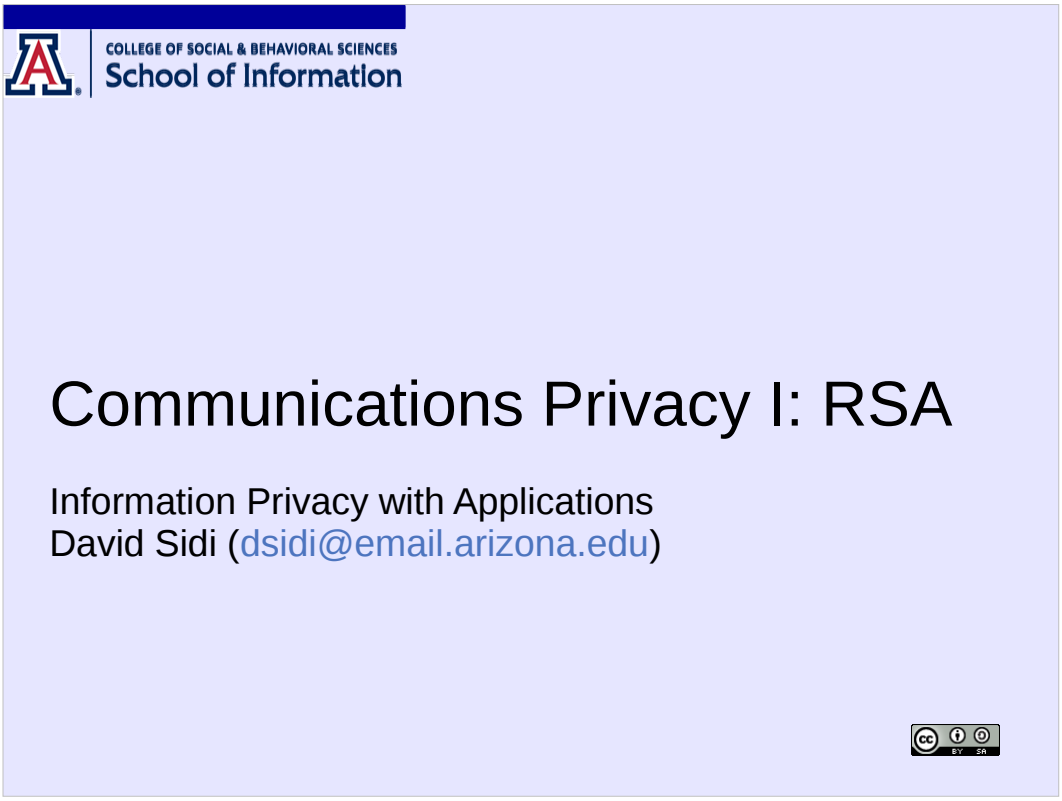
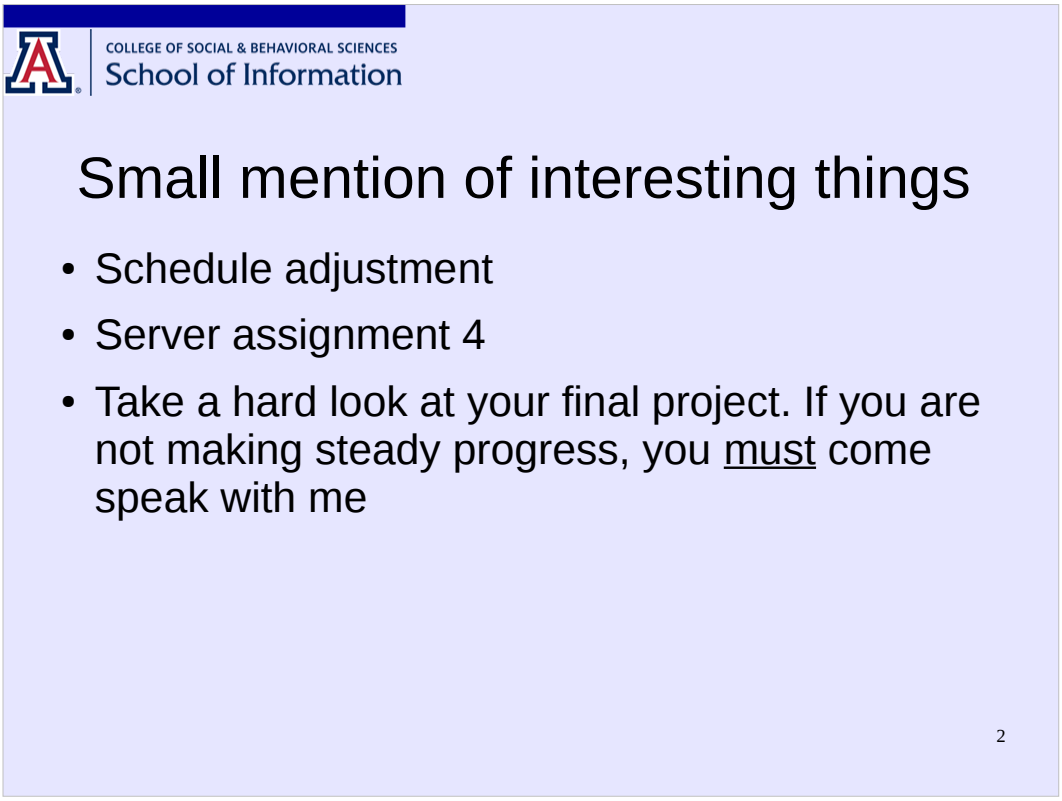



two topics from cryptography came up last time: Kerckhoff's Law and Diffie-Hellman. We will talk about those to set the stage for our main focus today, which is RSA. It's going to be a little bit short, to allow us to do some mitmproxy work, so we'll return to RSA next time.




We are going to skip the Boneh article for next time, and move to a discussion of TLS.

Now that we have finished the Tor material, we are ready for assignment 4: it will be due in one week.

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

# Cryptography is useful when it is difficult to secure a channel

- Confidentiality in FF voice communication requires that no unwanted third party is listening in
- However hard that is, phone communication presupposes it too. In addition, it requires that the call isn't intercepted while in transit
- Interception: Face to face < Copper wire < Radio link < Optical link (**POTL** 11)

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

# Different cryptographic systems have different strengths

- “A cryptosystem is considered secure when an opponent cannot break it under reasonable circumstances, in a reasonable amount of time, at a reasonable cost. The term “reasonable” is perforce vague.” (POTL 26)

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

# “Reasonable Circumstances”

- Ciphertext only
- Known plaintext: attacker can observe a plaintext and its encryption
- Chosen plaintext: attacker picks the plaintext to be encrypted
- Chosen ciphertext: attacker picks the ciphertext to be decrypted
  - non-malleability: the attacker cannot change the ciphertext so that the correspondnig plaintext is changed in a controlled way

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

# “Reasonable Time”

- “Workfactor:” number of operations required to break a cryptographic system
- What ‘operations’ means depends: they may not be elementary computer instructions
  - For example, if searching space of keys: operations are encryptions, which could be several hundred instructions

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

# Workfactors and their significance

- Assume perfectly parallel problems
- $2^{30}$: trivial (minutes) by one computer at 1 GhZ
- $2^{60}$: 11 to 12 days (with $2^{20}$ processors in parallel at $2^{20}$ instructions per second)
- $2^{90}$: 30 years (with $2^{30}$ processors in parallel at $2^{30}$ instructions per second)
- $2^{120}$: 30,000 years (with $2^{60}$ processors in parallel at $2^{60}$ instructions per second)

# Estimating workfactor significance: what could go wrong?

- RSA challenge: in 1977, it was estimated that the time taken to factor the 426 bit number would be **$4 \times 10^{16}$ years**
- n = 114381625757888867669235779976146612010218296721242362562561842935706935245733897830597123563958705058989075147599290026879543541
- This is from Martin Gardner's Scientific American article, and came to be known as RSA-129
- Solved in 1994 (~ **17 years** later)

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

# “Reasonable time” is relative

- How long is too long for decryption?
  - The Venona messages were studied for nearly 40 years in hopes that they would reveal the identities of spies who had been young men in the 1930s and who might have been the senior intelligence officers of the 1970s
- Sometimes keys are ephemeral, so are only helpful for a small window of messages going forward
  - What’s an example of an ephemeral key from Tor?

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

# “Reasonable cost”

- Example: NSA’s Utah Data Center
- The planned structure provides 1 to 1.5 million square feet (90,000–140,000 m2), with 100,000 square feet (9,000 m2) of data center space and more than 900,000 square feet (84,000 m2) of technical support and administrative space. It is projected to cost $1.5–2 billion. A report suggested that it will cost another $2 billion for hardware, software, and maintenance.

- “cryptography can best be thought of as a mechanism for extending the confidentiality and authenticity of one piece of information (the key) to another (the message).” (POTL 34)

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

# Key compromise means different things depending on the key's use

- Authentication keys can be revoked, and no authentications will still go through with those keys
- Keys used for privacy can also be revoked, but all messages ever sent with a key must be regarded as compromised
- A revoked key can be used in the future for old encryptions, but there is no corresponding notion of "old authentications"

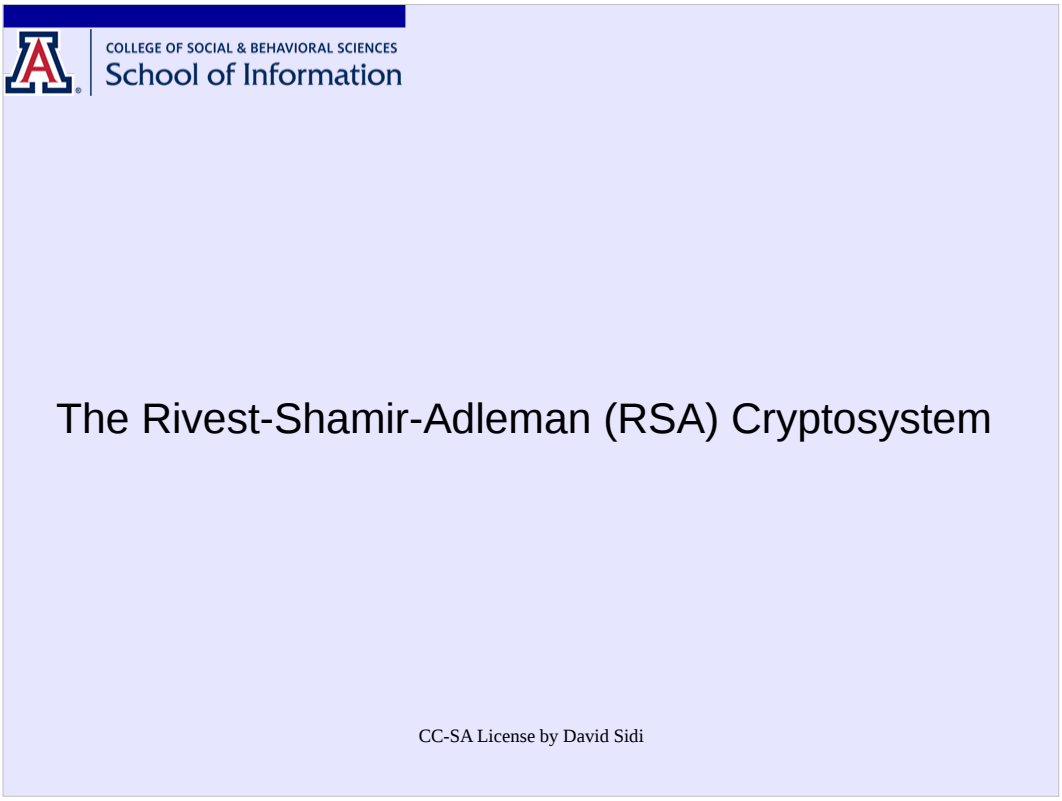
COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

"Cryptography is the only technique capable of providing security to messages transmitted over channels entirely out of the control of either the sender or the receiver." (POTL 35)

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

# Public Key Cryptography

- Key idea: Encryption key is public, decryption key is private
- Question: The private key is sometimes used by the *sender*, and the public one by the *recipient*. When?

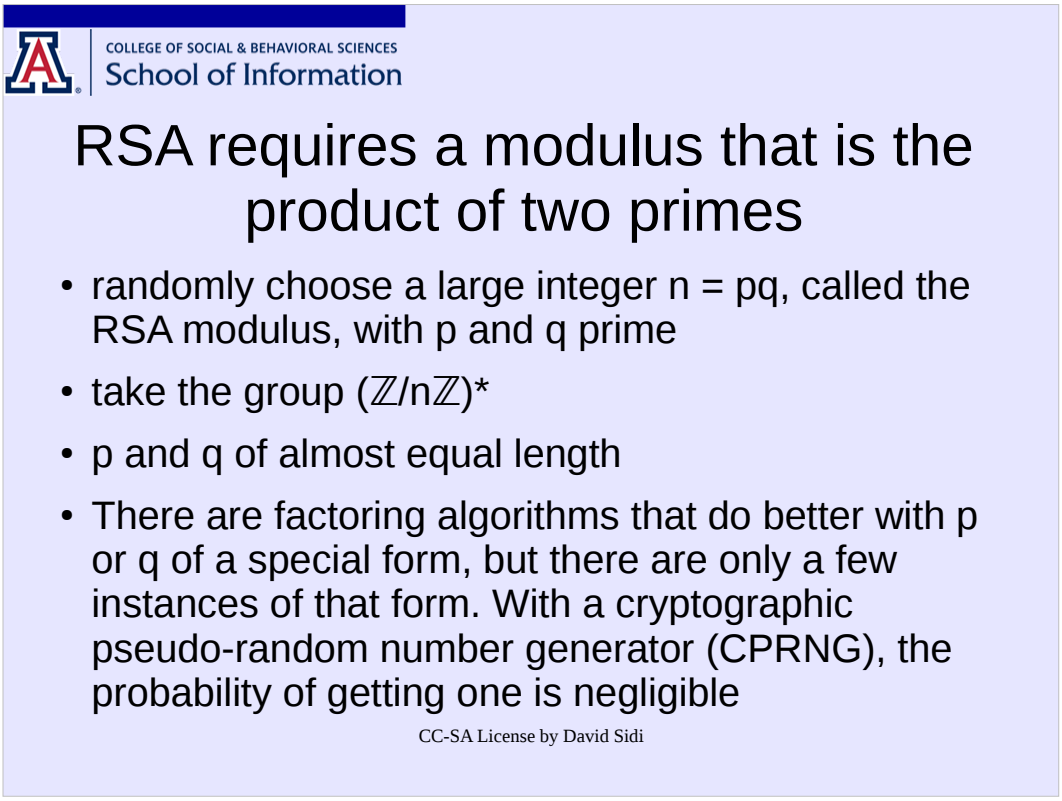



We're thinking about residue class rings here, as in **Z/nZ**. There, elements like **a,b,n** above are residue classes (quickly on board. NB: representation by smallest non-negative member is common).

Division works in rings like it does in the integers: loosely, **a** divides **n** if **a** can be multiplied by some **b** to get **n.**

When an element **n** of a residue class divides **1 (mod m)**, we call it `invertible.' (i.e., when **ax ≡ 1 (mod m)**)

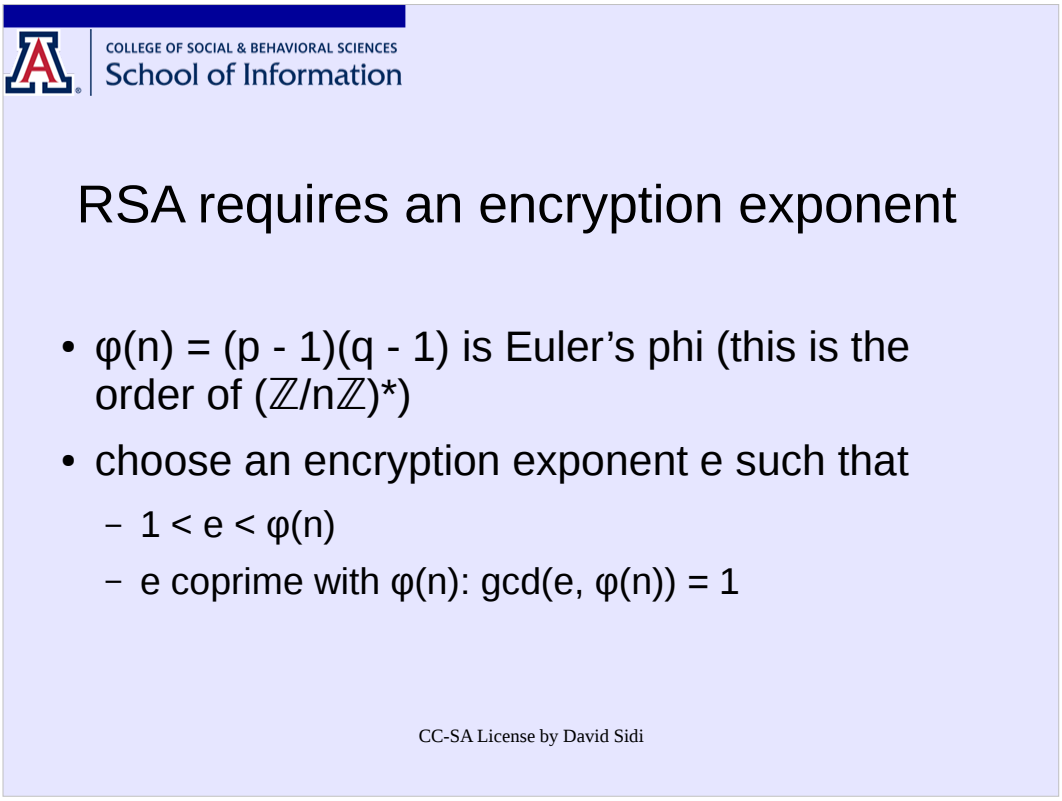
In general in a ring there is a question about which elements are invertible. In the unit group for a ring, that question evaporates: all elements are invertible. We'll work in the unit group for **Z/mZ** (that is, integers modulo m).




Here we're picking a modulus that has two primes as factors, and using it to form the unit group from the ring **Z/nZ**. (NB: **n** is the key length)

In general it is hard to factor a composite positive integer that is the product of two primes. Choosing p and q to be almost of equal length makes it harder, as do a few other choices.

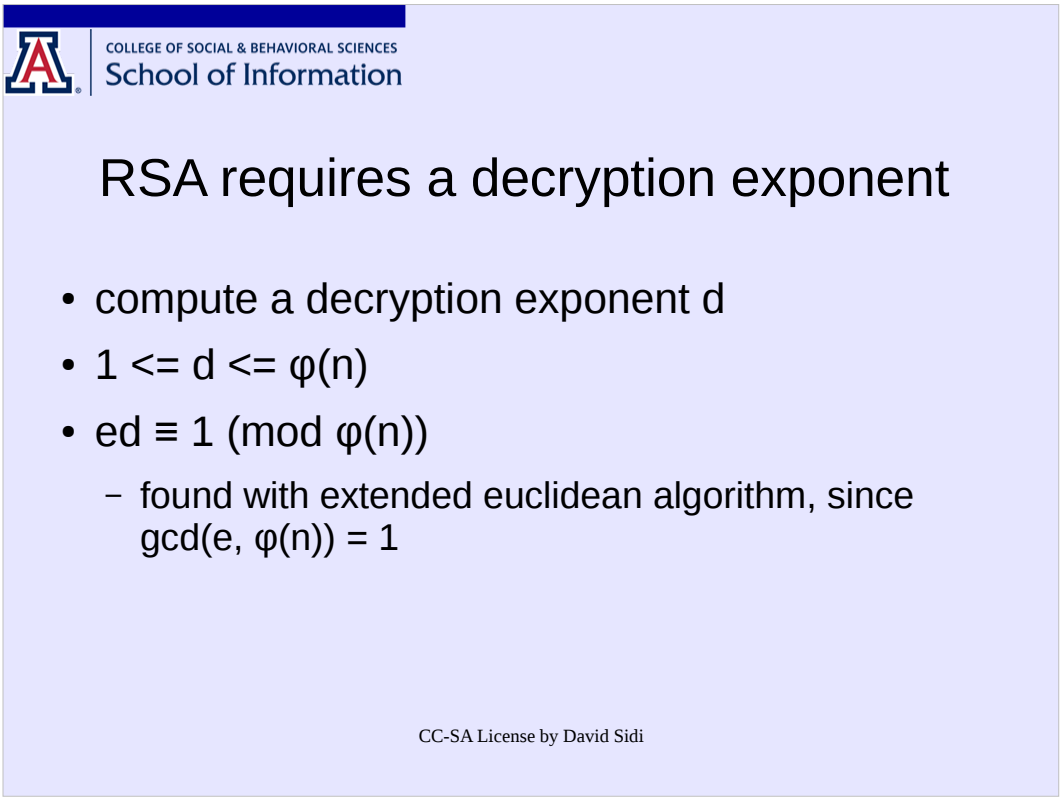
The infeasibility of factoring **n** is important here, since with **p** and **q** you can get an important value, **φ(n)=(p-1)(q-1)**, which can be used to compute the secret key from the public one.




When you pick **n** in the way we did as the modulus, there is a relationship between the prime factors of the residue class ring's modulus and the number of elements in the ring's unit group. That relationship is given by Euler's phi.

The restriction on choice of **e** here ensures that an encryption exponent **e** is chosen so that things powered by it are in the unit group: **e** is less than φ(n), and it's coprime with φ(n) (so invertible)

**(n,e)** becomes the public key.




to get d, we need the inverse of **e** in the group formed from the units of the previous group. There is a general way to get inverses when the thing to invert is coprime with the modulus, called the extended euclidean algorithm. Note: working in the unit group makes this work.

**d** becomes the private key

NB: if you know the value of φ(n), then you can compute the solution to ed ≡1 (mod φ(n)) efficiently using the extended Euclidean algorithm.

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

# RSA encrypts messages encoded as integers

- message is an integer m with 0 <= m < n
  - can encode $m_1 m_2 \cdots m_k$ as such an m; a block version of RSA
- encryption of a message m is $m^e \pmod n$; decryption is $(m^e)^d \pmod n$
- we need to show that $(m^e)^d = m$

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

# RSA relies on the difficulty of prime factorization

- choose a large number $n = pq$, with p,q prime
- $\varphi(n)$ is Euler's phi
- choose exponents e,d such that
  - $1 <= e,d <= \varphi(n)$
  - e coprime with $\varphi(n)$
  - $ed \equiv 1 \pmod{\varphi(n)}$
- message is an integer m with $0 < m < n$
- encryption of m is

  $m^e \pmod n$
- decryption is

  $(m^e)^d \pmod n$

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

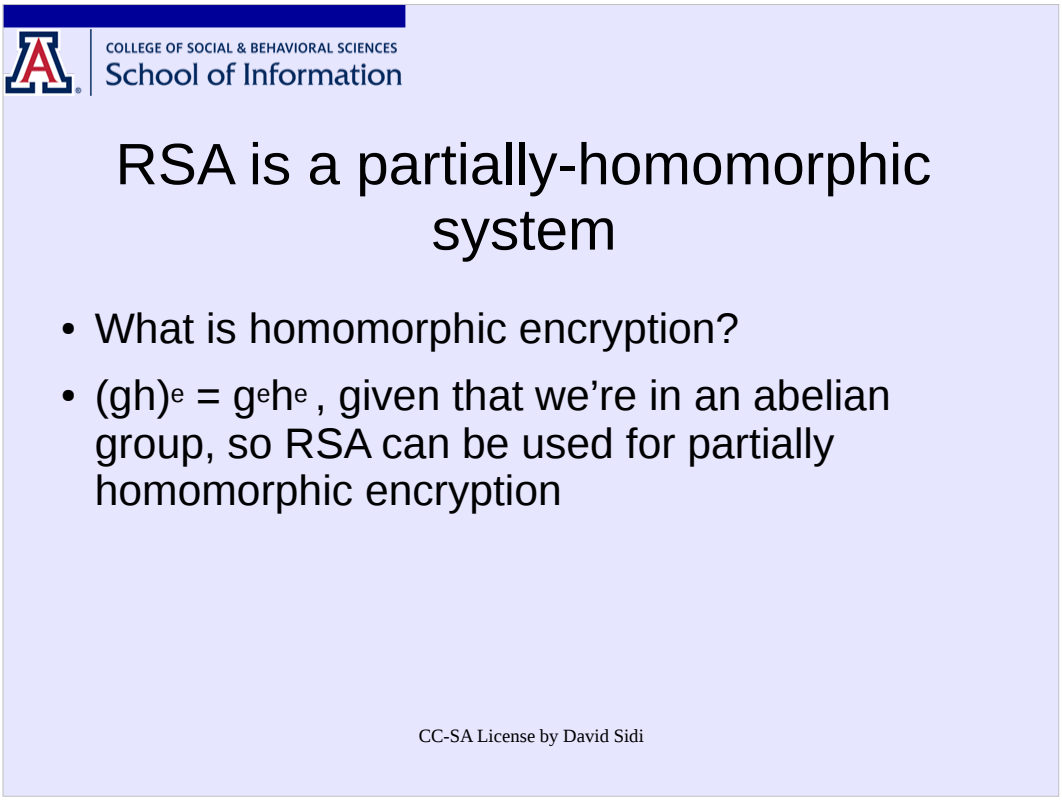
# RSA is a cryptosystem

- To show that RSA is a cryptosystem, we need to show that the encryption operation is invertible

# RSA is a cryptosystem

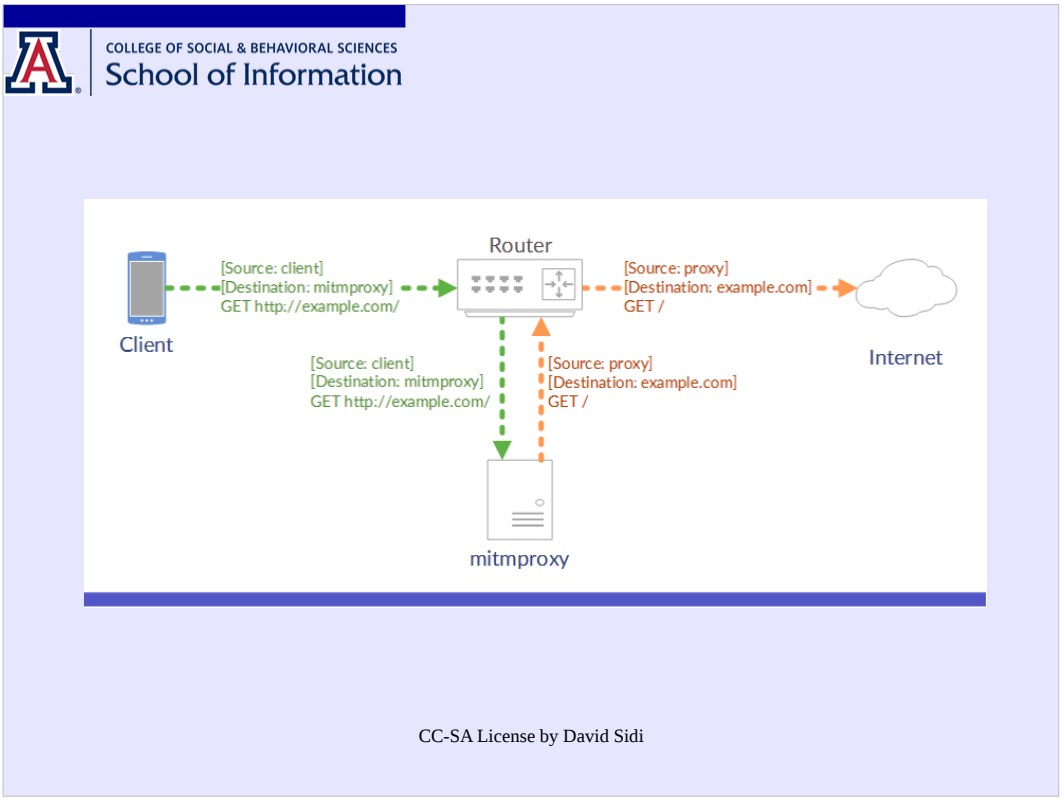
- $ed \equiv 1 \pmod{\varphi(n)}$ implies $ed = 1 + \ell\varphi(n)$
- $(m^e)^d = (m^{1+\ell\varphi(n)}) = m(m^{\ell\varphi(n)}) = m(m^{\ell(p-1)(q-1)}) = m(m^{(p-1)})^{\ell(q-1)}$
- If $p \mid m$, $(m^e)^d \equiv m \pmod{p}$ is trivial. (why?)
- Otherwise, by Fermat's little theorem, $m^{(p-1)} \equiv 1 \pmod{p}$, so $m(m^{(p-1)})^{\ell(q-1)} \equiv m \pmod{p}$
- The case is exactly similar for q, so we have $(m^e)^d \equiv m \pmod{pq}$
- $0 < m < pq$, so it is established that $(m^e)^d = m$

- The core idea of RSA is from algebra: exponentiation by an element coprime with the unit group order is an invertible automorphism
  - Klaus Lux occasionally teaches a cryptography course here; take it to learn more




Unit group is an abelian group.


D.C.'09

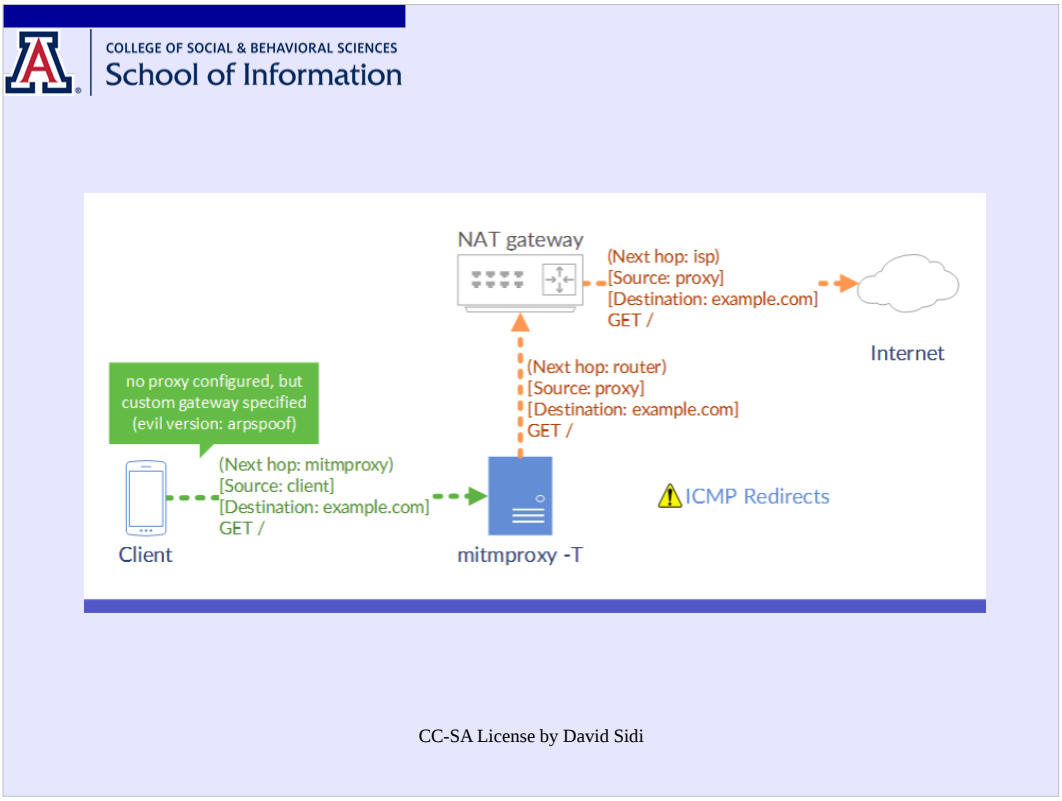



transparent vs. nontransparent proxying

nontransparent, remote proxy case

This is a remote proxy. What is the local proxy case?

Let's set up local nontransparent proxying on ourselves. (1) install conda, pip, and python 3.x if you don't have these already (2) create and activate a new environment called "mitm_sandbox" (3) install requisites for mitmproxy (4) install mitmproxy (5) set up proxy settings on your browser (6) run mitmproxy (8) access httpS://webauth.arizona.edu

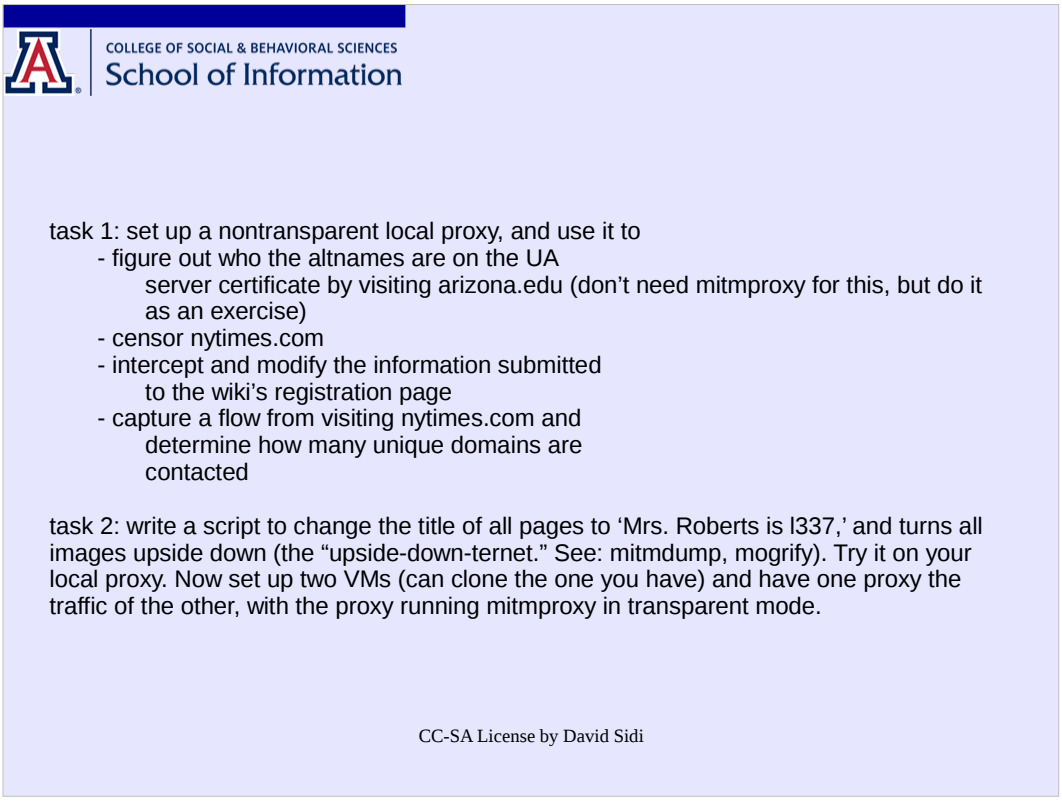



transparent case (note: -T): what does it mean to say traffic is “directed at the network layer?”

mentioned in the docs without comment: “arp spoofing”

What is arp (from last time)? So arp spoofing (AKA arp cache poisoning)?

Simple defense, whipped up in an hour or so: antipineapple




set up:

install VM (optional, but recommended for windows)

install conda

install mitmproxy (pip)

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

Done with that? Use stem to retrieve the full set of descriptors from a running tor process.

Consider: ControlPort, authentication method, getting full descriptor list (not just the ones being used), all in torrc

Using stem to connect to the Control port, then get the router status entries from the network status documents. Print whether the router is an exit, and its nickname and fingerprint.