

The Elements Program at UT

Preparing for Life in the Digital World

Dr. Bill Young
Department of Computer Sciences
University of Texas at Austin

Orientation: June 11, 2009

Thought Experiment

Imagine you have a friend who lives in some foreign country where the secret police spy on everyone and everything, and you want to send a valuable object to this friend. You have a strongbox big enough to hold the object. The box has a lock ring that can accommodate several locks. But your friend does not have the key to any lock that you have.

You can't send a key in the mail because the secret police will intercept and copy it. You can't leave the box unlocked, because the object is too valuable. Ideally, you'd lock the object in the box and mail it to your friend, so that he can open it, but the secret police can't. *How could you do it?*

One Possible Answer

You might take the following sequence of steps:

- 1 You put your treasure into the box, attach your lock to the clasp, and mail the box to your friend.
- 2 He adds his own lock, for which he has the key, and mails the box back to you.
- 3 You remove your lock and mail the box back to him. He now removes his lock and opens the box.

What's This Got to do with Computing?

The procedure just described could be regarded as a *protocol* – a structured dialog intended to accomplish some communication-related goal.

What goal: To send some content secretly in the context of a hostile or untrustworthy environment, when the two parties don't already share a secret/key

In fact, you could implement the “same” protocol to send a message confidentially across the Internet. Here,

- the *valuable thing* is the contents of a secret message;
- the *locks* are applications of a cryptographic algorithm (cipher) with appropriate *cryptographic keys*.

But for this to work in the computing world there's a particular feature that the ciphers have to satisfy. *Can you see what it is?*

What is the Property?

Imagine that instead of putting on another lock, your friend puts your lockbox inside another locked box. Our protocol wouldn't work because you couldn't reach inside his box to take off your lock in step 3.

You have to be able to “reach inside” his encryption to undo yours. One way this would be true is if the ciphers *commute*.

$$\text{Cipher}_1(k_1, \text{Cipher}_2(k_2, \text{msg})) = \text{Cipher}_2(k_2, \text{Cipher}_1(k_1, \text{msg}))$$

Most encryption algorithms don't have this property. But one that does is: exclusive or (XOR) your message with a randomly generated string (key) of the same length.

What's Exclusive Or?

To XOR a message means to apply the following function on a bitwise basis:

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Example:

Message:	11101001010111
Key:	10110101110010
	<hr/>
Encrypted message:	01011100100101

To decrypt the message, you just XOR again with the same Key.

Properties of XOR

To encrypt a message by XOR'ing it with a randomly generated string of the same length is called using a *one-time pad*. It is a *theoretically unbreakable encryption algorithm*.

Here are some of the algebraic properties of XOR:

$$x \oplus 0 = x$$

$$x \oplus x = 0$$

$$x \oplus y = y \oplus x$$

$$(x \oplus y) \oplus z = x \oplus (y \oplus z)$$

One implication of these rules is that if you have two x 's anywhere within a nest of XORs, they cancel each other out.

So Here's the Protocol

Let K_a be a random string generated by A, and K_b be a random string generated by B.

- 1 $A \rightarrow B : M \oplus K_a$
- 2 $B \rightarrow A : (M \oplus K_a) \oplus K_b$
- 3 $A \rightarrow B : ((M \oplus K_a) \oplus K_b) \oplus K_a$

In step 3, the two applications of K_a “cancel out,” leaving $(M \oplus K_b)$, which B can easily decrypt with his own key K_b .

Whoops!

Even though the one-time pad is a theoretically unbreakable cipher, there's a reason it's called "one-time." Our protocol is fundamentally flawed. *Can you see why?*

Whoops!

Even though the one-time pad is a theoretically unbreakable cipher, there's a reason it's called "one-time." Our protocol is fundamentally flawed. *Can you see why?*

An eavesdropper who records the three messages can XOR combinations of them to extract any of M , K_a , and K_b . Try it for yourself.

This puzzler is a problem I sometimes pose to my students in computer security. To solve it you need:

- some technical knowledge, and
- the ability to think creatively.

That's true of much of computer science. The concepts are often simple, but incredibly powerful and useful in a huge range of disciplines.

Computer science is the science of information and computation and their implementation and application in computer systems.

Computing and computer technology are part of just about everything that touches our lives. No matter what field you're entering, you simply can't be a fully functioning and educated citizen without some fluency in computing.

Computing jobs are among the highest paid and have the highest job satisfaction. The Bureau of Labor Statistics says computing has the “greatest potential for new jobs through 2014.”

Goals of the Elements Program

The Elements Program is for non-CS majors interested in learning about the enabling technology of our age. It aims to:

- Teach thinking and problem solving skills using computing technology.
- Provide an understanding of the technologies that are ubiquitous in the modern world and workplace.
- Provide technical skills to complement your major, *whatever your major may be.*

The Elements Certificate

For UTCS certification, students must complete 12 semester hours of Elements courses, including CS 303E or equivalent, and at least 6 hours of upper division courses. If you take 18 hours, UT will record your certification on your official transcript.

Two entry points to the Elements Program:

CS303E: Introduction to Programming Assumes the student has little or no programming background. Introduces Java or a scripting language.

CS313E: Elements of Software Design Typical starting point for a student with a year or more of programming in high school. Introduction to Java programming.

These provide a basic knowledge of programming and skills needed in subsequent courses.

CS301K: Foundations of Logical Thought

A basic introduction to logical thinking. What does it mean to solve a problem systematically? What are the tools that can be employed in thinking critically?

CS302: Computer Fluency

Basic familiarity with what computers are, how they work, and how they can be used to solve everyday problems through the study of algorithms – precise instructions for solving a problem. Also touches on the impact of computers on people and society.

Upper Division Courses

The upper division courses allow the student to explore topics relevant to his/her major. Programming is required for some classes.

CS320N: Topics in Computer Science

- **Computers From the Ground Up**: build and program your own computer
- **Contemporary Issues in Computing**: address topics of the interaction of computing and society
- **Great Ideas in Computing**: what are the intellectual innovations that have driven the great success of computing
- **Visual Programming**: build complex “worlds” using a powerful graphical interface

CS320N may be repeated when the topic varies.

CS324E: Elements of Graphics and Digital Media

Understand basic 2-D and 3-D computer graphics systems and how to manipulate digital media. Study animation, game design, graphical user interfaces, and visual information presentation. (requires CS313E)

CS326E: Elements of Networking

Principles and basic concepts of the Internet and World Wide Web, including wired and wireless networks, security, privacy, and file sharing.

CS327E: Elements of Databases

Introduction to SQL and fundamentals of database technology to facilitate information searches. (requires CS313E)

CS329E: Topics in Elements of Computing

- **Elements of Computing in Society:** investigate the role of computing in society. Topics may include privacy, free speech on the Internet, newspapers vs. bloggers, defending against hackers, and copywrite laws
- **Elements of Navigating Cyberspace:** peek beneath the surface of the Internet to understand how cyberspace really works
- **Elements of Web Programming:** learn how to create your own Web presence

CS329E may be repeated if the topic varies.

Advantages of Elements Certificate

Becoming certified may help you to:

- Understand your world better
- Gain a competitive advantage in the workplace
- Become a more productive, more informed citizen
- Impress your friends and family

What Past Students Have Said

Employers like candidates with some technical background.

Helped me get an internship and excel in this job.

The Database class helped me during an internship, while the Networking class has enabled me to understand the Internet a lot better.