CSc 466/566

Computer Security

3 : Physical Security

Version: 2014/09/10 08:46:56

Department of Computer Science
University of Arizona

collberg@gmail.com

Christian Collberg

Outline

Physical vs. Digital Interface

- We access computers

Physical vs. Digital Interface

- We access computers
  - over the network

Physical vs. Digital Interface

- We access computers
  - over the network
  - keyboard

Physical vs. Digital Interface

- We access computers
  - over the network
  - keyboard
  - other well-defined digital interfaces

Physical vs. Digital Interface

- We access computers
  - over the network
  - keyboard
  - other well-defined digital interfaces
- Right?

Physical vs. Digital Interface

- We access computers
  - over the network
  - keyboard
  - other well-defined digital interfaces
- Right?
- Or with a

Physical vs. Digital Interface

- We access computers
  - over the network
  - keyboard
  - other well-defined digital interfaces
- Right?
- Or with a
  - sledge hammer, a bottle of liquid nitrogen, ...

Physical vs. Digital Interface

- We access computers
  - over the network
  - keyboard
  - other well-defined digital interfaces
- Right?
- Or with a
  - sledge hammer, a bottle of liquid nitrogen, ...
- We need to protect access to computers physically as well as digitally.

Physical Security

Definition (physical security)

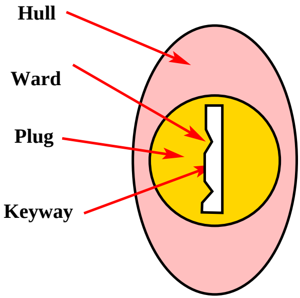
The use of physical measures to protect valuables, information, or access to restricted resources.

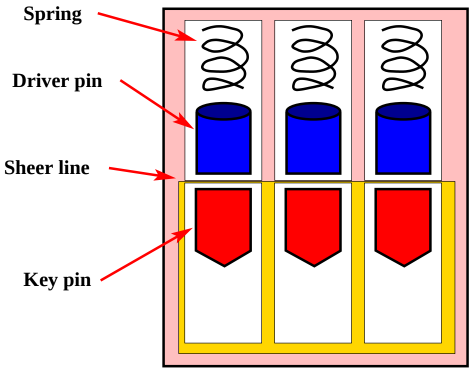
1. Location protection: protecting the location where hardware resides;
2. Physical intrusion detection: detecting intrusion into the location where hardware resides;
3. Hardware attacks: attacks against hard drives, CPUs, etc.;
4. Eavesdropping: attacks that monitor signals from or between computers;
5. Physical interface attack: exploiting weaknesses in a system's physical interface.

Outline

Locks and Safes: Terminology

- plug: the cylinder that contains the keyway and turns when the proper key is inserted
- keyway: where the key is inserted
- ward: sticks out of the sides of the keyway to restrict what keys will fit
- hull: the non-rotating part of the lock
- key pin: they pin that touches the key, also lifts the driver pin
- driver pin: This pin sits on top of the key pin
- sheer line: The space where the hull and plug meet
- spring: pushes the driver pin into the plug.


Hull
Ward
Plug
Keyway

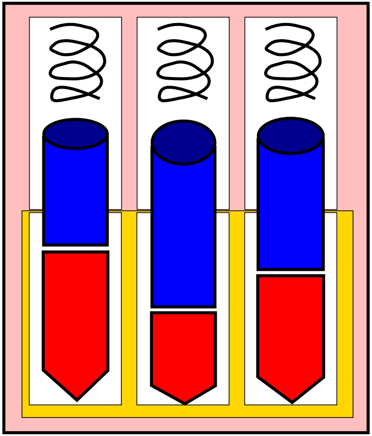

Spring
Driver pin
Sheer line
Key pin

Lock Layout

A lock consists of

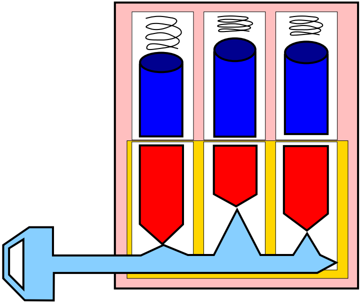
1. a hull and a plug, where the plug sits inside the hull such that rotating it opens the lock;
2. a keyway inside the plug that gives the key access to the pins;
3. a set of pins:
   - driver pins prevent the plug from rotating;
   - key pins allow the key to push the driver pins above the sheer line.

Locked Lock



- In a locked lock, the driver pins are stuck between the sheer line, stopping the plug from rotating.

Opened Lock



- When the proper key is inserted the key pins will push the driver pins above the sheer line allowing the plug to be rotated and the lock to be opened.
- An incorrect key will leave some of the driver pins stuck between the sheer line, stopping the plug from rotating.

Picking a lock: Tools of the Trade

- Terminology:
  - setting a pin: The act of trapping the driver pin above the sheer line even though the key pin is not holding it in place.
  - binding: scissoring (pinning) a pin between the plug and the hull.
- Lock picking requires two tools:
  - A pick for moving the pins
  - A tension wrench for moving the plug.

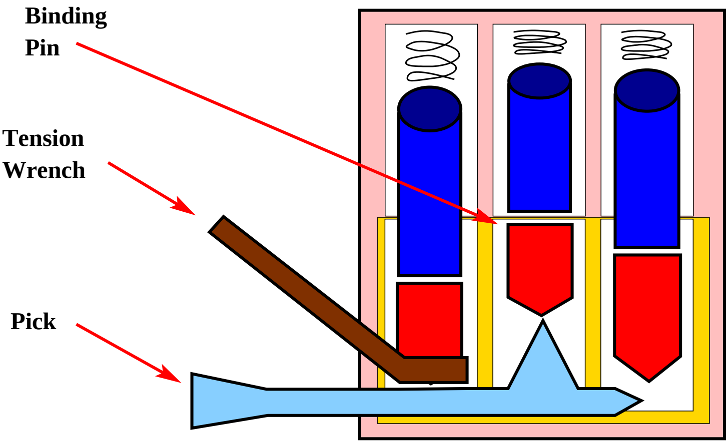
Lockpicks

- http://www.southord.com/Lock-Picking-Tools/Lock-Pick-Set-8-Piece-Metal-Handles-MPXS-08.html
- $29.95 Buy Now!

Technique

- The following technique is used to pick a lock one pin at a time:
  1. Apply a sheer force (torque from the tension wrench);
  2. Find the pin that is binding the most (the binding pin);
  3. Push that pin up until you feel it set at the sheer line;
  4. Go to step 2.

Technique



Technique: Scrubbing

- Scrubbing tries to set multiple pins each time the pick is inserted or removed from the keyway.
- The tension wrench is used to bind pins and then a pick is bounced along the pins.
- Technique:
  1. Insert a snake pick (designed to lift multiple pins at the same time) into the keyway;
  2. Move the pick back and forth in the keyway;
  3. Gradually increase the pressure on the pins;
  4. Gradually increase the torque from the tension wrench (to keep pins set);
  5. Pick remaining pins manually.

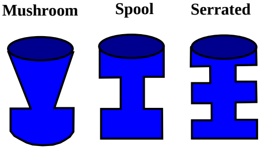
Demo

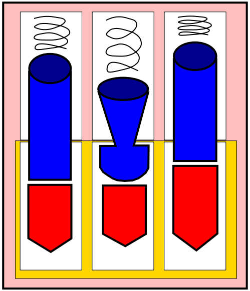
- Watch: http://www.youtube.com/watch?v=JZJe23UD8wU

Vibration Picking with Lockpicking Guns

- `http://www.lockpickshop.com/PKX-GUN.html`
- $74.95 Buy Now!
- Watch: `http://www.youtube.com/watch?v=UCBxqKnA8mo`
- You can do vibration picking manually as well, called lock bumping.

Countermeasures





- Security pins:
  - Special driver pins in an attempt to make lock picking harder.
  - These pins well cause a low false set.
  - Particularly damaging to vibration picking.
- Countermeasure to the countermeasure: Use less torque and more pressure with the pick.

Locks with Master Keys

- Certain locks can be opened with two different keys.
- Terminology:
  - Change key: the regular key for the lock.
  - Master key: Can also open other locks.
  - Grandmaster key: Can open any lock in the organization.
  - Control key: Can remove the entire cylinder, for rekeying.
- These locks add a spacer pin between the driver pin and the key pin.
- The master key pushes the spacer and driver pins above the sheer line.
- The change key only pushes the driver pin.

Assignment: Learn to Pick Locks!

- http://www.southord.com/Lock-Picking-Tools/Locksmith-School-In-A-Box-ST-23.html
- \$99.95 Buy Now!
- We have three of these, for you to check out and practice on.

Assignment: Learn to Pick Locks!

- http://www.southord.com/Lock-Picking-Tools/Practice-Lock-Cutaway-Visible-Locks-ST-34.html
- \$39.95 Buy Now!
- And we have three of these, too. . . .

In-Class Exercise: Goodrich & Tamassia C-2.3

- A group of n pirates has a treasure chest and one unique lock and key for each pirate.
- Using hardware that is probably already lying around their ship, they want to protect the chest so that any single pirate can open the chest using his lock and key.
- How do they set this up?

In-Class Exercise: Goodrich & Tamassia C-2.4

- A group of n red pirates and a group of n blue pirates have a shared treasure chest and one unique lock and key for each pirate.
- Using hardware that is probably already lying around their two ships, they want to protect the chest so that any pair of pirates, one red and one blue, can open the chest using their two locks and keys.
- No group of red or blue pirates can open the chest without having at least one pirate from the other group.
- How do they set this up?

In-Class Exercise: Goodrich & Tamassia C-2.5

- A group of four pirates has a treasure chest and one unique lock and key for each pirate.
- Using hardware that is probably already lying around their ship, they want to protect the chest so that any subset of three of these pirates can open the chest using their respective locks and keys, but no two pirates can.
- How do they set this up?

Outline

Means of Authentication

We identify someone by a combination of

1. something they have — smart card, radio key fob, . . .
2. something they know — password, mother's maiden name, first pet's name . . .
3. something they are — fingerprint, retina scan, . . .

- Here we'll look at: something physical you posses, or something you are (biometrics).

Barcodes

- Uses for grocery checkout, postage, etc.
- Easy to duplicate.
- On boarding passes:
  - Barcode holds internal unique identifier;
  - Hard to forge, since only airline knows $\mathrm{ID} \rightarrow \mathrm{passenger}$ mapping.

3597C052[BR]

9 54220762TA - 6590-8721B

9 0 7 8 5 9 8 6 >

00-0432786423-94-7

(42)78C052(54)BR

A53023-5X23-AB2300-96653

4 9 0 8 3   7 0 9 8 5

220762454

56700 AB78 76YD 00054000 5D7832 89000-0097 7YH48 800-7AC5432 905643 W9

7 052890735421 7342901715 45

123456789 123456789

ISBN 9543-76-23-8700-1

9 54220762 65908721

In-Class Exercise: Goodrich & Tamassia C-2.12

- The government gives the airlines a no-fly list of names of people not allowed to fly.
- Consider the following security measures for airline travel:
  1. Before entering the departure area of the airport, passengers go through a security check where they have to present a government-issued ID and a boarding pass.
  2. Before boarding a fight, passengers must present a boarding pass, which is scanned to verify the reservation.
- Show how someone who is on the no-fly list can manage to fly provided boarding passes can be printed online.
- Which additional security measures should be implemented in order to eliminate this vulnerability?

Magnetic Stripe Cards

- Developed in the late 60s.
- Debit cards, credit cards, drivers' licenses, ID cards, ...
- Three tracks, error correcting code (parity bit) to deal with worn magnetic stripes.

Track 1

Full name, account #, format, ...

79 characters, 6 bits+1 parity bit/character

Track 2

Account #, expiration date, issuing bank, ...

40 characters, 5 bits+1 parity bit/character

Track 3: Not often used

Magnetic Stripe Cards: Vulnerabilities

- Easy to read.
- Easy to reproduce.
- Some vendors use the card as a stored value card, storing money, points, transportation credits, etc. — cloning attack.

Magnetic Stripe Cards: Countermeasures

1. Embed hologram in the card.

Magnetic Stripe Cards: Countermeasures

1. Embed hologram in the card.
2. Customer signature.

Magnetic Stripe Cards: Countermeasures

1. Embed hologram in the card.
2. Customer signature.
3. PIN code.

Magnetic Stripe Cards: Countermeasures

1. Embed hologram in the card.
2. Customer signature.
3. PIN code.
4. Secret data formats (security-through-obscurity).

Magnetic Stripe Cards: Countermeasures

1. Embed hologram in the card.
2. Customer signature.
3. PIN code.
4. Secret data formats (security-through-obscurity).
5. Cryptographic signature algorithms to validate data intergrity.

In-Class Exercise: Goodrich & Tamassia C-2.11

- A bank wants to store the account number of its customers (an 8-digit number) in encrypted form on magnetic stripe ATM cards.
- We assume the account number is supposed to be secret.
- We assume the attacker can read the magnetic stripe.
- How secure are these methods:
  1. Store a cryptographic hash of the account number;
  2. Store the ciphertext of the account number encrypted with the bank's public key;
  3. Store the ciphertext of the account number encrypted with the bank's secret key using a symmetric cryptosystem.

Smart Cards

- Mass transit, prepaid phone cards, identification cards, *SIM cards*, pay-TV set-top boxes, credit cards.
- Disk encryption: smart card stores the key.
- Chip-and-pin: credit cards with smart card technology.
- Electronic wallet.
- Prepaid phone cards.

Smart Cards

- Trade-off between tamper-resistance and cost.
- Protected memory in which a secret can be stored.
- Cryptographic capabilities: generate and store public-key key-pairs, perform RSA encryption, compute SHA-1 hashing, ...
- Newer card types are contactless.

Smart Cards

- Gets power and clock from Card Acceptance Device (CAD).
- The CAD has no direct access to the internals of the card, including its memory.
- CAD and card communicate over 1-bit serial link.

Gemalto TOP DM GX4

- JavaCard virtual machine interpreter, 68KB of persistent RAM, 78KB EEPROM, 3DES/AES/RSA encryption, SHA-1 cryptographic hash, and asymmetric key pair generation.
- JavaCard specifies a subset of the Java language and standard libraries designed specifically for smart card programming, along with a virtual machine instruction set optimized for size.

the platform implements most advance security countermeasures enforcing protection of all sensitive data and function in the card. . . . includes multiple hardware and software countermeasure against various attacks:

- *Side channel attacks*
- *Invasive attacks*
- *Advanced fault attacks*
- *Other types of attack.*

Invasive vs. non-invasive attacks

- Invasive attack:
  1. expose the bare chip,

Invasive vs. non-invasive attacks

- Invasive attack:
  1. expose the bare chip,
  2. probe the surface to extract information

Invasive vs. non-invasive attacks

- Invasive attack:
  1. expose the bare chip,
  2. probe the surface to extract information
  3. poke the surface to modify the chip

Invasive vs. non-invasive attacks

- Invasive attack:
  1. expose the bare chip,
  2. probe the surface to extract information
  3. poke the surface to modify the chip
- Non-invasive attack:
  - monitor execution characteristics (power, radiation, execution time) etc.

Invasive vs. non-invasive attacks

- Invasive attack:
  1. expose the bare chip,
  2. probe the surface to extract information
  3. poke the surface to modify the chip
- Non-invasive attack:
  - monitor execution characteristics (power, radiation, execution time) etc.
  - watch normal operations or induce faults

Invasive vs. non-invasive attacks

- An invasive attack, by definition, destroys the card.
- You can use the secret code and data that you collect to clone a new card.
- Invasive attacks are useful when you know very little about the card.
- They may require sophisticated and expensive equipment.
- However, once you've gathered enough information about the card you may be able to use it to devise a non-invasive attack that's easier, cheaper, and faster to deploy.

Smart Cards — Invasive attacks

Chipworks will provide reverse engineering service for you (http://www.chipworks.com):

> *Chipworks can extract analog or digital circuits from semiconductor devices and deliver detailed easy-to-understand schematics that document a single functional block or all the circuits. ...We decapsulate the chip and analyze the die to locate the circuit blocks of interest. Then, using our Image Capture and Imaging System (ICIS) we generate mosiacs for each level of interconnect. Finally, advanced software and expertise is used to extract the circuits for analysis.*

Invasive attacks: Step 1 — Depackaging

1. Remove the chip from the card itself by heating and bending it.
2. Remove the epoxy resin around the chip by dipping it in 60°C fuming nitric acid.
3. Clean the chip by washing it with acetone in an ultrasonic bath.
4. Mount the exposed chip in a test package and connect its pads to the pins of the package.

Invasive attacks: Step 2 — Deprocessing

5. Use an optical microscope to take large high-resolution pictures of the chip surface.
6. Identify major architectural features (ROM, ALU, EEPROM, etc.) and/or lower-level features such as busses and gates.
7. Remove the top metal track layer by dipping the chip in hydrofluoric acid in an ultrasonic bath.
8. Repeat from 5, for each layer.

Invasive attacks: Step 3 — Reverse Engineering

- Reverse engineer the chip
- Analyze the information collected
- Understand the functional units of the chip

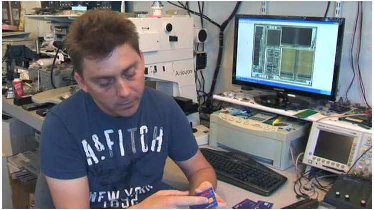
Invasive attacks: Step 4 — Microprobing

9. To allow the probe contact with the chip, use a laser cutter mounted on the microscope to remove (patches of) the *passivation layer* that covers the top-layer aluminum interconnect lines.
10. Record the activity on a few of the bus lines (as many as you have probes) as you go through a transaction with the card.
11. Repeat from 10 until you've collected the bus activity trace from all of the bus lines.

Invasive attacks: Summary

- Attacks get harder as features get smaller
- Rent a lab!
- Use your university lab!

Invasive attacks: Christopher Tarnovsky

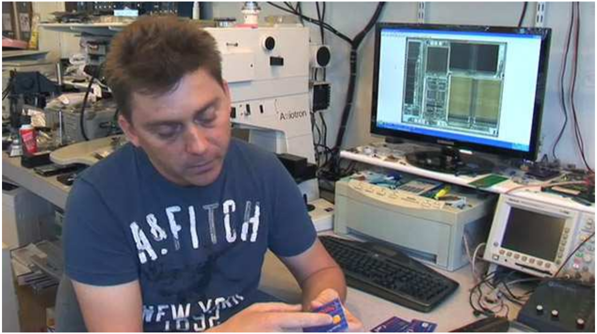


Dish Network is accusing News Corp . . . of hiring hacker Christopher Tarnovsky to break into Dish's network, steal the security codes, and use them to make pirated cards to flood the black market.

Tarnovsky admitted in court he was paid James Bond villain style, with $20,000 cash payments mailed from Canada hidden inside "electronic devices."

http://gizmodo.com/383753/news-corp-hires-hacker-to-break-into-dish-satellite-network-steal-security-cod

Invasive attacks: Christopher Tarnovsky



play

http://www.wired.com/politics/security/news/2008/05/tarnovsky?currentPage=all

Non-invasive attacks

Advantages over invasive attacks:

- No dangerous chemicals!
- Don't destroy the card!
- No expensive equipment!
- Once you have an effective attack against one particular card you can easily reuse it on another of the same model.

Non-invasive attacks

- Passive attack:
  - Watch what comes out of the chip
  - . . . , electromagnetic radiation, power consumption, execution time, . . .
- Active attack:
  - Feed carefully constructed data/power/clock/. . . to the chip,
  - *then* measure the chip's behavior.

Non-invasive attacks: Fault induction (glitch) attacks

- Methods:
  - generate a sharp voltage spike,
  - increase the clock frequency,
  - subject the chip to an electric field.
- Goal: Cause an error in the computation!
- Not every wrong instruction will cause an exploitable fault — use trial and error!

Non-invasive attacks: Fault induction (glitch) attacks

- This routine writes a region of memory to the I/O port:

```
void write(char* result, int length) {
    while (length > 0) {
        printf(*result);
        result++;
        length--;
    }
}
```

- Assume this routine is on the card.
- Goal: Force a fault in the boxed code, replacing it with any instruction that doesn't affect the `length` variable.
- Effect: The loop will cycle through all of memory, dumping it on the port!

Non-invasive attacks: Timing attacks

- Method:
  1. generate a large number of messages

     "please encrypt this file with your secret key";
  2. send them to the smart card;
  3. measure the time the operations take;
  4. deduce the key from the measurements.

Non-invasive attacks: Timing attacks

- This is a modular exponentiation routine that's used in many cryptographic operations, such as RSA encryption.
- x is the w bits long private key we want to recover.

```
s[0] = 1;
for(k=0; k<w; k++) {
    if (x[k] == 1)
        R[k] = (s[k]*y) mod n;
    else
        R[k] = s[k];
    s[k+1] = R[k]*R[k] mod n
}
return R[w-1];
```

Non-invasive attacks: Timing attacks

- Recover one bit at a time, starting with bit x_1:
  1. Construct a set of messages M_1 causing the boxed code to execute.

Non-invasive attacks: Timing attacks

- Recover one bit at a time, starting with bit x_1:
  1. Construct a set of messages M_1 causing the boxed code to execute.
  2. Construct a set of messages M_2 that make the code take the other branch.

Non-invasive attacks: Timing attacks

- Recover one bit at a time, starting with bit x_1:
  1. Construct a set of messages M_1 causing the boxed code to execute.
  2. Construct a set of messages M_2 that make the code take the other branch.
  3. Ask the smart card to encrypt all the messages, and record their time.

Non-invasive attacks: Timing attacks

- Recover one bit at a time, starting with bit x_1:
  1. Construct a set of messages M_1 causing the boxed code to execute.
  2. Construct a set of messages M_2 that make the code take the other branch.
  3. Ask the smart card to encrypt all the messages, and record their time.
  4. If the messages in M_1 take longer to encrypt than those in M_2, deduce that $x_1 = 1$, otherwise $x_1 = 0$.

Non-invasive attacks: Timing attacks

- Recover one bit at a time, starting with bit x_1:
  1. Construct a set of messages M_1 causing the boxed code to execute.
  2. Construct a set of messages M_2 that make the code take the other branch.
  3. Ask the smart card to encrypt all the messages, and record their time.
  4. If the messages in M_1 take longer to encrypt than those in M_2, deduce that $x_1 = 1$, otherwise $x_1 = 0$.
  5. Knowing x_1 continue to deduce x_2 in the same manner.

Non-invasive attacks: Timing attacks

- Recover one bit at a time, starting with bit x_1:
  1. Construct a set of messages M_1 causing the boxed code to execute.
  2. Construct a set of messages M_2 that make the code take the other branch.
  3. Ask the smart card to encrypt all the messages, and record their time.
  4. If the messages in M_1 take longer to encrypt than those in M_2, deduce that $x_1 = 1$, otherwise $x_1 = 0$.
  5. Knowing x_1 continue to deduce x_2 in the same manner.

Non-invasive attacks: Timing attacks

- Recover one bit at a time, starting with bit x_1:
  1. Construct a set of messages M_1 causing the boxed code to execute.
  2. Construct a set of messages M_2 that make the code take the other branch.
  3. Ask the smart card to encrypt all the messages, and record their time.
  4. If the messages in M_1 take longer to encrypt than those in M_2, deduce that $x_1 = 1$, otherwise $x_1 = 0$.
  5. Knowing x_1 continue to deduce x_2 in the same manner.
- Smaller difference in time between the two branches $\Rightarrow$ more samples.

Non-invasive attacks: Power analysis attacks

- Draw conclusions about the internal behavior of the chip from measurements of the power that it consumes.
- Different instructions consume different amounts of power
- Busses also draw power as bus lines change between 0 and 1: you can estimate the number of bits that changed on the bus by measuring the amount of power consumed.
- Easy attack to implement:
    1. Put a resistor on the chip's power supply line;

Non-invasive attacks: Power analysis attacks

- Draw conclusions about the internal behavior of the chip from measurements of the power that it consumes.
- Different instructions consume different amounts of power
- Busses also draw power as bus lines change between 0 and 1: you can estimate the number of bits that changed on the bus by measuring the amount of power consumed.
- Easy attack to implement:
    1. Put a resistor on the chip's power supply line;
    2. Put a high-resolution high-sampling-frequency volt meter over the resistor;

Non-invasive attacks: Power analysis attacks

- Draw conclusions about the internal behavior of the chip from measurements of the power that it consumes.
- Different instructions consume different amounts of power
- Busses also draw power as bus lines change between 0 and 1: you can estimate the number of bits that changed on the bus by measuring the amount of power consumed.
- Easy attack to implement:
  1. Put a resistor on the chip's power supply line;
  2. Put a high-resolution high-sampling-frequency volt meter over the resistor;
  3. Use a computer to store and analyze the current traces.

Non-invasive attacks: Power analysis attacks

- Draw conclusions about the internal behavior of the chip from measurements of the power that it consumes.
- Different instructions consume different amounts of power
- Busses also draw power as bus lines change between 0 and 1: you can estimate the number of bits that changed on the bus by measuring the amount of power consumed.
- Easy attack to implement:
  1. Put a resistor on the chip's power supply line;
  2. Put a high-resolution high-sampling-frequency volt meter over the resistor;
  3. Use a computer to store and analyze the current traces.

Non-invasive attacks: Power analysis attacks

- Draw conclusions about the internal behavior of the chip from measurements of the power that it consumes.
- Different instructions consume different amounts of power
- Busses also draw power as bus lines change between 0 and 1: you can estimate the number of bits that changed on the bus by measuring the amount of power consumed.
- Easy attack to implement:
  1. Put a resistor on the chip's power supply line;
  2. Put a high-resolution high-sampling-frequency volt meter over the resistor;
  3. Use a computer to store and analyze the current traces.
- Counter noise in the measurements by averaging over a large number of transactions.

Non-invasive attacks: Power analysis attacks

- There are two kinds of power analyses:
  1. Simple Power Analysis (SPA),
  2. Differential Power Analysis (DPA) uses statistical techniques.
- Paul Kocher et al. report that DPA allowed them to:

  > *extract keys from almost 50 different products in a variety of physical form factors.*

Non-invasive attacks: Countermeasures

1. Randomization: generate an internal clock signal by inserting random delays in the external one.

Non-invasive attacks: Countermeasures

1. Randomization: generate an internal clock signal by inserting random delays in the external one.
2. Obfuscation: insert bogus instructions in conditional branches.

Non-invasive attacks: Countermeasures

1. Randomization: generate an internal clock signal by inserting random delays in the external one.
2. Obfuscation: insert bogus instructions in conditional branches.
3. Interleave multiple threads of control (difficult on smart cards with limited computational resources).

Non-invasive attacks: Countermeasures

1. Randomization: generate an internal clock signal by inserting random delays in the external one.
2. Obfuscation: insert bogus instructions in conditional branches.
3. Interleave multiple threads of control (difficult on smart cards with limited computational resources).
4. Environmental sensors: Detect if an attacker lowers the clock signal in order to be able to more easily monitor the computations.

Non-invasive attacks: Countermeasures

1. Randomization: generate an internal clock signal by inserting random delays in the external one.
2. Obfuscation: insert bogus instructions in conditional branches.
3. Interleave multiple threads of control (difficult on smart cards with limited computational resources).
4. Environmental sensors: Detect if an attacker lowers the clock signal in order to be able to more easily monitor the computations.
5. Anticipate being attacked:

Non-invasive attacks: Countermeasures

1. Randomization: generate an internal clock signal by inserting random delays in the external one.
2. Obfuscation: insert bogus instructions in conditional branches.
3. Interleave multiple threads of control (difficult on smart cards with limited computational resources).
4. Environmental sensors: Detect if an attacker lowers the clock signal in order to be able to more easily monitor the computations.
5. Anticipate being attacked:
   1. Smart Cards should be one part of a complete security architecture.

Non-invasive attacks: Countermeasures

1. Randomization: generate an internal clock signal by inserting random delays in the external one.
2. Obfuscation: insert bogus instructions in conditional branches.
3. Interleave multiple threads of control (difficult on smart cards with limited computational resources).
4. Environmental sensors: Detect if an attacker lowers the clock signal in order to be able to more easily monitor the computations.
5. Anticipate being attacked:
    1. Smart Cards should be one part of a complete security architecture.
    2. Architect your systems to detect anomalies and to minimize losses.

Non-invasive attacks: Countermeasures

1. Randomization: generate an internal clock signal by inserting random delays in the external one.
2. Obfuscation: insert bogus instructions in conditional branches.
3. Interleave multiple threads of control (difficult on smart cards with limited computational resources).
4. Environmental sensors: Detect if an attacker lowers the clock signal in order to be able to more easily monitor the computations.
5. Anticipate being attacked:
    1. Smart Cards should be one part of a complete security architecture.
    2. Architect your systems to detect anomalies and to minimize losses.
    3. Design your system with upgradable security.

SIM Cards

- Subscriber Identity Module Card.
- Issued by the network provider.
- Contains personal information allowing the user to authenticate themselves to the network.

IC−CID: integrated circuit card ID

Used for hardware identification

18 digits + 1 check digit

IMSI: Int’l mobile subscriber ID

Owner’s country, network, personal ID

64 bits

PUK: Personal unblocking key

If the user forgets his 4−digit PIN

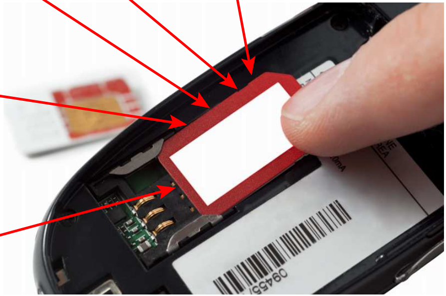
8 digits

Contact list

Secret Key

Authenticates the phone to the network

128 bits



SIM Cards: ICC-ID Example

89	91	10	1200	00	320451	0
Telecom ID	country code	network code	MM/YY of manu-facturing	switch config. code	SIM number	check digit

- The check digit is calculated using the Luhn Sum algorithm.

SIM Cards: IMSI Example

310	150	123456789
USA Mobile Country Code (MCC)	AT&T Mobile Network Code (MNC)	Mobile Subscription Identification Number (MSIN)

- There are some differences between different countries.

SIM Cards: Luhn Sum

```
double :: [Int] -> [Int]
double [] = []
double (x:xs) = 2*x : double xs

everyOther :: [Int] -> [Int]
everyOther [] = []
everyOther [x] = [x]
everyOther (x:y:xs) = x : everyOther xs

sumDigits :: Int -> Int
sumDigits n = n `mod` 10 + n `div` 10
```

SIM Cards: Luhn Sum. . .

```haskell
sumDigitsList :: [Int] -> Int
sumDigitsList [] = 0
sumDigitsList (x:xs) = sumDigits x +
                       sumDigitsList xs

luhnSum :: [Int] -> Int
luhnSum xs = sumDigitsList
               (double (everyOther
               (tail (reverse xs)))
             ++ (everyOther (reverse xs)))
```

GSM Challenge-Response Protocol

- ID = IMSI (the phone's ID); K = 128-bit secret key; C = 128-bit random challenge; A3, A5, A8 = secret encryption algorithms.
- Protocol:
  1. The phone sends ID to the base station;

GSM Challenge-Response Protocol

- ID $=$ IMSI (the phone's ID); K $=$ 128-bit secret key; C $=$ 128-bit random challenge; A3, A5, A8 $=$ secret encryption algorithms.
- Protocol:
  1. The phone sends ID to the base station;
  2. The base station generates and sends C to the phone;

GSM Challenge-Response Protocol

- ID = IMSI (the phone's ID); K = 128-bit secret key; C = 128-bit random challenge; A3, A5, A8 = secret encryption algorithms.
- Protocol:
  1. The phone sends ID to the base station;
  2. The base station generates and sends C to the phone;
  3. The phone sends $V = E_K^{\mathrm{A3}}(C)$ to the base station;

GSM Challenge-Response Protocol

- ID = IMSI (the phone's ID); K = 128-bit secret key; C = 128-bit random challenge; A3, A5, A8 = secret encryption algorithms.
- Protocol:
  1. The phone sends ID to the base station;
  2. The base station generates and sends C to the phone;
  3. The phone sends $V = E_K^{\mathrm{A3}}(C)$ to the base station;
  4. The base station looks up ID's key K in its database;

GSM Challenge-Response Protocol

- ID = IMSI (the phone's ID); K = 128-bit secret key; C = 128-bit random challenge; A3, A5, A8 = secret encryption algorithms.
- Protocol:
  1. The phone sends ID to the base station;
  2. The base station generates and sends C to the phone;
  3. The phone sends $V = E_K^{\mathrm{A3}}(C)$ to the base station;
  4. The base station looks up ID's key K in its database;
  5. The base station compares $V \stackrel{?}{=} E_K^{\mathrm{A3}}(C)$. If they are the same, the phone is authenticated;

GSM Challenge-Response Protocol

- ID = IMSI (the phone's ID); K = 128-bit secret key; C = 128-bit random challenge; A3, A5, A8 = secret encryption algorithms.
- Protocol:
  1. The phone sends ID to the base station;
  2. The base station generates and sends C to the phone;
  3. The phone sends $V = E_K^{\mathrm{A3}}(C)$ to the base station;
  4. The base station looks up ID's key K in its database;
  5. The base station compares $V \stackrel{?}{=} E_K^{\mathrm{A3}}(C)$. If they are the same, the phone is authenticated;
  6. The phone and base station both compute a session key $K_{\mathrm{session}} = E_K^{\mathrm{A8}}(C)$;

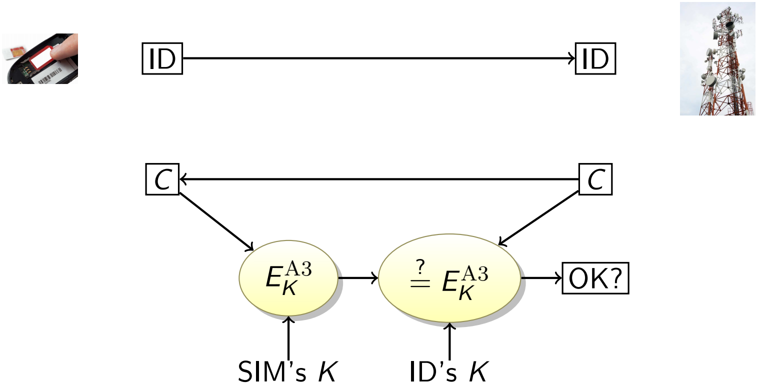
GSM Challenge-Response Protocol

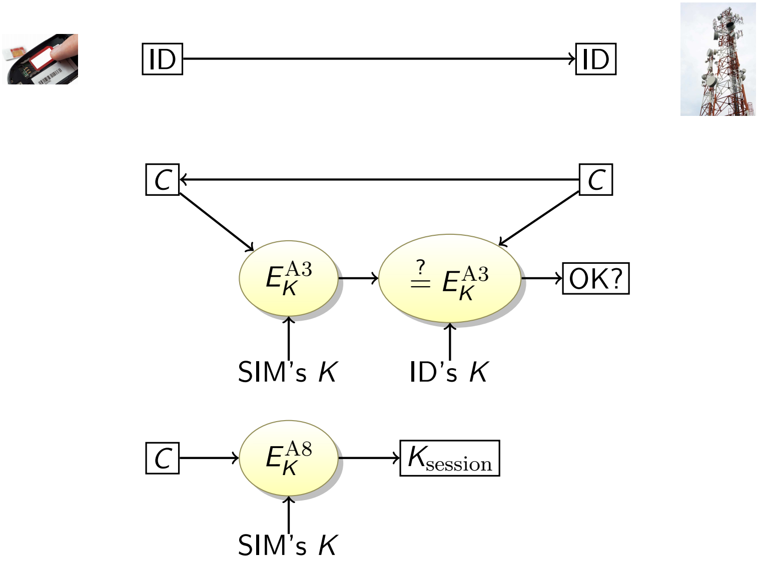
- ID = IMSI (the phone's ID); K = 128-bit secret key; C = 128-bit random challenge; A3, A5, A8 = secret encryption algorithms.
- Protocol:
  1. The phone sends ID to the base station;
  2. The base station generates and sends C to the phone;
  3. The phone sends $V = E_K^{\mathrm{A3}}(C)$ to the base station;
  4. The base station looks up ID's key K in its database;
  5. The base station compares $V \stackrel{?}{=} E_K^{\mathrm{A3}}(C)$. If they are the same, the phone is authenticated;
  6. The phone and base station both compute a session key $K_{\mathrm{session}} = E_K^{\mathrm{A8}}(C)$;
  7. The phone uses A5 to encrypt data.

$\boxed{\text{ID}} \longrightarrow \boxed{\text{ID}}$

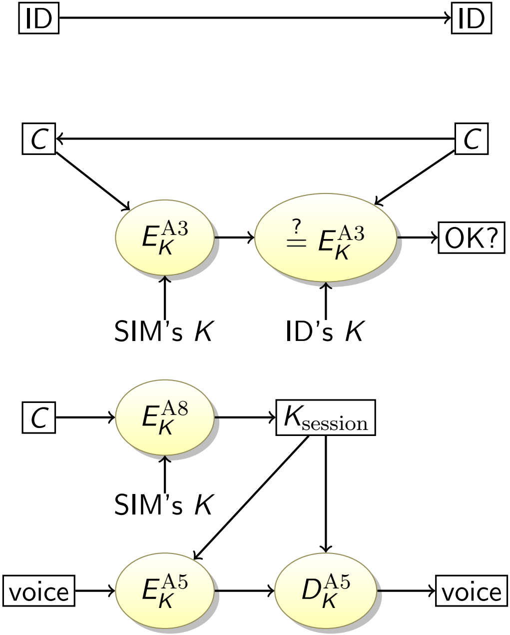
$$\boxed{\mathrm{ID}} \longrightarrow \boxed{\mathrm{ID}}$$

$$\boxed{C} \longleftarrow \boxed{C}$$


ID
ID
C
C
E_K^{A3}
$\stackrel{?}{=} E_K^{\mathrm{A3}}$
OK?
SIM's K
ID's K



ID
ID
C
C
E_K^{A3}
$\stackrel{?}{=} E_K^{\mathrm{A3}}$
OK?
SIM's K
ID's K
C
E_K^{A8}
K_{session}
SIM's K


ID
ID
C
C
E_K^{A3}
$\stackrel{?}{=} E_K^{\mathrm{A3}}$
OK?
SIM's K
ID's K
C
E_K^{A8}
K_{session}
SIM's K
voice
E_K^{A5}
D_K^{A5}
voice

GSM Vulnerabilities

- A3, A5, A8 were chosen over standard cryptographic algorithms for efficiency reasons.
- The A3/A8 were reverse engineered and found to be insecure:
  - Given certain input (over the air!) the attacker can discover the card's key.
  - Given the key, a new SIM card can be cloned.
- A5 implementations have also had flaws, allowing eavesdropping on conversations.

RFIDs

- RFID = Radio Frequency IDentification.
- IC for storing information + coiled antenna.
- Many RFIDs are passive (no battery).
- Range: a few centimeters to a few meters.
- Uses: tracking products, theft detection, track animals.
- In 2004 night clubs in Barcelona implanted RFID chips under the skin of their VIP customers, to identify them and allow them to pay for drinks. http://news.bbc.co.uk/2/hi/technology/3697940.stm.
- Harder to clone than barcodes.

RFID Vulnerabilities

- Privacy issues: RFID tags can be read from a distance.
- Important to protect against unauthorized readers.

Remote Automobile Entry

- The RFID and the car lock have the same pseudo-random number generator (PSRNG).
- Both generate the same sequence of random numbers.
- What happens if the devices become desynchronized?

Remote Automobile Entry: Desynchronization

$$\text{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

Remote Automobile Entry: Desynchronization

$$\text{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

$$42 \longrightarrow 42$$

Remote Automobile Entry: Desynchronization

$$\text{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

$$42 \longrightarrow 42$$

$$99 \longrightarrow 99$$

Remote Automobile Entry: Desynchronization

$$\mathrm{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

$$42 \longrightarrow 42$$

$$99 \longrightarrow 99$$

$$27 \longrightarrow 27$$

Remote Automobile Entry: Desynchronization

$$\text{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

$$42 \longrightarrow 42$$

$$99 \longrightarrow 99$$

$$27 \longrightarrow 27$$

$$63$$

Remote Automobile Entry: Desynchronization

$$\text{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

$42 \longrightarrow 42$

$99 \longrightarrow 99$

$27 \longrightarrow 27$

63

$82 \longrightarrow 63$

Remote Automobile Entry: Hopping (Rolling) Codes

- The car lock keeps track of the next 256 random numbers, and skips to the next one that matches.
- If the key-fob is pressed more the 256 times without connecting to the car: factory reset!

Remote Automobile Entry: Hopping (Rolling) Codes

$$\text{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

Remote Automobile Entry: Hopping (Rolling) Codes

$$\text{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

42 ⟶ 42

next=42,99,27,63,82

Remote Automobile Entry: Hopping (Rolling) Codes

$$\text{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

$42 \longrightarrow 42$ $\boxed{\text{next}=42,99,27,63,82}$

$99 \longrightarrow 99$ $\boxed{\text{next}=99,27,63,82,32}$

Remote Automobile Entry: Hopping (Rolling) Codes

$$\text{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

$42 \longrightarrow 42$	next= 42 ,99,27,63,82
$99 \longrightarrow 99$	next= 99 ,27,63,82,32
$27 \longrightarrow 27$	next= 27 ,63,82,32,66

Remote Automobile Entry: Hopping (Rolling) Codes

$$\text{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

42 ⟶ 42	next= 42 ,99,27,63,82
99 ⟶ 99	next= 99 ,27,63,82,32
27 ⟶ 27	next= 27 ,63,82,32,66
63	

Remote Automobile Entry: Hopping (Rolling) Codes

$$\text{PRNG} = \langle 42, 99, 27, 63, 82, 32, 66, 87, 11, 24, \ldots \rangle$$

$42 \longrightarrow 42$ $\boxed{\text{next}=42,99,27,63,82}$

$99 \longrightarrow 99$ $\boxed{\text{next}=99,27,63,82,32}$

$27 \longrightarrow 27$ $\boxed{\text{next}=27,63,82,32,66}$

63

$82 \longrightarrow 82$ $\boxed{\text{next}=\cancel{63},82,32,66,87}$

Remote Automobile Entry: Replay Attack

Remote Automobile Entry: Replay Attack

42 ⟶ next=42

- Replay attack: jam the radio signal, collect the PRNG sequence, play it back to the car.

Remote Automobile Entry: Replay Attack

42 ⟶ next=42

99 ⟶ next=42,99

- Replay attack: jam the radio signal, collect the PRNG sequence, play it back to the car.

Remote Automobile Entry: Replay Attack

$42 \longrightarrow \text{next}=42$

$99 \longrightarrow \text{next}=42,99$

$27 \longrightarrow \text{next}=42,99,27$

- Replay attack: jam the radio signal, collect the PRNG sequence, play it back to the car.

Remote Automobile Entry: Replay Attack

42 ⟶ next=42

99 ⟶ next=42,99

27 ⟶ next=42,99,27 ⟶

- Replay attack: jam the radio signal, collect the PRNG sequence, play it back to the car.

Remote Automobile Entry: KeeLoq

- KeeLoq is a proprietary code hopping algorithm using a 32-bit key.

Remote Automobile Entry: KeeLoq

- KeeLoq is a proprietary code hopping algorithm using a 32-bit key.
- Reduced key-space attack:

Remote Automobile Entry: KeeLoq

- KeeLoq is a proprietary code hopping algorithm using a 32-bit key.
- Reduced key-space attack:
  1. Note: some car models share common key bits;

Remote Automobile Entry: KeeLoq

- KeeLoq is a proprietary code hopping algorithm using a 32-bit key.
- Reduced key-space attack:
  1. Note: some car models share common key bits;
  2. collect many transmissions;

Remote Automobile Entry: KeeLoq

- KeeLoq is a proprietary code hopping algorithm using a 32-bit key.
- Reduced key-space attack:
  1. Note: some car models share common key bits;
  2. collect many transmissions;
  3. calculate for days.

Remote Automobile Entry: KeeLoq

- KeeLoq is a proprietary code hopping algorithm using a 32-bit key.
- Reduced key-space attack:
  1. Note: some car models share common key bits;
  2. collect many transmissions;
  3. calculate for days.
- Side channel attack:

Remote Automobile Entry: KeeLoq

- KeeLoq is a proprietary code hopping algorithm using a 32-bit key.
- Reduced key-space attack:
  1. Note: some car models share common key bits;
  2. collect many transmissions;
  3. calculate for days.
- Side channel attack:
  1. Use power analysis to extract the manufacturer's (e.g. Chrysler's) "master key" from an encoder;

Remote Automobile Entry: KeeLoq

- KeeLoq is a proprietary code hopping algorithm using a 32-bit key.
- Reduced key-space attack:
  1. Note: some car models share common key bits;
  2. collect many transmissions;
  3. calculate for days.
- Side channel attack:
  1. Use power analysis to extract the manufacturer's (e.g. Chrysler's) "master key" from an encoder;
  2. intercept two messages from any encoder (up to 100 meters);

Remote Automobile Entry: KeeLoq

- KeeLoq is a proprietary code hopping algorithm using a 32-bit key.
- Reduced key-space attack:
  1. Note: some car models share common key bits;
  2. collect many transmissions;
  3. calculate for days.
- Side channel attack:
  1. Use power analysis to extract the manufacturer's (e.g. Chrysler's) "master key" from an encoder;
  2. intercept two messages from any encoder (up to 100 meters);
  3. clone the encoder!

Remote Automobile Entry: KeeLoq

- KeeLoq is a proprietary code hopping algorithm using a 32-bit key.
- Reduced key-space attack:
  1. Note: some car models share common key bits;
  2. collect many transmissions;
  3. calculate for days.
- Side channel attack:
  1. Use power analysis to extract the manufacturer's (e.g. Chrysler's) "master key" from an encoder;
  2. intercept two messages from any encoder (up to 100 meters);
  3. clone the encoder!
- Newer designs use longer keys.

RFID Passports (E-Passports)

- Since 2006, US passport have RFID tags, containing personal information + a digital picture.
- Skimming: With special equipment you can read the the passport from 10m.
- Countermeasures to skimming:
  1. A thin metal lining.
  2. To read the RFID, a PIN (printed on the passport data page) has to be entered into the reader.
  3. The communication is encrypted.

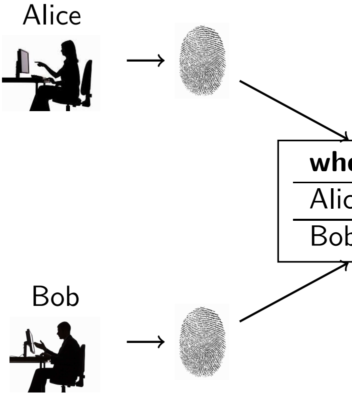
Biometrics

Definition (biometric)

Any measure used to uniquely identify a person based on biological or physiological traits.

- Biometric verification — biometrics supplement other means of identification (smartcard, etc.).
- Biometric identification — biometrics is the only means of identification.

Biometrics: Collecting Reference Vectors



who	reference vector
Alice	$\langle 5, 7, 3, 1, 6, 8 \rangle$
Bob	$\langle 3, 1, 5, 1, 6, 9 \rangle$

- For every user, extract a *reference vector* from their biometric measurement.

Biometrics: Matching Feature Vectors

Alice

 $\longrightarrow$  $\longrightarrow \boxed{\langle 4, 7, 3, 2, 6, 8 \rangle}$

- Extract a *feature vector* from the biometric measurement and do a fuzzy match against stored reference vectors.

Biometrics: Matching Feature Vectors

Alice

$\langle 4, 7, 3, 2, 6, 8 \rangle$

who	**reference vector**
Alice	$\langle 5, 7, 3, 1, 6, 8 \rangle$
Bob	$\langle 3, 1, 5, 1, 6, 9 \rangle$

match?

- Extract a *feature vector* from the biometric measurement and do a fuzzy match against stored reference vectors.

Biometrics: Matching Feature Vectors

Alice

 $\longrightarrow$  $\longrightarrow$ $\langle 4, 7, 3, 2, 6, 8 \rangle$

who	**reference vector**
Alice	$\langle 5, 7, 3, 1, 6, 8 \rangle$
Bob	$\langle 3, 1, 5, 1, 6, 9 \rangle$

match?

Alice/Bob/neither

- Extract a *feature vector* from the biometric measurement and do a fuzzy match against stored reference vectors.

Biometrics: Which Features?

- Fingerprints:
  - features: ridges, line splits, ...
  - collectability: easy
  - distinctiveness: high
  - permanence: change slightly over time
  - spoofability: gummy bears!

Biometrics: Which Features?

- Fingerprints:
  - features: ridges, line splits, ...
  - collectability: easy
  - distinctiveness: high
  - permanence: change slightly over time
  - spoofability: gummy bears!
- Voice recognition:
  - collectability: easy
  - distinctiveness: low
  - permanence: changes from year to year
  - spoofability: tape recorders!

Biometrics: Which Features?

- Fingerprints:
  - features: ridges, line splits, ...
  - collectability: easy
  - distinctiveness: high
  - permanence: change slightly over time
  - spoofability: gummy bears!
- Voice recognition:
  - collectability: easy
  - distinctiveness: low
  - permanence: changes from year to year
  - spoofability: tape recorders!
- Face recognition:
  - features: ridge of eyebrows, edges of mouth, tip of nose, ...
  - collectability: easy
  - permanence: facial hair, ...

Biometrics: Which Features?

- Fingerprints:
  - features: ridges, line splits, ...
  - collectability: easy
  - distinctiveness: high
  - permanence: change slightly over time
  - spoofability: gummy bears!
- Voice recognition:
  - collectability: easy
  - distinctiveness: low
  - permanence: changes from year to year
  - spoofability: tape recorders!
- Face recognition:
  - features: ridge of eyebrows, edges of mouth, tip of nose, ...
  - collectability: easy
  - permanence: facial hair, ...
- Eye scanning:
  - Retinal scan: uncomfortable lighting of retina
  - Iris scan: photograph of the surface

Biometrics: Privacy Concerns

- Biometric data is the same over a lifetime.
- Must not be compromised!
- Just store and compare cryptographic hashes!

$$h(\text{feature vector}) \stackrel{?}{=} h(\text{reference vector})$$

  Uh, no. We need to do approximate matching.
- AMAC — Approximate Message Authentication Codes:
  - Can easily determine similarity between two AMACs;
  - Given $\mathrm{AMAC}(M)$ it's hard to find a message M' such that $\mathrm{AMAC}(M') \approx \mathrm{AMAC}(M)$.

Outline

Direct Attacks Against Computers

- What kind of damage can an adversary cause if
  1. he has direct physical access to it?
  2. he is in close physical proximity to it?
- It is usually assumed that the user of a computing system is trusted — but the reality is often different!

Eavesdropping

Definition (Eavesdropping)

Secretly listening in on another person's conversation.

- Not really a "computer security" issue — we need to protect the environment in which the system is used.
- Passive wiretapping: monitoring or eavesdropping on communication.
- Active wiretapping: modifying or creating bogus communcation.

Eavesdropping: Shoulder Surfing

- Shoulder surfing:
  - installing small hidden cameras,
  - watch with binoculars through a window,
  - ...
- Countermeasures:
  - ATM machine displays have limited viewing angle,
  - ATM keypads shields the keypad from view,
  - Alter the physical location of the keypad keys after each keypress.

Eavesdropping: Wiretapping

- Coaxial cable, twisted pair:
  - measure the leaked electrical impulses
  - cut cable, splice in secondary one
- Ethernet cable:
  - briefly disconnect, insert passive listening device
- Fiber optic cable:
  - bend the cable, read the leaked light with an optical sensor
  - cut the fiber, reconnect it with an 80/20 splitter (80% goes through, 20% is used to monitor) in line ($100).
- Microwave/satellite communication:
  - an attacker close to receiver can read the communication

Eavesdropping: Countermeasures to Wiretapping

- Countermeasures:
  - Detect brief disconnect of cables
  - Detect drop in signal strength
  - End-to-end encryption.
- Countermeasures to the countermeasures:
  - Reboost the signal to make up for signal loss
  - Perform the attack at night when it is less likely to be detected.

Eavesdropping: Monitoring Emissions

- Electromagnetic radiation:
  - Monitor CRT displays
- Optical emissions:
  - CRT displays emit light pulses that can be monitored with a photosensor, and the screen image can be reconstructed.
- Acoustic emissions:
  - Listening to typing can reconstruct 79% of keystrokes.
  - Listening to a CPU can reveal the instructions it executes.

Eavesdropping: Hardware Keyloggers

- USB-to-USB connector, installed between keyboard and computer.
- Logs passwords to flash memory.
- Attacker can retrieve the logger or data can be transmitted wirelessly.
- Could capture BIOS passwords giving full control over the machine.

Hardware Keyloggers: KeyGrabber Wi-Fi Premium

This wireless keylogger is packed with state-of-the-art electronics: two powerful processors, a full TCP/IP stack, a WLAN transceiver, and 2 Gigabytes of memory. How does it work? Besides standard PS/2 and USB keylogger functionality, it features remote access over the Internet. This wireless keylogger will connect to a local Wi-Fi Access Point, and send E-mails containing recorded keystroke data. You can also connect to the keylogger at any time over TCP/IP and view the captured log. All this in a device less than 2 inches (5 cm) long!

Hardware Keyloggers: KeyGrabber Wi-Fi Premium...

- Applications:
  - Observe WWW, E-mail & chat usage by children and employees
  - Monitor employee productivity
  - Protect your child from on-line hazards and predators
- $148.99 

Hardware Keyloggers: KeyGrabber Wi-Fi Premium...

- Features:
  - Background connection to the Internet over a local Access Point
  - Automatic E-mail reports with recorded keyboard data
  - On-demand access at any time through TCP/IP
  - Support for WEP, WPA, and WPA-2 encryption
  - 2 Gigabytes of internal memory in all versions
  - No software or drivers required, Windows, Linux, and Mac compatible
  - Ultra compact and discrete, less than 2 inches (5 cm) long
  - Internal clock and battery with over 7 years lifetime guaranteed!

Hardware Keyloggers: KeyGrabber Wi-Fi Premium...

Is this legal?

> *Technically speaking, you should contact a lawyer to get detailed information about the local laws, and the application for which you intend to use this device for. Generally it's permitted to monitor your own computer, meaning you can watch what your kids and family are doing on the computer. If you want to monitor your employees, or perform any other type of surveillance, you should display a clear notice about this fact. It is obviously NOT LEGAL to use this device for any type spying, or stealing confidential data.*

- http://www.keelog.com/wifi_hardware_keylogger.html

TEMPEST

Definition (TEMPEST)

U.S. government standards for limiting electromagnetic intelligence-bearing signals from computing equipment.

- NATO SDIP-27 zones of protection:
  1. Level A: almost immediate access (neighbour room, 1 m distance).
  2. Level B: 20 m distance (or similar level of building material attenuation).
  3. Level C: 100 m distance (or equivalent attenuation).
- Countermeasures:
  - Block the emissions
  - Modify the emissions.

TEMPEST: Emanation Blockage

- Block visible light:
  - Windowless room
- Block acoustic emanations:
  - Line room with sound-dampening materials
- Block electromagnetic radiation:
  - Line room with copper mesh with holes smaller than the wavelength we want to block (Faraday Cage).

TEMPEST: Emanation Masking

- Broadcast random noice signals so that the information-carrying signals are lost in the noice.

Live CDs

Definition (Live CD)

A bootable computer operating system stored on external media (CD, DVD, USB drive) allowing a computer to be booted without a hard disk drive.

- An attacker can
  1. boot a computer from a Live CD bypassing the native operating system,
  2. bypass any authentication mechanisms,
  3. read and modify the hard disk data.
- Countermeasures:
  - Install BIOS passwords, so the computer can't be booted without authentication.
  - Hard drive password.
  - Hard drive encryption.

Computer Forensics

Definition (Computer Forensics)

Identifying, preserving, recovering, analyzing and presenting facts and opinions about the information found on digital storage media, to be used in legal proceedings.

- Forensic techniques can be used by attackers to extract information from computer equipment.
- Recover "deleted" files: most OSs only remove meta data, don't overwrite the file itself.
- Overwritten files can be recovered: magnetic traces may remain.
- Countermeasures:
  - overwrite files with multiple passes of random data
  - physically destroy the disk.

Computer Forensics: Cold Boot Attack

- Cold boot attack:
  1. Freeze DRAM on running computer
  2. power off computer
  3. boot from Live CD
  4. extract disk encryption key from RAM
- Countermeasures:
  - Don't store encryption keys in cleartext in RAM.

Outline

Readings and References

- Chapter 2 in *Introduction to Computer Security*, by Goodrich and Tamassia.
- Marshall Brain and Tom Harris, *How Lock Picking Works*,
  http://home.howstuffworks.com/home-improvement/household-safety/security/lock-picking2.htm
- Ted the Tool, *MIT Guide to Lock Picking*,
  http://www.lysator.liu.se/mit-guide/MITLockGuide.pdf

Acknowledgments

Material and exercises have also been collected from these sources:

1. Christian Collberg, Jasvir Nagra, *Surreptitious Software, Obfuscation, Watermarking, and Tamperproofing for Software Protection*,
   http://www.amazon.com/Surreptitious-Software-Obfuscation-Watermarking-Tamperproofing/dp/0321549252
2. Tom Olzak, *Protect your network against fiber hacks*,
   http://www.techrepublic.com/blog/security/protect-your-network-against-fiber-hacks/222
3. Bruce Schneier,
   http://www.schneier.com/blog/archives/2007/09/eavesdropping_o_1.html