UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Security II

2016-2017
CST Part II
Lent 2017

Dr Frank Stajano
Reader in Security and Privacy
Head, Academic Centre of Excellence in Cyber Security Research

Based on a deck of slides originally by Professor Ross Anderson, but blame me for the changes, including any new bugs

Overall Course Structure

WGB, LT2, M-W-F, 1200-1300

Security Engineering

- **Dr Frank Stajano** (x6): Security, human factors and psychology. Security policies. Passwords. Physical security.
  2017-01-... 20F, 23M, 25W, 27F, 30M, 2017-02-01W
- **Dr Richard Clayton** (x1): Security economics
  2017-02-03F
- **Dr Steven Murdoch** (x1): Anonymity and censorship resistance
  2017-02-06M

Cryptography

- **Dr Markus Kuhn** (x8): Secure hash functions + applications. Key distribution problem. Number theory. Discrete logarithm problem. Trapdoor permutations. Digital signatures.
  2017-02-... 08W, 10F, 13M, 15W, 17F, 20M, 22W, 24F

Introduction

Aims

Give you a thorough understanding of security engineering as a systems discipline

- Policy (what should be protected)
- Mechanisms (cryptography, hardware security…)
- Attacks (malicious code, exploiting users…)
- Assurance (assessing how secure it is)

Objectives

By the end of the course, you should be able to tackle an information protection problem by

- drawing up a threat model
- formulating a security policy and
- designing specific protection mechanisms to implement the policy

Broad range of topics

Not an "axiom, theorem, exercise" subject
You must learn to think outside the box

- Human factors, Security Policy, Crypto, Protocols, Incentives etc
- Guest lectures, to broaden horizons:
  - Dr Richard Clayton, security economics
  - Dr Steven Murdoch, anonymous communications

Resources

for my part of the course

- Anderson, *Security Engineering*
  - developed from lecture notes
  - free online from author's page
- Course web page (has research papers)
- Further reading:
  - Gollmann, *Computer Security*
  - Cialdini, *Influence: science and practice*
  - Stajano, *Security for ubiquitous computing*
  - Kahneman, *Thinking fast and slow*

Further resources

- History:
  - David Kahn: *The Codebreakers*
  - Gordon Welchmann: *The Hut Six Story*
  - Glenn Greenwald: *No Place to Hide*
- **Use the Source, Luke:**
  - Read the original papers (books come after papers)
- Lab:
  - Security seminars: Tuesdays, LT2, 1400-1500
  - Security group meetings: Fridays, FW11, 1600-1700

http://www.cl.cam.ac.uk/teaching/current/SecurityII/materials.html

We also run national and international hacking competitions, with *serious* prizes.

INTER-ACE CYBERCHALLENGE

Cambridge 2 Cambridge cybersecuritychallenge

InterACE 2016

Cambridge2Cambridge 2016

LIVE CTF

1. **Johnny Cached**
   Cheng Chen, MIT
   Alex Dalgleish, University of Cambridge
   Julian Fuchs, MIT
   Gábor Szarka, University of Cambridge
2. **Class Warfare**
   Michael Choi, MIT
   Rahul Sridhar, MIT
   Daniel Wong, University of Cambridge
   Priyesh Patel, University of Cambridge
3. **Total Recursion**
   Cecilia Testart, MIT
   Ronald Gil, MIT
   Jan Ondras, University of Cambridge
   Will Shackleton, University of Cambridge
4. **Hacking**
   Ashley Wang, MIT
   Witchakorn Kamolpornwijit, MIT
   Daniel Chatfield, University of Cambridge
   William Moses, MIT
   Anish Athalye, MIT
5. **JGRTSec**
   Theodor Nedelcu, University of Cambridge
   Ray Wang, MIT
   Josiah Yan, University of Cambridge
   Gregory Falco, MIT
6. **Paragon**
   Samuel Yeom, MIT
   Zachary Neely, MIT
   Devin Neal, MIT
   Brett Gutstein, University of Cambridge

RAPID FIRE

1. Julian Fuchs, MIT
2. Will Shackleton, University of Cambridge
3. Witchakorn Kamolpornwijit, MIT
4. Rahul Sridhar, MIT

CRACKING CRYPTO

1. Cheng Chen, MIT
2. Ashley Wang, MIT
3. Will Shackleton, University of Cambridge
4. Devin Neal, MIT
5. Priyesh Patel, University of Cambridge
6. Julian Fuchs, MIT

POINT OF SALE

1. **Total Recursion**
   Cecilia Testart, MIT
   Ronald Gil, MIT
   Jan Ondras, University of Cambridge
   Will Shackleton, University of Cambridge
2. **Johnny Cached**
   Cheng Chen, MIT
   Alex Dalgleish, University of Cambridge
   Julian Fuchs, MIT
   Gábor Szarka, University of Cambridge
3. **Hacking**
   Ashley Wang, MIT
   Witchakorn Kamolpornwijit, MIT
   Daniel Chatfield, University of Cambridge
   William Moses, MIT
   Anish Athalye, MIT
4. **JGRTSec**
   Theodor Nedelcu, University of Cambridge
   Ray Wang, MIT
   Josiah Yan, University of Cambridge
   Gregory Falco, MIT
5. **Class Warfare**
   Michael Choi, MIT
   Rahul Sridhar, MIT
   Daniel Wong, University of Cambridge
   Priyesh Patel, University of Cambridge
6. **Paragon**
   Samuel Yeom, MIT
   Zachary Neely, MIT
   Devin Neal, MIT
   Brett Gutstein, University of Cambridge

BREAKING BINARIES

3. **Hacking**
   Ashley Wang, MIT
   Witchakorn Kamolpornwijit, MIT
   Daniel Chatfield, University of Cambridge
   William Moses, MIT
   Anish Athalye, MIT
4. **JGRTSec**
   Theodor Nedelcu, University of Cambridge
   Ray Wang, MIT
   Josiah Yan, University of Cambridge
   Gregory Falco, MIT
6. **Paragon**
   Samuel Yeom, MIT
   Zachary Neely, MIT
   Devin Neal, MIT
   Brett Gutstein, University of Cambridge
1. **Total Recursion**
   Cecilia Testart, MIT
   Ronald Gil, MIT
   Jan Ondras, University of Cambridge
   Will Shackleton, University of Cambridge
5. **Class Warfare**
   Michael Choi, MIT
   Rahul Sridhar, MIT
   Daniel Wong, University of Cambridge
   Priyesh Patel, University of Cambridge
2. **Johnny Cached**
   Cheng Chen, MIT
   Alex Dalgleish, University of Cambridge
   Julian Fuchs, MIT
   Gábor Szarka, University of Cambridge

LOCK PICKING

1. Cheng Chen, MIT
2. Josiah Yan, University of Cambridge
3. Will Shackleton, University of Cambridge
4. Priyesh Patel, University of Cambridge
5. Anish Athalye, MIT
6. Witchakorn Kamolpornwijit, MIT

Thank you to all of our 2016 participants! Congratulations!

InterACE and C2C 2017

- InterACE 2017
  - Between the 13 ACE-CSRs in the UK
  - Each university sends a team of 4
- Cambridge2Cambridge 2017
  - Top UK and US universities
  - We mix the teams
  - Even newcomers may win top prize
- Why?
  - Make cyber security fun
  - Security practice as well as theory
  - Build a network of good people

INTER-ACE CYBERCHALLENGE

18 March 2017

Cambridge 2 Cambridge cybersecuritychallenge

24-26 July 2017

Be part of it!
http://cambridge2cambridge.cl.cam.ac.uk/
Student society: http://cybersoc.soc.srcf.net/

UNIVERSITY OF CAMBRIDGE | Massachusetts Institute of Technology | GCHQ | Cabinet Office

What is Security Engineering?

Security engineering is about building **systems** to remain dependable in the face of malice, error and mischance.

As a discipline, it focuses on the tools, processes and methods needed to design, implement and test complete **systems**, and to adapt existing **systems** as their environment evolves.

Systems!

- A *system* can be:
  - a product or component (PC, smartcard,…)
  - some products plus O/S, comms and infrastructure
  - the above plus applications
  - the above plus internal staff
  - the above plus customers / external users
- Common failing: policy drawn too narrowly
  - Want a secure system? You need to consider *users*

Security, human factors and psychology

Why Johnny can't encrypt

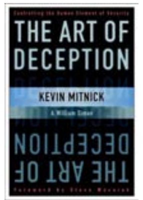
Whitten, Tygar: 'Why Johnny Can't Encrypt', 1999.

- Study of encryption program PGP – showed that 90% of users couldn't get it right give 90 minutes
- Private / public, encryption / signing keys, plus trust labels was too much – people would delete private keys, or publish them, or whatever
- Security is hard – unmotivated users, abstract security policies, lack of feedback…
- Geeky "security experts" would rather deal with machines than with unpredictable people. They miss the point.

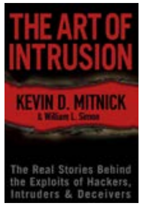
Users are not the enemy

Adams, Sasse: "Users are not the enemy", 1999.

- Insufficient communication with users produces unusable systems
- Users forced to comply with password mechanisms incompatible with work practices will look for workarounds
- Vicious circle:
  - Security departments think users are inherently insecure
  - Users think security departments get in the way of real work
- But "users never motivated to behave securely" is wrong!
- Treat users as stakeholders and they'll cooperate
- Provide feedback, guidance, awareness; and usable security



Think like an attacker



- Mitnick: *The art of deception*, 2003
  and *The art of intrusion*, 2005.
  - You don't have to pick the lock or break into the server: get someone on the inside to open the door for you
  - Pretext calls surprisingly effective
- Traditional responses:
  - mandatory access control
  - operational security

But why do the attacks work?

Phishing

- Started in 2003 with six reported (there had been isolated earlier attacks on AOL passwords)
- By 2006, UK banks lost £35m (£33m by one bank) and US banks maybe $200m
- Early phish crude and greedy; but phishermen learned fast
- E.g. ‘Thank you for adding a new email address to your PayPal account’
- The banks make it easy for them

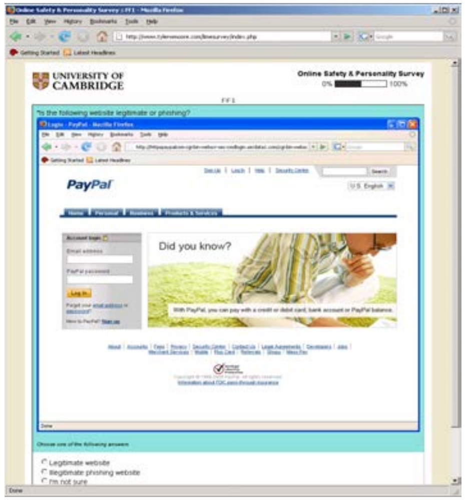
Types of phishing website

- Misleading domain name
  `http://www.banckname.com/`
  `http://www.bankname.xtrasecuresite.com/`
- Insecure end user
  `http://www.example.com/~user/www.bankname.com/`
- Insecure machine
  `http://www.example.com/bankname/login/`
  `http://149.32.40.1/bankname/login/`
- Free web hosting
  `http://www.bank.com.freespacesitename.com/`

Fraud and Phishing Patterns

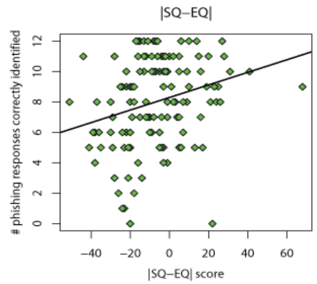
- Fraudsters do pretty well everything that normal marketers do
- The IT industry has abandoned manuals – people learn by doing, and marketers train them in unsafe behaviour (click on links…)
- Banks’ approach is ‘blame and train’ – long known to not work in safety critical systems
- Their instructions ‘look for the lock’, ‘parse the URL’ are easily turned round, and discriminate against nongeeks

A simple user study



Results

- Ability to detect phishing is (albeit rather loosely) correlated with SQ-EQ (S=systemizer, E=empathizer)
- It is (independently) correlated with gender
- So the gender HCI issue applies to security too



Think like a victim

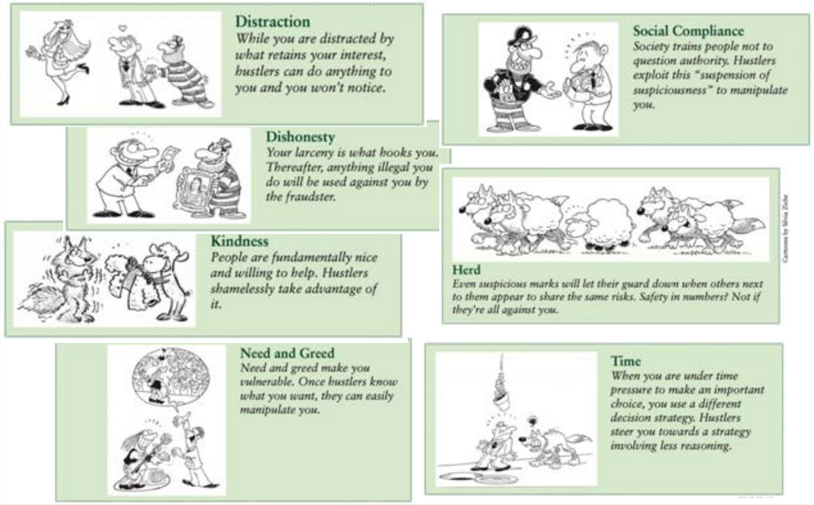
Stajano, Wilson: “Understanding scam victims”, 2009

- Learn from fraudsters: they know how to push the victims' buttons
- The Real Hustle (BBC3): hundreds of scams recreated for hidden cameras

What makes these scams work?

Understanding scam victims

Seven principles for systems security

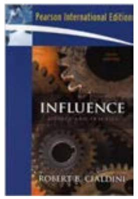


Seven principles for systems security

- Principles for sales, scams and other "persuasion" contexts
- Those principles predate computers! Rooted in human nature
- Thus, each also applies to computer *systems* security
  - Phishing: user is fixated on task completion (e.g. finding why new payee on PayPal account)
  - Advance fee frauds (419) take this to extreme lengths!
- You have to accept them, almost like the laws of physics
  - That's the way people work, no matter what you tell them to do
  - Which exact principles… is not even that important

It is arrogantly idiotic for security engineers to whinge that "users are gullible". Certain behavioural patterns are simply human nature. Smart security engineers must acknowledge their inevitability and design the system to prevent their exploitation.

Weapons of influence



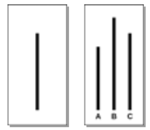
Cialdini: *Influence: science and practice*, 5th ed 2009.

Based on undercover study of salesmen "and other compliance professionals"
Salesmen are like scam artists, except legal (perhaps)

– Reciprocation: they'll feel compelled to respond
– Commitment and consistency: "but you previously said X"
– Social proof: like to do what others do
– Liking: want to deal with people they can relate to
– Authority: will defer to authority figure
– Scarcity: less is best and loss is worst

Framing effects include 'Was £8.99 now £6.99' and the estate agent who shows you a crummy house first. Take along an ugly friend on a double date.

Social Psychology



Theory

– Solomon Asch, 1951 (conformity experiments). 2/3 of subjects deny obvious facts to conform to group
– Stanley Milgram, 1964: a similar number will administer torture if instructed by an authority figure
– Philip Zimbardo, 1971 (Stanford prison experiment). You don't need authority: the subjects' situation or context is enough

Cfr Herd principle and Authority principle

Practice

– The Officer Scott case: a "police officer" phones a fast food restaurant and persuades the manager to strip-search and sexually humiliate an employee
– Abu Ghraib

What should you do with users you can't train (your customers)? Cfr phishing.

Framing effects

- Imminent outbreak of some Asian Disease is expected to kill 600
- Two programs to combat the disease have been proposed
  - Program A:
    - 200 people saved
  - Program B:
    - 1/3 chance that 600 saved
    - 2/3 chance that 0 saved
  - Program A':
    - 400 people die
  - Program B':
    - 1/3 chance that 0 die
    - 2/3 chance that 600 die

Framing effects

- Imminent outbreak of some Asian Disease is expected to kill 600
- Two programs to combat the disease have been proposed
  - Program A:
    - 200 people saved
  - Program B:
    - 1/3 chance that 600 saved
    - 2/3 chance that 0 saved
  - Program A':
    - 400 people die
  - Program B':
    - 1/3 chance that 0 die
    - 2/3 chance that 600 die

Substantial majority favours A: let's save those 200 (risk-averse)

Substantial majority favours B': let's not kill those 400 (risk-seeking)

Why the switch?

Attitudes towards risk

Which would you choose?
- Win £900 for sure
- 90% chance of winning £1000

(For a rational agent, there should be no difference; but what will you do if it's your own real money?)

Which would you choose?
- Lose £100 for sure
- 10% chance of losing £1000

Again: in theory, no difference. But with your real money?

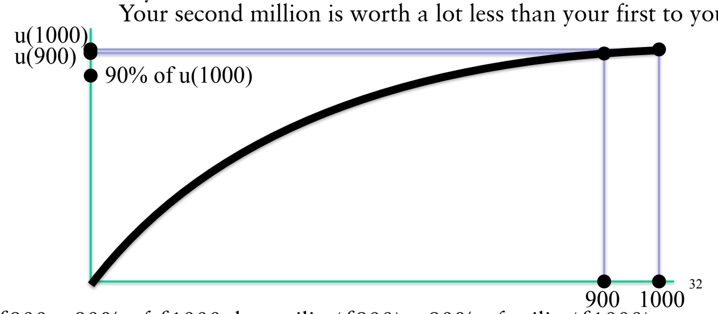
Attitudes towards risk

Which would you choose? *Give me the certain money!*
- Win £900 for sure **risk-averse**
- 90% chance of winning £1000

Bernoulli's Expected Utility theory explains why

Utility of wealth is ***less than linear***

Your second million is worth a lot less than your first to you



£900 = 90% of £1000; but utility(£900) > 90% of utility(£1000)

Attitudes towards risk

Which would you choose? *Give me the certain money!*

- Win £900 for sure **risk-averse**
- 90% chance of winning £1000

Bernoulli's Expected Utility theory explains why.
Utility of wealth is less than linear.

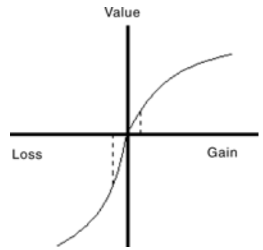
"A pay rise of 10 k£" is a big deal if you're on 20 k£; less if you're on 80 k£.
"A pay rise of 50%" is a big deal either way. (So, log makes sense for U(w))

Which would you choose? *Let me try to get away without paying!*

- Lose £100 for sure **risk-seeking**
- 10% chance of losing £1000

But note! If I give you £1000 after the loss, expected values are the same as in the top problem!
Bernoulli's utility of wealth *can't explain* why you switch.
The difference is only *framing* (as a gain or a loss). *WTF?*

Prospect theory

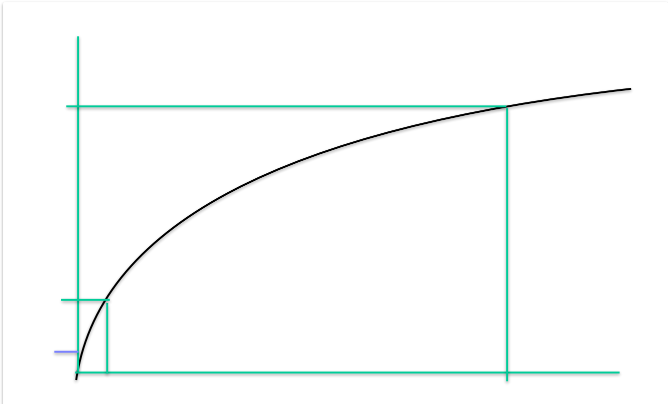


- Kahneman & Tversky, 1979
  - *Critique* of Expected Utility theory
  - *Descriptive* analysis of decision under risk
  - Importance of *framing*
  - Cfr *perception* (hot or cold water)

- Utility of wealth? No, of changes from reference point *(gains and losses matter, not overall wealth)*
- Both curves lean towards the horizontal *(both higher gains and higher losses are progressively less relevant)*
- But steeper for losses than for gains *(we dislike a loss much more than we like a win; and by about 2x)*
- Concave for gains *(risk-averse)*, convex for losses *(risk-seeking)*

Why concave for gains, convex for losses?

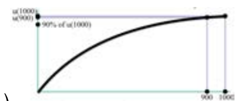
Consider: "win 100" vs "10% chance of winning 1000"



Consider: "lose 100" vs "10% chance of losing 1000"

Decision under risk: a summary

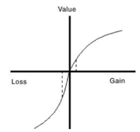
We have seen 3 "generations" of explanations:



- Raw probabilistic computation
- Expected Utility theory (Bernoulli)
- Prospect theory (Tversky and Kahneman)

"This is what might happen. How much is it worth to you?"

What does a decision theory claim to indicate?



- This is how people should behave
- This is how smart people actually behave

Tversky & Kahneman point out that Bernoulli thought he was doing both…

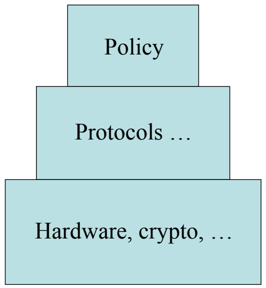
Security policies

Managing security

- Security awareness: measures must have, and be seen to have, full support of management
- Measuring security is hard
  - Measure security bugs, attack surface, attack cost…
- Risk analysis
  - Assets, vulnerabilities, threats, probabilities
  - That's quantitative, but inputs are usually guesswork
- Security policy: an instrument of communication

Design Hierarchy

- What are we trying to do?
- How?
- With what?



Terminology

Trust is hard to define accurately

1. a warm fuzzy feeling
2. a trusted system or component is one that can break my security policy
3. a trusted system is one I can insure
4. a trusted system won't get me fired when it breaks

We'll use the NSA definition – number 2 above – by default.

E.g. an NSA man selling key material to the Chinese is *trusted* but *not trustworthy* (assuming his action was unauthorised)

Terminology

- A *security policy* is a succinct statement of protection goals – typically less than a page of normal language
- A *protection profile* is a detailed statement of protection goals – typically dozens of pages of semi-formal language
- A *security target* is a detailed statement of protection goals applied to a particular system – and may be hundreds of pages of specification for both functionality and testing

What often passes as 'Policy'

1. This policy is approved by Management.
2. All staff shall obey this security policy.
3. Data shall be available only to those with a 'need-to-know'.
4. All breaches of this policy shall be reported at once to Security.

What's wrong with this?

Policy: Multi Level Security

- Multilevel Secure (MLS) systems are widely used in government / intelligence / military contexts
- Basic idea: a clerk with 'Secret' clearance can read documents at 'Confidential' and 'Secret' but not at 'Top Secret'
- 1960s/70s: problems with early mainframes
- First security policy to be worked out in detail following Anderson report (1973) for USAF which recommended keeping security policy and enforcement simple

Levels of Information

- Levels include:
  - **Top Secret:** compromise could cost many lives or do exceptionally grave damage to operations. E.g. intelligence sources and methods, battle plans
  - **Secret:** compromise could threaten life directly. E.g. weapon system performance, combat reports
  - **Confidential:** compromise could damage operations
  - **Restricted:** compromise might embarrass

  else "unclassified"
- Resources have classifications
- Principals have clearances
- Information flows upwards only

Context of Multilevel Security

- Information mustn't leak from High to Low
- Enforcement must be independent of user actions
- Perpetual problem: careless staff
- 1970s worry: operating system insecurity
- 1990s worry: virus at Low copies itself to High and starts signalling down (e.g. covert channel)

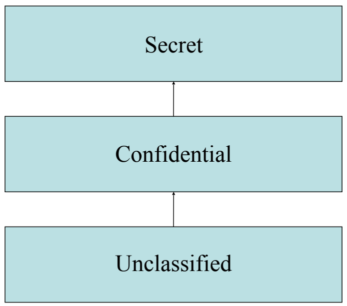
Manning (2010) and Snowden (2013)
show us how things actually go wrong in practice...

Context of Multilevel Security

Nagaraja, Anderson 'The Snooping Dragon', 2009.

- September 2008: Dalai Lama's office realised there had been a security failure
- Initial break: targeted email with bad pdf
- Then: took over the mail server and spread it
- About 35 or their 50 PCs were infected
- Fix (Dharamsala): take 'Secret' stuff offline
- Fix (UKUSA agencies): use MLS mail guards and firewalls to prevent 'Secret' stuff getting out

Authorized Information Flow

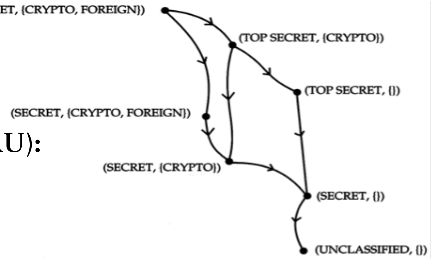


Formalising the Policy

- Bell-LaPadula (1973):
  - *simple security property*: no read up
  - **-property*: no write down
- With these, one can prove that a system which starts in a secure state will remain in one
- Ideal: minimise the Trusted Computing Base (set of hardware, software and procedures that can break the security policy) so it's verifiable
- 1970s idea: use a reference monitor

The Lattice Model

- Intelligence agencies manage 'compartmented' data by adding categories. Label = (level, {set of categories})
- Basic idea: BLP requires only a partial order (*dominates*).
- X dominates Y iff level(X) ≥ level(Y) and cat(X) ⊇ cat(Y)



- **BLP simple property** (**NRU**): X can read Y iff X dominates Y
- **BLP *property** (**NWD**): X can write Y iff X is dominated by Y

Objections to BLP

- Some processes, such as memory management, need to read and write at all levels
- Fix: put them in the trusted computing base
- Consequence: once you put in all the stuff a real system needs (backup, recovery, comms…) the TCB is no longer small enough to be easily verifiable

Objections to BLP

- John MacLean's "System Z": as BLP but lets users request temporary declassification of any file
- Fix: add tranquility principles
  - Strong tranquility: labels never change
  - Weak tranquility: they don't change in such a way as to break the security policy
- Usual choice: weak tranquility using the "high watermark principle" – a process acquires the highest label of any resource it's touched
- Problem: have to rewrite apps (e.g. license server)

Objections to BLP

- High can't acknowledge receipt from Low
- This blind write-up is often inconvenient: information vanishes into a black hole
- Option 1: accept this and engineer for it (Morris theory) – CIA usenet feed
- Option 2: allow acks, but be aware that they might be used by High to signal to Low
- Use some combination of software trust and covert channel elimination

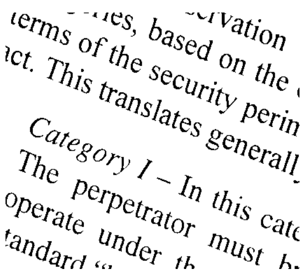
Covert Channels

- In 1973 Butler Lampson warned BLP might be impractical because of covert channels: “neither designed not intended to carry information at all”
- A Trojan at High signals to a buddy at Low by modulating a shared system resource
  - Fills the disk (storage channel)
  - Loads the CPU (timing channel)
- Capacity depends on bandwidth and S/N. So: cut the bandwidth or increase the noise
- But it’s really hard to get below 1 bit/s or so…

Downgrading

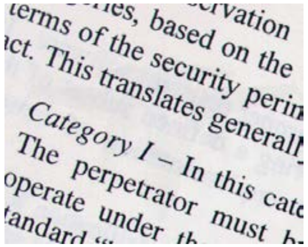
- A related problem to the covert channel is how to downgrade information
- Analysts routinely produce Secret briefings based on Top Secret intelligence, by manual paraphrasis
- Also, some objects are downgraded as a matter of deliberate policy – an act by a trusted subject
- For example, a Top Secret satellite image is to be declassified and released to the press

Downgrading



Text hidden in least significant bits of image

Downgrading

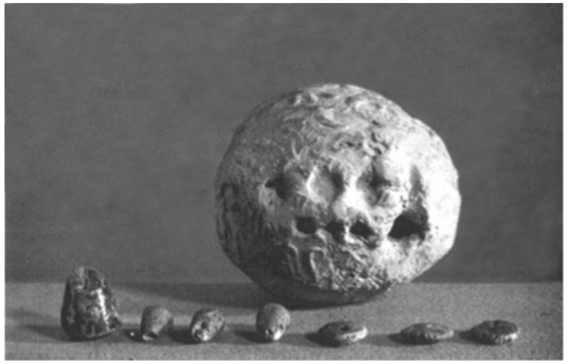


Picture hidden in three least significant bits of text

Multilevel Integrity

- The Biba model – data may flow only down from high-integrity to low-integrity
- Dual of BLP: (“BLP upside down”)
  - **Simple integrity property:** subject may write to object iff object has same or lower label as subject
  - ***-integrity property:** subject may read object iff object has same or higher label as subject
- So you have low watermark properties, etc
- Example: medical equipment with two levels, “calibrate” and “operate”

Bookkeeping, c. 3300 BC



Bookkeeping c. 1100 AD

- How do you manage a business that’s become too large to staff with your own family members?
- Double-entry bookkeeping – each entry in one ledger is matched by opposite entry in another
  - E.g. firm sells £100 of goods on credit – credit the sales account, debit the receivables account
  - Customer pays – credit the receivables account, debit the cash account

  (Some of these may sound backwards but make sense to accountants)
- So bookkeepers have to collude to commit fraud

Banking Security Policy

- Threat model:
  - 1% of staff go bad each year
  - Mistakes happen – 1 in 500 paper transactions
  - There are clever fraudsters too
  - Loss of confidence means ruin
- Protection goals:
  - Deter/prevent the obvious frauds
  - Detect the rest as soon as possible
  - Be able to defend the bank’s actions in court

The Clark-Wilson Policy Model

Work by David Clark (MIT) and David Wilson (Ernst & Whinney) in 1986 to model double-entry bookkeeping

- In addition to the normal objects in your system, which we call unconstrained data items (UDIs), you add constrained data items (CDIs)
- CDIs are acted on by special programs called transformation procedures (TPs) that preserve the invariants
- IVPs (integrity verification procedures) verify the validity of CDIs (eg that the books balance)
- Mental model: a TP in a bank must increase the balance in one CDI (account) by the same amount that it decrements another

Clark-Wilson rules

1. There's an IVP to validate integrity of each CDI
2. Applying a TP to a CDI maintains integrity
3. A CDI can only be changed by a TP
4. Subjects can use only certain TPs on certain CDIs
5. Triples (subject, TP, CDI) enforce separation of duty
6. Special TPs on UDIs can produce CDIs
7. Each TP application must be logged to special append-only CDI
8. System must authenticate subjects that attempt to launch a TP
9. Only special subjects (admins) can change auth lists

Clark-Wilson importance

- First influential security policy model not based on BLP
- Application-level security state
  - The audit log (with enough info to reconstruct each TP)
  - The triples
- Separation of duties
  - In parallel (require 2 signatures, e.g. for large and irreversible transactions)
  - In series (different people for raising an order, accepting delivery, paying invoice, balancing budget)

Ubiquitous computing

- Authentication and device pairing without infrastructure
  - Two devices meet for the first time
  - No online servers available
    - Can't do the key distribution protocols studied later in this course
    - Can't use PKI, because you can't check for revoked keys
  - One device wants the other to "do something"
  - Authentication as temporary master-slave pairing: *Secure Transient Association* e.g. Smart Home devices + Universal Controller (phone)

Big Stick policy

(Stajano, 2000)

Whoever has physical control of the device
is allowed to take it over

- E.g. you can press the hard reset button on your router and the admin password is restored to factory default
- A trivial policy; but effective because cynically realistic
- Works fine for your lawnmower, pocket calculator (stateless) or for your fridge or router (inside the home)
- But not good for devices with valuable state that may be left unattended (e.g. a vending machine, a wireless sensor)

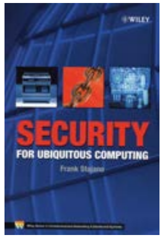
Imprinting

- Inspired by Konrad Lorenz (1973 Nobel prize)
- First moving subject seen by duckling becomes its mother
- Duckling stays faithful to mother until death

Out of metaphor:

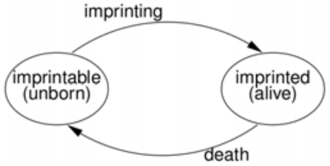
- Slave device starts as imprintable
- First device that gives it a key becomes its master
- Bootstrap with unmediated physical channel

...but what if you then want to sell your Blu-Ray player?



Resurrecting Duckling policy

Stajano, Anderson 1999



- **Two-state:** *duckling* can be imprintable or imprinted
- **Imprinting:** transition to imprinted when someone (henceforth *mother duck*) sends imprinting key over secure channel
- **Death:** transition to imprintable when mother duck orders it (like seppuku of the samurai)
- **Assassination:** uneconomical for attacker artificially to cause transition to imprintable (implies tamper resistance)

Most work on bootstrapping security associations refers to this

Chinese Wall policy

Brewer, Nash 1989

Intellectual elegance of BLP is appealing: inspired many followers
But even for something that simple, getting the details right is hard

- Simple rule: Read or write access to object o2 by subject s is granted if and only if, for all objects o1 to which s has had access, we have: (class(company(o1)) != class(company(o2)) or (company(o1) = company(o2)).
- *-rule: Write access to object o2 by subject s is granted if and only if access is granted by the simple rule and there does not exist any unsanitized object o1, readable by s, for which company(o1) != company(o2).

Written up in several textbooks. Yet contains a disastrous flaw.
Can you spot anything wrong?

Availability Policies

- Until recently, security researchers ignored availability. But it's where the money goes!

	research	industry
confidentiality	90%	1%
integrity/authenticity	9%	9%
availability	1%	90%

- Availability matters a lot for, e.g., burglar alarms (more on this later)

Policy

- Bell-LaPadula, Biba and Clark-Wilson are only three early examples of policy
- Many industries develop their own policies, and may get Protection Profiles evaluated
- Many things go wrong – people protect the things they can, not the things they should
- We often see deception at the policy level!

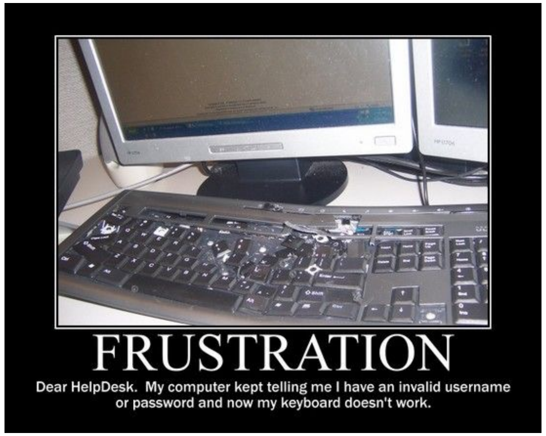
Passwords

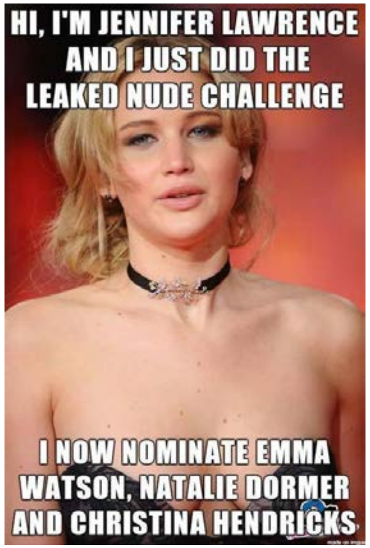
and liberating humanity from them

Passwords

- Have many usability shortcomings
- Also have many security shortcomings
- But continue to be dominant

"There is no doubt that over time, people are going to rely less and less on passwords. People use the same password on different systems, they write them down and they just don't meet the challenge for anything you really want to secure"
(Bill Gates, keynote @ RSA conference, 2004)






theguardian

Adobe

Did your Adobe password leak? Now you and 150m others can check

Leak is 20 times worse than the company initially revealed, and could put huge numbers of peoples' online lives at risk

Adobe's HQ. The company leaked over 100m users' details. Photograph: PAUL SAKUMA/ASSOCIATED PRESS

Passwords Must:

- Be a minimum of nine characters in length
- Contain each of the following in the first nine characters:
  - Two Uppercase Letters
  - Two Lowercase Letters
  - Two Special Characters (except ? which is reserved)
  - Two Numeric Characters
- Be changed every 90 days

(USAF portal, 2007)

How passwords work "for dummies" (?)

Prover

Verifier

userid	pwd
alice	pa$$word
bob	123456
charlotte	letmein
derek	qwerty
emily	123456

Verifier could store a file with the password of each user

But all passwords exposed (bad idea) if server is compromised

The hash: a one-way function

x ↦ **h(x)** is easy to compute
(a bit like computing the cube)

h(x) ↦ **x** is v. hard to compute
(a bit like extracting the cubic root)

h(123456) =
f447b20a7fcbf53a5d5be013ea0b15af

h(123457) =
91d3af515b5f077bb56d3efb8d162232

h(to be or not to be) =
f9d804763c3031cc22323d79e165b562

Now the bad guy who compromises the server can't read off the passwords

userid	*h(pwd)*
alice	h(pa$$word)
bob	h(123456)
charlotte	h(letmein)
derek	h(qwerty)
emily	h(123456)

Brute-forcing hashed passwords

- Try hashing all possible passwords and verify each guess. How long does it take?
  charset^length * guessTime
- That's where those annoying password policies come from

	4	6	8	10	12
26 abc	0.5 s	5 min	2 d	5 y	3 ky
36 abc123	2 s	36 min	32 d	116 y	150 ky
52 abcABC	8 s	5 h	1 y	4500 y	12 My
64 abcABC123%@	17 s	19 h	9 y	36 ky	149 My

guessTime = 1 μs

Side note: attacks only get better

- Moore's law makes computers twice as fast every 2 years
- 1000x faster after 20 years
- Attacker's computer keeps getting more powerful
- Defender might compensate by iterating the hash

	4	6	8	10	12
26 abc	0.5 s	5 min	2 d	5 y	3 ky
36 abc123	2 s	36 min	32 d	116 y	150 ky
52 abcABC	8 s	5 h	1 y	4500 y	12 My
64 abcABC123%@	17 s	19 h	9 y	36 ky	149 My

guessTime = 1 μs goes down to 1 ns
but you may h(h(h(h(h(...h(x)...))))) a thousand times to compensate

Smart guessing of passwords

Normal people don't use passwords like zM%3Dz*S

Bad guys are not stupid: they check the plausible ones first

Bad guys can also check ALL the passwords of a certain shape up to some length

userid	*h(pwd)*
alice	h(pa$$word)
bob	h(123456)
charlotte	h(letmein)
derek	h(qwerty)
emily	h(123456)

Hashes of passwords (c) Bad Guys Association

hash	*password*
h(password)	password
h(123456)	123456
h(qwerty)	qwerty
h(abc123)	abc123
h(letmein)	letmein
h(monkey)	monkey
h(myspace1)	myspace1
h(password1)	password1

(searchable by hash)

Salting and hashing

userid	salt	h(salt‖pwd)
alice	OTRh	h(OTRhpa$$word)
bob	OTUx	h(OTUx123456)
charlotte	Yjcy	h(Yjcyletmein)
derek	ZjAx	h(ZjAxqwerty)
emily	OTky	h(OTky123456)

?

Hashes of passwords
(c) Bad Guys Association

hash	password
h(password)	password
h(123456)	123456
h(qwerty)	qwerty
h(abc123)	abc123
h(letmein)	letmein
h(monkey)	monkey
h(myspace1)	myspace1
h(password1)	password1

(searchable by hash)

Password strength

The crucial distinction is between **offline** and **online** guessing

Florencio & Herley: draconian password policies are just incompetent websites covering their backside

userid	salt	h(salt‖pwd)
alice	OTRh	h(OTRhpa$$word)
bob	OTUx	h(OTUx123456)
charlotte	Yjcy	h(Yjcyletmein)
derek	ZjAx	h(ZjAxqwerty)
emily	OTky	h(OTky123456)

Password Manager Friendly

Stajano, Spencer, Jenkinson, Stafford-Fraser, 2014

Normal people don't use passwords like zM%3Dz*S

But a password manager (or Pico) can remember even NDUrNDQzMTQsYTkwYjcwMDFiMzcyOGZkZGVhNWVkZTM!

Impossible to brute-force from the leaked password file, even with all the graphics cards in the world

But stupid websites will reject this password because it has no numbers or symbols (even though they accept Pa$$w0rd)

PMF: a simple standard to allow password managers to interact with websites, robustly and without pointless guesswork

- This is a login page, a signup page, an error page
- This is the username field, the password field, the submit button
- If the password is over 64 characters, just take it as is

http://pmfriendly.org

The compliance budget

Beautement, Sasse, Wonham, 2008.

- Users assess cost/benefits of security measures and put up with some inconvenience for the good of the company
  - With virus scanner on, program takes longer to build
  - Encrypting USB stick might prevent me from giving talk
- But only up to a point!
  - User patience is finite: "the compliance budget"
  - Once exhausted, user quickly stops cooperating

Must be managed like any other budget

Single Sign-On

One single password to rule them all

- Doesn't even have to be a password
- Will it really rule "all"?

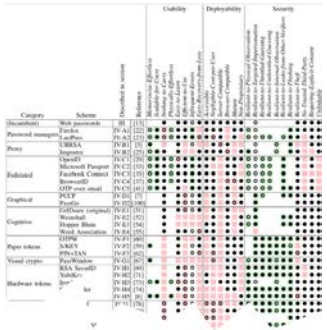
A taxonomy of SSO (Pashalidis and Mitchell, 2003)

	pseudo-SSO	*true-SSO*
local		
proxy-based		

Seen that way, password managers are SSO

Will we ever get rid of passwords?

- Cheapest for implementers
  - No need to explain them to users
  - Cost to support one more user is negligible (Facebook reached 1 M users before external funding)
  - Can't be that bad: everyone else uses them too ("tragedy of the commons")
- Very many alternatives have been proposed
- "easier to remember" schemes: can they scale?
- Nice properties: Single Sign-On; pwd managers.



Unsustainable in the long run, but still...
expect password to be around for a while

Bonneau, Herley, van Oorschot, Stajano.
"The quest to replace passwords", 2012

UNIVERSITY OF CAMBRIDGE

Pico

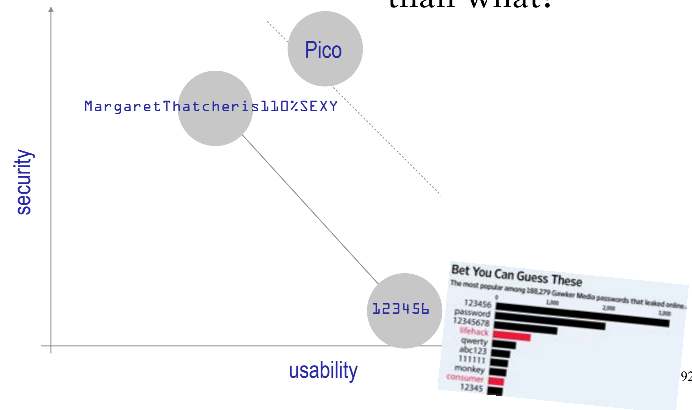
http://mypico.org

UNIVERSITY OF CAMBRIDGE

A challenge for Pico

more usable and more secure

than what?



Where is the pain?

Inertia
of verifiers
and
of provers

Pico: thanks and credits to…

Current Pico staff:

David Llevellyn-Jones
Claudio Dettoni Jr
Seb Aebischer
Kat Krol

Former Pico staff:

Graeme Jenkinson
Jeunese Payne
Max Spencer
Quentin Stafford-Fraser
Chris Warrington

http://mypico.org
has our papers and videos

Pico students:

Bo Tian (BA 2012)
Oliver Stannard (BA 2012)
Anders Bentzon (MPhil 2013)
Fabian Krause (MPhil 2014)
Jonathan Millican (BA 2014)
Christian Toader (MPhil 2014)
Daniel Low (BA 2015)
Alex Dalgleish (summer intern 2015)
Agnes Cameron (summer intern 2015)
Fin Brown (summer intern 2015)
Spencer Thang (BA 2016)
Antonaela Siminiuc (BA 2016)
Adam Roberts (BA 2017)
James Brashko (BA 2017)

UNIVERSITY OF CAMBRIDGE

European Research Council
Established by the European Commission
Supporting top researchers from anywhere in the world

Physical security

The Physical Security Revolution

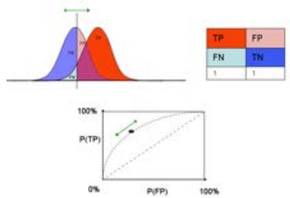
- IT security is rooted in physical security (for server rooms, crypto boxes etc)
- Old model: firms like Chubb with proprietary fire / burglar alarm systems; locks with master-keying systems
- New model: sensors run off ethernet like everything else
- We should be able to do better than metal systems
- Should be much easier to manage too – but many tensions (manageability, dependability)

Burglar Alarms

- Good example of a service where availability matters!
  - If there's a burglar in my vault I want the alarm company to be told!
  - Not bothered about confidentiality: you can tell other people too
  - Nor about authenticity: I don't care who tells them
- Wide range of systems: homes – supermarkets – jewellery stores – banks – nuclear facilities
- Wide range of standards, from Underwriters' Labs to the IAEA

How to Steal a Painting (1)

- Hollywood idea of art theft: cut through roof, climb down rope, grab painting without stepping on pressure mat (i.e. sensor defeat), get girl…
  - Response to this perceived threat is: more, fancier sensors
  - There are limits: set by false alarm rates and environmental conditions
  - Critical science: the Receiver Operating Characteristic (ROC) curve
  - Multisensor data fusion is really hard!
- But most high-grade attacks don't defeat sensors



How to Steal a Painting (2)

- More common type of art theft: hide in broom cupboard, come out at midnight, grab the Rembrandt and head for the fire exit
  - Understand the service you're supplying: deter – detect – alarm – delay – response
  - Don't rely on tech too much: 'Titanic effect'
- Or just toss in a smoke grenade. The fire alarm turns off the burglar alarm. Dash in and grab the Rembrandt
  - If caught, claim you were passing by and dashed in to save the national heritage

How to Steal a Painting (3)

- Wait for a dark and stormy night, when false alarms will be common. Create several (fence rattling). Wait till guards stop responding
  - Typical police force blacklists a property after 3–4 false alarms
  - Fix: multiple sensors, e.g. CCTV inside
  - Problem: we want best sensors on the outside for delay, but on the inside for low false alarm rate
- This is the standard way for professionals to do a bank vault! (Attack trust in the system)

How to Steal a Painting (4)

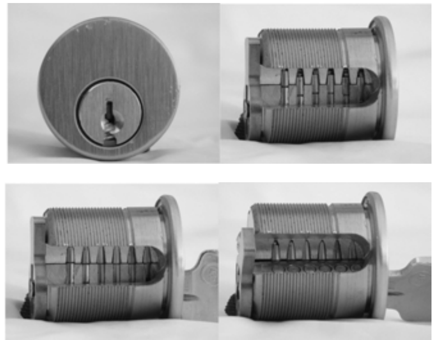
- Cut the wire from the sensor to the controller
- Connect a bogus controller to the phone line
- Cut the communications to the controller
- Cut the communications to many controllers
  - 2 independent channels for risks over £20m
  - Armed response force on premises for plutonium
- Insurance companies would like resilient anonymous communications to make service denial hard

Alarms – Lessons Learned

- Dealing with service denial is becoming more important, and harder
- Trade–off between false alarm rate and missed alarm rate is central
- You need to be clear what the service you're supplying (or buying) is – is it about sounding the alarm, or more?
- Critically, we need to design the system around the limitations of the human response. E.g. in airport screening, you insert deliberate false alarms. But what more can be said?

Really physical security

Basics of pin locks



When all pins align at the shear line, the cylinder can rotate and the lock opens.

B/W images courtesy of Matt Blaze

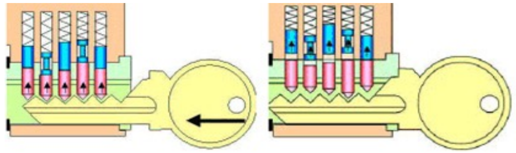
Basic lockpicking

- While applying torque, pick one pin at a time, in order of stickiness
- Cost is linear in number of pins (not exponential!)
- Requires a modest (but non-zero) amount of skill

When all pins align at the shear line, cylinder can rotate and lock opens.

Lock Bumping

With lock bumping, that's not necessary

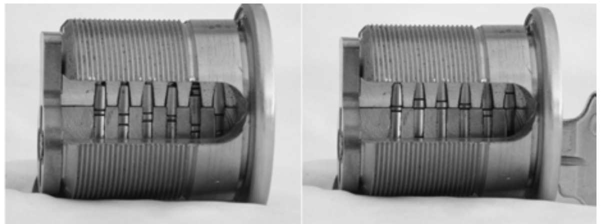


- Cut a key down to the (0,0,0,0,0) bitting
- Put it in the keyway, apply torque, and tap
- Pins bounce up to shear line and cylinder turns

Lock Bumping

- With fancy locks, like sidebar here: break that separately first
- pick it, or steal or photograph a key
- Enthusiasts have now defeated most mechanical locks

Master-key locks



- Intent: allow a single *master key* to open many locks that have different *change* keys
- Implementation: some or all pins are cut in more than one place

Example:

- Lock A opened by 11111
- Lock B opened by 22222
- Both opened by 44444

(but note unintentional cross-keying:
14114 opens A;
22442 opens B...)

Blaze attack on master key systems

Matt Blaze, “Rights Amplification in Master-Keyed Mechanical Locks”. *IEEE Security & Privacy* 1(2): 24-32 (2003). http://www.crypto.com/papers/mk.pdf

- **Preconditions:** Attacker (insider?) has one change key, several blanks, access to (own?) lock
- **Outcome:** Attacker recovers master key for whole system
- **Strategy:** For each pin, try all possible cuts, but copy the known key on the other pins (easy to find the “other” cut for this pin!)

Last words

First of all, participate in InterAce and C2C 2017

Still many interesting unsolved challenges in security waiting for smart people to solve them

Want to improve the world?
Want to create the content of future security textbooks?

Do a PhD with us!

Expecting a First? I definitely want to hear from you!