Cryptography

CSE 143 Exploration Session

Melissa Winstanley

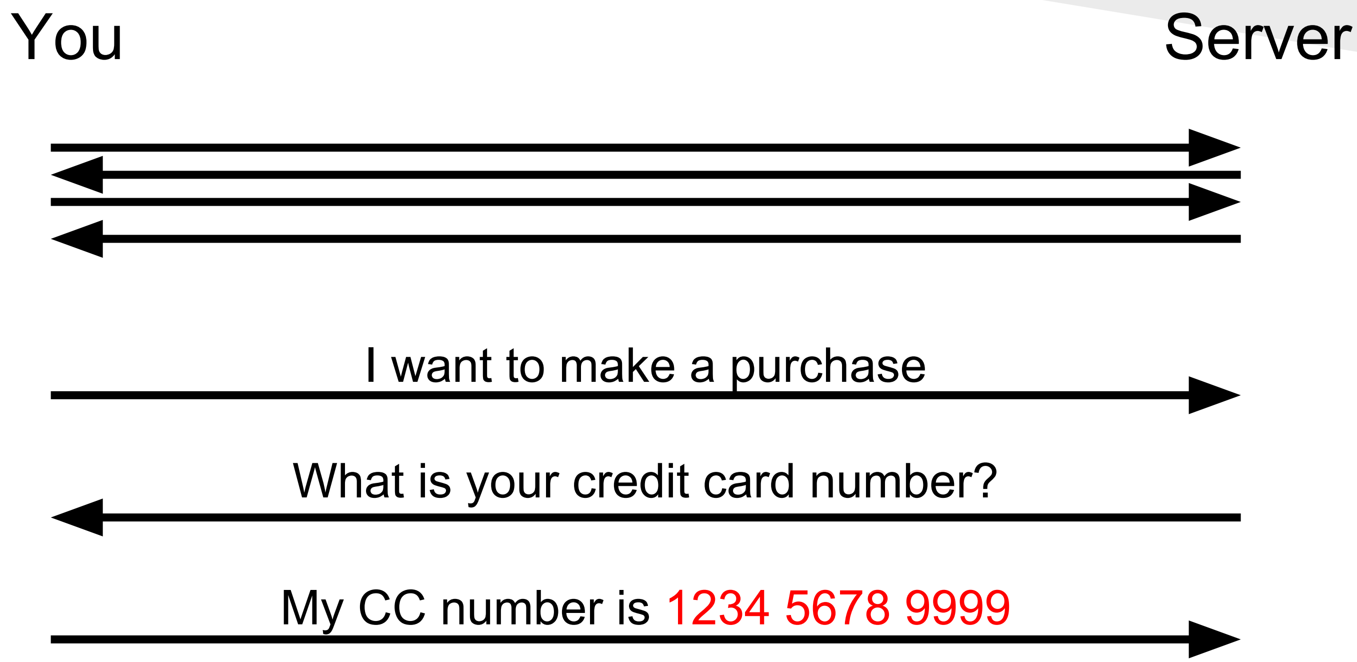
Slides based on presentation by Josh Benaloh

Internet Security

- The Internet was NOT designed for security.

- Sending data through the Internet is like sending a postcard through the mail...
  - ...when you don't trust the post office

A typical internet session



Basic encryption

- Can we AT LEAST protect the credit card number so it won't be revealed to anybody except the merchant?

Kerckhoff's Principle (1883)

- The security of a cryptosystem should depend only on the key.
- You should assume that attackers know everything about your system except the key

Some terminology

- Informally…
  - A PIN is a 4-6 digit speed bump
  - A password is a short, user-chosen, usually guessable selection from a small dictionary.
  - A key is an unguessable, randomly chosen string – usually at least 128 bits

Off-Line Attacks

- Don't even think about using user-chosen passwords as encryption keys.
- Don't even think about using keys derived deterministically from user-chosen passwords.
- Given the ciphertext, an attacker can do a (guided) exhaustive search through the space to find the password.

Symmetric cryptography

- If the client has a pre-existing relationship with the merchant, the two parties may have a shared secret key K – known only to these two.
  - User encrypts private data with key K.
  - Merchant decrypts data with key K.
- Two classes
  - Stream ciphers
  - Block ciphers

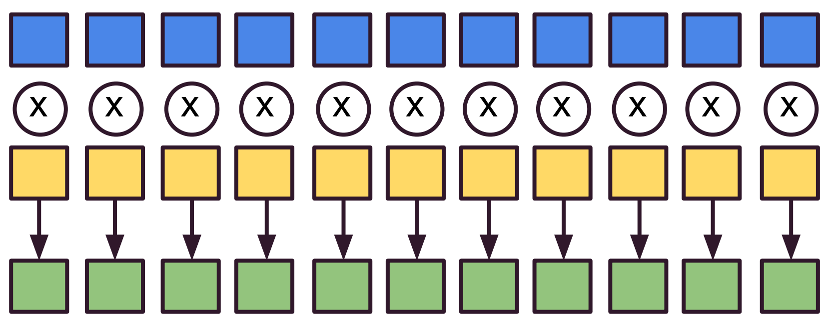
Stream Cipher

- RC4, A5/1, SEAL, etc.
- Use the key as a seed to a pseudo-random number-generator.
- Take the stream of output bits from the PRNG and XOR it with the plaintext to form the ciphertext.
    - (1x1->0, 1x0->1, 0x1->1, 0x0->0)

Plaintext:

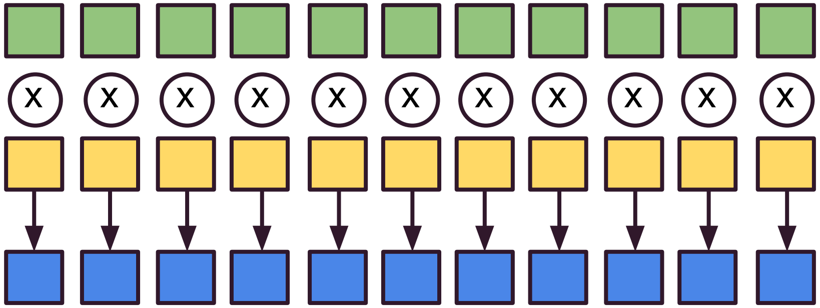
PRNG (seed):

Ciphertext:



Stream Cipher Decryption

Plaintext:
PRNG (seed):
Ciphertext:



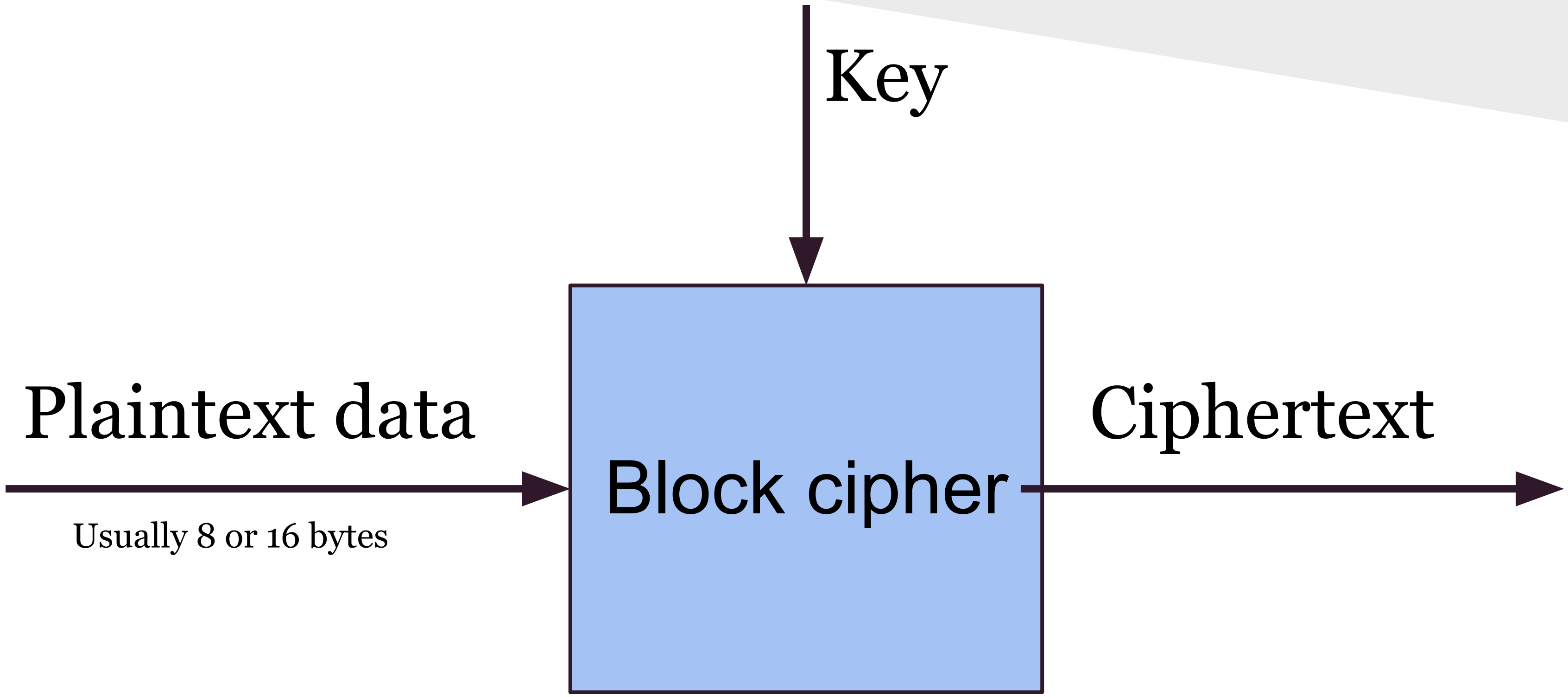
Stream cipher evaluation

- The good
  - Usually fast
  - Usually simple
  - Same function for encrypt and decrypt
- The bad
  - Hint: Something XOR'ed with itself disappears, which is why decryption works
  - If the same PRNG seed is ever reused...
  - (PT1 x PRNG) x (PT2 x PRNG) = (PT1 x PT2)

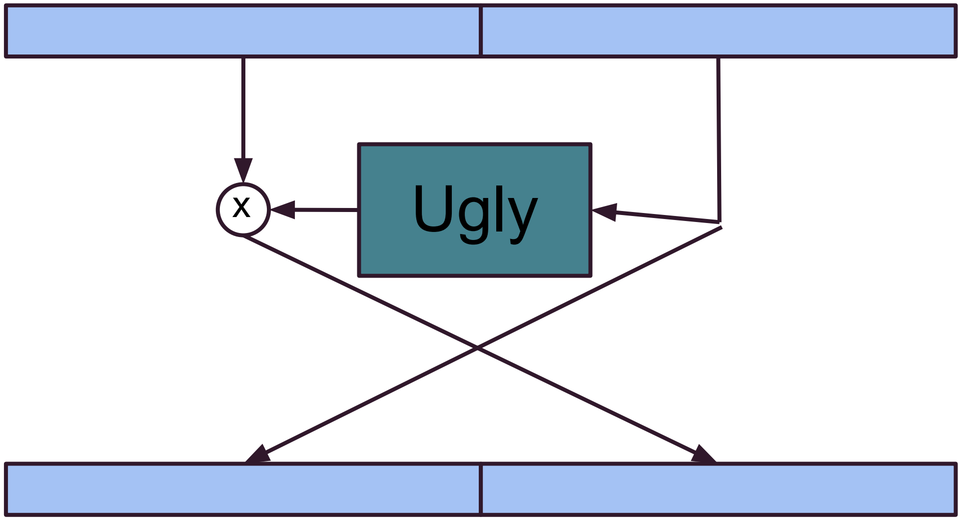
More bad

- It is easy for an adversary (even one who can't decrypt the ciphertext)to alter the plaintext in a known way.
- Eg,.Bob to Bob's Bank:
  - Please transfer $0,000,002.00 to the account of my good friend Alice.
  - Please transfer $1,000,002.00 to the account of my good friend Alice.

Block Cipher

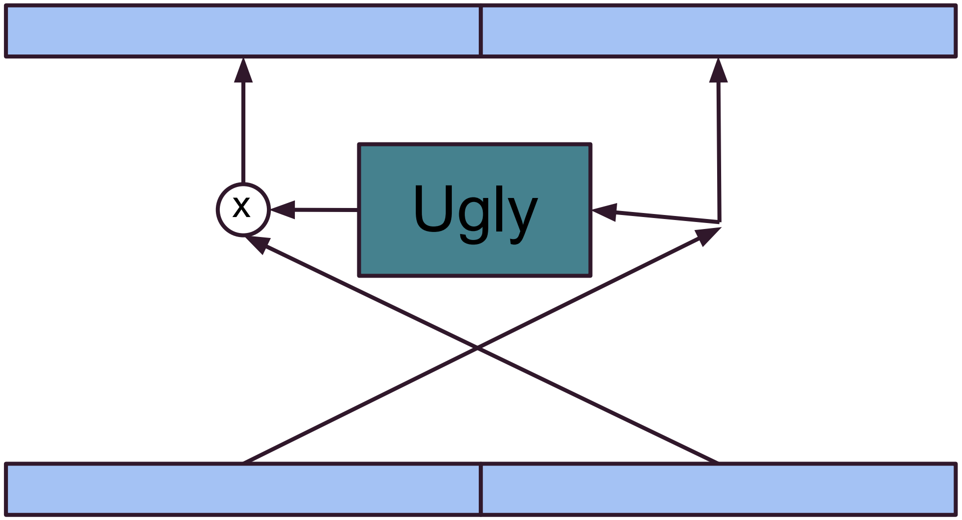


Feistel cipher



Encoding

Feistel cipher

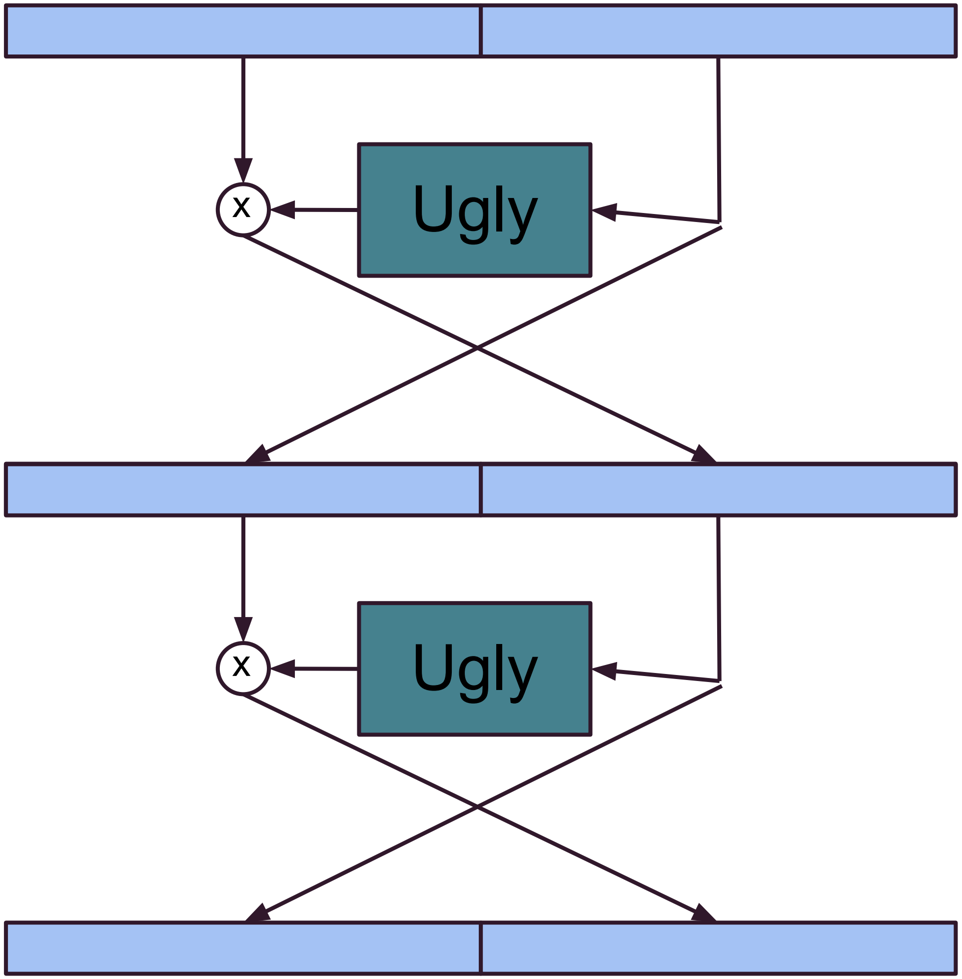


Decoding

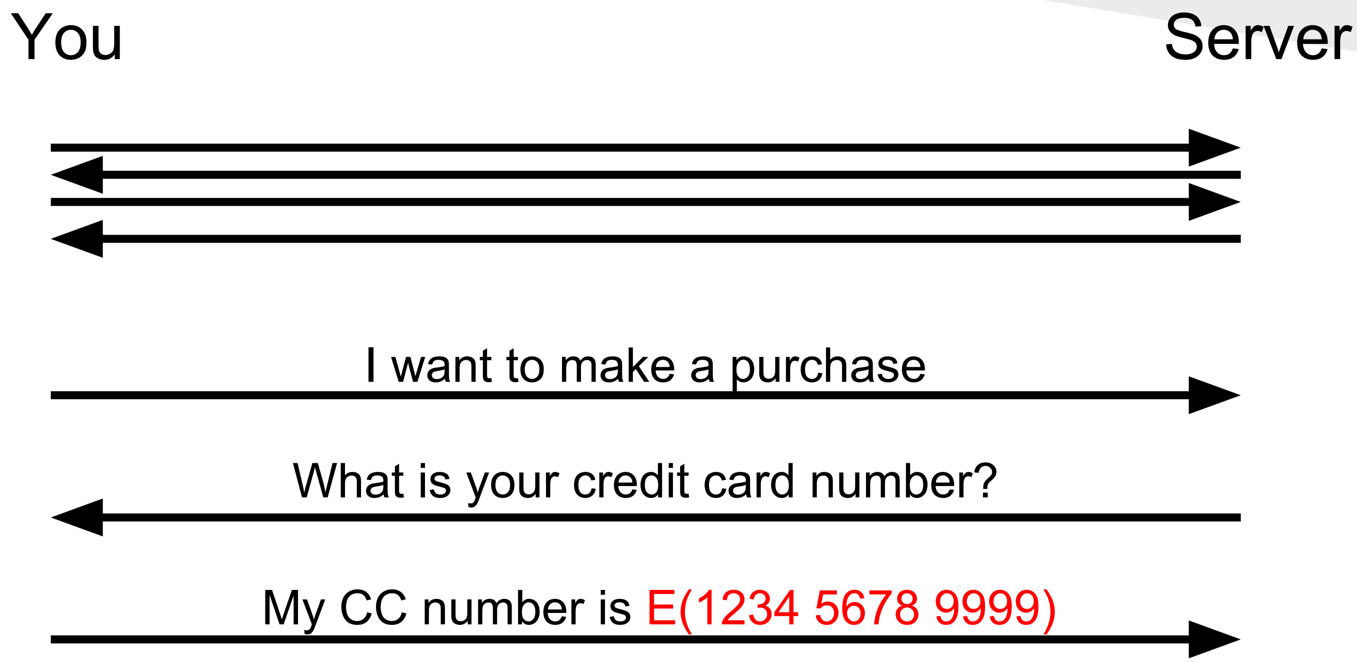
Feistel performance

- Typically, Feistel ciphers are iterated for about 10-16 rounds.
- Different "sub-keys" are used for each round.
- Even a weak round function can yield a strong Feistel cipher, if iterated sufficiently.

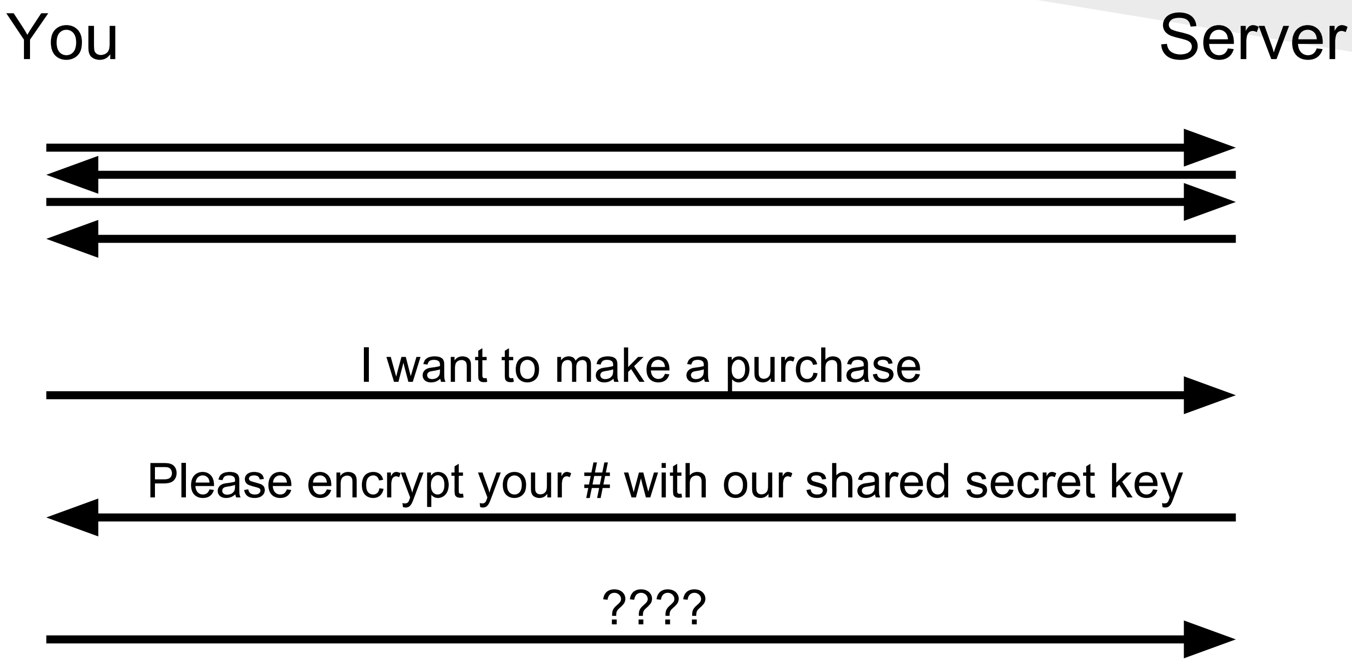
Feistel cipher



Our new encoded system



Our new encoded system



Asymmetric cryptography

- What if the user and merchant have no prior relationship?
- Asymmetric encryption allows someone to encrypt a message for a recipient without knowledge of the recipient’s decryption key.
- Usually involves lots of math.

The Fundamental Equation

$$Z = Y^X \bmod N$$

The Fundamental Equation

$$Z = Y^X \bmod N$$

If Z is unknown, it can be computed efficiently.

The Fundamental Equation

$$Z = Y^{X} \bmod N$$

If X is unknown, the problem is called the *discrete logarithm* and is generally hard to solve

The Fundamental Equation

$$Z = Y^x \bmod N$$

If Y is unknown, the problem is called the *discrete root finding* and is generally hard to solve, without factorization of N.

The Fundamental Equation

$$Z = Y^X \bmod N$$

If N is unknown, the problem is not well studied.

RSA encryption

- Pick two primes p and q, compute n = pq
- Pick two numbers e and d, such that:
  - ed = (p-1)(q-1)k + 1 (for some k)
- Publish n and e (public key), encode with:
  - $(\text{original message})^e \bmod n$
- Keep d, p and q secret (private key), decode with:
  - $(\text{encoded message})^d \bmod n$

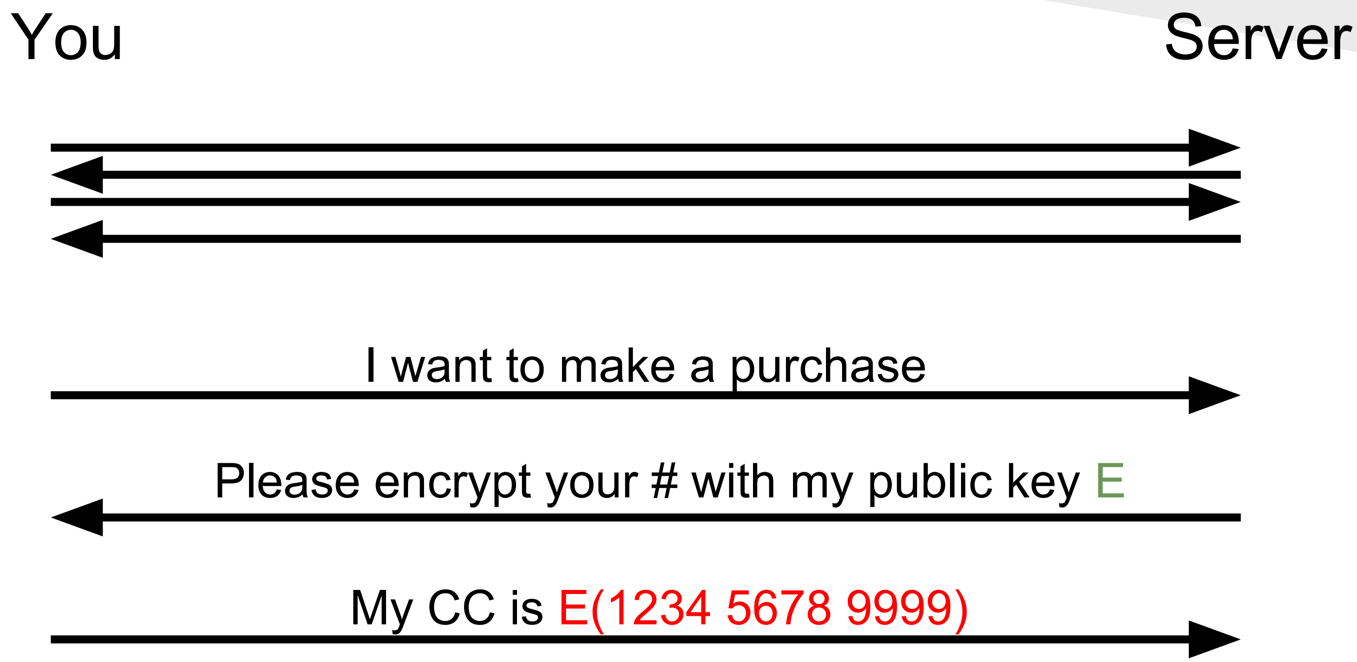
Why does it work?

- Original message is carried to the e power, then to the d power:
  - $(msg^{e})^{d} = msg^{ed}$
- Remember how we picked e and d:
  - $msg^{ed} = msg^{(p-1)(q-1)k + 1}$
- Apply some simple algebra:
  - $msg^{ed} = (msg^{(p-1)(q-1)})^{k} \text{ x } msg^{1}$
- Applying Fermat's Little Theorem:
  - $msg^{ed} = (1)^{k}\ msg^{1} = msg$

A brief history of RSA

- British discovered RSA first but kept it secret
- Phil Zimmerman tried to bring cryptography to the masses w/PGP
  - Investigated as an arms dealer by FBI and a grand jury
- Shor's algorithm would break RSA if only we had a quantum computer
- The NSA hires more mathematicians than any other organization

Our RSA based system



Problems

- Man-in-the-middle attack
  - Someone pretends to be the server
  - Solution: Certificates
  - Need *certificate authorities*
  - Must guarantee the certificate authorities
- Replay attack
  - Someone repeats your encoded message
  - Solution: a unique *nonce* (number)