City Research Online

City, University of London Institutional Repository

Citation: Arunkumar, S., Soyluoglu, B., Sensoy, M., Srivatsa, M. & Rajarajan, M. (2017). Location attestation and access control for mobile devices using GeoXACML. Journal of Network and Computer Applications, 80, pp. 181-188. doi: 10.1016/j.jnca.2016.11.028

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/17317/

Link to published version: https://doi.org/10.1016/j.jnca.2016.11.028

Copyright: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

Reuse: Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

City Research Online: http://openaccess.city.ac.uk/ publications@city.ac.uk

Location Attestation and access control for Mobile Devices using GeoXACML

Saritha Arunkumar[†], Berker Soyluoglu[‡], Murat Sensoy[‡], Mudhakar Srivatsa, Muttukrishnan Rajarajan[*]
IBM Hursley Labs, UK[†] University of Ozyegin, Turkey[‡] IBM Research, USA City University, London[*]
saritha.arun@uk.ibm.com, msrivats@us.ibm.com, R.Muttukrishnan@city.ac.uk

***Abstract*—Access control has been applied in various scenarios in the past for negotiating the best policy. Solutions with XACML for access control has been very well explored by research and have resulted in significant contributions to various sectors including healthcare. In controlling access to the sensitive data such as medical records, it is important to guarantee that the data is accessed by the right person for the right reason. Location of access requestor can be a good indication for his/her eligibility and reasons for accessing the data. To reason with geospatial information for access control, Geospatial XACML (eXtensible Access Control Markup Language) is proposed as a standard. However, there is no available implementation and architecture for reasoning with Geospatial XACML policies. This paper proposes to extend XACML with geohashing to implement geospatial policies. It also proposes an architecture for checking reliability of the geospatial information provided by clients. With a case study, we demonstrate how our framework can be used to control the privacy and data access of health service data in handheld devices.**

I. Introduction

Mobile devices have been used in the modern world to access information, exchange information and to store information. With new applications and new solutions coming to existence every day , handheld devices are being used more and more to completely take over the functionality of a wallet, laptop, computer, briefcase and books. With the increasing demand on the mobile devices usage, the number of threats and issues related to security with regards to the handheld devices are doubled. Information exchange is a key requirement for every sector and handheld devices are being used for the very same purpose more frequently than ever before. How can one ensure that the data being accessed is being accessed by the right user of the data. How can one ensure that the data requestor is the genuine requestor of the data and has the right to access the data being requested.

This is a very interesting problem which highlights the very existence of access control mechanisms. The process of making sure that data is accessed by the right source for the right requirement is called access control. Making use of access control in order to ensure that the right user is getting hold of the right data has always been a concern in all major areas of business including healthcare. Healthcare data involves a lot of personal data of the end users and hence it is critical to ensure that the data is not misused and is not mishandled. It is equally important to ensure that data is being accessed by the right user and does not fall into the hands of a malicious user who would mishandle the data.

Researchers in the past have introduced solutions around access control for healthcare with XACML. By making use of XACML policy, and through policy negotiation, relevant data is provided to the requestor based on the information provided in the XACML policy. The solution has taken care of the access control of the data but something very important has been forgotten in this solution. The data can be requested from anywhere and there is no way to stop an outsider from requesting the data. This introduces the importance of contextual information for the requestor of the data. This paper specifically considers the location information of the data requestor before granting access to the data. This has been achieved by making use of geospatial attributes of the handheld device from where the request is made. Once the geographical coordinates are verified, the policy is then applied on the request and then access to the data is eventually granted. This solution has been implemented by using GeoXACML policy. This is the first of a kind implementation of GeoXACML for access control in mobile healthcare.

The organization of the paper is as follows: Section I provides a brief introduction to access control and GeoXACML. Section II covers the related work. Section III covers the background about XACML, GeoXACML and the importance of global attestation. Section IV covers the system model for global attestation. Section V describes the Global attestation scheme. Section VI describes the implementation of GeoXACML. Section VII describes the implementation of the Geospatial access control for mobile healthcare in an iphone application. Section VIII shows the experiments and results of the global attestation scheme. Finally, Section IX concludes the paper.

Research was partly sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. Dr. Şensoy thanks the U.S. Army Research Laboratory for its support under grant W911NF-14-1-0199 and The Scientific and Technological Research Council of Turkey (TUBITAK) for its support under grant 113E238.

II. Related work

Jansen et al. [12], [13] describe an implementation of assigning and enforcing policies on handheld devices using Java smartcards. The organization policy is distributed via tamper-proof smartcards. All the devices are smartcard enabled. The policy to be enforced is read from the smartcard, which requires authentication by a username and pin. After authentication a card monitor continuously monitors the existence of the smartcard. If the smartcard is removed the device reverts to the default policy of the device. [13], [14] describe the use of a policy specification language, a policy distribution mechanism and certificate representation.

An XACML-based architecture is proposed [21] to tackle the problems of compromise to the requesters data confidentiality and integrity, and the issue of applicability of reputation data. The traditional XACML polices, used for user access control in distributed environments, can be used for mobile agents access control [10]. Such policies are used to manage delegation of access rights from users to agents while at the same time following the core principles of the XACML standard. [10] proposes a combination of policies that map users to their mobile agents and make access control decisions for mobile agents by evaluating complex policy sets. [4] deals with the use of P3P policy by extending it for data access control and use of XACML policy [5] in the mobile device for data access control.

GeoXACML [1] is a geo-specific extension to XACML 2.0 and it is standardised by Open GeoSpatial Consortium (OGC). GeoXACML supports the declaration and enforcement of geo-specific access rights. It also makes sure that it controls access to services, data and other information in a service oriented type architecture. GeoXACML [2] has been standardised but there has not been any open standard implementation of this to control the access rights to resources. We have implemented the required parts of GeoXACML to fit our needs and implemented a proof of concept version for some of the functions and attributes.

Saroiu et al. [19] have described location proofs as a new mechanism that enables the existence of mobile applications that needs proof of the user's location. It is handled by the wireless access point to mobile devices. The solution is mainly based on users and wireless access points (APs) exchanging their signed public keys to create time-stamped location proofs.The paper describes an implementation of the location proofs. The paper shows six potential applications that would be used by an infrastructure that provides location proofs. It also showcases a protocol that is demonstrable over WiFi and characterizes security properties of the design. The paper details the difficulties that come from a collusion attacks such as when sharing devices with one another. VeriPlace [16] is a location proof architecture that takes care of the challenges involving user trying to fool the systems by receiving location proofs for locations where they are not located. These solutions take care of the user's privacy and is capable of detecting cheating. VeriPlace used cryptographic techniques in order to achieve system security. VeriPlace needs three types of trusted entities that are run by different parties to avoid collusion. To protect users privacy, each trusted entity is aware if either a users identity or the location, but not both of them. VeriPlace uses Cheating Detection Authority (CDA) to check if any cheating has occured. Using the encrypted access point information, the CDA decrypts it and checks whether any two APs are far from each other, if yes it indicates cheat.

Zhu et al [22] proposes a Privacy-Preserving Location proof Updating System (APPLAUS) in which co-located mobile devices which are bluetooth enabled generate location proofs and then transfers the changes to a location proof server. In order to protect source location privacy pseudonyms which are changed periodically are used. The solution also shows a model in which the users can assess their location privacy levels and when and whether to receive the location proof requests. The paper also shows a way to secure from colluding attacks by presenting betweenness ranking based and correlation clustering based approaches for outlier detection. Implementation and deployment of APPLAUS is easily done in bluetooth enabled devices and doesn't require a lot of computation cost.

Spatial-Temporal provenance Assurance with Mutual Proofs (STAMP) scheme [3] is a solution that gives security and privacy assurance to mobile users proofs for their past location visits. STAMP is based on mobile devices in vicinity to mutually generate location proofs. Some of the key features of STAMP have been to maintain the integrity and non-transferability of the location proofs and location privacy of the users.The implementation also shows that it requires very low computational costs to execute this on the mobile devices.

Hasan et al.[11] analyzed the secure location provenance problem and introduced methods for making location proofs resistant against collusion. The proofs let the user prove the location in different granularity to a number of auditors. The paper shows two schemes using hash chains and bloom filters. Some of the experimentation results from the proof of concept shows that these schemes are realistic in today's mobile environments.

Another interesting paper [6] shows that they have designed two privacy-preserving alibi schemes: one for corroborators who have no personal privacy issues and another one for corroborators who want to keep control over the disclosure of their identities. The paper also demonstrates that schemes are implementable and usable in today's mobile devices.

The main contribution of our work in attestation compared to the state of the art is around the Global consistency check. Our work details how attestation can be achieved at a global level by making use of trust models. EigenTrust and PeerTrust models are used to show the results and benefits around global attestation. Global trust models are a crucial part of the consistency check in our solution. This technique proves that user's true location can be confirmed using global attestation scheme much more accurately in comparison to the local attestation schemes.

III. Background

XACML (eXtensible Access Control Markup Language), was formed by the OASIS (Organization for the Advancement

of Structured Information Standards) standards consortium. XACML is a simple, flexible way to express and enforce access control policies in a variety of environments, using a single language. The XACML language in effect protects content from unauthorized use in enterprise data exchanges. XACML is mainly derived around and written in, XML, which is understood in most global environments. OASIS, which drives the development, convergence, and adoption of e-business standards, has ratified XACML. XACML gives an extensive and powerful set of features to the developers. It allows an organization to create and deploy authorization policies to match its mix of assets and business use-cases, then plug in additional policies as the business and its standards evolve. XACML helps in resolving issues related to security applications and there have been a number of papers published in order to prove the same. Xuebing et al. [21] detailed in their paper how XACML can be used to solve some of the issues with mobile environment. Arunkumar and Rajarajan introduced XACML in the mobile environment and hence proposed a new architecture for data access control [5].

It is interesting to note that the implementation of GeoX-ACML for access control requires 2 main pieces to function correctly. The first piece is to ensure that the right user is getting the right access to the right resource. In order to achieve this, global attestation of location is a must. This paper will detail the global attestation scheme and how the user's location information can be verified by other user's in the vicinity.

The second important piece for geospatial access control to work is to ensure that the location information being passed to the server is passed using geohash. Geohash is a latitude/longitude geocode system invented by Gustavo Niemeyer and has been made public. It is a hierarchical spatial data structure which subdivides space into buckets of grid shape. Using geohash, the geographical co-ordinates can be sent to the server without having to send them as is.

IV. System Model

There are increasing number of mobile applications being used by end users to access all kinds of services. Location based services are mainly used for accessing location related capabilities. To ensure that the right user is receiving the location based services, the requirement of location proofs have been commonly seen and understood. So by making use of the Access Points, the location proofs are provided by the APs in the form of a proof to ensure that the mobile device requesting a service is at a particular location at a particular point in time. This has been very well implemented and researched by various researchers including Saroiu et al. [5], Zhu et al [22] and through VeriPlace in [4].

When the user and the AP are colluding, it clearly indicates that the user is in the vicinity of the AP and hence it is convenient for the AP to attest for the user's location. Another example for location attestation could be where a user is in the vicinity of 3 APs nearby. Individually each of the APs can attest the user's location and the attestation could be verified. However, what are the consequences when a global check is done for capturing inconsistency. If all the 3 APs are put together and if a consistency check is done, it could either result in a positive feedback where all the points add up or it could also result in a negative feedback where all the points don't add up. This proves that one of the locations provided by the user is a false location. By doing a global check through a mechanism named global attestation, it is possible to check whether the user has been lieing all through the path.

This brings us to a strong point that even if local attestation passes the check, does it mean that its correct. If it is correct, when multiple local attestations are done, will all the points add up? Our system model using global attestation scheme proves that a global consistency check is very crucial to prove the location proofs add up and that the user is not faking the location information through the entire journey of the request. Bitcoin has been using Block chain as the transaction database shared by all the nodes in a system [18] . This has been used as part of our system model for the global consistency check. A global log of the contacts is maintained in the database similar to block chain which is used to ensure that the locations reported by the entities themselves and other contacts in the proximity add up to result in a positive/negative feedback.

V. Global Attestation scheme

In our model, each device (e.g., users and access points) provides reports about their locations. These devices register to our system with their Bluetooth or WiFi MAC addresses and each is given a unique ID. A location report from an entity x does not only contains its location, but also the MAC addresses sensed in the proximity. Therefore, if two devices are close in location, they may sense and report each others' MAC addresses.

If all of the devices honestly report their locations and others they sensed, we may easily confirm the locations of these devices by cross checking the reports from different devices. However, these devices may not be honest, and even they may collude to mislead the system. Therefore, we do not assume the reliability of devices and compute their trustworthiness while reporting their and others' locations.

There are various statistical trust models. On the other hand, most of them requires some sort of ground truth to come up with positive and negative evidence (or feedback) for the behavior of entities. In these models, each report from an entity is evaluated with respect to ground truth. The report serves as a positive feedback for the trustworthiness of the entity if it complies with the ground truth. Similarly, it serves as a negative feedback if it does not comply with the ground truth. While the computation of positive and negative feedback is trivial when ground truth is available, in our setting, we do not have ground truth for the location of devices.

We propose to use report consistency instead of ground truth to derive positive and negative feedback for the computation of trust. Our system uses a global log of location reports in the system. This global log can be implemented similar to blockchain [8], which is a transaction database shared by all nodes participating in Bitcoin protocol [18].

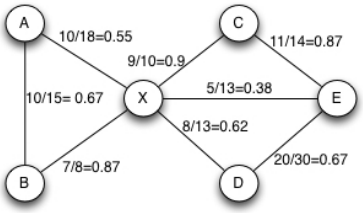

Fig. 1: Trust Feedback Graph.

Let us consider two devices i and j that provide a number of location reports over time. At time t, let us assume that they share their reports R_i^t and R_j^t, which include the locations of these devices as l_i^t and l_j^t, respectively. If l_i^t and l_j^t are in proximity, report of each device may confirm the existing of other by including its MAC address. If l_i^t and l_j^t are not in proximity, report of each device may not include the MAC address of other. In these cases, the reports are considered consistent; otherwise, they are not.

We use consistent reports from pairs of devices as positive feedback for their trustworthiness. Similarly, inconsistent reports serve as negative feedback. If devices i and j have n positive and m negative feedback, their overall positive feedback ratio is computed as $c_{i,j} = n/(n+m)$. Using positive and negative feedback, we compose a feedback graph where edges are weighted based on the computed overall positive feedback ratios. Figure 1 demonstrates a sample feedback graph.

Once a feedback graph is computed, we can use existing graph-based trust models to calculate trustworthiness of the nodes in the graph. While various trust models can be used, in our system, we use two specific graph-based trust models: EigenTrust [15], [9] and PeerTrust [20], [17]. These are models that compute global trust values for the nodes of a graph based on their local trust values, e.g., edge weights.

EigenTrust [15] provides an efficient and robust method for computing global trust values. The calculation of the trust values are similar to the ranking calculations of the well-known page rank algorithm. It generates a matrix C whose each each entry $C(i,j)$ corresponds to

$$\frac{c_{i,j}}{\sum_k c_{i,k}}.$$

Then, the principal eigenvector of the feedback matrix C gives the global trust values for the nodes.

PeerTrust [20] computes a node's trust value based on the number of feedback and the credibility of feedback. The credibility of feedback for pairs of nodes may be measured by the personalized similarity of these nodes. In order to compute the similarity between two nodes i and j, we first create their feature vectors $\mathbf{f}_i$ and $\mathbf{f}_j$, respectively. The k^{th} element of $\mathbf{f}_i$ is set as $c_{i,x}$ – the weight of the edge between i and k in the feedback graph. Then, $cos(\mathbf{f}_i, \mathbf{f}_j)$ – the cosine distance between $\mathbf{f}_i$ and $\mathbf{f}_j$ – is computed as

$$cos(\mathbf{f}_i, \mathbf{f}_j) = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{||\mathbf{f}_i|| \times ||\mathbf{f}_j||}$$

and taken as the personalized similarity of the nodes. Details of the PeerTrust algorithm can be found in [20].

After calculating global trust scores for the entities, our location attestation algorithm considers these trust scores as follow. When an entity reports its location, we find a set of positive reports (reports that concur with the location claimed by the entity) and negative reports (reports that claim that the entity was located elsewhere, or a lack of report from a trusted entity close to the claimed location). Then, we select the report from the most trusted entity (in this set) as the consensus report. If the consensus report agrees with the entity's claimed location then it is accepted; otherwise it is rejected. If the report from the most trusted entity conflicts with the reports from other entities with similarly high trust values, the consensus report could not be created. The lack of consensus report is considered same as the lack of report from a trusted entity close to the claimed location.

VI. Implementing Geospatial XACML

The main contribution of this paper is to restrict access to data from mobile devices based on geospatial attributes of the requester. XACML policy is applied to check the access rights before making a decision on an access request. However, it does not have operators and construct to handle geospatial attributes. There are several well-established open source XACML libraries, such as SUN's implementation[1]. In an XACML policy, several attributes of a requester are checked against their values to grant access. In this paper, we propose to extend the existing XACML implementations with the ability of handling geospatial attributes efficiently. For this purpose, we introduced new functions and attributes that support geospatial functionality.

We have two main *attribute* extensions: GeoPoint and GeoPolygon. A GeoPoint attribute represents a particular point which stores a latitude and a longitude. For instance, position of an access requester is represented as a GeoPoint instance. A GeoPolygon instance represent a region or geospatial boundaries, e.g., a building, or a room. This can be achieved by storing multiple GeoPoints, which when combined create a polygon. The polygon generation can be done with polygonization provided by existing topology libraries such as JTS[2]. In addition to these attribute extensions, we implemented a function called *geo-contains*,which takes a GeoPoint and a GeoPolygon as arguments and checks to see if the point is contained by the polygon.

With these extensions, the requester may send the current location together with the access request using XACML request to a Policy Decision Point (PDP). The PDP checks geospatial policies to decide if requester may be allow the requested access or not. Once the access is allowed, Policy Enforcement Point (PEP) provides the requested resource. A simple example policy and access request are listed in Figure 2 and Figure 3, respectively. In the example policy, it is stated that doctors can access medical records within the geospatial boundaries defined through a geoPolygon object. In the request example, a doctor with name John requests medical records

[1]http://sunxacml.sourceforge.net/

[2]http://www.vividsolutions.com/JTS

```
<Policy PolicyId="ExamplePolicy" ...>
   <Target><AttributeValue>/medical/reports/*</AttributeValue></Target>
   <Rule RuleId="ReadRule" Effect="Permit">
     <Target>
       <Subjects>  <AnySubject/> </Subjects> <Resources> <AnyResource/></Resources>
       <Actions> <Action>...
            <AttributeValue DataType="...XMLSchema#string">read</AttributeValue>
       ...</Action>  </Actions>
     </Target>
   <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
           <SubjectAttributeDesignator AttributeId="doctor" DataType="...#string" />
         </Apply>
         <AttributeValue DataType="...XMLSchema#string">John</AttributeValue>
       </Apply>
       <Apply FunctionId="geo-contains">
         <Apply FunctionId="geoPoint-one-and-only">
           <SubjectAttributeDesignator AttributeId="location" DataType="geoPoint" />
         </Apply>
         <AttributeValue DataType="geoPolygon">
                         41.027514,29.189435;
                         41.029514,29.189435;
                         ...
         </AttributeValue>
       </Apply>
   </Condition>
   </Rule>
</Policy>
```

Fig. 2: A GeoSpatial policy example.

```
<Request>
  <Subject>
    <Attribute AttributeId="doctor"
               DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>John</AttributeValue>
    </Attribute>
    <Attribute AttributeId="patient"
               DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>Alice</AttributeValue>
    </Attribute>
    <Attribute AttributeId="location"
               DataType="geoPoint">
      <AttributeValue>41.028514,29.190435</AttributeValue>
    </Attribute>
  </Subject>
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      <AttributeValue>http://example.com/medical/reports/alice</AttributeValue>
    </Attribute>
  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>read</AttributeValue>
    </Attribute>
  </Action>
</Request>
```

Fig. 3: A GeoSpatial request example.

for patient named Alice. The PDP may examine the request and provide grand based on the policy.

In the policy example above, the geospatial boundaries are unnamed. In other words, we explicitly define the geospatial boundaries instead of addressing this region. This makes it harder to author policies, since the policy author may not correctly enter the geopoints composing the polygon. Authoring policies become an important issue as the number of policies. To handle this, we propose to store geoPolygon objects in a separate database and use alias to refer to them. For instance, the geoPolygon object used in the example policy is stored in a database table with alias *CentralHospital*. Then, the alias can be used while defining the policy, instead of explicitly using the geoPolygon object. Therefore, the policies could be more human friendly and less error prone.

The *geo-contains* function is responsible for checking if the location of the requester is within a region defined by geoPolygon object. This may be a costly operation if we represent the polygon as a set of geopoints. However, if we use *GeoHashing* [7] to represent geoPolygon objects instead of multiple geopoints, we can very efficiently perform the containment test. Geohash is a geocode system invented by Gustavo Niemeyer. For each geopoint, geohashing produces a code. By gradually removing characters from the end of the code will reduce the precision and the code would be representing a region instead of a single point. Therefore, geohash codes belonging to nearby points may have similar

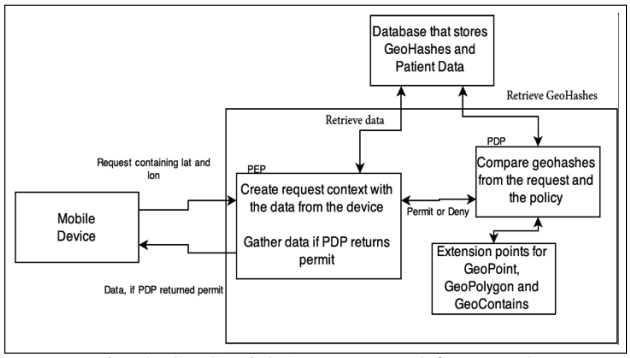

Fig. 4: GeoSpatial Access control framework

prefixes. That is, the longer a shared prefix is, the closer the two points are. In other words, we can efficiently check if a point falls into a region by checking the prefixes. As a result, we propose to store geohashing prefixes for geoPolygon objects and exploit these while checking if requesters' location falls into predefined regions.

Figure 4 shows the components of our framework. Policies and requests are composed based on GeoXACML standards using the introduced techniques. In this framework, geospatial attributes are handled in preprocessing step in which a GeoXACML policy is converted into a regular XACML policy (with equality/prefix/range constraints on geohash) and during policy verification step the input geospatial attributes of requesters are converted to their geohashes and the regular XACML policy can be applied. The main advantage of this solution is that we can use an existing XACML engine to implement a geoXACML engine very easily and perform geospatial policy reasoning efficiently.

VII. Geospatial Access Control in Healthcare

In this section, we will introduce a scenario from healthcare domain to motivate and demonstrate our framework for geospatial access control in mobile devices. Medical records contains important information for diagnosing and curing various health problems. Authorised doctors with expertise may access medical information about their patients while fulfilling their responsibilities such as diagnosing, monitoring, and curing patients. However, these doctors may not access these records for a different purpose, since they may contain sensitive data. There may be a relation between purpose of access and the location of access requester. For instance, a doctor may be in the hospital while diagnosing or examining an outpatient. Therefore, she may request medical reports of the patient from the hospital, e.g., her office. It is less likely that the doctor will use the requested information for the right purpose (i.e., diagnosis) if he/she is trying to access it outside of the hospital. Doctors may access their patients' records only when they are in the right location. In order to restrict the access based on attributes and location of the requester, we may use policies written in GeoXACML.

An example policy and related request are listed in Figure 2 and Figure 3, respectively.

In order to demonstrate how our solution is deployed for this and similar problems, we implemented a mobile iOS

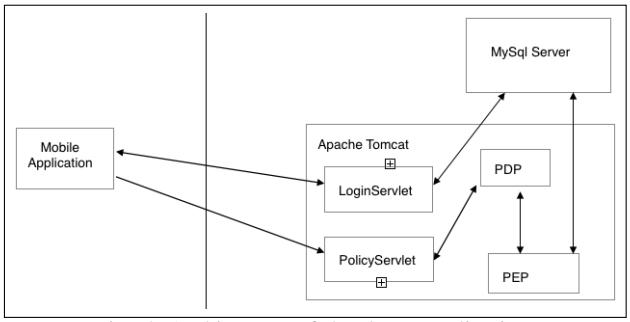

Fig. 5: Architecture of the demo application.

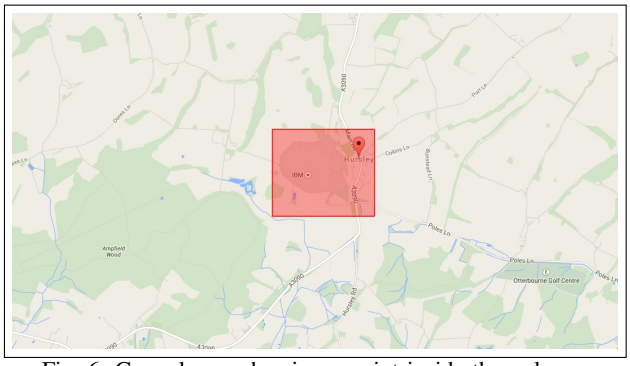

Fig. 6: Geopolygon showing a point inside the polygon

application. Architecture of this demo application is shown in Figure 5. In the PDP, we store policies such as "Doctors must be in the premises of their hospitals if they are required to access particular patients records". Then, we have extended SUN's XACML implementation as described in the previous section to implement GeoXACML policy decision and enforcement points. In order to store patient records, we have used a relational database at the back-end.

Figure 6 shows the polygon defined and a point within the polygon which refers to the location of the doctor making the request. Since the doctor's location is within the polygon, the doctor will be allowed to access the records. One of the snapshot from our demo application is shown in Figure 7. The figure shows the patient details that is visible to the doctor, once he has passed the policy check.

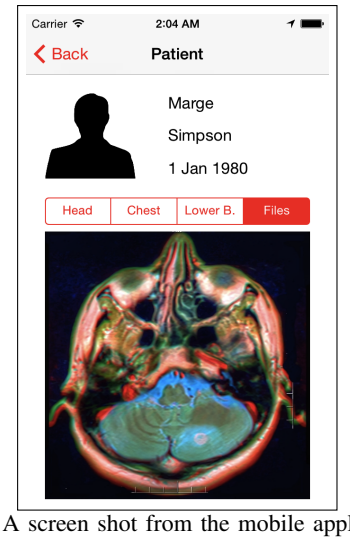

Fig. 7: A screen shot from the mobile application.

VIII. Experiments and Results

In this section we present an experimental evaluation of the proposed global attestation scheme approach using three publicly available datasets (see Figure 8). The `San Francisco taxicab` dataset includes GPS location traces from about 500 taxicabs over a period of 30 days. These cabs covered a spatial extent of about 600 km^2 around the city of San Francisco. Contacts between cabs were synthetically induced when two cabs happened to be within 600 meters of each other. When the spatial resolution of contacts was reduced below 200 meters, contacts were very infrequent and that over 1200 meters induced a large number of contacts; hence the choice of 600 meters for experimentation. The `MIT Reality Mining` dataset includes WiFi location traces from about 75 entities (typically personal laptop and handheld devices) over a period of 30 days. These entities covered a spatial extent of about 10 km^2. Contacts between two entities were naturally induced when two entities connected to the same WiFi access point. The `Infocom06` dataset includes Bluetooth contacts from about 78 entities (a subset of attendees of IEEE Infocom 2006 conference) over a period of 4 days. These entities covered a small indoor spatial extent of under 1 km^2. Contacts between two entities were naturally induced by pairwise Bluetooth contacts. We remark that these datasets were intentionally chosen with varying degrees of contact density (number of contacts per entity per unit time). The taxicab trace from San Francisco has the least contact density, while the indoor Bluetooth contact trace from Infocom06 has the largest contact density.

In our experiments we emulate both honest and dishonest entity behavior. A honest entity faithfully reports contacts with other entities; e.g., in the taxicab data a honest taxicab would report contacts with all the other taxicabs that are within the chosen threshold distance of 600 meters. A dishonest entity can report both false positives and false negatives: a false positive report is one wherein a dishonest entity reports contact, that which has not really occurred (e.g., a dishonest taxicab a claims that it is in contact with taxicab b when b is currently more than 600 meters from a); a false negative report is one where a dishonest entity fails to report a true contact (e.g., a dishonest taxicab a fails to report contact with taxicab b when b is within the threshold 600 meters from a).

In our experiments we also emulate both collusive and non-collusive settings for the dishonest entities. In a non-collusive setting a dishonest entity randomly chooses to induce a false positive or a false negative report. In a collusive setting a dishonest entity is more strategic: a dishonest entity would also concur with its colluder, i.e., if a dishonest taxicab a reports a contact with its colluder b, then b would also report a contact with a. Further, the choice of the location in the contact report between two colluding entities could be arbitrarily chosen (including a location that which neither a and b are currently located). In this setting dishonest entity would continue to randomly induce false positive and false negative reports against non-colluders (i.e., honest entities).

When two entities a and b concur (e.g., a reports a contact with b at location l at time t and b reports a contact with

a at location l at time t) then the trust management system would treat this as a positive feedback between a and b. When two entities a and b fail to concur the trust management system would treat this as a negative feedback. Indeed based on one instance of non-concurrence it is impossible to say whether a is dishonest or b is dishonest or both are dishonest (both being dishonest is feasible only under the non-collusive setting). We combine multiple positive and negative feedbacks to determine a trust score (between 0 and 1) in an entity using the EigenTrust and the PeerTrust algorithms.

Figures 9 and 10 show the error in trust score under non-collusive and collusive settings (respectively). Given the trust score estimate for an entity a ($\hat{ts}_a$) and the ground truth trust score (ts_a = 0 for dishonest entity and 1 for honest entity), the error in trust estimate is computed as the root mean square error in the estimate. We also compare the approaches with random, that assumes uniform trust in all the entities. We observe that in a non-collusive setting, then both the EigenTrust and the PeerTrust approach is very effective in estimating trust even when the fraction of dishonest entities is large (and hence do not offer corroborating evidence). Also, the localized approach to trust estimation helps PeerTrust outperform the EigenTrust solution that attempts to compute a global trust score for each entity. However, in a collusive setting both the EigenTrust is generally effective when we the fraction of dishonest entities is small than 0.5. The PeerTrust approach (again due to its localized trust estimation strategy) is relatively more robust. Nonetheless both the approaches are ineffective (compared to the baseline random strategy) when an overwhelmingly large fraction of entities are dishonest.

Figures 11 and 12 show the error in location attestation under non-collusive and collusive settings (respectively). Location attestation considers the trust scores of entities that report the target entity's location as described in the previous section. That is, we select the report from the most trusted entity (in this set) as the consensus report. If the consensus report agrees with the entity's claimed location then it is accepted; else rejected. We also check for absence of reports from highly trusted nodes that are in the vicinity (e.g., within 600 meters of the location claimed by a taxicab in San Francisco dataset). Location attestation error is captured as the root mean square error in accepting / rejecting a location claim, with accepts assigned a numerical value of one and rejects assigned a numerical value of zero.

We observe that under a non-collusive setting both the EigenTrust and the PeerTrust solution are effective even when a large fraction of entities are dishonest. Since location estimation relies on the entity with the highest trust score, under a collusive setting we observe that when an overwhelming fraction of entities are malicious, neither of the solutions are more effective than the baseline random strategy. However, for most practical settings wherein the fraction of dishonest entities is under 0.5, the EigenTrust and the PeerTrust solution offer a viable solution for robust location attestation.

A summary of key observations from our experiments are as follows:

- In general, the trust estimation error and location attestation error are acceptably small ($< 15\%$) when we have fewer than one-third dishonest entities.
- In general as the contact density increases the error in trust estimation and location attestation decreases. Recall that the San Francisco dataset has the lowest contact density, while the Infocom06 dataset has the highest contact density.
- The PeerTrust approach of computing the trust score is more robust when we have a large fraction of dishonest nodes. The PeerTrust approach can handle more dishonest nodes because it uses personalized trust scores for each entity, rather than a global trust score (as in EigenTrust).

	San Francisco	**MIT Reality**	**Infocom06**
Num Entities	512	75	78
Location Source	GPS	WiFi	Bluetooth
Sampling Interval (secs)	30	300	120
Spatial Extent (km^2)	600	10	1
Temporal Extent (days)	30	30	4
Num contacts	28,241	18,665	182,951

Fig. 8: Datasets

We have run 100000 tests and implemented 3 additional test, which are the objects used in GeoXACML extension. In the run with GeoXACML extensions we have created 3 Attribute objects and 1 FunctionBase object so we have 268 nanoseconds just object creation overhead. The rest is execution time for checking if a point is located within bounds of a polygon in which we have a database access. Figure 13 shows the results highlighted from the tests:

Some of the potential limitations of the global attestation scheme could be if all the entities are dishonest users. In such a scenario, the scheme will not work as per the requirement. For the global attestation scheme to work, there is also a requirement of trusted contact entities. If the trusted entities turn to be malicious users, then other methods of ensuring that the same entities are not used repetitively to get the consensus report should be considered.

IX. Summary

With increasing volumes of data tagged with space and time (e.g., from smartphones) it is becoming increasingly important to support contextual (e.g., location-based) access control to data and resources. In this paper we have explored solutions to realize the GeoXACML access control model that allows a security administrator to specify location-based access control policies. This paper presents the first implementation and architecture for GeoXACML. The key novelty in our approach is the ability to use geohashes to translate a GeoXACML policy into a conventional XACML policy - this allows us to fully reuse existing implementations of the XACML engine. The paper also describes a case study in the context of healthcare services wherein access control to handheld devices is moderated based on the location of the device. As a part of our future work we will explore solutions for location attestation (to authenticate the location of a device) and scalable enforcement of GeoXACML policies (as the volume of machine-to-machine traffic increases).

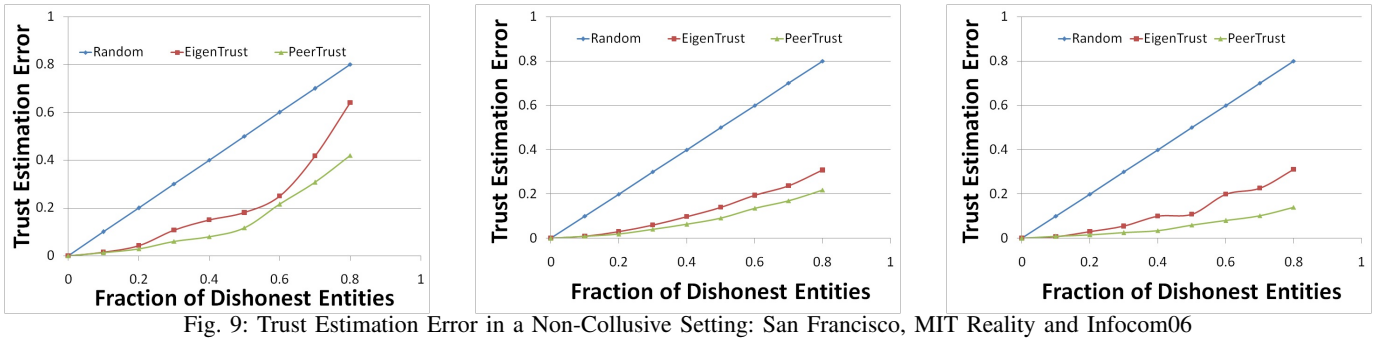

Fig. 9: Trust Estimation Error in a Non-Collusive Setting: San Francisco, MIT Reality and Infocom06

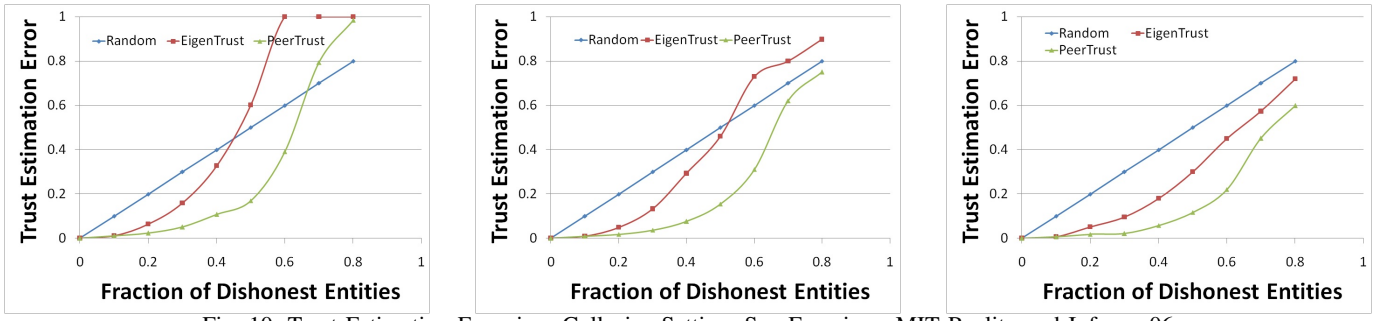

Fig. 10: Trust Estimation Error in a Collusive Setting: San Francisco, MIT Reality and Infocom06

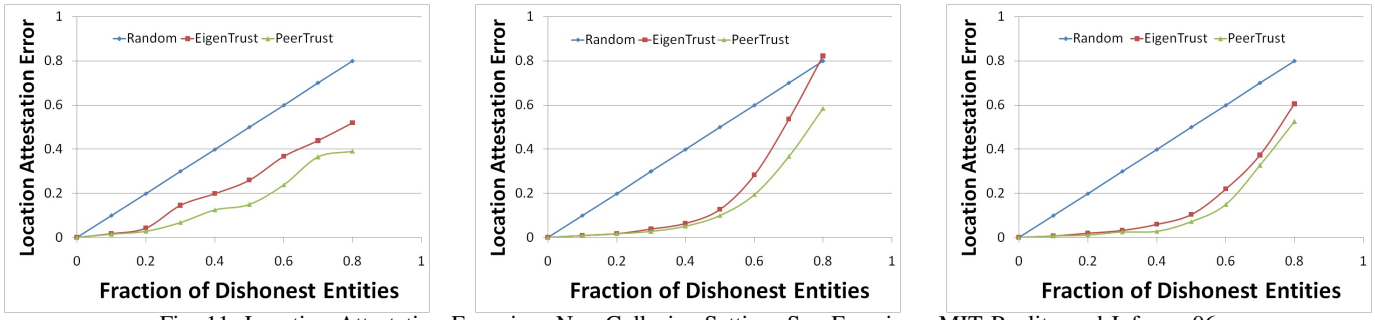

Fig. 11: Location Attestation Error in a Non-Collusive Setting: San Francisco, MIT Reality and Infocom06

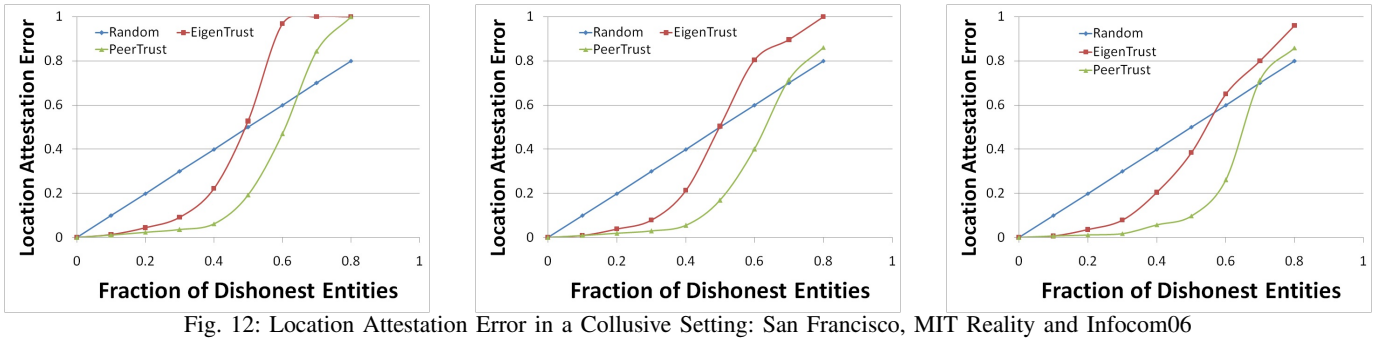

Fig. 12: Location Attestation Error in a Collusive Setting: San Francisco, MIT Reality and Infocom06

Run Count	GEO Extensions	Without GEO Extensions	Creation of FunctionBase Object	Creation of Attribute Object	Creation of Finder Object
100000	6608 ns	3109 ns	76 ns	140 ns	70 ns

Fig. 13: Results showing the overhead with and without GeoXACML.

REFERENCES

[1] http://portal.opengeospatial.org/files/?artifact_id=42734.

[2] http://www.opengeospatial.org/projects/groups/geoxacmlswg.

[3] T. Abdelzaher, X. Wang, and R. Ganti. Stamp: Ad hoc spatial-temporal provenance assurance for mobile users. In *Proceedings of The 21st IEEE International Conference on Network Protocols, ICNP 13, Gottingen, Germany, October 2013*, 2013.

[4] S. Arunkumar, A. Raghavendra, and M. Rajarajan. Policy extension for data access control. In *6th IEEE workshop on Secure Network Protocols (NPSec)*, pages pp55–60, 2010.

[5] S. Arunkumar and M. Rajarajan. Healthcare data access control using xacml for handheld devices. In *Developments in E-systems Engineering (DESE), 2010. IEEE, 2010,*, pages pp. 35–38, 2010.

[6] H. C. B. Davis and M. Franklin. Privacy preserving alibi systems. In *ACM ASIACCS*, 2012.

[7] Z. Balkić, D. Šoštarić, and G. Horvat. Geohash and uuid identifier for multi-agent systems. In *Agent and Multi-Agent Systems. Technologies and Applications*, pages 290–298. Springer, 2012.

[8] J. DuPont and A. C. Squicciarini. Toward de-

anonymizing bitcoin by mapping users location. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, pages 139–141. ACM, 2015.

[9] F. Z. Filali and B. Yagoubi. Global trust: A trust model for cloud service selection. *Computing*, 3(18):19, 2015.

[10] A. Giambruno, M. A. Shibli, S. Muftic, and A. Lioy. Magicnet: Xacml authorization policies for mobile agents. In *Internet Technology and Secured Transactions, 2009*, pages pp 1– 7. International Conference for Publication, 2009.

[11] R. Hasan and R. Burns. Where have you been? secure location provenance for mobile devices. In *CoRR*, 2011.

[12] W. Jansen, T. Karygiannis, S. Gavrila, and V. Korolev. Assigning and enforcing security policies on handheld devices. In *In Proceedings of the Canadian Information Technology Security Symposium*, 2002.

[13] W. Jansen, T. Karygiannis, S. Gavrila, and V. Korolev. Security policy management for handheld devices. In *In The 2003 International Conference on Security and Management(SAM'03)*, 2003.

[14] W. Jansen, T. Karygiannis, V. Korolev, and S. Gavrila. Policy expression and enforcement for handheld devices. Technical report, NIST, 2003.

[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM, 2003.

[16] W. Luo and U. Hengartner. Veriplace: a privacy-aware location proof architecture. In *ACM GIS*, 2010.

[17] S. Moalla and M. Rahmouni. Trust path: a distributed model of search paths of trust in a peer-to-peer system. *Security and Communication Networks*, 8(3):360–367, 2015.

[18] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.

[19] S. Saroiu and A. Wolman. Enabling new mobile applications with location proofs. In *ACM Hotmobile*, 2009.

[20] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *Knowledge and Data Engineering, IEEE Transactions on*, 16(7):843–857, 2004.

[21] Q. Xuebing and A. Carlisle. "xacml-based policy-driven access control for mobile environments ",. In *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on Digital Object Identifier:*, number 10.1109/CCECE.2006.277617, pages pp. 643 – 646, 2006.

[22] Z. Zhu and G. Cao. Towards privacy-preserving and colludingresistance in location proof updating system. In *IEEE Transactions on Mobile Computing*, 2011.