Toward Trustworthy Mobile Sensing

Peter Gilbert
Duke University
Durham, NC, USA
gilbert@cs.duke.edu

Landon P. Cox
Duke University
Durham, NC, USA
lpcox@cs.duke.edu

Jaeyeon Jung
Intel Labs Seattle
Seattle, WA, USA
jaeyeon.jung@intel.com

David Wetherall
University of Washington
Seattle, WA, USA
djw@cs.washington.edu

ABSTRACT

Commodity mobile devices have been utilized as sensor nodes in a variety of domains, including citizen journalism, mobile social services, and domestic eldercare. In each of these domains, data integrity and device-owners' privacy are first-class concerns, but current approaches to secure sensing fail to balance these properties. External signing infrastructure cannot attest to the values generated by a device's sensing hardware, while trusted sensing hardware does not allow users to securely reduce the fidelity of readings in order to preserve their privacy. In this paper we examine the challenges posed by the potentially conflicting goals of data integrity and user privacy and propose a trustworthy mobile sensing platform which leverages inexpensive commodity Trusted Platform Module (TPM) hardware.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Security kernels

General Terms

Security

Keywords

Location privacy, mobile computing, participatory sensing, trusted platform module

1. INTRODUCTION

Many projects have proposed using consumer devices equipped with commodity sensors as a platform for highly-scalable, low-cost sensing. Domestic eldercare [7], citizen journalism [1], traffic monitoring [20], price-dispersion monitoring [13], mobile social services [18, 23], and environmental monitoring [25] are just a few contexts in which useful sensing can be performed by inexpensive consumer devices. Ensuring the privacy of participants in these services is a first-order concern. Data collected from human-operated devices should not inadvertently reveal any private information about the contributors. Unfortunately, consumer devices are not *trustworthy*, which prevents many services that would benefit from anonymous, user-generated content from fully utilizing it.

Many services currently rely on non-technical economic and legal frameworks to establish trust in images, video, and other forms of sensed data. For example, the New York Times requires freelance photographers to work under a contract stipulating that submitted images be consistent with the paper's "Guidelines on Integrity" [2, 4]. These guidelines specify how photographs and images can be altered, and a photographer is held accountable for adhering to the guidelines through their contract with the paper. However, relying on legal contracts to enforce image integrity is not scalable, and limits the paper's ability to cover fast-moving and dangerous stories such as the recently contested elections in Iran.

Events in Iran during the summer of 2009 were covered by a relatively small number of professional photographers due to the difficulty of entering the country and risk of violence against journalists by the government. At the same time, the large protests in Tehran and elsewhere were widely documented by thousands of anonymous sources with commodity mobile camera phones and video recorders. Anonymity was crucial for documenting these events; according to Reporters without Borders, 19 Iranian journalists have been imprisoned since the June 12th election [3]. If the New York Times and other news organizations could use a technical framework to establish trust in freelance photographs and video rather than relying on the legal system, they could take advantage of these large pools of user-generated content without compromising their integrity requirements.

Several recent projects have sought to address these issues by strengthening the integrity guarantees of user-generated content. One approach is to rely on trusted, co-located infrastructure to serve as a witness to the time and location of a device or data item [22, 27]. Unfortunately, relying on external location witnesses is limited to areas where infrastructure has already been deployed. Furthermore, simply verifying the location of a device or a submitted data item gives no assurance as to how it has been altered or manipulated; devices can easily submit "pre-manipulated" sensor data to the infrastructure for signing.

Alternatively, a consumer device could be paired with trusted hardware sensing peripherals, which sign their raw readings [15]. This allows a remote entity to verify that a reading was generated by a trusted peripheral using the peripheral's public key. Combining signed readings from multiple peripherals, such as a camera and GPS radio, would allow a service provider to verify the authenticity of both a reading and the context in which it was taken. Unfortunately, while this approach satisfies the needs of service providers, it is impractical for consumer devices. First, for many kinds of sen-

HotMobile '10 Annapolis, Maryland USA

sor data, uploading raw readings is too expensive, particularly for a mobile device. For example, sound and video must be uploaded in highly compressed formats to save energy. Second, even when transmitting raw readings is feasible, users may reveal too much private information by doing so. For example, users may wish to reduce the precision of their signed GPS readings to preserve their location privacy.

As a result, building a trustworthy mobile-sensing platform with consumer devices that satisfies the the privacy requirements of users and the integrity requirements of service providers is not possible with existing solutions. Our insight is that, for many applications, mobile-sensing platforms can become more trustworthy by: 1) allowing services to verify that submitted data was generated by a mobile device capturing data directly from hardware sensors, and 2) allowing the platform to apply a sequence of trusted transformations to raw readings in software before passing results to a service.

Establishing trust in code running on an otherwise untrusted platform can be enabled by Trusted Platform Module (TPM) hardware [6] included in nearly all commodity PCs on the market today. Recently, specifications for Mobile Trusted Module (MTM), which provides the essential functionality of TPM for mobile phone platforms, have been released [5]. We expect that consumer devices will ship with MTM hardware in the near future.

Our position is that trustworthy mobile sensing can resolve existing tensions between privacy and data authenticity, and can be realized through on-device TPM technology. In this paper, we examine several representative sensing applications that could benefit from stronger authenticity guarantees, describe how TPM can be leveraged to enable trusted sensing, outline properties that such a system should provide, propose a possible system architecture, and explore tradeoffs associated with various design decisions.

2. APPLICATION SCENARIOS

In this section we briefly explain how trustworthy consumer devices would improve service quality in three mobile-sensing domains: citizen science and journalism, mobile social services, and health monitoring.

2.1 Citizen science and journalism

Numerous projects have proposed augmenting conventional monitoring of political, scientific, and economic phenomena with user-generated content from consumer devices [1, 11, 13, 25, 26]. User-generated content from consumer devices offers a low-cost data source for monitoring large-scale phenomena such as global warming and dangerous events such as contested election results. Unfortunately, existing bases for trust simply cannot scale to these vast new data pools.

Trust in journalism and the sciences is based on economic, legal, and social frameworks that assign reputations to contributors and provide individuals with strong incentives to adhere to well-defined integrity standards. Contributors of user-generated content are difficult to fit within these existing frameworks because their individual trustworthiness is difficult to evaluate: content may come from users with no established reputation, users who wish to remain anonymous due to privacy concerns, or users whose reputations are based on systems that are prone to Sybil-style gaming [14].

A trusted platform for mobile sensing would alleviate this problem by decoupling the reputations of individual contributors from the integrity of the data they produce. Such a platform would allow users to present attestations that their sensed data was authentic the first time they contribute and without forcing them to reveal any sensitive information about themselves. Of course, as is the case in all existing frameworks, motivated adversaries will always be able to undermine trust by forging sensor data through "analog" attacks on the sensors themselves (e.g., staging a photograph or putting a temperature sensor in a refrigerator), but we believe that these kinds of attacks will be uncommon. Rather, our proposed platform is intended to give consumers of user-generated content stronger integrity guarantees than any framework currently supports.

2.2 Mobile social services

Mobile social services also stand to benefit from more trustworthy data. In a mobile social service, users forward their location information to a service provider, which coordinates interactions among participants based on the locations it receives. Localization is usually performed by the device using its GPS, Wifi, or GSM radios. Live, remote query systems such as Micro-Blog [18] could leverage trustworthy localization to inform users when their location-based query is answered by someone who can prove that they are present in the area of interest. Similarly, game-oriented services such as Foursquare [17], in which users earn points, titles, and prizes by "checking in" from certain locations, could use signed location information to prevent cheating.

2.3 Domestic healthcare

Finally, mobile devices have been proposed in numerous domestic healthcare scenarios [7, 8, 9]. For example, a home-based care model has been proposed as an alternative to traditional cardiac rehabilitation programs that have been underutilized due to inadequate service provisioning, high cost, and long-term commitment required by patients [9]. The home-based cardiac rehabilitation program collects a patient's health status via sensors monitoring physiological signals (e.g., ambulatory ECG monitors) and body movements (e.g., accelerometers) and provides personalized feedback remotely as to an appropriate exercise level and the use of preventive medications. A patient's mobile phone can serve as an ideal platform to process various sensor data for continuously monitoring the cardiac condition and for regularly uploading the sensor data for a clinician's review. With the increasing computing power and always-on connectivity, commodity mobile phones could be easily adopted for such a home-based healthcare program, thus further reducing the cost.

However, as with the previous examples, the lack of trustworthy mobile platforms could impede these potentially high-value domestic healthcare services. First, stronger assurance of a patient's privacy is required since unlike in-clinic treatment, a patient's sensor data is continuously collected in his daily life. For instance, without clear benefits, a patient may not be willing to share his location data with a clinician while participating in the rehabilitation program and would not enroll in the program without these privacy guarantees. Second, data-authenticity guarantees are required for these services to be accountable and audited. For instance, an insurance company needs to verify that the claimed service was provided and that data was properly collected and processed. Auditing could be simplified if a patient's mobile phone generated signed event logs that could be compared against later claims.

3. RELATED WORK

Several different approaches have been proposed for verifying the integrity of sensor data collected by mobile devices. Dua et al. propose a system which allows a mobile device to attest to the integrity of sensor readings using a trusted hardware sensing peripheral [15]. Each mobile device is paired with a peripheral containing sensors and a TPM chip used to attest to the integrity of the software running on the peripheral and to sign sensor readings. This allows the TCB to remain free of any code running on the mobile

device itself. However, the use of specialized hardware may be a barrier to widespread deployment, and the system only allows a device to attest to the integrity of raw data captured from sensors.

Other approaches address the problem of verifying the geographical location of a participating mobile device. The association between a sensor reading and the location at which it was collected is essential in sensing applications. In [22] and [27], timestamped location certificates signed by wireless infrastructure are issued to co-located mobile devices. A user can collect certificates and later provide them to a remote party as verifiable proof of her location at a specific time. However, the applicability of these infrastructure-based approaches for mobile sensing is limited as cooperating infrastructure may not be present in remote or hostile environments of particular interest to some applications.

The technique of verifying input from hardware devices has also been applied to the problem of detecting human activity—specifically, detecting whether email messages generated by a potentially-compromised client machine were produced by a bot or by a legitimate user. Not-a-bot [19] runs trusted code on the client machine which captures keyboard and mouse input and provides attestations for outgoing messages only when recent keyboard or mouse input is detected. The heuristic of proximity in time between human activity and message transmission was chosen because it proved to be difficult to determine whether the message contents were produced by a given sequence of keyboard and mouse activity. In contrast, in the domain of mobile sensing, data is often generated automatically by the sensing application through a sequence of device reads and transformations such as compression or feature extraction. In this case, it is feasible to verify that a data item was produced by a series of trusted operations.

Flicker [24] uses the late launch capability supported by recent commodity processors to allow an application-specific block of trusted code to be executed at runtime on a client machine running an untrusted software platform. Flicker can generate a verifiable attestation of the integrity of the code as well as its inputs and outputs. While Flicker is lightweight and adds only a few hundred lines of code to the TCB, it is not designed for complex operations like extracting data from hardware devices, which spans many layers of the software stack, including kernel-level driver code.

Finally, Nexus [29] is a TPM-based operating system that removes device drivers from the trusted computing base by using trusted reference monitors to enforce driver-safety specifications. The Nexus device driver architecture is relevant to trusted mobile sensing platforms because it provides a framework for enforcing safe behavior by the driver code that interacts with physical sensing devices. It is not clear if the Nexus safety-specification language is expressive enough to build a trusted path from the raw sensor readings generated by hardware to user-friendly images and geotags, but enforcing safety properties of untrusted driver code rather than including it in the trusted computing base is an appealing approach.

4. CHALLENGES

The problem of building trustworthy services on mobile sensing is different from many traditional trusted computing scenarios in that it requires providing assurances to multiple stakeholders whose interests may at times conflict. Service providers wish to gain assurance that contributed data is authentic, or extracted from sensor hardware and manipulated only by trusted software. Participating device owners must be assured that a sensing application releases only expected, acceptable data, and that it does not pose an unnecessary privacy risk. In fact, as previously mentioned, it may be important for content contributors to remain totally anonymous in some cases.

These requirements present a unique tension and pose several new challenges for trustworthy mobile computing. We categorize the major challenges as those related to achieving two goals: 1) enabling a service provider to trust content generated by software running on a mobile device, and 2) defining and ensuring privacy properties for participating device owners.

4.1 Local data processing

Enabling a service provider to trust that a reported sensor reading was indeed captured by a sensor attached to a mobile device is a key feature of trustworthy mobile sensing. A previous approach, attractive in its simplicity, is to employ a piece of trusted hardware or low-level software to sign raw data captured from hardware sensors [15, 19]. However, data reported by a mobile device is often several steps removed from raw sensor data: it may be compressed, or processed to extract high-level features or preserve privacy. Data compression or feature extraction may be necessary due to the high costs of wireless communication for energy-constrained mobile devices; for example, transmitting uncompressed image or video data will certainly result in excessive drain on the battery.

Sensor data may also be processed locally to preserve the privacy of the device owner. For example, a user may wish to report that she is located in the city Durham or the state of NC rather than revealing her precise location by reporting GPS coordinates or the results of a Wifi scan. In this case, location data captured from sensors is transformed by applying a fidelity reduction locally before submitting it to the service.

In general, content may need to be processed locally by a number of applications or libraries before being sent to a remote service provider. As a result, in order to verify the authenticity of the data, a service must not only gain assurance that the original input was captured directly from hardware sensors, but also establish trust in the applications which processed the data. Ideally, this should be enabled without excessively increasing the size of the TCB. These requirements are unique to building a trusted mobile sensing platform and necessitate a novel system design.

4.2 Device owner privacy

Mobile phones store a wealth of personally identifying information such as phone numbers and IMEI numbers. Furthermore, devices are increasingly likely to be equipped with localization technologies such as GPS and Wifi which are capable of determining the geographical location of the device with high precision. Consequently, an application which has access to privacy-sensitive data or certain hardware sensors could compromise the anonymity or location privacy of the device owner.

This issue illustrates a fundamental tension which arises when trying to build a trustworthy mobile platform: there exist multiple stakeholders, with potentially conflicting interests, who wish to gain assurances about aspects of the software platform running on the mobile device. This includes the mobile device owner, services which want to establish trust in application code running on the device, and potentially even network carriers. We focus on the conflict which exists between the need of a sensing application to ensure data integrity and the desire of the device owner to not disclose her location or identity.

Providing participating device owners with meaningful privacy assurances presents a challenge. We believe that the simple approach of requiring device owners to place unconditional trust in application code provided by service providers is insufficient. Economic and social incentives may exist for a service provider to collect additional user data beyond the stated purpose of the service. Furthermore, service providers who are not particularly con-

cerned with user privacy may compromise user privacy through negligence. As a result, we would like to protect against privacy threats posed by both malicious service providers as well as unintentional disclosures perpetrated by poorly designed applications. The system should provide privacy assurances to device owners beyond unsubstantiated claims offered by service providers.

On the other hand, attempting to provide strong privacy guarantees about software running on the mobile device may preclude applications which are useful and appropriate. For example, traditional information flow techniques can provide strong guarantees; however, if paired with overly strict policies, they could impose restrictions on disclosure which prevent legitimate applications. Instead, we want to allow potentially privacy-sensitive sensor data to be released in a way that is acceptable to the device owner. The key challenge is to enable rich applications while assuring the device owner that a sensing application exposes "no more than is expected and needed." In some cases, contributors may wish to remain totally anonymous; in other cases it may be sufficient to not reveal precise location data. We must provide privacy assurances which are compatible with both application needs and user expectations.

5. TRUSTED PLATFORM APPROACH

As discussed in the previous section, trustworthy mobile sensing services face the challenge of providing assurances to both service providers and device owners. Enabling a service provider to trust that contributed sensor data is authentic requires providing verifiable proof that the data was generated by trusted software, running in isolation on a mobile device. Trusted Platform Module (TPM) hardware, commonly provided in commodity PCs, can be leveraged to help provide this assurance. To address the problem of protecting the privacy of data contributors, we consider techniques such as requiring explicit authorization for applications to access local resources and formulating and enforcing access control policies.

This section outlines our proposed approach for building a trustworthy mobile sensing platform. It provides background on the mechanisms provided by TPM which can be applied to allow a service provider to establish trust in data-processing software running on a mobile device. Discussions follow about how the service provider's TCB could be structured as well as how to generate attestations. Finally, we consider potential approaches for protecting the privacy interests of contributing device owners.

5.1 TPM background

A TPM [6] is a relatively inexpensive hardware component used to facilitate building trusted software systems. Our proposed system leverages the TPM functionality of *attesting* to the integrity of software running on a device to a remote *verifier*. A TPM can be used to enable trusted boot, where each piece of code loaded from boot-time is measured via SHA-1 cryptographic hash before loading. Each measurement is recorded into a Platform Configuration Register (PCR) using the TPM *extend* operation: the value of the PCR is updated with the hash of the current value of the PCR concatenated with the measurement. Static PCRs can only be modified through the extend operation or reset via a reboot.

The TPM can attest to the software platform running on the machine by providing a signed *quote* of its PCR(s) in response to a challenge from a remote verifier. The TPM can generate quotes anonymously by signing them with a private Attestation Identify Key (AIK); the public component can be verified using a certificate issued by a trusted third-party certificate authority. The remote verifier can validate the software stack by comparing the PCR values provided in the quote to the expected values.

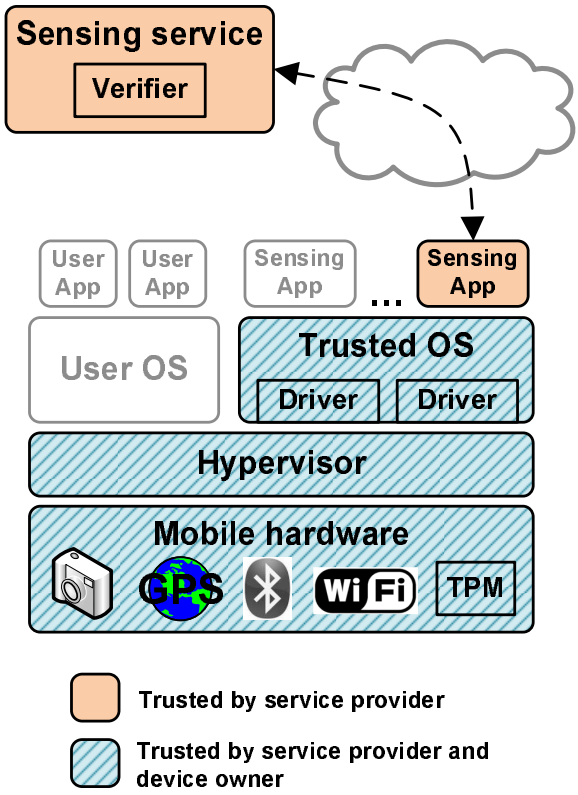

Figure 1: Hypervisor-based system architecture

Our proposed system utilizes another operation provided by a TPM, *sealed storage*, which can encrypt data and bind it to a specific software state. The TPM will decrypt sealed data only when the proper software state can be verified through one or more PCR values. Sealed storage can be used to protect private keys which should only be accessed by specific trusted software.

Specifications have been released for Mobile Trusted Module (MTM) [5], which provides the essential functionality of TPM but is aimed at mobile platforms rather than PCs.

5.2 Structuring the service provider's TCB

A fundamental question in designing any trusted platform is that of how to structure the TCB. In general, system designers embrace the principle that a system can be made more secure by reducing the size of the TCB as much as possible to include only security-critical code. In our case, the service provider's TCB must include at a minimum any code which has the opportunity to modify sensor data on the mobile device at any point from when it is extracted from hardware until it is signed by trusted local code before being submitted to a remote verifier. Therefore, sensor device drivers as well as any running applications or libraries which are able to modify the data must be included.

Because device drivers and applications necessarily rely on and invoke OS code, our assurance requirements imply that the OS upon which the drivers and applications run must be included in the TCB as well. Despite being designed for resource-constrained devices, current commodity mobile OSes are comparable in complexity to those designed for traditional PCs and could add millions of lines of code to the TCB, posing a serious threat to the integrity of the system. To make matters worse, commodity mobile OSes typically do not ensure strong isolation among applications, poten-

tially requiring all applications running on the mobile device to be added to the service provider's TCB as well. It is clear that such a design would result in a system which is neither practical nor trustworthy.

Our approach to limiting the size of the TCB, depicted in Figure 1, is similar to that employed by the Not-a-bot system [19], which avoids including the user's OS and applications in the TCB by instead attesting to the trusted boot of a secure, minimal hypervisor. Secure mobile hypervisors enforce strong isolation among multiple virtual machines, or domains, each capable of running a full commodity mobile OS [12]. The use of a secure hypervisor could allow trusted driver, application, and library code run inside a special domain, running a minimal trusted OS with privileged access to hardware. The device owner's untrusted OS and applications could run in a separate guest domain, with the hypervisor enforcing strong isolation among domains. This design results in a TCB for the service provider which contains only the minimal hypervisor and trusted OS in addition to critical driver, application, and library code used by the mobile sensing service. In [19], this approach is demonstrated using a stripped-down version of the Xen hypervisor and mini-OS on a PC. Similar virtualization platforms are emerging for mobile architectures [10].

5.3 Generating attestations

To provide assurance to a sensing service provider that data received from a mobile device is authentic, the mobile platform must present verifiable proof of two properties: 1) that the mobile device correctly loaded the trusted software platform from boot (i.e., hypervisor, trusted OS and drivers, sensing applications), and 2) that the data item in question was generated by the trusted sensing application.

To meet these requirements, each data item submitted by a mobile device must be accompanied by an attestation consisting of four components: a log of all software loaded into the TCB since booting, a TPM quote over the PCR used to store measurements of loaded software, a cryptographic hash over the contents of the data item, and a digital signature over the first three items. The signature is computed by the sensing application using the private half of a key pair associated with the local application, generated at install-time. The private key can be protected from untrusted software running on the device using the TPM sealed storage feature.

The provider runs a remote verifier on its own servers which can verify the integrity of the software log by repeating the computation that generated by final PCR value and comparing results, and can verify the value of the data hash using the public key associated with the specific sensing application installation. If the log and hash are valid, the provider can then evaluate the authenticity of the data item based on the list of software loaded into the TCB. A public database of certificates validating the integrity of well-known software, issued by a trusted authority, could serve as a reference to aid this evaluation.

5.4 Protecting device owner privacy

Privacy issues always accompany mobile sensing services since mobile device uses are tightly integrated with the device owner's everyday life. For example, even seemingly innocuous data such as Wifi scan results can be used to track the movement of a device owner if released with a unique identifier such as a MAC address or IMEI. Although the possibility of such an incident may be low, potential privacy risks could be a barrier to wide adoption of mobile sensing services, especially if the service relies on high-fidelity sensing data such as audio [21].

As Figure 1 shows, our system design fundamentally separates any third party sensing applications from the trusted core of the user's system. In addition, we argue that two important platform features are required for protecting device owner privacy. First, the system must prevent sensing applications from accessing local resources without explicit authorization. The authorization process could happen during the install as employed by Android phones or at the start of an application as on iPhones. The controlled resources should include hardware, sensors, user data, and system configuration information.

Second, the platform must provide a mechanism to monitor whether sensing applications inadvertently or maliciously release private information and block such attempts. For instance, while users may allow a price-comparison application to access camera for bar code processing, they would loathe to see raw photos that may contain personal images transmitted to the application's server. We believe that the proposed architecture will greatly simplify such enforcement as untrusted sensing applications run in isolation inside the trusted OS. Mandatory access control (MAC) operating systems [31, 16] can provide information-flow control mechanisms to block information leaks. If these are not an option, the hypervisor can be instrumented to support dynamic information-flow security [28, 30] for personal data protection.

Besides standard systems challenges associated with implementing these techniques on a mobile platform, implementing them without compromising the privacy needs of service providers poses an additional challenge. For example, while the dynamic information-flow tracking technique has proved effective for providing fine-grained control of sensitive data, the same technique can be used for reverse engineering of service provider's potentially proprietary code. Moreover, the aforementioned attestations would be also affected if the trusted OS invokes a privacy checker irrespective of a sensing application's desire.

6. CONCLUSION

In this paper, we have argued that mobile sensing applications which rely on participants with commodity devices to contribute data could benefit from stronger data authenticity and privacy assurances. Service providers would like to verify that contributed data was extracted directly from hardware sensors and manipulated only through trusted software operations. Device owners must be assured that their privacy is not unnecessarily compromised. Due to this tension between the interests of service providers and device owners, building services which provide both of these properties presents a challenge.

We proposed building a trustworthy sensing platform for commodity mobile devices using TPM hardware and techniques such as access control policies and explicit user authorization to protect the privacy of participants. We believe that such a platform would increase the value of mobile sensing services for both service providers and participating device owners through improved data authenticity and privacy assurances.

7. ACKNOWLEDGMENTS

We would like to thank our shepherd, M. Satyanarayanan, and the anonymous reviewers for their helpful comments and insight. This material is based in part on work supported by the National Science Foundation under NSF awards 0747283 and 0916649.

8. REFERENCES

[1] iReport. `http://www.ireport.com/`.

[2] New York Times: Guidelines on Integrity. http://www.nytco.com/company/business_units/integrity.html.

[3] Reporters Sans Frontières. http://www.rsf.org/.

[4] The Photographers' Room: New York Times Photo Ethics Policy. http://www.thephotographersroom.com/blog/?p=615.

[5] Trusted computing group - mobile - specifications. http://www.trustedcomputinggroup.org/developers/mobile/specifications/.

[6] Trusted computing group - trusted platform module - specifications. http://www.trustedcomputinggroup.org/developers/trusted_platform_module/specifications/.

[7] *Handbook of Digital Homecare*, chapter A Multi-Modal Health and Activity Monitoring Framework for Elderly People at Home, pages 287–298. In [8], 2009.

[8] *Handbook of Digital Homecare*. Springer Berlin Heidelberg, 2009.

[9] *Handbook of Digital Homecare*, chapter A Home-Based Care Model of Cardiac Rehabilitation Using Digital Technology, pages 329–352. In [8], 2009.

[10] S. bum Suh. Secure architecture and implementation of xen on arm for mobile devices. Xen Summit, Spring 2007, IBM T.J. Watson.

[11] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Rmanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. In *World-Sensor-Web*, 2006.

[12] L. P. Cox and P. M. Chen. Pocket hypervisors: Opportunities and challenges. In *HOTMOBILE '07: Proceedings of the Eighth IEEE Workshop on Mobile Computing Systems and Applications*, pages 46–50, Washington, DC, USA, 2007. IEEE Computer Society.

[13] L. Deng and L. P. Cox. Livecompare: grocery bargain hunting through participatory sensing. In *HotMobile '09: Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, 2009.

[14] J. R. Douceur. The sybil attack. In *IPTPS*, 2002.

[15] A. Dua, N. Bulusu, W.-c. Feng, and W. Hu. Towards trustworthy participatory sensing. In *HotSec '09: Proceedings of the Usenix Workshop on Hot Topics in Security*, 2009.

[16] P. Efstathopoulos, M. Krohn, S. VanDeBogart, C. Frey, D. Ziegler, E. Kohler, D. Mazières, F. Kaashoek, and R. Morris. Labels and event processes in the asbestos operating system. In *SOSP*, 2005.

[17] Foursquare. http://www.foursquare.com.

[18] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: sharing and querying content through mobile phones and social participation. In *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 174–186, New York, NY, USA, 2008. ACM.

[19] R. Gummadi, H. Balakrishnan, P. Maniatis, and S. Ratnasamy. Not-a-bot: improving service availability in the face of botnet attacks. In *NSDI '09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, 2009.

[20] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *MobiSys*, pages 15–28, New York, NY, USA, 2008. ACM.

[21] P. Klasnja, S. Consolvo, T. Choudhury, R. Beckwith, and J. Hightower. Exploring privacy concerns about personal sensing. In *Pervasive '09: Proceedings of the 7th International Conference on Pervasive Computing*, pages 176–183. Springer-Verlag, 2009.

[22] V. Lenders, E. Koukoumidis, P. Zhang, and M. Martonosi. Location-based trust for mobile user-generated content: applications, challenges and implementations. In *HotMobile '08: Proceedings of the 9th workshop on Mobile computing systems and applications*, 2008.

[23] J. Manweiler, R. Scudellari, and L. P. Cox. SMILE: Encounter-based trust for mobile social services. In *Proceedings of ACM CCS 2009*, November 2009.

[24] J. M. Mccune, B. J. Parno, A. Perrig, M. K. Reiter, and H. Isozaki. Flicker: An execution infrastructure for tcb minimization. In *Eurosys '08: Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems*, 2008.

[25] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *MobiSys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, 2009.

[26] E. Paulos, M. Foth, C. Satchell, Y. Kim, P. Dourish, and J. H. jeong Choi. Ubiquitous sustainability: Citizen science and activism. In *UbiComp Workshops 2008*, 2008.

[27] S. Saroiu and A. Wolman. Enabling new mobile applications with location proofs. In *HotMobile '09: Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, 2009.

[28] N. Vachharajani, M. J. Bridges, J. Chang, R. Rangan, G. Ottoni, J. A. Blome, G. A. Reis, M. Vachharajani, and D. I. August. RIFLE: An architectural framework for user-centric information-flow security. In *MICRO*, 2004.

[29] D. Williams, P. Reynolds, K. Walsh, E. G. Sirer, and F. B. Schneider. Device driver safety through a reference validation mechanism. In *OSDI*, 2008.

[30] H. Yin, D. Song, M. Egele, C. Kruegel, and E. Kirda. Panorama: capturing system-wide information flow for malware detection and analysis. In *CCS*, 2007.

[31] N. Zeldovich, S. Boyd-Wickizer, E. Kohler, and D. Mazières. Making information flow explicit in HiStar. In *OSDI*, 2006.