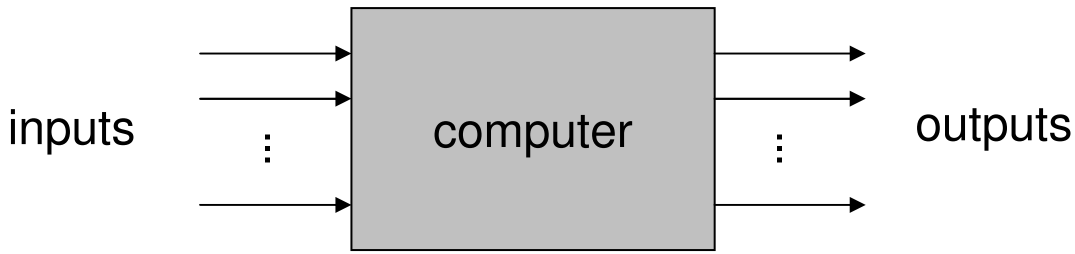
Theory of Computation

National Chiao Tung University
Chun-Jen Tsai
6/13/2012

Fundamental CS Questions

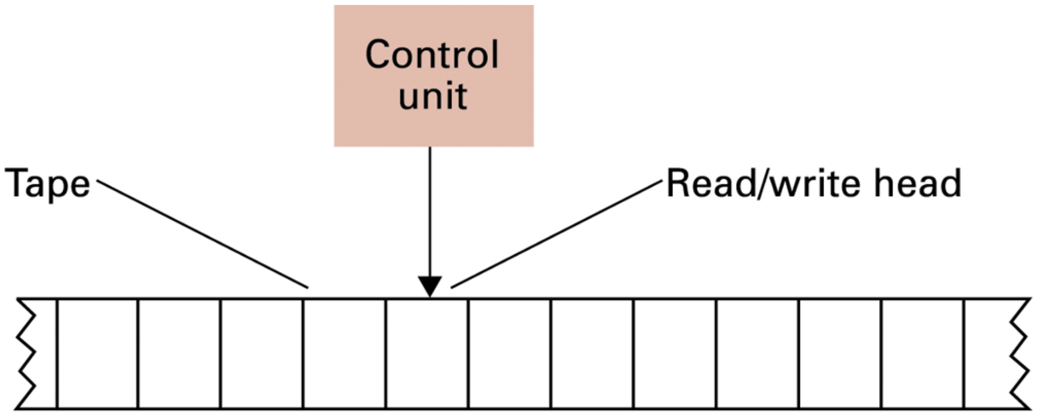
- What "problems" are solvable? By machines with what capability?
  - All computers only compute functions. That is, mapping input values to output values



  - A function is computable if the mapping is unique and can be calculated by the computer

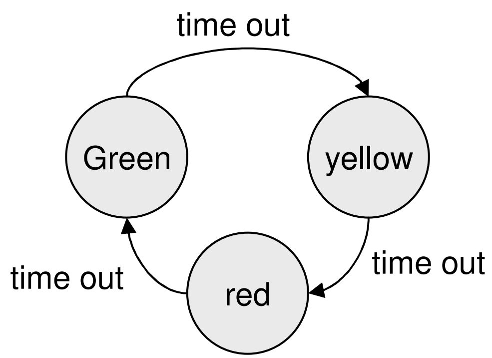
Turing Machines

- Turing machines are proposed by A. Turing in 1936:
  - Study the minimal computers for function computation
- A Turing machine is composed of
  - A control unit that can read and write symbols on a tape
  - A tape with infinite length and records symbols sequentially
  - A set of symbols that the machine can read/write/store – this is called the alphabet of the machine



Turing Machine Operations

- A Turing machine is a state machine
  - State machine example: traffic lights



- Inputs at each step
  - State
  - Value at current tape position
- Actions at each step
  - Write a value at current tape position
  - Move read/write head
  - Change state

Example of Computations

- A Turing machine can add one to a binary value as follows:

Current state	Current cell content	Value to write	Direction to move	New state to enter
START	*	*	Left	ADD
ADD	0	1	Right	RETURN
ADD	1	0	Left	CARRY
ADD	*	*	Right	HALT
CARRY	0	1	Right	RETURN
CARRY	1	0	Left	CARRY
CARRY	*	1	Left	OVERFLOW
OVERFLOW	*	*	Right	RETURN
RETURN	0	0	Right	RETURN
RETURN	1	1	Right	RETURN
RETURN	*	*	No move	HALT

- **Church-Turing Thesis**: A Turing machine can compute any computable functions
  - This a generally accepted conjecture, not a proven theory

Universal Programming Language

- **An universal programming language** is a language that can express a program to compute any computable functions
  - Most popular programming languages are universal programming languages
- Most programming languages are feature-rich. But, what is the minimal elements a languages needs to be "universal?"

The Bare Bones Language

- Bare Bones is a simple, yet universal language
- Statements
  - **`clear`** ***`name`*****`;`**
  - **`incr`** ***`name`*****`;`**
  - **`decr`** ***`name`*****`;`**
  - **`while`** ***`name`*** **`not 0 do; … end;`**
- Researchers have shown that the Bare Bones can be used to compute all Turing-computable functions

Examples of Bare Bones Programs

- Computing $X \times Y$:

```
clear Z;
while X not 0 do;
   clear W;
   while Y not 0 do;
      incr Z;
      incr W;
      decr Y;
   end;
   while W not 0 do;
      incr Y;
      decr W;
   end;
   decr X;
end;
```

Copying “Today” to “Tomorrow”

```
clear Aux;
clear Tomorrow;
while Today not 0 do;
   incr Aux;
   decr Today;
end;
while Aux not 0 do;
   incr Today;
   incr Tomorrow;
   decr Aux;
end;
```

Non-computable Functions

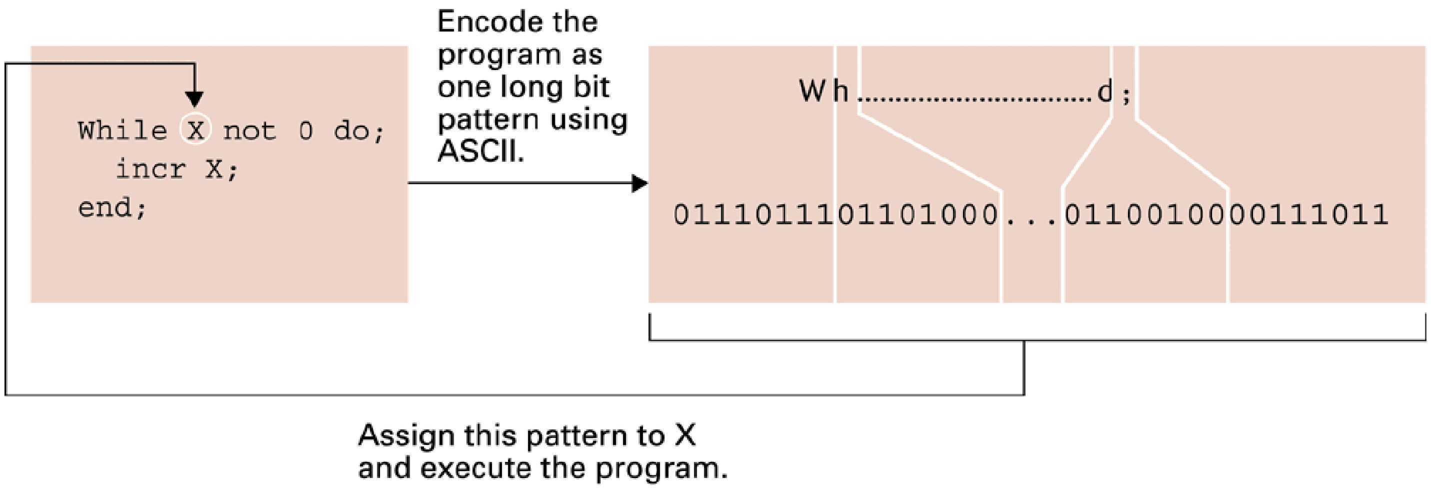
- A classical non-computable function is the “Halting Problem:”

  Given the encoded version of any program, return 1 if the program will eventually halt, or 0 if the program will run forever

- The solution to the halting problem is important, but there is no way to compute such a function

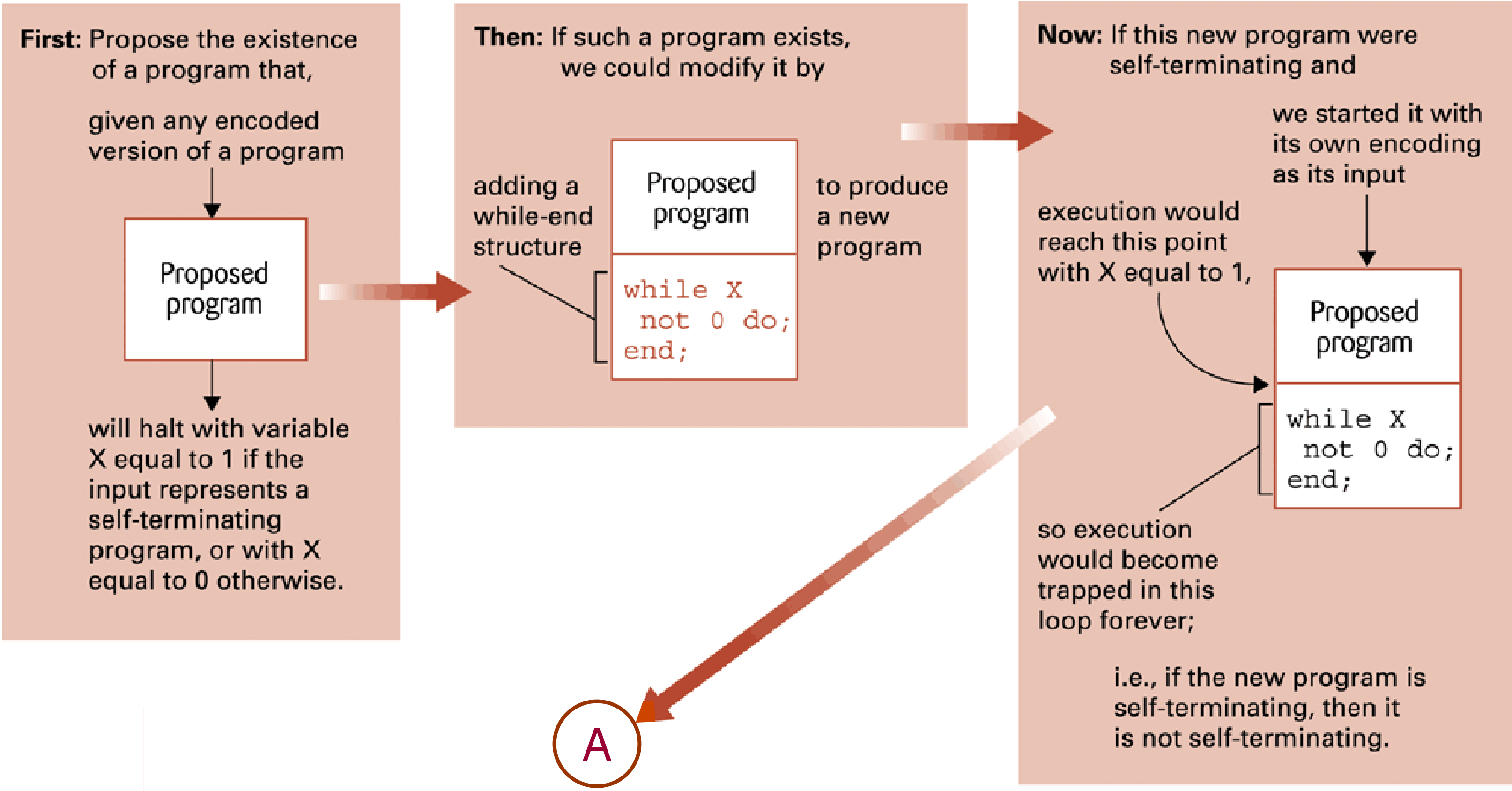
Self-Terminating

- Let's consider a simple Bare Bones program:

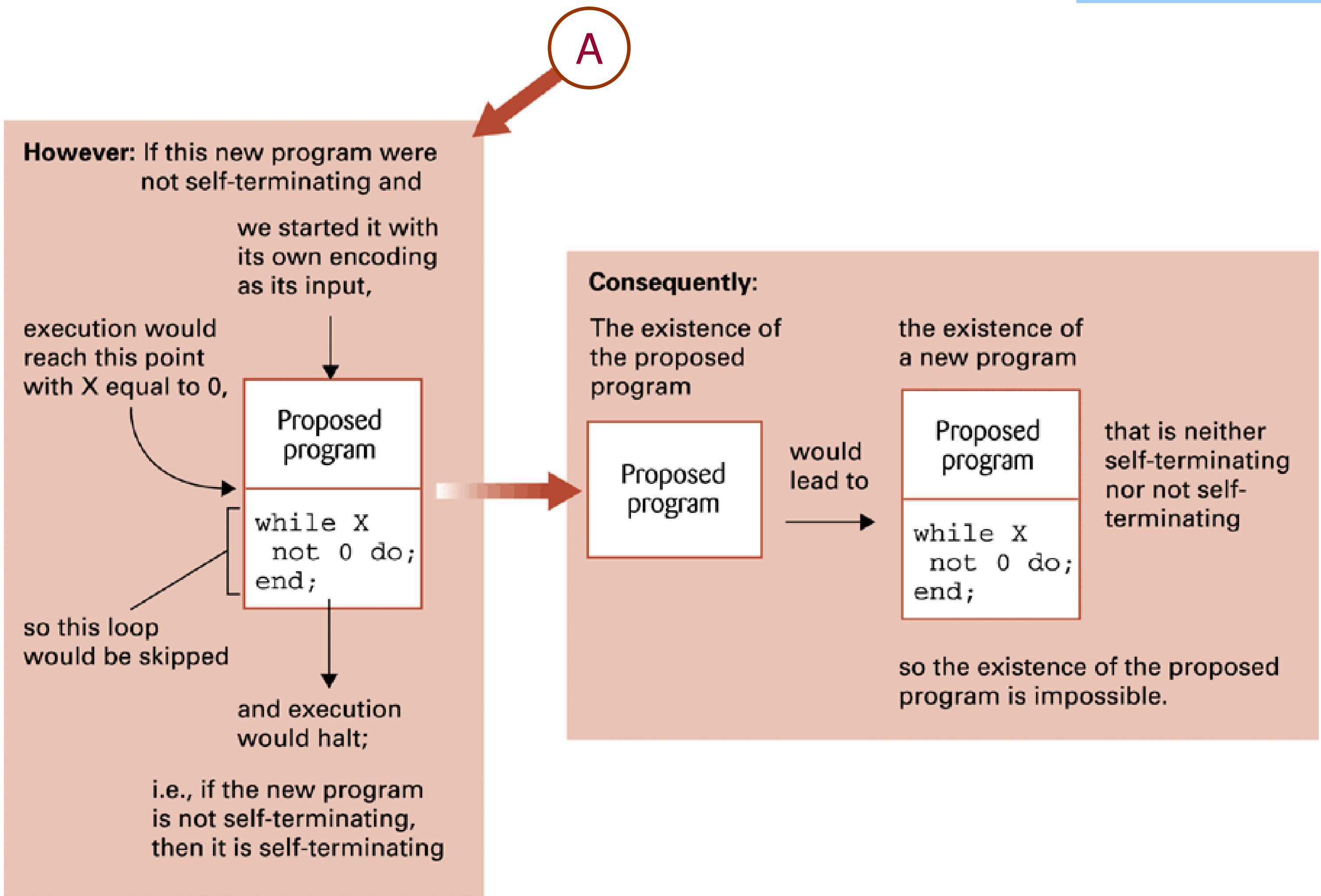


- If the program terminates when the initial value is set to the encoded version of itself, then it is called a self-terminating program

Insolvability of Halting Problem (1/2)



Insolvability of Halting Problem (2/2)



Complexity of Problems

- Time-complexity of a problem is the time it takes to find the solution of a problem
  - From machine's point of view, this is equivalent to the number of machine instructions it must perform when executing ***a best algorithm*** that solves the problem
- Recall that the notation $\Theta(f(n))$ can be used to denote the time-complexity of a problem
  - $\Theta(n^2)$ means that the complexity increases as fast as a 2nd-order polynomial when the input size n increases linearly

Class P Functions

- Class P functions are all problems in any class $\Theta(f(n))$, where $f(n)$ is a polynomial
- Intractable functions are all problems too complex to be solved practically
  - Most computer scientists consider all problems not in class P to be intractable

Class NP Functions

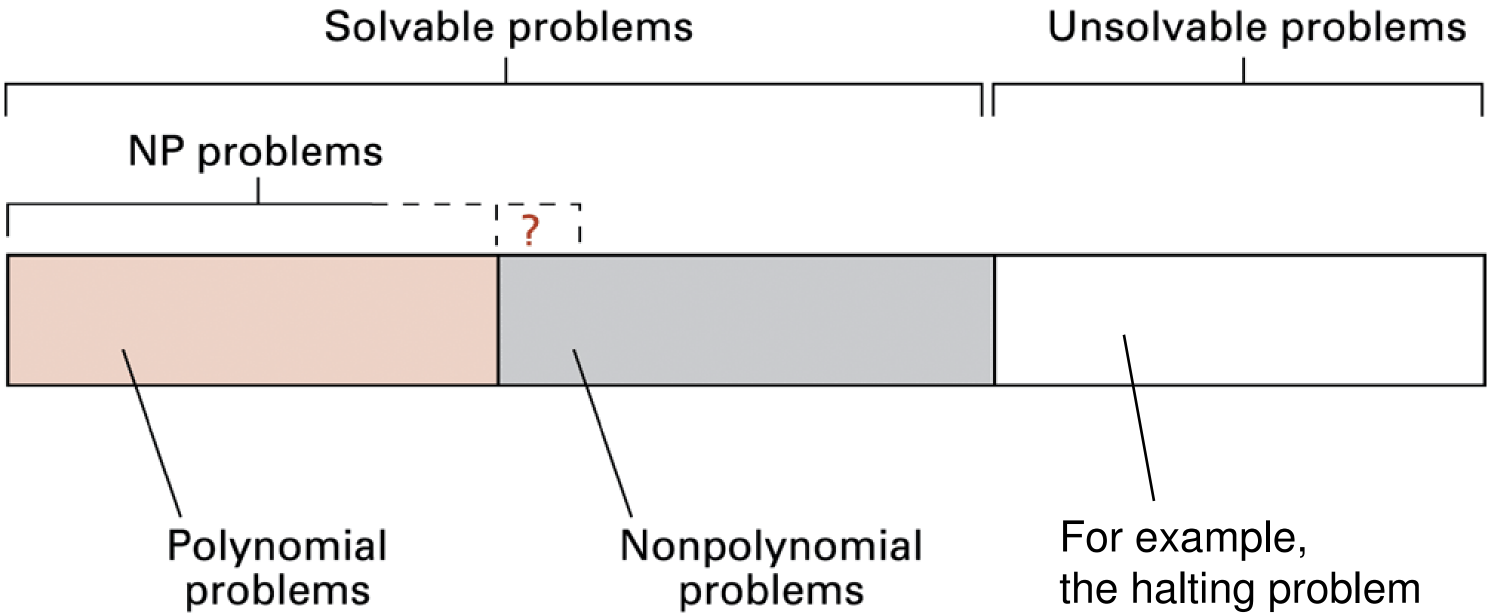
- Class NP functions are all problems that can be solved by a nondeterministic algorithm in class P
  - A nondeterministic algorithm is an “algorithm” whose steps may not be uniquely and completely determined by the process state
  - A nondeterministic algorithmic step can be executed by a hypothetical intelligent machine; for example:

    “Go to the next intersection and turn either left or right to get to a drug store.”

- Whether the class NP is bigger than class P is currently unknown

Summary on Complexity

- A classification of computing problems are as follows:

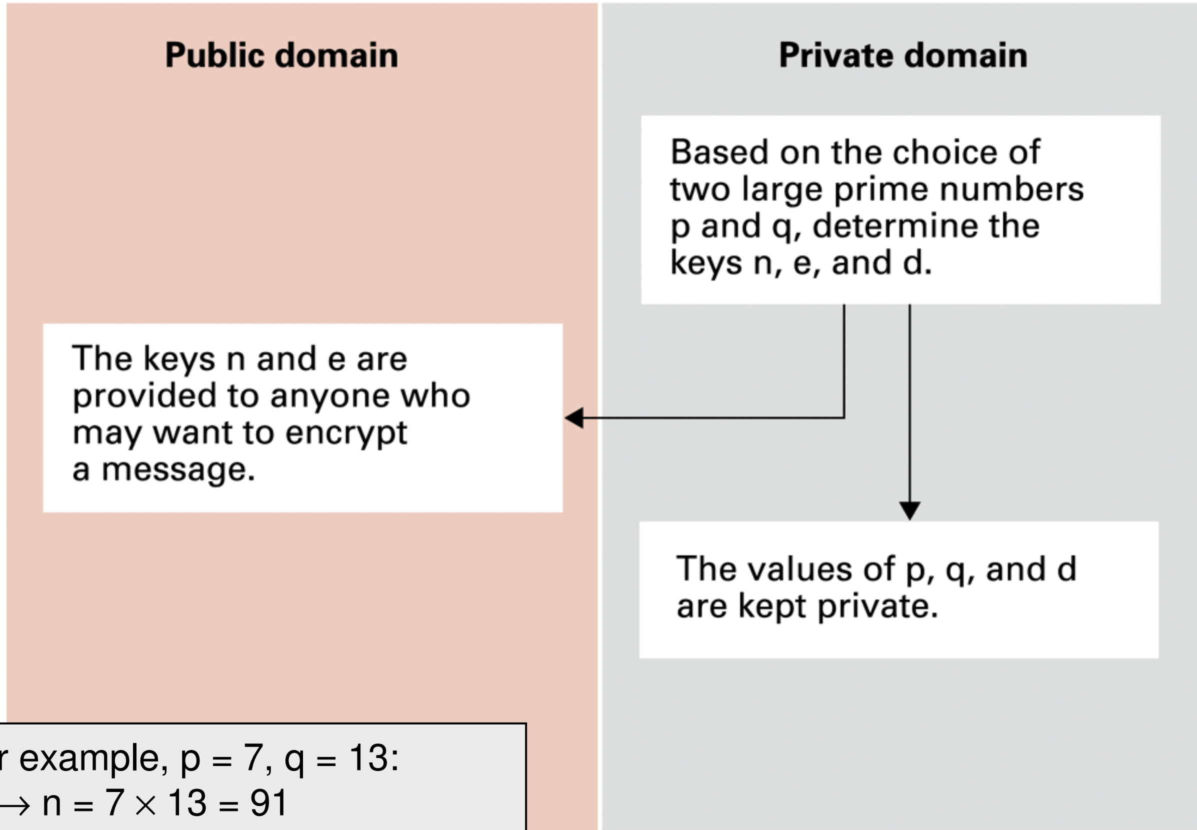


Complexity and Cryptography

- In old days, encryption of a message requires the encryption key to be kept secret $\rightarrow$ not secure since both the message sender and receiver need the key
- **RSA** is a popular public key cryptographic algorithm that relies on the (presumed) intractability of the problem of factoring large numbers
  - Public key is used for encryption, and can be given to anyone
  - Private key is used for decryption, and is only available to the receiver

Public Key Cryptography

- RSA works as follows:



For example, $p = 7$, $q = 13$:

$\rightarrow n = 7 \times 13 = 91$

$\rightarrow 5 \times 29 = 2(7-1)(13-1) + 1$

Encrypting the Message 10111

- Encrypting keys: n = 91 and e = 5
- $10111_{two} = 23_{ten}$
- $23^e = 23^5 = 6{,}436{,}343$
- $6{,}436{,}343 \div 91$ has a remainder of 4
- $4_{ten} = 100_{two}$
- Therefore, encrypted version of 10111 is 100.

Decrypting the Message 100

- Decrypting keys: d = 29, n = 91
- $100_{two} = 4_{ten}$
- $4^d = 4^{29} = 288{,}230{,}376{,}151{,}711{,}744$
- 288,230,376,151,711,744 ÷ 91 has a remainder of 23
- $23_{ten} = 10111_{two}$
- Therefore, decrypted version of 100 is 10111.