trol decisions, and to classify photos to determine access control decisions. We measure the time taken by an Android app to access stored photos in various scenarios. Table 1 shows the times taken to access each photo, averaged over 50 random photos from the Caltech Faces 1999 dataset. Scenario 1 establishes the baseline time taken to access each stored photo without CHIPS. While it takes 3.6 seconds to perform a privacy check for a photo without result caching (Scenario 2), we believe that the majority of photo access decisions will have been precomputed and cached by CHIPS's background service. Hence, for the majority of image file accesses for which photo access decisions have been cached, CHIPS would add 94.2 ms (98% overhead) to the critical execution path of accessing the photo (Scenario 3). Finally, Scenario 4 shows that for whitelisted apps, CHIPS adds only 21.2 ms (22% overhead) to the critical path of photo access.

5. RELATED WORK

Android Permissions. Apex [10] and Jeon et al. [7] propose finer-grained permissions for Android, but do not specifically target stored photos, unlike CHIPS. AppFence [5] modified the Android framework to preserve privacy by covertly substituting shadow data for sensitive data. AppFence protects only the camera, and does not protect stored photos, unlike CHIPS. Aurasium [16] mediates third-party apps using intercepts at the C and Java library level, whereas CHIPS uses mediation in the kernel to prevent apps from directly invoking system calls to bypass mediation.

Photo Privacy. P3 [11] protects the privacy of photos stored on third-party Photo-sharing Service Providers (e.g. social networks, photo-sharing sites). Darkly [6] is a privacy-preserving computer-vision library based on the OpenCV library [4], and it protects users from privacy loss due to continuously-sensing perceptual applications. PlaceAvoider [13] proposed new image analysis techniques for recognizing sensitive places in video streams from first-person cameras. PlaceAvoider focuses on image analysis, whereas CHIPS focuses on the systems architecture needed for enforcing stored photo privacy. Klemperer et al. [8] designed a series of user-studies which evaluated the effectiveness of using user-assigned tags to build access control rules for photos.

6. CONCLUSION AND FUTURE WORK

We have presented CHIPS, a fine-grained, face-recognition-based run-time access control system for stored photos on Android smartphones, which overcomes Android's current all-or-nothing access model for stored photos. We have demonstrated that CHIPS's privacy enforcement prevents unauthorized access to privacy-sensitive photos in unmodified real-world Android apps (Facebook), and that this enforcement imposes acceptable overheads of just 94.2 ms (98% overhead) per accessed photo when results are cached. We have also demonstrated that existing face detection and face recognition algorithms are sufficiently accurate, so that we can identify if a given face belongs to a trained database with a false-negative rate of 8%, and with a false-positive rate of 8%, and that they require minimal training, attaining optimal performance with just 4 training images per person.

In future, we intend to expand the CHIPS framework to support other types of media such as audio/video, along with other algorithms. For instance, we can extend CHIPS to run optical character recognition (OCR) algorithms on accessed photos to search for sensitive information, such as credit-card numbers and addresses, to proactively block access to photos containing such information. We also intend to explore the use of content-type checks to robustly identify files requiring privacy checks without relying on file extensions.

Acknowledgements

This research is funded in part by CMU-SYSU Collaborative Innovation Research Center and the SYSU-CMU International Joint Research Institute. We would like to thank the anonymous reviewers and our shepherd, Apu Kapadia, for their comments and constructive feedback. We would also like to thank Anupam Datta for his feedback on earlier versions of this work.

7. REFERENCES

[1] Androguard. https://code.google.com/p/androguard/.
[2] Computational Vision at CalTech. http://www.vision.caltech.edu/archive.html.
[3] CyanogenMod. http://www.cyanogenmod.org.
[4] OpenCV. http://opencv.org/.
[5] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall. These Aren't the Droids You're Looking For: Retrofitting Android to Protect Data from Imperious Applications. In *ACM CCS*, 2010.
[6] S. Jana, A. Narayanan, and V. Shmatikov. A Scanner Darkly: Protecting User Privacy From Perceptual Applications. In *IEEE Security and Privacy*, 2013.
[7] J. Jeon, K. Micinski, J. Vaughan, A. Fogel, N. Reddy, J. Foster, and T. Millstein. Dr. Android and Mr. Hide: Fine-grained Permissions in Android Applications. In *IEEE SPSM*, 2012.
[8] P. Klemperer, Y. Liang, M. Mazurek, M. Sleeper, B. Ur, L. Bauer, L. Cranor, N. Gupta, and M. Reiter. Tag, You Can See It! Using Tags for Access Control in Photo Sharing. In *ACM SIGCHI*, May 2012.
[9] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Li. Learning Multi-scale Block Local Binary Patterns for Face Recognition. In *International Conference on Biometrics (ICB)*, 2007.
[10] M. Nauman, S. Khan, and X. Zhang. Apex: Extending Android Permission Model and Enforcement with User-defined Runtime Constraints. In *ASIACCS*, 2010.
[11] M. Ra, R. Govindan, and A. Ortega. P3: Toward Privacy-Preserving Photo Sharing. In *NSDI*, 2013.
[12] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev, and C. Glezer. Google android: A comprehensive security assessment. *IEEE Security and Privacy*, March 2010.
[13] R. Templeman, M. Korayem, D. Crandall, and A. Kapadia. PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces. In *NDSS*, 2014.
[14] M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.
[15] C. Write, C. Cowan, S. Smalley, J. Morris, and G. Kroah-Hartman. Linux Security Modules: General Security Support for the Linux Kernel. In *USENIX Security Symposium*, Aug 2002.
[16] R. Xu, H. Saidi, and R. Anderson. Aurasium: Practical Policy Enforcement for Android Applications. In *USENIX Security Symposium*, 2012.