A System for Privacy-Preserving Access Accountability in Critical Environments

Francesco Buccafurri
University of Reggio Calabria (Italy)

Gianluca Lax
University of Reggio Calabria (Italy)

Serena Nicolazzo
University of Reggio Calabria (Italy)

Antonino Nocera
University of Reggio Calabria (Italy)

In this paper, we present a BeagleBone-based system to increase the security and safety level of critical environments by tracking people movements. Our solution is privacy aware, as it is possible to know who accessed a zone at a given time, with an uncertainty degree of accountability.

The problem of accountability of people's access to physical environments is receiving great attention in recent times, also because of the emergent risk related to terrorism. Specifically, indoor tracking to allow a-posteriori identification of the authors of an illegal action is an important issue.

Consider, for example, a security/safety incident occurred in a museum, which hosts thousands of visitors per day. Even though we have registered the entrance of all people and we are able to restrict the instant of the incident to a brief time interval, the number of suspects is so high that the adopted security measures are in fact ineffective. Besides video surveillance, an effective possibility of tracking people is based on the use of RFIDs, to have logs reporting people localization at any time and to restore precise information in a faster way than video surveillance. This kind of solution requires that people entrance in the surveillance environment is registered, but this is an acceptable (often already adopted) requirement in case of critical environments (e.g., government buildings, museums, safety-critical environments, tribunals).

Unfortunately, in most cases, a similar solution is intolerable for privacy reasons, often not compliant with law requirements.

In this paper, we study this issue by considering a surveillant environment in which people internal accesses must be logged for security purposes. However, a certain degree of uncertainty has to be introduced for privacy reasons. Specifically, it is required that, given an instant of time τ

and a monitored location, it should be possible to guess who was the person inside this location at the time τ with a probability at most k^{-1}, where k is a positive integer number representing a privacy requirement. [1]

OUR PROPOSAL

This paper proposes an RFID/BeagleBone-based system to generate logs allowing us to trace people with a suitable degree of uncertainty, in such a way that privacy is fully preserved. Logs fulfill a k-anonymity property, [1] so that we are able to guess who accessed a place, at a given time, with probability at most k^{-1}. Consequently, in case of a security/safety incident, we can restrict the pool of suspects from the whole population to k people.

Our approach is based on the use of movement sensors, RFID tags and readers, a server, and a suitable communication infrastructure. Preliminarily, the environment is partitioned into two types of zones: monitored and non-monitored zones. Moreover, each visitor is identified and associated with a passive RFID tag, which is attached to something that the individual must keep with him (e.g., to the visitor pass in a government building).

Each monitored zone is equipped with a movement sensor and a RFID reader to detect the presence of a tag (thus, of a person). Sensors are able to detect the physical movement of a person entering a zone in such a way that in case this movement is not associated with the reading of a RFID tag, an alert is triggered. When a person enters a monitored zone, the reader associated with this zone reads the identifier of the RFID tag (called EPC). Clearly, data received from readers allow the exact (i.e., with no uncertainty) tracking of people, provided that the association between tag EPC and person identity is known and this can result in a breach of privacy.

To avoid this, the received EPC is transformed into a new number in such a way that k people are associated with the same number (which we call qID – quasi identifier) and such a transformation is one way. All sensors are connected to a local network, so that the obtained qIDs can be sent to a server, which manages a log file.

System architecture and implementation

Our solution leverages two mechanisms: people identification and physical access monitoring. People identification is done by passive RFID tags that operate at a frequency of 865-868 MHz in Europe and 902-928 MHz in America, have a read range up to 10 meters and a cost of about 0.15$. Readers are built by exploiting a BeagleBone Black [2] equipped with a single-board computer equipped with the Texas Instrument's TRF7970ATB multi-protocol RFID transceiver and a high resolution cam Logitech C920 (see Figure 1).

As for physical access monitoring, typical solutions are based on turnstiles, which are too invasive. In our design, we adopt video counting, which is the most enhanced and powerful solution. A BeagleBone board is installed at the entrance of each monitored zone: each time the cam observes the movement of a person by running a suitable image-recognition software, the reader of a RFID tag is expected. If this does not occur, the system sounds the alarm to notify the entrance of a person with no RFID tag. Otherwise, the BeagleBone sends to a server a privacy-preserving information related to this entrance. To reduce privacy issues related to the people images, our solution does not store any image. Both the RFID people identification and the physical access monitoring modules are installed on the same BeagleBone Black board.

Trace Anonymization

The notion of *k-anonymity*, introduced above, means that the probability of identifying a person by accessing the stored location traces is k^{-1}, where k is a given privacy requirement. Our system reaches this goal by suitably transforming the received EPC into a new number, called qID (i.e., quasi identifier), belonging to a domain smaller than the number of people, so that more people are associated with the same qID. In particular, this domain is the interval $[1, d]$, where d is a positive integer system parameter.

Figure 1. The BeagleBone and the Logitech C920 webcam

The log file stores four types of events, which are described in the following.

1. *Entering the building*. When a person enters a government building, one RFID tag is associated with him and integrated into the document issued to permit the access to the building. Observe that, RFID tags used in our proposal have a life cycle. Indeed, in a real scenario, there must be a large number of RFID tags ready to be given to people. After a tag is associated with a person, it starts to work and we say it becomes *working* (active). Each BeagleBone stores the EPC of the currently *working* tags in a sorted set *S*. When a tag becomes working, a message containing the timestamp, the type of event (activation), and the assigned EPC is sent through the network. This way, each BeagleBone can add into *S* the new EPC.
2. *Leaving the building*. When a person leaves the building and returns the tag, this tag ends its life becoming *inactive*. Also in this case, a message containing the timestamp, the type of event (deactivation), and the tag EPC is sent through the network and all BeagleBones remove from *S* this EPC.
3. *Entering a zone*. When a person enters a controlled zone, the BeagleBone associated with this zone detects this entrance by the cam and reads the *EPC* of his tag, which is typically a 96-bit identifier. A privacy-preserving qID associated with this tag is computed from the EPC by means of three transformations.
   The first transformation maps the 96-bit domain of EPCs, which is highly sparse (because the number of tags used is much smaller than 2^{96}) into the dense domain $[1, |S|]$. Specifically, the received EPC is mapped to the integer p, which is its position in the sorted set S, with $1 \leq p \leq |S|$.
   The second transformation is done by the random permutation function *RPF* called *Algorithm 235*, [3] an efficient method for generating a particular permutation of the set *S*.
   Algorithm 235 is very simply:

```
for i from |S| downto 1 do
    exchange S[j] and S[i]
```

   where j = R mod i and R is a random integer picked from a PRNG whose seed is known only by the BeagleBones.
   Called S' the set permuted by Algorithm 235, the position p is transformed into the permuted position $p'=RPF(p)=S'[p]$. As a consequence, the access done by the p-th user is logged as done by the p'-th user. This swap hides the real identity of the user accessing a zone to all parties that do not know the seed used by the PRNG. Observe that $RPF^{-1}(p')$ can be done by searching for a p such that $RPF(p)=p'$ with $1 \leq I \leq |S|$, thus in linear time w.r.t. the number of users.
   In the third transformation, a hash function H having domain $[1,|S|]$ and codomain $[1,d]$ is adopted, where $d \leq t$. This function has the purpose of reducing the domain of the output (i.e., the stored qIDs) and generating the collisions among more qIDs necessary to implement the k-anonymity approach. Consequently, it is important that every hash value in the output range is generated with roughly the same probability. Among the several implementations of this function, we used the simplest hash function $qID = 1 + (p' \bmod d)$.
   When a user accesses a monitored zone, by these three transformations a new qID is generated for him and a message containing the timestamp, the type of event (entrance), the reference to the zone entered and this qID is sent through the network. Moreover, the BeagleBone involved in the reading adds into a local map T the pair (EPC, qID). This information is used in the next step.

4. *Leaving a monitored zone.* When a person exits a monitored zone, the involved BeagleBone searches for the read EPC into the map T to obtain the qID previously associated with this EPC. If such a query does not give any result, an alarm is generated to report the fault situation. Otherwise, a message containing the timestamp, the type of event (leaving), the reference to the zone and this qID is sent through the network. Moreover, the pair (EPC, qID) is deleted from T.

All the messages sent through the network are appended to the log file stored by a server. The log is encoded by XML format and does not include the information necessary to associate a person with a tag, which is expected to be stored elsewhere. Each log event is mapped into an element LogInfo. The attribute type specifies the typology of event, which can be activation or deactivation of a tag, or the reading of a tag when a person enters or leaves a zone. LogInfo subelements are: the timestamp of the event, the read EPC, the identifier of the involved zone and the generated qID.

An example of this file, concerning events occurred in the morning of 15 June 2018, is reported in Figure 2. The first two LogInfo elements show that two people accessed the building at about 9 and the tags with EPC 119 and 182 (again, for simplicity, we represent an EPC by 3 digits) were assigned to them. The next fragment of the log reports four entries into the zones Z3, Z8, Z4 and again Z3, occurred at about 10. Observe that the last information is the qID generated according to the procedure described at Step 3. Then, the log registered the leaving from zone Z3 of a person associated with the qID=1. Finally, the last LogInfo element shows that the person, to whom the tag with EPC=119 has been assigned, has left the building at 13:00.

```
<?xml version="1.0"
      encoding="UTF-8"?>
<root>
...
<LogInfo type=Activation>
      <Time>2018-06-15
            09:00:26</Time>
      <epc>119</epc>
</LogInfo>
<LogInfo type=Activation>
      <Time>2018-06-15
            09:00:29</Time>
      <epc>182</epc>
</LogInfo>
...
<LogInfo type=Entering>
      <Time>2018-06-15
            10:00:05</Time>
      <IdRoom>Z3</IdRoom>
      <id>2</id>
</LogInfo>
<LogInfo type=Entering>
      <Time>2018-06-15
            10:00:06</Time>
      <IdRoom>Z8</IdRoom>
      <id>3</id>
</LogInfo>
```

```
<LogInfo type=Entering>
      <Time>2018-06-15
            10:00:06</Time>
      <IdRoom>Z4</IdRoom>
      <id>2</id>
</LogInfo>
<LogInfo type=Entering>
      <Time>2018-06-15
            10:00:07</Time>
      <IdRoom>Z3</IdRoom>
      <id>1</id>
</LogInfo>
<LogInfo type=Leaving>
      <Time>2018-06-15
            11:00:00</Time>
      <IdRoom>Z3</IdRoom>
      <id>1</id>
</LogInfo>
...
<LogInfo type=Deactivation>
      <Time>2018-06-15
            13:00:00</Time>
      <epc>119</epc>
</LogInfo>
...
</root>
```

Figure 2. A partial representation of a log file.

We conclude this section by observing that there exist some attacks on privacy based on temporal correlation and knowledge of the building map. For instance, suppose that the reader of zone far from the building exit registers the entrance of a qID corresponding to Alice or Bob and that after few seconds, the system registers that Alice leaves the building. This allows an adversary to guess that Bob was in that zone. The solution we applied to prevent this attacks is based on temporal cloaking: [4] exact timestamps are perturbed and stored with an approximation time error e such that $e \cdot v \geq d$, where d is the distance between the building exit and the considered room and v is the user velocity (typically, v is assumed to be about 5 km/h). [4] Consequently, the timestamps considered so far have to be considered approximate timestamps.

Log Analysis

In case of need, it is possible to elaborate the log file to find the set of the users who could have entered a given zone at a given time T. We say "could" because this operation returns a set of k possible users, in order to comply with the privacy requirement. For example, consider the third LogInfo item of the log file fragment reported in Figure 2, we could be interested in knowing (with uncertainty) who is the person entered with qID=2 the zone Z3 at T =10:00:05.

```
Constant  d: system parameter
Input  n: the line index of the log file to analyze
Input  T: a timestamp
Input  S: the set of active tags at timestamp T
Output  U: the set of candidates
Variable  X: a set of integers
Variable  p: an integer
 1: for n times do
 2:     read next line L from log
 3:     if L.timestamp > T then
 4:         if L.type = activation then
 5:             S.add(L.EPC)
 6:         end if
 7:         if L.type = deactivation then
 8:             S.remove(L.EPC)
 9:         end if
10:     end if
11: end for
12: X = {x | (1 + x mod d = L.qID }
13: for each x ∈ X do
14:     p = RPF^-1(x)
15:     U.add(S[p])
16: end for
```

Figure 3. Algorithm EPSRestoring

In Figure 3, the algorithm solving this problem is presented. It receives three inputs. The first one is the index n of the log line to analyze. The other two inputs (i.e., T and S) are optional and have as default value T=0 and S={} (their use will be clarified later). The algorithm computes the set S of active tags at time T (Lines 1-11). This is done by elaborating log files from the beginning and adding to (Line 6) or removing from (Line 8) S an EPC each time an event of tag activation or deactivation is found. The optional inputs are used to speed up this computation: it is possible to create periodically a *check point*, i.e., to save the set S of the active tags at a given timestamp T. In this case, giving as optional inputs S and T, then the algorithm skips all log lines related to the events occurred before T (Line 3). Consequently, it is possible to start the construction of the set of active tags from this point instead of the beginning of the file. Then, the inversion of the 3rd (Line 12) and 2nd (Line 14) transformation return the position p of the considered EPC in S. Finally, this EPC is added to the output. Concerning the complexity of Algorithm 1, it is linear in the input parameter n (Lines 1-11) and linear in the number of users $|S|$ (Line 15), since we observed that RPF^{-1} is $O(|S|)$.

For example, with reference to the log reported in Figure 2, suppose we want to know (with uncertainty) who is the person referred by the third LogInfo item. The first operation to do is to find the numbers $x \in [1, 6]$ such that $1 + x \bmod d = 2$. As $d = 3$, we have two candidates, $x_1 = 4$ and $x_2 = 1$. Then, by inverting the random permutation function, we find that x_1 and x_2 derives from 1 and 2. Thus, the requested EPCs are the first and the second tags of S, which are 119 and 182. We gather that at 10:00:05 of 2018-06-15, the person associated with the EPC equal to 119 or 182 was in the zone Z3.

EXPERIMENTAL STUDY

In this section, we describe some experiments carried out to show how our system behaves and to test its performance. We observe that in the literature no real-world dataset of human traces exists to be used in our experiments: indeed, as stressed in this paper, such data are sensitive,

thus their storing, elaboration, and sharing result in an important privacy issue. Consequently, in this study, we used traces generated by a simulator.

We simulate the application of our approach to a real Italian government building, namely the Ariano Irpino law court, which consists of 2 floors and 56 rooms (zones). Concerning the setting of RFID readers, we assume that there is a reader at the entrance of each monitored room of the building and that the adjustable range of the RFID readers is set to detect tags which are at a maximum distance of 50 cm (about half the size of a door) from the reader. Each time a person crosses the entrance of a room, a reader reads the EPC from the RFID tag and triggers the execution of our technique for creating privacy preserving logs.

To simulate people movements inside the building, we built a java prototype implementing a mobility model based on the Random Waypoint Model, [5] which has been largely used in the literature. In this model, at the beginning of the simulation, each person randomly selects a destination room inside the building. Then, he moves towards it with a velocity selected uniformly at random in the interval (S_{min}, S_{max}). After the end-point is reached, he stops for a time period that varies in the interval (P_{min}, P_{max}). Then, he chooses another room and starts moving towards it. In our simulation, we set S_{min}=0.1 m/s, S_{max} = 2.1 m/s, P_{min} = 0.0 seconds and P_{max} = 1 hour.

Setting the system parameter

The first experiment aims at discussing about how to set the value of the parameter *d* in such a way to obtain the desired privacy requirement. Observe that the trivial setting *d=1* always solves the problem because all people are associated with the same qID but no meaningful information is provided. Thus, we should find the greatest d satisfying the above property. In Figure 4.(a), we measure the greatest value of *d* guaranteeing *k*-anonymity for different values of *k*, depending on the (changing) number of people.

We observe that all the curves have similar growing trend, because the number of collisions increases as the number of people increases, so that a higher value of *d* can be selected. The oscillations from the exactly-linear trend are due to the non-ideal behavior of the hash functions and the collisions of people with the same qID in the same room. The analysis of this figure allows us to fix the parameter *d* on the basis of the current number of users in the environment to guarantee a given privacy requirement. For example, to guarantee *k=6* with *200* people, we set *d=20*. Moreover, we can derive the upper bound for the anonymity degree as the fraction between the number of people and the parameter *d*. For example, setting as lower bound *k=6*, with *200* people and *d=20*, the upper bound is *k=10*, this means that we guarantee *k*-anonimity with *k* ranging from *6* to *10*.

Impact of d

Now, we want to measure the impact of the size of the domain of possible qIDs on the capability of our approach of guaranteeing the privacy requirements. For this purpose, we consider the same four values of the parameter *k* as done in the previous experiment, we set the number of people in the building to *200* and we measure the number of times, called Violation, in which the privacy requirement has not been satisfied. Observe that, this metrics takes also into account the situation in which two users with the same qID are in the same room. Indeed, this event does not generate uncertainty and reduces the chances to satisfy the privacy requirement. Figure 4.(b) shows the obtained results. We observe that a small variation of the parameter *d* may result in high performance variations. For instance, consider the curve *k=10*: we have that *d=8* is the value guaranteeing the privacy requirement when the number of people is 200. By means of this experiment, we find that by varying *d* from *8* to *22* the level of privacy given by the system quickly decreases and reach the lowest value, as people are uniquely identified in all cases.

Impact of the number of monitored zones

In this experiment, we study the effect of the percentage of monitored zones w.r.t. the total number of monitored zones in the building, assuming to have *200* people and *k=4*. We vary the percentage of monitored zones MZ from *25%* to *100%* and we measure the value of Violation, against different values of the parameter *d* (see Figure 4.(c)). We observe that the number of

monitored zones slightly influences the capability of our approach to guarantee the privacy requirement. Indeed, only when MZ assumes the lowest value (i.e., 25%) it is necessary to reduce the optimal value *d=35* (estimated in the previous experiments) to *d=28*. In all the other cases, the variation of the number of monitored zones has negligible impact. In conclusion, this experiment shows that in the setting of the system parameters it is sufficient to consider only the geometry of the building and not the number of monitored zones.

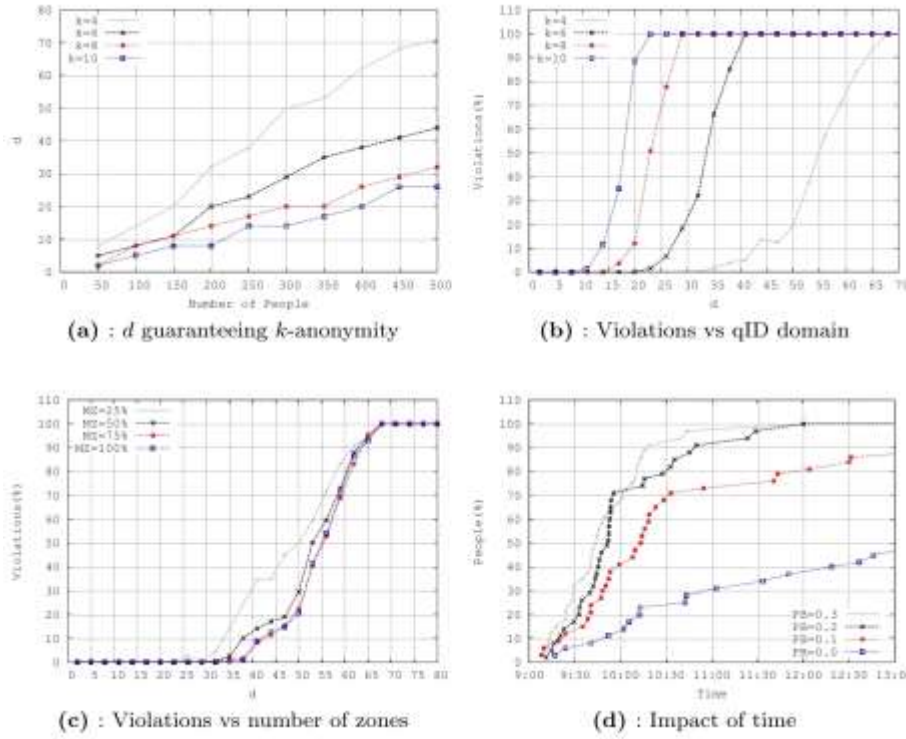

Figure 4. Experiment results.

Impact of time

So far, we have tested the capability of our system to guarantee the privacy requirements by analyzing a single log item. Now we provide a further study aiming at measuring the uncertainty in the identification of a person who entered a zone over time. As a matter of fact, at each access, *k* possible candidates are determined. Clearly, when more people enter a zone over time, the number of candidates grows. In this experiment, we focus on a given zone and measure the number of candidates (as fraction of the total number of people inside the building) who access it over time. Clearly, the result of this experiment is strongly influenced by the access frequency of this zone. Therefore, we consider four typologies of zone, each characterized by a different bias of access probability (PB, for short). In particular, when *PB=0* we mean that all building zones have the same probability of being accessed, whereas *PB>0* means that the probability of accessing this zone is increased by *PB* w.r.t. the uniform probability. The analysis is conducted by considering a time window from 9.00 AM to 01.00 PM and fixing the number of people in the building to *200* and *d=35*. The results of this experiment are reported in Figure 4.(d). We observe that, at 01.00 PM, the set of people who could have accessed a zone with no bias (*PB=0*) is about *47%*, whereas when the zone access frequency is higher (*PB=0.3*), the set of candidates coincides with the whole population at about 12.00 AM. Thus, given an acceptable uncertainty level, this experiment allows us to estimate at which time this level of uncertainty is reached, thus allowing to perform suitable countermeasures. For example, to limit to *15%* the possible suspects of a malicious action with PB=0, we should schedule an inspection per hour to detect damages.

RELATED WORK

The general idea of k-anonymity location may be implemented by means of a trusted third-party. [6,7] The performance of these approaches strongly depends on the user density, so that many users have to use this service, otherwise the user's locations are given with a too high approximation in order to guarantee their privacy. Moreover, such approaches are ineffective when the number of malicious users is not negligible. Our proposal does not suffer from these issues: indeed, RFID readers independently generate quasi-IDs. We remark that the distributed fashion of our methodology is the basis of its effectiveness from the point of view of privacy. Indeed, any centralized solution would not be able to protect data from attacks in which we assume that the adversary accesses the server.

In the approaches based on location cloaking [6–7] location data are perturbed by introducing random noise. However, a spectral filter exploiting the theoretical properties of random matrices can be used to estimate the actual positions.

The idea of deleting identifiers from location data is not effective and several techniques [8–9] have been presented to guess the home location and the identity of a person. However, a spectral filter exploiting the theoretical properties of random matrices [10] can be used to estimate the actual positions.

In Location-Based-Service applications, GPS is used to continuously track the node's location. [11–13] Differently, in our proposal the use of RFID does not allow continuous tracking.

Some proposals use encryption to address the privacy issues in the data collection phase. [16-18] In some sense our approach is more general than an encryption-based approach. Basically, what we do is to implement a random permutation function with compression in such a way that k actual IDs are mapped to 1 virtual ID. Without the knowledge of the seed of the PRNG, the probability to guess the mapping between an actual ID and a virtual ID is $1/n$ (if n is the number of actual IDs), while, by knowing the seed, this probability increases to $1/k$. If we encrypt actual IDs, without the knowledge of the secret key, the probability to guess the mapping between an actual ID and a virtual ID is still $1/n$, and the probability to guess this mapping once the key is known is 1, obviously. Therefore, our method is equivalent to that based on encryption if k is set to 1, thus in the case we disable the feature of privacy against honest-but-curious providers.

In a recent paper [19] the authors propose a technique to localize patients of assisted living facilities by preserving their privacy. This solution addresses a problem different from the one dealt with in our case and is based on the use of active RFID.

Finally, we observe that a very preliminary version of the approach described in this paper has been presented. [20] In the current work, the original idea has been extended by a detailed and effective implementation of the approach, through a deep hardware-software-system design experience. Moreover, this implementation adds to the approach a number of security features relying on technological components to make the system resistant to misbehaving attacks (for example, decoupling of user and RFID tag positions). Also log analysis is enhanced and now it is presented in a complete and detailed way. Finally, the experimental analysis (only sketched in the original paper) has been deepened also by applying our approach to a real-life environment to effectively validate the proposal.

CONCLUSION

The main contribution of this paper concerns the solution of the trade-off between privacy and security/safety requirements, which is a relevant problem in the context of physical control of the territory. Importantly, the model is translated into a concrete system whose technological components play a significant role in guaranteeing the correct working and the security of the approach. As a consequence, the hardware-software-system design experience is inherently part of the novelty and significance of the proposal. Moreover, it demonstrates the technical and economic feasibility of the model, providing its validation by a number of experiments, which show that privacy and security requirements can be reached by suitably setting the system parameters and that the system appears robust against the possible attacks.

REFERENCES

1. L. Sweeney. k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(05):557–570, 2002.
2. Beagle Board Black Website. http://beagleboard.org/BLACK, 2017.
3. R. Durstenfeld. Algorithm 235: random permutation. Communications of the ACM, 7(7):420, 1964.
4. G. Ghinita, M. L. Damiani, C. Silvestri, and E. Bertino. Preventing velocity-based linkage attacks in location-aware applications. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 246–255. ACM, 2009.
5. F. Bai and A. Helmy. A survey of mobility models. Wireless Adhoc Networks. University of Southern California, USA, 206, 2004.
6. M. Wernke, P. Skvortsov, F. Durr, and K. Rothermel. A classification of location privacy attacks and approaches. Personal and ubiquitous computing, 18(1):163–175, 2014.
7. Y. Ye, C. Pan, and G. Yang. An improved location-based service authentication algorithm with personalized k-anonymity. In China Satellite Navigation Conference (CSNC) 2016 Proceedings: Volume I, pages 257–266. Springer, 2016.
8. Y. Xiao, L. Xiong, S. Zhang, and Y. Cao. Loclok: location cloaking with differential privacy via hidden markov model. Proceedings of the VLDB Endowment, 10(12):1901–1904, 2017.
9. G. Yang and Y. Cai. Full location privacy protection through restricted space cloaking. Journal of Information Processing, 25:756–765, 2017.
10. H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. Random-data perturbation techniques and privacy-preserving data mining. Knowl. Inf. Syst., 7(4):387–414, 2005.
11. C. Lin, G. Wu, and C. Wu Yu. Protecting location privacy and query privacy: a combined clustering approach. Concurrency and Computation: Practice and Experience, 27(12):3021–3043, 2015.
12. T. Hashem and L. Kulik. “Don’t trust anyone”: Privacy protection for location-based services. Pervasive and Mobile Computing, 7(1):44–59, 2011.
13. X. Li, E. Wang, W. Yang, and J. Ma. DALP: A demand-aware location privacy protection scheme in continuous location-based services. Concurrency and Computation: Practice and Experience, 2015.
14. L. Yao, L. Kang, F. Deng, J. Deng, and G. Wu. Protecting source–location privacy based on multirings in wireless sensor networks. Concurrency and Computation: Practice and Experience, 27(15):3863–3876, 2015.
15. J.N. Moutinho, R.E. Araùjo, and D. Freitas. Indoor localization with audible sound-Towards practical implementation. Pervasive and Mobile Computing, 2015.
16. X. Yi, A. Bouguettaya, D. Georgakopoulos, A. Song, and J. Willemson. Privacy protection for wireless medical sensor data. IEEE transactions on dependable and secure computing, 13(3):369–380, 2016.
17. T. Borgohain, U. Kumar, and S. Sanyal. Survey of security and privacy issues of internet of things. arXiv preprint arXiv:1501.02211, 2015.
18. R. L. Lagendijk, Z. Erkin, and M. Barni. Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation. IEEE Signal Processing Magazine, 30(1):82–105, 2013.
19. F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera. A privacy-preserving localization service for assisted living facilities. IEEE Transactions on Services Computing, 2016.
20. F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera. A Privacy-Preserving Solution for Tracking People in Critical Environments. In Proc. of the International Workshop on Computers, Software & Applications (COMPSAC’14), pages 146–151. IEEE Computer Society.