Cryptographic Protocols 2

Luke Anderson

luke@lukeanderson.com.au

5th May 2017

University Of Sydney

Overview

1. Secret Splitting and Sharing

1.1 Secret Splitting

1.2 Secret Sharing

2. Commitment Protocols

2.1 Bit Commitment

2.2 Fair Coin Flipping

2.3 Mental Poker

Secret Splitting and Sharing

Secret Splitting

Problem: You are the CEO of Coca-Cola. You're responsible for keeping the formula secret from Pepsi's industrial spies. You could tell your most trusted employees, but ...

- ○ They could defect to the opposition.
- ○ They could fall to rubber hose cryptanalysis.

How can a secret be split among multiple parties such that each piece by itself is useless?

Secret Splitting with XOR

Suppose Trent wants to protect the message m:

1. Generate a random string r, the same length as m.
2. Compute $s = m \oplus r$.
3. Give Alice r, and give Bob s.

Each piece r, s is called a *shadow* of the message m. To reconstruct m, Alice and Bob can XOR their shadows together:

$$s \oplus r = m$$

If r is truly random, the system is perfectly secure (similar to One Time Pad).

The scheme may be extended to n people by generating $n - 1$ random strings $r_1, \ldots, r_{n-1}$. Give the first person r_1, the second person r_2, and so on up to r_{n-1}, and give the nth person $r_1 \oplus \cdots \oplus r_{n-1} \oplus m$.

Secret Splitting with XOR

Secret splitting aims to enhance reliability without increasing risk through distributing trust.

Issues: (with XOR)

- The system is adjudicated by Trent.
    - Trent can hand out rubbish and say it's the secret.
    - Trent can say he's splitting a secret 4 ways, but only splitting it between the first two people.
- All parties know the length of the message.
- The message is malleable: by flipping bits in any part, the recovered message changes.
- All parties are required to recover the message (*bus factor* $= 1$).

Secret Sharing

Problem: You are responsible for a small country's nuclear weapons.

- Ensure that no single lunatic can launch a missile.
- Ensure that no pair of lunatics can launch a missile.
- You want at least three of five officers to be lunatics before a missile can be launched.

This is called a $(3, 5)$-threshold scheme.

Secret Sharing

Shamir's Secret Sharing is an algorithm for dividing a secret into m pieces, where only n of them are required to reconstruct the original secret.

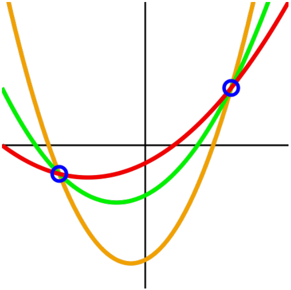
A polonomial of degree n can be uniquely defined by plotting $n + 1$ points on that polynomial.

Example

$$y = ax^2 + bx + c$$

Infinite possibilities for the coefficients with only 2 points.

3 points, and a, b, and c are uniquely determined.



Shamir's (t, n) threshold scheme

Fact: A polynomial $f(x)$ of degree $t-1$ is uniquely determined by t distinct points $(x, f(x))$ lying on the curve. This works over $\mathbb{Z}_p$, not just $\mathbb{R}$!

Trent wishes to distribute a message m amongst n users, where any group of t users $(1 \leqslant t \leqslant n)$ can recover m. (*bus factor* $= n - t + 1$)

1. Choose a prime $p > \max\{m, n\}$.
2. Create the polynomial $f(x) = m + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$, where the $0 \leqslant a_i < p$ are random and independent.
3. Trent selects n distinct points x_i, with $1 \leqslant x_i < p$.
4. Trent gives $(x_i, f(x_i))$ to person i.

Once t people pool their $(x_i, f(x_i))$ points, then the polynomial $f(x)$ can be reconstructed and m recovered. This can be done by *Lagrange interpolation*, for example.

Commitment Protocols

Bit Commitment

Problem:

- Alice wants to sell Bob information regarding police informants within his Mafia empire.
- Alice doesn't trust Bob enough to tell him the rats without getting paid first.
- Bob thinks the deal is a police setup, and won't give her the money until she commits to names.

Bit Commitment

Commitment:

1. Bob generates a random string r, and sends it to Alice.
2. Alice generates a random key k, and sends Bob $E_k(r \parallel m)$.

Revelation:

1. Alice sends Bob the key k.
2. Bob decrypts the message with k, and verifies r.

Discussion:

- r is needed for freshness, and to stop Alice from finding colliding messages, with $E_{k_1}(m_1) = E_{k_2}(m_2)$. She needs to commit at the time she receives r.
- Bob does not know k until revelation, so cannot brute force the message space.

Bit Commitment with Hash Functions

Commitment:

1. Alice generates random strings r, s, and computes $x = h(r \parallel s \parallel m)$. ($x$ is called a *blob*.)
2. Alice sends Bob (r, x).

Revelation:

1. Alice sends Bob the remaining data of (s, m).
2. Bob verifies that $h(r \parallel s \parallel m)$ is the same as the x he received.

Discussion:

- Bob does not have to send any messages.
- Alice sends a message to commit, and a message to reveal.
- Since h is a crypto hash function, Alice cannot find t such that $h(r \parallel s \parallel m) = h(r \parallel t \parallel m)$.
- s is kept secret so that Bob can't brute force the message space.

Fair Coin Flipping

Problem: Alice and Bob are arguing over the internet about who will be white (and therefore play first) in a game of online chess. They want to flip a coin to resolve the situation.

- Alice doesn't trust Bob to flip the coin.
- Bob doesn't trust Alice to flip the coin.

How can a coin be flipped fairly?

Fair Coin Flipping

转硬币有两种方法，这个是第一种

To flip a coin fairly:

1. Alice commits to a bit b using a commitment scheme.
2. Bob tries to guess the bit.
3. Alice reveals the bit: if Bob guessed correctly, he wins the toss. Otherwise, Alice does.

Discussion:

- The security of this algorithm lies in the security of the commitment scheme. *In particular*, the blob of the commitment scheme should not give away anything about the message inside (such as low-order bits).

Fair Coin Flipping using Public Key Crypto

第二种

We require a *commutative* public key cryptosystem, for example ElGamal, so that

$$E_A(E_B(m)) = E_B(E_A(m))$$

To perform a fair coin flip:

1. Alice and Bob generate keypairs A, B respectively.
2. Alice generates two random numbers r_T and r_H.
3. Alice sends Bob $m_1 = E_A(\text{heads}, r_H)$ and $m_2 = E_A(\text{tails}, r_T)$ in a random order.
4. Bob selects one of Alice's messages, call it x, and sends Alice $E_B(x)$.
5. Alice decrypts Bob's message and sends it back: $D_A(E_B(x))$.
6. Now Bob is left holding $E_B(m_1)$ or $E_B(m_2)$: he can decrypt this and send it back to Alice.
7. Alice verifies that Bob's response matches up with her r_T or r_H.

Fair Coin Flipping using Public Key Crypto

Discussion:

- The algorithm is self-enforcing: either party can detect the other cheating, without requiring a trusted third party.
- Bob learns the result of the coin flip before Alice. He can't change the result, but he may delay it ("flipping the coin into a well").
- Coin flipping has use in session key generation, as neither party can influence the result of each flip. For example, in Diffie-Hellman, one party selects an exponent after the first.

Mental Poker

Problem: Alice and Bob want to play poker over email.

- Alice doesn't trust Bob.
- Bob doesn't trust Alice.

How can Alice and Bob deal hands fairly?

Mental Poker

Use a commutative public key cryptosystem.

1. Alice and Bob generate keypairs A, B respectively.
2. Alice encrypts the 52 messages $m_1 = (\text{ace of spades}, r_1), \ldots$ using her public key, and sends these blobs $x_1, \ldots, x_{52}$ to Bob.
3. Bob picks 5 of the blobs at random (or however he pleases), encrypts them with his public key, and sends them back to Alice.
4. Alice decrypts the messages with her public key, and sends them back to Bob.
5. Bob decrypts the messages: this is his hand.
6. At the end of the game, Alice and Bob may reveal their keys to ensure no-one cheated.

How is Alice's hand dealt?

alice 怎么弄？

Attacks against the Poker Scheme

Cryptosystems (especially ones based in number theory) tend to leak small amounts of information, if not used in conjunction with hash functions.

与。。。一起

For example, in RSA, if the number representing the card is a quadratic residue (a square number modulo the RSA modulus), then the encryption of the card is also a quadratic residue. This could be used by the dealer to “mark” certain cards.

ELEC5616 COMPUTER & NETWORK SECURITY

Lecture 14b: Covert Channels & Steganography

SUBLIMINAL CHANNELS

Problem:

Alice and Bob have been arrested for conspiracy to factor large numbers by the government.

Alice has been sent to a woman’s jail, Bob to a men’s jail.

The warden, Walter, is willing to let them communicate on the condition that messages are not encrypted.

How can Alice and Bob communicate secretly given Walter can read their messages and might attempt to deceive them by planting false messages?

SUBLIMINAL CHANNELS

Alice and Bob can set up a subliminal (or covert) channel

Simplest level: Alice and Bob can use ***steganography*** (information hiding) to place in hidden messages in places no-ones suspects a message exists

Steganography is not cryptography: it’s “security through obscurity”

Steganography is usually used together with cryptography

EXAMPLE OF STEGANOGRAPHY (TEXT)

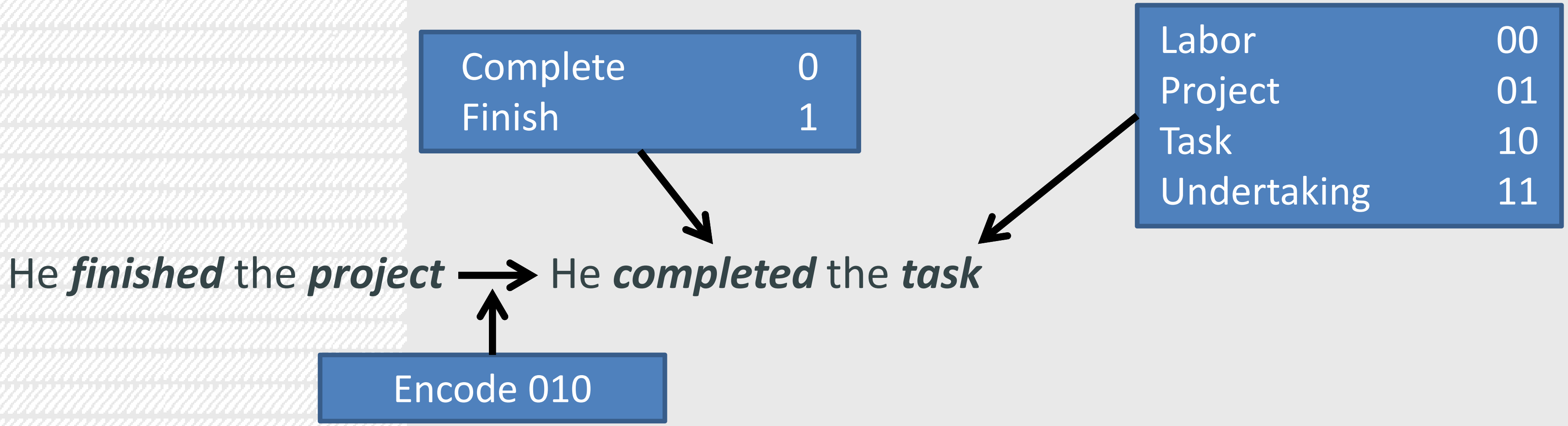
Naive, obvious and bloated: count(words in sent) => if even = 0, if odd = 1

Synonym substitution (0.67 bits per sentence) [Topkara et al.]

Syntactic substitution (0.5 bits per sentence) [Atallah et al., Topkara et al.]

Contextual Synonym Substitution and Vertex Color Coding [Chang & Clark]

Slightly higher than 1 bit per newspaper sentence



Complete	0
Finish	1

Labor	00
Project	01
Task	10
Undertaking	11

He ***finished*** the ***project*** → He ***completed*** the ***task***

Encode 010

EXAMPLE OF STEGANOGRAPHY (IMAGE)

Imagine a grayscale (8 bitsof gray) lossless 32x32 pixel icon

128	64	32	16	8	4	2	1

128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255 + (zero) => your 256 different grays

Imagine if you were happy with a slight trade off – get rid of one bit

128	64	32	16	8	4	2	Msg

Original and modified image will be imperceptibly different to the naked eye

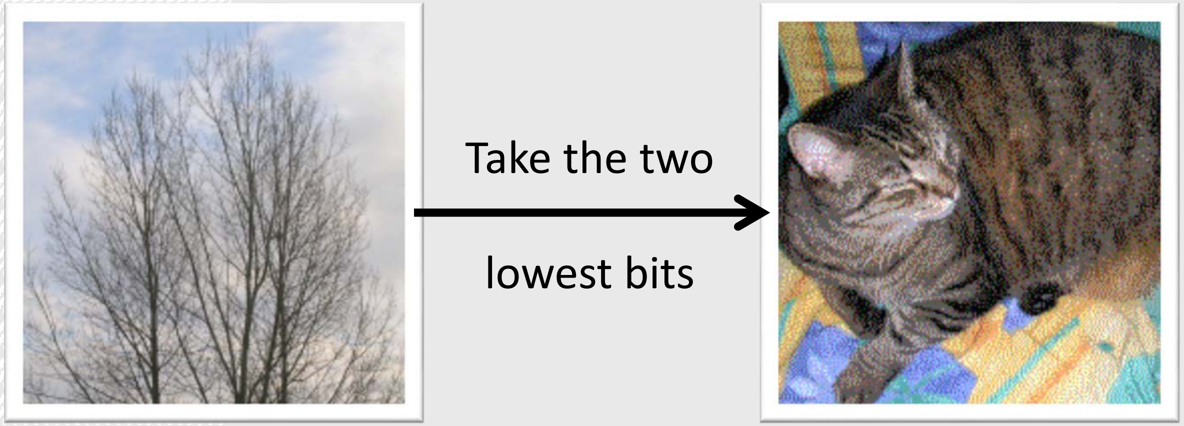
Yet you can now encode a secret message 1024 bits (128 characters) long!

EXAMPLE OF STEGANOGRAPHY (IMAGE)

If you were using a colour image (RGB) you can steal 3 bits per pixel

Double the resolution and you quadruple the amount of data you can hide

Settle for some lost quality and you can go to 6 bits per pixel (2 bits per color)

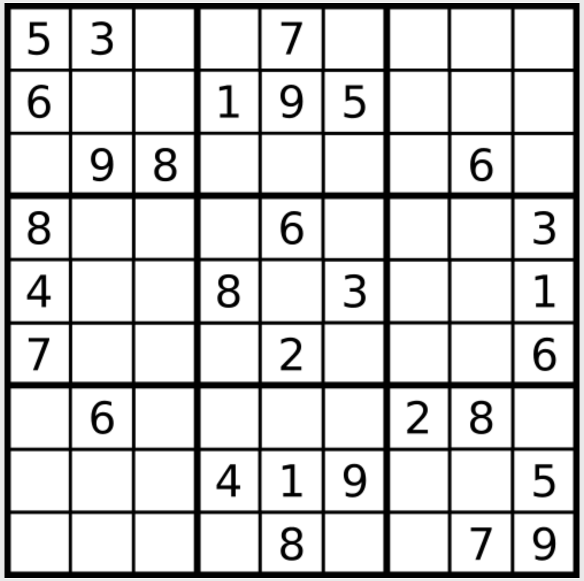


Take the two

lowest bits

Techniques exist that work with lossy recompression and even resizing

EXAMPLE OF STEGANOGRAPHY (CRAZY)



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Sudoku has approx 2^{70} different possible solutions, longer than DES key

A standard shuffled deck of cards is one of 52! or 2^{230} different combinations (see *Solitaire cipher* from Cryptonomicon & Bruce Schneier)

NETWORK STEGANOGRAPHY

Loki

- Daemon9, Alhambra (phrack/the guild)
- Bidirectional covert UNIX shell client using the data field in ICMP type 0 (Echo Reply) and type 8 (Echo Request) packets.

Daemonshell-UDP

- ICMP Echo Reply only (more stealthy)

ICMP Backdoor

- Reusable tunnel library
- Messages fragmented to look more like ping packets (multiples of 64 bytes)

Rwwwshell

- Backdoor emits requests as HTTP Response packets
- Output from commands return from the slave as cgi script HTTP GETs

B0CK

- IGMP multicast messages used as transport

AckCmd

- TCP ACK packets for request (port 80), TCP RESET packets for response (high port)

FIRESMITHING

The head of IT at XYZ tells you they're entirely leak safe:
"I disabled SSL, have a limited white-list of websites the users can access and monitor all their outgoing email for sensitive documents..."

Can data still leave this network? Yes, of course!

Ask Google Translate to access mattbarrie.com/?Firesmithing
Google Translate fetches the page in order to retrieve the content to translate

```
 [Google's IP] - - mattbarrie.com 193.239.120.148:80 [date]\
"GET /?Firesmithing HTTP/1.0" 200 7863 "-"\ "browser (via
translate.google.com)"
```

Use DNS: send data out by making requests to turn domain names into Ips
facebook.com => 173.252.110.27
answers-to-exam.smerity.com
Want SSH? You can get SSH over DNS. Not fast but entirely doable.

HOW DO WE PROTECT AGAINST THIS?

Start to see the problems with content filtering?

Consider national content filters

- Great firewall of China (and other middle eastern countries)
- Australian Government NetAlert

Corporate content filters

Net-nannies

What if malware used these techniques to communicate?

- Answer: they do

Could other internets be layered onto the Internet