

# DOTA2024:6

## Defense of the Ancients

### Sixth topic - Communication methods

---

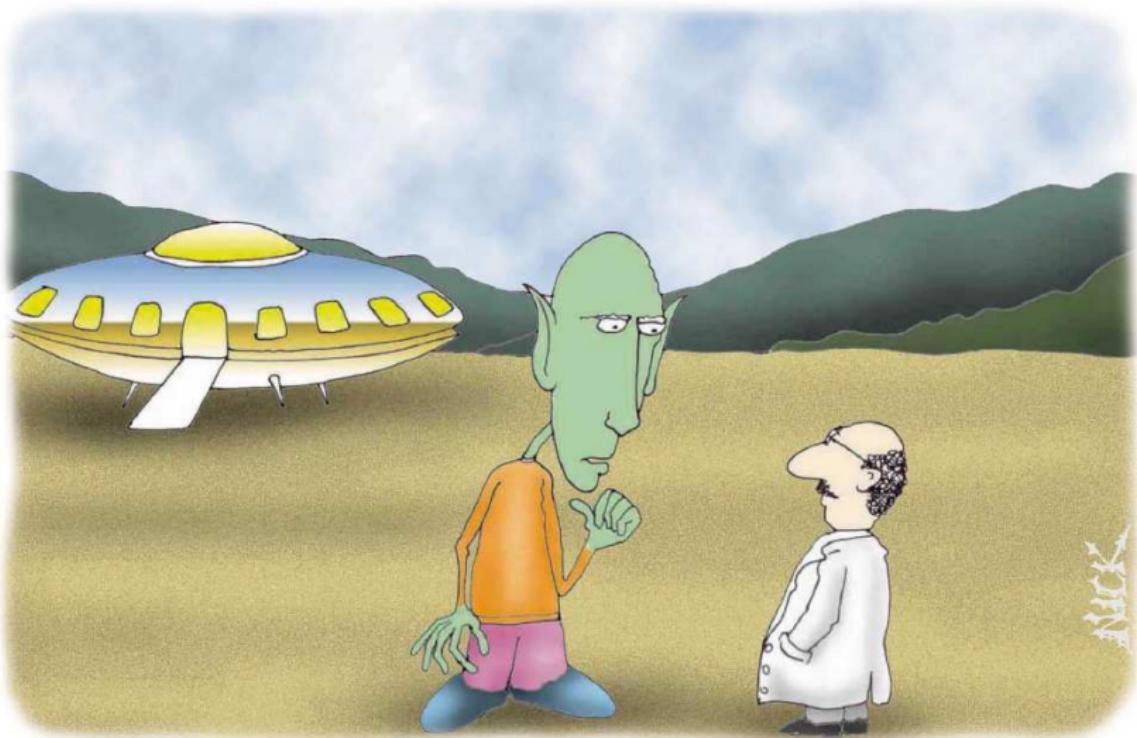
Hugh Anderson

National University of Singapore  
School of Computing

July, 2024



# TLAs...



*"That? No, that isn't a U.F.O. It's on the ground now,  
and you've identified it. That makes it an I.G.O."*

# Outline

## 1 Its not just the Internet

- Othernets
- The radio spectrum

## 2 Fighting back

- Awareness, evaluation, consideration

## 3 Padding oracle attack for Lab2

- Padding, release of partial information



# Other networks

## Hackable things are everywhere...

The Internet is not the only network of computers.

---

All around us, things we depend on are built on **less well known networks**:  
**Water** in the taps, **power, sewage**, gas, oil, automated factories, security systems. They are characterized by:

- ① **archaic** protocols (Allen Bradley...), security by **obscurity**
- ② Developers who claim that “Our engineers/technicians would **not tolerate passwords**”
- ③ Languages that should have died years ago (**Ladder languages** built from relays and switches), and the claim that “Our engineers/technicians would **not tolerate HLLs**”
- ④ **Radio links, no encryption, dial up** and leased phone lines

# OtherNets

**They are everywhere! Here is a (partial) list:**

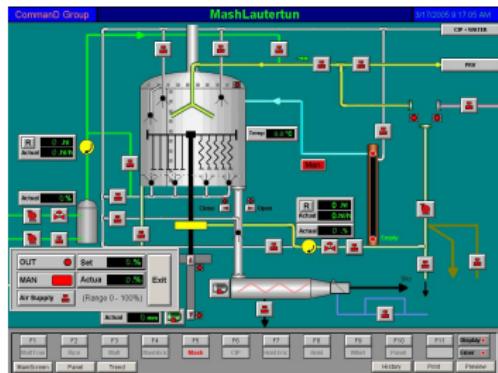
- Phone (Symbian, Android, iOS - all hacked)
- The old **wired phone network** (exchanges are computers these days)
- Embedded systems, PLC networks (in your **plane**, your **car**, your **factory**)
- The wireless **phone network** (wireless links, base stations)
- Anarchic/private control networks/systems
  - **public utilities** - dam control, power reticulation, sewage (dial-up, radio links, SCADA, Allen Bradley...)...
  - **commercial** - petrol pumps, cash registers, shop stocklist systems (dial-up, leased lines or backboned over Internet)
- **Home networks** (security, environment control ...)

# Othernet hacking scenario #1

Welcome to the wonderful world of SCADA/PLCs...

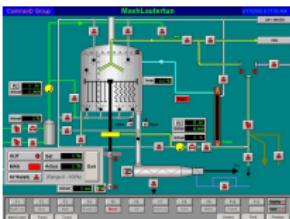
SCADA (supervisory control and data acquisition) is everywhere. Kind of McDonalds but for industrial control, automation, remote management. SCADA equipment controls Singapore's water, gas pipelines, sewage, security systems and so on.

Unfortunately security is seldom addressed, and otherwise sensible people still secure their systems with obscurity.



# Othernet hacking scenario #1

Welcome to the wonderful world of PLCs. 2 examples:

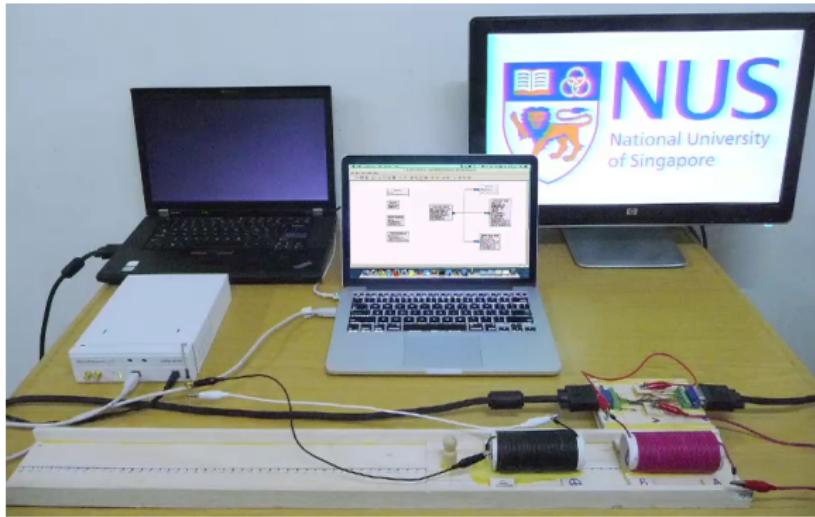


- ➊ An [angry man](#) (now jailed) attacked the Maroochy Shire sewage control system (Queensland, Australia), and [released a million litres of raw sewage](#) into the environment, using a computer and a special radio.
- ➋ A sophisticated program may have damaged a fifth of Iran's nuclear centrifuges. [Stuxnet](#) specifically targeted [SCADA](#) and Siemens [PLCs](#), and was probably targeted at Iran.

# Balloons should be afraid; very very afraid



# Topic: Insecurity



## Tempest - Resurrecting images from cable radiation

Paper at:

<https://hugh.comp.nus.edu.sg/DOTA/Files/JinZhouFYP.pdf>

# Othernet hacking scenario #1

## Spooky CIA stuff...

The CIA claim that there have been multiple instances of successful blackmail of power utility companies around the world. (Note that this may be part of a general strategy of convincing people the world is “dangerous”, so perhaps we should take it with a grain of salt).

The idea is that the blackmailer threatens to turn off the power grid, and (according to the CIA) have turned off power to cities to demonstrate.

---

Two documented instances of similar activities:

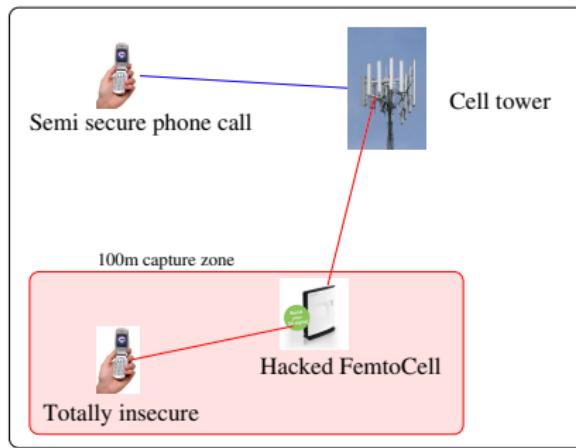
- ① Russia's gas utility control system being taken over for a short period, and also
- ② Disgruntled employees in the US hacked traffic light systems in LA to cause chaos

# Othernet hacking scenario #2

## Welcome to the wonderful world of femtocells...

Vodafone's [SureSignal](#) is a neat product which makes cell signals stronger in houses and buildings. It is a small cell base station, called a femtocell.

SureSignal was hacked in 2012. A hacked system can forge calls, record calls, and collect detailed data about the phone/SIM so it can be copied.



# Othernet hacking scenario #3

**If you cant see it, it must be secret...**

## **The tram switch hack:**

A Polish teenager hacked the tram system in the city of Lodz and derailed four trams. Many people were injured. He modified a TV remote control to change track points...

TV-begone... Tram-begone?

# Othernet hacking scenario #4

## Welcome to the wonderful world of RFIDs...

MIFARE Classic, a [contactless smartcard](#) widely used in Europe.

---

It uses a set of [proprietary](#) protocols/algorithms, that were reverse-engineered in 2007. It turns out that the encryption [algorithms](#) are already known to be [weak](#) (using only 48bits) and breakable

<http://en.wikipedia.org/wiki/MIFARE>. All software and hardware details are open, and available for download.

## Othernet hacking scenario #4



London underground Oyster card, Malaysia Touch&Go, BMWs  
etc

# Othernet info leaks, projects last sem...

**CS3235-01 Measure Population with GSM**

**Project Code:** [Aman Sundararan, He Molin, Liu Xuxin, Ng Chi Hau, Suresh Venkatesh](#)

**Abstract**  Our project explores GSM-based methods to infer the population density of an area by intercepting unencrypted cell tower broadcasts. We use this information to make predictions about the number of individuals in a location and correlate with popular bars and restaurants in the area.

**Introduction**  GSM is a 2G cellular network. Making use of the fact that most people have a GSM network.

**Methods** 

- Find out the frequency for each location
- Capture encrypted SMS/Toll and decode it
- Find number of users around a BTS

**Analyses** 

We wrote programs to automatically record the frequency and strength of all possible signals and analyse which ones are the strongest.

**Sponsors** 

**Conclusion**  Due to the nature of global networks, we managed to infer the traffic in the network. The results of population density can be put in good use.

**CS3235-08 GSM-BASED INTRUDER DETECTION & IDENTIFICATION SYSTEM**

**Introduction**  GSM is a primary used to detect intruders with the help of the mobile phones using GSM technology.

**Application: Intrusion Detection System** 

**GSM Authentication** 

**Members** 

**Aim of Experiments** 

To verify the capacity of our base station to monitor a GSM (2G) network through three experiments: 1) Manual Association, 2) Automatic Association, and 3) Performance of Base Station.

**THE EXPERIMENT**

**Manual Association** 

In this method, we detect the signal from handset as our target. Once the phone is placed within the broadcast radius, it will connect to our GSM network to associate with our base station.

**Automatic Association** 

Automatic association automatically associates with our base station. Once the phone is placed within the broadcast radius, it will connect to our GSM network to associate with our base station.

**Sponsors** 

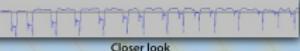
**CS3235-10 NFC Sniffing Via SDR**

**Motivation behind this project**  Have you ever wonder what is going on in your student card or e-link card when you tap on a reader?

Is the information really safe in the card?

**Sniffing data** 

**Sniffed data waveform** 

**Closer look** 

**Actual Data** 

Project by:  CHEAH KIT WENG  JOANNE MAH JIA WEN  TERENCE GAN SHIEN PIN  QIWER JIE PING  LIM ZHEN MING 

Hardware hacking using \$25 and \$500 software defined radios, social engineering, inference...

# Summary

**It sounds bad. Many systems could affect us...**

Not only routers, printers, servers, PCs, but also PLCs, control systems. In each of these complex systems, they are only **as strong as their weakest link**.

Taking the cases of Internets and othernets - There is no administrative centre for the Internet, little security (and no administration) on Othernets.

Just organizations trying to make money, cutting corners.

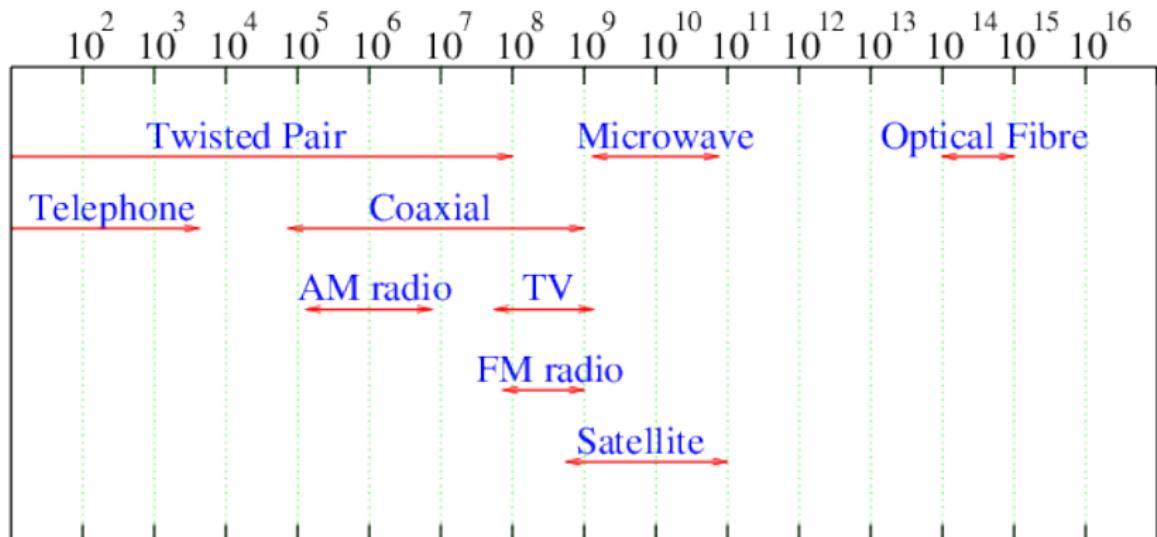


It seems like we cannot trust the machines!

Who knew?



# From Hertz, to Terahertz...



# Tools to access the radio spectrum...



or



Hardware hacking using \$25 and \$500 software defined radios. A range from 1MHz to 5GHz.

# Lets see what is at 1090MHz

The screenshot shows a map of Southeast Asia with a focus on Singapore and its surroundings. The map includes labels for Johor, Pasir Gudang, Nusajaya, Jurong Island, Pulau Batam, Pulau Kapalajernih, and Pulau Bulan. Overlaid on the map is a flight tracking interface. On the left, there's a legend for roads and water bodies. In the center, a small airplane icon is positioned near the Singapore Strait. To the right of the map, there are two circular clock-like displays: one for Local Time and one for UTC Time, both showing the current time. Below these are buttons for '[ Reset Map ]' and '[ Settings ]'. A section titled 'DUMP1090' provides real-time flight information: Altitude: n/a, Speed: n/a, Track: n/a, Squawk: n/a, ICAO (hex): n/a, and Lat/Long: n/a. Below this is a table of flight data:

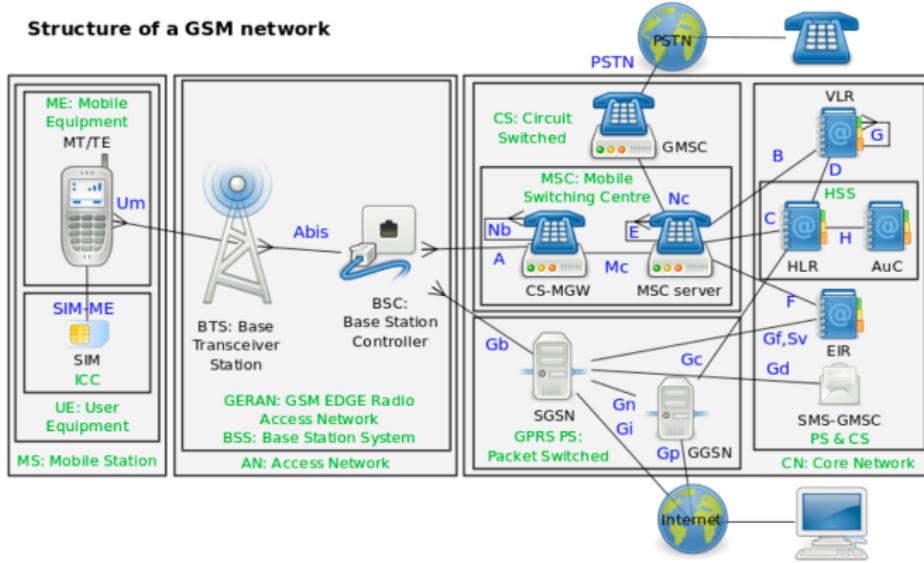
ICAO	Flight	Squawk	Altitude	Speed	Track	Msg	Seen
75025f			0	491	297	1	45
75012b	A3H716	2265	9575	291	355	197	1
76ce65	BIA117	2112	10500	309	82	283	10
800733		0162	11475	346	320	17	4
76cdcc		0136	20000	0		10	26

At the bottom of the map area, there are links for 'Google', 'Map data ©2016 Google', 'Terms of Use', and 'Report a map error'.

Radio signals all around us are supplying us with interesting information.

# Phones and GSM

## Mobile stations with SIM...



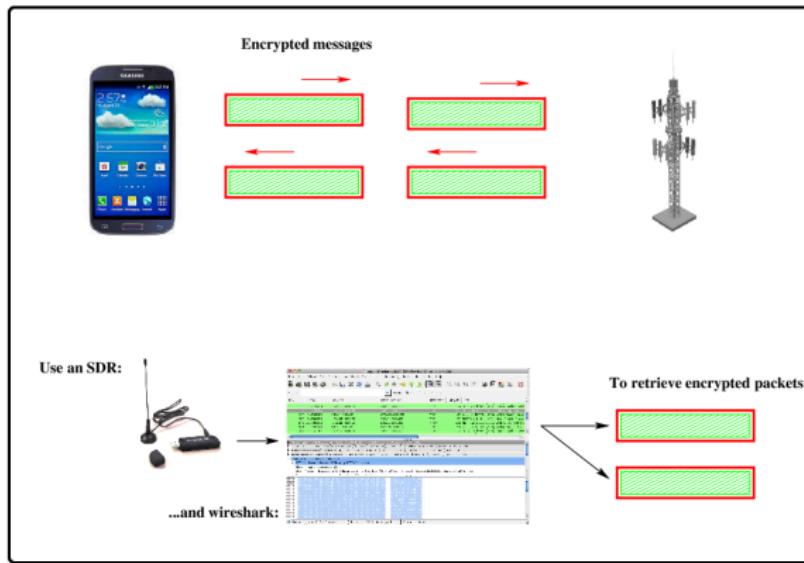
The SIM keeps the private identity of a phone...

Base stations authenticate the SIM/phone.

GSM phone networks on 900/1800 MHz - can we look at them with an SDR?

# An attack on phones and GSM

Step 1: get the packets/frames off the air.....



The packets/frames are 124bits, and some are always the same. In particular, my students found that in Singapore, a “system information type 6” message was found always 306 frames away from another fixed frame. The decoded frame was all 0x2b!

# An attack on phones and GSM

## If the plaintext is known...

... but the key is not, this is like a hash function. We have seen it before.

What can we do?

- Keysize is large (128 bits), but build a rainbow table. The one we have was built in Germany a few years ago, and is 2TB.
- The rainbow table, with good probability, discovers the input to the hash function i.e. the key!

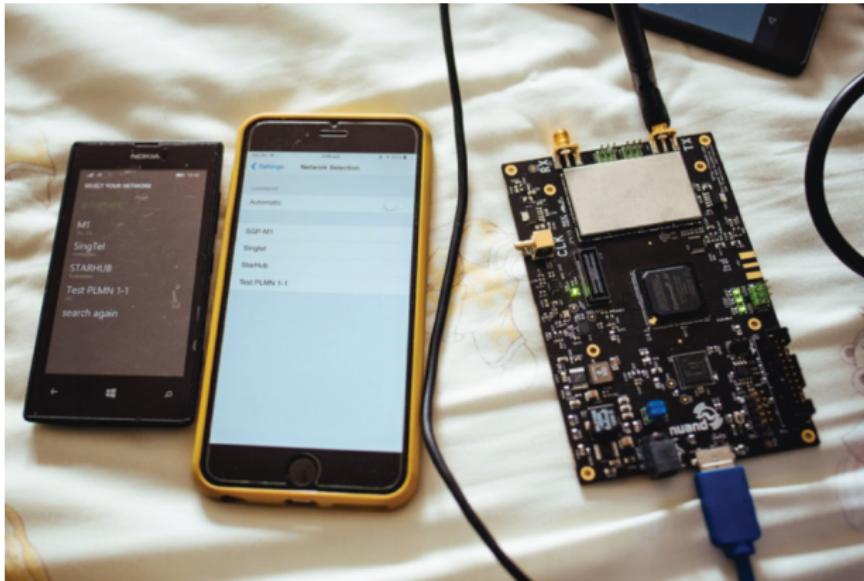
---

As a result, we have the key used by the phone, and are able to decode this session - [the students successfully decrypted SMS messages](#).

They were not able to do voice, because the simple SDR they were using could only receive a single channel at a time.

# GSM man-in-the-middle

Using software defined radios...



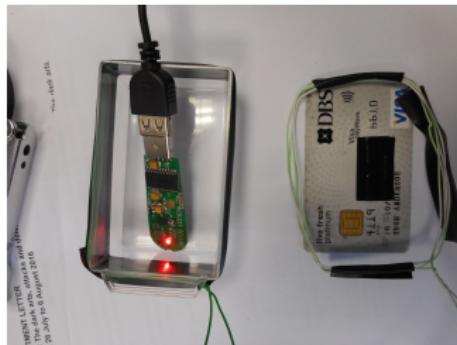
The SDR on the right is pretending to be a base station, the phones on the left can see it... With some care, you can make the phones associate to the SDR instead of Singtel/Starhub, and retrieve interesting information. Students developed a social engineering SMS sequence...

## More things to worry about:

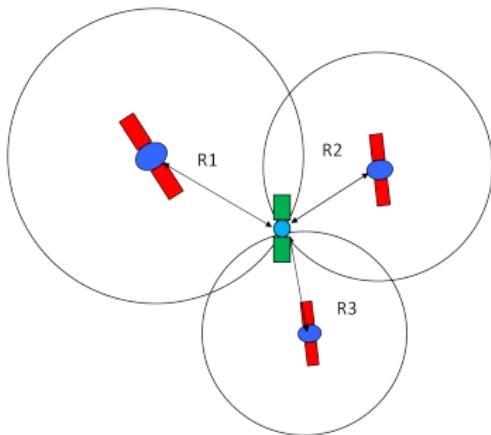
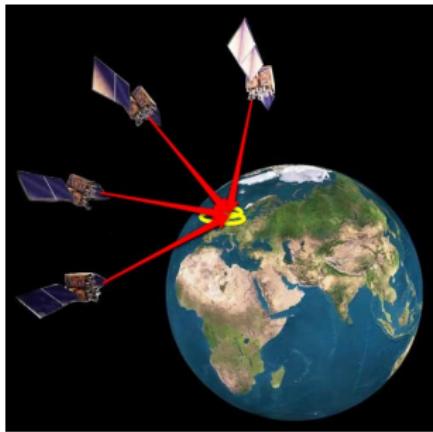


INSIDE  
newspaper

# A demo with NFC...



# Not sure if this fixes the pokemon GO situation...



---

The satellites are synchronized in time, transmitting on-the-nanosecond signals, which are received in your phone.

The difference in time (signals travel at  $300\text{m/uS}$ ), can locate the phone to about 15m.

# Surface 4 attack, simulate GPS...

## A recipe...

Download the `gps_sdr_sim` from <https://github.com/osqzss/gps-sdr-sim>.

---

Get the latest ephemeris ffrom

`ftp://cddis.gsfc.nasa.gov/gnss/data/daily/`.

(An ephemeris gives the positions of astronomical objects, in this case the satellites, which are decaying and moving about due to gravity variations).

I got `ftp://cddis.gsfc.nasa.gov/gnss/data/daily/2016/brdc/brdc2160.16n.Z`.

---

Generate a simulated set of signals from the ephemeris, and a latitude and longitude...

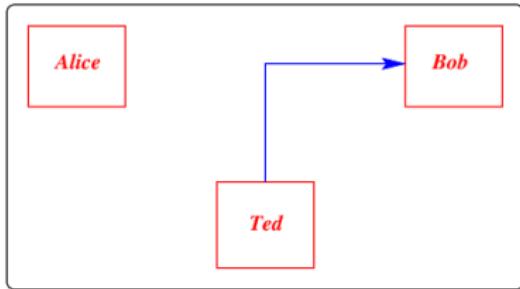
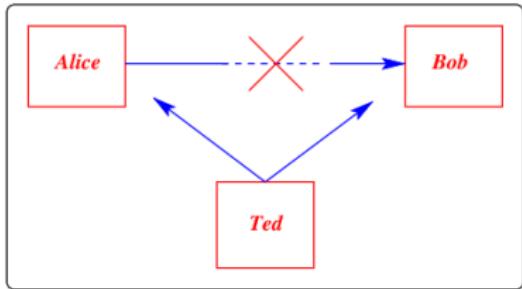
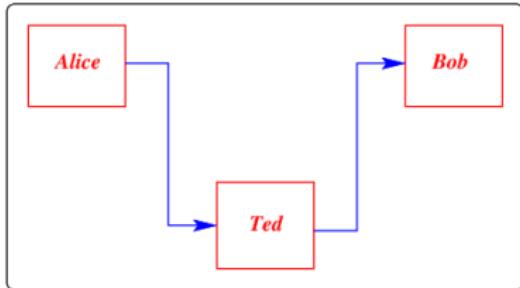
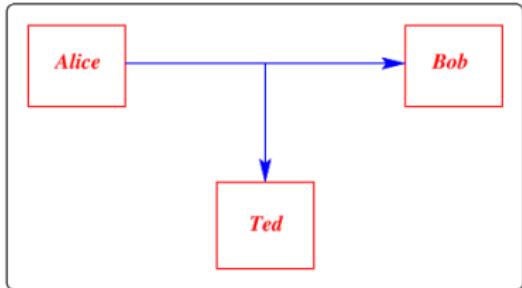
```
./gps-sdr-sim -v -e brdc2160.16n -b 16 -s 2500000 -l -39.332079,175.342537
Hughs-MacBook-Pro:gps-sdr-sim-master hugh$ ls -al gpssim.bin
-rw-r--r-- 1 hugh staff 2999000000 Aug  4 09:11 gpssim.bin
Hughs-MacBook-Pro:gps-sdr-sim-master hugh$
```

---

And then attach an SDR, and start sending it...

```
./tx_samples_from_file --args="master_clock_rate=50e6" --file gpssim.bin \
--type short --rate 2500000 --freq 1575420000 --gain 0 --repeat
```

# Attacks and defences anywhere in the spectrum



(... or ... Interception, Modification, Interruption, Fabrication)

# Evaluation

## **It is not magic:**

We should evaluate and manage this level of our systems in the same way as we assess and control our computer systems.

---

Assuming we had a system that made use of communication infrastructure, we should:

- evaluate it's security,
- do a risk assessment,
- ensure we have appropriate controls...

# Standards

## NIST Special Publication 800-53

For security controls for all U.S. federal information systems. Its approach seems relevant to assessing any security issue, including ones like this.

Within the framework:

- **Technical issues** include access control, audit and accountability, identification and authentication, and system and communication protection.
- **Operational issues** include awareness and training, contingency planning, incident response, maintenance and physical protection, and even personnel.
- **Management issues** include certification, risk assessment and so on.

# Summary

## **It is still not magic:**

Defences against communication attacks can encompass a wide range of techniques.

---

Unfortunately, there is no magic bullet for defending our communication schemes, except perhaps to

- reach for end-to-end security, where even a corrupted communication network cannot result in malicious activities.

However, even in this scenario, a DoS attack may be possible.

# Bit and byte padding for block ciphers...

## PKCS #7 - a byte-wise padding technique...

One technique is to add a single 1 bit, followed by enough 0 bits to fill out the block.

Another is the PKCS #7 paddings for the final block. Each byte is given as 2 hex digits:

### Valid PKCS #7 Padding:

Last block data

48	65	6c	6c	6f	20	77	01
----	----	----	----	----	----	----	----

48	65	6c	6c	6f	20	02	02
----	----	----	----	----	----	----	----

48	65	6c	6c	6f	03	03	03
----	----	----	----	----	----	----	----

48	65	6c	6c	04	04	04	04
----	----	----	----	----	----	----	----

48	65	6c	05	05	05	05	05
----	----	----	----	----	----	----	----

48	65	06	06	06	06	06	06
----	----	----	----	----	----	----	----

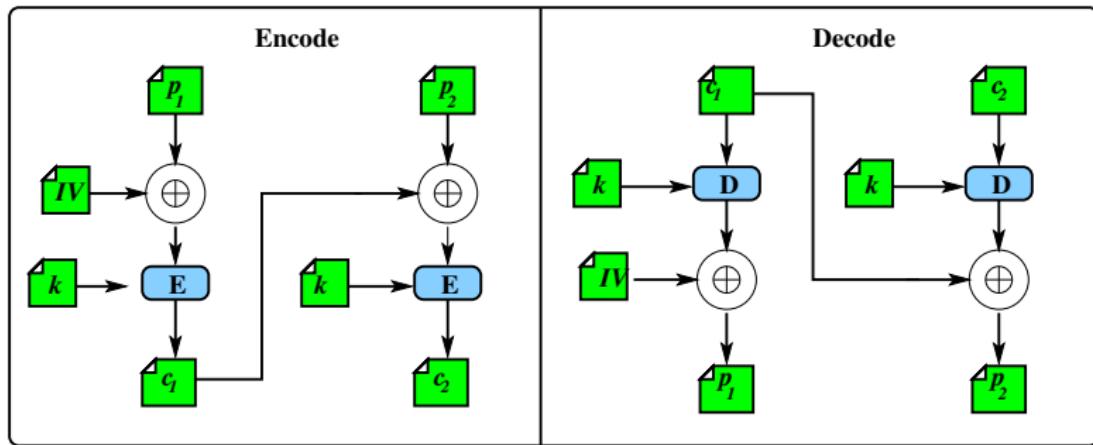
48	07	07	07	07	07	07	07
----	----	----	----	----	----	----	----

08	08	08	08	08	08	08	08
----	----	----	----	----	----	----	----

Padding

# CBC: encoding and decoding...

Encode is XOR, then encrypt...



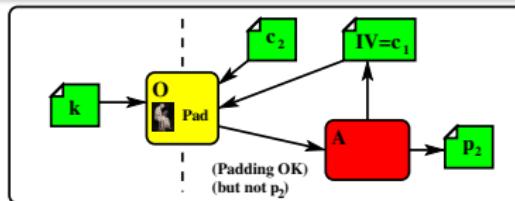
(...and of course, decode is decrypt, then XOR).

Note that when you are decoding, the final plaintext  $p_2$  is the decrypted ciphertext exclusive-ored with  $c_1$ .

---

If there was only one block, the final plaintext  $p_1$  is the decrypted ciphertext exclusive-ored with  $IV$

# Padding oracle...



## What is a padding oracle?

Imagine that you have a bit of software, that, given  $c_1$  and  $c_2$ , will tell you if the padding is correct in  $p_2$ . If the message was only one block long, given  $IV$  and  $c_1$  (as in the lab), it will tell you if the padding is correct in  $p_1$ . In general, given  $c_{n-1}$  and  $c_n$ , it will tell you if the padding is correct.

---

Is this realistic? Well, yes it is.

When you send a padded, encrypted message to another system, after decrypting, the other system will check the padding, and if it is not valid, will fail immediately. Otherwise it will fail (or succeed) later.

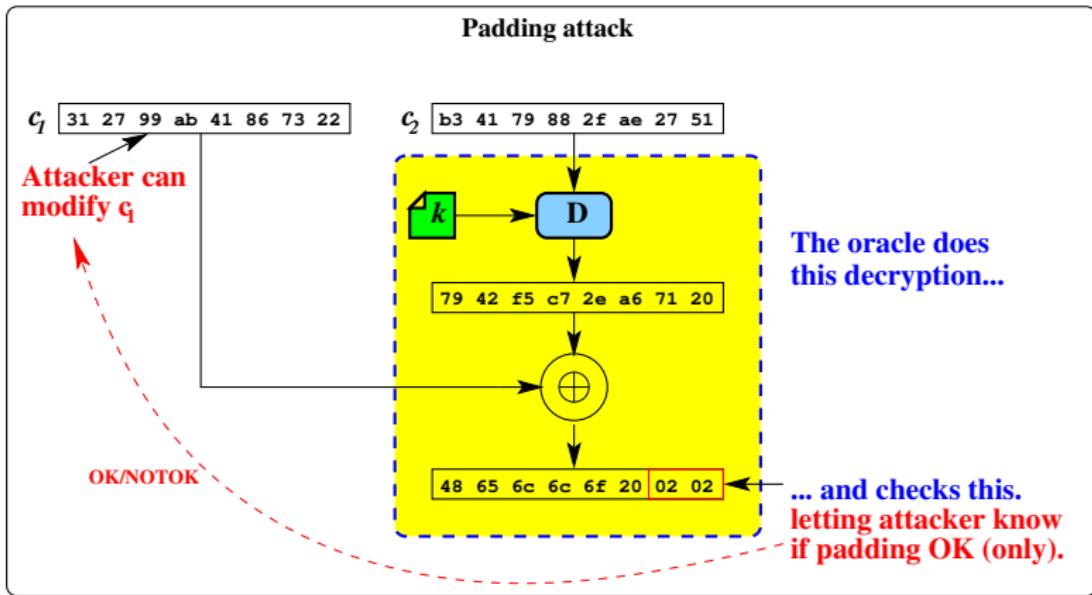
Normal receiving software may leak the correctness of the padding. In the lab, we will give you a program that is a padding oracle.

---

Note also that you can directly change each bit/byte of the final plaintext for  $p_n$ , because the attacker has control over the input  $c_{n-1}$  (or IV as in the lab).

# Using the oracle in an attack...

The attacker does not get to see the decrypted data...

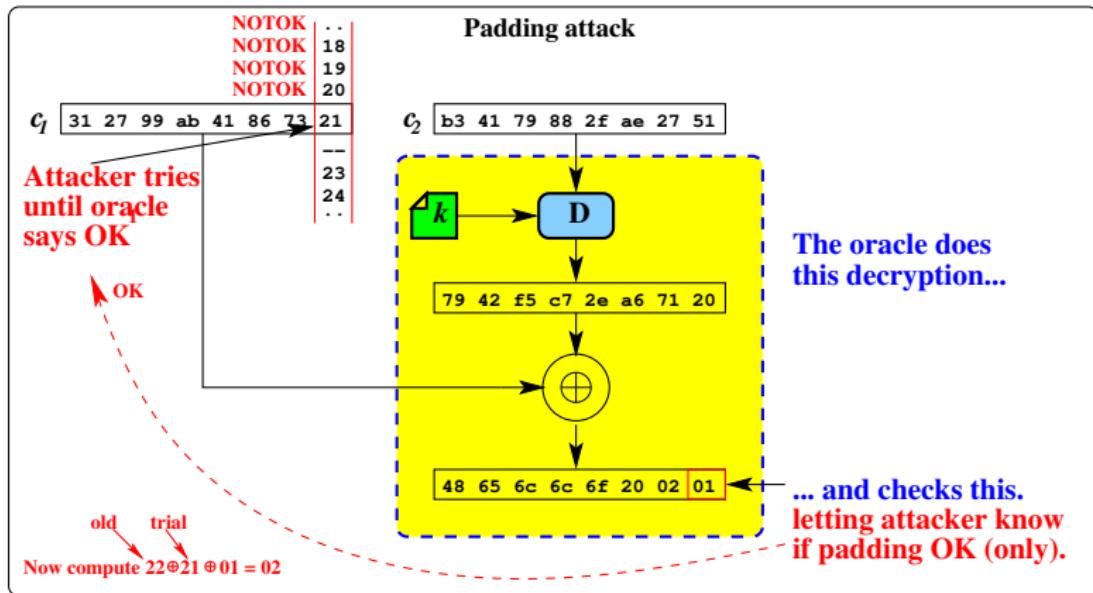


The oracle only learns if the last bytes are correct - they could be 01, 0202, 030303, and so on.

The attacker starts by trying a series of values for the last byte of  $c_1$ :

# Using the oracle in an attack...

Attacker discovers last byte...

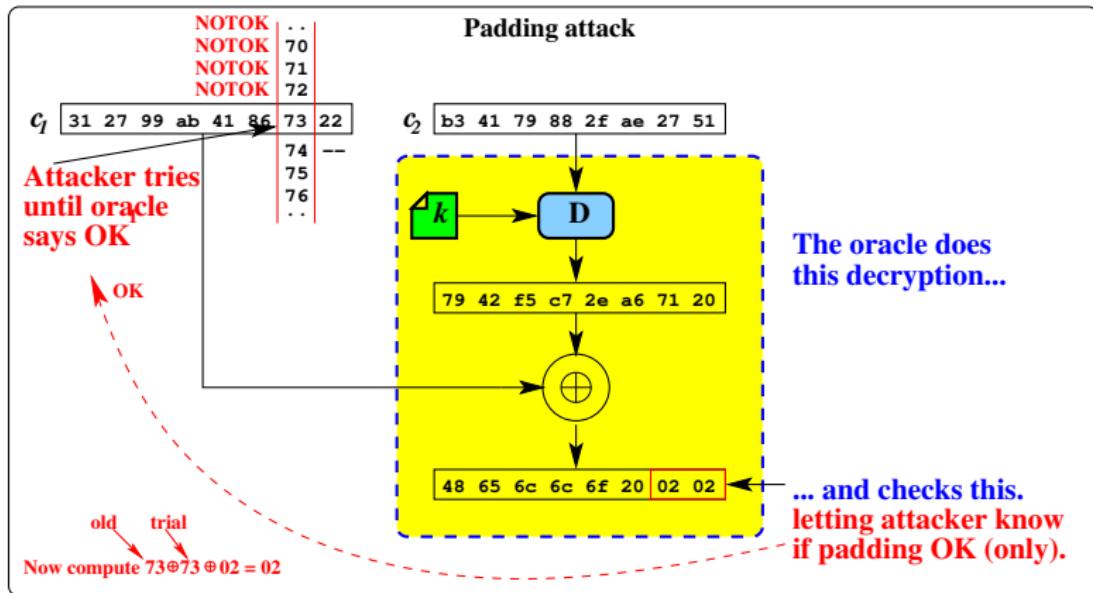


The attacker is trying values in the last byte of  $c_1$  until 01 is the last byte. The oracle will return padding OK...

Attacker now sets the last byte to 02, and tries for the second-last byte of  $c_1$ :

# Using the oracle in an attack...

Attacker discovers second-to-last byte...

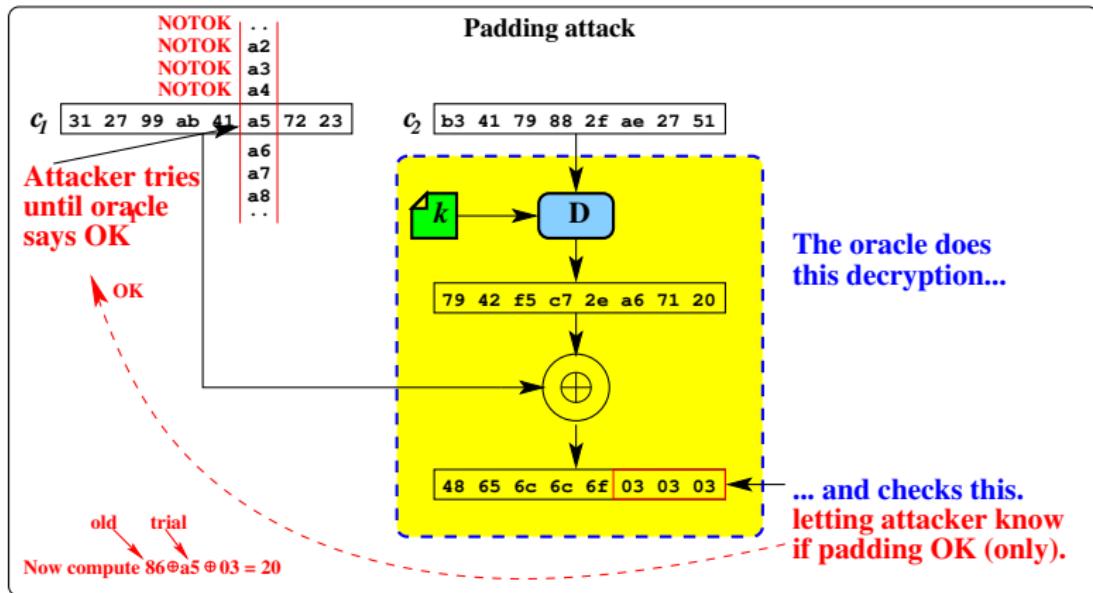


Since we know (now) the last byte (02), and the original last byte of  $c_1$ , we can set the last byte to anything we want.

We try all the second-to-last values in  $c_1$ , and uncover a second byte.

# Using the oracle in an attack...

Attacker discovers third-to-last byte...

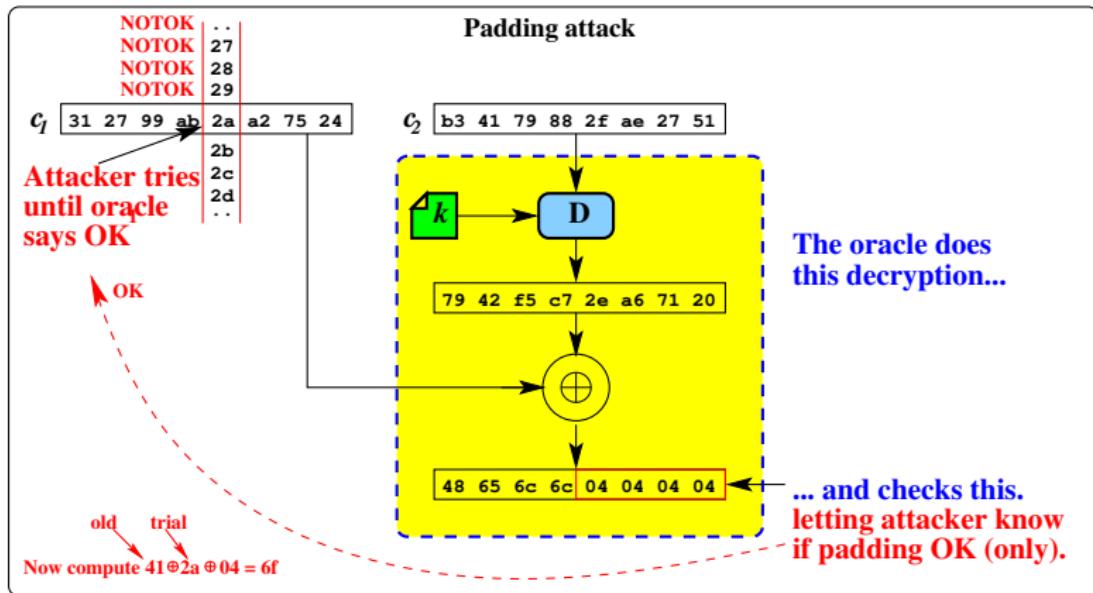


Since we know (now) the last two bytes (02 02), we can set the last two bytes to 03 03.

Now we try all the third-to-last values in  $c_1$  and uncover 20.

# Using the oracle in an attack...

Attacker discovers fourth-to-last byte...



Since we know (now) the last three bytes (20 02 02), we can set the last three bytes to 04 04 04.

Now we try all the fourth-to-last values in  $c_1$  and uncover 6f.

# Using the oracle in an attack...

## Attacker discovers all bytes...

The attacker keeps moving through the block, and has to try a maximum of 255 trials for each byte recovered, so this is an  $\mathcal{O}(n)$  attack, where  $n$  is the number of bytes in the block.

This attack can be avoided if...

- There is no oracle (i.e. the receiving code does not give any indication if the padding is correct or not)
- CFB is used instead.
- A different padding scheme is used...

That is it - a little aside into cracking that makes use of much of what we have learned.