The case for Usable Mobile Security

N. Asokan,
Nokia Research Center

Joint work with Cynthia Kuo (NRC)

August 2012
MORDAC, THE PREVENTER OF INFORMATION SERVICES.

SECURITY IS MORE IMPORTANT THAN USABILITY.

IN A PERFECT WORLD, NO ONE WOULD BE ABLE TO USE ANYTHING.

To complete the log-in procedure, stare directly at the sun.
Outline

• Why worry about usable security?
• What is special about mobile?
• Some examples of mobile usable security problems we face
  – A look back: The “First Connect” story
  – Current problems
    – Local (user) authentication
    – Mobile CAPTCHA
    – Trustworthy installation
    – Theft resistance and data/credential recovery
  – ….

• Conclusions
Why worry about usable security

Lack of security usability
• harms security, eventually
• lowers overall attractiveness of the device/service, eventually
• costs money!

In many cases, the source of the "cost" is surprising
What is special about mobile?
Your mobile phone: Not a smaller version of your PC
Your mobile phone: Not a smaller version of your PC

Mobile phone applications have different requirements due to

1. Smaller physical screen size
   → Less room for security indicators, notifications etc.

20"

3.5"

2.4"

10.1"

20”
Your mobile phone: Not a smaller version of your PC

Mobile phone applications have different requirements due to
1. Smaller physical screen size
2. Different input mechanisms

Directional pad + keyboard

Touch screen

Keyboard + mouse
Your mobile phone: Not a smaller version of your PC

Mobile phone applications have different requirements due to
1. Smaller physical screen size
2. Different input mechanisms
3. Limited battery life
4. More prone to theft/loss
5. Slower and less reliable network connectivity
6. (Comparatively) limited computational power
Example: Setting up the first connection

• **First Connect**: setting up contexts for subsequent communication.
  - Typically for proximity communications between personal devices, e.g.:
  - Pairing a Bluetooth phone and headset
  - Enrolling a Phone or PC in the home WLAN
  - More instances to come: Wireless USB, WiMedia

• **Problem (circa 2006)**: Secure First Connect for personal devices
  - Initializing security associations (as securely as possible)
  - No security infrastructure (no PKI, key servers etc.)
  - Ordinary non-expert users
  - Cost-sensitive commodity devices
Prevalent mechanisms were not intuitive

SSID? WPA? Passcode?

Example: First Connect
Cracking the Bluetooth PIN*

Yaniv Shaked and Avishai Wool

School of Electrical and Computer Engineering
Tel Aviv University, Ramat Aviv, Israel
shakedy@eng.tau.ac.il

Abstract
This paper describes the implementation of an attack on the Bluetooth security mechanism. Specifically, we de-

Security Weaknesses in Bluetooth

Markus Jakobsson and Susanne Wetzel

Lucent Technologies - Bell Labs
Information Sciences Research Center
Murray Hill, NJ 07974
USA
{markusj,agwetzel}@research.bell-labs.com

Abstract. We point to three types of potential vulnerabilities in the Bluetooth standard, version 1.0B. The first vulnerability opens up the system to an attack in which an adversary under certain circumstances is able to determine the key exchanged by two victim devices, making
Naïve usability measures damage security

Pictures taken with mobile phone showed up on neighbour's TV

- Default password must be changed when starting to use Bluetooth-equipped devices; read the manual

elsewhere as well. It is, therefore, absolutely essential that the password is changed immediately when the device is first installed."

"This is clearly printed in the user's manual", Rosenberg points out. How often have we heard that before?

"Once the digital receiver's password has been changed, the new password also has to be entered in the transmitting device, in this case a mobile phone. The user is only warned about the need to change the password when the device is turned on. This causes the password to be reactivated if it is changed. The case is, however, not unique in the consumer electronics market. The instructions often contain vague or unclear information instead of clearly stating that a digital password has to be activated for the new device to function."

Example: First Connect
Naïve security erodes usability

- Car kits allow a car phone to retrieve and use session keys from a mobile phone smartcard
- Car kit requires a higher level of security
  ➢ users have to enter 16-character passcodes

More secure = Harder to use?

Cost:

Calls to Customer Support

Example: First Connect
Wanted: intuitive, inexpensive, secure first connect

• Two (initial) problems to solve
  − Peer discovery: finding the other device
  − Authenticated key establishment: setting up a security association

• Assumption: Peer devices are physically identifiable
Key establishment for first connect ~2006

- **Key transport via OOB channel**
  - Symmetric crypto only
    - Authenticated
    - Unauthenticated
  - Asymmetric crypto
    - Authenticated
    - Unauthenticated

*Short keys vulnerable to passive attackers*

*Secure against passive attackers*

**Example: First Connect**
Authenticating key agreement

• Use an auxiliary channel to transfer information needed for authentication
• Two possibilities for realizing secure auxiliary channel
  – User assistance
  – Other out-of-band secure communication channels:
    – E.g., Near Field Communication, infrared, …
Authenticating key agreement: user-assisted

- User “bandwidth” is low (4 to 6 digits)
- Directionality depends on available hardware (1-way or 2-way)
- Security properties (integrity-only, or integrity+secrecy)

Example: First Connect
User as the secure channel

• Peer discovery by “user conditioning”: introduce a special first connect mode
  – E.g., Press a button to put device into the special mode
  – Demonstrative/indexical identification

• Authentication of key agreement by
  – Comparing short non-secret check codes (aka “short authentication string”), and
  – entering a short secret Passkey

• Short key/code should not hamper security
  – Standard security against offline attacks
  – Good enough security against active man-in-the-middle
Authentication by comparing short strings

\( v_A \leftarrow H(A, B, PK_A|PK'_B) \)

\( v_B \leftarrow H(A, B, PK'_A|PK_B) \)

\( v_A \) and \( v_B \) are short strings (e.g., 4 digits),
User approves acceptance if \( v_A \) and \( v_B \) match

A man-in-the-middle can easily defeat this protocol

Example: First Connect
MitM in comparing short strings

Guess a value $SK_{C2}/PK_{C2}$ until $H(A, B, PK_A|PK_{C2}) = v'_B$

If $v'_B$ is $n$ digits, attacker needs at most $10^n$ guesses; Each guess costs one hash calculation

A typical modern PC can calculate 100000 MACs in 1 second

Example: First Connect
Authentication by comparing short strings

Choose long random $R_A$
Calculate commitment $h_A^\leftarrow h(A, R_A)$

$\begin{align*}
v_A &\leftarrow H(A, B, PK_A | PK'_B, R_A, R'_B) \\
h_A^\leftarrow h(A, R_A)
\end{align*}$

key agreement: exchange $PK_A, PK_B$

Send commitments $h_A, R_B$

Open commitment $R_A$

Choose long random $R_B$

Verify commitment $h'_A^\leftarrow h(A, R'_A)$
Abort on mismatch $v_B^\leftarrow H(A, B, PK'_A | PK_B, R'_A, R_B)$

User approves acceptance if $v_A$ and $v_B$ match

$2^{-l}$ ("unconditional") security against man-in-the-middle (l is the length of $v_A$ and $v_B$)

$h()$ is a hiding commitment; in practice SHA-256

$H()$ is a mixing function; in practice SHA-256 output truncated

Example: First Connect
Authentication by comparing short strings

Choose long random $R_A$
Calculate commitment $h_A \leftarrow h(A, R_A)$

Choose long random $R_B$

User approves acceptance if $v_A$ and $v_B$ match

$2^{-l}$ ("unconditional") security against man-in-the-middle ($l$ is the length of $v_A$ and $v_B$)

$h()$ is a hiding commitment; in practice SHA-256

Example: First Connect

Key agreement: exchange $PK_A$, $PK_B$

Send commitments $h_A$

Verify commitment $h'_A \leftarrow h(A, R'_A)$
Abort on mismatch

$R_B$

Open commitment $R_A$

$A$

$B$

$V_A \leftarrow H(A, B, PK_A|PK'_B, R_A, R'_B)$

$V_B \leftarrow H(A, B, PK'_A|PK_B, R'_A, R_B)$

$O_k/no-t-0k$

$O_k/no-t-0k$
Authentication using interlocking short passkeys

Choose long random $R_{Ai}$

Calculate commitment $h_A \leftarrow h(A, PK_A|PK'_B, Pi, R_{Ai})$

Verify commitment $h'_B \overset{?}{=} h(B, PK'_A|PK_B, Pi, R_{Bi})$

One-time passkey $P$ is split into $k$ parts ($k > 1$): next 4-round exchange repeated $k$ times

$h()$ is a hiding commitment; in practice SHA-256

Up to $2^{-(l-1)}$ ("unconditional") security against man-in-the-middle (l is the length of $P$)

Originally proposed by Jan-Ove Larsson [2001]: essentially multi-round MANA III

Example: First Connect
Key establishment for first connect

Key establishment

Key transport via OOB channel

Key agreement

Asymmetric crypto

Authenticated

Unauthenticated

Symmetric crypto only

Authenticated

Unauthenticated

Authentication by integrity checking

Short string comparison

User-assisted

Example: First Connect

Authentication by shared secret

User-assisted
Key establishment for first connect ~2008

<table>
<thead>
<tr>
<th></th>
<th>Unauthenticated Diffie-Hellman</th>
<th>Authenticated Diffie-Hellman</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>short-string comparison</td>
</tr>
<tr>
<td>WiFi Protected Setup</td>
<td>“Push-button”</td>
<td>□</td>
</tr>
<tr>
<td>Bluetooth 2.1</td>
<td>“Just-works”</td>
<td>□</td>
</tr>
<tr>
<td>Wireless USB</td>
<td></td>
<td>□</td>
</tr>
</tbody>
</table>

“Security associations for wireless devices” (Overview, book chapter)
“Standards for security associations in personal networks: a comparative analysis” IJSN 4(1/2):87-100 (survey of standards)
First Connect: today

• Widely deployed (Bluetooth SSP, WiFi Protected Setup)
• Improving usability/security → fundamental protocol changes
  – Did it really help?
• Recent research exploiting properties of radio communication looks promising
First Connect: A cautionary tale

Short pass keys were intended to be one-time

• Fixed pass keys are sometimes unavoidable
• Use of fixed pass key must be accompanied by suitable techniques to thwart online guessing attacks
  − Enter a 1-minute lock-out period after 3 failed guesses (WiFi Protected Setup)
  − Use an authenticated tunnel (a la server-authenticated TLS)
    − fixed public key (+ authenticator) to protect
    − Can you work out such a protocol?
    − (WUSB 1.1 Fixed Passkey Association Model)
December 27, 2011

Wi-Fi Protected Setup PIN brute force vulnerability

Filed under: advisories — Stefan @ 3:00 am

A few weeks ago I decided to take a look at the Wi-Fi Protected Setup (WPS) technology. I noticed a few really bad design decisions breaking the security of pretty much all WPS-enabled Wi-Fi routers. As all of the more recent router models come with WPS enabled by default, many users have been left with a relatively easy to exploit back door into their home network.

I reported this vulnerability to CERT/CC and provided them with a list of (confirmed) affected vendors. CERT/CC has assigned VU=723755. To my knowledge none of the vendors have reacted and released firmware with mitigations in place.

Detailed information about this vulnerability can be found in this paper: Brute forcing Wi-Fi Protected Setup - Please keep in mind that this is a list of affected devices.

I would like to thank the guys at CERT for coordinating this vulnerability.

Update (12/29/2011 - 20:15 CET)

As you probably already know, this vulnerability was independently discovered by Craig Heffner (/dev/ttyS0, Tactical Network Solutions) and released information about it first. Craig and his team have now released their tool "Reaver" over at Google Code.

My PoC Brute Force Tool can be found here. It's a bit faster than Reaver, but will not work with all Wi-Fi adapters.

Update (12/31/2011 - 14:25 CET)

http://www.kb.cert.org/vuls/id/723755

Example: First Connect
Local user authentication: need new methods

Need alternatives that are:
- Faster
- More enjoyable
- Secure enough

Cost: users avoid using apps that mandate local authentication (work e-mail!)

Cost: weak PINs

Biometrics
Wearables
Local user authentication: a cautionary tale

@koush @koush
The face recognition unlock thing is really easily hackable. Show it a photo.

@Tim Bray @timbray
@koush Nope. Give us some credit.

http://youtu.be/BwfYSR7HttA
CAPTCHA on mobile devices

Cost:
Estimated 15% drop-off rate when encountering a CAPTCHA on mobile devices
Alternatives to standard CAPTCHA?

- The problem is real
- Can it be solved without CAPTCHA? (device authentication)
- Mobile-friendly CAPTCHA variants?

Select all **Frogs** and press **Next**

Mobile CAPTCHA by Alex Smolen, Becky Hurwitz, Dhawal Mujumdar, UC Berkeley i213 Spring 2010
Long tail: app/content creation made easier

Create your app for Ovi in minutes.
It’s free.

Get Started

Join the expanding list of global and local brands using Ovi App Wizard to reach consumers in over 180 countries.
Plenty of choice for the user

"Is this App Safe?"
A Large Scale Study on Application Permissions and Risk Signals
(WWW 2012)
Can “clique-sourcing” help?

Secure Installer for Nokia N810
Pern-Hui Chia (NTNU) et al

Friend App Rating (Facebook app + Firefox plugin)
Jo Mehmet Øztarman & Pern-Hui Chia, NTNU
Mobile devices can help security/privacy

• Mobility and portability can help in surprising ways: e.g.,
  – PayPal Bump
  – ”Mobility helps security in ad hoc networks”, Čapkun et al, MobiHoc ’03
  – ...

• Mobiles can sense location, motion, ambient light, noise level, ...
  – Cues from context/history to set sharing, access control policies
  – ”CRePE: Context-Related Policy Enforcement for Android”, Conti et al, ISC ’10
  – ISAC (Intuitive and Sensible Access Control) project at NRC
    – SocialCom ’12 Paper (to appear), older tech report, PerCom ‘11 Demo
    – AISec ‘10 position paper.
Summary

• Usable mobile security is a challenging goal
  – Lack thereof results in surprising costs
  – Requires changes under-the-hood (protocols, algorithms, ...)
• No satisfactory solutions yet for a number of specific instances
  – First Connect?
  – Local (user) authentication
  – Mobile CAPTCHA
  – Trustworthy installation
  – [Theft resistance and data/credential recovery]
  – ....
• One promising avenue: intuitive security/privacy policy configuration by using the context and history of the user’s mobile device
How to make it possible to build trustworthy information protection mechanisms that are simultaneously easy-to-use and inexpensive to deploy while still guaranteeing sufficient protection?