

EECS 388 Final Review!!

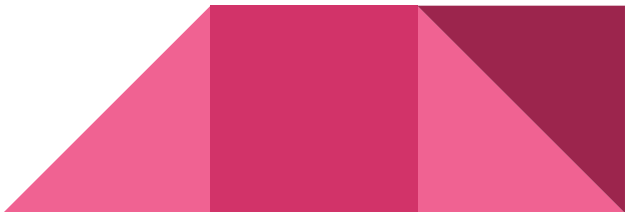
December 12th, 2018

Logistics

- Final Exam

- Thursday, December 14th, 7-9pm
- Room assignments in pinned Piazza post, “Final Exam Rooms”
- The exam DOES NOT start on Michigan Time. **We begin promptly at 7pm.**
- No cheat sheet
 - Pro-tip: **make one anyway to study!**
- If you need special accommodations, you should have gotten an email

- Extra office hours?

- Wednesday, 12-3pm, DOW 1005 (Dean and Becca)
 - Wednesday, 3-5:30pm, Location TBD (Gabby)
- 

Crypto! - Basic Building Blocks (functions)

PRF:

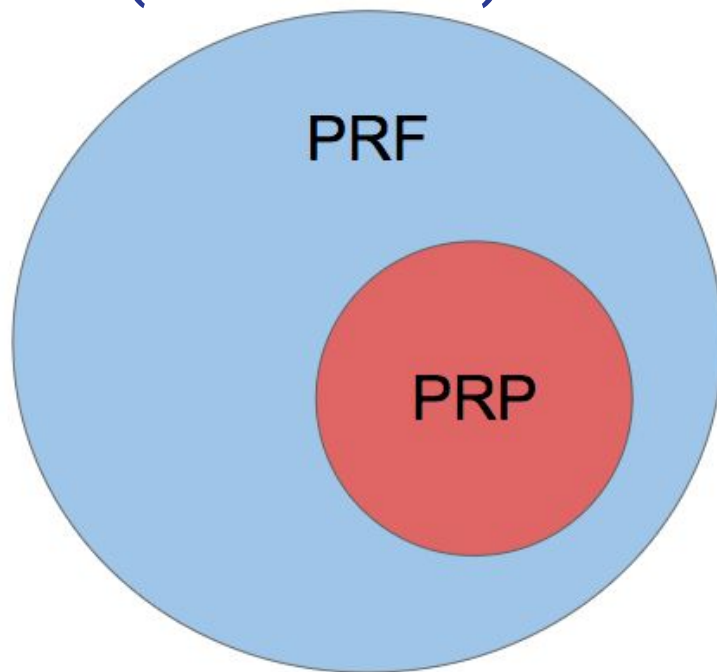
$F_k(x) = y$ //mapping is pseudorandom

PRP:

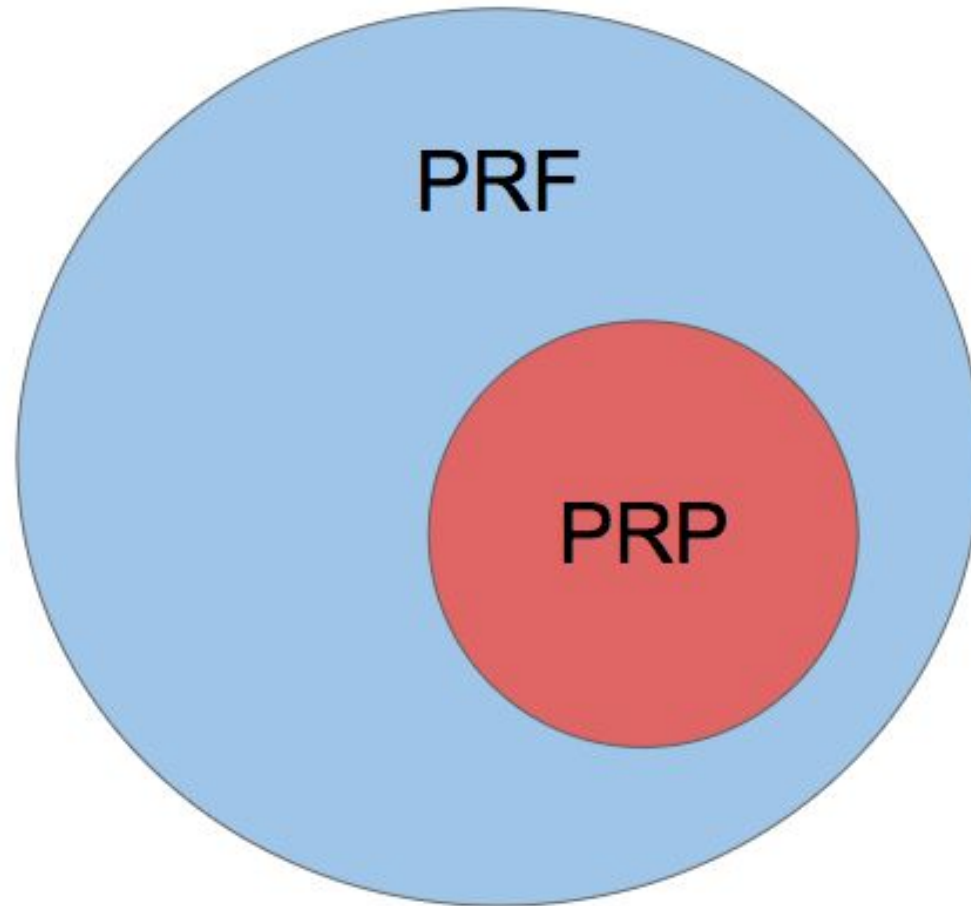
Given : $F_k \rightarrow \exists F_k^{-1}$ such that

$$F_k(x) = y$$

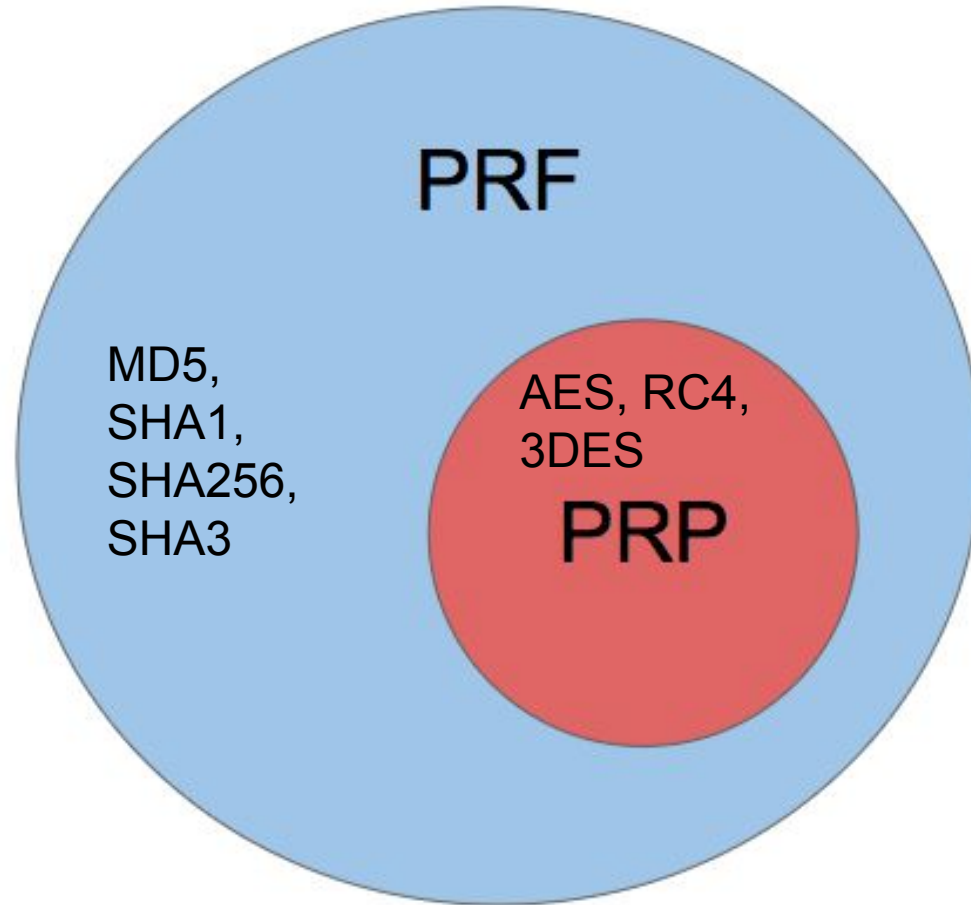
$$F_k^{-1}(y) = x$$



Crypto!



Crypto!

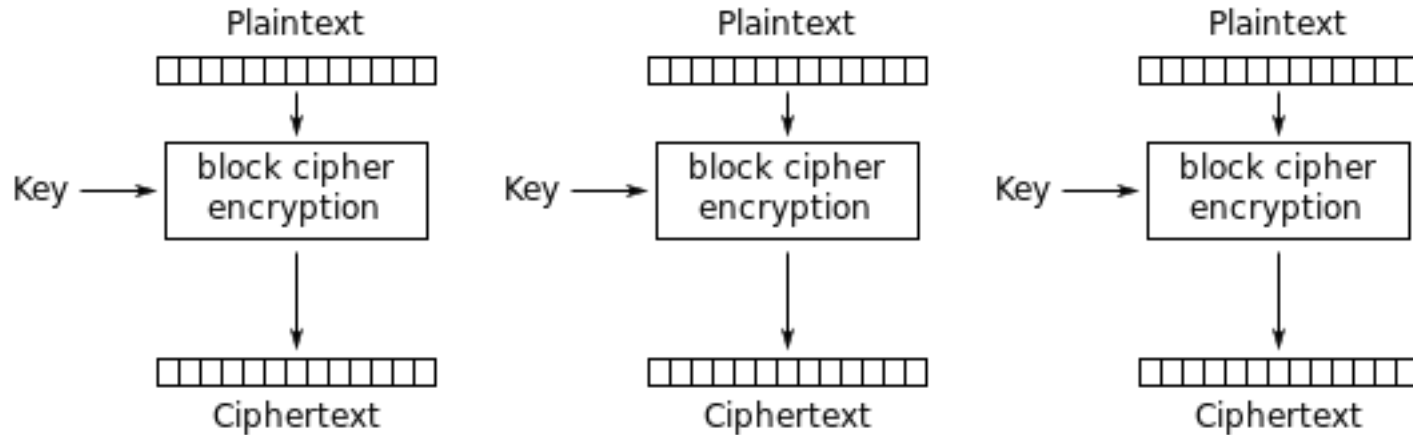


Crypto! - Block Cipher Modes!

- Block cipher = break message into blocks
- ECB
- CBC
- CTR



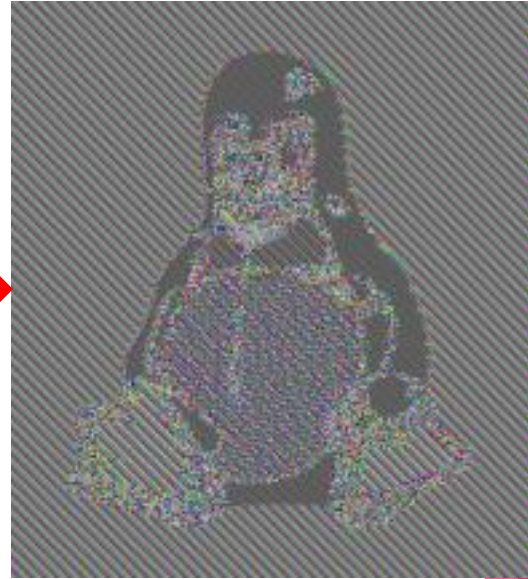
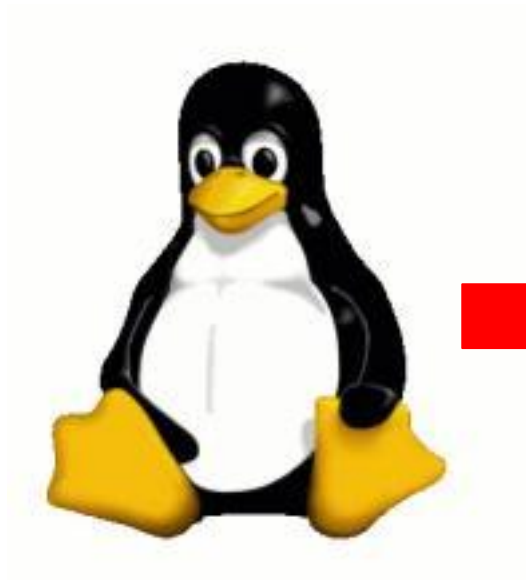
Crypto! - ECB (Electronic Codebook)



Electronic Codebook (ECB) mode encryption

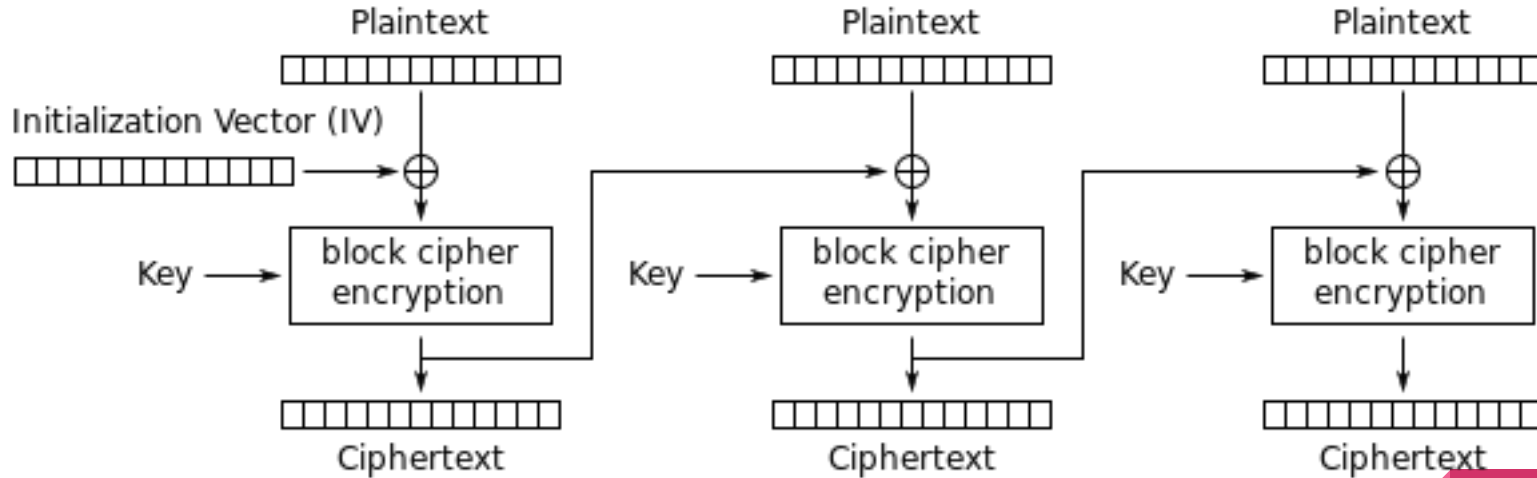
Problem?

Crypto! - ECB (Electronic Codebook)



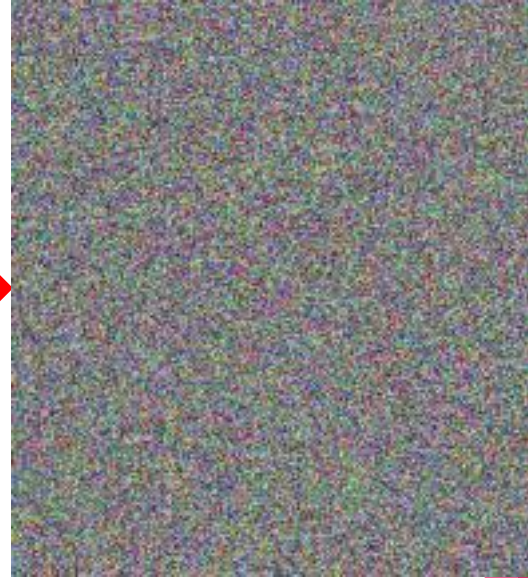
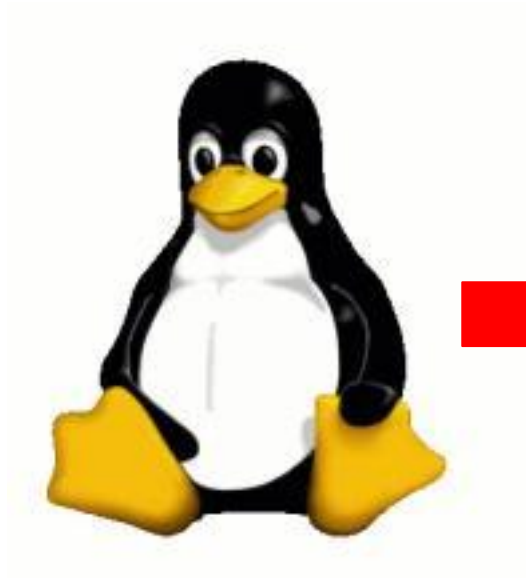
Crypto! - CBC (Cipher Block Chaining)

To fix, let's make each block rely on the previous:



Cipher Block Chaining (CBC) mode encryption

Crypto! - CBC



Problem still?


Crypto! - CBC

Padding Oracle!

If a padding oracle exists, then you have no security!

A padding oracle is:

An agent whose behavior let's slip whether or not a given ciphertext has valid or invalid padding



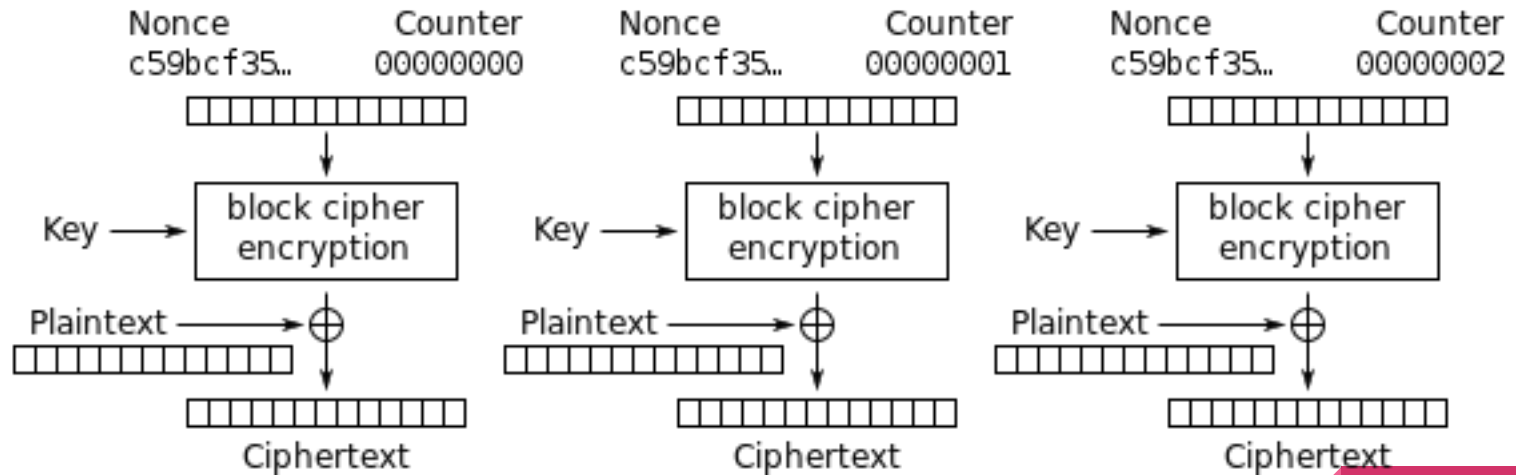
Crypto! - CBC

Example on board!



Crypto! - CTR (Counter)

Effectively a stream cipher! - no ecb or cbc probs



Counter (CTR) mode encryption

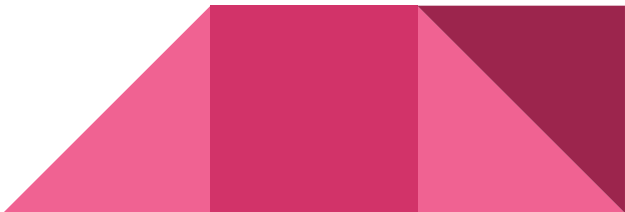
Crypto! Encrypt MAC

Why should we Encrypt then Mac?

Mac then Encrypt: $MAC_k(m) = tag$
 $Enc_k(tag) = C$

Encrypt and Mac: $Enc_k(m) = C$
 $MAC_k(m) = tag$

Encrypt then Mac: $Enc_k(m) = C$
 $MAC_k(C) = tag$



Crypto! Diffie Hellman (Key Exchange)

DH Protocol

Standard g (generator), and p (prime, or modulus)

Alice picks
secret a



$$g^a \bmod p$$

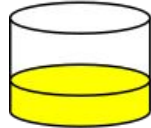
$$g^b \bmod p$$

Bob picks
secret b



$$g^{ab} \bmod p = g^{ba} \bmod p$$

Alice



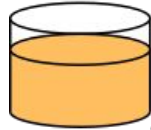
Common paint

+

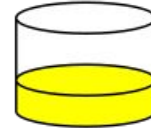


Secret colours

=



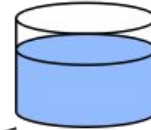
Bob



+

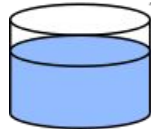


=



Public transport

(assume
that mixture separation
is expensive)



+



Secret colours

=



+



=



Common secret

Crypto! RSA (Rivest-Shamir-Adleman)

How RSA Works

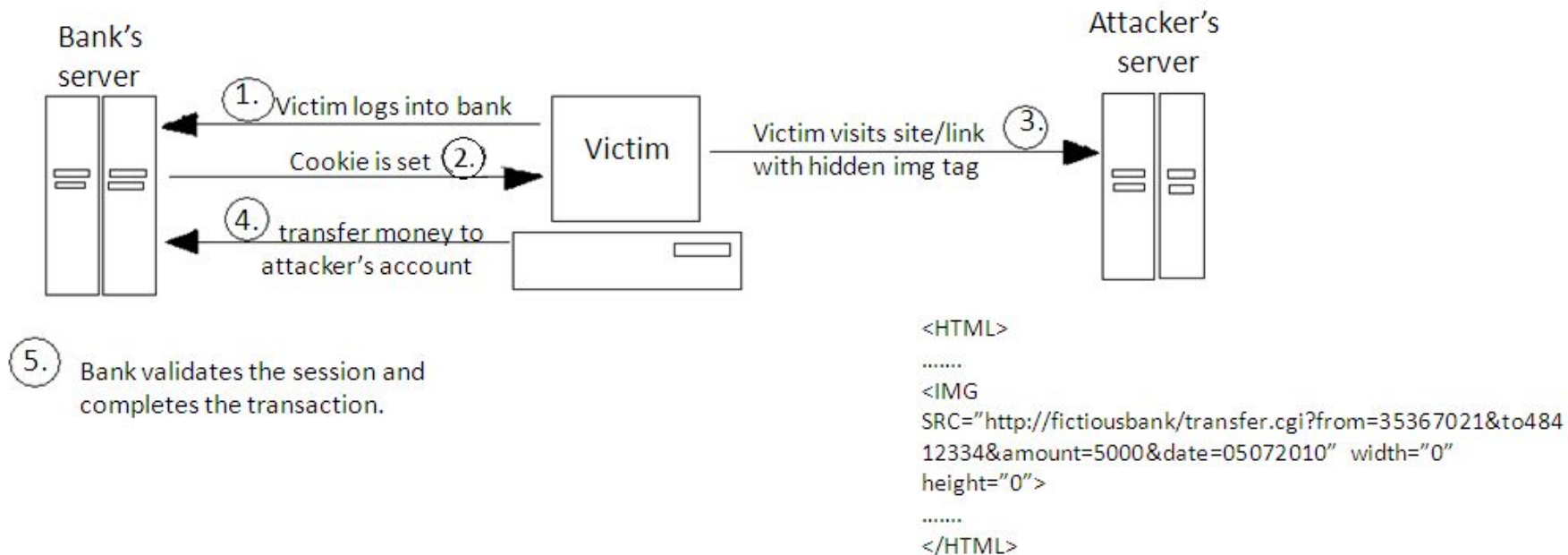
1. Pick two large (say, 2048 bits) random primes **p** and **q**.
2. **N** := **p** * **q** (RSA does multiplication mod **N**)
3. Pick **e** to be relatively prime to **(p - 1) * (q - 1)**.
4. Find **d** so that **(e * d) mod ((p - 1) * (q - 1)) = 1**
5. Public key := (**e**, **N**) ← the key is actually a pair of values
Private key := (**d**, **N**)
6. **Encryption:** **Enc_e(x) = x^e mod N**
Decryption: **Dec_d(x) = x^d mod N**

CSRF

- CSRF exploits the *browser*
 - Trick the user into submitting a malicious request, usually with *social engineering*
 - Browser should enforce the same origin policy
- Same origin policy
 - Scripts contained in one page are allowed to access a resource belonging to a second page if and only if both web pages have the same origin
- What gives something the same origin?
 - Same protocol, host, and port



CSRF



XSS

- XSS exploits the *site not properly distinguishing data from code*
 - Inject a payload into the site's HTML
 - Search bar, forum comment, etc.

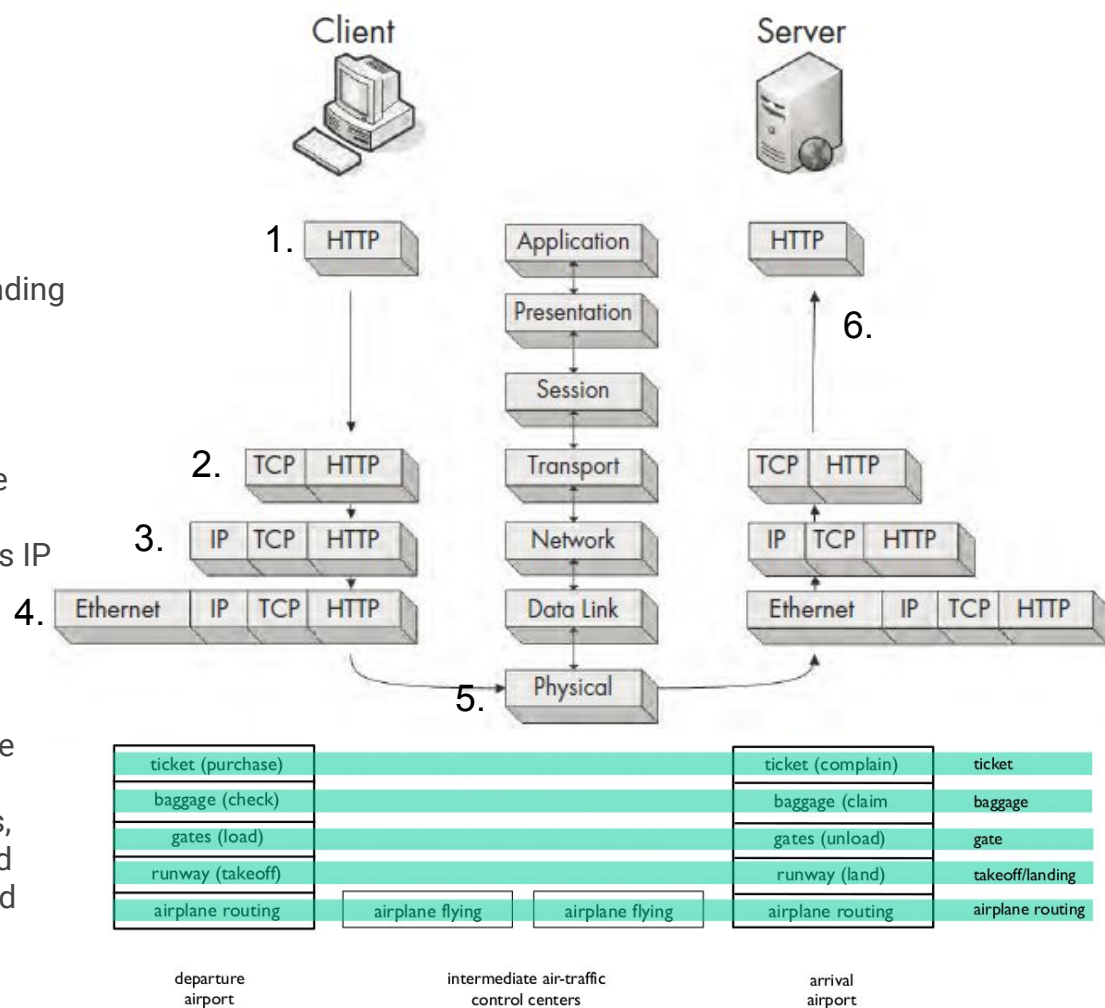


Networking Theory

Know basic packet encapsulation (this isn't 489; high-level is fine). Each layer talks to its corresponding layer on the other host.

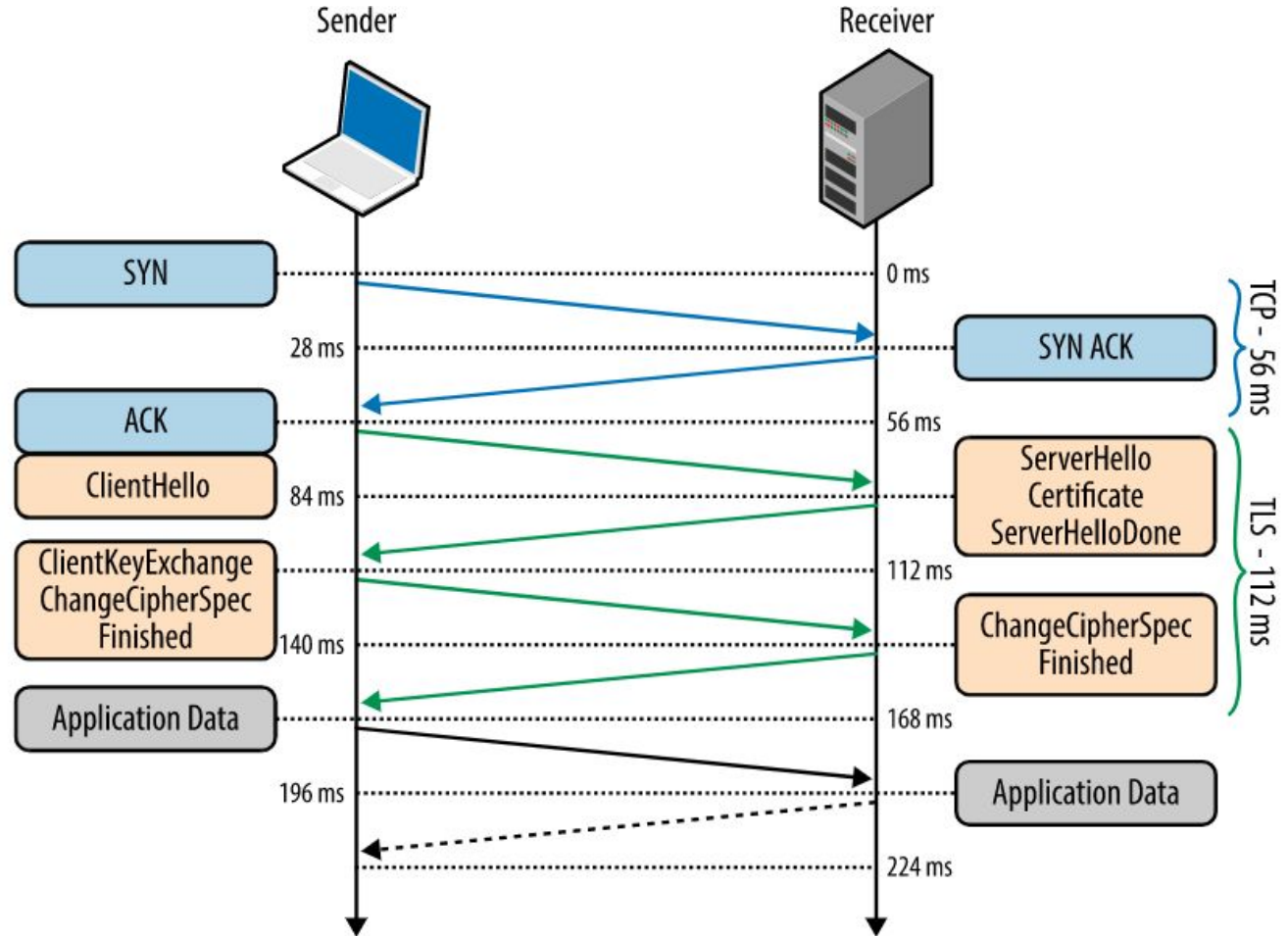
Ex. Sending a GET request to a Web Server

1. Client's browser makes the HTTP request
2. The kernel wraps it in TCP (it's more reliable than UDP), destination port 80
3. The kernel wraps that in IP (the destination's IP address)
4. The kernel wraps that in Ethernet (MAC Address) and sends it to the router.
5. Physical routers between the source and destination check the IP address and get the packet to its destination network
6. The destination network peels off the layers, eventually getting the packet to the intended Web Server, so it can serve the webpage and send it back the same way.



Networking!

TCP & TLS Handshakes



Applied Network Security

Getting on the network

- Aircrack
- Social engineering

Mapping the network

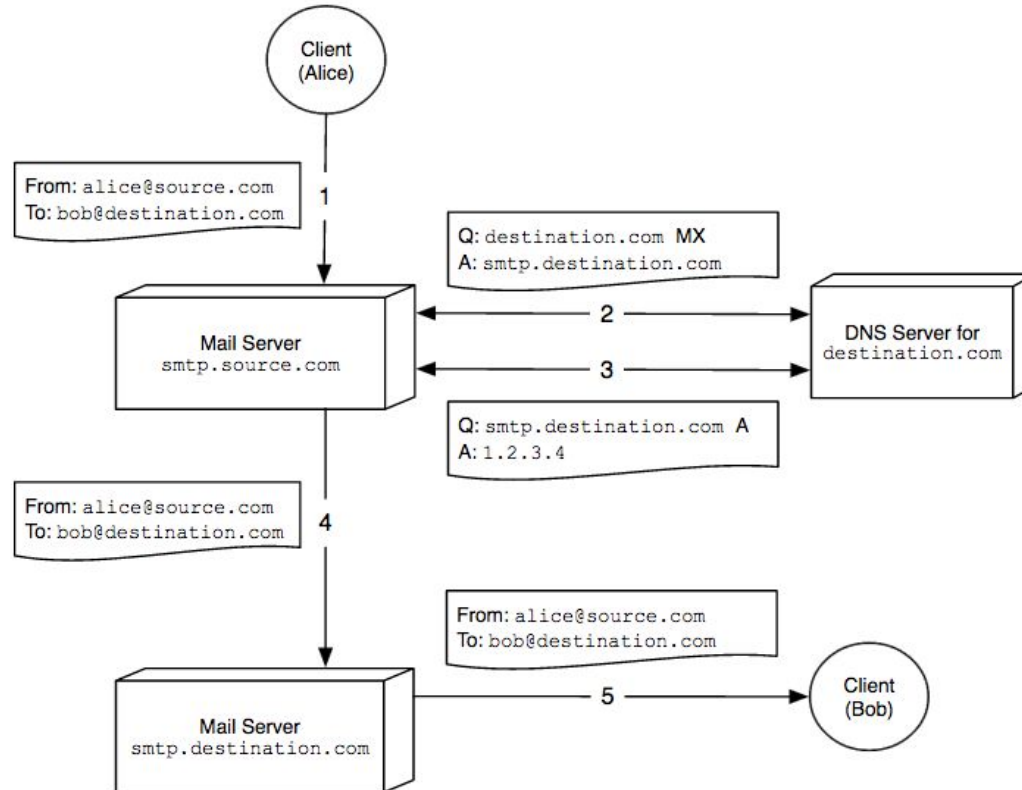
- Nmap - seeing what devices are responding on a network, and which ports they have open (what services they're running)
- Know the common ports and what they mean.
 - 80, 443, 22, 25

Listening on a network

- Wireshark, tcpdump



E-Mail in the real world

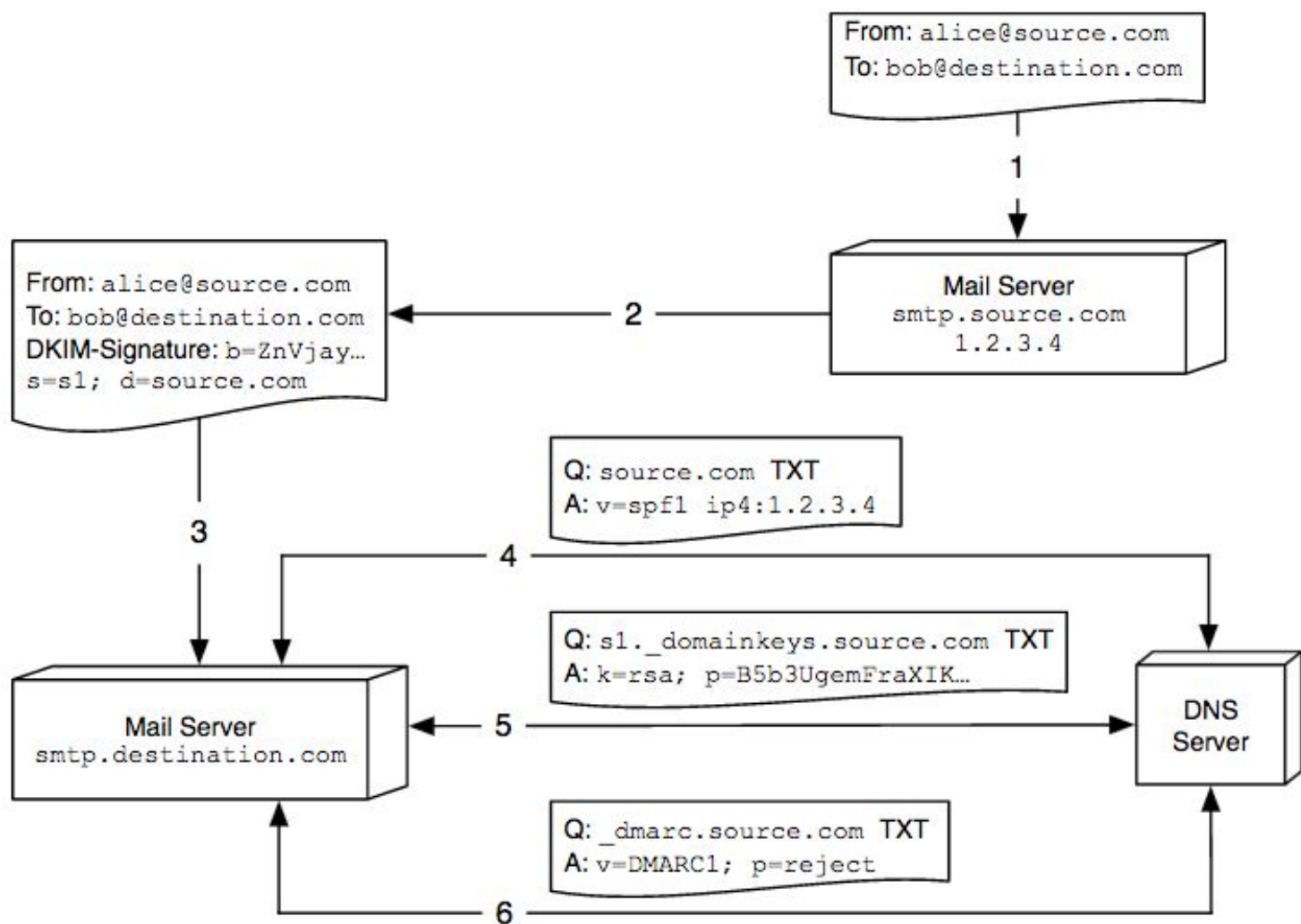


E-mail Security

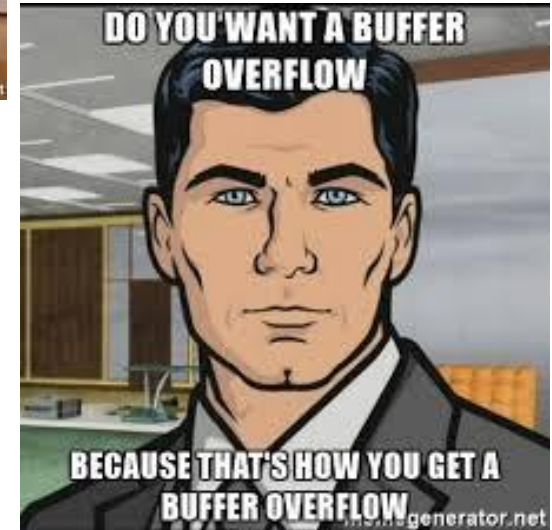
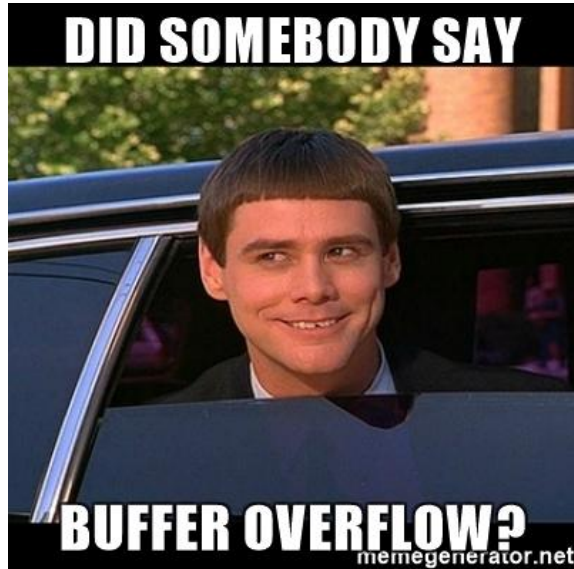
- User-based
 - PGP
 - S/MIME
- Infrastructure-based
 - STARTTLS
 - SPF
 - DKIM
 - DMARC

Email Security

- DKIM
 - Authentication, integrity
 - Your provider signs your email with their private key before sending
- DMARC
 - Nothing, it's just a set of rules of what to do with SPF and DKIM results
- SPF
 - Authentication
 - Verifies that the IP address sending the email is allowed to send emails from that domain
- STARTTLS
 - Confidentiality
 - Negotiates TLS between two servers before talking SMTP (mail language)
 - Fails open and only protects individual hops



Buffer Overflow Demo!



Questions??