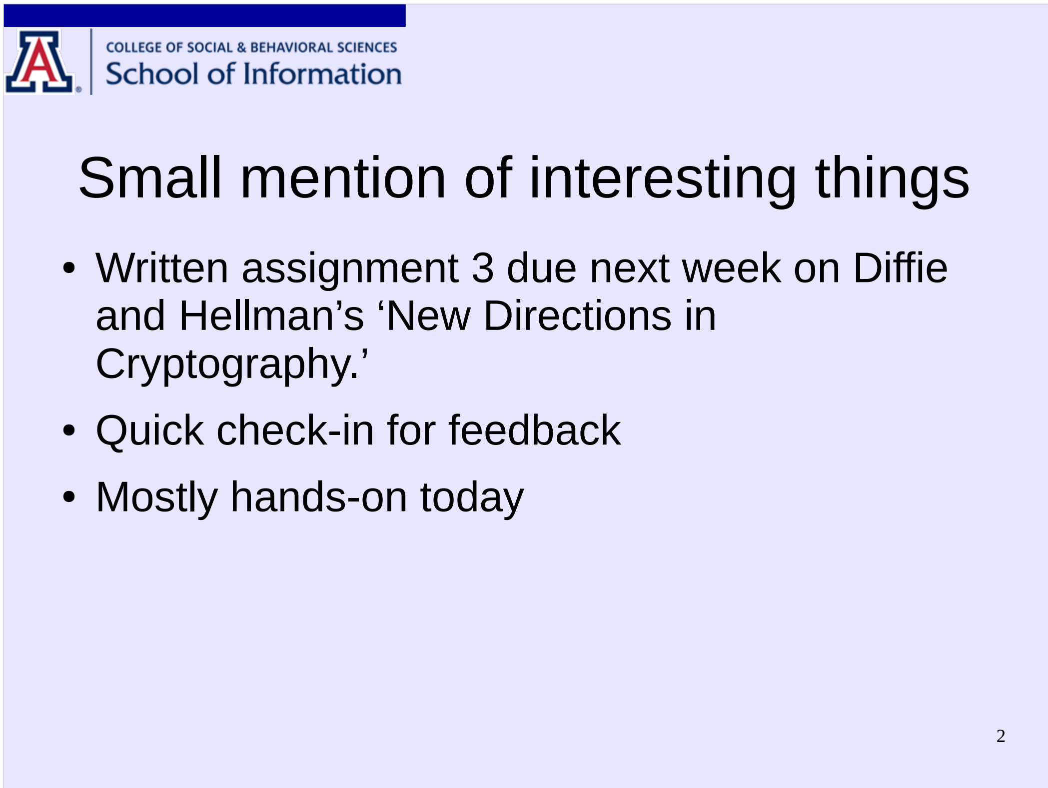
Communications Privacy II: RSA

Privacy Technology in Context
David Sidi (dsidi@email.arizona.edu)

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

The Rivest-Shamir-Adleman (RSA) Cryptosystem

CC-SA License by David Sidi

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 $ax \equiv 1 \pmod{m}$)

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).

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

RSA requires a modulus that is the product of two primes

- randomly choose a large integer n = pq, called the RSA modulus, with p and q prime
- take the group $(\mathbb{Z}/n\mathbb{Z})^*$
- p and q of almost equal length
- There are factoring algorithms that do better with p or q of a special form, but there are only a few instances of that form. With a cryptographic pseudo-random number generator (CPRNG), the probability of getting one is negligible

CC-SA License by David Sidi

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.

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

RSA requires an encryption exponent

- $\varphi(n) = (p - 1)(q - 1)$ is Euler's phi (this is the order of $(\mathbb{Z}/n\mathbb{Z})^*$)
- choose an encryption exponent e such that
  - $1 < e < \varphi(n)$
  - e coprime with $\varphi(n)$: $\gcd(e, \varphi(n)) = 1$

CC-SA License by David Sidi

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 $\varphi(n)$, and it's coprime with $\varphi(n)$ (so invertible)

(n,e) becomes the public key.

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

RSA requires a decryption exponent

- compute a decryption exponent d
- $1 <= d <= \varphi(n)$
- $ed \equiv 1 \pmod{\varphi(n)}$
  - found with extended euclidean algorithm, since $\gcd(e, \varphi(n)) = 1$

CC-SA License by David Sidi

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 $\varphi(n)$, then you can compute the solution to $ed \equiv 1 \pmod{\varphi(n)}$ 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 $(m^e)^d = m$

CC-SA License by David Sidi

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

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

CC-SA License by David Sidi

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

CC-SA License by David Sidi

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 $(m^e)^d = m$

CC-SA License by David Sidi

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

- 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

CC-SA License by David Sidi

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

RSA is a partially-homomorphic system

- What is homomorphic encryption?
- $(gh)^e = g^e h^e$, given that we're in an abelian group, so RSA can be used for partially homomorphic encryption

CC-SA License by David Sidi

Unit group is an abelian group.

D.C. '09

COLLEGE OF SOCIAL & BEHAVIORAL SCIENCES
School of Information

task 1: set up a nontransparent local proxy, and use it to
- figure out who the altnames are on the UA server certificate by visiting arizona.edu (don’t need mitmproxy for this, but do it as an exercise)
- censor nytimes.com
- intercept and modify the information submitted to the wiki’s registration page
- capture a flow from visiting nytimes.com and determine how many unique domains are contacted

task 2 (optional, or you can move to the tor task): write a script to change the title of all pages to ‘Mrs. Roberts is l337,’ and turns all images upside down (the “upside-down-ternet.” See: mitmdump, mogrify). Try it on your local proxy. Now set up two VMs (can clone the one you have) and have one proxy the traffic of the other, with the proxy running mitmproxy in transparent mode.

CC-SA License by David Sidi

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.

CC-SA License by David Sidi