Human Factors in Security

(a brief introduction to usable security and privacy)
(and, a short presentation on of my research projects)

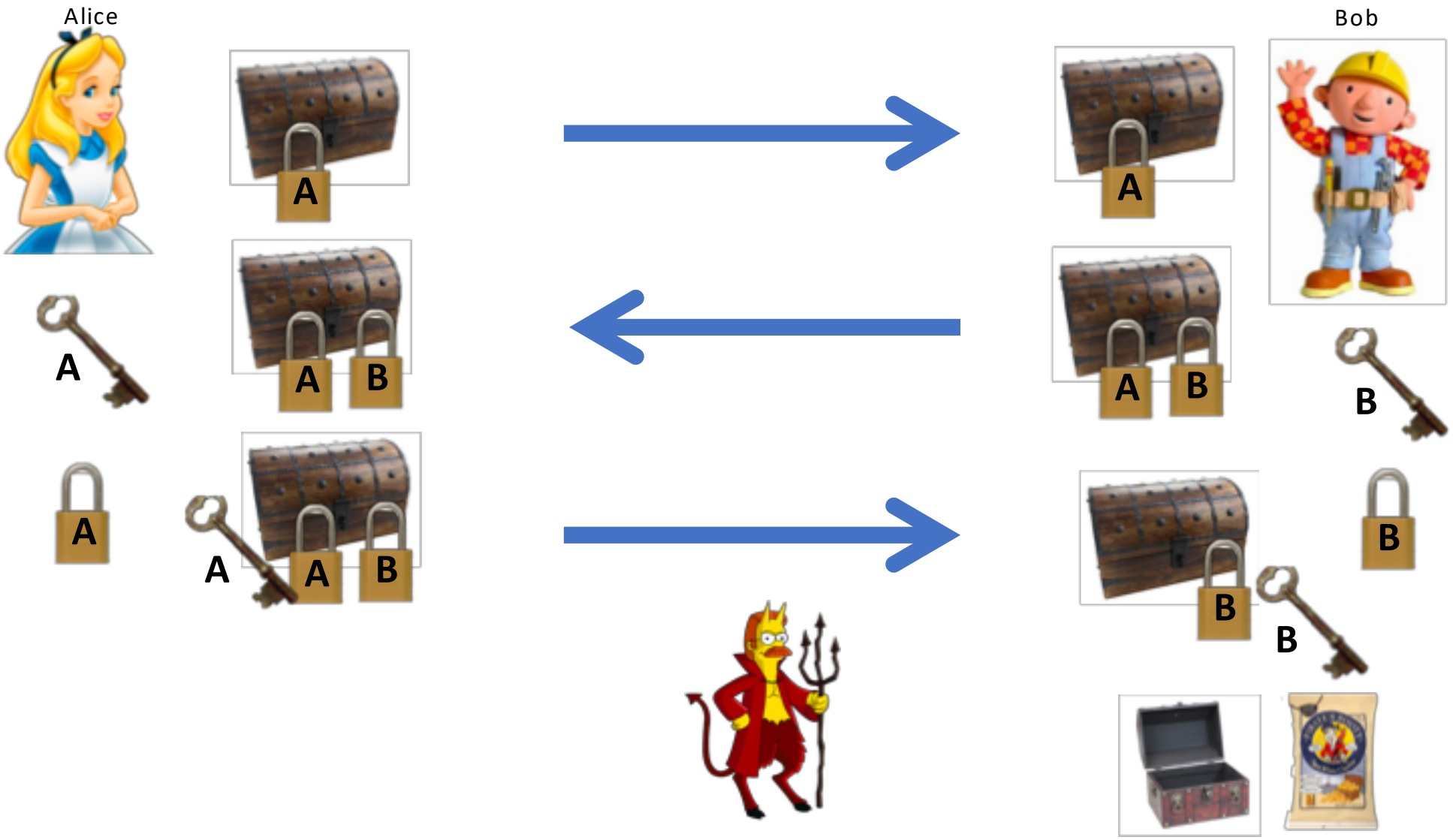
Adam J. Aviv
Asoc. Prof. of Computer Science
aaviv@gwu.edu / adamaviv.com

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

DEPARTMENT OF
COMPUTER SCIENCE
SCHOOL OF ENGINEERING & APPLIED SCIENCE

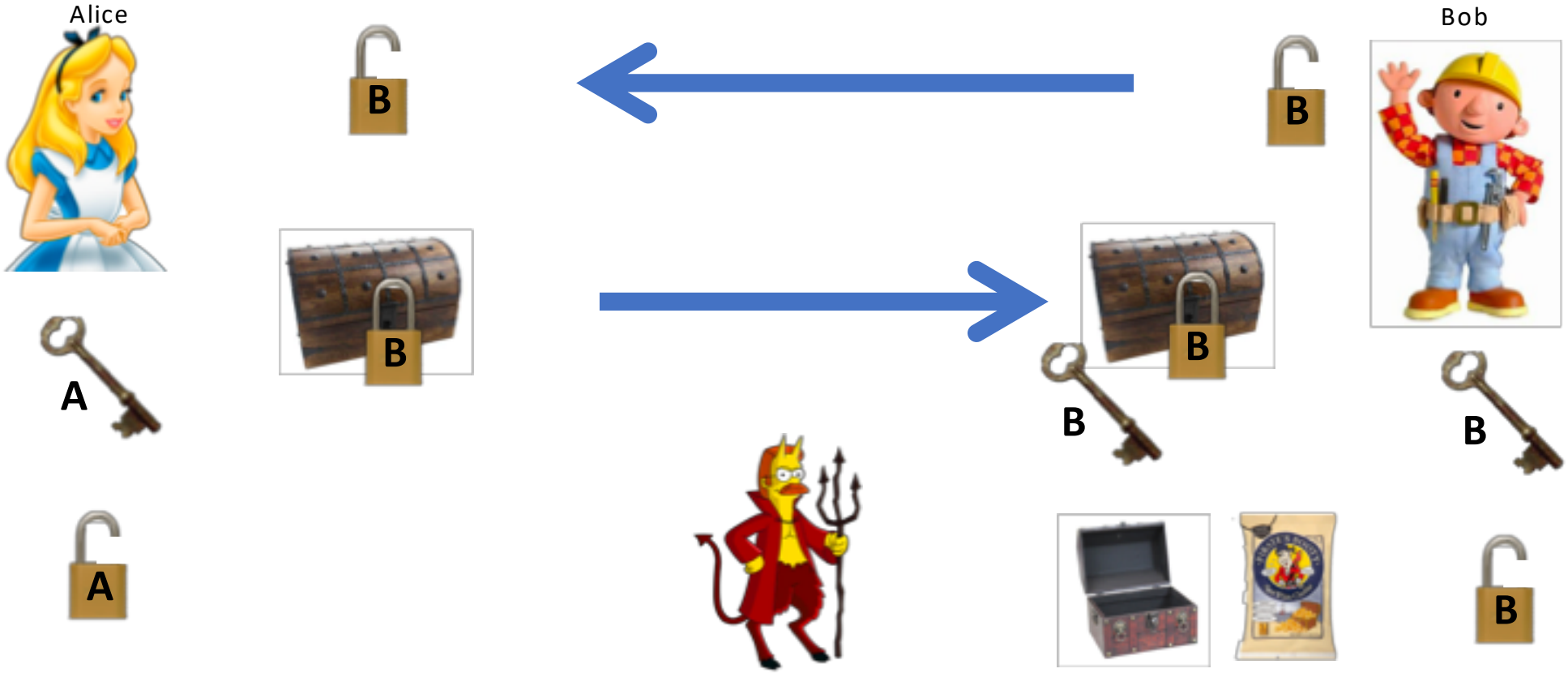
Computer Security is actually pretty amazing

- Example: Public Key Cryptography
  - Ability to encrypt a message you cannot decrypt!
  - Consider the alternative …


Alice
Bob
A
A
B
A
A
B
A
A
B
A
B
B
B
B
B

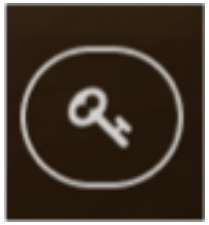
Public Key Cryptography

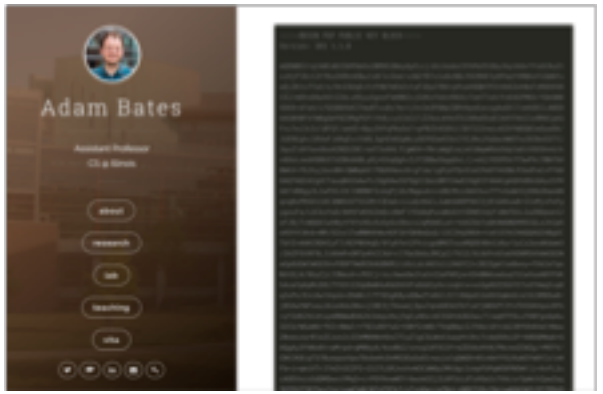
Amazing; that's it!
So simple…



Public Key Cryptography and Email

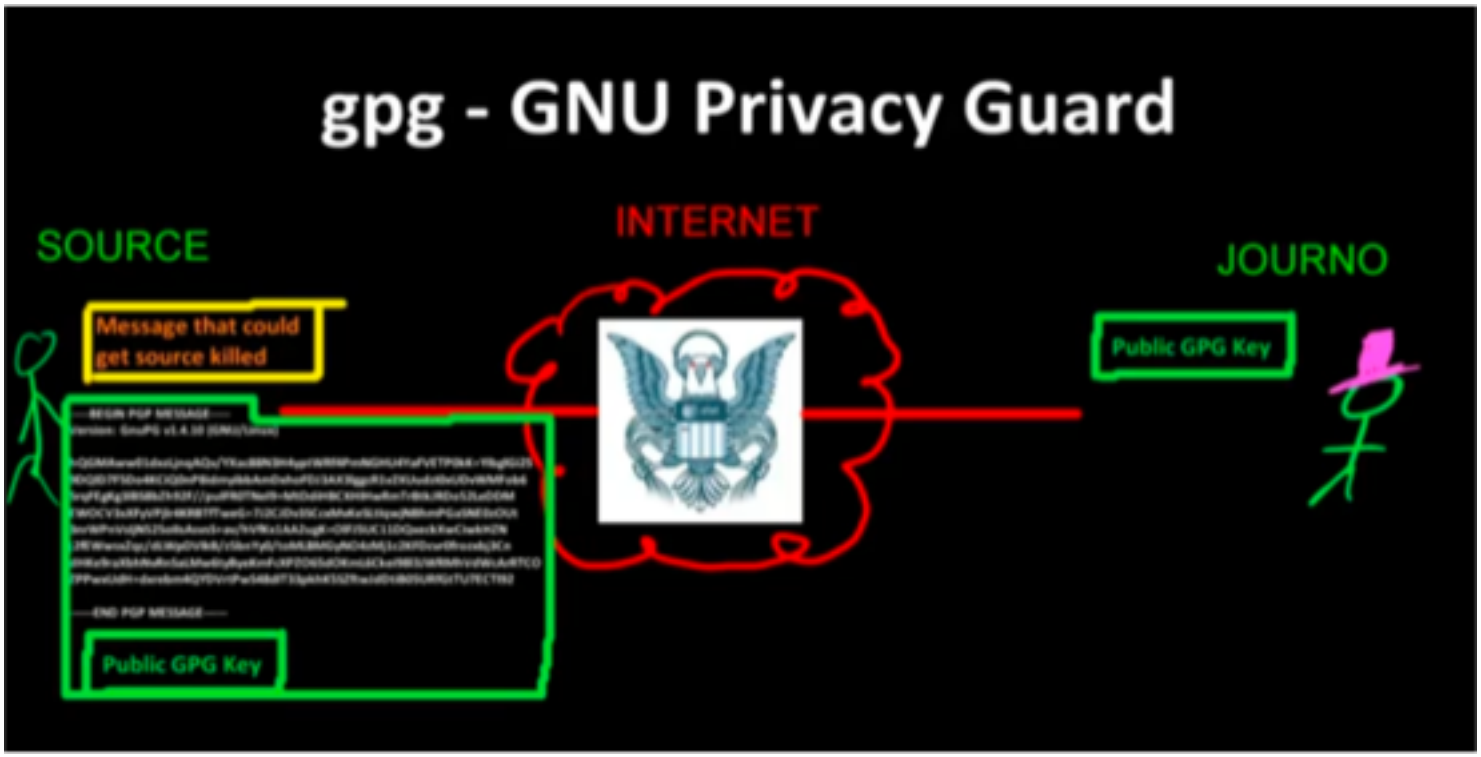
To: aaviv@gwu.edu
From: adamb@illinois.edu





So, then why couldn’t Glenn encrypt

Tutorial for GPG for Journalist



http://vimeo.com/56881481

“And yet, Greenwald still didn't bother learning security protocols. ‘The more he sent me, the more difficult it seemed,’ he says. ‘I mean, now I had to watch a f***ing video . . . ?’”

http://www.rollingstone.com/politics/news/snowden-and-greenwald-the-men-who-leaked-the-secrets-20131204
http://www.dailydot.com/politics/edward-snowden-gpg-for-journalists-video-nsa-glenn-greenwald/

Edward Snowden

Laura Poitras

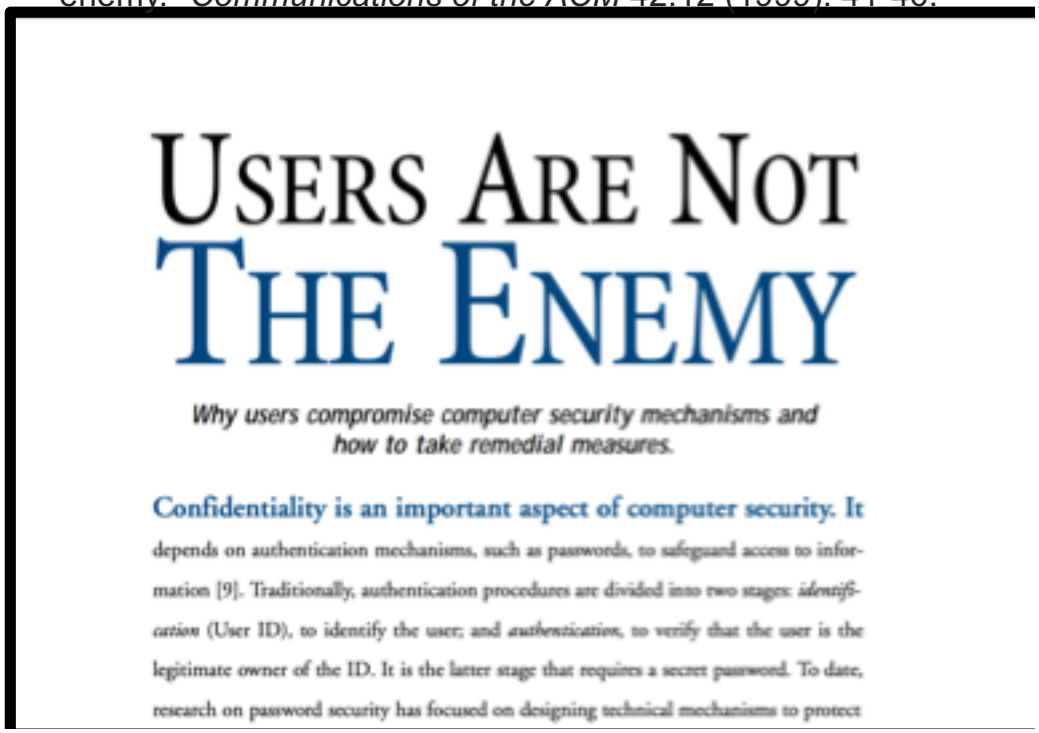
What made this so difficult for Glenn?

Is it even his fault?

Users are not the enemy!

- We have to design security systems with users in mind
- Users may be unable to complete security tasks for a variety of reasons, including the design of the system.

Adams, Anne, and Martina Angela Sasse. "Users are not the enemy." *Communications of the ACM* 42.12 (1999): 41-46.



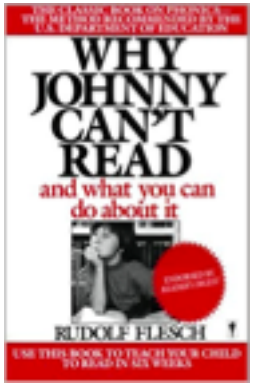
USERS ARE NOT THE ENEMY

Why users compromise computer security mechanisms and how to take remedial measures.

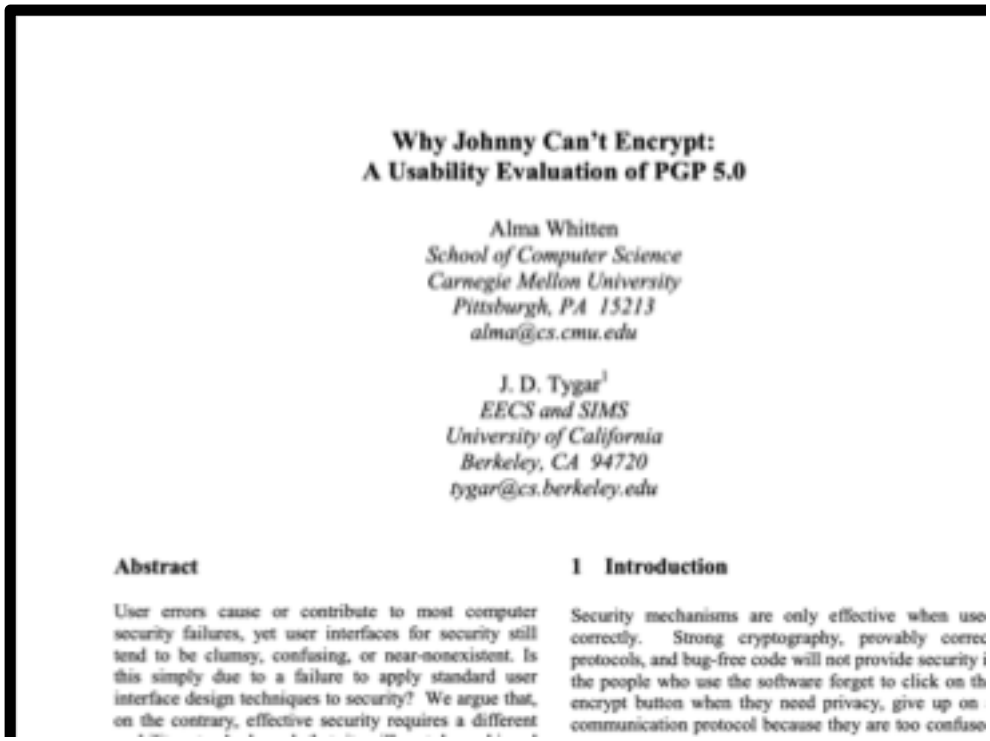
Confidentiality is an important aspect of computer security. It depends on authentication mechanisms, such as passwords, to safeguard access to information [9]. Traditionally, authentication procedures are divided into two stages: *identification* (User ID), to identify the user; and *authentication*, to verify that the user is the legitimate owner of the ID. It is the latter stage that requires a secret password. To date, research on password security has focused on designing technical mechanisms to protect

Why Johnny Can’t Encrypt

- Evaluated PGP 5.0 using a lab study with “novice” users
  - Cognitive walkthrough
- Defined a concept of “usable security” and evaluated it with a real system



Whitten, Alma, and J. Doug Tygar. "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0." *USENIX Security Symposium*. Vol. 348. 1999.



Why Johnny Can’t Encrypt:
A Usability Evaluation of PGP 5.0

Alma Whitten
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
alma@cs.cmu.edu

J. D. Tygar[1]
EECS and SIMS
University of California
Berkeley, CA 94720
tygar@cs.berkeley.edu

Abstract

User errors cause or contribute to most computer security failures, yet user interfaces for security still tend to be clumsy, confusing, or near-nonexistent. Is this simply due to a failure to apply standard user interface design techniques to security? We argue that, on the contrary, effective security requires a different

1 Introduction

Security mechanisms are only effective when use correctly. Strong cryptography, provably correc protocols, and bug-free code will not provide security i the people who use the software forget to click on th encrypt button when they need privacy, give up on communication protocol because they are too confuse

Usable Security for a Software System

***Definition*: Security software is usable if the people who are expected to use it:**

- ***Are reliably made aware of the security tasks they need to perform***
- ***Are able to figure out how to successfully perform those tasks***
- ***Don't make dangerous errors***
- ***Are sufficiently comfortable with the interface to continue using it***

Essentially, *users are not the enemy*, but rather the system is not usable!

What makes Usable Security Difficult?

- The unmotivated user
  - Security may be a secondary goal
  - “I just want to send this email!”
  - “I have to watch a f**ing video?!”
- Abstraction
  - Security policies and rules are often too abstract for many users
- Lack of feedback
  - How do you properly tell a user that they’ve done something securely, or insecurely?
- The barn door
  - “locking the barn door after the house is gone”
  - Ensuring high-risk mistakes are avoided by users
- Weakest Link
  - Everything can be great, but …

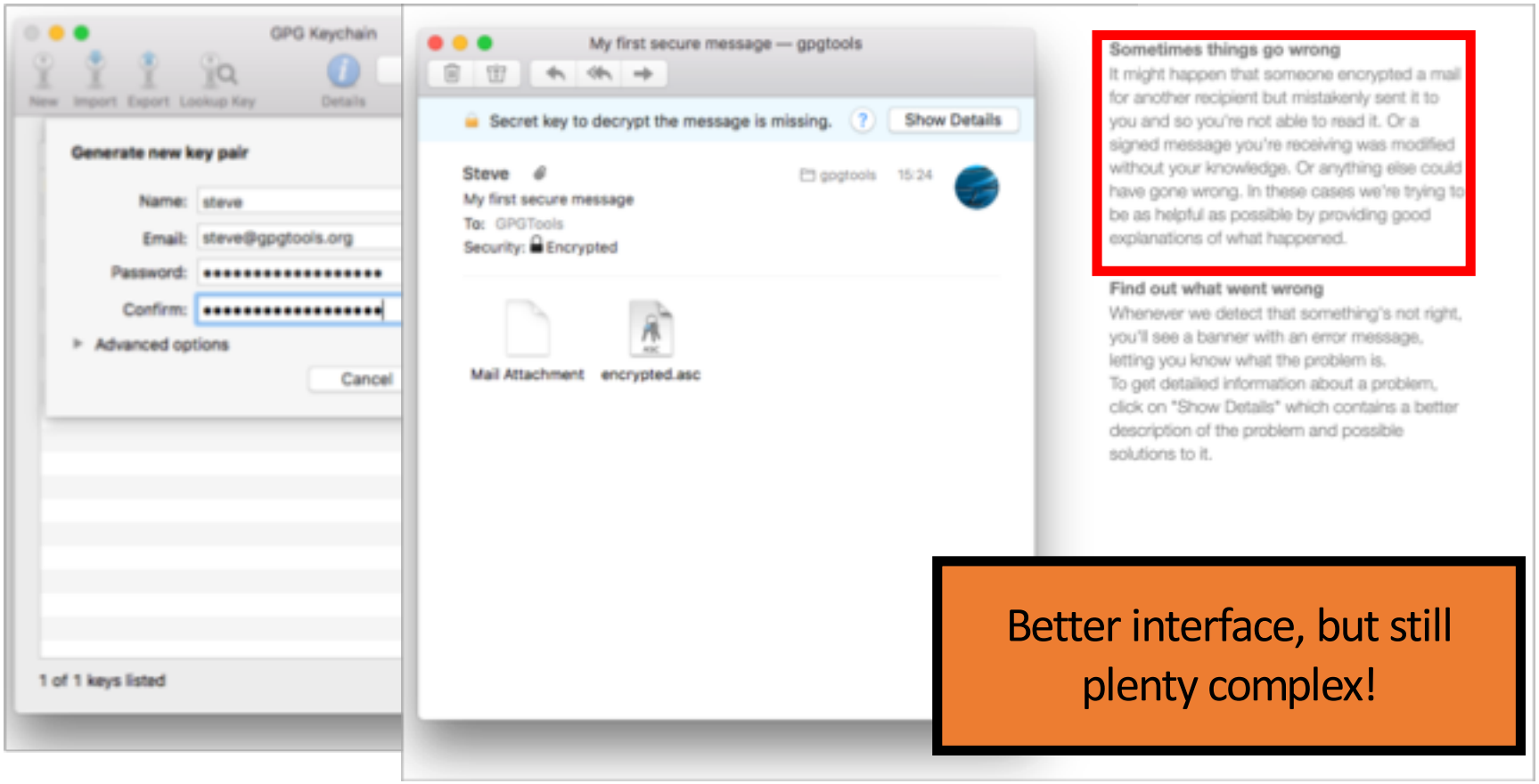
What is usability for PGP?

- Understand that **privacy is achieved by encryption**, and figure out how to encrypt email and how to decrypt email received from other people
- Understand that **authentication is achieved through digital signatures**, and figure out how to sign email and how to verify signatures on email from other people
- Understand that in order to sign email and allow other people to send him encrypted email, **a key pair must be generated**, and figure out how to do so
- Understand that in order to allow other people to verify his signature and to send him encrypted email, he must **publish his public key**, and figure out some way to do so
- Understand that in order to verify signatures on email from other people and send encrypted email to other people, he **must acquire those people's public keys**, and figure out some way to do so
- Manage to **avoid such dangerous errors** as accidentally failing to encrypt, trusting the wrong public keys, failing to back up his private keys, and forgetting his passphrases
- • Be able to succeed at all of **this within a few hours of reasonably motivated effort**

Usability for PGP is actually more work than it seems!

PGP Tools (circa, 1998)

GPG Tools (circa, 2019)



Better interface, but still plenty complex!

User Study (12 participants)

- Gave participants a test scenario
  - Campaign coordinator for a political campaign
  - Send updates to other campaign staff team by email using PGP for privacy and security
- PGP does not do email directly, so participants were provided with a separate email client
- User is tasked to send the campaign itinerary to a set of user names and emails of five team members

- To complete task …
  - Generate a key pair
  - Get the team members public keys
  - Make their own public key available
  - Type a short secret message into an email
  - Sign the email using their private key
  - Encrypt the message using the team member's public key
  - Send the resulting encrypting message to each of the team members

Remember, participants may be totally unfamiliar with PGP at the start of the study!

So, what happened?

- Three participants accidently emailed the secret message without encryption.
  - Two realized immediately afterwards that they did so
  - One thought that security was suppose to be transparent and the encryption had already taken place
- One participant forgot the passphrase to the key pair
  - Most participants choose standard passwords instead of passphrases
- One participant couldn't encrypt at all, two others took >25 min

Misunderstanding of the Public Key Model

- 7 of 12 participants ONLY encrypted emails using their own public key
  - Two eventually realized they needed to use the public key of the recipient
  - One ended up using only one of the public keys and set encrypted messages to wrong participants who couldn't decrypt it
- One participant generated key pairs for the other team members
- 10 of 12 could publish their public key
- 8 of 12 were able to obtain other's public key

How do we improve this situation?

- Better UX
- Better tools
- Better training
- Transparent encryption

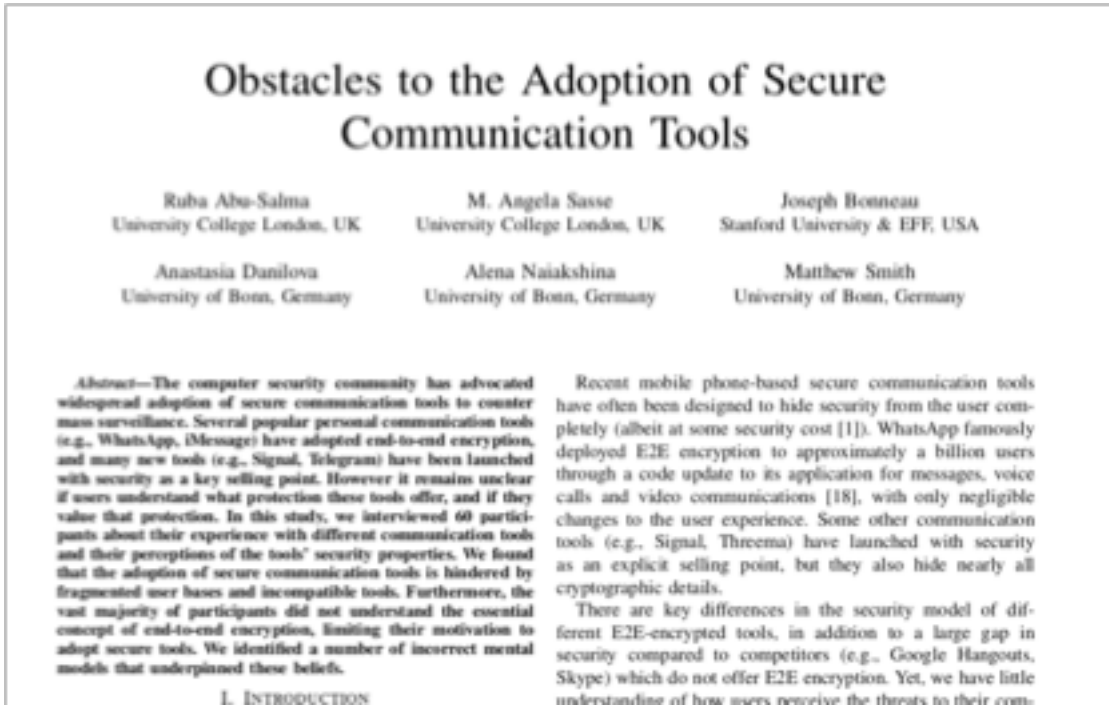
We use end-to-end encryption a lot of communications, but the process is nearly transparent to the end user.

Why not use these services all the time?

- Qualitative, interview based study of how and why users choose (or choose not to) use secure messaging tools

- Findings in brief…
  - Usability not the main obstacle
  - Fragmented user base
  - Low QoS
  - Perceptions of futility
  - Low motivation

IEEE S&P 2017
(Oakland)



Obstacles to the Adoption of Secure Communication Tools

Ruba Abu-Salma
University College London, UK

M. Angela Sasse
University College London, UK

Joseph Bonneau
Stanford University & EFF, USA

Anastasia Danilova
University of Bonn, Germany

Alena Naiakshina
University of Bonn, Germany

Matthew Smith
University of Bonn, Germany

***Abstract*—The computer security community has advocated widespread adoption of secure communication tools to counter mass surveillance. Several popular personal communication tools (e.g., WhatsApp, iMessage) have adopted end-to-end encryption, and many new tools (e.g., Signal, Telegram) have been launched with security as a key selling point. However it remains unclear if users understand what protection these tools offer, and if they value that protection. In this study, we interviewed 60 participants about their experience with different communication tools and their perceptions of the tools' security properties. We found that the adoption of secure communication tools is hindered by fragmented user bases and incompatible tools. Furthermore, the vast majority of participants did not understand the essential concept of end-to-end encryption, limiting their motivation to adopt secure tools. We identified a number of incorrect mental models that underpinned these beliefs.**

I. Introduction

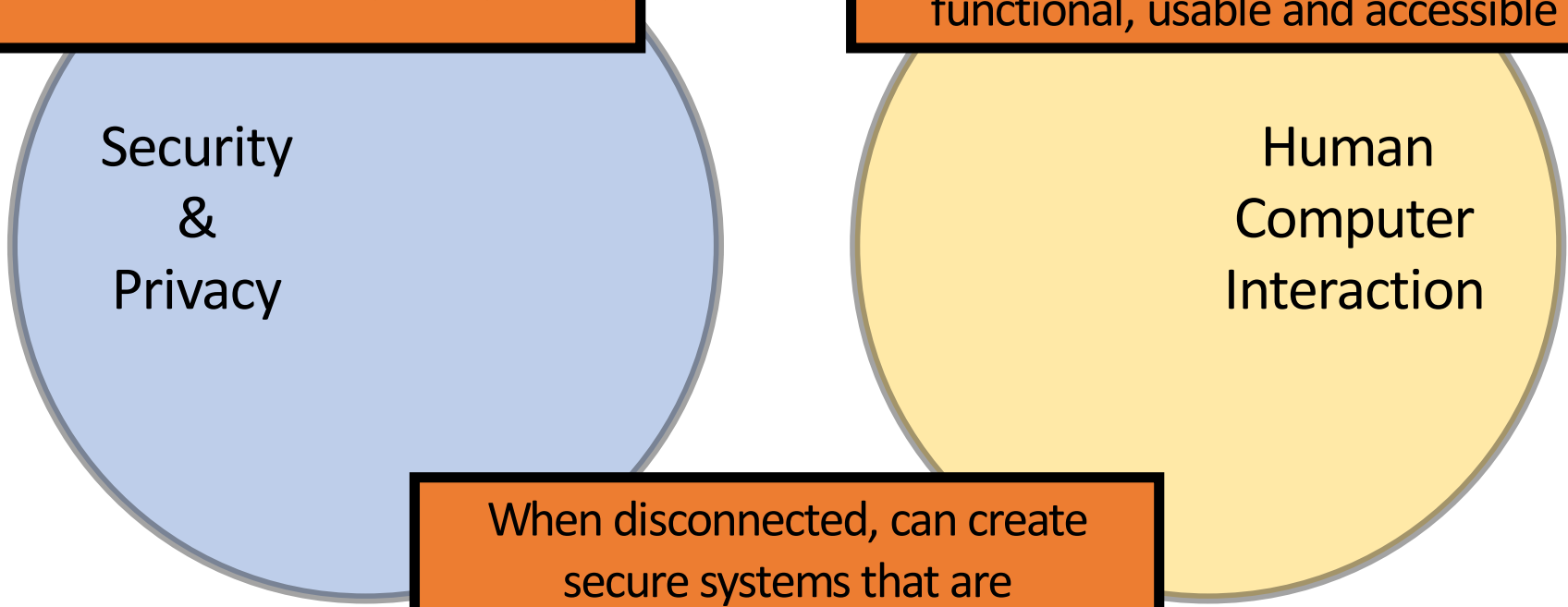
Recent mobile phone-based secure communication tools have often been designed to hide security from the user completely (albeit at some security cost [1]). WhatsApp famously deployed E2E encryption to approximately a billion users through a code update to its application for messages, voice calls and video communications [18], with only negligible changes to the user experience. Some other communication tools (e.g., Signal, Threema) have launched with security as an explicit selling point, but they also hide nearly all cryptographic details.

There are key differences in the security model of different E2E-encrypted tools, in addition to a large gap in security compared to competitors (e.g., Google Hangouts, Skype) which do not offer E2E encryption. Yet, we have little

The disconnect …

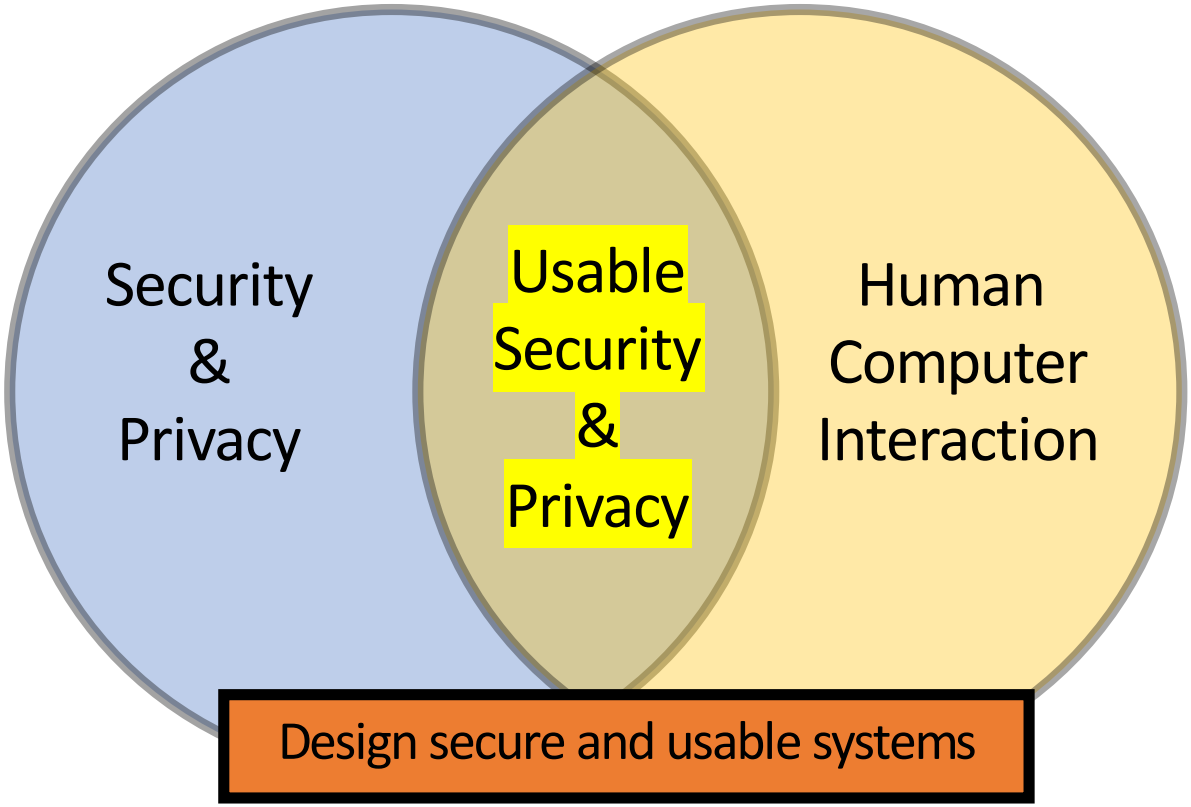
Design of secure systems that meet some definition of security and privacy

Understanding how humans/users use computing systems to make them more functional, usable and accessible



When disconnected, can create secure systems that are unusable/incomprehensible to users

Usable Security and Privacy

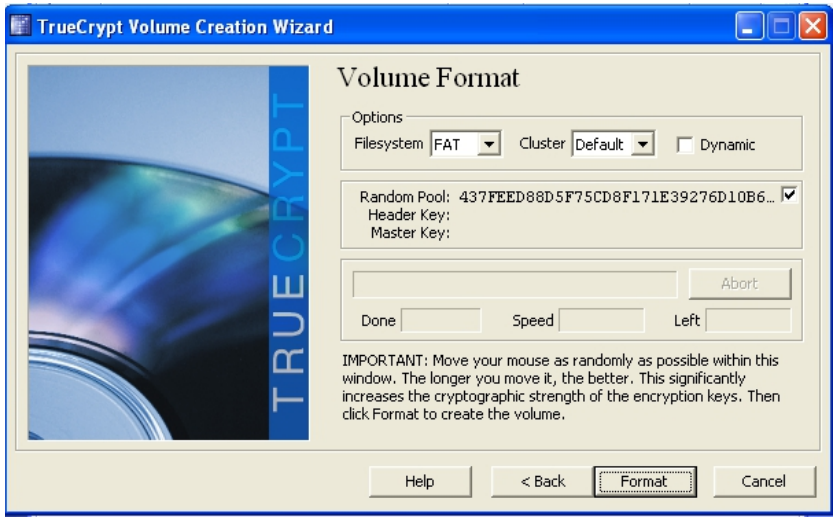


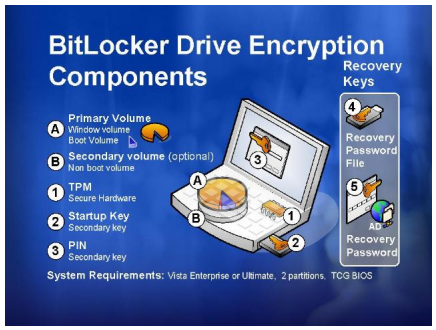
More reasons why usable security is hard

- Humans … (as we've seen)
  - And how they act in the presence of an adversary
- Usability is not enough. We also need systems that remain secure when:
  - Attackers try to fool users
  - Users behave in predictable ways
  - Users are acting under stress
  - Users are careless, unmotivated, busy

Usable Encryption

- Why don’t people encrypt their email or their sms or much of anything?
- Or maybe they do and don’t know it?





Passwords

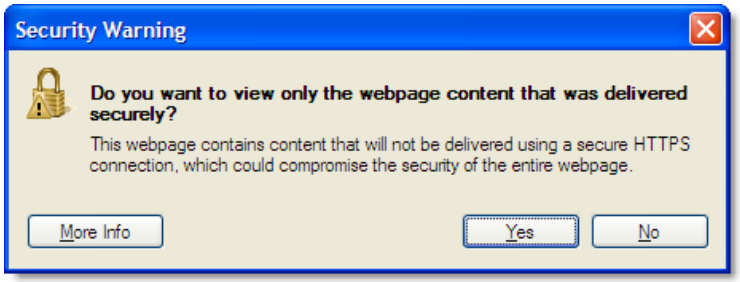
- Why do people choose such terrible, awful passwords?
- Can we help them to do better?

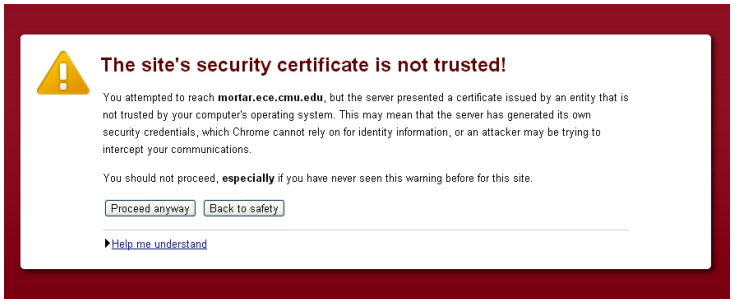
Phishing and Mental Models

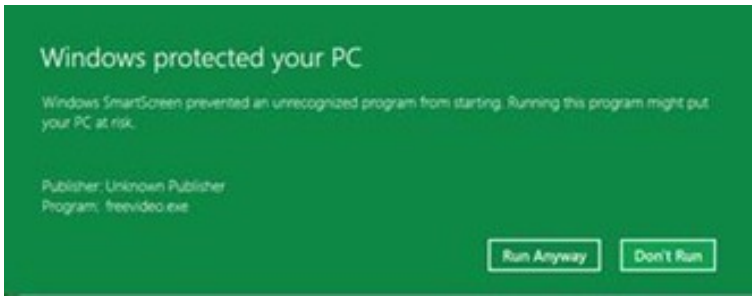
- How do non-technical people think about privacy and security, and how can we better support them?
  - Or, whatever happened to that Nigerian prince scam?
  - Or, how do we respond to telephone scams?

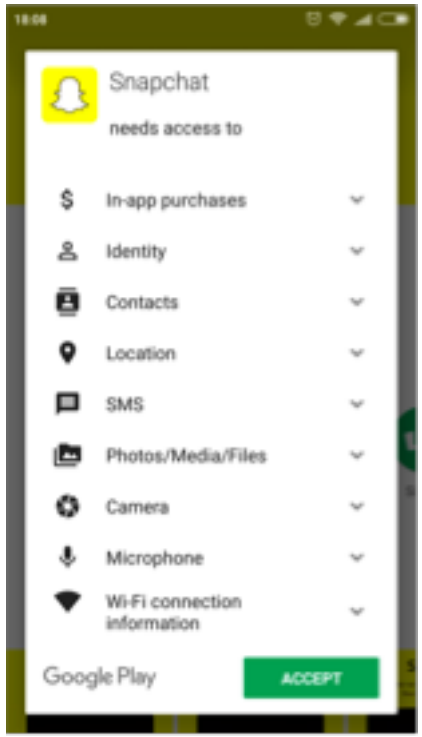
Security Warnings and Permissions

- Do you ever say no?



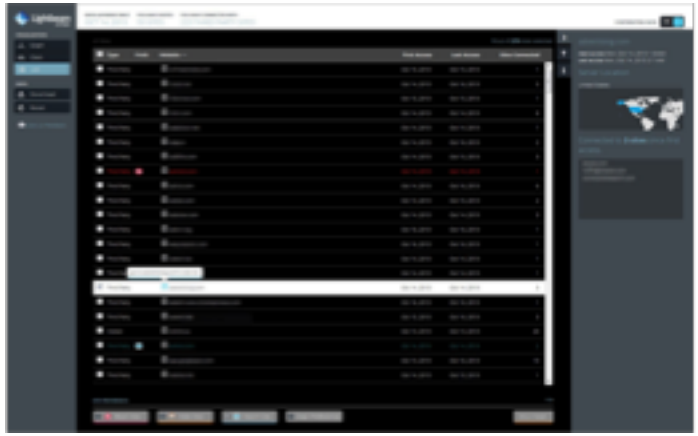






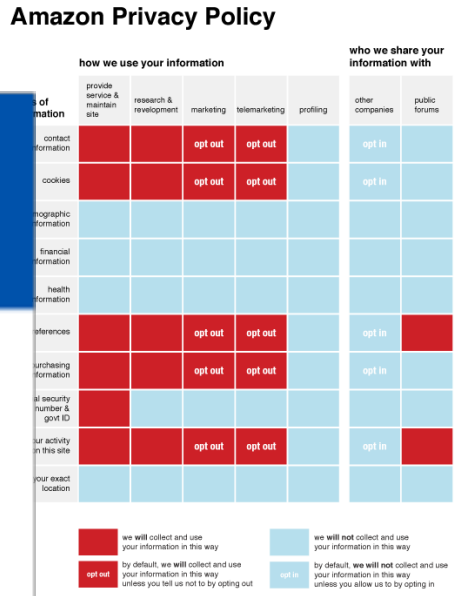
Web and Browser Security & Privacy

- How do we keep the web secure and private, and how do we keep users aware of what's happening as they browse?



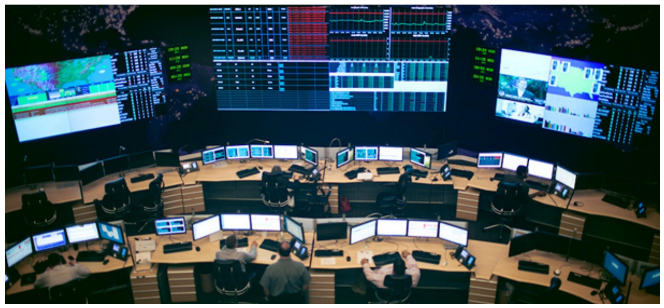
Privacy Notices

- How do we communicate about privacy critical information?



Developer Studies

- The developers are users too, and their choices matter for security.

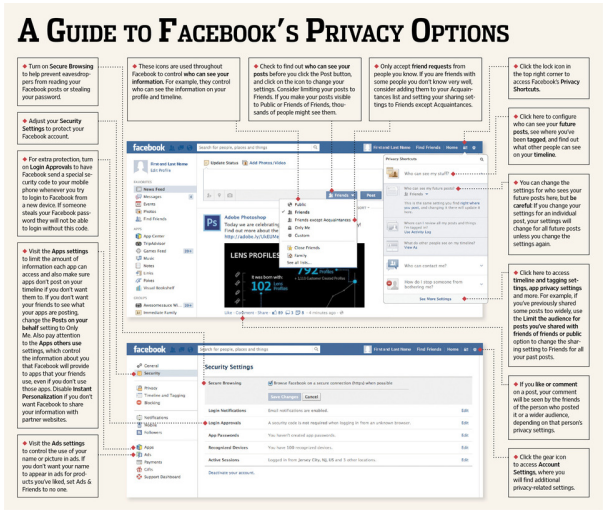


Underrepresented Groups

- How do we design security and privacy for all?
  - Age, Abilities, Culture
- How can technology affect some groups than others?

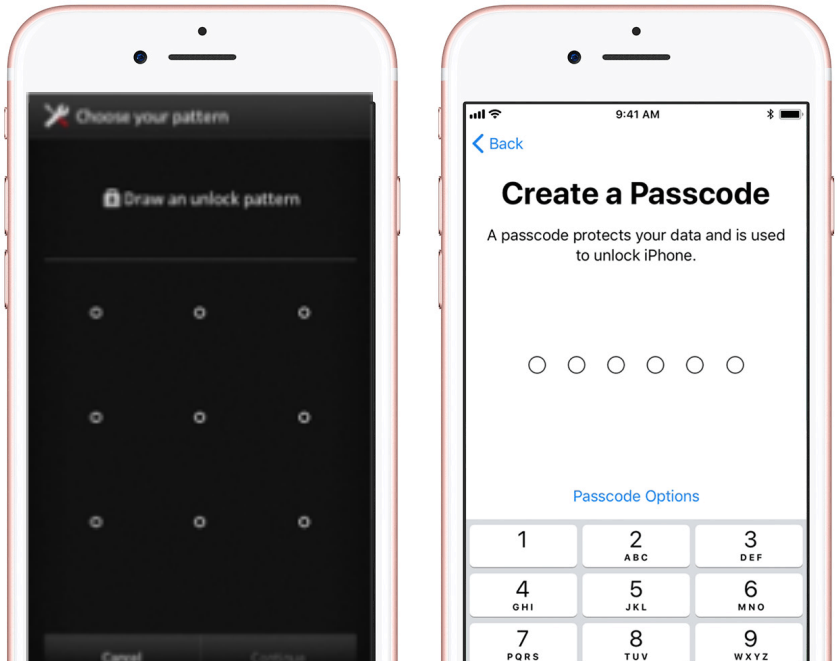
And it keeps going …

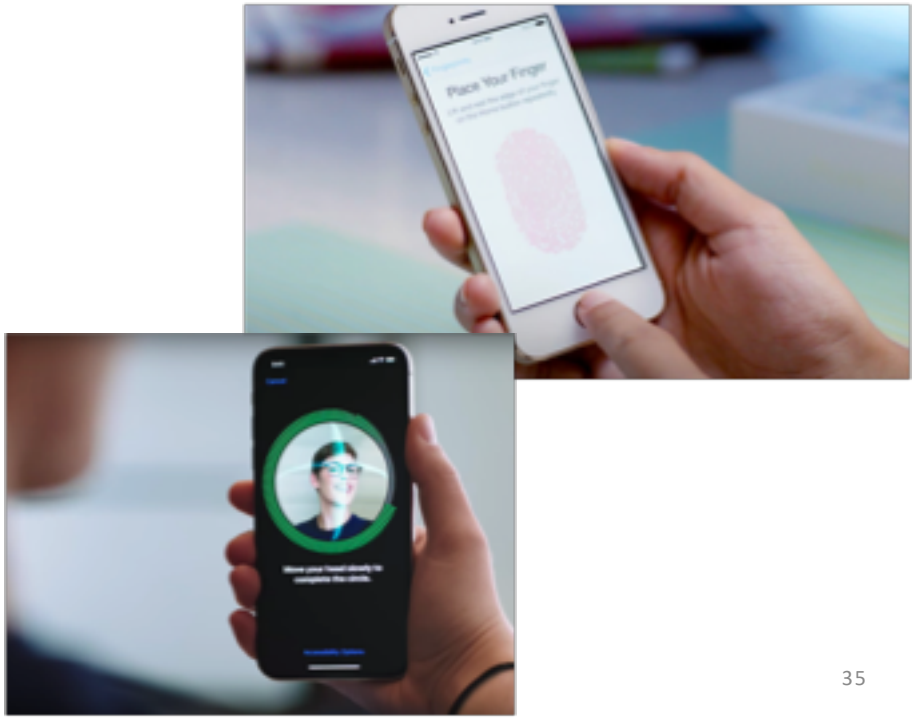
- Smart Homes
- Expert Users
- Digital Advocacy
- IoT
- Social Media



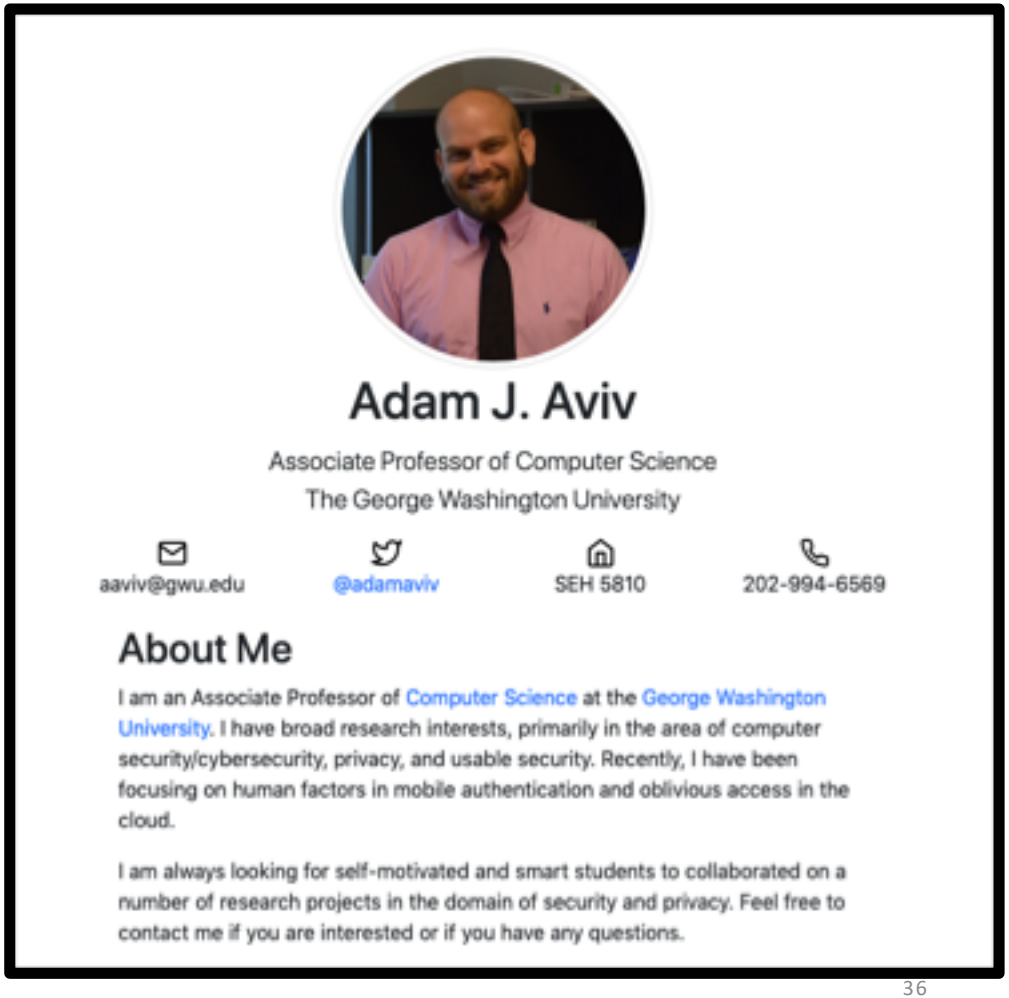
Mobile Authentication

- Your smartphone is your most personal computer … have you thought about your PIN lately?





Who am I?



Adam J. Aviv

Associate Professor of Computer Science
The George Washington University

aaviv@gwu.edu | @adamaviv | SEH 5810 | 202-994-6569

About Me

I am an Associate Professor of Computer Science at the George Washington University. I have broad research interests, primarily in the area of computer security/cybersecurity, privacy, and usable security. Recently, I have been focusing on human factors in mobile authentication and oblivious access in the cloud.

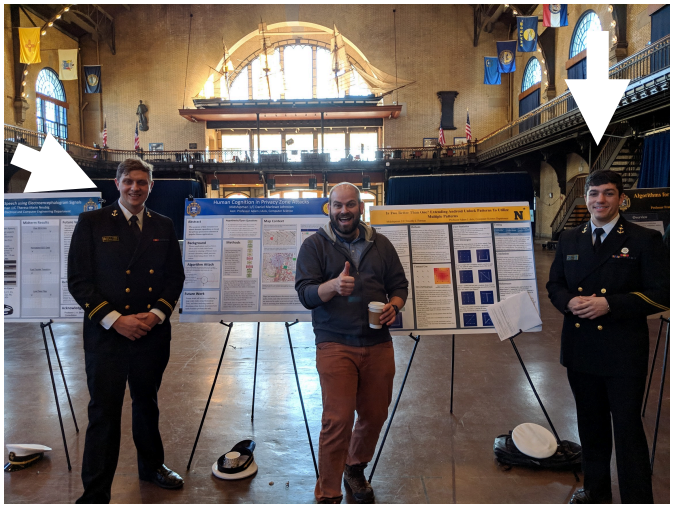
I am always looking for self-motivated and smart students to collaborated on a number of research projects in the domain of security and privacy. Feel free to contact me if you are interested or if you have any questions.

United States Naval Academy 2013-2019

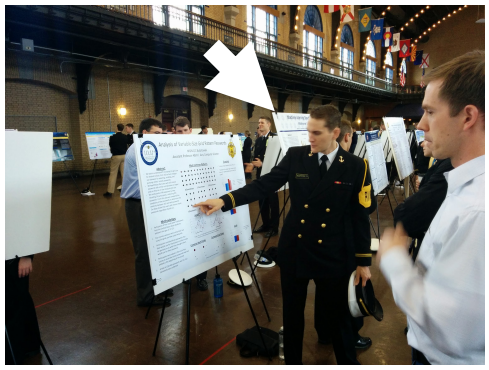
- Undergraduate Program
  - ~4,000 Midshipman
- Teaching and Research Mission
- No-Cost Appointment at UMBC
  - Graduate Advising

Tim Forman

Dan Johnston

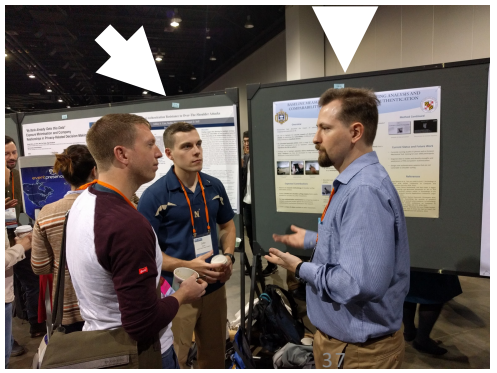


Devon Budzitowski

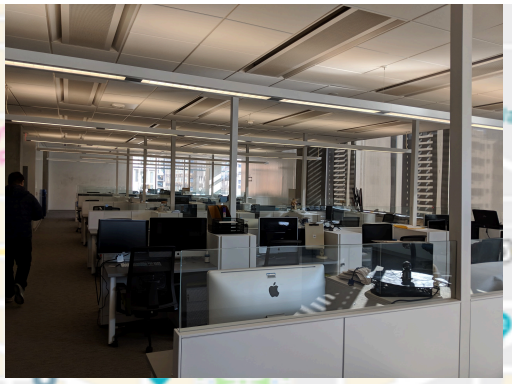


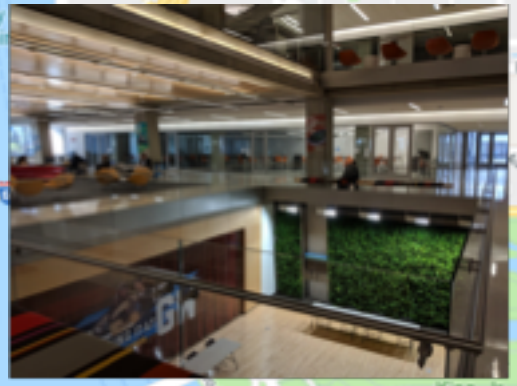
John Davin

Flynn Wolf



THE GEORGE WASHINGTON UNIVERSITY

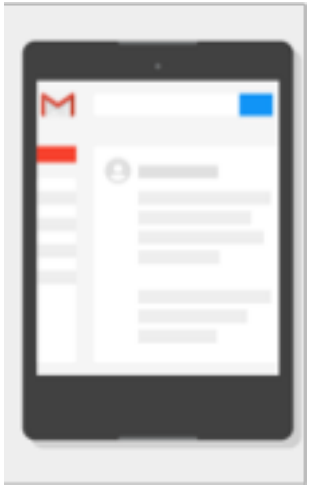


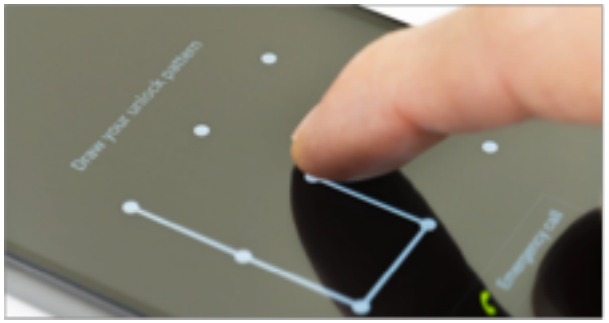


I'm looking to admit two-to-three graduate students!

Why Focus on Mobile Authentication?

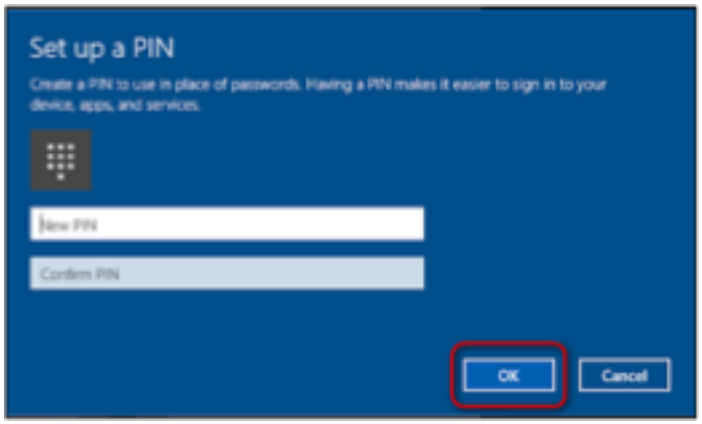
- Mobile Devices store or have access to a wealth of private information
  - ***Mobile Device Authentication is a proxy for other authentication***
- Mobile Devices (Smartphones and Tablets) are, perhaps, the most prevalent forms of Internet connected computers
  - Especially in lower socio-economic classes and the developing world
  - Local authentication proxies for remote authentication
- Significant Human Factors in Authentication
  - Humans can be the *hardest* part of security!
- Different interaction model
  - Held and touch and carried: *incredibly personal devices!*





PINs

PINs are also used in a lot more contexts…



Experiments

- Conducted a *large* study of PINs collected on Mobile Devices
  - Amazon Mechanical Turk (Mturk)
  - n=1220
- Considered different selection scenarios
  - Control: 4-digit and 6-digit PINs
  - Blacklisting – disallowing some PINs
- Security Analysis based on throttled and unthrottled attacker
  - Limited number of guesses
  - Analyzed the right size and composition for a blacklist

Priming and PIN Entry

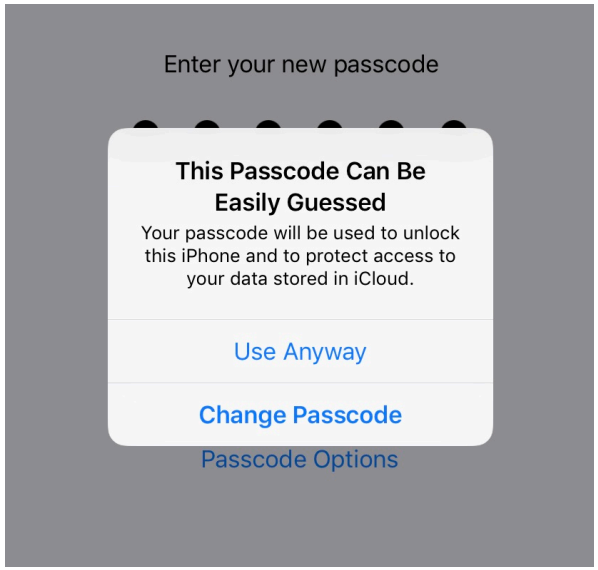
Figure 1: Priming information provided during the survey.

Figure 2: The design of the page on which we asked the participants to create a PIN.

Blacklisting ….

Kanye definitely ignores warnings…

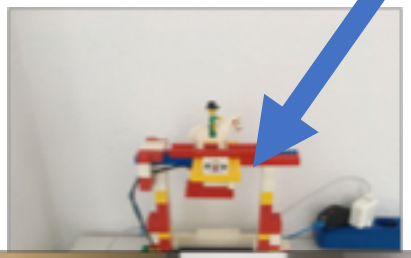
… because Apple has been using a non-enforcing blacklist for some time



Learning Apple's PIN Blacklist

Raspberry PI Controller to input touches

Camera connect to computer



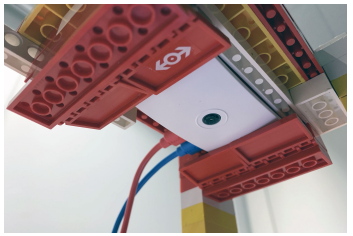
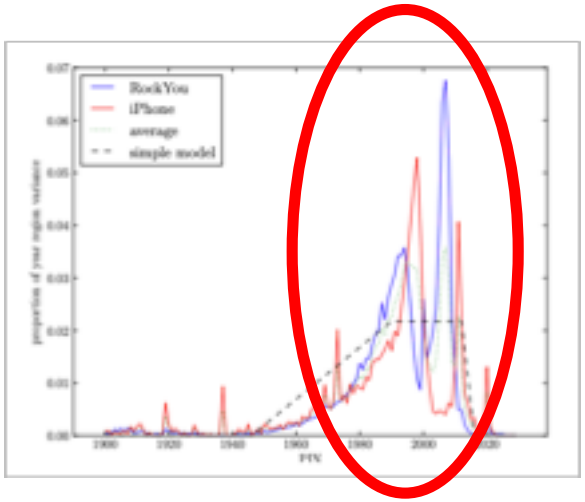


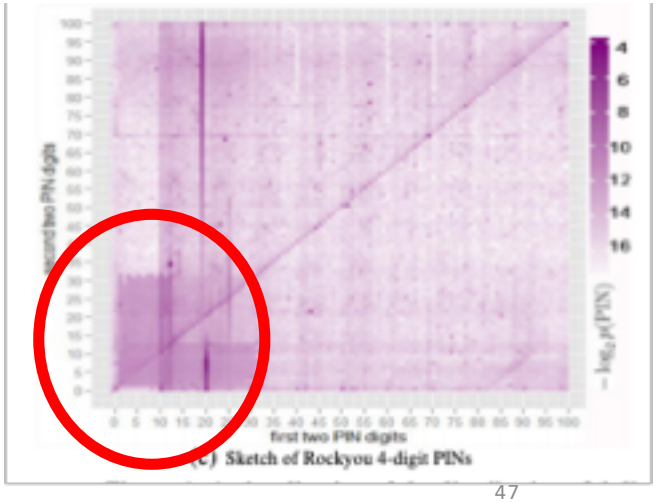
Image recognition for the blacklist warning

Then … Brute force and wait!

The iOS Blacklists

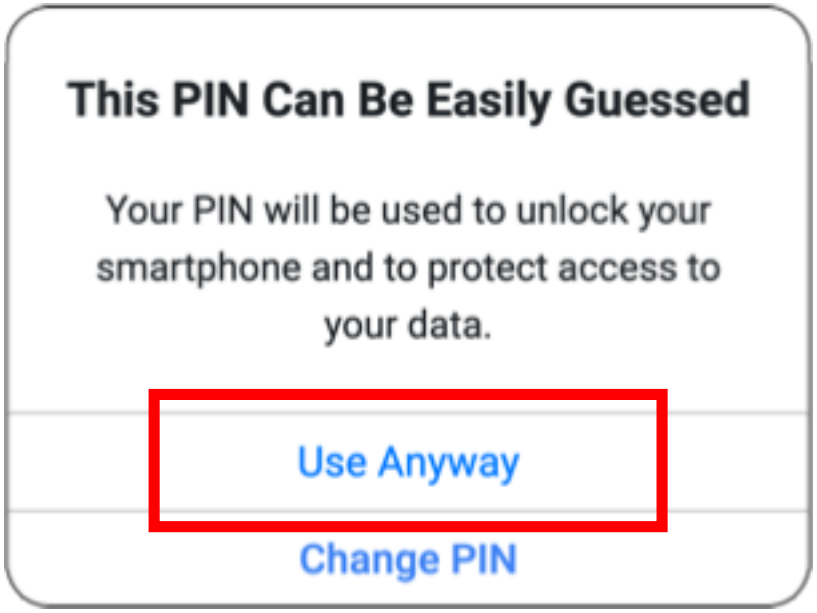
- 4-digit Blacklist (274 PINs)
  - Based on the suggestions from Bonneau et al.
  - Basic Patterns, Dates, “common” PINs
- 6-digit Blacklist (2910 PINs)
  - More Ad-hoc – not much to build on
  - Mostly Pattern based and “common” PINs





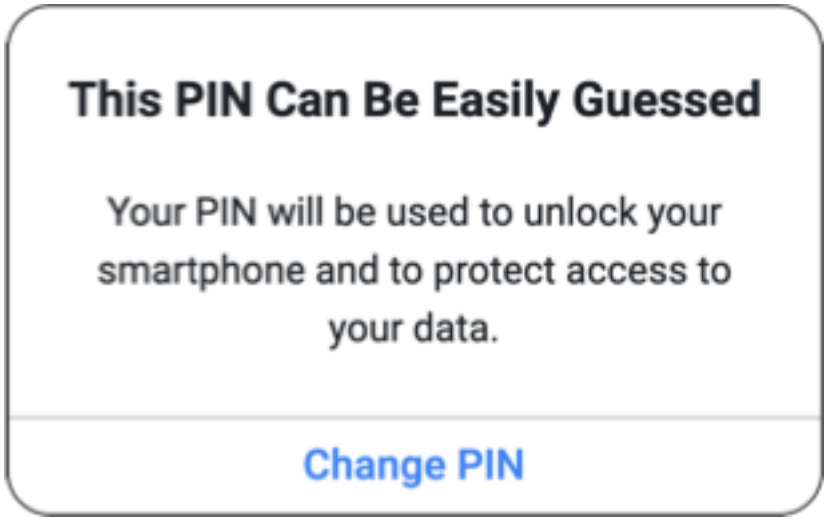
Blacklist Enforcement

Non-Enforcing



With Clickthrough

Enforcing



Without Clickthrough

Data Driven Blacklist

- Significantly Smaller (10x)
  - 27 PINS
- Significantly Larger (10x)
  - 2740 PINs

- Amitay Dataset
  - 204,432 4-digit PINs
  - Collected from an iOS application
  - Top-N most frequent

Blacklist Treatment

- iOS-4-Digit-wCt (ios-4-wC)
  - 4-digit iOS Blacklist **with** an option for participants to **clickthrough**
- iOS-4-Digit-nCt (ios-4-nC)
  - 4-Digit iOS Blacklist **without** an option for participants to **clickthrough**
  - **Non-Enforcing**
- iOS-6-Digit-wCt (ios-6-wC)
  - 6-digit blacklist **with** an option for participants to **clicthrough**
  - **Encorcing**
- DD-4-digit-wCt (DD-4-27)
  - Data driven blacklist
  - **Enforcing**
- DD-4-digit-wCt (DD-4-2740)
  - Data driven blacklist
  - **Enforcing**
- Always blacklist first choice!
  - Placebo-4-digit (Pla-4)
  - Placebo-6-digit (Pla-5)
  - **Enforcing**

Does the presence of the blacklisting message, itself, change behavior?

4- and 6- Digit Data Sets (no-blacklist)

- Control PINs
  - PINs selected without any blacklist intervention
  - Control-4-digit: 231
    Control-6-digit: 127

- First Choice PINs
  - PINs selected before/without any blacklist intervention
  - **First-Choice-4-digit: 851**
  - **First-Choice-6-digit: 369**

Focus on a **Throttled Attack Model**

Throttled Attacker only has a limited number of guesses based on the lockout period of the device.

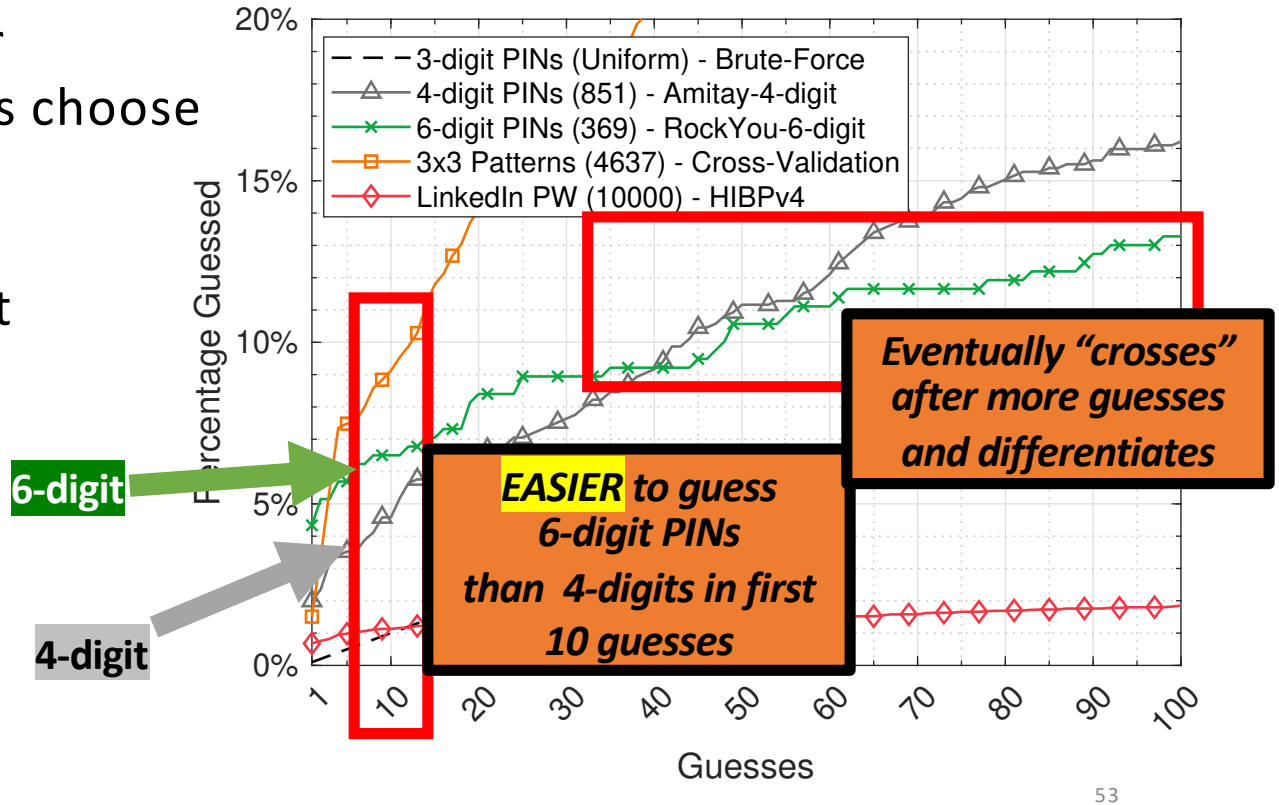
Table 1: Rate-Limiting on Mobile Operating Systems

To Make *n* Guesses	Accumulated Waiting Time Android 7, 8, 9	iOS 9, 10, 11, 12
1-5 guesses	0 s	0 s
6 guesses	30 s	1 m 0 s
7 guesses	30 s	6 m 0 s
8 guesses	30 s	21 m 0 s
9 guesses	30 s	36 m 0 s
10 guesses	30 s	1 h 36 m 0 s
30 guesses	10 m 30 s	-
100 guesses	10 h 45 m 30 s	-
200 guesses	67 d 2 h 45 m 30 s	-

Guessing Strategy

False Sense of Security?

- Attacker has some prior knowledge of how users choose PINS
  - 4-digit: Amitay Dataset
  - 6-digit: Rockyou Dataset
- Always guesses in frequency order



Impact of Blacklists?

- Assume that the attacker knows the blacklist
  - Afterall, we extracted the iOS blacklist
- Does that knowledge actually help?
  - E.g., for an enforcing BL: Avoid guessing anything on the blacklist

For non-enforcing, attacker "looses" by avoiding too many common PINs on the blacklist

Table 6: Attacker's Gain From Blacklist Knowledge

Treatment	10 Guesses No.	10 Guesses %	100 Guesses No.	100 Guesses %	Guess No. Median	Knowledge Beneficial
Pla-4	±0	±0 %	±0	±0 %	±0	–
iOS-4-wC	-3	-2 %	-9	-8 %	-303	✗
iOS-4-nC	+3	+2 %	+3	+2 %	+245	✓
DD-4-27	+4	+3 %	+5	+4 %	+27	✓
DD-4-2740	±0	±0 %	+1	+1 %	+2740	✓
Pla-6	±0	±0 %	±0	±0 %	±0	–
iOS-6-wC	-9	-7 %	-8	-6 %	-7322	✗

For Enforcing blacklist there is a benefit

Throttled Guessing Results w/Blacklists

Clicked-through PINs are very easily guessed…

… but because attacker can't leverage blacklist, doesn't affect guessing performance

Impact of small and iOS blacklists are limited …

… but the largest blacklist mostly neutralizes a throttled attacker

Higher % means MORE PINs guessed and is easier for the attacker

Treatment	Participants	Blacklist Hits	10 Guesses No.	10 Guesses %	100 Guesses No.	100 Guesses %	Guess No. Median
First-Choice-4-digit	851	-	39	5 %	138	16 %	1 330
Clicked-through-4	19	19	5	26 %	13	68 %	50
Control-4-digit	231	-	11	5 %	39	17 %	1 185
Placebo-4-digit	122	122	5	4 %	19	16 %	2 423
iOS-4-digit-wCt	124	28	5	4 %	18	15 %	1 405
iOS-4-digit-nCt	126	21	4	3 %	14	11 %	1 747
DD-4-digit-27	121	5	4	3 %	18	15 %	1 928
DD-4-digit-2740	127	88	0	0 %	1	1 %	2 871
First-Choice-6-digit	369	-	24	7 %	49	13 %	39 389
Clicked-through-6	10	10	9	90 %	9	90 %	1
Control-6-digit	127	-	7	6 %	18	14 %	36 822
Placebo-6-digit	117	117	3	3 %	10	9 %	154 521
iOS-6-digit-wCt	125	15	9	7 %	13	10 %	40 972

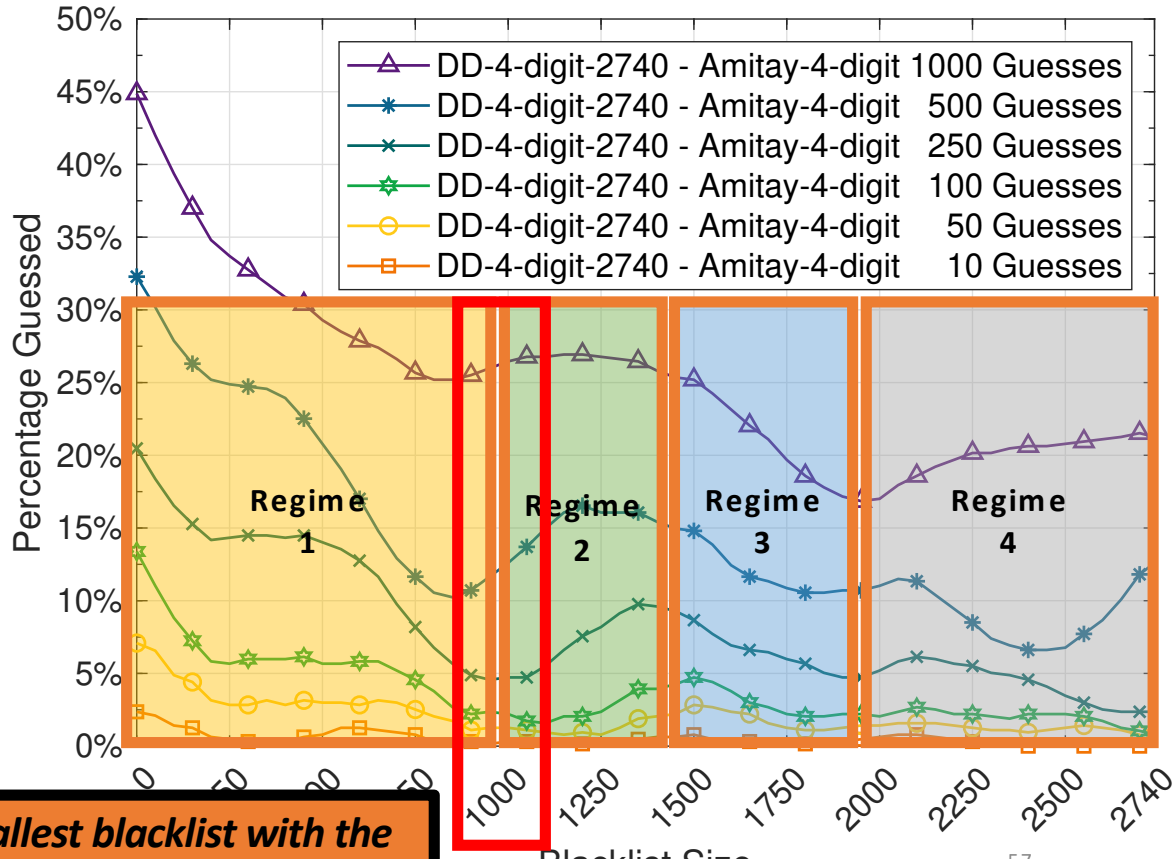
Can we find the *"right"* Blacklist size?

- Use the largest blacklist to simulate smaller blacklists
  - Assume first choice of participant not on the simulated size was their choice
  - Example:
    - Blacklist of size 1,000 would have top 1,000 PINs (of the 2,740)
    - First PIN in selection order not in those 1,000, assume the participant choose that one!
  - Consider the throttled attacker at a given size blacklist w/knowledge

- Not clear that really large blacklists will always help
  - Eliminating some PINs means an attacker can focus on others
  - Goal: Find a balance

Simulating Blacklist Sizes

- Regime 1: Negative Attacker
  - Blacklist forces more second choices but attacker still guessing first choice
- Regime 2: Positive Attacker
  - Blacklist contains all first choices, attacker can focus on just second choice
- Regime 3: Negative Attacker
  - Blacklist forces more third choices, but attacker still guessing second choices
- Regime 4: Positive Attacker
  - Blacklist is getting so large, it's easier to just guess from the remaining



Smallest blacklist with the most gain at ~10%

PIN Research Takeaways

- With a throttled attacker, **the benefits of 6- over 4-digit PINs is negligible**
  - Less diverse models for choosing 6-digit PINs
  - 6-digit PINs much better in a unthrottled setting

False Sense of Security?

- **Clickthrough blacklists are both good and bad**
  - Good: Less user frustration and blacklist can't be used by attacker effectively
  - Bad: PINs that are clicked-through are very, very easy to guess

- **Blacklists, as used, have no meaningful impact in the throttled attack setting**
  - Would need to be MUCH larger, but this would lead to higher user frustration
  - Recommend a blacklist that is ~10% of the key space may offer a tradeoff

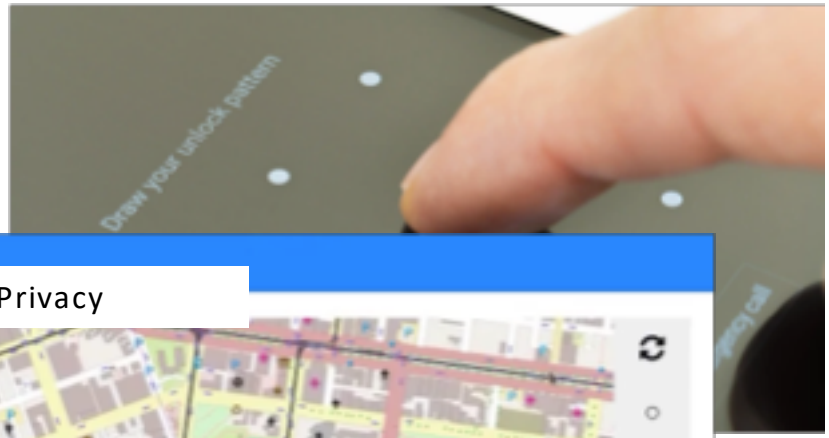
- Placebo Blacklists
  - **Placebo only works as long as no one knows …**
  - Suggest that second choices are generally better than first

One of many projects!

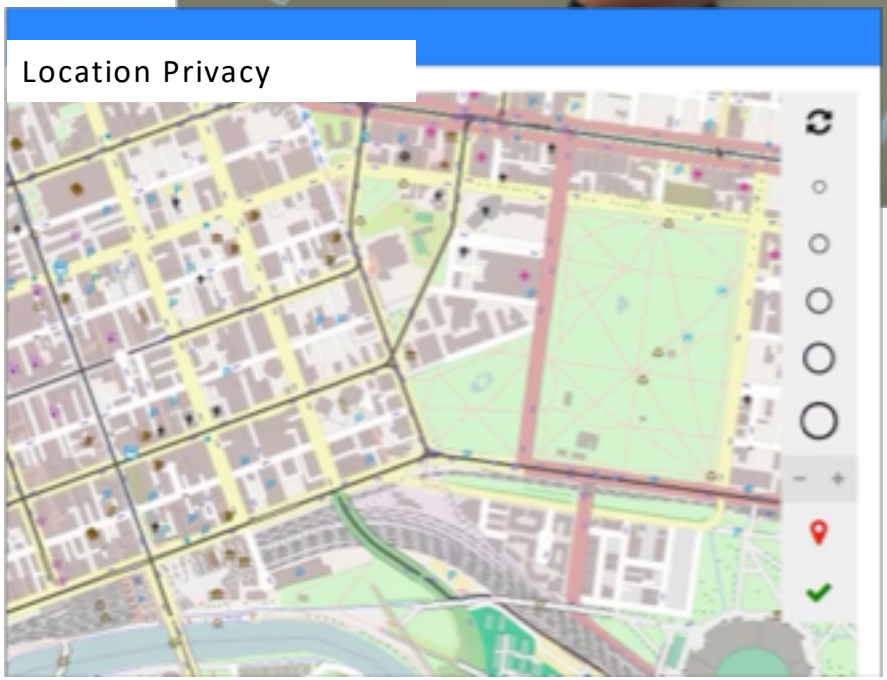
Privacy for Older Adults

Small Business Security

Android Patterns



Location Privacy



Lot of great work to do!

Are you Interested in getting a PhD in
computer security and/or usable security and privacy.

Feel free to reach out.

Adam J. Aviv
Asoc. Prof. of Computer Science
aaviv@gwu.edu / adamaviv.com

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

DEPARTMENT OF
COMPUTER SCIENCE
SCHOOL OF ENGINEERING & APPLIED SCIENCE