknesses may result in severe penalties, up to and including expulsion, civil fines, and jail time. Our policy in this class is that you must respect the privacy and property rights of others at all times, **or else you will fail the course**.

Acting lawfully and ethically is your responsibility. Carefully read the Computer Fraud and Abuse Act (CFAA), a federal statute that broadly criminalizes computer intrusion. This is one of several laws that govern "hacking." Understand what the law prohibits — you don't want to end up like this guy. If in doubt, we can refer you to an attorney.

Please review CU's acceptable use policy of IT resources for guidelines concerning proper use of information technology, as well as the Engineering Honor Code.

Source: Eric Wustrow

Material

- Security overview
- Encryption techniques, one time pad, symmetric encryption, asymmetric encryption
- Diffie-Hellman
- Hashes, Message Authentication Codes (MACs)
- Attack vectors: AES, key, man-in-the-middle, replay
- Key protection, hardware techniques
- Side-channel attacks
- Chain of Trust
- Examples of poor security implementations
- How web browsers establish a secure connection
- Blockchains

MEANT TO SAY "CRYPTO CURRENCIES"

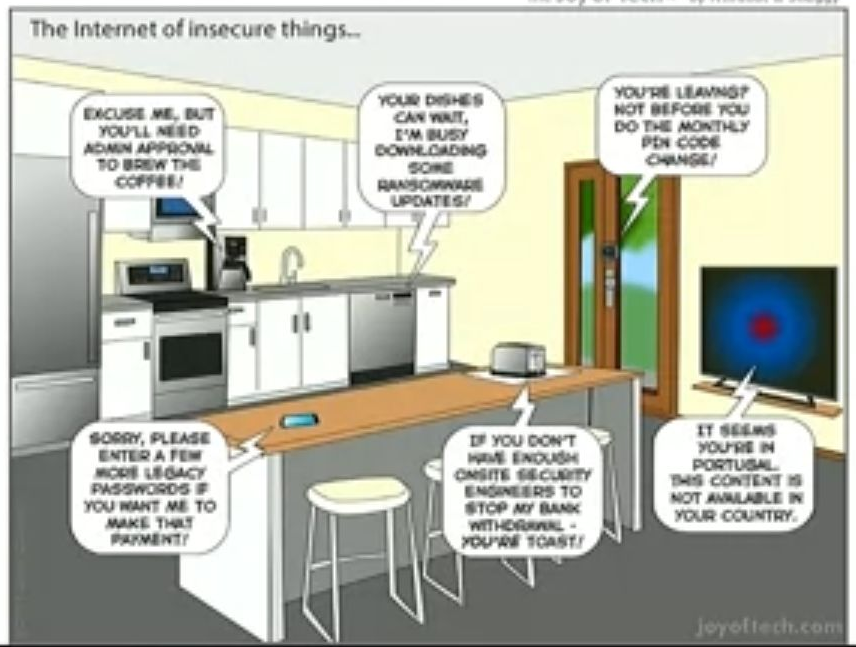
It's the basis of bitcoin and other cyber currencies.

Learning Outcomes

- Develop a "Security" mindset
- Address security at all levels and at all interfaces in a system
  - Make it difficult for attackers to walk through the door.
- Difference between symmetric and asymmetric encryption, Diffie-Hellman, Hashes, MACs, key protection schemes, man-in-the-middle and replay attacks
- Awareness of US security standards

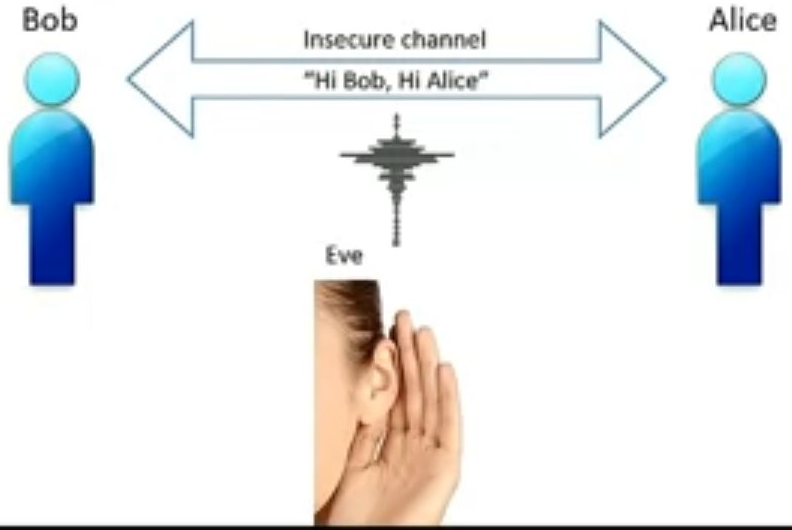
around the world but I think many countries follow the US NIST standards.

But first, a comic



Take a second to read that. Is it big enough to see?

What does it mean to be secure?



an insecure channel and we've got this person Eve eavesdropping on their conversation,

The Security Mindset - Bruce Schneier

- https://www.schneier.com/blog/archives/2008/03/the_security_mi_1.html

"Security requires a particular mindset. Security professionals -- at least the good ones -- see the world differently. They can't walk into a store without noticing how they might shoplift. They can't use a computer without wondering about the security vulnerabilities. They can't vote without trying to figure out how to vote twice. They just can't help it.

SmartWater is a liquid with a unique identifier linked to a particular owner. "The idea is for me to paint this stuff on my valuables as proof of ownership," I wrote when I first learned about the idea. "I think a better idea would be for me to paint it on *your* valuables, and then call the police." Really, we can't help it.

This kind of thinking is not natural for most people. It's not natural for engineers. Good engineering involves thinking about how things can be made to work; **the security mindset involves thinking about how things can be made to fail**. It involves thinking like an attacker, an adversary or a criminal. You don't have to exploit the vulnerabilities you find, but if you don't see the world that way, you'll never notice most security problems."

linked to his website and he says,

The Security Mindset - Dave Sluiter

- When working in security, it is an unwise mental mindset to make statements such as: "That's impossible", or "No one will ever figure this out" and other such absolute statements.
- A better mindset is one that blurs the line between TRUE and FALSE, mental positions such as likely/unlikely, probable/improbable, practical/impractical.
- The world is full of some very clever and well funded people.
  - Examples:
    - WWII German Enigma machine
    - US NSA
    - Israeli Mossad
- There is no 100% security, only approaches/solutions deemed "good enough".
- "Security through obscurity is not security" - courtesy of Don Matthews

no one's ever going to figure it out."

Kerckhoffs's Principle

- Stated by Dutch cryptographer Auguste Kerckhoff in the 19th century:
  - A crypto system should be secure even if everything about the system, **except the key**, is public knowledge.

to keep in the back of your mind when you're working in security.

The Security Mindset

- What can we learn from wood-block puzzles?

out the first time.

The Security Mindset

- What can we learn from wood-block puzzles?
- How to think "orthogonal" or unconventional

understanding where the pieces are, fixed on the inside,

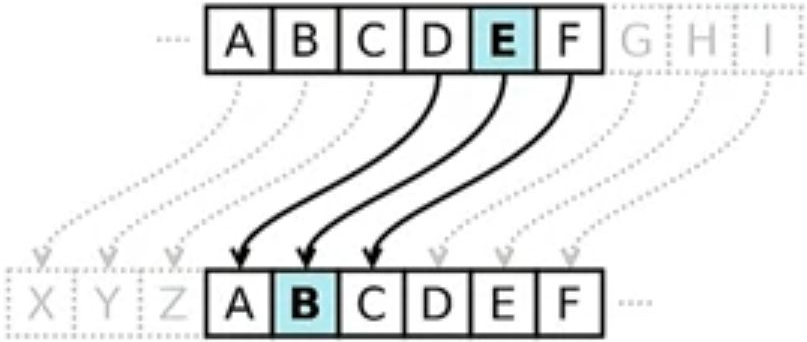
Terminology

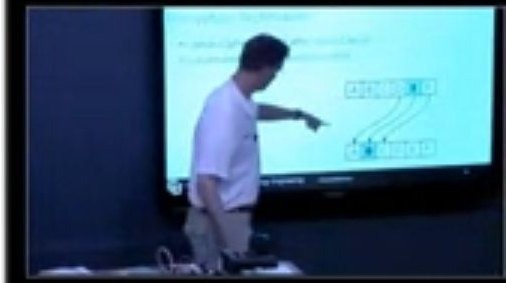
- **Plaintext**, often denoted by P (sometimes called clear-text), is what you want to protect/encrypt
- **Ciphertext**, often denoted by C, is what you get after encryption

is what you want to protect or what you want to encrypt.

Encryption Techniques

- Caesar Cipher, named after Julius Caesar
- Substitution cipher, based on a shift





A maps to X this is offset by three,

Encryption Techniques

- One time pad (OTP)
- So-called “perfect” encryption
- Impractical for real-world

A=0, B=1, C=2, … Z=25											
Plain text	M	E	E	T	T	O	N	I	G	H	T
	12	4	4	19	19	14	13	8	6	7	19
Key	D	Z	H	S	U	I	M	W	E	K	C
	3	25	7	18	20	8	12	22	4	10	2
Sum	15	29	11	37	39	22	25	30	10	17	21
Sum mod 26	15	3	11	11	13	22	25	4	10	17	21
Cipher Text	P	D	L	L	N	W	Z	E	K	R	V
Cipher text	P	D	L	L	N	W	Z	C	K	R	V
	15	3	11	11	13	22	25	4	10	17	21
Key	D	Z	H	S	U	I	M	W	E	K	C
	3	25	7	18	20	8	12	22	4	10	2
Diff	12	-22	4	-7	-7	14	13	-18	6	7	19
Sum mod 26	12	4	4	19	19	14	13	8	6	7	19
Plain text	M	E	E	T	T	O	N	I	G	H	T

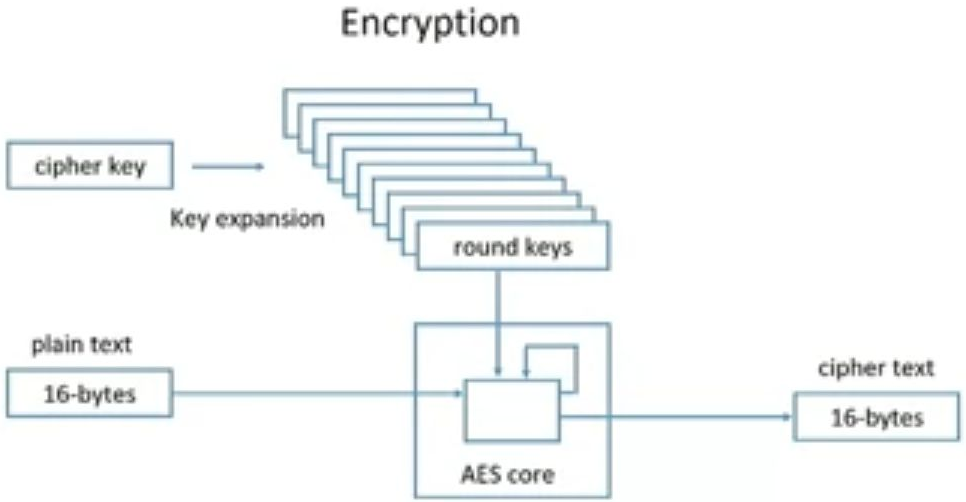
Assuming no one understands what you're using for the key.

AES (Advanced Encryption Standard)

- Established by US NIST (National Institute of Standards)
- Block cipher: 16-bytes in, 16-byte out
- 3 key lengths: 128-, 192-, 256-bits
- High-level description
  - 1) With N=number of rounds, round keys are extracted from the cipher key (where N = 10, 12, 14, for 128-, 192- or 256-bits)
  - 2) Round 0
  - 3) Rounds 1 to N-2
  - 4) Final round N-1
- Believed to be secure, but we don't know how to prove it

we just believe it to be secure.

AES Encryption



The first thing that happens in the AES algorithm is this key,

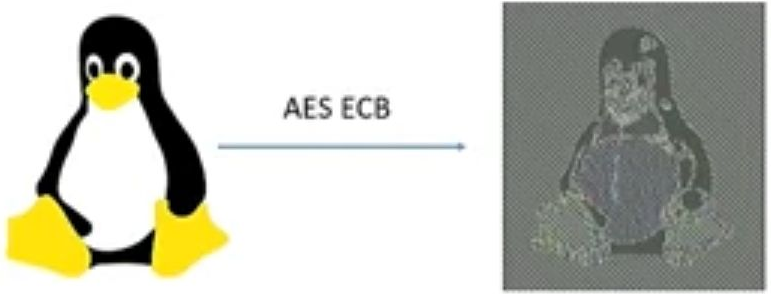
AES

- When AES is implemented as per specification, it is known as Electronic Code Book (ECB)
- Shouldn't this be good enough?
- Can you think of any issues that might arise?

Can you think of any issues that might arise by

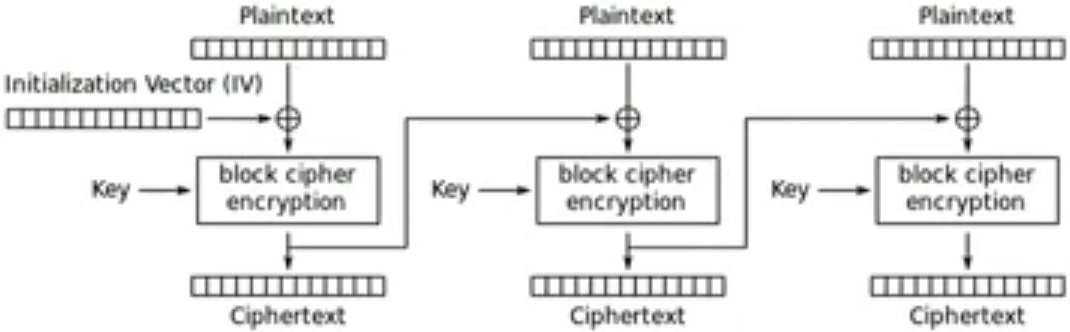
AES ECB mode

- The same plain-text always encrypts to the same cipher-text
- The result is, it leaks information



but we can kind of see what it is,

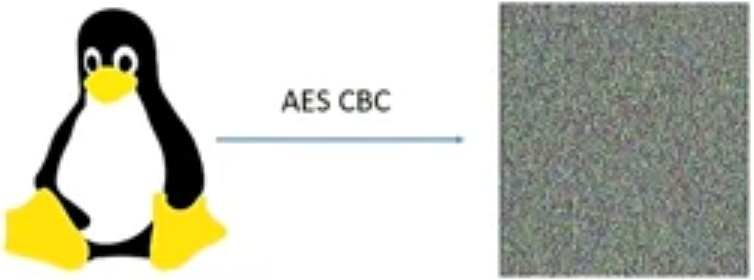
AES CBC (Cipher Block Chaining Mode)



Cipher Block Chaining (CBC) mode encryption

So what happens here,

AES CBC mode



It does not leak information.

AES XTS mode

- NIST added XTS mode for storage devices
- Specified in SP800-38E
- Deemed "better than" CBC
- AES is known as a **symmetric** encryption algorithm because the same key is used for encryption and decryption

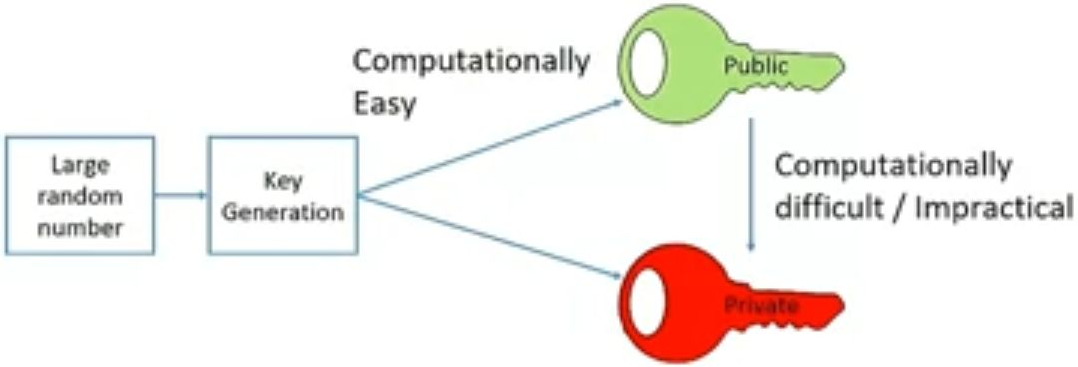
encryption algorithm because it uses

Asymmetric Encryption

- Also known as Public Key Encryption (Public Key Cryptography)
- Uses a pair of keys: 1 public, 1 private
- Provides two functions:
  - Encryption
  - Authentication, verifies the message came from the holder of the matching private key

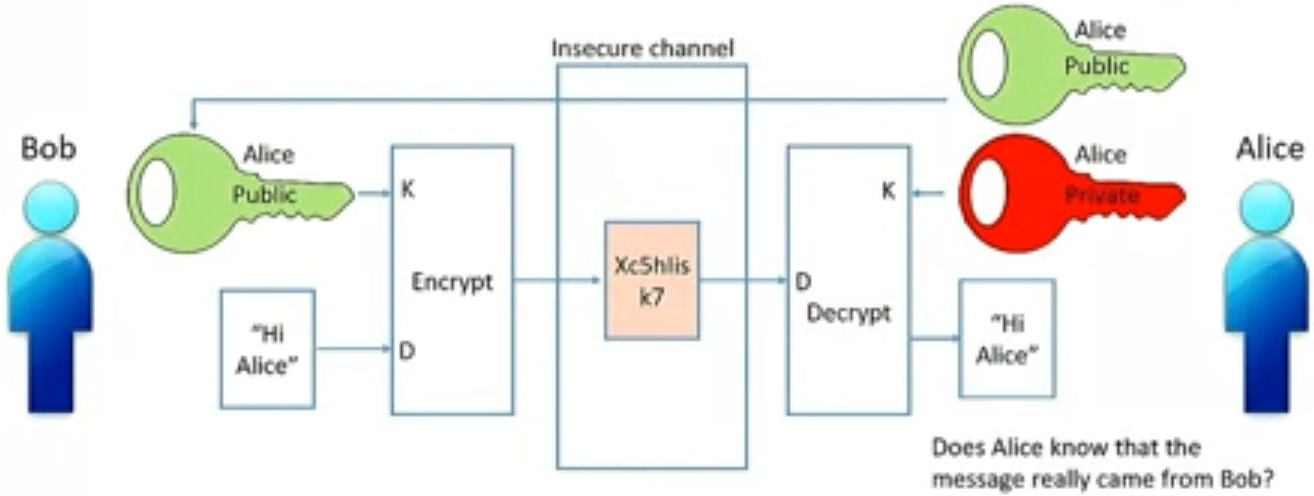
matching private key or public key depending on which one gets exchanged.

Asymmetric Encryption



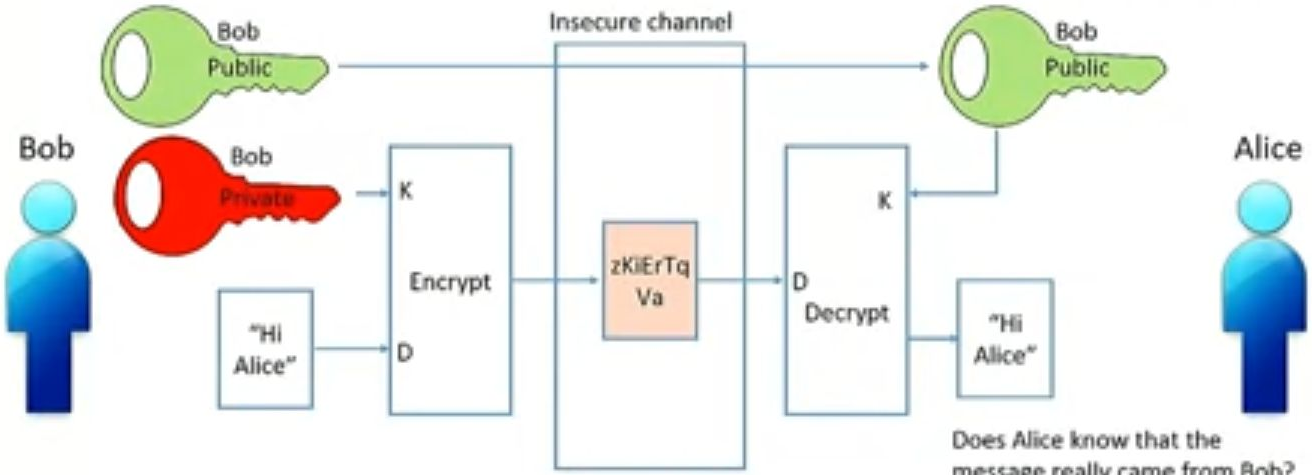
anyone can get access to it generally, and systems,

Asymmetric Encryption



So, Alice wants to receive secure communication from Bob.

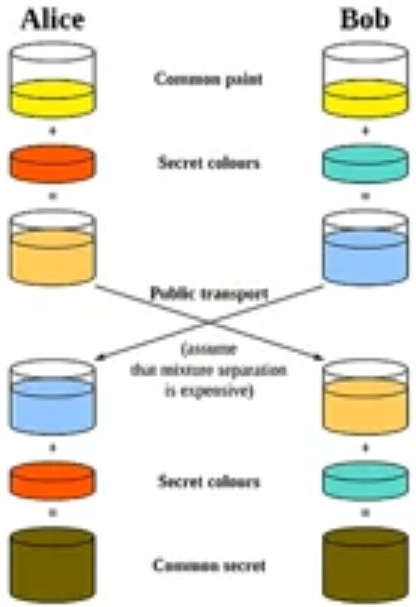
Asymmetric Encryption



So, Alice here still wants to receive secure communication from Bob,

Diffie-Hellman

A method to securely establish a known secret (a "key") between 2 parties over an insecure channel.



Source: https://en.wikipedia.org/wiki/Diffie–Hellman_key_exchange

Diffie-Hellman

The simplest and the original implementation of the protocol uses the multiplicative group of integers modulo p, where p is prime, and g is a primitive root modulo p. These two values are chosen in this way to ensure that the resulting shared secret can take on any value from 1 to p–1. Here is an example of the protocol, with non-secret values in blue, and secret values in red.

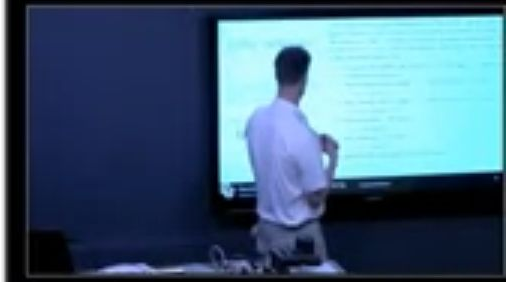
1. Alice and Bob agree to use a modulus $p = 23$ and base $g = 5$ (which is a primitive root modulo 23).
2. Alice chooses a secret integer $a = 6$, then sends Bob $A = g^a \bmod p$
   - $A = 5^6 \bmod 23 = 8$
3. Bob chooses a secret integer $b = 15$, then sends Alice $B = g^b \bmod p$
   - $B = 5^{15} \bmod 23 = 19$
4. Alice computes $s = B^a \bmod p$
   - $s = 19^6 \bmod 23 = 2$
5. Bob computes $s = A^b \bmod p$
   - $s = 8^{15} \bmod 23 = 2$
6. Alice and Bob now share a secret (the number 2).

Note: Larger values of a, b and p would be needed to make this secure, g is usually a small number.

Also Note:

$$(g^a)^b = (g^b)^a$$

Source: https://en.wikipedia.org/wiki/Diffie–Hellman_key_exchange
Primitive root mod n: http://math.stackexchange.com/questions/795414/what-are-primitive-roots-modulo-n



If you want to understand what a primitive root modulo n is,

Diffie-Hellman

Alice		Bob		Eve	
Known	Unknown	Known	Unknown	Known	Unknown
$p = 23$		$p = 23$		$p = 23$	
$g = 5$		$g = 5$		$g = 5$	
$a = 6$	b	$b = 15$	a		a, b
$A = 5^a \bmod 23$		$B = 5^b \bmod 23$			
$A = 5^6 \bmod 23 = 8$		$B = 5^{15} \bmod 23 = 19$			
$B = 19$		$A = 8$		$A = 8, B = 19$	
$s = B^a \bmod 23$		$s = A^b \bmod 23$			
$s = 19^6 \bmod 23 = 2$		$s = 8^{15} \bmod 23 = 2$			s

Source: https://en.wikipedia.org/wiki/Diffie–Hellman_key_exchange

So all Alice doesn't know is the secret number that Bob picked was b.

Diffie-Hellman

- Alice and Bob now use symmetric encryption (AES XTS for example) on an insecure channel, using s as the common encryption/decryption key.

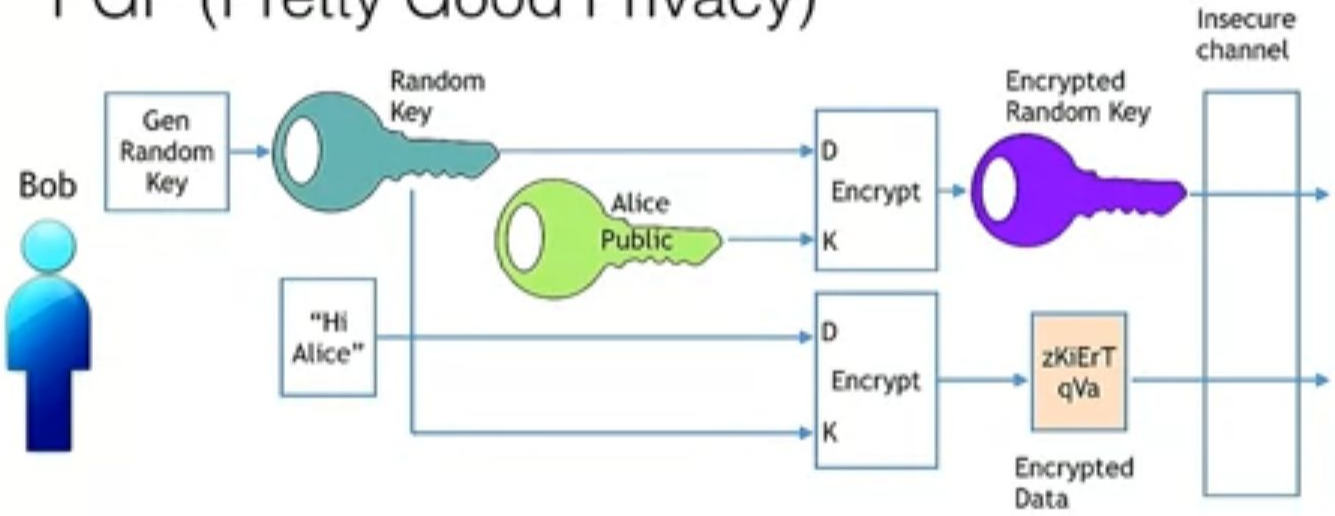
an insecure channels using S as
a common encryption and decryption key.

Diffie-Hellman

- Alice and Bob now use symmetric encryption (AES XTS for example) on an insecure channel, using s as the common encryption/decryption key.

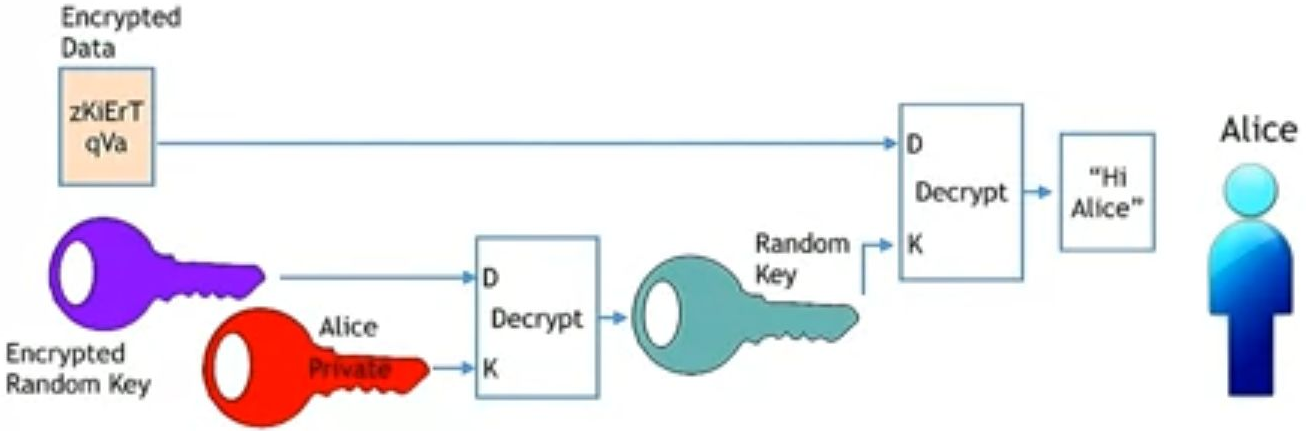
knowledgeable about it to see if there are any threat vectors or

PGP (Pretty Good Privacy)



how it works.

PGP (Pretty Good Privacy)

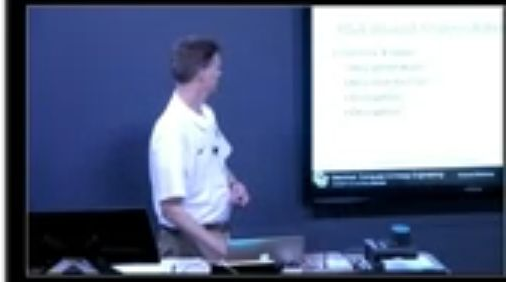


So Alice uses her private key, which is pair of the public key she gave and

RSA (Rivest-Shamir-Adleman)

- Involves 4 steps
  - Key generation
  - Key distribution
  - Encryption
  - Decryption

Source: https://en.wikipedia.org/wiki/RSA_(cryptosystem)



Key generation, key distribution, encryption, and decryption.

RSA

- It is practical to find 3 very large positive integers e, d and n such that with modular exponentiation for all integers m, we have:

$$(m^e)^d \equiv m(\mathrm{mod}\, n)$$

and

$$(m^d)^e \equiv m(\mathrm{mod}\, n)$$

$b^y \% m$

Integer b raised to y, divided by the modulus m to produce a remainder

And that even knowing e, m and n, it can be very difficult to determine d

So the tenant of RSA is that it is practical to find three very large

RSA Key Generation

- Choose two distinct prime number p and q
  - Chosen at random, similar magnitudes
- Compute $n = p*q$
- n is used as the modulus, this is the key length
- Compute $\lambda(n) = lcm(p-1, q-1)$ kept private
- Choose an integer e, such that $1 < e < \lambda(n)$
  - where e and $\lambda(n)$ are coprime
- Determine d as $d \equiv e^{-1} (\mathrm{mod}\, \lambda(n))$

So key generation process starts off
by choosing two distinct prime numbers,

RSA Encryption

- Bob wants to send Alice a message M after receiving n and e from Alice
- Bob reduces his message to an integer m where 0 <= m <= n by using a previously agreed upon reversible protocol known as a *padding scheme*. Cipher text is then computed as:

$$c \equiv m^e \pmod{n}$$

For padding explanation, see: https://en.wikipedia.org/wiki/Optimal_asymmetric_encryption_padding

that's what you would send across the link.

Hash Functions

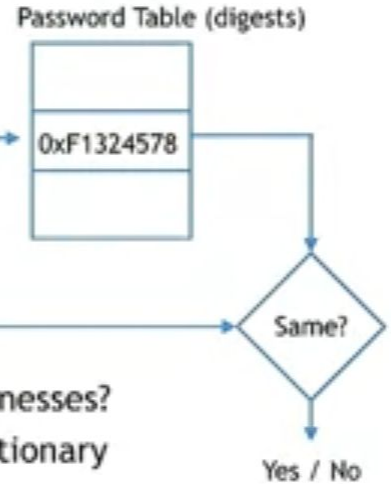
- Maps a message of arbitrary length to a fixed number of bits, referred to as the hash value (or message digest, or simply **digest**)
- "Good" cryptographic hash functions have 5 properties:
  - Deterministic: same input, same output every time
  - Quick to compute, computationally inexpensive
  - ***Impractical to determine a message from the hash value***
    - ***Has the notion of being a "one-way" function***
  - A small change in input results in a large change in the output
  - Impractical to find two messages that map to the same hash value
- Used to check the integrity of data transmission

See: https://en.wikipedia.org/wiki/Comparison_of_cryptographic_hash_functions for a list of cryptographic hash functions

Hash functions can be used to check the integrity of data transmission.

Uses for Hash Functions

- Saving passwords
  - Login: UserName
  - Password: Xh8_i27vZ



- See also the SHA-2 family of hash functions as per FIPS 180-4
- Well studied, haven't spotted a problem yet

- Are there weaknesses?
- What about dictionary attacks?

Like Snowden, I'm not exactly sure how he got his data but he was able

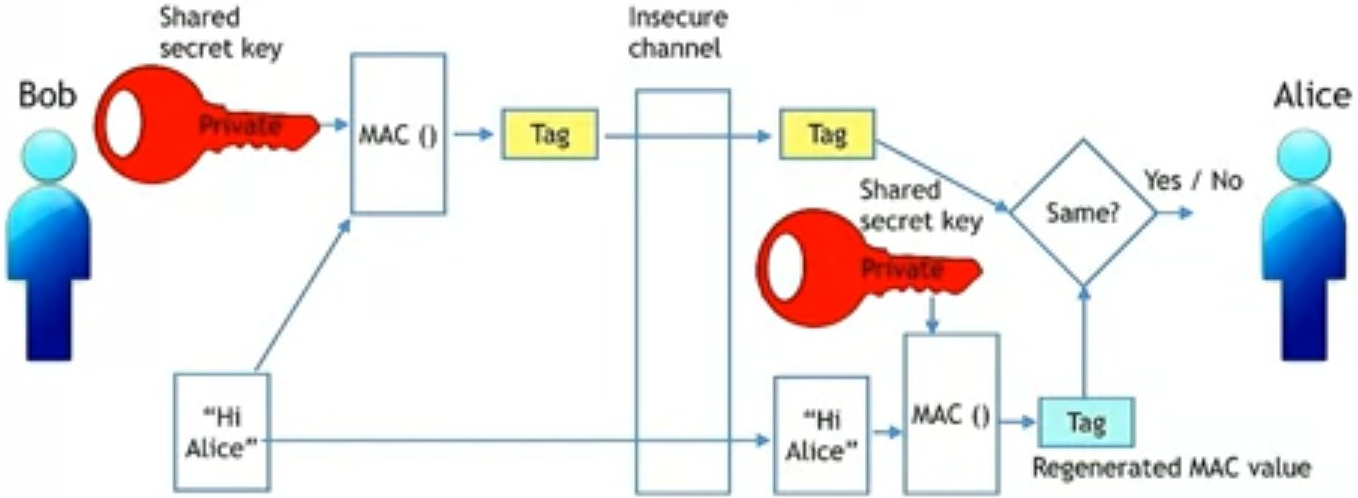
Message Authentication Codes (MACs)

- Accepts as inputs:
  - A secret key
  - A message of arbitrary length
- Outputs a fixed-size MAC value (also know as a **tag**)
- A cryptographic hash function is one method to generate a MAC value.
- See HMAC-SHA256, RFC 2104
- Often used for authentication

This one has these, including that one,

MAC Usage

Here we are concerned about **authenticity** - did this message really com from Bob?
And **integrity** - has the message been modified in transit?



So, here we are concerned about the authenticity.

Open source software downloads

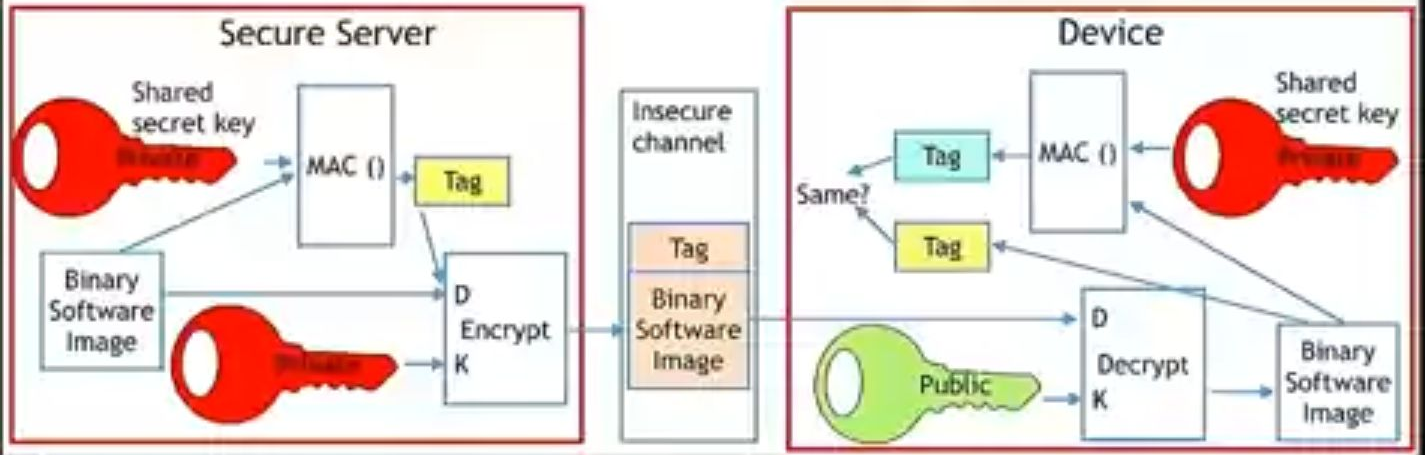
- How many of you have seen/used an MD5 checksum?
- Vulnerabilities have been discovered that make it unsuitable for use as a cryptographic hash function, but can still be used for data integrity.
  - For example: Has anyone messed with the .zip file I downloaded?

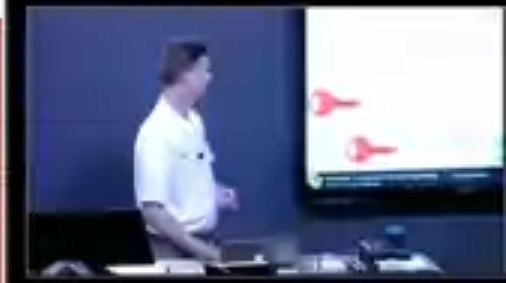
Has anyone messed with my zip file?

Software/Firmware Updates

- Should always be authenticated!
- Build this into the system from day one
- Asymmetric encryption would be a fair design choice

- Authentication
- Integrity
- Security

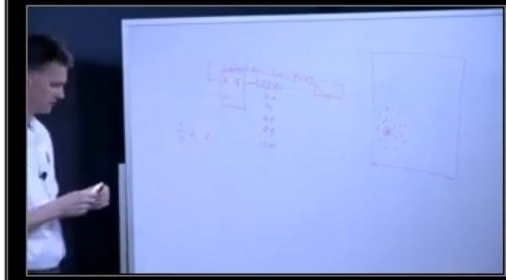




It handles authentication, integrity, and security altogether.

Key Generation

- In order to be effective, the keys used for security should be random numbers so an adversary cannot guess them.
- My advice is to steer away from pseudo-random number generation because these have a predictable pattern.
- Very hard to generate “true” random numbers.



So, that was better than we had anticipated.

Key Generation

- In order to be effective, the keys used for security should be random numbers so an adversary cannot guess them.
- My advice is to steer away from pseudo-random number generation because these have a predictable pattern.
- Very hard to generate “true” random numbers.
- Use the output or “measures” of physical systems that are inherently entropic, such that they are indistinguishable from “true” random numbers. Choices may include:
  - Temperature
  - Vibration
  - Flow rate
  - Ring oscillator
- Then mix these sources into your key generation

talked about and mix all these sources together.