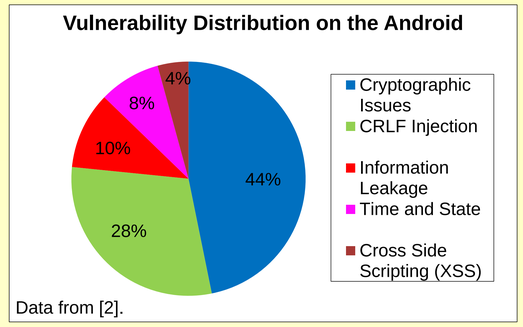
Securing Public-key Cryptography on the Android Platform

Hannah Gommerstadt
Harvard University
Advisors: Stephen Chong and Aslan Askarov

Motivation

- The Android Platform runs on 48.6% percent of smart phones in the United States and handles a plethora of sensitive data which makes it a minefield of security bugs [1].
- A recent report found that 44% of errors on the Android platform were due to cryptographic vulnerabilities and that out of 10,000 applications 40% had hardcoded cryptographic keys [2].
- Public-key cryptography is integral to many transactions on the Android.



Data from [2].

Cryptographic Errors

- Most developers rely on third party libraries for encryption and decryption
- Vulnerabilities can arise due to misuse of correctly implemented cryptographic protocols
- Possible vulnerabilities:
  - Hardcoded cryptographic keys
    - Example: the key appears in the program text
  - Output of private keys onto insecure channels
    - Example: the key is read out to a file instead of being stored in a secure key store

Results

Theory

- Defined a semantic security condition for public-key cryptography
  - Used symbolic definition of public-key cryptography
  - Adapted standard definition of non-interference to handle non-determinism of encryption
- Developed an enforcement mechanism
  - Designed a type system that allows for safe encryption, decryption and key generation
- Proved that our type system enforces the security condition
- The security condition and type system is based on work done by Askarov et al [3]

Implementation

- Developed Cryptflow, which is an information flow analysis of Java code that detects misuses of cryptography
  - Based on formal type system, but is also flow-sensitive
- Cryptflow can currently analyze snippets of Java code and identify simple vulnerabilities
  - Output of private keys to System.out.println()
  - Cryptographic keys hardcoded into the program text
- Cryptflow is built on top of the Objanal framework which facilitates program analyses of Java code [4].

Conclusion

It is necessary to guarantee and enforce the security of public-key cryptographic protocols, especially on the Android platform. Cryptflow, and the theory behind it, is a step towards this goal.

Future Directions

We are currently working on having Cryptflow analyze the Java source code of entire Android applications to detect misuses of public-key cryptography.

More Information

A full copy of my senior thesis is available at http://anyag.net/docs/thesis.pdf.

References

1. Chloe Albanesius. Almost half of u.s. smartphones running android., March 2012.
2. Veracode. State of software security report: The intractable problem of insecure software, December 2011.
3. Aslan Askarov, Daniel Hedin, and Andrei Sabelfeld. Cryptographically-masked flows. Theor, Comput. Sci., 402(2-3):82-101, July 2008.
4. Stephen Chong, Andrew Johnson, Scott Moore, and Owen Arden. Accrue ObjAnal, 2013. http://people.seas.harvard.edu/~chong/accrue.html.