CSE 127 Computer Security

Stefan Savage, Spring 2018

Lecture 2: Security Foundations

Review

- What is a secure system?
  - "System that remains dependable in the face of malice" – Ross Anderson
- *Reflections on Trusting Trust* by Ken Thompson
- Chapter 1 from *The Craft of System Security*
- First project is due next week (4/9 @ 10pm)
  - Getting comfortable with the debugger and project submission system

Lecture Objectives

- Develop a mental framework for:
  - Thinking about security risks
  - Evaluating potential mitigation options
  - Analyzing tradeoffs

  - Some bonus material about locks and the mental mistakes we make

Threat Model

- ***Threat Model*** defines ***Security Goals***: What are we trying to protect from whom?
  - ***Assets*** and ***Attackers***
- Let's return to a definition of a secure system
  - "System that remains dependable in the face of malice" -Ross Anderson
- What does "remains dependable" mean?
  - ***Confidentiality***, ***Integrity***, and/or ***Availability (C.I.A.)*** of particular system, component, or data are preserved
  - These are the ***Assets*** we need to protect
    - "An attacker must not be able to compromise the [Confidentiality | Integrity | Availability] of…"
  - Assets have value: Must understand value in order to decide how much to spend on protection
- What does "malice" mean?
  - Malice from whom? Curious roommate? Criminal? Law enforcement? Foreign military?
  - These are potential ***Attackers***
    - Defined by ***Capability*** and ***Intent***

Threat Model

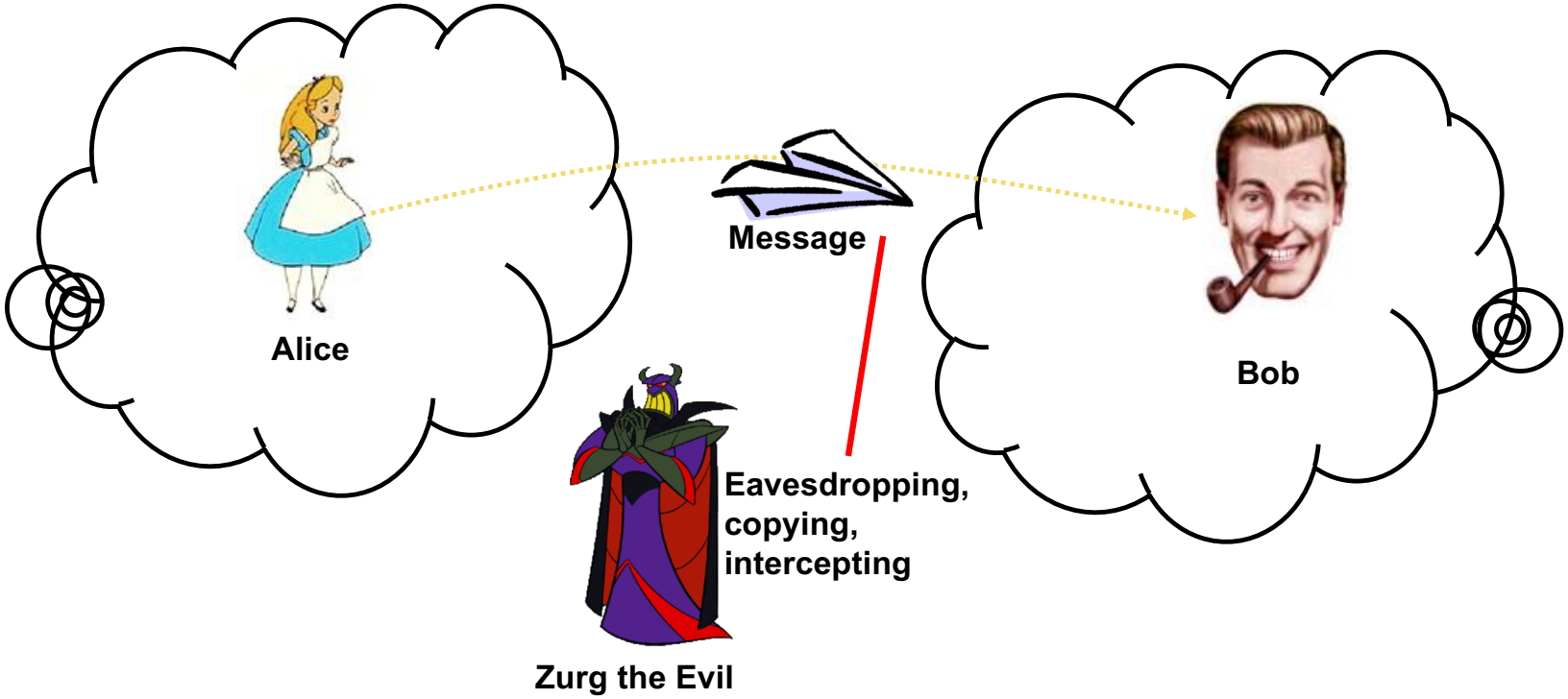
- ***Threat Model*** defines ***Security Goals***: What are we trying to protect from whom?
  - ***Assets*** and ***Attackers***
    - ***Confidentiality***, ***Integrity***, and/or ***Availability (C.I.A.)*** of particular system, component, or data

Confidentiality

- Prevention of unauthorized access to information
  - Unauthorized parties can't view protected information
  - aka Secrecy
- What are some scenarios where you want secrecy?

Confidentiality

- Protection of information; secrecy



Integrity (& Authenticity)

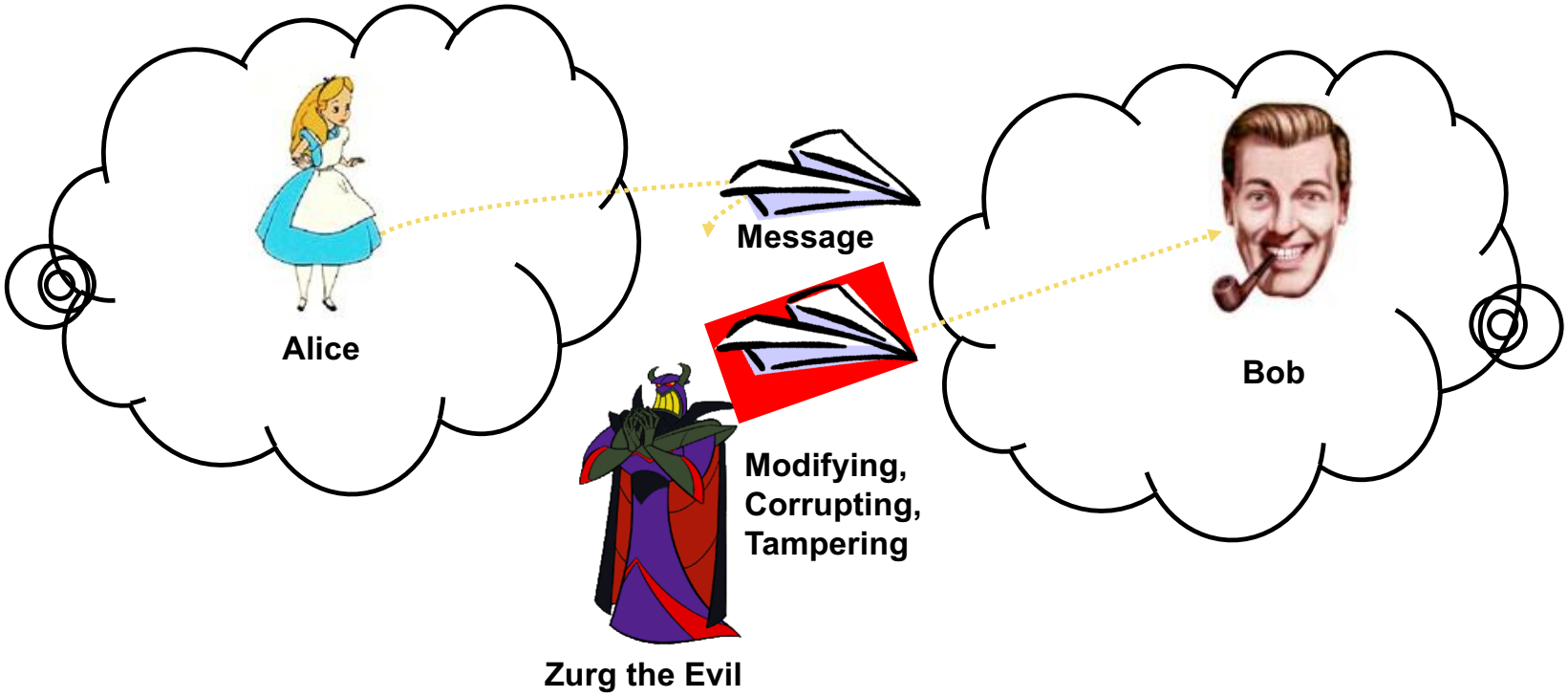
- Integrity: Prevention of unauthorized changes
- Authenticity: Identification and assurance of origin

Integrity (& Authenticity)

- Prevention of unauthorized modification of information, process, or function
  - Unauthorized parties can't modify protected information or process
- Examples
  - Increasing your bank account balance without depositing money
  - Getting snacks from vending machine without paying for them
- What are some other examples?

Integrity

- Prevention of unauthorized changes

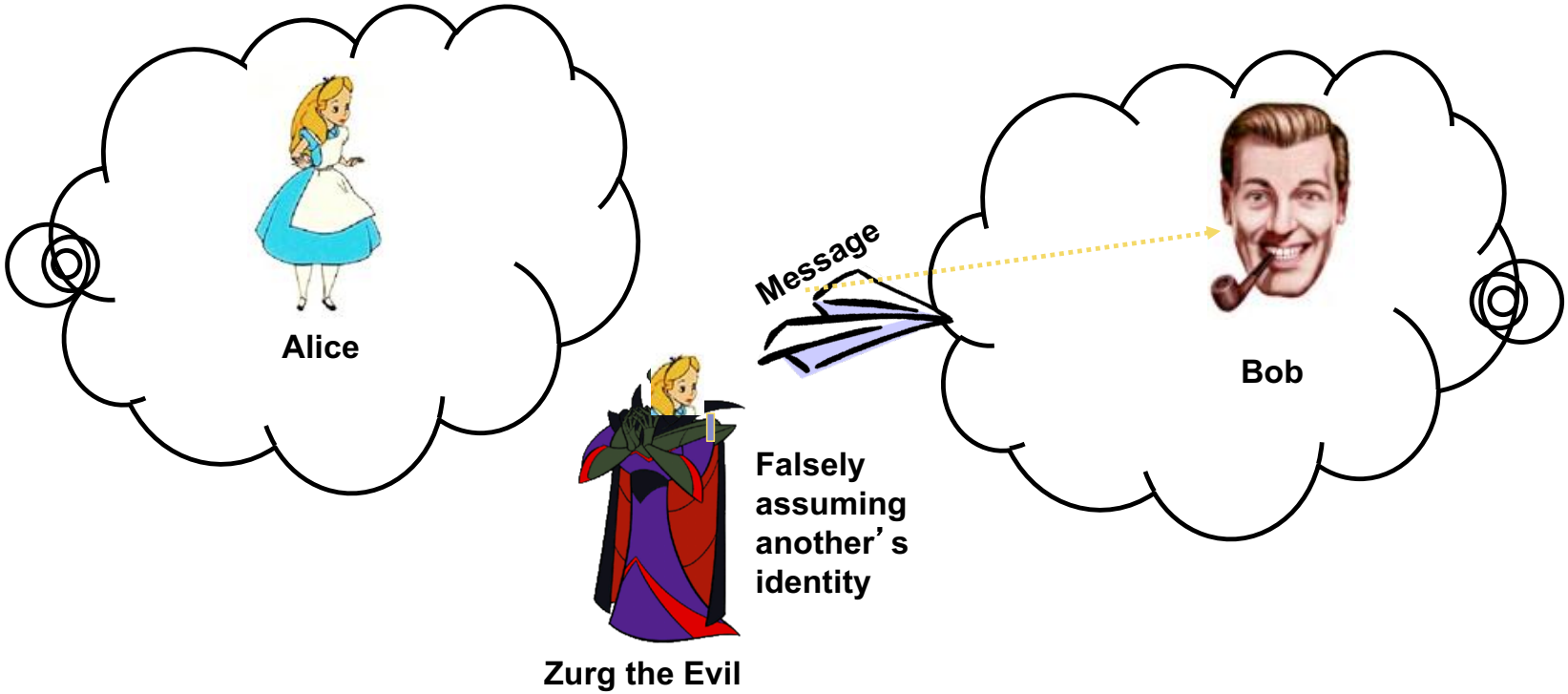


Integrity (& Authenticity)

- Prevention of impersonation of another principal
  - aka origin integrity
- Example
  - Getting money from someone else's bank account using their credentials
- What are some other examples?

Authenticity

- Identification and assurance of origin

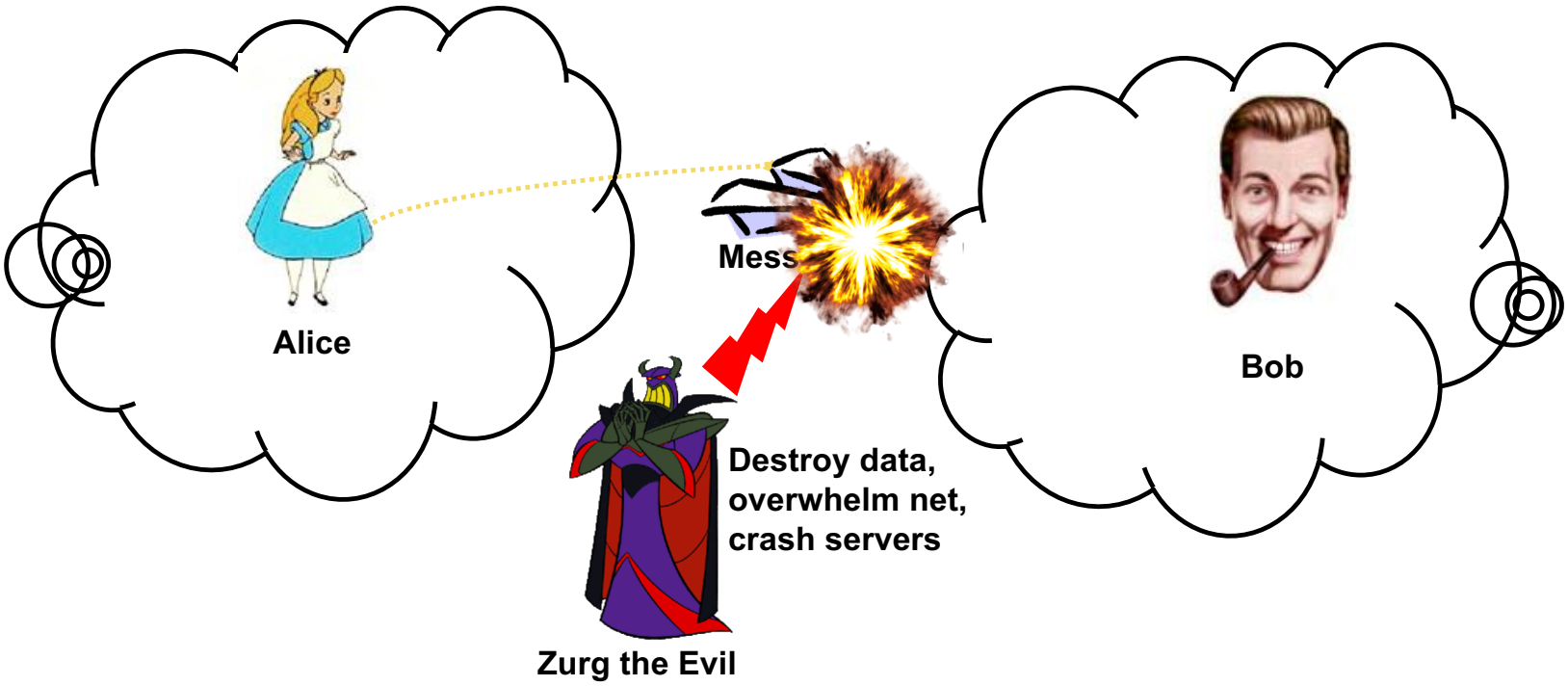


Availability

- Prevention of unauthorized denial of service to others
  - Unauthorized parties can't deny access to system for other users
- Examples
  - Physically rendering ATM unusable
  - Network denial of service attacks
- What are other examples of denial of service?

Availability

- Ability to use information or resources desired



C.I.A. + Privacy

- Privacy: A person's right or expectation to control the disclosure of his/her personal information, including activity metadata
- What is the difference between privacy and secrecy?
  - Secret from whom?
  - Activity metadata
    - What can you figure out about a person just from their location history?

C.I.A. + Privacy

- Which security property is violated if someone …
  - unplugs your alarm clock while you're sleeping?
  - changes the time on your alarm clock?
  - installs a camera in your room?

Vulnerabilities

- Weaknesses that could be exploited to cause damage to assets (typically where CIA + Privacy are not enforced as intended)
  - Default password left intact ("password")
  - Implementation flaws in software
  - Debug interface left open to network
  - Cryptography based on weak keys
- In particular, look to *assumptions* in system

- Known vulnerabilities (Bugtraq, full-disclosure, NVD) vs unknown vulnerabilities (0-days)

Threat Model

- ***Threat Model*** defines ***Security Goals***: What are we trying to protect from whom?
  - ***Assets*** and ***Attackers***
    - Defined by ***Capability*** and ***Intent***

Attackers/Adversaries

- Types
  - Individual
    - Outsider
    - Insider
    - Trusted/Privileged Insider
  - Group
    - Ad hoc
    - Established
  - Organization
    - Competitor
    - Supplier
    - Partner
    - Customer
  - Nation-State
- Capabilities
  - Time
  - Money
  - Training
  - Access
- Motivation/Intent
  - Curiosity
  - Fame
  - Money
  - National interest

Trust Computing Base (TCB)

- Trusted Computing Base
  - Set of systems/components/people/entities that your security depends on
  - Security dependency
- Trusted != Trustworthy
- Reflections on Trusting Trust

Trust/Security Boundary & Attack Surface

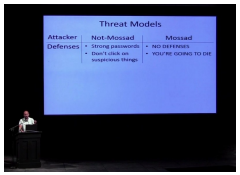
- Security Boundary
  - Perimeter around components of same trust level
  - Any data or signals coming in from outside is untrusted and potentially malicious
- Attack Surface
  - Set of interaction points across a security boundary
  - Parts of your system handling input from or otherwise interacting with less trusted and potentially malicious entities
  - Port, IPC, API, parser, etc.
  - Some highly sensitive systems are "air-gapped" to minimize the attack surface

Threat Model

- Your very first question in any security discussion should be

What's the threat model?

- Do not argue about attacks or defenses without understanding the threat model
- The threat model defines the problem to be solved.
  - If there is no consensus on the problem, there will be no consensus on solutions



https://vimeo.com/95066828#t=976s

Threat Model Example: Personal Items

- Asset
  - TV, jewelry
- Trust Boundary
  - Spouse, roommate
- Attacker
  - Roommate, thief
  - Do you have to worry about nation state attackers?



Threat Model

- A cautionary note
  - Your threat model is your problem scope.
  - Attackers don't care about your threat model.
  - Just because an asset or an attacker are outside your threat model, does not mean that they do not exist.
    - It just means that you have explicitly decided that you will not address them in your solution.

Risk Assessment

- Now that we know what we want to protect and from whom, we can reason about what risks attackers can pose to our assets
- Security is rarely binary
- Everything has some measure of risk
  - A rough, informal scale: from Very Low to Very High
  - Calculated as some combination of:
    - Likelihood – the probability that a security threat will materialise
    - Impact – the adverse affects that will occur if a threat does materialise
- We evaluate security risk relative to value
  - Is this risk worth taking?
  - Similar to health and financial risks

Risk Assessment (idealized)

- Risk assessment/risk analysis process for assessing risk in a system:
  1. Start by understanding system requirements
  2. Identify assets and attackers
  3. Establish security requirements
  4. Evaluate system design
  5. Identify threats and classify risks
  6. Address identified risks

1. Understand system requirements

- You can't make a system secure if you don't know what it is supposed to do
- What is the system supposed to do?
  - What is the security boundary?
  - Look for vague or open-ended requirements. These will likely be interpreted in different ways by different architects and developers.
- What is the lifecyle? How will it be deployed?

2. Identify assets and attackers

- Who are the stakeholders?
  - Who has a vested interest in the system running correctly/securely.
- What needs to be protected?
  - What is the value? To whom?
- From whom does it need to be protected?

3. Establish security requirements

- Think about the CIA+P classification
- "An attacker must not be able to compromise the [Confidentiality | Integrity | Availability] of…"

4. Review system design

- You can't make a system secure if you don't understand how it works
- What are the different components in the system?
  - How will the components communicate?
  - How will they store and/or pass data?
- Draw a block diagram of the system
  - Indicate security boundaries and information flow
  - Describe protocol interactions
- Again, look for ambiguity and hidden assumptions
- This is only the starting point
  - The design will evolve as risks are identified and mitigated
- Later, make sure that implementation matches the design

5. Identify threats and classify risks

- Using the security goals and current design, identify ***Threats*** to security
  - Possible ways in which attackers (as defined by your threat model) could exploit vulnerabilities in system design or implementation to compromise the assets (as defined by your threat model)
- For each threat, determine likelihood and impact and convert to a risk "score"

Identifying Threats

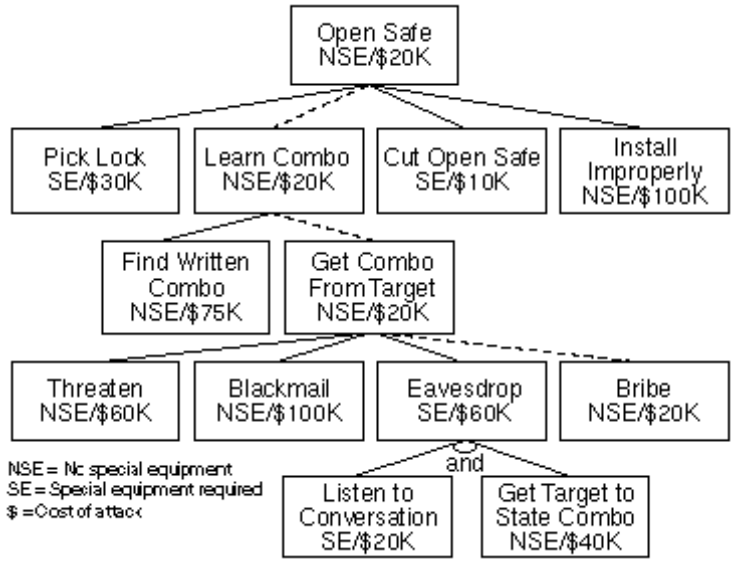
- Adopt an adversarial mindset
  - An attacker does not have to obey any rules established by the defender.
    - Requirements are for builders, not breakers.
    - It doesn't matter what is *supposed* to happen.
      - Review the code to see what actually happens.
  - Assume that the untrusted parts of the system work in the worst possible way – i.e. assume adversarial control.
    - That includes any external components, like storage, etc.
  - Look for assumptions about data coming across the security boundary and try to violate them.
    - If the documentation says that commands have to have a certain format, then evaluate what can happen when the format is incorrect.
    - If the documentation says that some commands are supposed to be executed in a certain order, then evaluate what can happen when the order is wrong.

Identifying Threats

- Strategies
  - Start with the assumption that there is yet another way to attack the system
    - You just need to find it
  - Attack the weakest link
    - Focus on parts that are ambiguous or seem overly complex.
    - Look for ambiguity in requirements and design and see if different parts of the system were implemented using different interpretations.
  - Be curious, ask stupid questions.
  - Consider the security goals one-by-one and look for different types of attacks against that goal.
    - Attack Trees
  - Consider the types of attack one-by-one and look for different ways that these attacks can affect a goal.
    - STRIDE

Identifying Threats

- Attack Trees
  - Start with an attacker's goal as the root of a tree
    - "Read secret …" or "Modify value…"
  - Different ways of achieving a goal (sub-goals) become child nodes
  - Can be annotated with likelihood, cost, etc.



https://www.schneier.com/academic/archives/1999/12/attack_trees.html

Identifying Risks

- STRIDE: Microsoft-developed model
  - Start with possible attack types and evaluate how they can violate security goals
  - Spoofing identity.
    - An example of identity spoofing is illegally accessing and then using another user's authentication information, such as username and password.
  - Tampering with data.
    - Data tampering involves the malicious modification of data. Examples include unauthorized changes made to persistent data, such as that held in a database, and the alteration of data as it flows between two computers over an open network, such as the Internet.
  - Repudiation.
    - Repudiation threats are associated with users who deny performing an action without other parties having any way to prove otherwise—for example, a user performs an illegal operation in a system that lacks the ability to trace the prohibited operations. Nonrepudiation refers to the ability of a system to counter repudiation threats. For example, a user who purchases an item might have to sign for the item upon receipt. The vendor can then use the signed receipt as evidence that the user did receive the package.
  - Information disclosure.
    - Information disclosure threats involve the exposure of information to individuals who are not supposed to have access to it—for example, the ability of users to read a file that they were not granted access to, or the ability of an intruder to read data in transit between two computers.
  - Denial of service.
    - Denial of service (DoS) attacks deny service to valid users—for example, by making a Web server temporarily unavailable or unusable. You must protect against certain types of DoS threats simply to improve system availability and reliability.
  - Elevation of privilege.
    - In this type of threat, an unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system. Elevation of privilege threats include those situations in which an attacker has effectively penetrated all system defenses and become part of the trusted system itself, a dangerous situation indeed.

https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx

Assessing Risk

- We can "score" identified risk based on combination of Likelihood and Impact:
- Risk assessment is subjective
  - Use your knowledge and common sense to make sure that the risk ranking is reasonable.

		Impact				
		Very Low	Low	Moderate	High	Very High
Likelihood	Very Low	Very low	Very Low	Very Low	Low	Low
	Low	Very Low	Very Low	Low	Low	Moderate
	Moderate	Very Low	Low	Moderate	Moderate	High
	High	Low	Low	Moderate	High	Very High
	Very High	Low	Moderate	High	Very High	Very High

https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final

6. Address identified risks

- Once risk is identified and classified we must decide what to do about it
- ***Avoid***: Remove component that creates the risk
  - Example: Disable the feature
- ***Mitigate***: Add measures that decrease likelihood or impact
  - Example: Better locks
- ***Transfer***: Make it someone else's problem
  - Example: Insurance
- ***Accept***: Do nothing
  - There's always residual risk we must accept
  - https://www.youtube.com/watch?v=9IG3zqvUqJY
    - Pwnie-nominated performance https://pwnies.com/

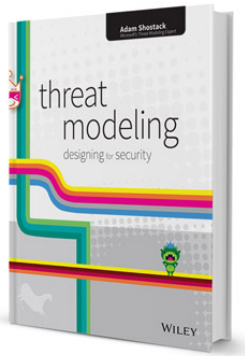


Risk Acceptance

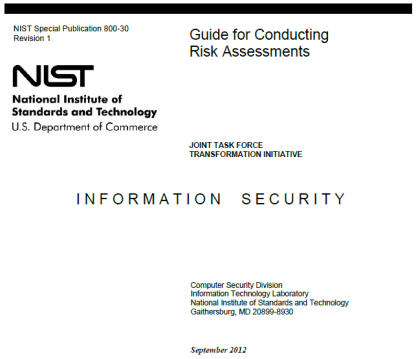
- It is impossible to offer functionality without risk.
- ***Residual risk*** is left over when all the controls and protections have been installed.
- Decision to accept risk should be made by someone who has sufficient authority to do so.
- *Accepting the risk does not make it go away.*
  - Regardless of why you choose to accept it
  - If there's no money or time to deal with a threat, does not make the risk disappear

Additional Resources

- *Threat Modeling: Designing for Security*
  - by Adam Shostack
  - https://threatmodelingbook.com/



- *NIST Guide for Conducting Risk Assessments*
  - SP 800-30 Rev. 1
  - https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final



Review

- ***Threat Model*** defines ***Security Goals***: What are we trying to protect from whom?
  - ***Assets***: ***Confidentiality***, ***Integrity***, and/or ***Availability (C.I.A.)*** of particular system, component, or data are preserved
  - ***Attackers***: ***Intent*** and ***Capability***
- Security is not binary, everything has some measure of ***Risk***
  - Calculated as some combination of ***Likelihood*** and ***Impact***
  - Is this risk worth taking?

Review

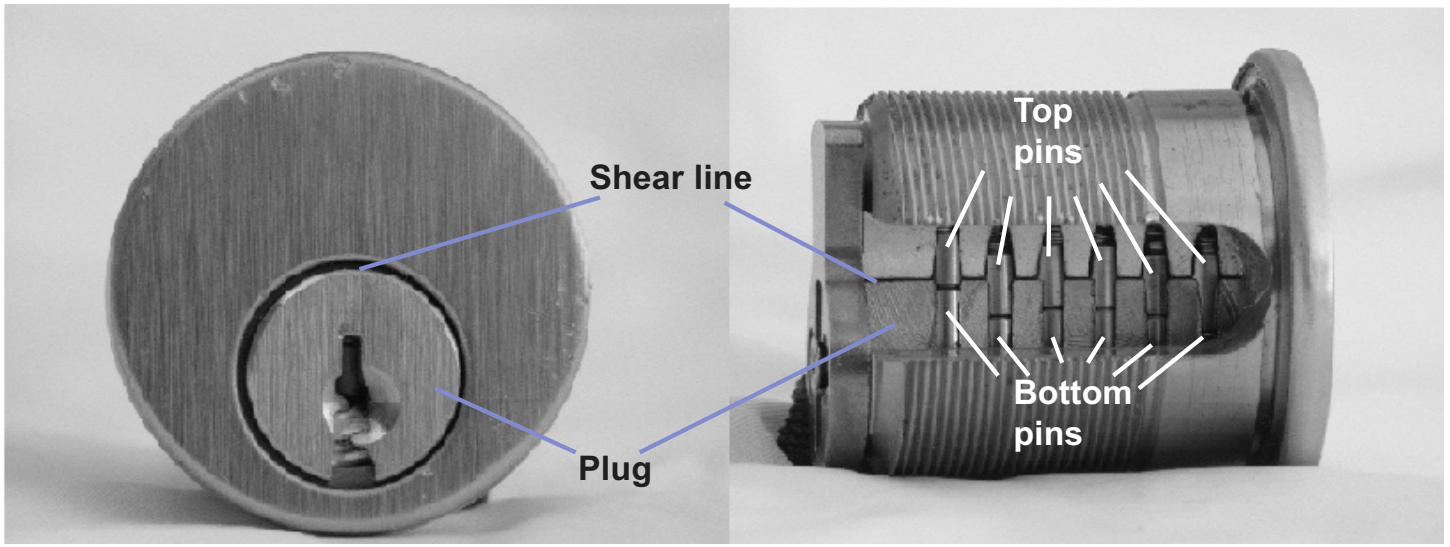
- Risk assessment/risk analysis process for assessing risk in a system.
- Adversarial Mindset
  - An attacker does not have to obey any rules established by the defender.
  - Assume that the untrusted parts of the system work in the worst possible way.
  - Violate assumptions.
  - An attack is possible, you just need to find it.
- Once risk is identified and classified we must decide what to do about it:
  - Avoid, Mitigate, Transfer, Accept

Bonus: using the adversarial mindset

- We depend on physical locks to secure our property

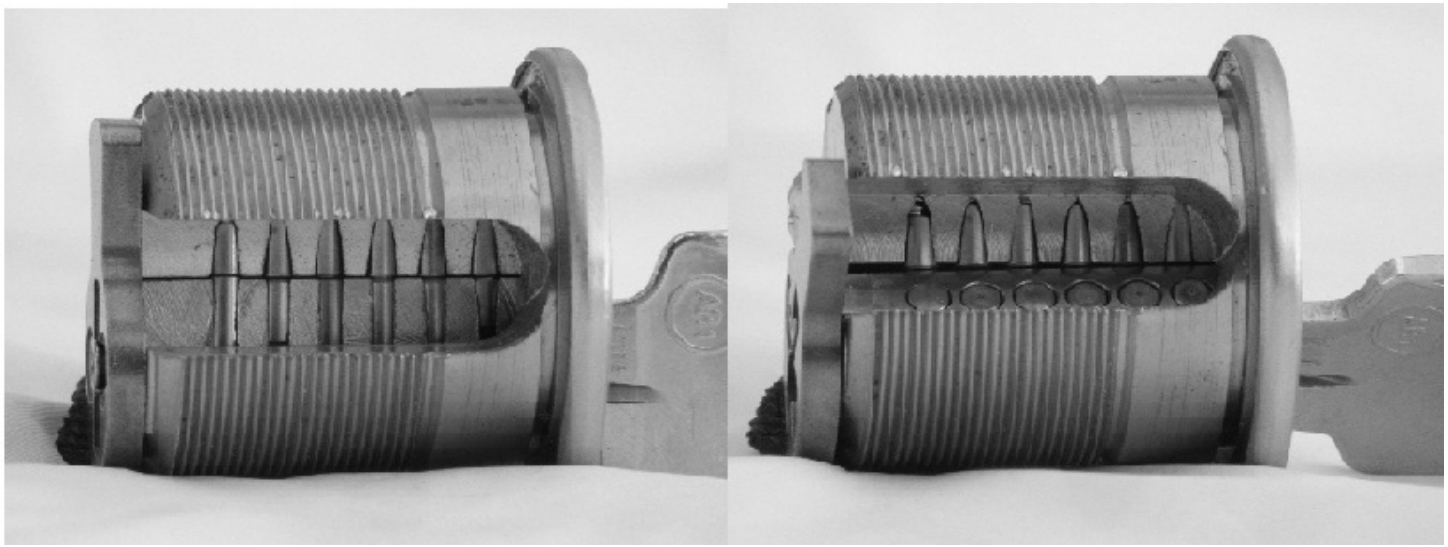


How physical locks work



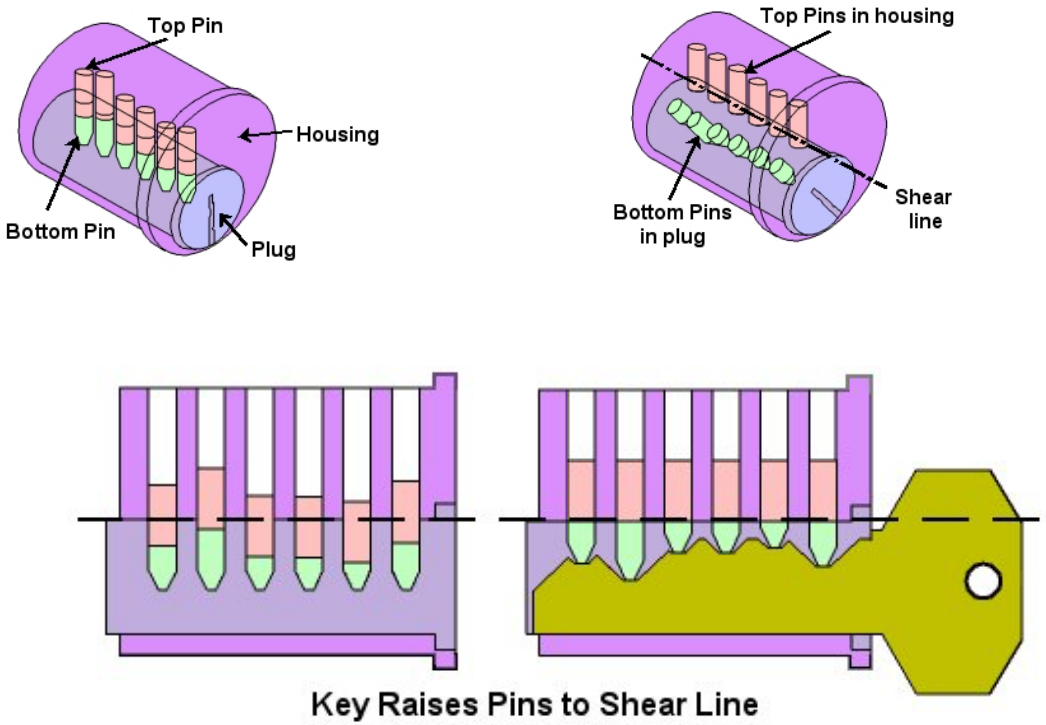
courtesy Matt Blaze

How physical locks work



courtesy Matt Blaze

Another visualization



Shared secrets

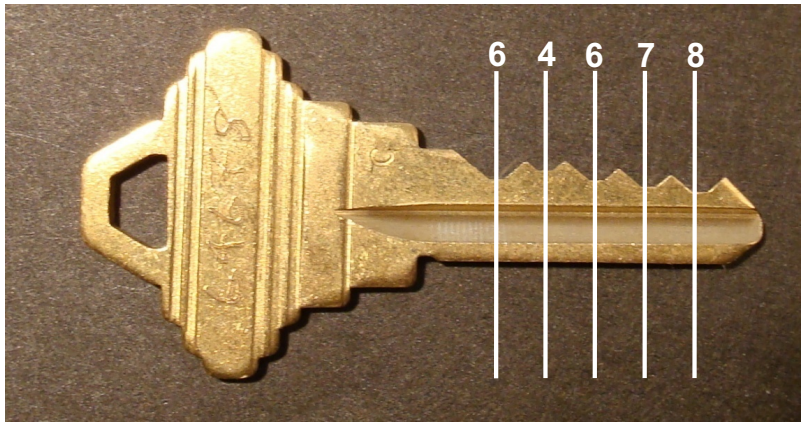
- There is a shared secret between the lock and the key… its shape



- In fact, it's a digital code

Bitting codes

- A key can be precisely described with a discrete code
  - Cuts at regular intervals (4-6 cuts)
  - Depth of cuts quantized in standard fashion (typically 6-9 bins)
  - 4-6 digits sufficient to describe most keys



Design assumptions

- If you don't know the secret code, you can't open the lock
- The secret code is secret
- If you can't open the lock, everything is fine

Design assumptions

- **If you don't know the secret code, you can't open the lock**
- The secret code is secret
- If you can't open the lock, everything is fine

Lock bypass via manipulation



Picking & Raking



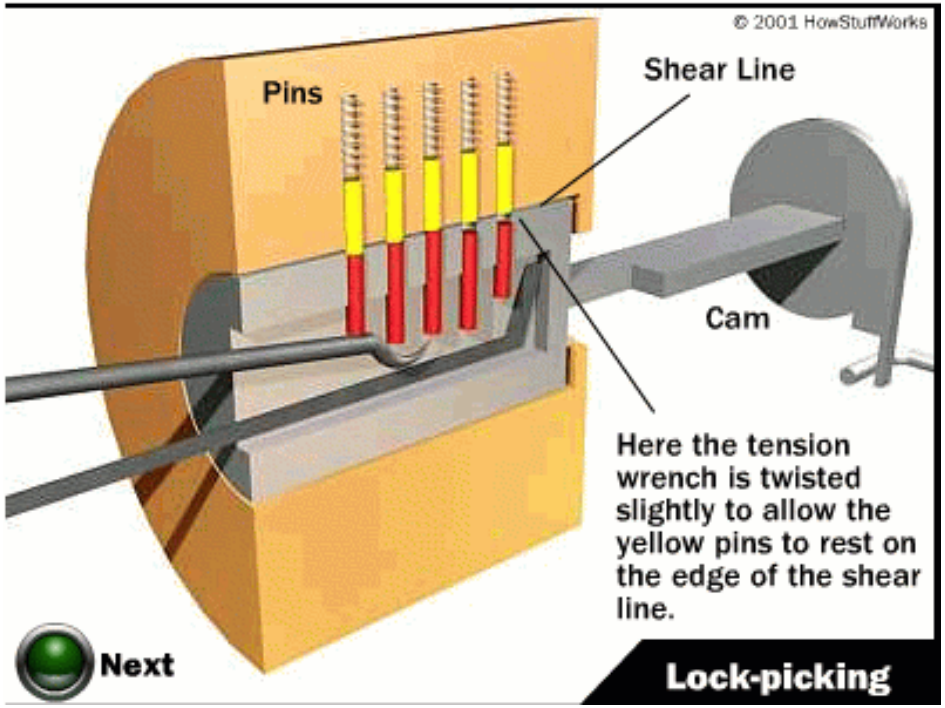
Bumping

Picking



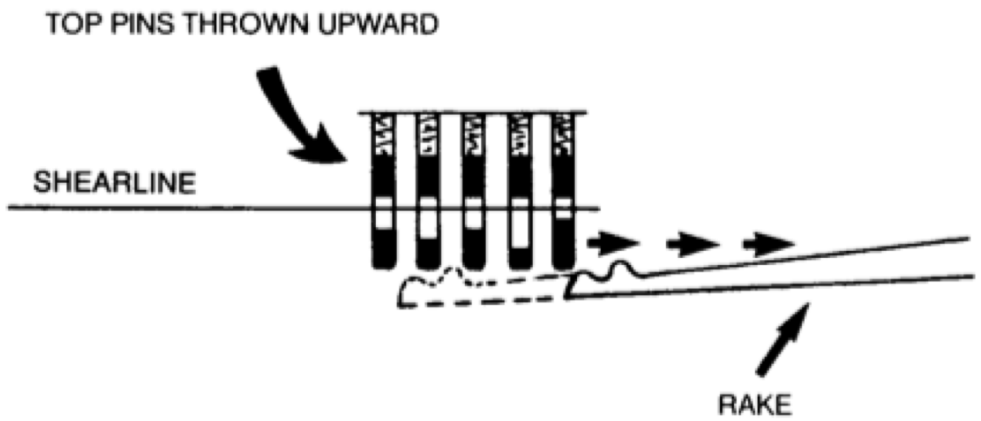
- Two parts
  - Tension wrench used to apply *slight* lateral force on plug
  - Pick used to lift individual bottom pins to the shear line
- Tension causes top pins to bind above shear line

Picking


© 2001 HowStuffWorks
Pins
Shear Line
Cam
Here the tension wrench is twisted slightly to allow the yellow pins to rest on the edge of the shear line.
Next
Lock-picking

Raking

- Similar idea, but less finesse…
- Rake pick moved in and out quickly imparts force to bottom pins; driver pins bind
- Quick & easy

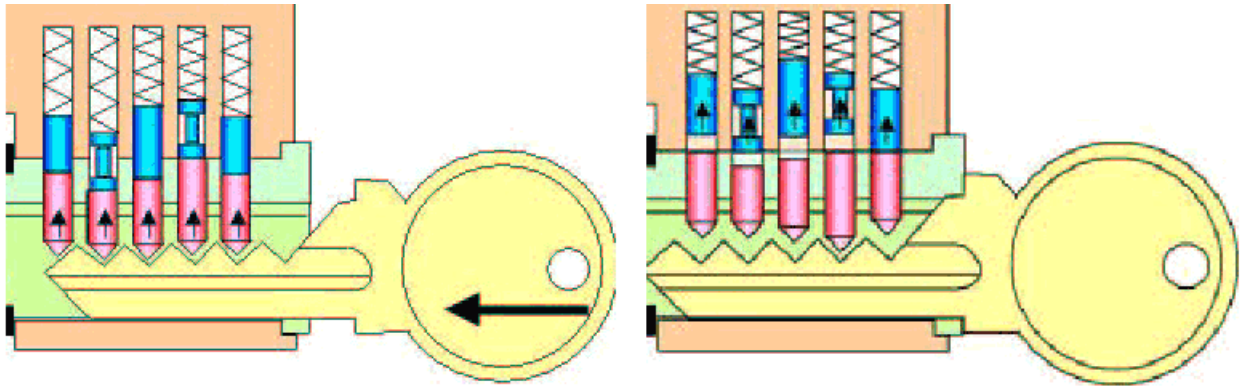


Bumping

- Similar idea to raking, but does all pins in parallel; super easy to do
- Max-depth key (bump key) used to impart force to bottom pins who transfer energy to driver pins (think billiards)



Bumping



Design assumptions

- **If you don't know the secret code, you can't open the lock**
- The secret code is secret
- If you can't open the lock, everything is fine

Design assumptions

- If you don't know the secret code, you can't open the lock
- **The secret code is secret**
- If you can't open the lock, everything is fine

Problem

- The bitting code is only secret if the key is kept secure
- What if I "borrow" your key?



Lock bypass via surreptitious duplication

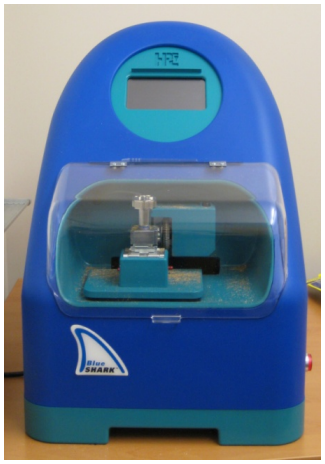


Field casting



Decoding

The power of decoding



Code key cutting machine

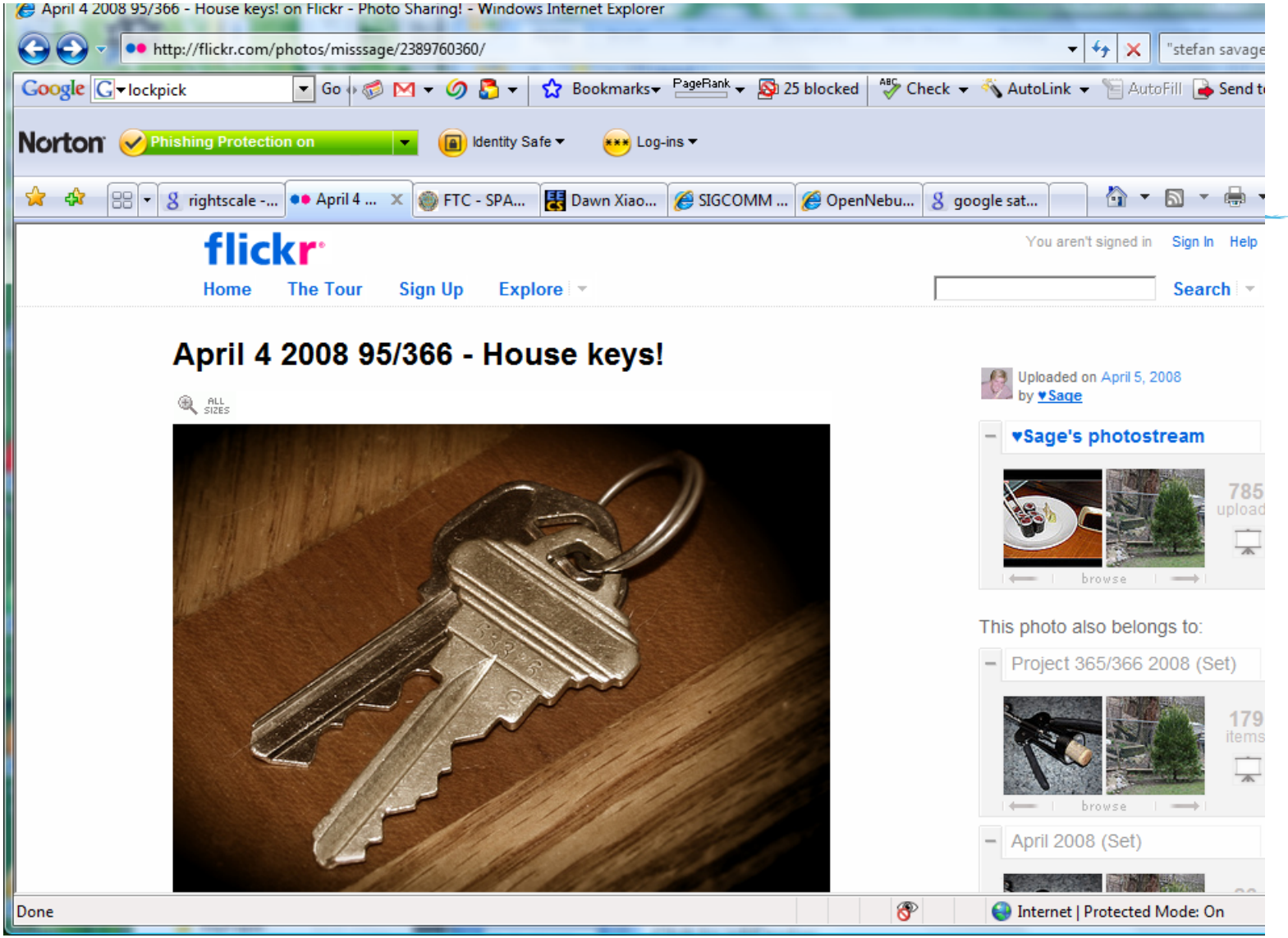
+ 

Key Blank

+ 64678 =



Key replica


April 4 2008 95/366 - House keys! on Flickr - Photo Sharing! - Windows Internet Explorer
http://flickr.com/photos/misssage/2389760360/
"stefan savage
Google
lockpick
Go
Bookmarks
PageRank
25 blocked
Check
AutoLink
AutoFill
Send t
Norton
Phishing Protection on
Identity Safe
Log-ins
rightscale -...
April 4 ...
FTC - SPA...
Dawn Xiao...
SIGCOMM ...
OpenNebu...
google sat...
flickr
You aren't signed in
Sign In
Help
Home
The Tour
Sign Up
Explore
Search
April 4 2008 95/366 - House keys!
ALL SIZES
Uploaded on April 5, 2008
by ♥Sage
♥Sage's photostream
785
upload
browse
This photo also belongs to:
Project 365/366 2008 (Set)
179
items
browse
April 2008 (Set)
Done
Internet | Protected Mode: On




5
3
3
8
6

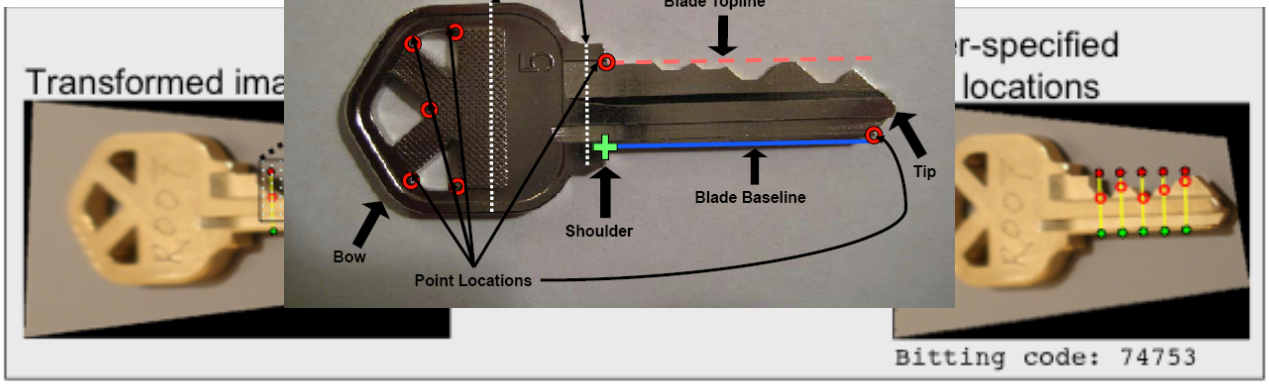
Optical decoding

- Decode keys semi-automatically from photos
- Traditional computer vision problem (photometry)
  - Normalize for scale and rotation



Sneakey: UCSD

- Reference key measured at control points
- User supplies correspondences between target key and reference image
- Image norm , cut locations identified ar es)



Works really well

- Almost perfectly from up close photos
  (e.g., cell phone cameras, etc)
- But that's no fun… what would James Bond do?


CELESTRON

Where's the Key?



Design assumptions

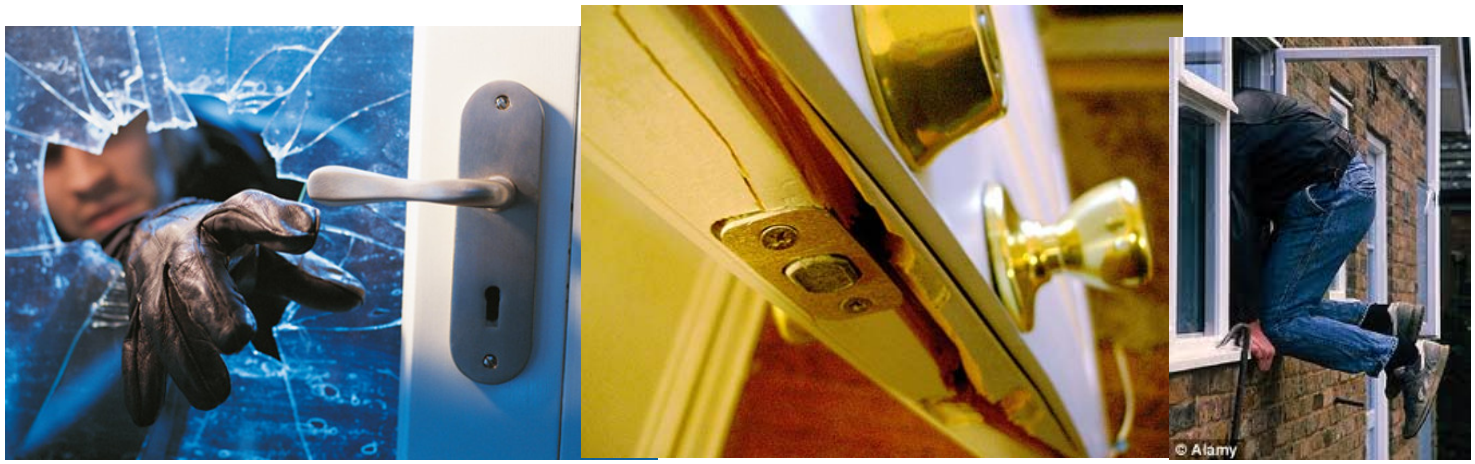
- If you don't know the secret code, you can't open the lock
- **The secret code is secret**
- If you can't open the lock, everything is fine

Design assumptions

- If you don't know the secret code, you can't open the lock
- The secret code is secret
- **If you can't open the lock, everything is fine**

Threat model example: home safety

- What is the threat?
  - Capabilities, resources, goals
  - Faster than the bear or faster than the next guy?
- What are all the ways the adversary might get access (the "attack surface")?



Aside: For those interested…

- Check out
  - Matt Blaze's work
    - *Safecracking for the Computer Scientist*
    - *Cryptology and Physical Security: Rights Amplification in Master-Keyed Mechanical Locks*
    - *Notes on Picking Pin Tumbler Locks,*
  - MIT Guide to Lockpicking
  - Locksport International (http://locksport.com/)
- However…
  - ***NEVER*** pick a lock you do not own
  - ***ALWAYS*** know the local law about using such tools

Homework

- First project is due next week (4/9 @ 10pm)
  - Getting comfortable with the debugger and project submission system
- Read *Smashing The Stack For Fun And Profit* by Aleph One
  - http://phrack.org/issues/49/14.html#article
- Read Chapter 6 from *The Craft of System Security*

Next Lecture…

Software exploitation: buffer overflows