future internet

MDPI

Article

An Adaptive Privacy Protection Method for Smart Home Environments Using Supervised Learning

Jingsha He [1,2,*], Qi Xiao [1], Peng He [2,*] and Muhammad Salman Pathan [1]

[1] Faculty of Information Technology & Beijing Engineering Research Center for IoT Software and Systems, Beijing University of Technology, Beijing 100124, China; xqnssa@163.com (Q.X.); salman_htbk@hotmail.com (M.S.P.)

[2] College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China

* Correspondence: jhe@bjut.edu.cn (J.H.); hpeng@ctgu.edu.cn (P.H.); Tel.: +86-135-0102-9135 (J.H.); +86-139-7200-2550 (P.H.)

Academic Editor: Georgios Kambourakis
Received: 23 December 2016; Accepted: 1 March 2017; Published: 5 March 2017

Abstract: In recent years, smart home technologies have started to be widely used, bringing a great deal of convenience to people's daily lives. At the same time, privacy issues have become particularly prominent. Traditional encryption methods can no longer meet the needs of privacy protection in smart home applications, since attacks can be launched even without the need for access to the cipher. Rather, attacks can be successfully realized through analyzing the frequency of radio signals, as well as the timestamp series, so that the daily activities of the residents in the smart home can be learnt. Such types of attacks can achieve a very high success rate, making them a great threat to users' privacy. In this paper, we propose an adaptive method based on sample data analysis and supervised learning (SDASL), to hide the patterns of daily routines of residents that would adapt to dynamically changing network loads. Compared to some existing solutions, our proposed method exhibits advantages such as low energy consumption, low latency, strong adaptability, and effective privacy protection.

Keywords: smart home; privacy; FATS attack; supervised learning

1. Introduction

The most profound technologies are those that eventually disappear. They weave themselves into the fabric of every day life, until they become indistinguishable. Wireless sensors are becoming ubiquitous in smart home applications and residential environments. Smart home applications integrate multiple Internet of Things (IoT) devices and services that store, process, and exchange data. Micro-controllers can be used to analyze the status of the sensors for identifying events or the activities of residents. They then respond to these events and activities by controlling certain mechanisms that are built within the home. A simple example of such a smart behavior is to turn on the lights when a person enters a room [1]. This can be realized by triggering an infrared sensor when the person enters the room, and the micro-controller can combine the activity and the brightness of the room, to determine whether the lights should be turned on.

There will be a multitude of devices in a wireless sensor network (WSN) around residential environments. These monitoring devices can be classified into three categories: sensors, physiological devices, and multimedia devices. Sensors are used to measure the environmental parameters. Physiological devices monitor health conditions and vital signs. Multimedia devices capture audiovisual information and provide an interface between the system and the user [2].

A sensor in a residence is a simple autonomous host device that can sense a phenomenon, convert signals into data, process the data, and then transmit it to a sink node for further analysis [3]. However,

the societal concerns of smart home technology evolution, in relation to the privacy and security of the citizen, appear to be at an embryonic stage [4]. Although smart home technologies can bring a great deal of convenience to residents, it is also possible that people's daily behaviors in such an environment would become exposed to attackers, who can use the smart home in a malicious way. Therefore, the issue of privacy protection in the smart home environment has become one of the most important challenges.

In the smart home scenario, almost all of the sensor nodes only transmit information when a related event is detected, which is called event-triggered transmission. A global adversary has the ability to monitor the traffic of the entire sensor network, and thus, can immediately detect the origin and time of event-triggered transmissions. Although encryption algorithms can be used to protect data in the transmission, the emergence of new types of attack methods can make such traditional approaches invalid. Such attacks only need access to the timestamp and fingerprint data of each radio message, and in a wireless environment, the fingerprint is a set of features of a radio frequency waveform that are unique to a particular transmitter. Thus, the primary attacks that we are concerned with here are the Fingerprint And Timing-based Snooping (FATS) attacks [5], which have been shown to be very effective in inferring the Activity of Daily Livings (ADLs) of the residents.

The most simple and effective way of resisting FATS attacks is to inject fake messages into the transmission sequence. There have been extensive studies on the protection of the privacy of residents in a smart home environment, by taking into consideration the limitations of communication bandwidth, battery energy, and computing power. Most solutions proposed so far are based on a fixed frequency or probabilistic model, which make it hard to identify the real messages in the sequence of messages, even if the attacker can access the global information. These solutions have a major drawback, however, i.e., the reporting of a real event could be delayed until the next scheduled transmission. The delay of sensed data can cause degradation of the Quality of Service (QoS) in many applications, especially in those with intelligent sensing, where sensor data need to be obtained in real time in order to make decisions. To address this problem of delay, Park et al. proposed a method based on behavioral semantics [6]. However, this method would depend on the accuracy of the prediction, meaning that, if the prediction of the next message was not accurate, the added fake messages would not be enough to disturb the statistical analysis, and the ADLs of the residents would still be exposed. In this paper, we propose a method to protect against FATS attacks. The method is based on sample data analysis and supervised learning, which can adapt to network loads, as well as to the common living habits derived from the real data.

The remainder of this paper is organized as follows. In Section 2, we describe the FATS attack model and introduce some existing solutions. In Section 3, we make some assumptions about the network environment and the adversary, and describe the requirements of the privacy protection method. In Section 4, we describe our method in detail. In Section 5, we compare our method to some existing methods, to demonstrate the advantages of our method. Finally, in Section 6, we conclude this paper and also describe our future work. At the end of the paper, we list all of the acronyms used throughout this paper as the appendix.

2. Related Work

In this section, we will first introduce the FATS attack model and then briefly describe some of the existing solutions for resisting FATS attacks. We will also analyze the deficiencies of the existing solutions.

2.1. The FATS Attack Model

The FATS attack model focuses on collecting the fingerprints and timestamps to gain access to the behavior of the residents, even if the sensor data is protected by using a sufficiently secure and reliable encryption method. The attack model is shown in Figure 1 and the four tiers of attacks are described below:

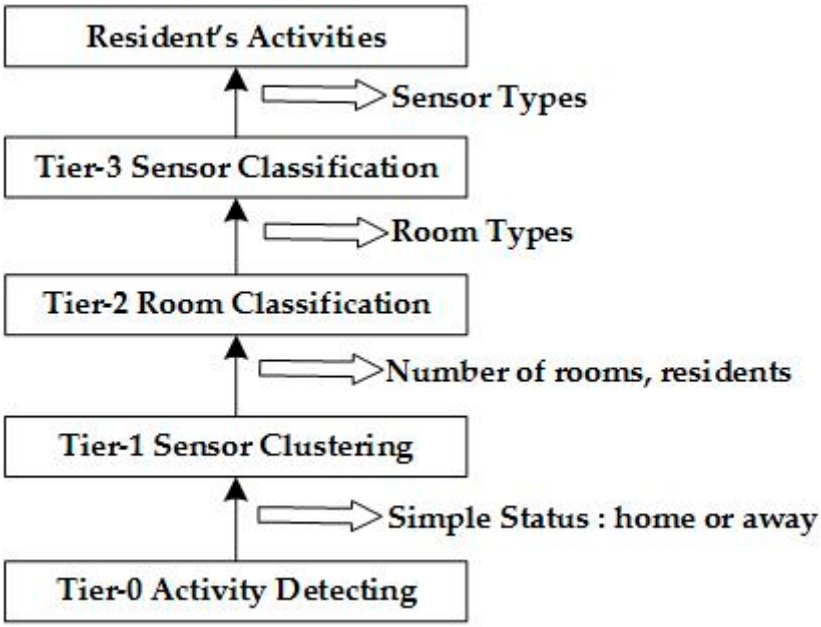


Figure 1. Four tiers in the FATS (Fingerprint And Timing-based Snooping) attack model.

Tier-0: General Activity Detection. The adversary can only detect very general activities, such as home occupancy or sleeping.

Tier-1: Sensor Clustering. It assumes that a particular sensor that is triggered during a timestamp will be very close in space to infer other sensors in the same room. In this tier, the number of rooms and people in the home can be predicted.

Tier-2: Room Classification. The main goal in this tier is to identify the features of the rooms via an analysis of the previous cluster in Tier-1. At this tier, the attacker can ascertain the layout of the house and can predict the residents in the rooms. The privacy of the residents can be infiltrated by the attacker.

Tier-3: Sensor Classification. In this tier, the goal is to identify the activities in the home, such as cooking, showering, and so on. The attacker can calculate a feature vector for each sensor from the answers in Tier-2 and can then classify each sensor by using Linear Discriminant Analysis (LDA).

Through analyzing the above tiers, residents' behavior can be exposed. First, a feature vector of every temporal activity cluster in every device can be calculated. Then, by importing the vector to the LDA classifier that was trained for other homes, the hand-labeled activity labels can be used to distinguish real activities. This approach can recognize many daily activities, including showering, washing, grooming, cooking, etc. [4].

2.2. Methods to Resist the FATS Attack

In the ConstRate (sending messages in constant rate) model, all of the sensor nodes in the network maintain the same frequency when sending messages, whether or not actual events happen. When a real event occurs, it has to wait until the next transmission. Therefore, the model can effectively resist the static analysis by the attacker with a global listening ability. This model also has a congenital deficiency, i.e., the average delay is half of the transmission interval and it is difficult to determine the transmission interval. If the transmission rate is low, the delay will be very high; whereas, if the rate is high, the delay will decline, but the number of fake messages will increase significantly, resulting in an increased energy consumption.

Yang et al. proposed a probability-based model called the FitProbRate model, that aims at reducing the latency of a fixed frequency transmission [7]. The main idea is that every sensor in the network sends messages with an interval that follows the exponential distribution. When a node

detects a real event, the algorithm needs to identify a minimum interval to obey the exponential distribution, and then waits to send the real event. Following this, all adjacent intervals will follow the same distribution, making the attacker unable to determine the real messages from the transmission sequence. This model can reduce the delay of transmission in some situations, e.g., when the time interval is relatively average and the sending interval is slightly longer. If the time interval is not uniform and is frequently triggered with small intervals, the delay will become high.

Park et al. proposed a behavioral semantics model to generate a small amount of fake data, to protect the activity that will happen in the near future [7]. Firstly, the model predicts the activity from a long-term history and then presents the sensor nodes with the forecast of the activity sequence. If an attacker listens to a sequence in order to monitor the sensors, the attacker can only predict misbehavior. This model adds fake messages to disturb the FATS attack in Tier-3 and privacy protection depends on the accuracy of the prediction of future behaviors. Therefore, if the prediction is not accurate, the added fake messages will not make much difference. Generally, the reliability of this scheme is lower than the ConstRate and FitProbRate models.

2.3. Summary of the Related Work

2.3.1. The intervals of the Send Sequence

In the ConstRate and FitProbRate models, the interval between the fake messages and the real messages is subject to the same distribution. The purpose of adding fake messages is to prevent the attacker from distinguishing between real and fake messages. If an attacker is not able to recognize the real messages from the message sequence sent by the sensor node, the purpose of adding noise is successfully achieved. Consequently, assuming that an adversary monitors the network over multiple time intervals, in which some intervals contain real event transmissions and some do not, if the adversary is unable to distinguish between the intervals with significant confidence, the real event will be hidden [8]. If the sensor nodes have a sufficient randomness to send fake and real messages, it makes it less likely that the adversary will be able to recognize the fake data from the transmission sequence, resulting in the analysis of the wrong ADL, and thus, the protection of the privacy of the residents.

2.3.2. The Traffic of the Whole Network

To reduce the delay of data transmission, the transmission interval should be reduced, resulting in a high traffic load, as well as a significant increase in the probability of collision. As Figure 2 shows, in an actual sensor network, the closer it gets to the sink node, the larger the amount of data that needs to be forwarded. If all of the sensor nodes send fake messages following the same model, the nodes that are closest to the sink node will have to assume a load which is too heavy to forward messages. In our method, a sensor node will adapt to its own network status when sending fake messages, i.e., a node will add a lower number of fake messages when the forwarding load appears to be heavy. Such an approach would result in the three types of nodes consuming energy at a more balanced level, prolonging the life of the network in comparison to other models.

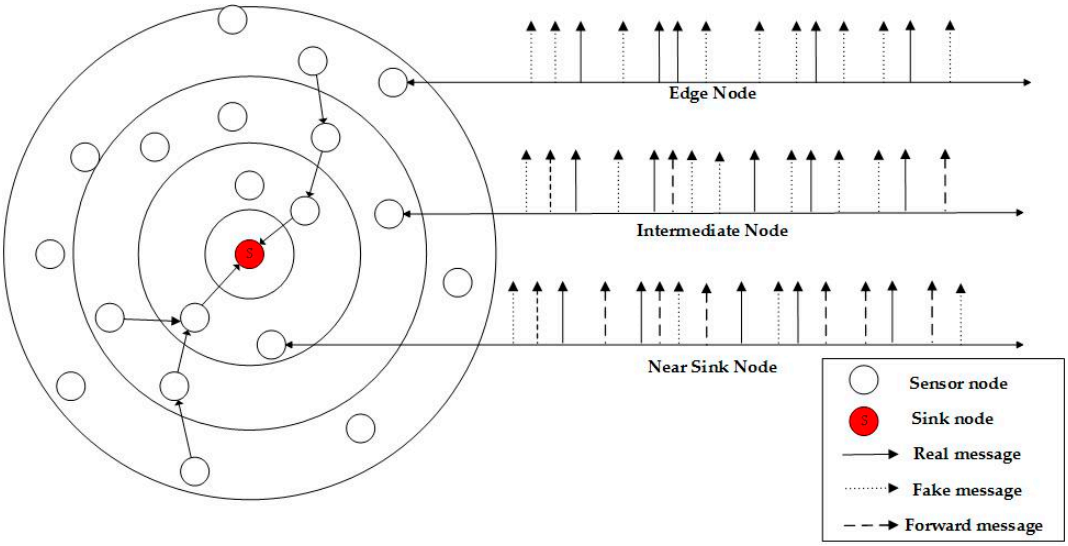


Figure 2. Traffic status of the sensors.

2.3.3. The Particularity of Smart Home Environment

In the smart home environment, when people go to sleep or go out to work, there are very few real events triggered during the corresponding time slots. It seems that there is no need to add fake messages during such periods, due to few regular activities in the smart home. Accordingly, the ideal situation should be that the sensor nodes only send fake messages when the residents are present, with some daily activities taking place. Thus, the number of fake messages can be reduced. However, the adversary can learn the routines of individual residents through analyzing the sending patterns of the sensor nodes. Following the ideas of k-anonymity, we will thoroughly analyze the data to obtain the routines of the public, so that message sending will be carried out in such a way that it would be very difficult for the attacker to detect the routines of individual residents.

3. Assumptions and Requirements

In this section, we describe the network model and some assumptions about the adversary in SDASL, as well as the requirements of privacy protection that will guide the design of our privacy protection method.

3.1. The Network Model

Similar to other WSNs [9], nodes in smart home consist of the sink node (only one sink node in the sensor network) and sensor nodes $N = \{N_1, N_2, N_3, N_4 \ldots N_n\}$. As shown in Figure 3, the smart home provider is a reliable service provider who can collect and analyze data from many homes. The sink node has a high enough computing power to take on complicated operations. The sensor node is the smallest unit in the WSN, and has a limited computing power and limited battery capacity. Sensor nodes can apply encryption algorithms to encrypt collected data [10], before sending it to the sink node. We assume that the encryption algorithms used are safe enough and that the attacker cannot acquire the original data through analyzing the cipher texts. We also assume that the sink node is actively powered, while being equipped with tamper-resistant hardware [11]. Consequently, it is reasonable to assume that the adversary cannot compromise the sink node.

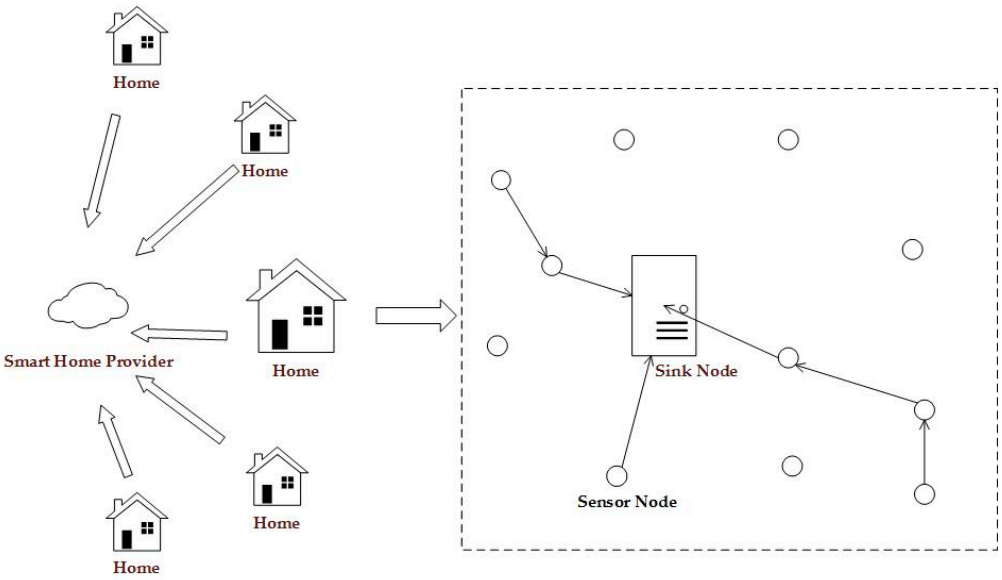


Figure 3. The framework of a smart home.

We assume that the dataset collected at the sink node is $D = \{X, y\}$ and $X = \{X_1, X_2, X_3 \cdots, X_n\}$, where X_i refers to the set of sensor data from sensor node N_i. A datum of X_i is denoted as $X_i^{(j)} = \{x_1, x_2, x_3, x_4\}$, in which x_1 is the sensor data, x_2 is the traffic status, x_3 is the timestamp of transmitting the sensor data, and x_4 is the flag to indicate whether the message is real or fake. Label y in D can only be a binary value {0,1}. The minimum unit of the collected datasets is $\left\{X_j^{(i)}, y^{(i)}\right\}$, which means that the i^{th} sensor data comes from sensor node N_j with label $y^{(i)}$.

3.2. The Adversary Model

The global adversarial model used in this paper is similar to the one that is considered as external, passive, and global [12,13]. In contrast to the passive adversary, an external adversary cannot control any nodes in the WSN. Instead, it can only monitor, eavesdrop, and analyze the communication in the network, via channel snooping. After obtaining the transmission status, the global adversary can identify the behavior of people in the smart home by applying static analysis, using a method such as the FATS Model.

3.3. Requirements of Privacy Protection

The particularity and sensitivity of the smart home makes it important to consider the privacy, energy efficiency, and latency of the sensor network in the design of privacy protection methods.

3.3.1. Privacy

The WSN in the smart home environment collects sensitive data about people living in the environment. A good privacy protection model should be robust to resist such attacks as FATS, as well as statistical analysis that can acquire private activities. The main purpose is that, even if an attacker can listen to a global message sequence along with the time of transmission, it is still not possible for the attacker to identify fake messages from the transmission sequence, i.e., the added fake messages would prevent the attacker from obtaining the desired results.

3.3.2. Energy Efficiency

It is not considered a good privacy protection scheme if the implementation of the scheme would reduce the lifetime of the entire WSN. A good protection scheme should keep the overhead as low as

possible, to extend the life of the WSN as much as possible. However, adding fake messages can incur extra energy consumption. In our method, we take the average load of the traffic into consideration. When the traffic load is high, the algorithm reduces the number and frequency of the fake messages accordingly. Packet loss and collision are also reduced as a result.

3.3.3. Low Latency

As the aging of the population becomes a serious issue, the smart home would make a great contribution to improving the quality of life of the elderly [14]. In such applications, it is necessary to trigger events in a timely manner, to immediately send the sensed data to the sink node, to determine or predict the abnormal situation (especially physical health status) of the elderly, and to take appropriate actions. If the latency was too long, it would take a lot of time for the sensed data to be received, lowering the efficiency of the timely treatment of elderly people. A good privacy protection model should keep the latency within the normal range of acceptance, allowing the sink node to make timely decisions, in order to meet the requirements of the applications.

4. The Proposed Model

In this section, we introduce our adaptive method, which is based on supervised learning. The proposed method consists of three separate algorithms. Algorithm 1 is designed for sample data analysis. The smart home provider uses the algorithm to analyze the sample dataset generated in real smart home scenarios. Through the analysis, we can use the frequency distribution of the RF radio (FDR) to simulate the distribution of fake messages, ensuring that the frequency rate is similar to that in the sample datasets. Algorithm 2 is designed for supervised learning. Firstly, we analyze the load and time characteristics of the collected sensor data and label the results of Algorithm 1 accordingly. Then, we use supervised learning to generate the parameters of the prediction model. Finally, the sink node sends the parameters to all of the sensor nodes. Algorithm 3 is designed to allow the sensor nodes to update the parameters and to send fake messages. In the rest of this section, we will describe the three algorithms in detail.

4.1. Sample Data Analysis

In the smart home environment, the likely categories of privacy protection include scenarios of the user's going to work, coming back home, or going to sleep, etc. If fake data messages are only sent when the user is at home, to protect the user's behavior, then the behavior during the time at work, sleep, and other privacy-related periods, can still be leaked. We therefore design an algorithm that produces a transmission sequence which resembles that of a large amount of real data, to achieve better privacy protection. The adversary can only attain a general pattern of the people through monitoring the frequency of radio signals, thus protecting the privacy of individual residents. Consequently, we use FDR to describe the send frequency of the sensor network, which consists of one or more elements in the form [(start time, end time), average frequency].

Taking into account the different habits among people in different regions, the provider should include sample data from different regions in the analysis, and the procedure is shown in Figure 4. Firstly, we should count the number, as well as the time of message sending per minute, for which the answer is denoted as $F = \{(t_1, f_1), (t_2, f_2), (t_1, f_3) \ldots, (t_n, f_n)\}$. Then, we will use formula (1) to standardize f, for which the answer is denoted as $F^{\wedge\prime} = \{(t_1, f_1'), (t_2, f_2'), \ldots, (t_n, f_n'))\}$. Secondly, we use the K-means clustering algorithm to cluster the elements in F', producing a group of periods over the 24-hour interval. Finally, we calculate the average frequency of sending in each of the periods and update the average frequency of sending, as well as the period p to the FDR. After the analysis, we will ascertain the FDR that will be sent to the smart home devices in every family.

$$f_i' = \frac{f_i - \min(F)}{\max(F) - \min(F)} \quad (1)$$

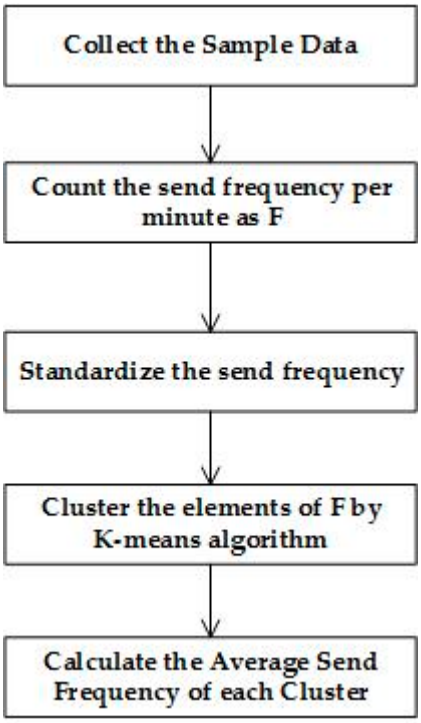


Figure 4. Analysis of sample datasets.

4.2. Supervised Learning

The supervised learning process is composed of three steps: collecting data, labeling collected data, and executing the learning algorithm and updating the parameters in the sensor nodes. All of the sensor nodes send fake messages in accordance with the initial time window, and the sink node labels the collected data in accordance with the FDR, as well as the traffic status. The learning algorithm will train the labeled data from different sensor nodes and the sink node will hand out the parameters to the sensor nodes.

4.2.1. Data Collection

Each sensor node works in accordance with Algorithm 1 of Figure 5 to generate information and decides whether or not to send fake messages. At the beginning of the algorithm, all of the sensor nodes in the WSN will send fake messages using a fixed time window. After training the learning algorithm, the prediction model will be plugged with a new θ, as shown in Figure 5. Each sensor node needs two variables to input into the prediction model, in order to determine whether or not to send fake data. The two variables are the traffic status and time, represented by x_1 and x_2, respectively. Traffic status is calculated using Formula (2), where ft represents the number of messages forwarded and st represents the number of messages sent within the time window.

$$x_1 = \frac{ft + st}{st}; \qquad x_2 = \frac{\text{nowTimeStamp} - \text{dateStamp}}{60 * 60 * 24} \tag{2}$$

To normalize the time, we map the current timestamp in the range [0,24], which is represented using x_2. In x_2, the nowTimeStamp denotes the current timestamp and the dateStamp denotes the starting time of the day. These parameters are used as inputs for the prediction model. The prediction model consists of a series of operations, along with a hypothesis function. If the result of the prediction model is not smaller than 0.5, a fake massage should be sent after a random time delay. If the result is smaller than 0.5, the algorithm ascertain whether a fake message has been sent during the previous K

iterations. If so, the time window will be doubled. Once real data is sent, the original time window will be resumed.

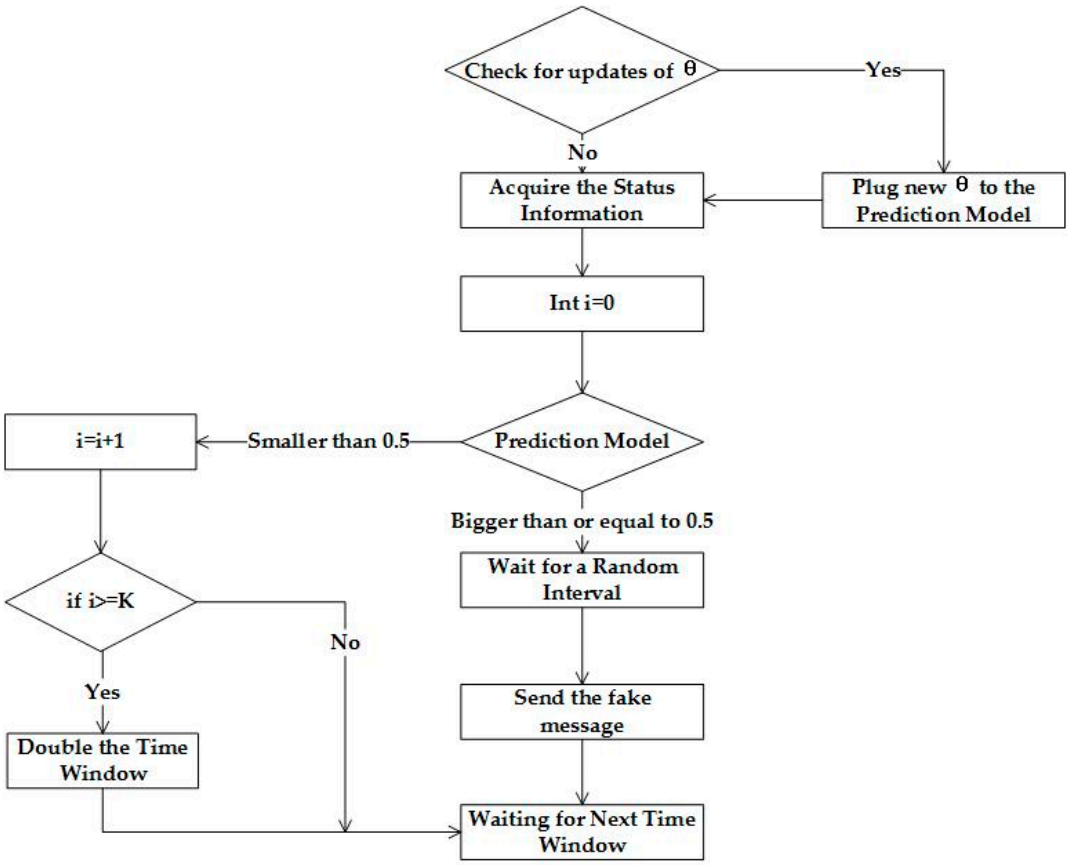


Figure 5. Sensor node determines whether or not to send fake messages.

4.2.2. Sensor Data Labeling

The sink node would classify received data into two categories: fake data and real data, which are marked using a flag. The main purpose of labeling is to mark the fake data with label 1, meaning that fake messages should be sent, and to use label 0 otherwise. The basis of the classification is the FDR and the traffic status. Firstly, we label fake messages with 0 during the period of sending messages sparsely, and with 1 during the period of sending messages at a higher rate. After labeling the fake messages, supervised learning will take place.

4.2.3. Learning and Parameter Updating

Supervised learning is one type of learning method, in which a model learns from the training data and then predicts new instances of an event. It results in unlabeled data to be labeled through previous experience and then applies the labeled data to the learning algorithm. After training, the parameters are generated to update the prediction model. Song et al. presented a solution for supervised learning [15] and our algorithm has been inspired by this solution.

Algorithm 2 is shown in Figure 6. The algorithm inputs the datasets of sensor data and the FDR. The learning algorithm will deal with all of the data and calculate θ for each sensor node, respectively.

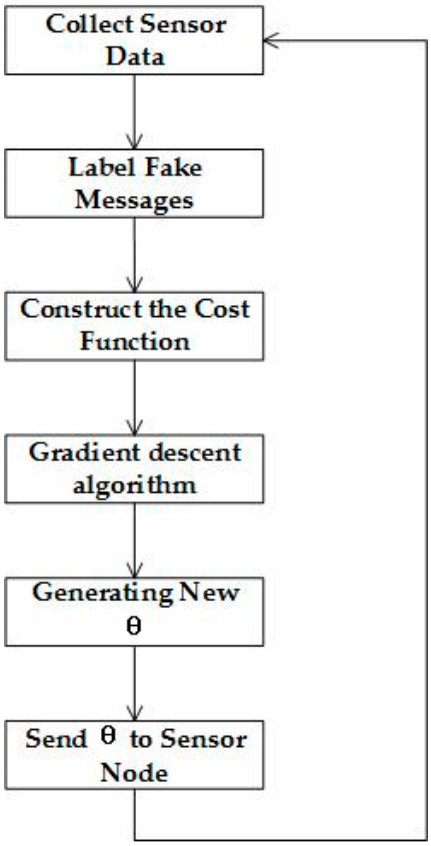


Figure 6. Supervised learning running in the sink node.

We use the logistic function as the hypothesis function, as expressed in Formula (3), where $\theta = [\theta_0, \theta_1, \theta_2 \cdots, \theta_w]^T$. In the formula, $h_\theta(x)$ is a $w+1$ dimensional parameter vector that we are learning from the training set, w is the number of features for the training, and θ_0 is the initial value for logistics regression. The learning algorithm and the hypothesis function can realize the goal of logistics regression.

$$h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)} \tag{3}$$

The main purpose of the learning algorithm is to find θ, to accurately classify the training data into two categories. When the cost function in Formula (4) reaches the global minimum, θ is the optimal solution of the learning algorithm. In the formula, m is the amount of training data. In the proposed SDASL method, every sensor node has its own training dataset, and thus, each sensor node has its own parameters.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} \cdot log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))] \tag{4}$$

The gradient descent algorithm is used here to obtain an appropriate θ for the hypothesis function which is described in Formula (5), in which α indicates the length of the gradient descent. Because $J(\theta)$ is a convex function, we can be sure of finding the local optimal value, which is also the global optimum.

$$\theta_i := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta) \tag{5}$$

5. Evaluation

In this section, we first describe the method and the setup of the experiment, and then present the results of the experiment in terms of privacy protection, delay, and energy consumption, in comparison to the ConstRate and the FitProbRate models.

5.1. Experiment Setup

We used the public dataset related to accurate activity recognition in a home setting [16] in the experiment. We also downloaded several datasets from WSU CASAS (Center for Advanced Studies in Adaptive Systems) as the sample datasets, to calculate the FDR for supervised learning in Algorithm 2.

The experiment was carried out in a macOS environment and a PHP language was used to process the original datasets, that are in the formats of txt or dat. To use the datasets conveniently, we used PHP to convert the datasets into a uniform format and stored the results in MySQL. The details of the datasets are listed in Table 1, which includes the sensor number, trigger time, sensor status, and real event. For example, M17 was triggered at 12:49:52 and the event is wash_hands begin. M17 was triggered again at 12:50:42 and the event is wash_hands end.

Table 1. Examples of the datasets.

ID	Sensor Number	Trigger Time	Sensor Status	Real Event
1	M17	2008-02-27 12:49:52	ON	Wash_hands begin
2	M16	2008-02-27 12:49:54	ON	-
3	AD1-B	2008-02-27 12:50:01	0.302934	-
4	M17	2008-02-27 12:50:42	OFF	Wash_hands end
5	M19	2008-02-27 12:51:01	ON	Cook begin
...	...	...	...	...
6	T10	2008-02-27 13:07:14.	26.5 °C	-

In evaluating the effectiveness of privacy protection, we propose using ACA, the average clustering accuracy. This is because, at Tier-1, the FATS attack would try to cluster the sensor data, which plays a critical role in the attack. Should a sensor node be clustered into the wrong group, the predicted ADL would be inaccurate [17], resulting in an ACA which falls within the range [0, 1]. The closer ACA gets to 1, the closer the clustering results will be to equaling the number of rooms and the distribution of the sensor nodes. Conversely, when the clustering result is inconsistent with the actual sensor distribution in the rooms, the ADL will be adequately protected.

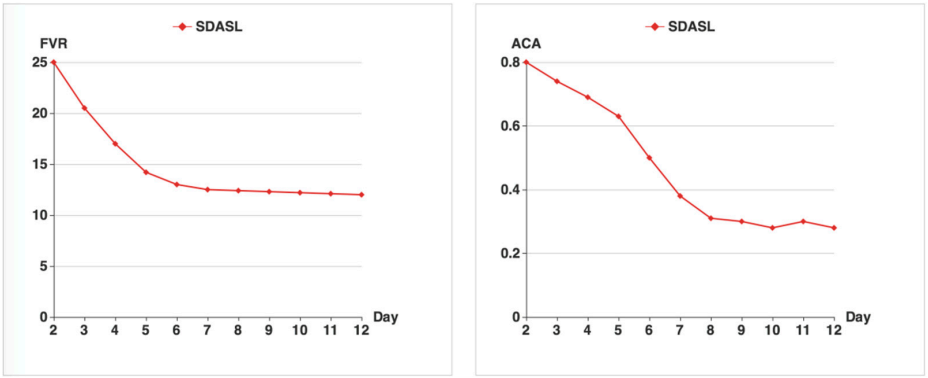
We also propose using FVR, which is the result of the total number of fake messages divided by the total number of real messages. Since it is hard to compare the three models under different conditions, we can use FVR to unify the main influencing factors. We compared the delay, energy consumption, and effectiveness of protection by the three models, through changing the FVR.

To compare the energy consumption between the three models, we compared the FVR under the condition that the same level of privacy protection is provided. Because all of the three models take a noise-based approach, the more noise in the WSN, the higher the energy consumption [18]. When considering the energy consumed by running the algorithm, it is generally recognized that energy consumption by wireless communication is much higher than the energy consumption by computation. It has been found that 3,000 instructions are needed for a sensor node to transmit 1 bit of data over a distance of 100 meters [19]. Therefore, we can ignore the energy consumption of the algorithm in our experiment.

In the evaluation of our model, the sample datasets were analyzed to derive the FDR and the time window. Then, through a number of iterations in supervised learning that would update θ, the transmission of real and fake data was collected by the sink node. Finally, we measured the FATS attack in Tier-1 to calculate the ACA.

5.2. Experiment Results

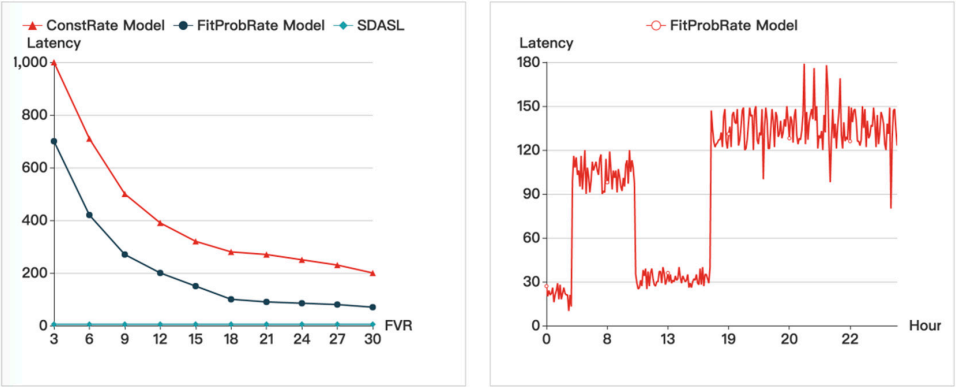
The results of the FVR are shown in Figure 7a. At the beginning, as the supervised learning is not performing, the FVR can achieve its maximum value, which is 25. As time goes by, the FVR gradually declines and eventually levels out at around 13. Figure 7b shows the results of ACA. As time goes by, the ACA decreases gradually, from 0.8 to 0.4, and then waves around 0.3. In the beginning, with the default parameters, the learning algorithm results in very poor privacy protection. Seven days later, the ACA arrives at a stably low level, indicating that good privacy protection has been achieved. As time goes by, the ACA gets lower, along with a continuous decrease in FVR thanks to the execution of the learning algorithm from the sink node and the sending algorithm from the sensor node.



(**a**) (**b**)

Figure 7. Relationship between FVR, ACA, and time in the SDASL model; (**a**) Description of the change of the FVR; (**b**) Description of the change of ACA.

Figure 8 shows that, with an increase in the FVR, the latencies of the ConstRate and the FitProbRate models gradually decrease. Since, in our method, real data is sent out without any delay, the latency is caused by the multi-hop forwarding time, which can be ignored, in contrast to the other two models. Figure 8 also provides the latency of a sensor node during a period of one day, which shows that the period from (0,6) to (9,17) has a lower latency due to a sparse transmission rate, and the period from (7,9) to (17,24) has a higher latency due to a dense transmission rate. This result indicates that the delay in the FitProbRate model is affected by the density of transmission, and as the transmission rate of the real event goes up, the delay will increase.



(**a**) (**b**)

Figure 8. Comparison of latency; (**a**) Description of the change of the Latency which with the change of the FVR; (**b**) Description of the change of Latency of FitProbRate Model which with the change of the hours in a day.

Figure 9 shows that, with an increase in FVR, the ACA gradually decreases. While in the ConstRate model, the ACA remains at a low level, it is affected by FVR in our model and in the FitProbRate model. When FVR increases, ACA declines. As in the FitProbRate model, if real events happen frequently, the overall transmission will become dense. Whereas, if real events happen sparsely, the sequence will become synchronized. Therefore, the adversary can infer the routine of the residents by analyzing the density of the transmission sequence. Accordingly, when the sequence is dense, people may be at home and pursuing activities. When the sequence is sparse, people may be asleep or at work. In our model and in the ConstRate model, this problem does not occur.

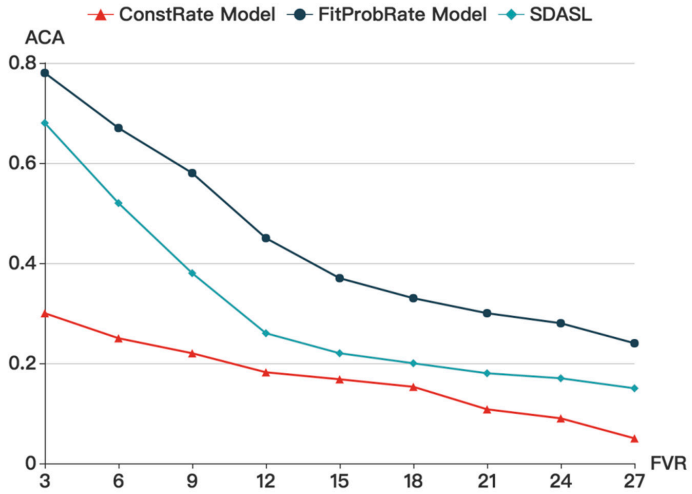


Figure 9. Comparison of ACA.

Figure 10 shows that with the increase in ACA, the FVR in all three models gradually decreases. Without considering the delay, the ConstRate model is the most energy-efficient model out of the three models. In the smart home, however, the delay must be controlled within an appropriate range. Compared to the FitProbRate and the SDASL models, for each ACA, the FVR in SDASL is much lower than that in FitProbRate, meaning that, under the same level of privacy protection, FitProbRate will incur 50% more noise data than SDASL. Since energy consumption mainly occurs during data transmission, SDASL will result in a 40% greater energy saving than FitProbRate, when the ACA is 0.2.

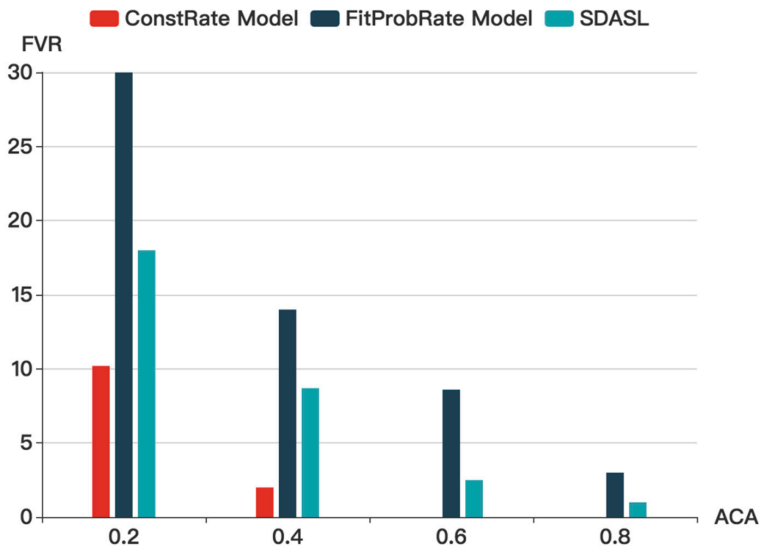


Figure 10. Comparison of energy consumption.

In summary, among the three models, ConstRate can provide the best privacy protection with an average latency and the longest delay. FitProbRate can reduce the delay, but a resident's routine may be revealed and the computation load in the sensor node would increase. Our SDASL model can overcome the shortcomings of the FitProbRate model, while reducing the latency.

6. Conclusions

In this paper, we proposed a new method to resist FATS attacks. Our method incorporates supervised learning to improve privacy protection, while analyzing sample data to provide the basis for data labeling. Compared to the ConstRate and the FitProbRate models, the experimental results clearly demonstrate the advantages of our SDASL model in terms of adaptiveness, low latency, and low power consumption, making it a better solution for smart home applications.

Following the approach in the proposed SDASL model, the collected data will be cached in the center of the smart home. When someone requests to access the data, the control module will decide whether or not to authorize the access. It may be true that the leakage of some insignificant information would not lead to the leakage of privacy. However, the attacker can still perform some analysis on such information, that could eventually lead to the leakage of privacy. In our future research, therefore, we will develop ways of resisting this type of attack, in order to improve privacy protection through access control.

Acknowledgments: The work in this paper has been supported by National High-tech R&D Program (863 Program) (2015AA017204).

Author Contributions: Jingsha He, ideas and general approach; Qi Xiao, model design, experiment and analysis; Peng He, general discussion and some contribution on experiment; Muhammad Salman Pathan, discussion and paper editing.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

To help readers understand the acronyms clearly, we provide the following Table A1 to list all of the acronyms used throughout this paper.

Table A1. List of Acronyms and Explanations.

The Acronym	Explanation
ACA	Average Clustering Accuracy
ADLs	Activity of Daily Livings
ConstRate	Model that sends fake messages in constant rate
FDR	Frequency Distribution of RF radio
FATS	Fingerprint and Timing-based Snooping
FitProbRate	Model that is based on probability and looks for fit time interval to send fake messages
FVR	FVR is calculated by dividing the number of fake messages by the number of real messages
IoT	Internet of Things
QoS	Quality of Service
SDASL	Model that is based on sample data analysis and supervised learning
WSN	Wireless Sensor Network
WSU CASCA	Washington State University Center for Advanced Studies in Adaptive Systems

References

1. De Silva, L.C.; Morikawa, C.; Petra, I.M. State of the art of smart homes. *Eng. Appl. Artif. Intell.* **2012**, *25*, 1313–1321. [CrossRef]
2. Alam, M.R.; Reaz, M.B.I.; Ali, M.A.M. A Review of Smart Homes-Past, Present, and Future. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 1190–1203. [CrossRef]
3. Ding, D.; Cooper, R.A.; Pasquina, P.F.; Fici-Pasquina, L. Sensor technology for smart homes. *Maturitas* **2011**, *69*, 131–136. [CrossRef] [PubMed]
4. Sanchez, I.; Satta, R.; Fovino, I.N.; Baldini, G. Privacy leakages in Smart Home wireless technologies. In Proceedings of the International Carnahan Conference on Security Technology, Rome, Italy, 1–6 May 2014.
5. Srinivasan, V.; Stankovic, J.; Whitehouse, K. Protecting your daily in-home activity information from a wireless snooping attack. In Proceedings of the 10th International Conference on Ubiquitous Computing, Seoul, Korea, 26 August 2008; pp. 202–211.
6. Park, H.; Basaran, C.; Park, T.; Sang, H.S. Energy-Efficient Privacy Protection for Smart Home Environments Using Behavioral Semantics. *Sensors* **2014**, *14*, 16235–16257. [CrossRef] [PubMed]
7. Yang, Y.; Shao, M.; Zhu, S.; Cao, G. Towards Statistically Strong Source Anonymity for Sensor Networks. *J. IEEE INFOCOM* **2008**, *9*, 51–55. [CrossRef]
8. Alomair, B.; Clark, A.; Cuellar, J.; Poovendran, R. Toward a Statistical Framework for Source Anonymity in Sensor Networks. *J. Mob. Comput.* **2011**, *12*, 1–6. [CrossRef]
9. Zhang, W.; Song, H.; Zhu, S.; Cao, G. Least privilege and privilege deprivation: Towards tolerating mobile sink compromises in wireless sensor networks. *Proc. ACM Mobihoc* **2005**, *4*, 2237–2258.
10. Klaoudatou, E.; Konstantinou, E.; Kambourakis, G. A Survey on Cluster-Based Group Key Agreement Protocols for WSNs. *Commun. Surv. Tutorials IEEE* **2011**, *13*, 429–442. [CrossRef]
11. Lin, X.; Lu, R.; Shen, X. MDPA: multidimensional privacy-preserving aggregation scheme for wireless sensor networks. *J. Wirel. Commun. Mob. Comput.* **2009**, *10*, 843–856. [CrossRef]
12. Chakravarty, S.; Portokalidis, G.; Polychronakis, M.; Keromytis, A.D. Detection and analysis of eavesdropping in anonymous communication networks. *J. Inf. Secur.* **2015**, *14*, 205–220. [CrossRef]
13. Mehta, K.; Liu, D.; Wright, M. Location Privacy in Sensor Networks Against a Global Eavesdropper. In Proceeding of the 20th IEEE International Conference on Network Protocols, Beijing, China, 16–19 October 2007; pp. 314–323.
14. Ghazal, B.; Al-Khatib, K. Smart Home Automation System for Elderly, and Handicapped People using XBee. *Int. J. Smart Home* **2015**, *9*, 203–210. [CrossRef]
15. Song, Y.; Cai, Q.; Nie, F.; Zhang, C. Semi-Supervised Additive Logistic Regression: A Gradient Descent Solution. *Tsinghua Sci. Technol.* **2007**, *12*, 638–646. [CrossRef]
16. Van Kasteren, T.; Noulas, A.; Englebienne, G.; Se, B. Accurate activity recognition in a home setting. In Proceedings of the International Conference on Ubiquitous Computing, Seoul, South Korea, 1–9 September 2008.
17. Park, H.; Park, T.; Son, S.H. A Comparative Study of Privacy Protection Methods for Smart Home Environments. *Int. J. Smart Home* **2013**, *7*, 85–94.
18. Siano, P.; Graditi, G.; Atrigna, M.; Piccolo, A. Designing and testing decision support and energy management systems for smart homes. *J. Ambient Intell. Humaniz. Comput.* **2013**, *4*, 651–661. [CrossRef]
19. Qian, T.F.; Zhou, J.; Wang, M.F.; Cao, C. Analysis of energy consumption in wireless sensor networks. *Comput. Tech. Appl. Prog.* **2007**, *3*, 1710–1714.