# An Evaluation of the Ozzie Proposal

Steven M. Bellovin
https://www.cs.columbia.edu/~smb

# The CLEAR Proposal

- What is it?
- Does it solve the problem?
- What are the problems with it?

# Ozzie's CLEAR Proposal

- At unlock time, the device takes the symmetric device decryption key and encrypts it (and, e.g., the WiFi MAC address) to the vendor's public key

- Law enforcement enters some special gesture on a seized phone; the devices displays  a QR code of the encrypted device key, and sends that to the vendor, e.g., by photographing it; the device also starts sending broadcast packets which LE picks up and includes in the request

- The vendor sends back the decryption key as another QR code

- LE shows that QR code to the device

# Receiving an Unlock Code

- The device decrypts its memory
- It then permanently “bricks” itself by blowing an internal JTAG fuse
- Seems cool—memory is no longer alterable, etc.
- There are problems…

# It's a Systems Problem

- Exceptional access is a systems problem
- It *isn't* just a cryptography problem; solutions have to look at all aspects
  - But the crypto protocol is hard enough
- And: different aspects can interact

# Eran Tromer's Attack

Attacker's goal: unlock a specific phone

1. Activate the exceptional access mechanism; capture the QR code

2. Root another phone of the same model so that it displays the *target* phone's QR code

3. Arrange for law enforcement to want to unlock the rooted phone

4. When the unlock image arrives, transmit it to the attackers

# Cryptography is Hard

- Ray Ozzie presented his design to a small group that included some *serious* security and cryptography experts

- None of us spotted the flaw

- In a presentation a few months later at Columbia University, Eran Tromer found that flaw in during the presentation

Cryptography is hard…

# Security is Hard

- Protecting the vendor infrastructure is hard
- Example: could a malicious QR code do nasty things?
  - Security flaws have just been found in fax machines—a malicious page sent to one can compromise it!
- HSMs have been shown to be insecure, too
- What about insider attacks?
- Above all, we need not just security but *assurance*—we need to *know* that something is secure

# Protecting the Vendor Key

- Vendors need a private key to handle unlock requests
- Ozzie argues that if they can protect code-signing keys (and all vendors have them), they can protect unlock keys
- But: code-signing keys are used a few times a year; exceptional access keys are used many times a day
    - You can add more layers of protection for signing keys, since the need for them is scheduled in advance
- How do you protect them against “routinization”?

# Authentication and Authorization

- How do vendors authenticate unlock requests?
- Which law enforcement agencies are allowed—by the laws of their jurisdiction—to request device unlocking?
  - (If there's no authorization mechanism, attackers don't need Tromer's attack)
- How do you do this internationally?

# International Aspects

- In what country are the unlock keys held?
- Will China be happy if US companies hold the keys?
- Will the US be happy if China holds them?
- A key per country? How do phones know which key to use?

# Missing Information

- Who pays?
  - Remember that unlocked phones are permanently disabled
- What about communications?
- Are there technical designs rendered impossible by this scheme?

# Questions?