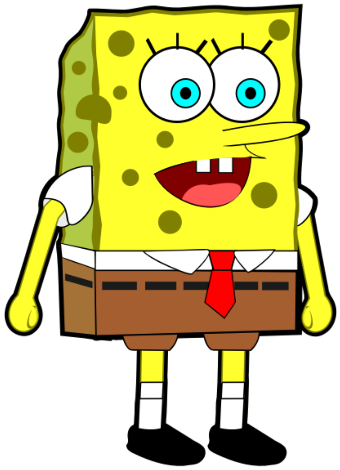
# Public-Key Cryptography Lecture 12a: 2022-04-04

MAT A02 – Winter 2022 – UTSC

Prof. Yun William Yu

# A Communications Story


(A)lice from Alice's Adventures in Wonderland
Illustration by Arthur Rackham, 1907



Sponge(B)ob Squarepants
https://freesvg.org/sponge-bob-squarepant

(E)ve by Lucas Cranach the Elder (1528)

# Symmetric ciphers

- Caesar shift: add a number to every letter mod 26.
  - The number you add is the “key”.
  - If you know the key, you can both encrypt and decrypt.

- Vigenère cipher: add a cyclically repeating word to the message mod 26.
  - The word you add is the “key”.
  - If you know the key, you can both encrypt and decrypt.

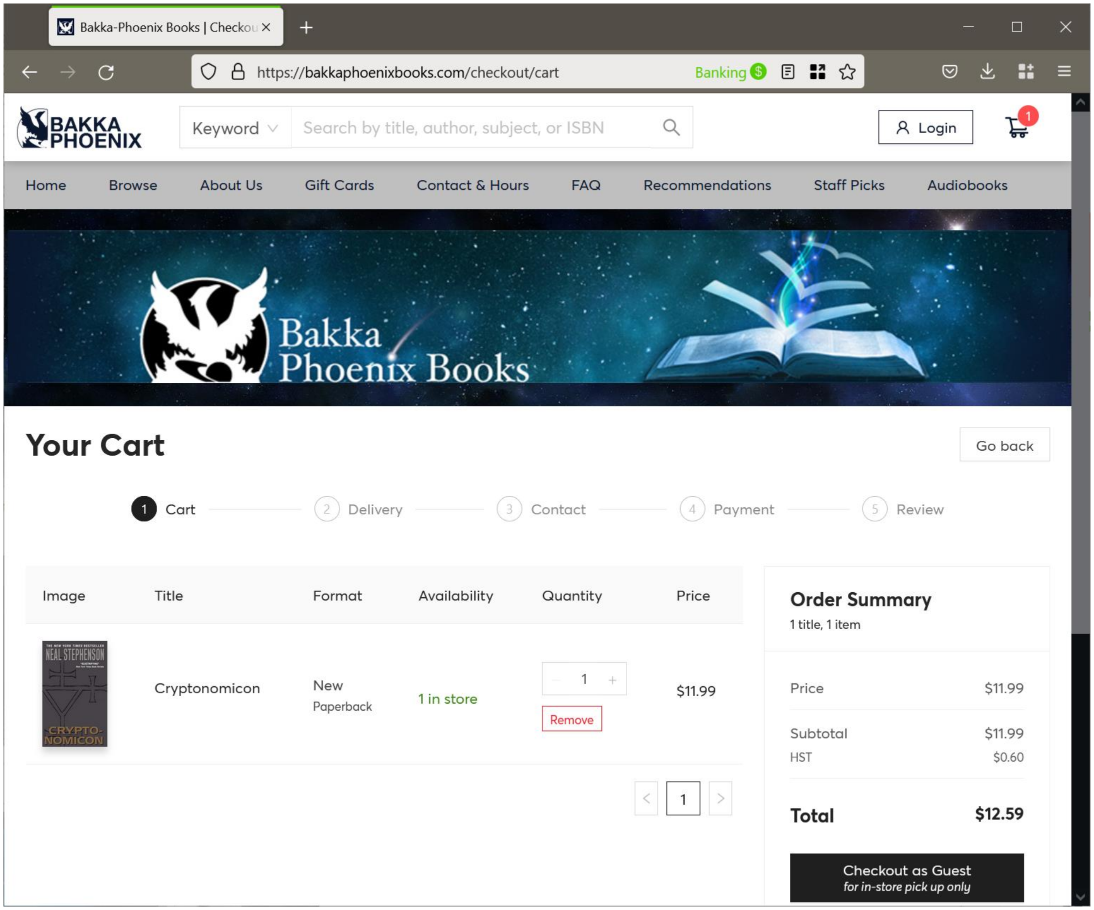
# Symmetric ciphers and eavesdropping

- Symmetric ciphers use the same key for encryption and decryption.
- They only work when Eve only hears part of the conversation.
- If Eve ever hears the key, then she can decrypt the entire conversation, both past and present.

Johann Georg Meyer von Bremen
(Germany, 1813 - 1886)

# Beating a perfect eavesdropper

- What if Eve knows everything Alice and Bob have ever said to each other, so there's no way for them to share a secret key without Eve knowing?
- In fact, let's say that Alice and Bob have never even met, but are just communicating on the Internet.



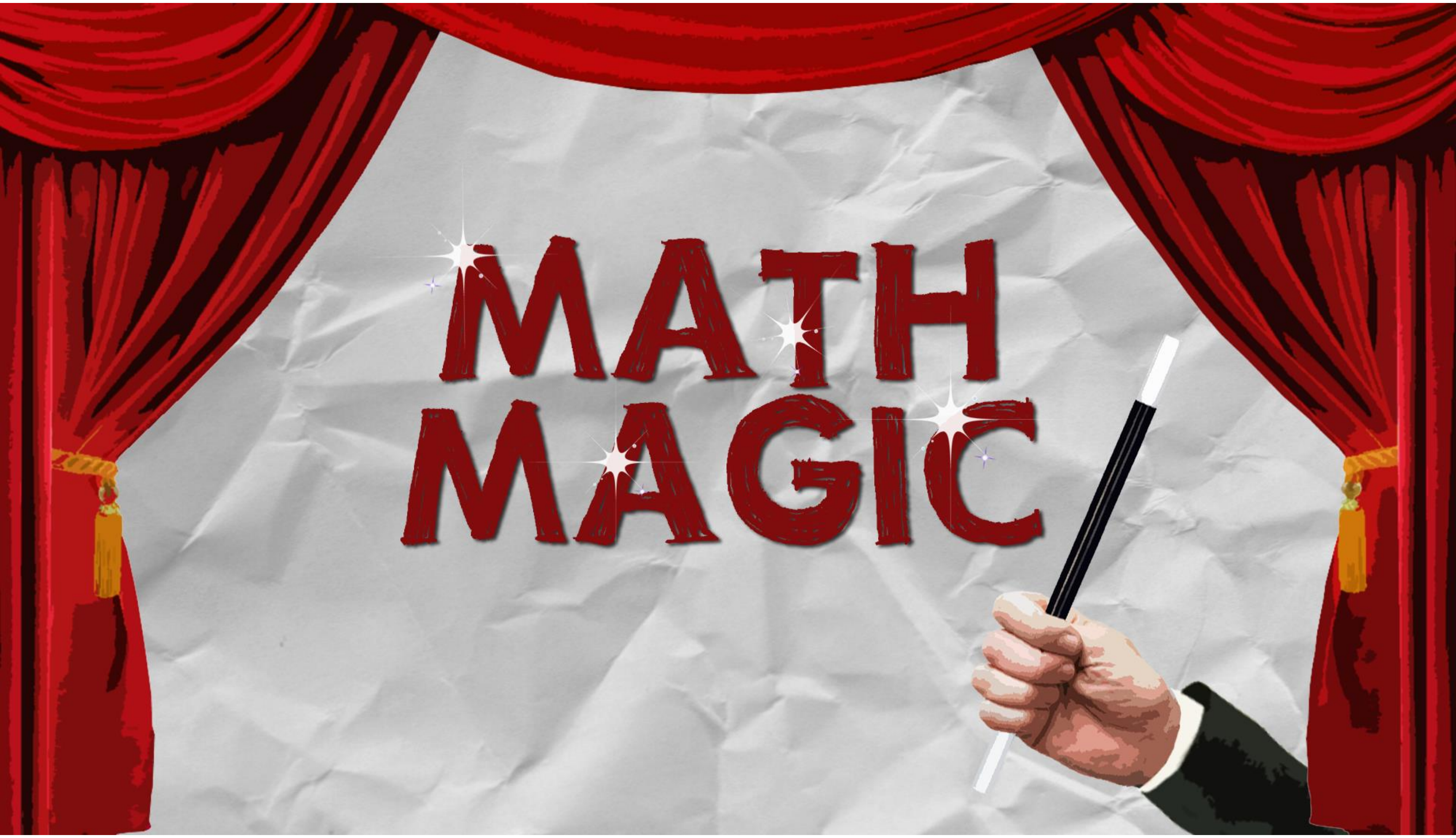
# Asymmetric encryption

- What if encryption and decryption use different keys? Or if encryption doesn’t need a secret at all?

Symmetric encryption

Asymmetric encryption

- Then, Bob could give everyone the encryption key, but not tell anyone the decryption key.




https://www.pbslearningmedia.org/resource/42439c47-de30-487c-9c10-563367a2c843/math-magic/

# Reversing is hard: factoring

- We define addition, multiplication, exponentiation, etc, and those are easy.

https://www.flickr.com/photos/nenadstojkovic/50446472706/in/photostream/

- Subtraction, division, and roots, are reversing those operations and sometimes much harder.

Floris de Wit; https://dribbble.com/shots/5039546-Moonwalk

# Factoring large numbers

- Figuring out if a number is prime is “easy” using probabilistic primality testing (e.g. Fermat)
  - We want to know if $n$ is prime.
  1. Pick a random number $a$ between 1 and $n-1$
  2. Compute $a^{n-1} \pmod{n}$
  3. If $a^{n-1} \not\equiv 1 \pmod{n}$, then $n$ is not prime and $a$ is a witness to this fact. Otherwise, $n$ passes the test, and you don’t know for certain, but you can repeat.

- Factoring a composite number is “hard” for large numbers if you don’t know any divisors.

# Finding roots with Euler's Theorem

- Algorithm for $\sqrt[k]{a} \pmod{n}$ using Euler's Theorem
  - Conditions: $\gcd(a, n) = 1$ and $\gcd\big(k, \phi(n)\big) = 1$.
  - Find 1 as a combination of $k$ and $\phi(n)$
    $$1 = km - l\phi(n)$$
  - Then $a^1 \equiv a^{1+l\phi(n)} \equiv a^{km}$.
  - So $\sqrt[k]{a} \equiv \sqrt[k]{a^{km}} \equiv a^m \pmod{n}$

# Reversing is hard: roots

- Computing $a^k \pmod{n}$ is needs $a, k, n$.

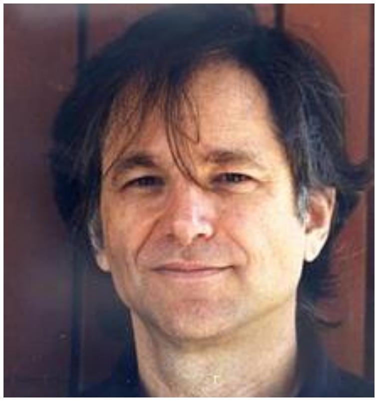
- Computing $\sqrt[k]{a} \pmod{n}$ is needs $a, k, n, \phi(n)$.

# RSA (Rivest-Shamir-Adleman, 1977)

- First public-key cryptosystem, which allows two parties who have never communicated before to send messages securely to each other.
- Made internet shopping and banking possible, because you can communicate securely with other computers without worrying about eavesdroppers.

Ron Rivest

Adi Shamir



Leonard Adleman

# RSA algorithm

1. Alice introduces herself to Bob.
2. Bob generates a two large random primes, $p, q$ and computes the product $n = pq$
3. Bob chooses an exponent $k$ with $\gcd\big(k, \phi(n)\big) = 1$.
4. Bob sends $(n, k)$ to Alice as a public key. Anyone who knows the public key can send messages to Bob that only he can decrypt.
5. Alice has a message $a$, with $\gcd(a, n) = 1$, so she sends $b \equiv a^k \pmod{n}$ to Bob
6. Bob can decrypt $a \equiv \sqrt[k]{b} \pmod{n}$


# Example

1. Alice introduces herself to Bob.
2. Bob generates a two large random primes, $p, q$ and computes the product $n = pq$
3. Bob chooses an exponent $k$ with $\gcd\big(k, \phi(n)\big) = 1$.
4. Bob sends $(n, k)$ to Alice as a public key. Anyone who knows the public key can send messages to Bob that only he can decrypt.
5. Alice has a message $a$, with $\gcd(a, n) = 1$, so she sends $b \equiv a^k \pmod{n}$ to Bob
6. Bob can decrypt $a \equiv \sqrt[k]{b} \pmod{n}$

# Example

- Alice wants to send the message “196” to Bob, but don’t want eavesdroppers knowing.
- Bob generates two prime numbers, 19 and 23, which he keeps as his secret key.
- He also chooses 61 as his exponent. Then he publishes (437, 61) as his public key.
- Alice sends $196^{61} \pmod{437} \equiv 9$ to Bob.
- Bob can then decrypt the message because he knows the factorization of $437 = 19 \cdot 23$, so he can compute $\phi(437) = 19 \cdot 23 \cdot \frac{18}{19} \cdot \frac{22}{23} = 18 \cdot 22 = 396$
- Eve cannot decrypt the message unless she is able to factor 437.

# Decryption in detail

- Eve intercepts an encrypted message “9” sent to a public key (437, 61).
- Being really clever, Eve breaks the private key and figure out that $437 = 19 \times 23$.
- What was the original unencrypted message?

# Try it out: encryption

- Encrypt the message 9 using the RSA public key $(n, k) = (77, 13)$, without factoring 77.

A: 12
B: 23
C: 58
D: 70
E: None of the above

# Try it out: decryption

- Decrypt a message encrypted using the RSA public key $(n, k) = (77, 13)$, with secret key $77 = 7 \cdot 11$.
- The encrypted message is 26.
- Helpful hint: $1 = 60 \cdot 5 - 23 \cdot 13$

A: 5
B: 12
C: 34
D: 67
E: None of the above

# Hybrid cryptosystems

- Public-private key encryption is often a lot harder than symmetric key encryption.
- This is true even for computers, which can do both, but are much slower at public-key encryption.
- Thus, in practice, Alice and Bob only use RSA to send a very short message containing a key for symmetric encryption.