Changes...

- The purpose of the Friday presentations is to explain the *concepts* that will be covered in the paper *without* going into lots of the notation and formalism
  - What background do we need to understand the paper?
  - What is the paper trying to show us how to do? What is the paper's solution?
  - How good is the paper's solution?
  - How does this paper fit in?

- The purpose of the reading guide is to make reading these notation-ridden papers easier
  - What does this funny symbol mean? How do we pronounce it?
  - How does the formal definition in the paper relate to the understandable definition?
  - Are there any errors in the paper?

- The purpose of Thursdays is to show how this theoretical stuff relates to the real world
  - Is it being used (correctly)?
  - Why do provably secure systems still break?
  - Why would we not use the secure constructions?

- Reading guides are due by Midnight *Monday*
- I'll have suggestions and corrections back to you by Midnight *Tuesday*
- Final submission is due by Noon *Thursday*
- Guides must be done in LaTeX. If no one in your group knows LaTeX, see me for an introduction

- Groups will now be *required* to give two practice presentations
  - The first on either Tuesday or Wednesday
  - The second on Thursday or Friday
- The presentation should be *complete* by the first run-through
- Short Thursday talks need to be run through once before Thursday

IND-CPA $\overset{?}{\leftrightarrow}$ IND-CCA

IND-CPA ← IND-CCA

IND-CPA + ? → IND-CCA

IND-CPA + INT-CTXT

→

IND-CCA

$$E(m \,||\, T(m))$$

$$E(m \,||\, T(m))$$

MAC-then-Encrypt

$$E(m) \,||\, T(E(m))$$

$$E(m) \,||\, T(E(m))$$

Encrypt-then-MAC

What is Encrypt-and-MAC?

What is Encrypt-and-MAC?

E(m) || T(m)

Encrypt-then-MAC is secure if the symmetric encryption is ? under plaintext attack and the MAC is ? under chosen message attack.

Encrypt-then-MAC is secure if the symmetric encryption is indistinguishable under plaintext attack and the MAC is strongly unforgable under chosen message attack.

Protocol	Scheme	Specifics	Security
SSH	Encrypt and MAC		
SSL	MAC then Encrypt		
IPSec	Encrypt then MAC	Doesn't matter	Secure

Protocol	Scheme	Specifics	Security
SSH	Encrypt and MAC		
SSL	MAC then Encrypt	Stream Cipher or CBC	
IPSec	Encrypt then MAC	Doesn't matter	Secure

MAC then Encrypt is secure if the encryption is a stream cipher (CTR) or CBC mode with a random IV

Protocol	Scheme	Specifics	Security
SSH	Encrypt and MAC		
SSL	MAC then Encrypt	Stream Cipher or CBC	Secure
IPSec	Encrypt then MAC	Doesn't matter	Secure

Protocol	Scheme	Specifics	Security
SSH	Encrypt and MAC	Encode then E and M	
SSL	MAC then Encrypt	Stream Cipher or CBC	Secure
IPSec	Encrypt then MAC	Doesn't matter	Secure

We have to look at specifics of SSH.

Protocol	Scheme	Specifics	Security
SSH	Encrypt and MAC	Encode then E and M	Broken
SSL	MAC then Encrypt	Stream Cipher or CBC	Secure
IPSec	Encrypt then MAC	Doesn't matter	Secure

CTR mode encryption is IND-CPA and HMAC is strongly unforgable. So we know that EtM with AES-CTR and HMAC-SHA-1 is secure. Right?

$$(ctr, C) = \text{AES-CTR}_e(ctr, M)$$

$$T = \text{HMAC-SHA-1}_m(C)$$

return (ctr, C, T)

$$(ctr, C) = \text{AES-CTR}_e(ctr, M)$$

$$T = \text{HMAC-SHA-1}_m(ctr||C)$$

return (ctr, C, T)

Is there a practical reason to use Encrypt and Mac?

Parallelizable: Can compute E(m) and M(m) simultaneously.

E(m) || M(m)

E(m || M(m))

E(m) || M(E(m)

As a practical matter, it would be nice to have something that is both secure and parallelizable.