COMP424
Computer Security

Prof. Wiegley
jeffw@csun.edu

Rivest, Shamir & Adelman (RSA)
Implementation

"Relatively prime"

- Prime: n, is "prime" if its only two factors are 1 and n. (and $n \neq 1$).
- Relatively prime: Two numbers x and y are relatively prime if x and y do not share any factors (other than 1) in common.
  - Example: 18 and 27 are not relatively prime because they share the factor 3.

$$\begin{aligned} 18 &= 2 \cdot 3 \cdot 3 \\ 27 &= 3 \cdot 3 \cdot 3 \end{aligned}$$

  - Example: 42 and 55 are relatively prime because none of 42's factors are a factor of 55. (Even though neither is prime.)

$$\begin{aligned} 42 &= 2 \cdot 3 \cdot 7 \\ 55 &= 5 \cdot 11 \end{aligned}$$

"Modulo"

- Programmers identify "modulo" as the operator that yields the remainder of an integer division.
- Mathematicians use "modulo" to define a mathematical space. Notation is slightly different:

$$b^x = 1 \bmod n$$

  means that in the space "modulo n", b^x is equal to 1. (As opposed to a programmer's perspective which might read b^x is equal to $1\%n$.)
- This requires some understanding of basic modulo algebraic rules.

"Modulo algebra (multiplication)"

- $ab \bmod n = (a \bmod n \cdot b \bmod n) \bmod n$

  The value of $ab \bmod n$ can be calculated by first calculating $a \bmod n$ and then $b \bmod n$, multiplying the results and taking the modulo of the result.

  - Example: let $n = 10$.

$$\begin{aligned} 16 \cdot 19 \bmod 10 &= (16 \bmod 10) \cdot 19 \bmod 10) \bmod 10 \\ &= (6 \cdot 9) \bmod 10 \\ &= 54 \bmod 10 \\ &= 4 \end{aligned}$$

    The outer modulo operator must remain because the product of the inner portions may exceed n.

"Modulo algebra (addition)"

- Similarly, $a + b \bmod n = (a \bmod n + b \bmod n) \bmod n$
  - Example: let $n = 10$ again.

$$\begin{aligned} 16 + 19 \bmod 10 &= (16 \bmod 10) + 19 \bmod 10) \bmod 10 \\ &= (6 + 9) \bmod 10 \\ &= 15 \bmod 10 \\ &= 5 \end{aligned}$$

Again, The outer modulo operator must remain because the sum of the inner portions may also exceed n.

"Modulo algebra (inverse)"

- In standard algebra the inverse of n is $\frac{1}{n}$. This allows that $n\frac{1}{n} = 1$.
- Similarly, in modulo space n, for some value, x, the value y is the inverse of x if $xy = 1 \bmod n$.
  - Example: Let $n = 10$ and $x = 3$. y is the inverse of x when $y = 7$. Because $7 \cdot 3 = 21$ and $21 = 1 \bmod 10$.

"RSA"

In order to encrypt and decrypt a message, RSA relies on three values:

$$n$$

$$e$$

$$d$$

- The public key (used to encrypt messages) is the tuple (e, n).
- The private key (used to decrypt messages) is the tuple (d, n).

The remaining slides present how n, e and d are computed.

Step 1: obtaining n

1. Pick a prime number, p.
2. Pick a different prime number, q

Any prime numbers will work though in practice, to be secure, p and q should not be twin primes or small.

n is simply the product of the two primes p and q.

$$n = pq$$

- Example: $p = 73$, $q = 61$ yields $n = 4453$

Step 2: obtaining e

1. Pick any value e such that e is relatively prime to $(p-1)(q-1)$

- Note: Any prime number other p or q is automatically relatively prime.
- However, when p, q are very large (1024 bits) then e should also be proportionately large and it becomes computationally difficult to test whether a chosen value is prime.
- Choosing an even value of e is futile. p and q are prime and therefore must be odd. $(p-1)(q-1)$ is therefore even and all even numbers share the factor 2. Thus e would not be relatively prime to $(p-1)(q-1)$.

But there is a method for quickly testing if a chosen value is relatively prime to $(p-1)(q-1)$.

Euclidean Algorithm (used to sift possible values of e).

1. Choose a possible value for e.
   - Example $e = 35$.
2. Test if e is relatively prime to $(p-1)(q-1)$ using the Euclidean algorithm.

   Start by setting up a series of equation of the form:

$$E_i : \quad \alpha_i - \beta_i \cdot \delta_i \quad = \quad \gamma_i$$

   where $\beta_i = \lfloor \frac{\alpha_i}{\delta_i} \rfloor$. For each equation $\alpha_i = \delta_{i-1}$ and $\delta_i = \gamma_{i-1}$. This is esentially subtracting the largest possible multiple of δ from α. δ and the remainder, γ, must share a common factor in order for α and δ to share a common factor. Continue the equations until $\gamma_i = 0$.

Euclidean Algorithm example

Begin with $\alpha_0 = (p-1)(q-1)$ and $\delta_0 = e$.

$$
\begin{array}{lrcrcrcl}
E_0: & (p-1)(q-1) & - & \alpha & \cdot & e & = & \beta \\
 & (73-1)(61-1) & - & \alpha & \cdot & 35 & = & \beta \\
 & 72 \cdot 60 & - & \alpha & \cdot & 35 & = & \beta \\
 & 4320 & - & 123 & \cdot & 35 & = & 15 \\
E_1: & 35 & - & 2 & \cdot & 15 & = & 5 \\
E_2: & 15 & - & 3 & \cdot & 5 & = & 0
\end{array}
$$

Since the second last result is 5 then e and $(p-1)(q-1)$ share the factor 5. So e is not relatively prime to $(p-1)(q-1)$.

Increment e by 2 and repeat test.

1. Since the first choice for e was unacceptable, increment e by 2 and test again. Repeat this procedure until e is relatively prime to $(p-1)(q-1)$.

- Example $e = 35 + 2 = 37$.

1. Test if e is proven to be relatively prime to $(p-1)(q-1)$.

$$
\begin{array}{lrcrcrcl}
E_0: & (p-1)(q-1) & - & \alpha & \cdot & e & = & \beta \\
 & (73-1)(61-1) & - & \alpha & \cdot & 37 & = & \beta \\
 & 72 \cdot 60 & - & \alpha & \cdot & 37 & = & \beta \\
 & 4320 & - & 116 & \cdot & 37 & = & 28 \\
E_1: & 37 & - & 1 & \cdot & 28 & = & 9 \\
E_2: & 28 & - & 3 & \cdot & 9 & = & 1 \\
E_3: & 9 & - & 9 & \cdot & 1 & = & 0
\end{array}
$$

Since the second last result is 1 then the smallest common factor of e and $(p-1)(q-1)$ is 1 and therefore e is relatively prime to $(p-1)(q-1)$.

Step 3: Calculating d

- d is the inverse of e mod $(p-1)(q-1)$. basically, $(d \cdot e = 1 \bmod n)$.
- By substituting values from the euclidean proof of e, d can be calculated:

$$
\begin{aligned}
e \cdot d &= 1 \bmod (p-1)(q-1) \\
&= 28 - 3 \cdot 9 \\
&= 28 - 3 \cdot (37 - 1 \cdot 28) \\
&= -3 \cdot 37 + 4 \cdot 28 \\
&= -3 \cdot 37 + 4 \cdot (4320 - 116 \cdot 37) \\
&= 4 \cdot 4320 - 467 \cdot 37 \\
&= 17280 - 467 \cdot 37 \\
&= 0 - 467 \cdot 37 \\
e \cdot d &= -467 \cdot e \\
d &= -467
\end{aligned}
$$

Negative d?

- A negative value for d is slightly disappointing.
- The modulo $(p-1)(q-1)$ space consists of the positive integers (0,. . . ,(p-1)(q-1)-1)
- if the modulo $(p-1)(q-1)$ space is thought of as a "ring" then traveling x units in the negative direction yields the same point as traveling $(p-1)(q-1)-x$ in the positive direction.
- if d is negative, subtract it from $(p-1)(q-1)$ to yield an equivalent positive value.

$$d = (p-1)(q-1) - d$$
$$d = 4320 - 467$$
$$d = 3853$$

$(e \cdot d = 37 \cdot 3853 = 142561 = 33 \cdot 4320 + 1 = 1 \bmod 4320)$

Final results

- Calculations for computing a public/private key are complete.
  - Public key: $(e, n) = (37, 4453)$
  - Private key: $(d, n) = (3853, 4453)$
- RSA is secure so long as n cannot be factored easily.
  - The attacker knows n because it is published as part of the public key.
  - n only has two factors, p and q. (p and q are prime.)
  - If an attacker could recover p and q then they have obtained $(p-1)(q-1)$.
  - Knowing e and $(p-1)(q-1)$ allows an attacker to easily compute d using the Euclidean algorithm since $e \cdot d = 1 \bmod (p-1)(q-1)$.

Encryption

So now we have a public key of $\{e, n\}$ and a private key of $\{d, n\}$ we can decrypt a message P by:

$$C = P^e \bmod n.$$

Similarly an encrypted message, C, can be decrypted to yield the original message, P, by:

$$P = C^d \bmod n.$$

This works because if $P = C^d$ and $C = P^e$ then

$$P = P^{e^d} = P^{e \cdot d} = P^1 = P.$$

ModPow

But P^e is going to be rediculously large. So large that modern calculators cannot carry out this computation.[a]

A method is needed to constrain the calculation to small numbers and we will use modulo arithmetic to provide this ability.

First, start with $P^1 \bmod n$. We can see from the axioms that

$$P^2 = (P^1 \bmod n)(P^1 \bmod n) \bmod n.$$

In general:

$$P^{a \cdot b} = (P^a \bmod n)(P^b \bmod n) \bmod n.$$

So let's work with powers of 2 to aid the calculation of P^e.

[a] Some calculators, such as the HP, series have modulo arithmetic function that can handle this calculation.

Powers of 2

Let's take an example using the key computed earlier:

$$P = 101, e = 75, n = 391.$$

First, notice that

$$P^e = 101^{75} = 101^{64} \cdot 101^8 \cdot 101^2 \cdot 101^1,$$

Where the power (75) has been broken down into powers of 2.

We could have broken it down in many ways but powers of two will decrease our work the most.

So first calculate all the values of 101 raised to a power of 2.

Calculating exponents of powers of 2

The first power of 2 is easy:

$$101^1 \bmod n = 101.$$

Now, for all other powers of two we can use the axiom as a trick:

$$101^x \bmod n = (101^{\frac{x}{2}} \bmod n)(101^{\frac{x}{2}} \bmod n) \bmod n.$$

Since we are only interest in powers of 2, $\frac{x}{2}$ will simply be the value computed in the previous iteration.

The computation

Remembering that $n = 391$, we have:

$$
\begin{aligned}
101^1 \bmod 391 &= 101 \\
101^2 \bmod 391 &= (101)(101) \bmod 391 &&= 35 \\
101^4 \bmod 391 &= (35)(35) \bmod 391 &&= 52 \\
101^8 \bmod 391 &= (52)(52) \bmod 391 &&= 358 \\
101^{16} \bmod 391 &= (358)(358) \bmod 391 &&= 307 \\
101^{32} \bmod 391 &= (307)(307) \bmod 391 &&= 18 \\
101^{64} \bmod 391 &= (18)(18) \bmod 391 &&= 324.
\end{aligned}
$$

Now, these values can be used to quickly compute 101^y where $y \leq 127$.

The computation

$$\begin{aligned} 101^{75} &= (101^{64} \cdot 101^{8} \cdot 101^{2} \cdot 101^{1}) \bmod 391 \\ &= (324 \cdot 358 \cdot 35 \cdot 101) \bmod 391 \end{aligned}$$

So life is a bit simpler but we still have a long string of factors that could produce a number larger than our calculator/computer can deal with.

To reduce the number of factors we can make use

$$P^{a \cdot b} = (P^{a} \bmod n)(P^{b} \bmod n) \bmod n.$$

Combining factors

This will enable us to combine factors two (or more) at a time.

$$\begin{aligned}(324 \cdot 358) \bmod 391 &= (324 \bmod 391 \cdot 358 \bmod 391) \bmod 391 \\ &= (324 \cdot 358) \bmod 391 \\ &= 115992 \bmod 391 \\ &= 256.\end{aligned}$$

Similarly:

$$\begin{aligned}(35 \cdot 101) \bmod 391 &= (35 \bmod 391 \cdot 101 \bmod 391) \bmod 391 \\ &= (35 \cdot 101) \bmod 391 \\ &= 3535 \bmod 391 \\ &= 16.\end{aligned}$$

Combining factors

So

$$
\begin{aligned}
101^{75} &= (101^{64} \cdot 101^{8} \cdot 101^{2} \cdot 101^{1}) \bmod 391 \\
&= (324 \cdot 358 \cdot 35 \cdot 101) \bmod 391 \\
&= (256 \cdot 16) \bmod 391 \\
&= 4096 \bmod 391 \\
&= 186
\end{aligned}
$$

The cipher text message, after encryption, is therefore 186.

The proof that decryption yields the original message is left as an exercise for the reader.

Proving the correctness of RSA

Lagrange's theorem states:

$$b^{\phi(n)} = 1 \bmod n.$$

Where $\phi(n)$ = the number of integers less than n that are relatively prime to n.

For RSA, let $n = pq$. (you can guess where RSA started their thinking now.)

Then, how many relatively prime integers are there in $\phi(pq)$?

Combining factors

Take pq, We know that p and q are prime so we can determine $\phi(pq)$

remove $p, 2p, 3p, 4p, \ldots, qp$, this removes q items.

remove $q, 2q, 3q, 4q, \ldots, pq$, this removes p items.

Now, you need back one of the pq that was removed twice.

So,

$$\begin{aligned} \phi(pq) &= pq - p - q - 1 \\ &= (p-1)(q-1). \end{aligned}$$

Therefore,

$$b^{\phi(n)} = b^{(p-1)(q-1)} = 1 \bmod n.$$

Keep that in mind, you'll need it in a second.

RSA proof

$$e \cdot d \quad = \quad 1 \bmod (p-1)(q-1)$$

We're working in modulo so we could have gone "around" the ring some arbitrary number of times. Let l be the number of wraps. Then

$$\begin{aligned}
e \cdot d &= 1 + l(p-1)(q-1) \\
P^{e^d} &= P^{e \cdot d} \\
P^{e \cdot d} &= P^{1+l(p-1)(q-1)} \\
&= P^1 \cdot P^{l(p-1)(q-1)} \\
&= P^1 \cdot (P^{(p-1)(q-1)})^l \\
&= P^1 \cdot (1 \bmod n)^l \quad \text{[By lagrange's substitution!]} \\
&= P^1 \cdot (1)^l \\
&= P^1 \cdot 1 \\
&= P^1 \\
&= P
\end{aligned}$$

Conclusion

By using Lagrange's theorem we have proven that

$$(P^e)^d = P,$$

and thus RSA works as advertised so long as our constraints about selecting p, q and e are met.