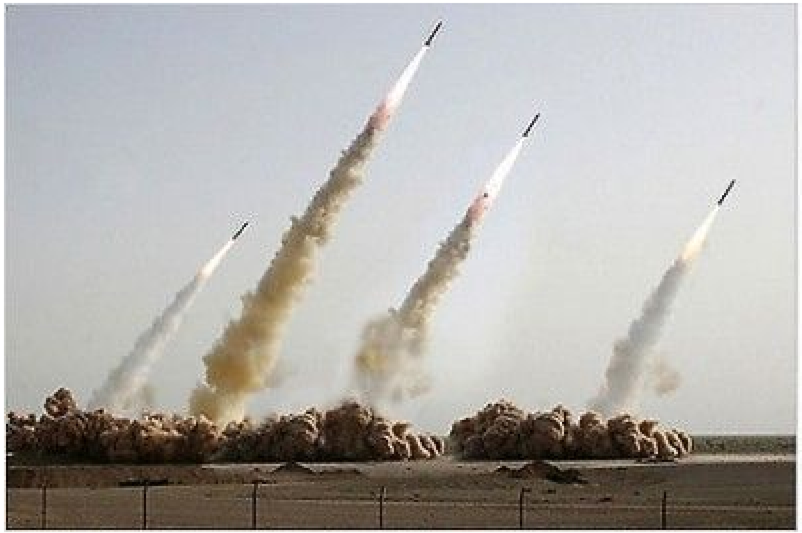
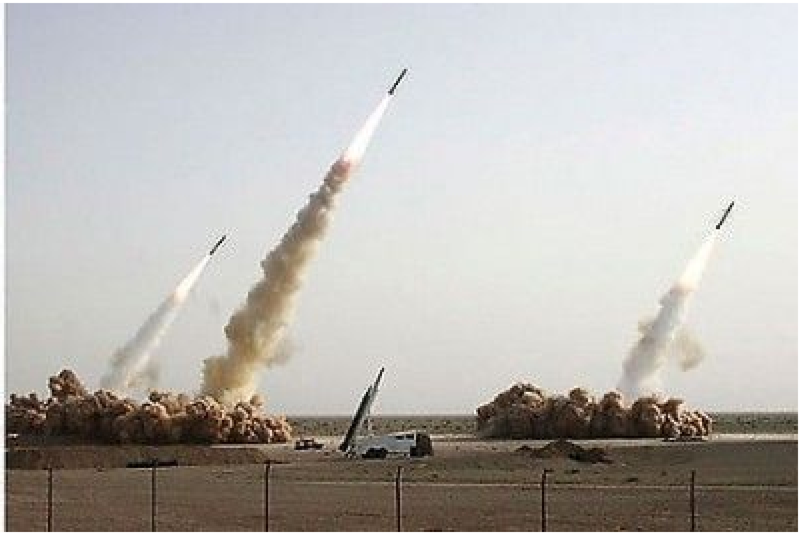
JULE 2008. IRAN'S PROVOCATIVE MISSILE TESTS. AS THE ILLUSTRATION SHOWS, THE SECOND MISSILE FROM THE RIGHT APPEARS TO BE THE SUM OF TWO OTHER MISSILES IN THE IMAGE.

FALSE



TRUE



MAY 2007. SO IF YOU WATCH THE IMAX VERSION OF HARRY POTTER, YOU GET TO SEE EMMA WATSON'S BREAST GET BIGGER AND IN 3D TOO!

FALSE

TRUE

YOUPROVE: AUTHENTICITY AND FIDELITY IN MOBILE SENSING

PETER GILBERT

DUKE UNIVERSITY

JAEYEON JUNG

MICROSOFT RESEARCH

KYUNGMIN LEE

DUKE UNIVERSITY

HENRY QIN

DUKE UNIVERSITY

ANMOL SHETH

TECHNICOLOR RESEARCH

DANIEL SHARKEY

DUKE UNIVERSITY

LANDON P. COX

DUKE UNIVERSITY

INTRODUCTION

The next generation of Internet platforms promises to support services like:

- Citizen journalism
- Mobile social networking
- Environmental monitoring
- Traffic monitoring

by pairing the ubiquitous sensing provided by mobile phones with the large-scale data collection and dissemination capabilities of the cloud.

Data authenticity is crucial for service correctness

Correctness is especially important for services such as Al Jazeera’s Sharek and CNN’s iReport. Deploying trusted reporters and photographers into events such as those recently experienced in Iran, Haiti, Tunisia, Egypt and Libya is difficult

"With the Arab Spring and the Iranian protests in 2009, we relied on citizen journalists for information," said Cox. "But as crowd-sourced content plays an increasingly important role in world affairs, falsified media could have severe consequences. It's important that we make sure the information we are getting is accurate."

Solution

Equip phones with trustworthy sensors capable of signing their readings and to require clients to return unmodified signed data to a service

But

Mobile clients require the flexibility to trade-off data fidelity for efficient resource usage and greater privacy

So

Trusted hardware such as a Trusted Platform Module (TPM) can serve as the foundation of a solution. A partnership between a device’s trustworthy hardware and its system software can produce digitally-signed statements about a data item’s “chain of custody” to a remote service.

The key to YouProve approach is type-specific analysis of derived sensor data.

- Type-specific analysis can be implemented by using well-known audio-analysis and computer-vision libraries to compare the content of a source item (e.g., an original audio clip or photo) to the content of a derived version of the item.

The goal of type-specific analysis is to allow client applications to apply fidelity-reducing modifications to data and to give services a basis for trusting that those modifications preserved the meaning of the source.

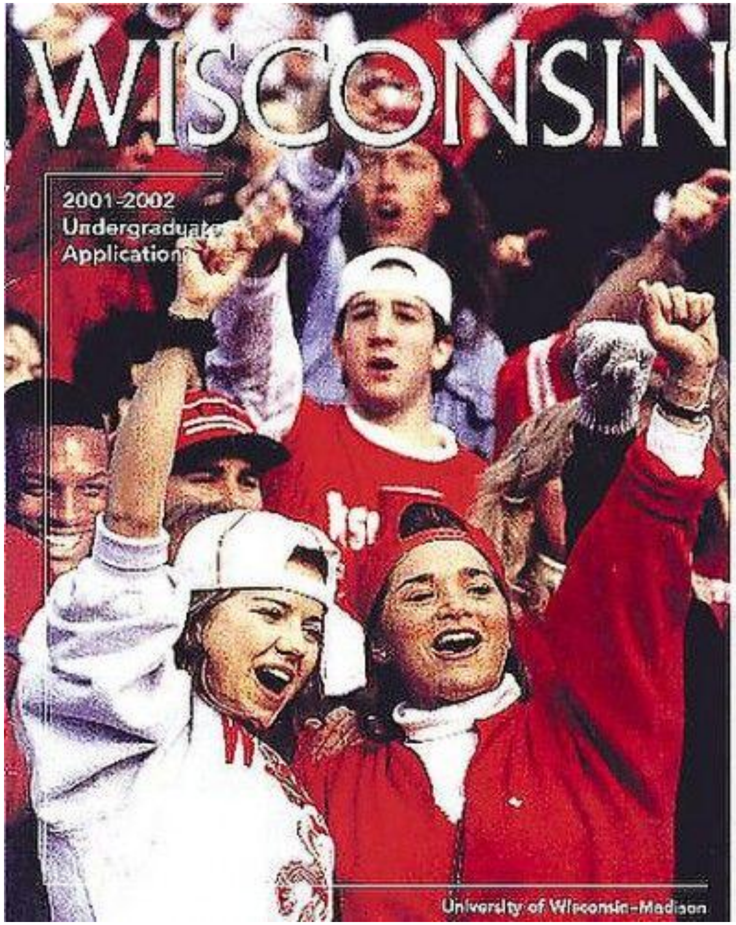
DESIGN CONSIDERATIONS

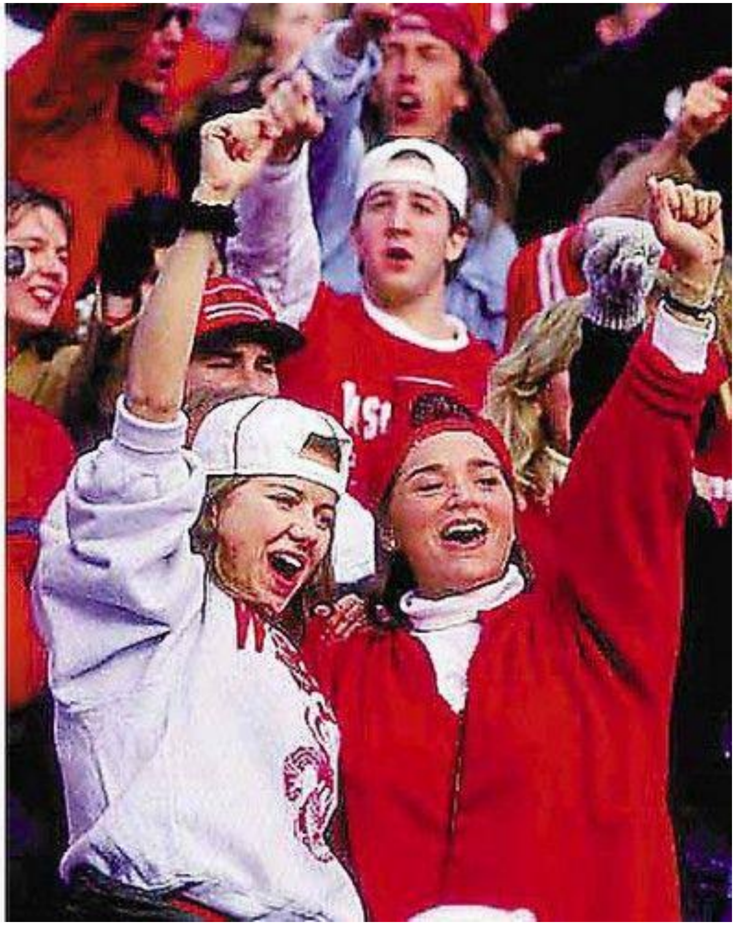
Mobile sensing services (also called participatory sensing) consist of servers that collect, aggregate, and disseminate geo-tagged sensor data such as audio and images from volunteer mobile clients.

The case for fidelity-aware mobile clients is well established, while the case for verifying the authenticity of sensing data has been made more recently. YouProve is a trustworthy sensing platform built on Android that allows a client to control the fidelity of data it submits and sensing services to verify that the meaning of source data is preserved across any modifications.

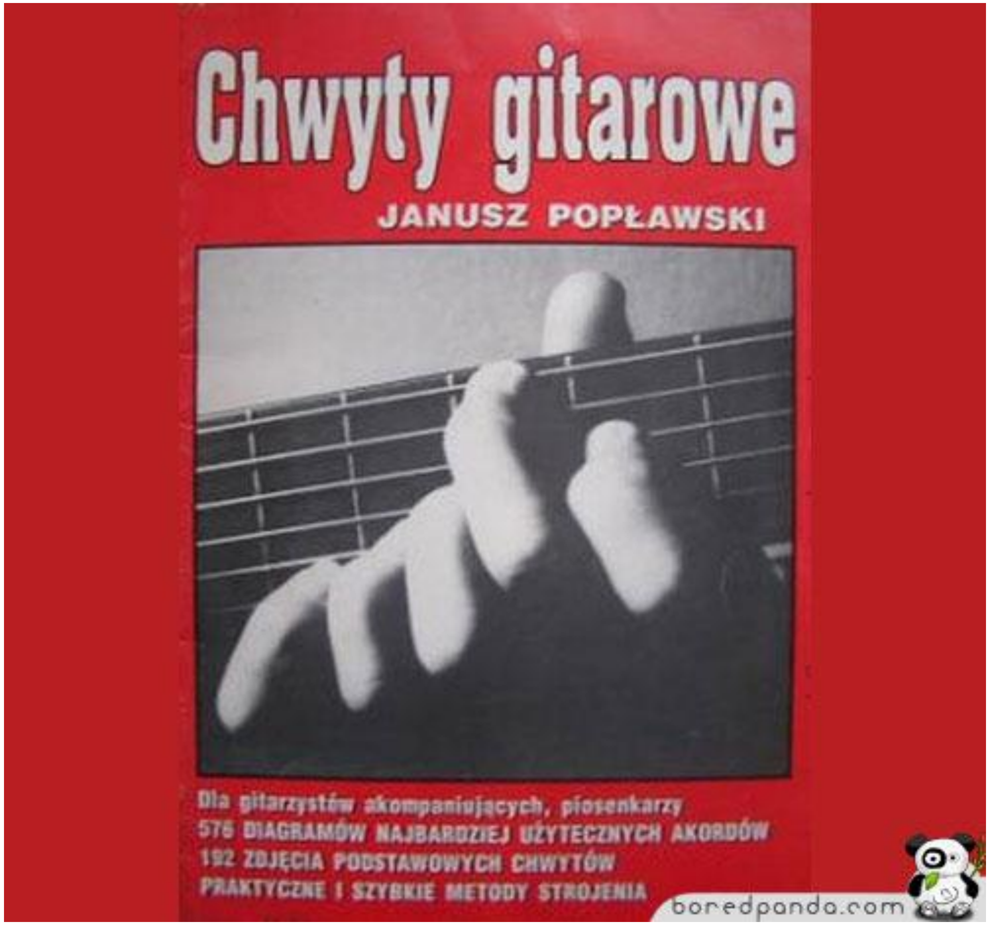
- Several groups have sought to decouple client reputations from data authenticity using trusted hardware such as a Trusted
- Platform Module (TPM) or ARM TrustZone. TPMs are included in most PCs sold today, and a specification for a Mobile Trusted Module (MTM) for mobile phones has been released.
- Similarly, most shipping ARM processors support TrustZone, which is a hardware-isolated, secure-execution environment that could be leveraged by phone manufacturers to implement functionality similar to a TPM’s.
- We take the presence of trusted hardware on mobile clients as a given, and we have designed YouProve as a set of software services running on top of such hardware.

SEPTEMBER 2000, WISCONSIN





GOOD GUITARIST HAVE 5 FINGERS, BEST ONES HAVE 6


Chwyty gitarowe
JANUSZ POPŁAWSKI
Dla gitarzystów akompaniujących, piosenkarzy
576 DIAGRAMÓW NAJBARDZIEJ UŻYTECZNYCH AKORDÓW
192 ZDJĘCIA PODSTAWOWYCH CHWYTÓW
PRAKTYCZNE I SZYBKIE METODY STROJENIA
boredpanda.com

YOUPROVE

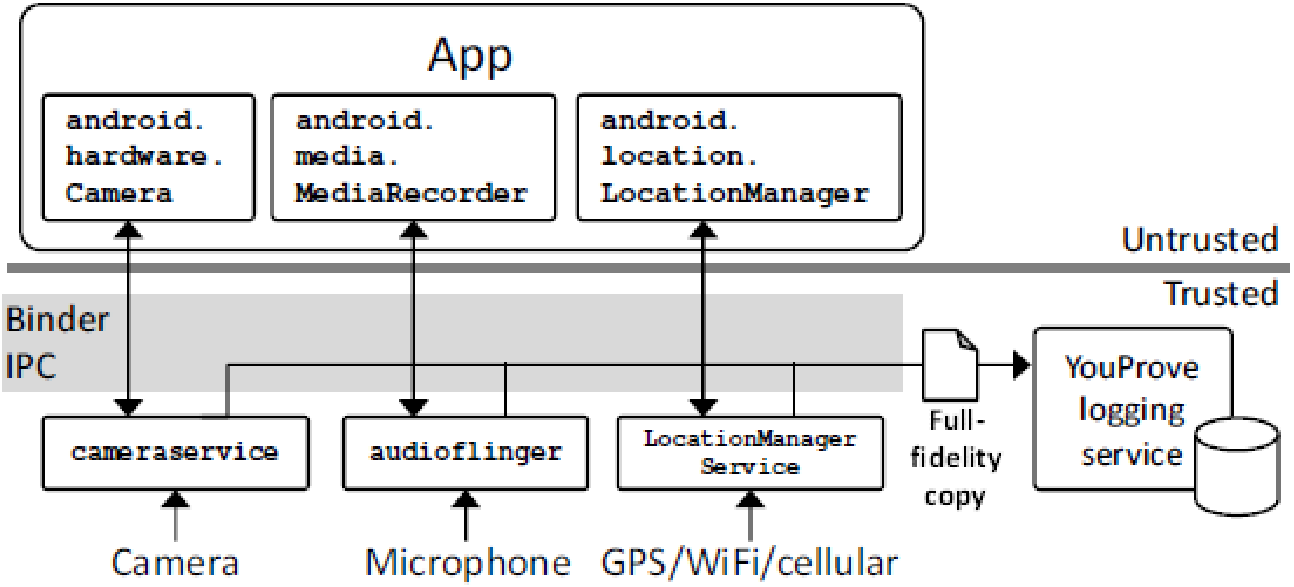
YouProve consists of four trusted software components responsible for performing the following tasks:

- *Logging* sensor data returned by the Android platform in response to requests from apps
- *Tracking* data derived from sensor readings as it is manipulated by untrusted third-party apps
- *Analyzing* the content of a derived data item and its source reading
- *Attesting* to the results of content analysis and the integrity of the software platform.

LOGGING SENSOR READINGS

YouProve makes trustworthy statements about the content of a derived data item by comparing it to source data captured by a sensor. To support this analysis, it is necessary for the trusted platform to collect a full-fidelity copy of any sensor reading returned to an application and to protect the integrity of the stored copy as long as a user wishes to generate fidelity certificates for data derived from the reading. YouProve's logging service provides this functionality.

LOGGING SENSOR READINGS



TRACKING DERIVED DATA

To enable comparisons YouProve relies on the *TaintDroid* information-flow monitoring framework as a lightweight means for tracking data dependencies throughout the Android system.

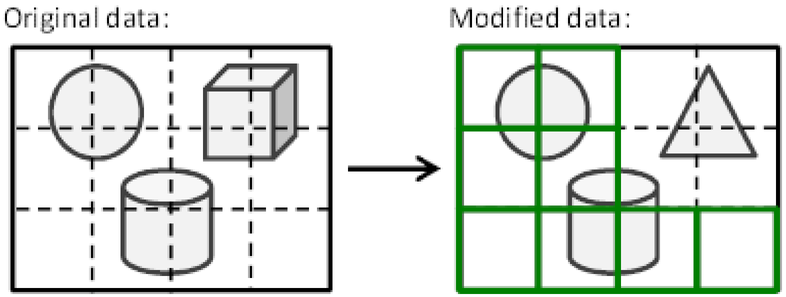
Before the platform returns sensor data to a user app, it attaches a taint tag encoding the *32-bit unique ID* assigned to the sensor reading—this ID serves as the primary key for the entry in the log database.

If tainted data is appended to a file or IPC message already marked with a different ID, the file or message is marked with a newly allocated ID and the mapping to the two previous IDs is recorded in the log.

ANALYZING CONTENT

For both photo and audio data, YouProve's approach is to divide a derived data item into smaller regions and then attempt to match the content of each region to that of a corresponding region in the source sensor reading. Photos are divided by rectangular grid, while audio analysis considers time segments.

OVERVIEW OF PHOTO AND AUDIO CONTENT ANALYSIS



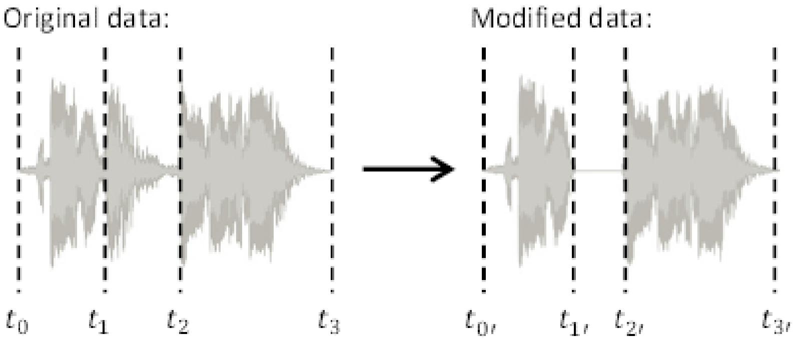
Output of photo analysis:

for $(i, j) \in ((0{,}0), (0{,}1), (1{,}0), (1{,}1), (2{,}*))$

$block_{i,j} \, \Delta \, block'_{i,j} \approx 0$

for $(i, j) \in ((0{,}2), (0{,}3), (1{,}2), (1{,}3))$

$block_{i,j} \, \Delta \, block'_{i,j} \gg 0$



Output of audio analysis:

$[t_0, t_1] \leftrightarrow [t_{0'}, t_{1'}]$

$[t_2, t_3] \leftrightarrow [t_{2'}, t_{3'}]$

YOUPROVE DEMO

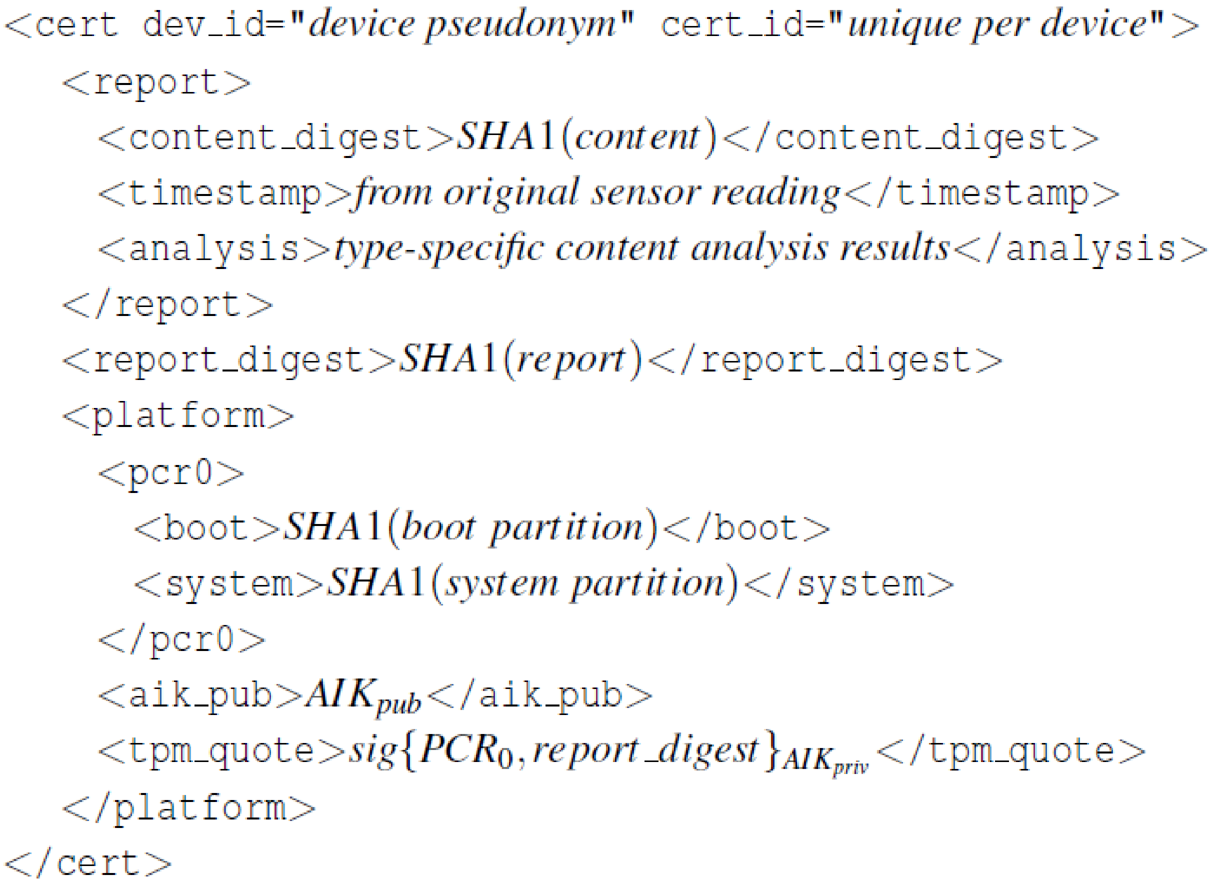
ATTESTING TO ANALYSIS AND PLATFORM

To enable data consumers to reason about the trustworthiness of a data item, YouProve's attestation service generates *fidelity certificates* that report the results of type-specific analysis and a timestamp for the original reading.

Fidelity certificates also contain information about the device's software platform, allowing remote verifiers to decide whether or not to trust reports generated by the device.

The two basic parts are a report that describes a data item and is bound to the data by a content digest, and a description of the platform software configuration, including a TPM quote that attests to the state of the software platform and binds the platformspecific part of the certificate to the content-analysis part.

FORMAT OF A FIDELITY CERTIFICATE



```
<cert dev_id="device pseudonym" cert_id="unique per device">
  <report>
    <content_digest>SHA1(content)</content_digest>
    <timestamp>from original sensor reading</timestamp>
    <analysis>type-specific content analysis results</analysis>
  </report>
  <report_digest>SHA1(report)</report_digest>
  <platform>
    <pcr0>
      <boot>SHA1(boot partition)</boot>
      <system>SHA1(system partition)</system>
    </pcr0>
    <aik_pub>$AIK_{pub}$</aik_pub>
    <tpm_quote>$sig\{PCR_0, report\_digest\}_{AIK_{priv}}$</tpm_quote>
  </platform>
</cert>
```

TYPE-SPECIFIC ANALYZERS

Photo content

Analysis utilizes two well-known techniques from computer vision:

- Speeded-Up Robust Features (SURF)
- Sum of Squared Differences (SSD)

SURF facilitates matching regions in two images by locating “keypoints” such as corners, and then computing a feature vector called a descriptor to represent each keypoint. A “matching” between the descriptors in two images can be found by computing the distance between the vectors.

SSD is useful for approximating the visual similarity of two equal-sized blocks of image content. It is a simple metric that computes the difference between the value of a pixel in the first image and the corresponding pixel in the second image, and then sums the square of these differences over all pixels in the block.

TYPE-SPECIFIC ANALYZERS

Audio content

At a high level, YouProve’s audio analyzer extracts sequences of spectral peak frequencies from the source and derived audio clips and applies local sequence alignment to find matching time segments.

The use of spectral peak analysis to compare audio data was inspired by the Shazam audio recognition system. The technique is well-suited for analysis because spectral peaks are a central feature in human hearing and thus are independent of audio encoding.

They are largely maintained across transformations that preserve the way audio will be perceived (e.g., compression). To identify time segments in a modified clip which preserve sequences of spectral peak frequencies from a source clip, YouProve uses a modified version of Smith-Waterman algorithm for local sequence alignment.

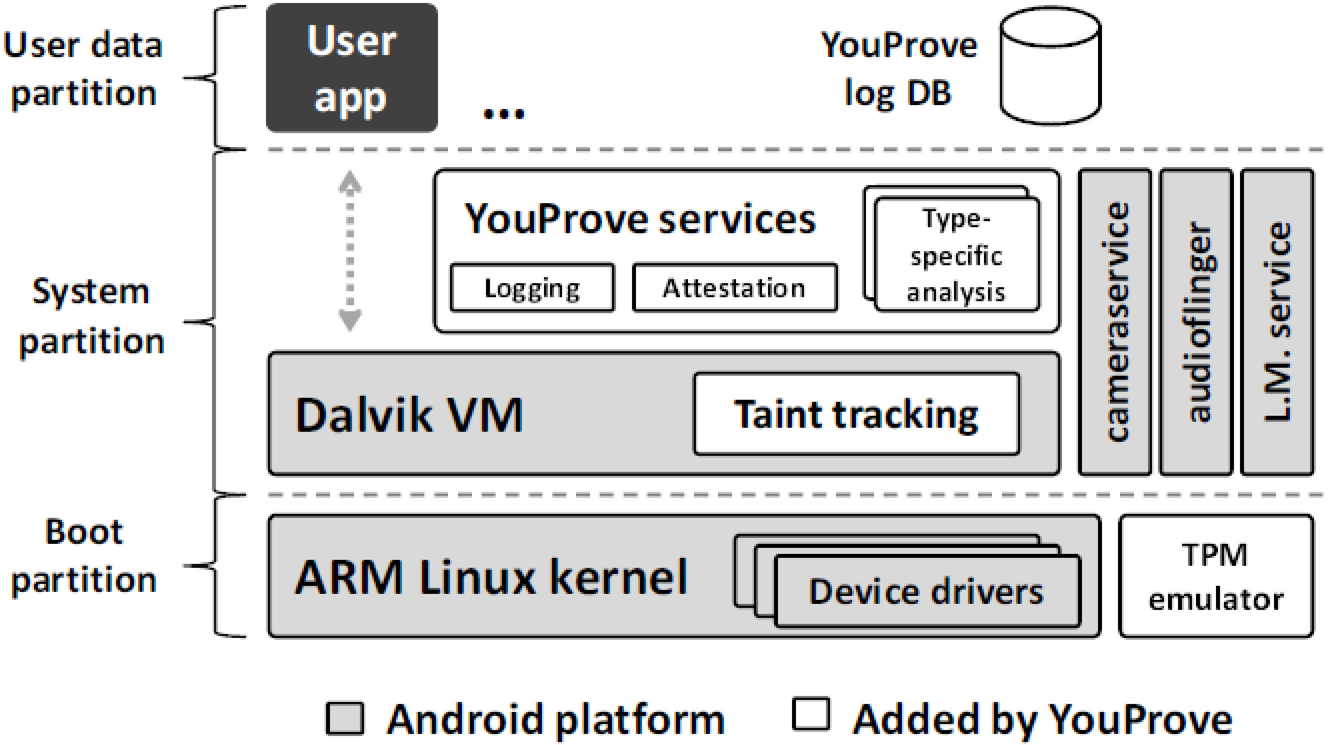
IMPLEMENTATION

- Nexus One smartphone on Android 2.2
- YouProve’s logging and attestation services are implemented in approximately 2,000 lines of Java code
- Nexus One does not include a TPM chip. Instead, was ported a popular open-source TPM emulator to Android,
- along with the TrouSerS open-source TCG stack

Prototype photo and audio analyzers are written in C++ and C, respectively

- The audio analyzer uses the open-source LibXtract and FFTW libraries for converting audio samples into frequency bins. We ported these libraries to the Android platform
- The photo analyzer uses the SURF implementation from the popular open-source computer-vision library OpenCV
- In addition, was used the open-source Taint-Droid information-flow tracking framework, which we updated from Android 2.1 to 2.2

YOUPROVE PROTOTYPE ARCHITECTURE



TESTING

The basis for our test dataset was a diverse collection of 69 photos taken on a college campus using a Nexus One. Subject matter included individual students, crowds, buildings, offices, landscapes, walls with flyers, and bookshelves full of books. The photos varied in level of detail, level of focus, and quality of lighting. All photos were taken at the default resolution of 1944x2592 pixels. We then used the ImageMagick image-editing tool to apply two classes of modifications to our photos.

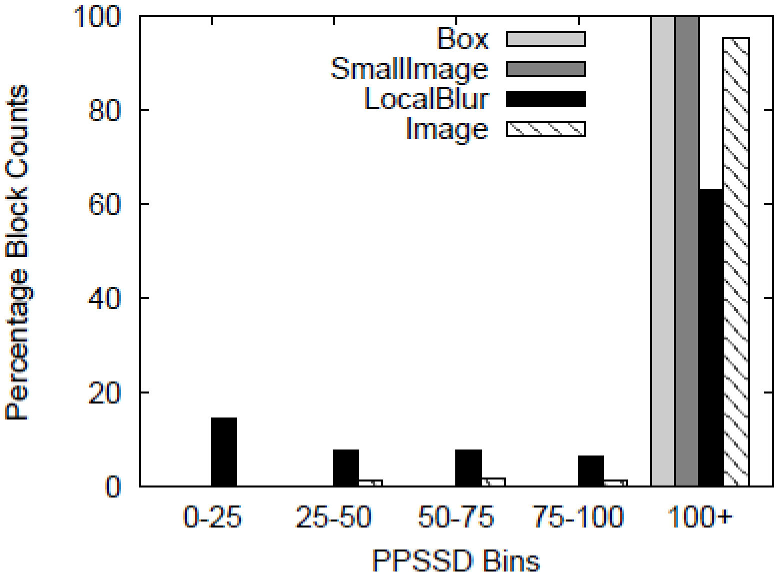
Global modifications included cropping, scaling, and JPEG compression. We consider these modifications to preserve the meaning of the source. Cropping test cases were created by cropping out either the top, bottom, left, or right half of an image, leaving a rectangular half-image. Scaling maintained aspect ratio while reducing each dimension to either 75%, 50%, or 25% of its original size. Compression produced JPEGs at 75%, 50%, and 25% quality.

Local modifications included overlaying a black box, pasting in a small photo, pasting in a large photo, or blurring a region. We consider these modifications to alter the meaning of the source.

EVALUATION

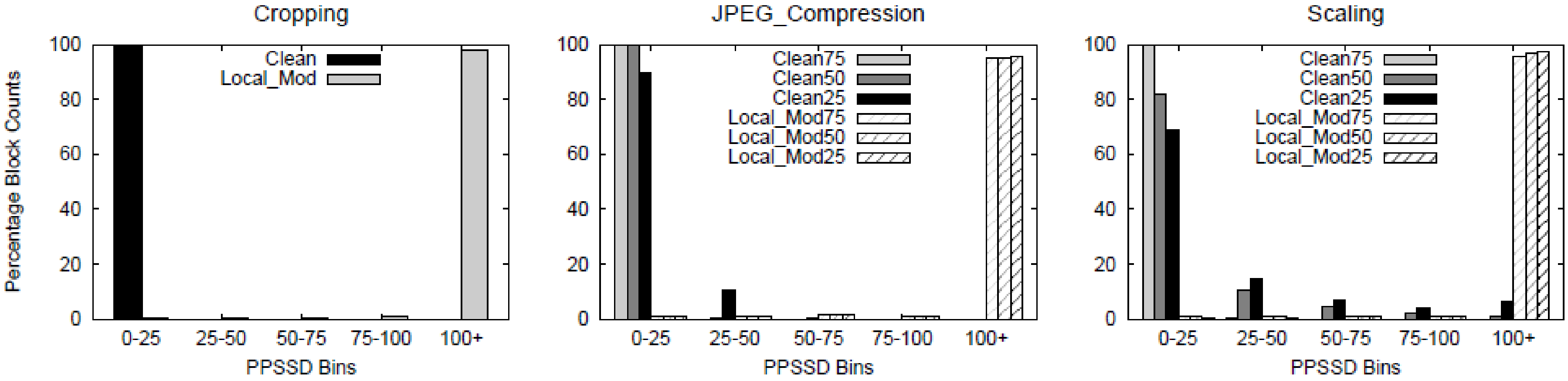
Analyzer Accuracy

Photo analysis



Block PPSSD for different local modifications.

PHOTO ANALYSIS



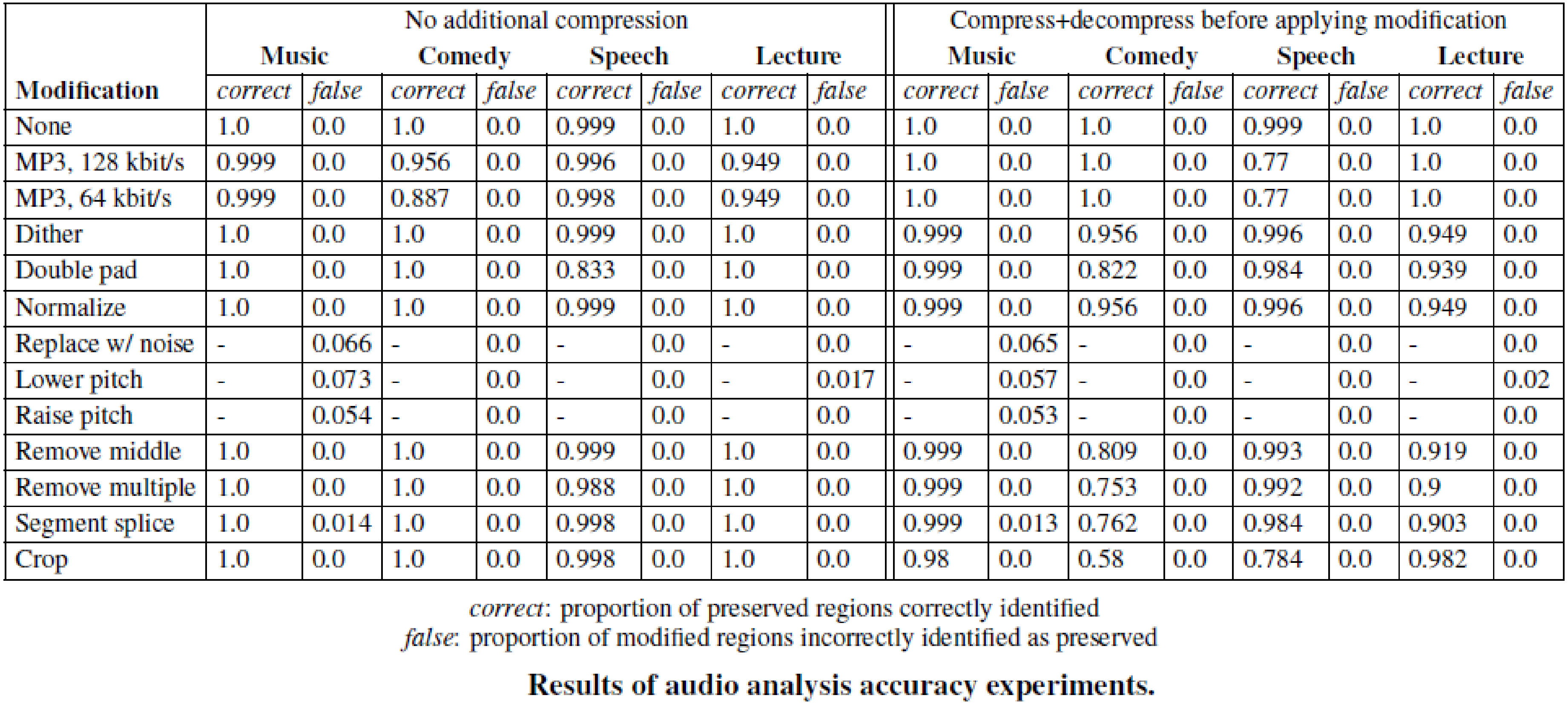
Results of photo analysis accuracy experiments.

AUDIO ANALYSIS

The test cases used to evaluate audio analysis were derived from 5-minute clips of an excerpt from Vivaldi’s The Four Seasons, a stand-up comedy show, an excerpt from the iconic “I Have A Dream” speech, and an undergraduate lecture.

Starting with these four clips, we used the Sound eXchange (SoX) tool to generate derivations from the following transformations: extracting and removing subclips, splicing in other audio, inserting silences, applying lossy (MP3) compression, dithering, normalizing, and pitch altering.

AUDIO ANALYSIS

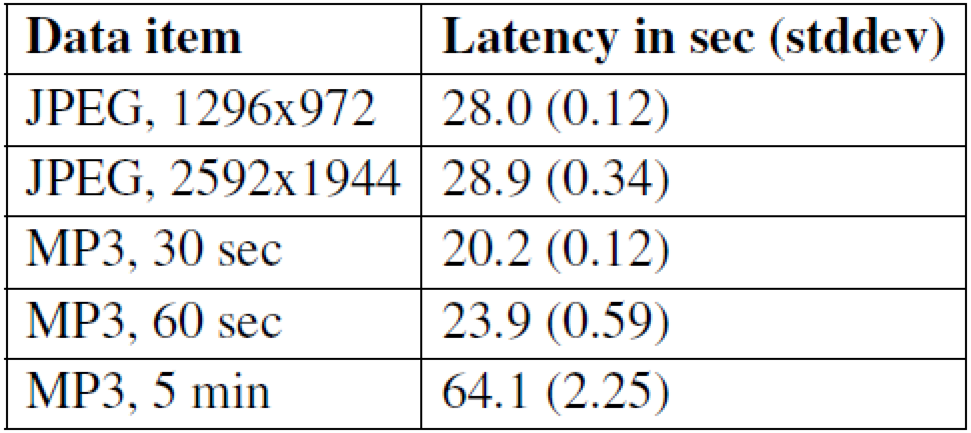


	No additional compression								Compress+decompress before applying modification							
	Music		Comedy		Speech		Lecture		Music		Comedy		Speech		Lecture	
Modification	*correct*	*false*	*correct*	*false*	*correct*	*false*	*correct*	*false*	*correct*	*false*	*correct*	*false*	*correct*	*false*	*correct*	*false*
None	1.0	0.0	1.0	0.0	0.999	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.999	0.0	1.0	0.0
MP3, 128 kbit/s	0.999	0.0	0.956	0.0	0.996	0.0	0.949	0.0	1.0	0.0	1.0	0.0	0.77	0.0	1.0	0.0
MP3, 64 kbit/s	0.999	0.0	0.887	0.0	0.998	0.0	0.949	0.0	1.0	0.0	1.0	0.0	0.77	0.0	1.0	0.0
Dither	1.0	0.0	1.0	0.0	0.999	0.0	1.0	0.0	0.999	0.0	0.956	0.0	0.996	0.0	0.949	0.0
Double pad	1.0	0.0	1.0	0.0	0.833	0.0	1.0	0.0	0.999	0.0	0.822	0.0	0.984	0.0	0.939	0.0
Normalize	1.0	0.0	1.0	0.0	0.999	0.0	1.0	0.0	0.999	0.0	0.956	0.0	0.996	0.0	0.949	0.0
Replace w/ noise	-	0.066	-	0.0	-	0.0	-	0.0	-	0.065	-	0.0	-	0.0	-	0.0
Lower pitch	-	0.073	-	0.0	-	0.0	-	0.017	-	0.057	-	0.0	-	0.0	-	0.02
Raise pitch	-	0.054	-	0.0	-	0.0	-	0.0	-	0.053	-	0.0	-	0.0	-	0.0
Remove middle	1.0	0.0	1.0	0.0	0.999	0.0	1.0	0.0	0.999	0.0	0.809	0.0	0.993	0.0	0.919	0.0
Remove multiple	1.0	0.0	1.0	0.0	0.988	0.0	1.0	0.0	0.999	0.0	0.753	0.0	0.992	0.0	0.9	0.0
Segment splice	1.0	0.014	1.0	0.0	0.998	0.0	1.0	0.0	0.999	0.013	0.762	0.0	0.984	0.0	0.903	0.0
Crop	1.0	0.0	1.0	0.0	0.998	0.0	1.0	0.0	0.98	0.0	0.58	0.0	0.784	0.0	0.982	0.0

correct: proportion of preserved regions correctly identified
false: proportion of modified regions incorrectly identified as preserved

Results of audio analysis accuracy experiments.

PERFORMANCE EVALUATION



Data item	Latency in sec (stddev)
JPEG, 1296x972	28.0 (0.12)
JPEG, 2592x1944	28.9 (0.34)
MP3, 30 sec	20.2 (0.12)
MP3, 60 sec	23.9 (0.59)
MP3, 5 min	64.1 (2.25)

Latency of generating fidelity certificates.

CONCLUSION

This paper has presented the design and implementation of YouProve, a system that enables mobile sensing services to verify that contributed data has not been manipulated in a way that alters its original meaning while allowing clients to use untrusted editing applications to directly control the fidelity of data they upload.

Verifying that contributed data preserves the meaning of original sensor readings is a key requirement for ensuring data authenticity in domains such as citizen journalism.

The key to YouProve's approach is providing analytic bases of trust for remote services. YouProve relies on trusted, content type-specific analyzers running on the mobile device to generate reports summarizing differences between a derived data item and an original sensor reading.

Results of experiments with a prototype are promising. Logging source data does not noticeably affect application responsiveness, and our content analyzers are accurate and complete their tasks in under 70 seconds for 5-minute audio clips and under 30 seconds for 5-megapixel photos.