Chapter 00

Introducing Foundations

Cryptography – The RSA Public Key Cryptosystem

Contents:

Cryptography – The RSA Public Key Cryptosystem

The Rivest-Shamir-Adleman (RSA) cryptosystem uses

all the ideas we have introduced in this lecture note.

It derives strong guarantees of security by ingeniously exploiting the wide gulf between

- the polynomial-time computability of certain number-theoretic tasks
  - modular exponentiation,
  - greatest common divisor,
  - primality testing and
  - the intractability of others (factoring).

Cryptography – The RSA Public Key Cryptosystem

How to encrypt and decrypt the message using the RSA cipher:

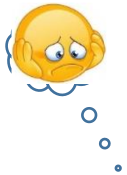
- Pick two large integers p and q,

  [say, in the order of several hundred digits each, and are

  virtually certain to be prime].

- To encrypt a plaintext message M using the RSA cipher, a person needs to know the publicly available values of
  - pq and
  - integer e
- Only the person, *who knows the individual values of p and q,* can *decrypt* an encrypted message M.

Cryptography – The RSA Public Key Cryptosystem



Case Study:

To set up an RSA cipher. Elain chooses:

- two prime numbers, $p = 5$, $q = 11$, and then computes $n = pq = 55$,
- a positive integer $e = 3$, which is relatively prime to $(p-1)(q-1) = 40$.

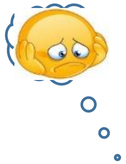
To be distributed widely are public keys:

- $n = 55$, the product of two numbers p and q
- $e = 3$.
- [Elain keeps p and q as the secret key.]

[The effectiveness of the system is

- the secrecy of the cipher: two distinct large integers p, q
  - both are in the order of several hundred digits each
  - both are virtually certain to be prime.
- And pick a very large e which is relatively prime to $(p - 1)(q – 1)$.]

Cryptography – The RSA Public Key Cryptosystem



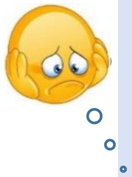
Case Study:

The RSA cipher works only on numbers.
Elain informs people how she will interpret the numbers in the message *according to the following guidelines:*

- Encodes letters of the alphabet the same way as was done for the Caesar cipher:
  A = 1, B = 2, C = 3, …, H = 8, I = 9, …, Z = 26.

- Divide long messages into blocks of messages
  - e.g., each block has a single, numerically encoded letter of the alphabet.

Cryptography – The RSA Public Key Cryptosystem



Case Study:

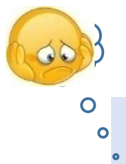
Sending Alex a message, she requires, according to the given guidelines:

- breaks the message into blocks,
    - each contains a single letter (or could be multiple letters).
- Finds the numeric equivalent for each block.
- Converts each block plaintext M into ciphertext C:

$$C = M^e \bmod pq. \qquad \text{........(RSA 0.4.5)}$$

- Anyone, who knows modular arithmetic, can use these public keys to encrypt a message to be sent to Alex since both pq and e are public keys.
- (Alex receives the ciphertext C for the plaintext M in a block of several blocks.)

Cryptography – The RSA Public Key Cryptosystem



Case Study:

Example 0.1.4.9 Encrypting a Message Using RSA Cryptography

For sending Alex a ciphertext of the message HI, Elain

- computes the ciphertext (i.e., the encrypted message) for the message HI.
  - Divide the message into two blocks: the H and the I.
  - Encode H as 08, or 8.
  - Use formula $C = M^e \pmod{pq}$ to compute the ciphertext for H:

    $$C = 8^3 \bmod 55 = 512 \bmod 55 = 17.$$

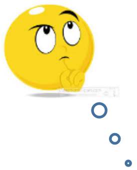
  - Encode I as 09, or 9.
  - Compute the ciphertext for I:

    $$C = 9^3 \pmod{55} = 729 \bmod 55 = 14.$$

Then, Elain sends Alex the encrypted message 1714.

RSA Cryptography

Case Study:



Example 0.1.4.10 Decrypting a Message Using RSA Cryptography

Received the encrypted message 1714. To obtain the plain message, Alex:

- computes the decryption key *d, a positive inverse to e modulo (p – 1)(q – 1).*
- Applies the formula

  $$M = C^d \bmod pq. \qquad \ldots\ldots\ (\text{RSA } 0.4.6)$$

  to decrypt the encrypted message (the ciphertext) C.

For guaranteeing the decryption to produce the original message,

- M must be less than pq because $M + k*pq \equiv M \pmod{pq}$.
  - The requirement of larger p and q (in the order of several hundred digits each) does not cause problems.
  - Break long messages into blocks of symbols to meet the restriction
    - Such as, including several symbols in each block to present decryption based on knowledge of letter frequencies.

RSA Cryptography

Case Study: Only Alex knows p = 5 and q = 11. Then he can computes (p-1)(q-1).

Example: Find a positive inverse for 3 modulo 40. (Note that e = 3; (p-1)(q-1) = 40;)

i.e., find a positive integer x such that $3x \equiv 1 (\text{mod } 40)$, or equivalently $x \equiv \boldsymbol{3^{-1}} (\text{mod } 40)$.

Solution:

Find a linear combination of 3 and 40 that equals 1.

$\gcd(\underline{40}, \underline{3})$ $\quad \underline{40} = 13 * \underline{3} + 1$. This yields $1 = 1*\underline{40} - 13*\underline{3}$. (1)

$= \gcd(\underline{3}, \underline{1})$ $\quad \underline{3} = 3 * \underline{1} + 0$. This yields $0 = 1*\underline{3} - 3*\underline{1}$ (2)

$= \gcd(\underline{1}, \underline{0}) = 1$ $\quad \underline{1} = 0 * \underline{0} + 1$. This yields $1 = 1*\underline{1} - 0*\underline{0}$ (3)

x = q * y + r
x = r + q *y
x ≡ r mod y

Take the 3rd equation, $1 = 1*\underline{1} - 0*\underline{0} = 1*\underline{1} = 1 * (1 * \underline{40} - 13 * \underline{3})$.

This $1 = 1 * (1 * \underline{40} - 13 * \underline{3})$ yields $(-13) * \underline{3} = 1 + (-1) * \underline{40}$, which is,

by definition of congruence modulo n,

$(-13) * \underline{3} \equiv 1 (\text{mod } 40)$, or, equivalently, $(-13) \equiv \boldsymbol{3^{-1}} (\text{mod } 40)$.

This result is: -13 is an inverse for 3 mod 40.

To find a positive inverse, compute -13 + 40 which yields 27, and

$27 \equiv (-13)(\text{mod } 40)$ because $27 - (-13) = 40$.

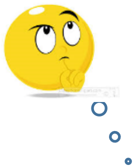
So, by Theorem 0.1.4.3(3), $ab \equiv cd \ (\text{mod } n)$,

$\mathbf{27} * 3 \equiv (-13) * 3 \equiv 1 \ (\text{mod } 40)$,

By the transitive property of congruence modulo n, 27 is a positive integer that is an inverse for 3 modulo 40.

a ≡ b mod n
iff a = b + kn
Iff n | (a – b)
iff a mod n = b mod n.

RSA Cryptography



Case Study:

Example 0.1.4.10 Decrypting a Message Using RSA Cryptography

- Alex knows $pq = 55$ and $e = 3$ as everyone has.
- Alex also knows the secret key: $p = 5$ and $q = 11$, allowing him to compute $(p-1)(q-1) = 40$.
- He finds the decryption key 27, a positive inverse for 3 modulo 40.
- He then decrypts the encrypted message C by computing

$$M = C^d \bmod pq = 17^{27} \bmod 55.$$

- The residues obtain when 17 is raised successively to $2^4 = 16$.

$$27_{10} = 11011_2 = 2^4 + 2^3 + 2 + 1 = 16 + 8 + 2 + 1$$

$$17 \bmod 55 = 17$$

$$17^2 \bmod 55 = 14$$

$$17^4 \bmod 55 = (17^2)^2 \bmod 55 = (17^2 \bmod 55)^2 \bmod 55 = (14)^2 \bmod 55 = 31$$

$$17^8 \bmod 55 = (17^4)^2 \bmod 55 = (17^4 \bmod 55)^2 \bmod 55 = (31)^2 \bmod 55 = 26$$

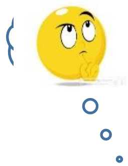
$$17^{16} \bmod 55 = (17^8)^2 \bmod 55 = (26)^2 \bmod 55 = 16$$

Then $17^{27} = 17^{16+8+2+1} = 17^{16} * 17^8 * 17^2 * 17^1$.

RSA Cryptography

Case Study:

Example 0.1.4.10 Decrypting a Message Using RSA Cryptography



…. $27 = 16 + 8 + 2 + 1 = 2^4 + 2^3 + 2 + 1.$

$17 \bmod 55 = 17$

$17^2 \bmod 55 = 14$

$17^4 \bmod 55 = (17^2)^2 \bmod 55 = (17^2 \bmod 55)^2 \bmod 55 = (14)^2 \bmod 55 = 31$

$17^8 \bmod 55 = (17^4)^2 \bmod 55 = (17^4 \bmod 55)^2 \bmod 55 = (31)^2 \bmod 55 = 26$

$17^{16} \bmod 55 = (17^8)^2 \bmod 55 = (26)^2 \bmod 55 = 16$

Then $17^{27} = 17^{16+8+2+1} = 17^{16} * 17^8 * 17^2 * 17^1.$

Thus, $17^{27} \bmod 55 = (17^{16} * 17^8 * 17^2 * 17^1) \bmod 55$

$=[(17^{16} \bmod 55)(17^8 \bmod 55)(17^2 \bmod 55)(17^1 \bmod 55)] \pmod{55}$

$=(16 * 26 * 14 * 17) \pmod{55}$

$=((16 * 26)\pmod{55} * (14 * 17)\pmod{55})\pmod{55}$ or $\equiv 99008 \pmod{55}$

$= 8 \pmod{55} \equiv 8.$

Hence $17^{27} \bmod 55 \equiv 8$.

Thus, the plaintext of the first part of Elain's message is 8 or 08.

In the last step, Alex finds the letter corresponding to 08, which is H.

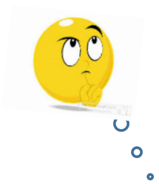
RSA Cryptography

Case Study:

Example 0.1.4.10 Decrypting a Message Using RSA Cryptography

Likewise, Alex found 14 to be 9, which corresponds to the letter I.

Alex uses the same decryption key 27, which is a positive inverse for 3 modulo 40.



For decrypting the ciphertext C, he computes $M = C^d \bmod pq = 14^{27} \bmod 55.$

$27 = 2^4 + 2^3 + 2 + 1 = 16 + 8 + 2 + 1.$

$14 \bmod 55 = 14$

$14^2 \bmod 55 = 31$

$14^4 \bmod 55 = (14^2)^2 \bmod 55 = (14^2 \bmod 55)^2 \bmod 55 = (31)^2 \bmod 55 = 26$

$14^8 \bmod 55 = (14^4)^2 \bmod 55 = (14^4 \bmod 55)^2 \bmod 55 = (26)^2 \bmod 55 = 16$

$14^{16} \bmod 55 = (14^8)^2 \bmod 55 = (16)^2 \bmod 55 = 36$

Then $14^{27} = 14^{16+8+2+1} = 14^{16} * 14^8 * 14^2 * 14^1$. Thus,

$14^{27} \bmod 55 = (14^{16} * 14^8 * 14^2 * 14^1) \bmod 55$

$= [(14^{16} \bmod 55)(14^8 \bmod 55)(14^2 \bmod 55)(14^1 \bmod 55)] \pmod{55}$

$= (36 * 16 * 31 * 14) \pmod{55}\ ((36 * 16) \bmod 55 * (31 * 14) \pmod{55}) \bmod 55$

$= (26 * 45) \bmod 55\ 1274 \pmod{55} = 9 \pmod{55} \equiv 9.$

Hence $14^{27} \bmod 55 \equiv 9.$

Thus, the plaintext of the first part of Elain’s message is 9 or 09.

Alex finds the letter corresponds to 09, which is I.

So Alex got Elain’s message HI.

RSA Cryptography

How secure it is?

- The computations it requires of Elain and Alex are elementary.
- But how secure is it against others?
- The security of RSA hinges upon a simple assumption:
  - *Give N, e, and* $y = x^e \bmod N$*, it is computationally intractable to determine x.*

For better understanding, read the following slides.

- Why Does the RSA Cipher Work?
- Brief summary with an example

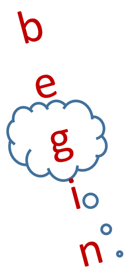
Otherwise, skip those slides and go to:

RSA Cryptography – Formalization

Application of the formalism of

RSA Cryptography

Why Does the RSA Cipher Work?



For the RSA cryptography method, the formula

$$M = C^d \bmod pq. \qquad \ldots\ldots\ (\text{RSA } 0.4.6)$$

is supposed to produce the original plaintext message, M, when the encrypted message is C.

How can we be sure that it always does so?

We require $M < pq$ and we know that

$$C = M^e \bmod pq. \qquad \ldots\ldots\ldots(\text{RSA } 0.4.5)$$

By substitution,

$$M = C^d \bmod pq = (M^e \bmod pq)^d \bmod pq$$

$$= M^{ed} \pmod{pq}$$

And so, it suffices to show $M \equiv M^{ed} \pmod{pq}$.

Why Does the RSA Cipher Work?

For the RSA cryptography method, the formula

$$M = C^d \bmod pq. \qquad \ldots\ldots. \text{ (RSA 0.4.6)}$$

is supposed to produce the original plaintext message, M when the encrypted message is C.

And so, it suffices to show $M \equiv M^{ed} \pmod{pq}$.

Recall that d was chosen to be a positive inverse for e modulo(p-1)(q-1), which exists because gcd(e, (p-1)(q-1)) = 1. In other words,

$$ed \equiv 1 \pmod{(p-1)(q-1)},$$

or equivalently,

$$ed = 1 + k(p-1)(q-1) \quad \text{for some positive integer } k.$$

Therefore,

$$M^{ed} = M^{1 + k(p-1)(q-1)} = M(M^{p-1})^{k(q-1)} = M(M^{q-1})^{k(p-1)}$$

Why Does the RSA Cipher Work?

…

Therefore,

$$M^{ed} = M^{1 + k(p-1)(q-1)} = M(M^{p-1})^{k(q-1)} = M(M^{q-1})^{k(p-1)}$$

If $p \nmid M$, then by Fermat's little theorem, $M^{p-1} \equiv 1 \pmod{p}$, and so

$$M^{ed} = M(M^{p-1})^{k(q-1)} \equiv M(1)^{k(q-1)} \pmod{p} = M \pmod{p}.$$

Likewise, if $q \nmid M$, then by Fermat's little theorem, $M^{q-1} \equiv 1 \pmod{q}$, and so

$$M^{ed} = M(M^{q-1})^{k(p-1)} \equiv M(1)^{k(p-1)} \pmod{q} = M \pmod{q}.$$

Thus, if M is relatively prime to pq,

$$M^{ed} \equiv M \pmod{p} \text{ and } M^{ed} \equiv M \pmod{q}.$$

Why Does the RSA Cipher Work?

…

If M is not relative prime to pq, then either $p \mid M$ or $q \mid M$. Without loss of generality, assume $p \mid M$. It follows that $M^{ed} \equiv 0 \equiv M (\text{mod } p)$. Moreover, because $M < pq$, $q \mid M$, and thus, as above $M^{ed} \equiv 0 \equiv M (\text{mod } q)$. Therefore, in this case also,

$$M^{ed} \equiv M \pmod{p} \quad \text{and} \quad M^{ed} \equiv M \pmod{q}.$$

By Theorem 0.1.4.1,

$$p \mid (M^{ed} - M) \quad \text{and} \quad q \mid (M^{ed} - M),$$

and by definition of divisibility,

$$(M^{ed} - M) = pt \text{ for some integer } t.$$

Why Does the RSA Cipher Work?

…

and by definition of divisibility,

$$(M^{ed} – M) = pt \text{ for some integer } t.$$

By substitution, $q \mid pt$,

and since q and p are distinct prime numbers, Euclid's lemma applies to give

$$q \mid t.$$

Thus, $t = qu$ for some integer u by definition of divisibility.

By substitution,

$$M - M^{ed} = pt = p(qu) = (pq)u,$$

where u is an integer, and so,

$$pq \mid (M - M^{ed})$$

Why Does the RSA Cipher Work?

…

where u is an integer, and so,

$$pq \mid (M - M^{ed})$$

by definition of divisibility. Thus

$$M - M^{ed} \equiv 0 \pmod{pq},$$

by definition of congruence, or, equivalently,

$$M \equiv M^{ed} \pmod{pq}.$$

Because $M < pq$, this last congruence implies that

$$M = M^{ed} \pmod{pq},$$

and thus the RSA cipher gives the correct result. QED

The RSA Cipher Works!

Brief summary with an example

RSA

The RSA scheme is based heavily on number theory. Think of

- *messages from Elain to Alex as numbers modulo N;*
- *messages larger than N can be broken into smaller pieces.*
- *The encryption function will then be a bijection on {0, 1, 2, 3, ..., N - 1}, and the decryption function will be its inverse.*
- *What values of N are appropriate, and what bijection should be used?*

Example 0.72:

Let N = 55 = 5*11.

Choose encryption exponent e = 3, which satisfies the condition

$$\gcd(e, (p-1)(q-1)) = \gcd(3, 40) = 1.$$

The decryption exponent is then $d = 3^{-1} \bmod 40 = 27$.

That is, $27 * 3 \equiv 1 \bmod 40$ if, and only if, $40 \mid (27*3 - 1)$.

Now for any message x mod 55, *the encryption of x is* $y = x^3 \bmod 55$, and *the decryption of y is* $x = y^{27} \bmod 55$.

For example:

if x = 13, then $y = 13^3 \bmod 55 = 52$. That is, $13^3 \equiv 52 \bmod 55$. and $13 = 52^{27} \bmod 55$. (This can be computed as in the following two slides.)

Show $13 = 52^{27} \bmod 55$.

$$
\begin{aligned}
52^{27} \bmod 55 &= (52 \bmod 55)^{27} \bmod 55 \\
&= (-3)^{27} \bmod 55 \\
&= (-3)^{9*3} \bmod 55 \\
&= (81 * 81 * -3)^{9*3} \bmod 55, \text{ where } 81 = (-3)^4 \\
&= (26 * 26 * -3)^3 \bmod 55 \\
&= (52 * 13 * -3)^3 \bmod 55 \\
&= (-3 * 13 * -3)^3 \bmod 55 \\
&= (117)^3 \bmod 55 \\
&= (7)^3 \bmod 55 \\
&= (343) \bmod 55 \\
&= 13
\end{aligned}
$$

Either this way or the way presented in the following slide.

The other way is as follows:

Show $13 = 52^{27} \mod 55$.

$27 = 16 + 8 + 2 + 1 = 2^4 + 2^3 + 2 + 1$

Then, $52^{27} = 52^{16+8+2+1} = 52^{16} * 52^{8} * 52^{2} * 52^{1}$

We can find the residues obtained when 52 is raised to successively higher powers of 2, up to $2^4 = 16$.

$$
\begin{aligned}
52 \bmod 55 &= 52 \\
52^2 \bmod 55 &= 9 \\
52^4 \bmod 55 &= (52^2)^2 \bmod 55 = (52^2 \bmod 55)^2 \bmod 55 \\
&= 9^2 \bmod 55 = 26 \\
52^8 \bmod 55 &= (52^4)^2 \bmod 55 = (52^4 \bmod 55)^2 \bmod 55 \\
&= 26^2 \bmod 55 = 16 \\
52^{16} \bmod 55 &= (52^8)^2 \bmod 55 = (52^8 \bmod 55)^2 \bmod 55 \\
&= 16^2 \bmod 55 = 36
\end{aligned}
$$

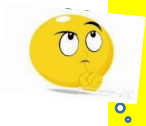
Thus, $52^{27} \bmod 55 = (52^{16} * 52^{8} * 52^{2} * 52^{1}) \bmod 55$

$\equiv [(52^{16} \bmod 55) * (52^{8} \bmod 55) * (52^{2} \bmod 55) * (52^{1} \bmod 55)] \pmod{55}$

$\equiv (36 * 16 * 9 * 52) \pmod{55} \equiv 13$

RSA Cryptography - Formalization

RSA Cryptography - Formalization



The *RSA public-key cryptosystem*- Formalization

Each participant creates their own public key and secret key according to the following steps:

1. Select two very large, non-public prime numbers p and q.
   The number of bits needed to represent p and q might be 1024.

2. Compute

   n = pq, where n is given to the public.

   φ(n) = (p – 1) (q – 1), where φ(n) is secret, and p and q are non-public.

   The formula for φ(n) is owing to the Theorem:

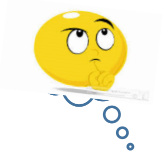
   - The number of elements in $z_n^* = \{\, [1]_n, [2]_n, \ldots\, , \; [n-1]_n \,\}$ is given by Euler's totient function, which is

     $$\varphi(n) = n \prod_{p:p|n} \left(1 - \frac{1}{p} \right),$$

     $$\boxed{n\left(1 - \frac{1}{p} \right)\left(1 - \frac{1}{q} \right) = n \left(\frac{(p-1)(q-1)}{pq}\right) = (p-1)(q-1)}$$

     where the product is over all primes that divide n, including n if n is prime.

The *RSA public-key cryptosystem*- Formalization…



3. Choose a small public prime number e as an encryption component, which is relatively prime to φ(n) such that

   gcd(e, φ(n)) = 1,

   gcd(e, (p-1)(q-1)) = 1.

   Both e and n are public.

4. Using Algorithm sL, compute $[h]_{\varphi(n)}$, the multiplicative inverse of $[e]_{\varphi(n)}$.

   That is,

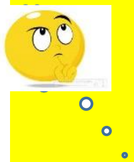
$$[e]_{\varphi(n)}[h]_{\varphi(n)} = [1]_{\varphi(n)}.$$

   e * h $\equiv$ 1 mod φ(n).
   h $\equiv$ e^{-1} mod φ(n).
   e.g., 3 * 27 $\equiv$ 1 mod 40.

   The inverse exists and is unique, according to Corollary s1.2.

   That is, the decryption component h, where h $\equiv$ e^{-1} mod φ(n).

The *RSA public-key cryptosystem*- Formalization…



5. Let pkey = {n, e | n = pq, e is a prime, and both e and n are relatively prime} be the public key. Let skey = {p, q, h | pq = n and $h = e^{-1} \bmod \varphi(n)$} be the secret key.

   For encoding message x mod n:

   - to transform a message x associated with a public key pkey = {n, e},
     - the encryption of x is $y = x^e \bmod n$.

   For decoding message y mod n:

   - to transform a ciphertext y associated with a secret key skey= {p, q, h},
     - the decryption of y is $x = y^h \bmod n$, where $e * h \equiv 1 \bmod \varphi(n)$, and h is the multiplicative inverse of e mod n.

End of the formalization of the *RSA public-key cryptosystem*

RSA Cryptography - Formalization

Takes $O(\log_2 n + \gcd(a, n))$ arithmetic operations

```
Algorithm Modular_Linear_Equation_sL(a, b, n)

    //Find all solutions to a modular linear equation: ax ≡ b (mod n)
    Inputs: positive integers a and b, and integer n.
    Outputs: if the equation [a]_n x = [b]_n is solvable, all solutions to it.
    //compute d = gcd(a, n), i' and j' such that d = ai' + nj',
    //showing that x' is a solution to the equation ax' ≡ d (mod n)
    (d, i', j') = Extended-Euclid(a, n);
    if (d | b){ //compute a solution x0 to the equation ax ≡ b (mod n)
        x0 = (i' *( b/d)) mod n // x0 is a solution of ax ≡ b (mod n).
        for (i = 0; l ≤ d-1; i++){
                output (x0 + i*(n/d))mod n;} //xi = (x0 + i*(n/d))mod n
    else output "no solution"; }
```

RSA Cryptography – Formalization

Takes $O(\log_2 n)$ arithmetic operations

```
Algorithm Extended_Euclid(a, b)
```

//Find d = gcd(a, b) = i*a + j*b.

Inputs: a positive integer a and non-negative b.

Outputs: gcd of a and b, and integers i and j such that d = i*a + j*b.

```
{ if (b == 0) { d = a; i = 1; j= 0;
                return(d, i, j);}
  else { int i’, j’, d’;
         (d’, i’, j’) = extended-Euclid(b, a mod b);
         d = d’;
         i = j’;
         j = i’ + └a/b┘ j’;
         return(d, i, j);}
}
```

Example:

Let a = 14, b= 30 and n = 100

Consider the equation $14x \equiv 30 \pmod{100}$.

Calling Extended-Euclid(14, 100), (d, i', j') = (2, -7, 1).

Since d = 2, b = 30, and 2 |30,

then $x_0 = (i' * (b/d)) \bmod n$

$= (-7 * (30/2)) \bmod 100$

$= -105 \bmod 100$

$= -105 + 2*100$

$= 95$

for i = 0, $(x_0 + i*(n/d)) \bmod n = 95 + 0 = 95$

for i = 1, $(x_0 + i*(n/d)) \bmod n = 95 + 1\,(100/2) = 45$

Therefore 95 and 45 are the solution for $14x \equiv 30 \pmod{100}$.

Check: $14*95 \equiv 30 \pmod{100}$. That is, $14*95 \pmod{100} = 1330 \pmod{100} = 30$

$14*45 \equiv 30 \pmod{100}$. That is, $14*45 \pmod{100} = 530 \pmod{100} = 30$

RSA Cryptography – Formalization (another way of writing)

```
Algorithm Modular_Linear_Equation_sL(n, m, k)
```

//Find all solutions to a modular linear equation.

Inputs: positive integers m and n, and integer k.

Outputs: if the equation $[m]_n\, x = [k]_n$ is solvable, all solutions to it.

```
index l ;
integer i, j, d;
extended-Euclid(n, m, d, i, j);
if (d | k)
    for (l = 0; l ≤ d-1; l++)
```

output $[\frac{jk}{d} + \frac{\mathtt{l}n}{d}]_n$; //equivalent classes modulo n

// $[\frac{jk}{d} + \frac{\mathtt{l}n}{d}]_n = [\frac{jk}{d}]_n + [\frac{\mathtt{l}n}{d}]_n$

RSA Cryptography – Formalization (another way of writing)

```
Algorithm Extended_Euclid(int n, int m, int& gcd, int& i, int& j)
    //Find gcd(n, m) = i*n + j*m.
    Inputs: a positive integer n and a nonnegative m.
    Outputs: gcd of n and m, and integers i and j such that gcd = i*n + j*m.
    { if (m == 0) { gcd = n; i = 1; j= 0;}
      else { int i', j', gcd';
             Extended-Euclid (m, n mod m, gcd', i', j');
             gcd = gcd';
             i = j';
             j = i' + ⌊n/m⌋ j';}
    }
```

Call	n	m	gcd	i	j
	↓	↓			
0	42	30	6	-2	3
1	30	12	6	1	-2
2	12	6	6	0	1
3	6	0	6	1	0
	→	→	↑	↑	↑

The time complexity is the same as recursive Euclid(n, m). The worst case is Worse(s, t) ∈ O(st) where s = floor(log n) and t = floor(log m). The number of recursive calls Calls(s, t) is θ(t).
The time complexity of Algorithm Modular_Linear_Equation_sL() is the time complexity of Algorithm Extended_Euclid(), plus the time complexity is worst-case exponential in terms of the input size. But this is required from the problem.

The table illustrates the flow of the Algorithm when the top-level call is Extended_Euclid(42, 30, gcd, i, j); The values returned at the top level are gcd = 6, i = -2, and j = 3. That means, gcd(n, m) = i*n + j*m.

The top-level call is labeled 0, the three recursive calls are labeled 1 – 3. The arrows show the order in which the values are determined.

++end

Call	n	m	gcd	i	j
	↓	↓			
0	42	30	6	-2	3
1	30	12	6	1	-2
2	12	6	6	0	1
3	6	0	6	1	0
	→	→	↑	↑	↑

gcd(42, 30) $\underline{42} = 1*\underline{30} + 12$ $12 = 1*\underline{42} - 1*\underline{30}$

gcd(30, 12) $\underline{30} = 2*\underline{12} + 6$ $6 = 1*\underline{30} - 2*\underline{12}$

gcd(12, 6) $\underline{12} = 2*\ \underline{6} + 0$ $0 = 1*\underline{12} - 2*\ \underline{6}$

gcd(6, 0) = 6 $\underline{6} = 0*\ \underline{0} + 6$ $6 = 1*\ \underline{6} - 0*\ \underline{0}$

$$\begin{aligned}
6 &= 1*\ \underline{6} - 0*\ \underline{0} \\
&= 1*\ \underline{6} - 0*\ (1*\underline{12} - 2*\ \underline{6}) = 0*\underline{12} + 1*\ \underline{6} \\
&= 0*\underline{12} + 1*\ (1*\underline{30} - 2*\underline{12}) = 1*\underline{30} - 2*\underline{12} \\
&= 1*\underline{30} - 2*\ (1*\underline{42} - 1*\underline{30}) = -2*\underline{42} + 3*\underline{30}
\end{aligned}$$

$6 = -2*\underline{42} + 3*\underline{30} = 6*(-2*7 + 3*5)$. Note that
$\gcd(x, y) = d = \min\{ix + jy \mid i, j \in Z \text{ and } ix + jy > 0\}$;
$(i * 7 + j * 5)6 > 0$ has a minimum value if and only
if $i * 7 + j * 5 = 1$, where $i, j \in Z$.
This implies that $i = -2$ and $j = 3$ can be a choice.

++end

Application of the formalism of RSA Cryptography

RSA Cryptography - Application

Example: Let e and g be interchangeably used.

Encipher, encode, encrypt

- convert (a message or piece of text) into a coded form.

Decipher, decode, decrypt

- convert (a text written in code, or a coded signal) into normal language.

Let the public key be denoted as pkey = {n, g} and secret key denoted as skey = {p, q, h}.

Consider an RSA cryptosystem using p = 7, q = 17, and g = 5.

I. What is the encode form for 13 mod 119? That is, encipher the message $[13]_{119}$.

II. What is the encode form for 39 mod 119? That is, encipher the message $[39]_{119}$.

Solution:

Given the public key pkey = {n = 119, g = 5} where the secret key p = 7 and q = 17 such that n = pq = 7*17 = 119; and g is relatively prime to (p-1)(q-1).

Encode(x) = y = $x^g \bmod n$.

I. Let x = 13. We encode the message 13 mod 119, which is as follows:

$$
\begin{aligned}
\text{Encode}(13) = y = \; & 13^5 \bmod 119 \\
& = (13^2 \times 13^2 \times 13) \bmod 119 \\
& = ((13^2 \bmod 119) \times (13^2 \bmod 119) \times 13) \bmod 119 \\
& = ((50 \times 50 \times 13) \bmod 119) \bmod 119 \\
& = ((50 \times 5 \times 10 \times 13) \bmod 119) \bmod 119 \\
& = ((250 \bmod 119) \times (130 \bmod 119)) \bmod 119 \\
& = (12 \times 11) \bmod 119 \\
& = (132 \bmod 119) \bmod 119 \\
& = 13 \bmod 119 = \underline{13}
\end{aligned}
$$

That is, the encryption of give message 13 is 13.

To decode this message 13 mod 119, it requires that

a) we know the public key pkey = {n, g} = {119, 5}, and
b) we find the private key skey {p, q, h}, where h can be calculated as follows:

Step 1: For public, n = 119 and g = 5, and the encoded form y = 13 are given.

In public, although it is known that n = p*q, and n = 119, it should not easily be derived that the non-public secret keys p = 7 and q = 17 from n.

For this case, n = 119 = 7*17 = p*q.

Step 2: φ(n) = (p – 1) (q – 1) = 6 * 16 = 96.

The reason is:

- The formula for φ(n) is owing to the Theorem:
  - The number of elements in

    $z_n^* = \{ [1]_n, [2]_n, \ldots, [n-1]_n \}$

    is given by Euler's totient function, which is

    $$\varphi(n) = n \prod_{p:p|n} \left(1 - \frac{1}{p} \right),$$

    where the product is over all primes that divide n, including n if n is prime.
  - That is, φ(n) = (p – 1) (q – 1) = $\varphi(n) = n \prod_{p:p|n} \left(1 - \frac{1}{p} \right)$.

Example: Let p = 7 and q =17 be primes such that 119 = 7*17.

φ(119) = (p – 1) (q – 1) = 119 * (1- $\frac{1}{7}$) (1- $\frac{1}{17}$) = 119* $\frac{(7-1)(17-1)}{7*17}$

= (7-1) (17-1) = 96.

Since g = 5, g is relatively prime to φ(n) = φ(119) = 96.

Step 3: Compute the multiplicative inverse $[h]_{\varphi(n)}$ of $[g]_{\varphi(n)}$.

That is, $[g]_{\varphi(n)}[h]_{\varphi(n)} = [1]_{\varphi(n)}$.

We know that $[1]_{\varphi(119)} = [1]_{96}$

$= \{ \ldots, -287, -191, -95, 1, 97, 193, 289, \ldots \}$.

Compute $g * h \equiv 1 \bmod \varphi(n)$.

$h * 5 \equiv 1 \bmod \varphi(119)$

$h * 5 \equiv 1 \bmod 96$

$h \equiv \frac{1}{5} \bmod 96$

$h \equiv 5^{-1} \bmod 96$.

To compute multiplicative inverse $[h]_{\varphi(n)}$ of $[g]_{\varphi(n)}$, we use Extended Euclid Algorithm.

Euclid(φ(n), g) = Euclid(96, 5) which is computed as follows:

Write gcd(m, n) in terms of $\underline{m} = q*\underline{n} + r$, where q is quotient and the remainder $0 \leq r < n$.

Iteration 1:	$\underline{96} = 19 * \underline{5} + 1$	gcd(96, 5) = gcd(5, 96 mod 5)
Iteration 2:	$\underline{5} = 5 * \underline{1} + 0$	gcd(5, 1) = gcd (1, 5 mod 1)
Iteration 3:	$\underline{1} = 1 * \underline{0} + 1$	gcd(1, 0) = 1
		Thus gcd(96, 5) = 1

Rewrite the last iteration $\underline{1} = 1 * \underline{0} + 1$ in terms of linear combination

$$1 = 1 * \underline{1} - 1 * \underline{0}.$$

Rewrite the iteration 2, $\underline{5} = 5 * \underline{1} + 0$ in terms of linear combination

$$0 = 1 * \underline{5} - 5 * \underline{1}.$$

Substituting $1 * \underline{5} - 5 * \underline{1}$ for the 0 in $1 = 1 * \underline{1} - 1 * \underline{0}$, we have

$$1 = 1 * \underline{1} - 1 * \underline{0}$$

$$1 = 1 * \underline{1} - 1 * (1 * \underline{5} - 5 * \underline{1})$$

$$1 = 1 * \underline{1} - 1 * \underline{5} + 5 * \underline{1}$$

$$1 = - 1 * \underline{5} + 6 * \underline{1}$$

Rewrite the iteration 1, $\underline{96} = 19 * \underline{5} + 1$ in terms of linear combination

$$1 = 1 * \underline{96} - 19 * \underline{5}.$$

Substituting $1 * \underline{96} - 19 * \underline{5}$ for the $\underline{5}$ in $1 = -1 * \underline{5} + 6 * \underline{1}$, it can be rewritten in terms of combination

$$1 = -1 * \underline{5} + 6 * (1 * \underline{96} - 19 * \underline{5})$$

$$1 = -1 * \underline{5} + 6 * \underline{96} - 114 * \underline{5}$$

$$1 = 6 * \underline{96} - 115 * \underline{5}.$$

The above equation $1 = 6 * \underline{96} - 115 * \underline{5}$ can be written as

$$1 \bmod 96 \equiv (6 * \underline{96} - 115 * \underline{5}) \bmod 96$$

$$1 \bmod 96 \equiv (6 * \underline{96} \bmod 96 - 115 * \underline{5} \bmod 96) \bmod 96$$

$$1 \bmod 96 \equiv (0 - 115 * \underline{5} \bmod 96) \bmod 96$$

$$1 \bmod 96 \equiv (-575 \bmod 96) \bmod 96$$

$$1 \bmod 96 \equiv ((-576 + 1) \bmod 96) \bmod 96$$

$$1 \bmod 96 \equiv (-576 \bmod 96 + 1 \bmod 96) \bmod 96$$

$$1 \bmod 96 \equiv (0 + 1 \bmod 96) \bmod 96$$

$$1 \bmod 96 \equiv (1 \bmod 96) \bmod 96$$

$$1 \bmod 96 \equiv 1 \bmod 96$$

Since $1 \bmod 96 \equiv (0 - 115 * \underline{5} \bmod 96) \bmod 96$

$1 \bmod 96 \equiv -115 * \underline{5} \bmod 96$

$$-115 \equiv \frac{1}{5} \bmod 96$$

$$-115 \equiv 5^{-1} \bmod 96.$$

Therefore, -115 is the multiplicative inverse of 5.

And -115 is one of the candidate of h, because $1 \bmod 96 = -115 * \underline{5} \bmod 96$ $= ((-115 \bmod 96) * (5 \bmod 96)) \bmod 96$.

This can be expressed in terms of $[h]_{96} * [5]_{96} = [1]_{96}$

Since h is the smallest positive number, we will find the equivalence class modulo of -115 to obtain the smallest positive number from its calls modulo.

We know that

$$[h]_{\varphi(119)} = [h]_{96} = \{ \ldots, -211, -115, -19, 77, 173, 269, 365, \ldots\}.$$

That means, $[77]_{96} * [5]_{96} = [1]_{96}$ or, should we way,

$[77]_{96}$ is the multiplicative inverse of $[5]_{96}$.

This means, $77 = 5^{-1} \bmod 96$.

This means, $1 \bmod 96 \equiv 77 * 5 \bmod 96 \equiv 77 * 5$

Therefore, the secret key skey = {p, q, h} = {7, 17, 77}.

Conclusion:
For a message x mod 119, the encryption of x is $y \equiv x^5 \bmod 119$.
The decryption of y is $x \equiv y^{77} \bmod 119$.

Example:

Given $x = 13$, $p = 7$, $q = 17$ and $g = 5$, the encryption of x is $y \equiv x^g \bmod n$.

$y \equiv 13^5 \bmod 119 \equiv 371293 \bmod 119 \equiv \underline{13} \bmod 119 = \underline{13}$.

The decryption of y is x, which is $x \equiv y^h \bmod 119$

$$
\begin{aligned}
x \equiv \underline{13}^{77} \bmod 119 &\equiv (13^5)^{15} * 13^2 \bmod 119 \\
&\equiv (13)^{15} * 13^2 \bmod 119 \\
&\equiv (13^5)^3 * 13^2 \bmod 119 \\
&\equiv (13)^3 * 13^2 \bmod 119 \\
&\equiv 13^5 \bmod 119 \\
&\equiv 13 \bmod 119 \\
&\equiv 13
\end{aligned}
$$

Note that encode (13 mod 119) = $\underline{13}$. Decode($\underline{13}$) = 13. We are luckily get the identical 13. But it should not always in this case. x and y can be different in value.

End of

Application of the formalism of RSA Cryptography