SPCS Cryptography Class Lecture 12

July 7, 2015

Secure Multi-party Computing

Example

Suppose Alice, Bob and Carol want to figure out their average age, without ever revealing his/her own age. How can they do that?

Solution

Secret sharing! Suppose their ages are A, B, C.

- *Alice randomly splits $A = A_1 + A_2 + A_3$. Similarly Bob splits $B = B_1 + B_2 + B_3$, Carol splits $C = C_1 + C_2 + C_3$.*
- *Alice sends Bob A_2, Carol A_3. Bob sends Alice B_1, Carol C_3, Carol sends Alice C_1, Bob C_2.*
- *Alice announces $A_1 + B_1 + C_1$, Bob announces $A_2 + B_2 + C_2$, Carol announces $A_3 + B_3 + C_3$.*
- *Average them.*

Secure Multi-party Computing

Example

Suppose the class wants to hold a yes-no vote, without ever revealing the individual vote. How can they do that?

Solution

This reduces to the sum problem as before, by giving a Yes value 1 and a No value 0.

Secure Multi-party Computing

Example

Suppose Bob wants to ask Alice out and vice versa. They want to find out if there is mutual interest, without running the risk of embarassment of one asking the other out but being turned down. Carol agrees to assist them with the process. How can they do that?

Solution

If a Yes is a 1, a No is a 0 - they only need to find out the product of Alice's and Bob's number. If Alice says 0, she never knows whether Bob started with 1 or 0.

Secure Multi-party Computing

Solution

Here is one way to approach the problem.

- *Alice splits her answer* $A = A_1 + A_2 + A_3$. *Similar Bob splits his answer* $B = B_1 + B_2 + B_3$.
- *Alice gives Bob* A_1, A_3, *and Carol* A_1, A_2. *Bob gives Alice* B_2, B_3, *and Carol* B_1, B_2.
- *Alice computes* $A_2B_2 + A_2B_3 + A_3B_2$. *Bob computes* $A_3B_3 + A_1B_3 + A_3B_1$. *Carol computes* $A_1B_1 + A_1B_2 + A_2B_1$. *They only need to add up the answer now.*
- *Use the previous problem to add the answer up. Alice and Bob now know the answer.*

Secure Multi-party Computing

Example (Andrew Yao's Millionaires Problem)

Suppose Alice and Bob want to know who is richer, without ever revealing his/her own wealth. How can they do that? For simplicity, assume their wealth are integers between 0 and 10^7.

Secure Multi-party Computing

Solution

Yao's solution:

- *Bob puts out one locked box-with-a-slit for each of the possible wealth.*
- *He throws away all the keys for the boxes except for the one corresponding to his wealth.*
- *Bob turns around.*
- *For each box more than Alice's wealth, she puts in a note saying that "You are richer than me".*
- *For the remaining boxes, she puts in a note saying "You are no richer than me."*
- *Bob turns back, and opens the box corresponding to his wealth. There he finds the answer.*

Secure Multi-party Computing

Implementation using RSA:

Solution

We use RSA. Let A be Alice's net worth, and B be Bob's net worth.

- *Alice publishes her public key (n, e), with private key d. Here $n = pq$ where p, q are primes of size N.*
- *Bob picks a random x of size N and calculates $C = E_{(n,e)}(x)$.*
- *Bob further calculates $m = C - B + 1 \mod n$, and sends it to Alice.*
- *Alice calculates $Y_i = D_{(n,d)}(m + i - 1)$, for each possible wealth value.*
- *Alice chooses a random prime P of size $\frac{N}{2}$, so that $Z_i = Y_i \mod P$ are at least 2 apart for different i's.*

Secure Multi-party Computing

Solution

- *Alice sets* $W_i = Z_i \bmod P$ *for* $i > A$*, and* $W_i = Z_i + 1 \bmod P$ *for* $i \leq A$*.*
- *Alice sends Bob the random prime* P*, and all the* $W_0, \cdots, W_{10^7}$*.*
- *Bob checks if* $W_B \equiv x \bmod P$*. If yes, Bob is richer. Otherwise, Bob's wealth does not exceed Alice's.*

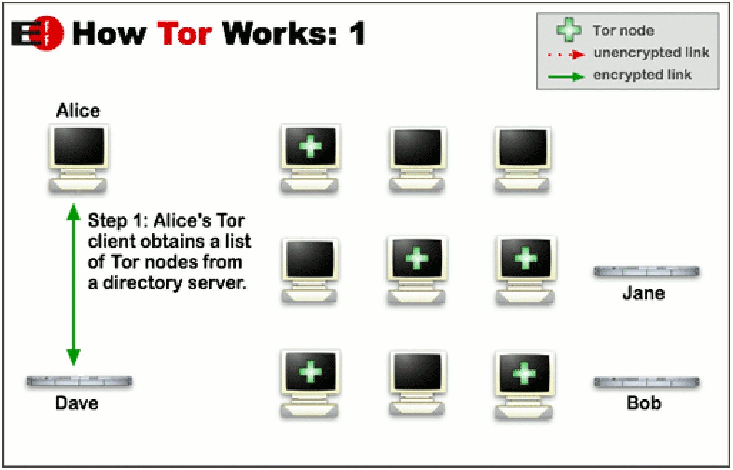
Secure Multi-party Computing

- This Millionaire problem started the field of *Secure Multi-Party Computing*.
- Each party has some secret information they don't want to reveal, yet they want to somehow combine the secret information to get something they all want. (e.g. Average age, Matchmaking, Who's Richer)
- Nothing about their secret should be revealed in this process, except the final result of the combination.
- All our solutions above used some secret-sharing schemes.
- Other methods are possible: oblivious transfer, homomorphic encryption, garbled circuits etc.
- Some applications: electronic voting (Helios), electronic bidding (Denmark sugar beet farmers), privacy-preserved data mining.

Tor Project

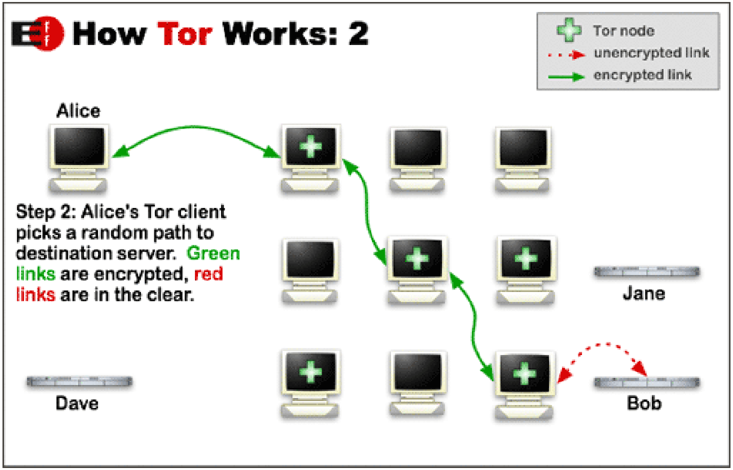
- Suppose that your network is being watched.
- Even if you encrypt your message, Eve can still know quite a bit about what you send/who you are talking to. (Header of data packet contains information about source, destination, size etc)
- Can you reduce the risk of such traffic analysis?
- Tor is a project trying to achieve this, by taking a twisted route to throw off somebody who is tailing you.
- Who needs such things?
    - Journalists under dictatorship
    - Opposition in autocratic regimes
    - Law enforcement, spies
    - Criminals, etc.

Tor Project


How Tor Works: 1
Tor node
unencrypted link
encrypted link
Alice
Step 1: Alice's Tor client obtains a list of Tor nodes from a directory server.
Jane
Dave
Bob

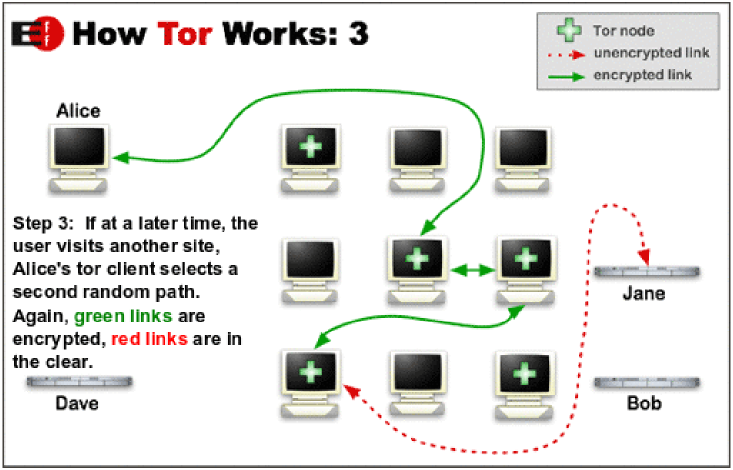
https://www.torproject.org/about/overview

Tor Project


How Tor Works: 2
Tor node
unencrypted link
encrypted link
Alice
Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.
Jane
Dave
Bob

https://www.torproject.org/about/overview

Tor Project

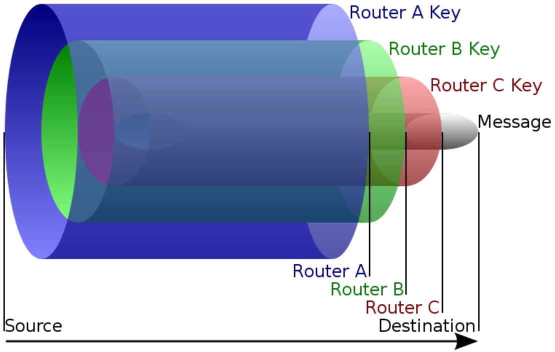


Step 3: If at a later time, the user visits another site, Alice's tor client selects a second random path. Again, green links are encrypted, red links are in the clear.

https://www.torproject.org/about/overview

Tor Project

- Multiple streams can use the same circuit.
- Onion cloud made up of anonymous volunteer-based onion routers (OR) which upload descriptors to directory servers.
- How does relaying between onion router work? Diffie-Hellman + AES.



Summary of the course

Alice wants to send a message to Bob.

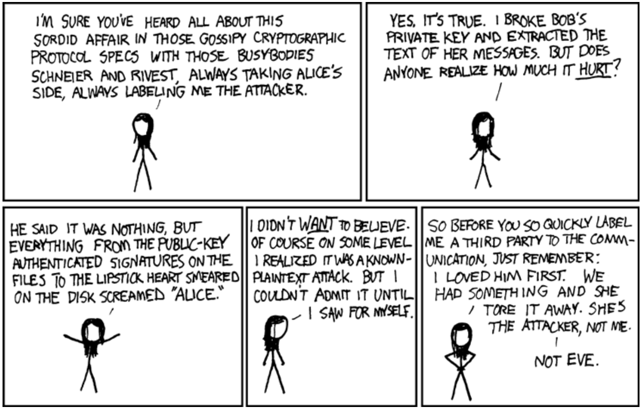
message to Bob
Alice
Bob

- Authentication: Alice wants to make sure she's talking to Bob.This can be done via Public Key Infrastructure (PKI) (e.g. digital certificates by trusted CAs), or some challenge-response schemes (Zero-knowledge proof).This step uses public-key cryptography. (e.g. RSA)

Summary of the course

- Once Alice is certain that she is talking to Bob, they establish an ephemeral(one-time), private, shared, session key. (e.g. by Diffie-Hellman)
- Alice can use *symmetric-key cryptography* (e.g. AES) to send messages to Bob, using the shared key.
- To make sure that Eve does not tamper with this message, Alice can also send a MAC (by symmetric key), or a digital signature (by public key), to check integrity of the message. It also serves as an extra layer of authentication.
- Bob decrypts the message using the shared key, and verifies it against MAC/signature.
- Alice and Bob communicate happily thereafter.

Summary of the course


I'M SURE YOU'VE HEARD ALL ABOUT THIS SORDID AFFAIR IN THOSE GOSSIPY CRYPTOGRAPHIC PROTOCOL SPECS WITH THOSE BUSYBODIES SCHNEIER AND RIVEST, ALWAYS TAKING ALICE'S SIDE, ALWAYS LABELING ME THE ATTACKER.
YES, IT'S TRUE. I BROKE BOB'S PRIVATE KEY AND EXTRACTED THE TEXT OF HER MESSAGES. BUT DOES ANYONE REALIZE HOW MUCH IT HURT?
HE SAID IT WAS NOTHING, BUT EVERYTHING FROM THE PUBLIC-KEY AUTHENTICATED SIGNATURES ON THE FILES TO THE LIPSTICK HEART SMEARED ON THE DISK SCREAMED "ALICE."
I DIDN'T WANT TO BELIEVE. OF COURSE ON SOME LEVEL I REALIZED IT WAS A KNOWN-PLAINTEXT ATTACK. BUT I COULDN'T ADMIT IT UNTIL I SAW FOR MYSELF.
SO BEFORE YOU SO QUICKLY LABEL ME A THIRD PARTY TO THE COMM-UNICATION, JUST REMEMBER: I LOVED HIM FIRST. WE HAD SOMETHING AND SHE TORE IT AWAY. SHE'S THE ATTACKER, NOT ME.
NOT EVE.

Summary of the course

What we did not do:

- Security of Protocol: It is often possible to quantify secrecy via Information Theory.
- Symmetric-Key Cryptography: something we did not touch upon.
- Pseudorandom generators: how do you actually generate random numbers? (In keys, salts, nonces, etc)
- Cryptographic Hash Functions: their actual constructions.
- More modern tools: Elliptic curves/Lattice-based cryptography/Quantum Computing,...
- All the details, basically.

Well, we didn't have that much time anyway...

Summary of the course

What next?

- Professor Dan Boneh's class on CourseEra. (Part I starting on Aug 3!)
  - First four weeks of Part I contains a discussion of symmetric-key cryptography that complements this class.
  - Last two weeks of Part I and early parts of Part II - you probably have seen many key ideas in this class but with little details.
- A list of great technical resources *http://blog.cryptographyengineering.com/p/useful-cryptography-resources.html*
- Learn some more math! For cryptography, perhaps some more Probability theory/Combinatorics.
- Of course, Number Theory is fascinating on its own.

> Mathematics is the queen of sciences and number theory is the queen of mathematics.

Hope you have a great summer!